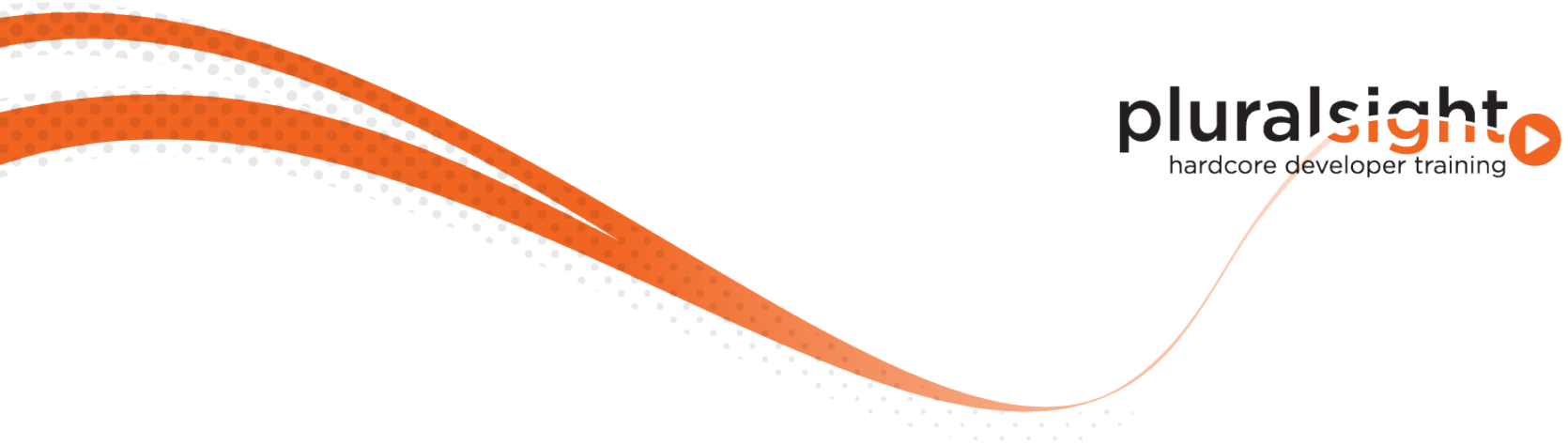
Elements of RESTful Architecture

Howard Dierking
http://codebetter.com/howarddierking



pluralsight
hardcore developer training

Overview

- Components and Connectors
- Resources
- Representations
- Control Data
- Hypermedia

Components

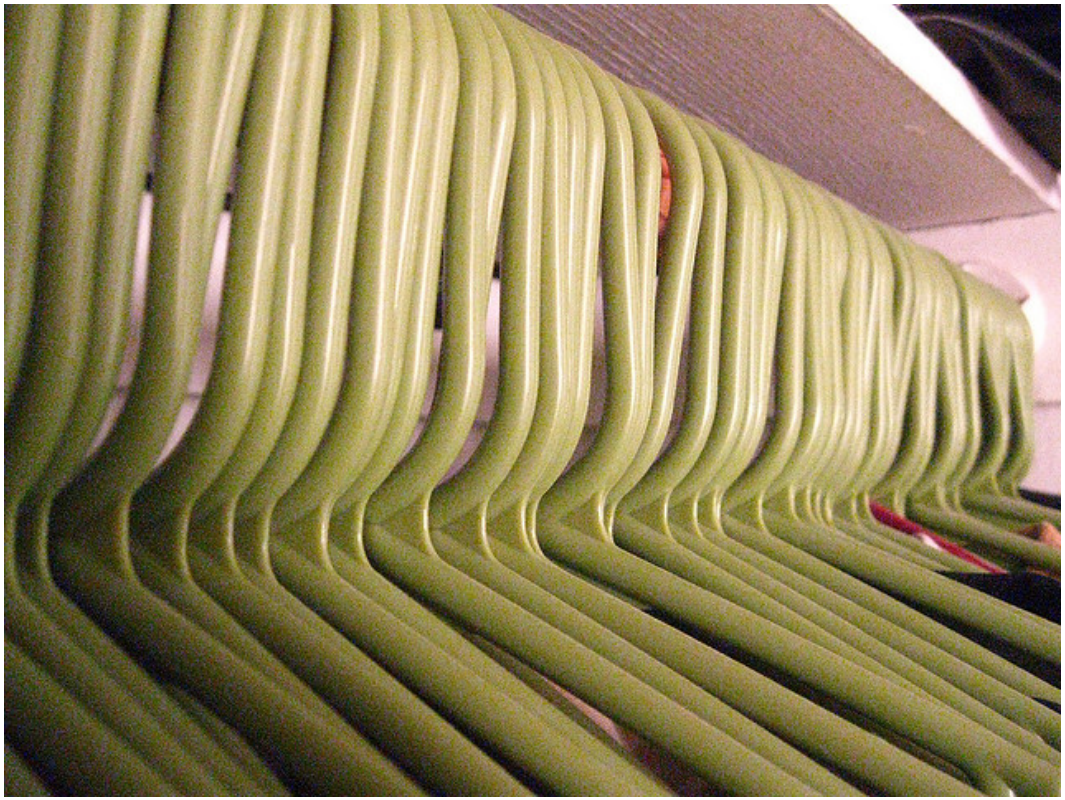
- **The processors of resource requests and representations in an application**
- **Categorized by role**
  - Origin server
  - User agent
  - Gateway
  - Proxy

Connectors

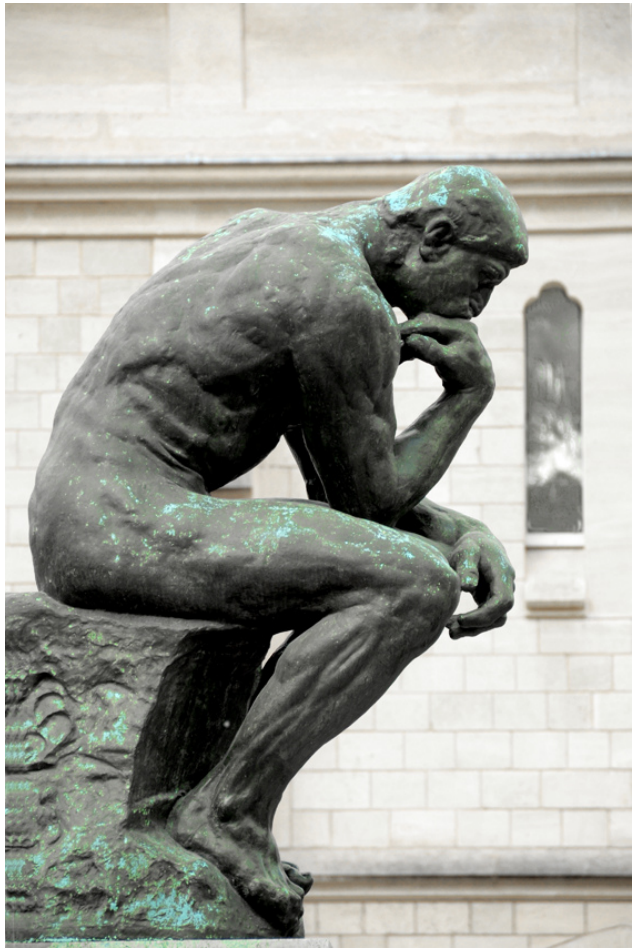
- **Represent the activities involved in accessing resources and transferring representations**
- **Roles provide an interface for components to implement**
- **Types**
  - Client
  - Server
  - Cache
  - Resolver
  - Tunnel

The Uniform Interface

- Identification of resources
- Manipulation through representations
- Self-descriptive messages
- Hypermedia as the engine of application state (HATEOAS)

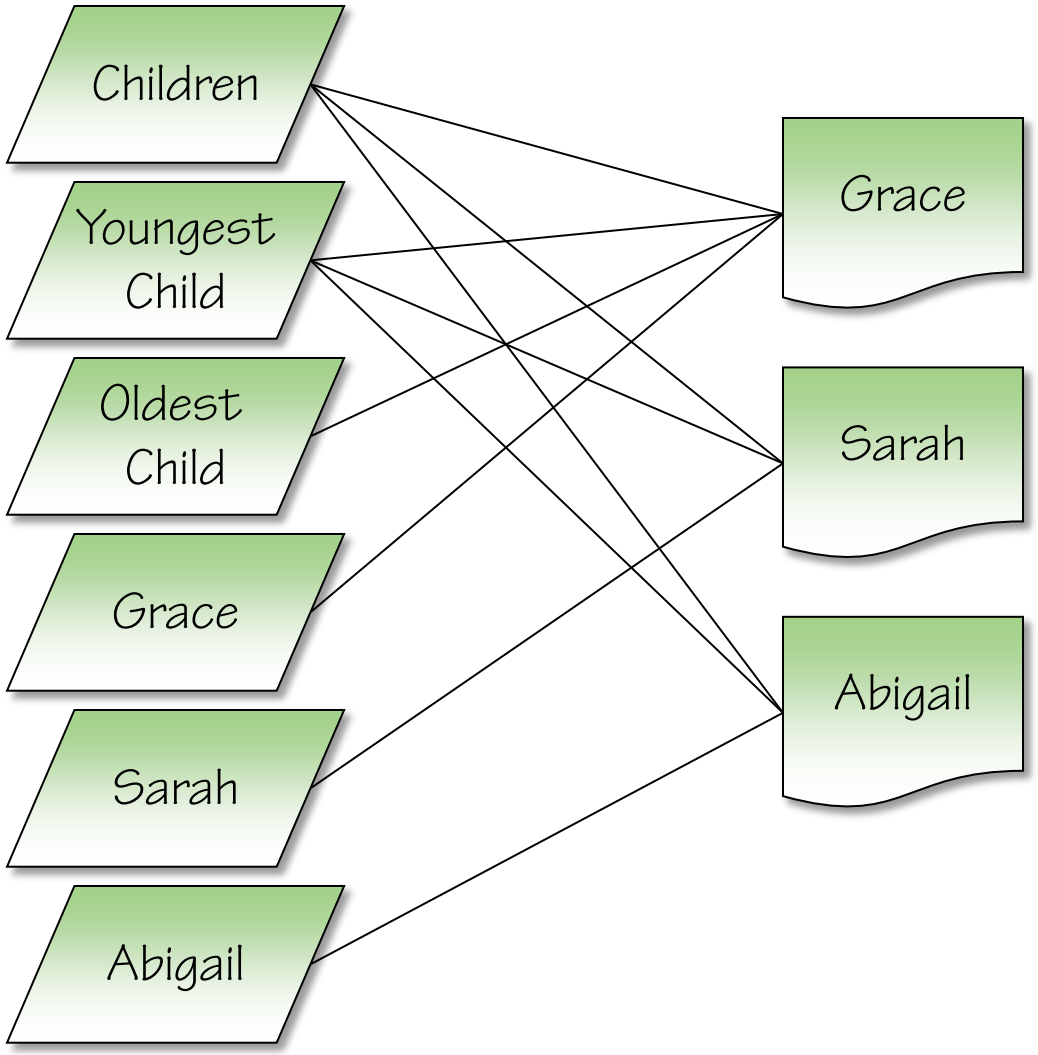


Resource



- Can be any named information or concept
- Maps a named concept to a set of entities over time
- Many to many relationship between named concepts and entities
- The relationship may be consistent over time, or it may change frequently

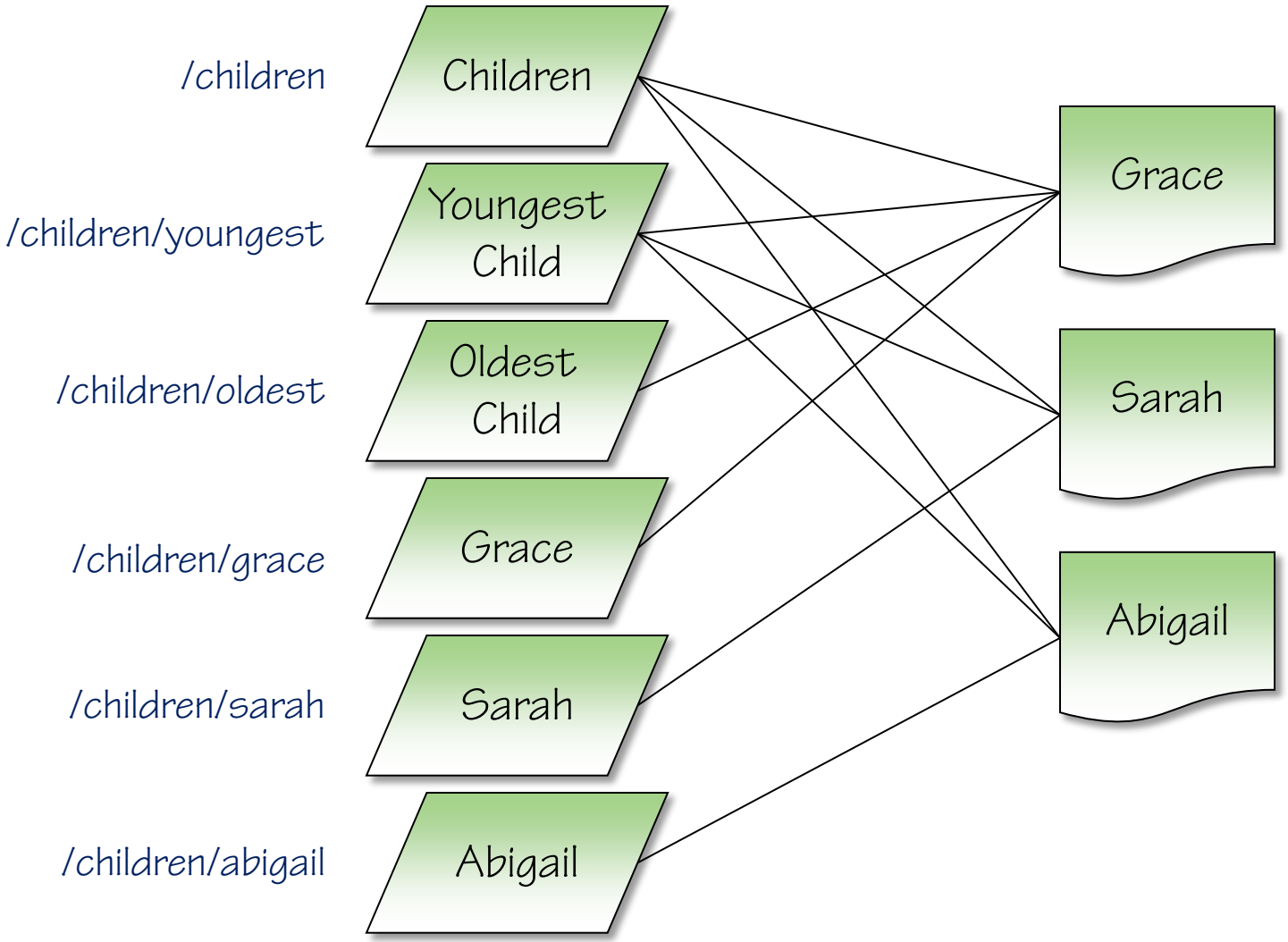
Mapping Concepts to Entities



Resource Identifier

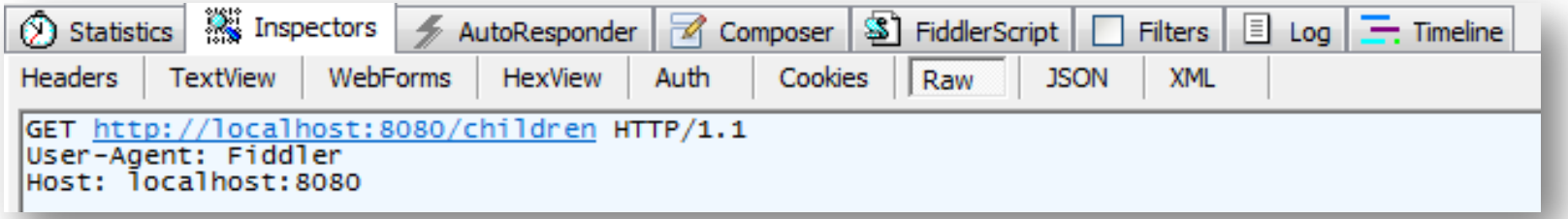
- Identifies a resource in a component interaction
- The server is responsible for ensuring that the mapping semantics between identifier and entities do not change
- Because the identify an abstraction rather than an entity, resource identifiers should change as infrequently as possible

Resource Identifiers and URIs



Resource Identifiers in HTTP

Crafting an HTTP request using Fiddler



Crafting an HTTP request using cURL

```
$ curl http://localhost:8080/children
```

Resource Metadata

- **Data that describes the resource**
- **Some examples:**
  - The location identifier of the resource
  - alternate resource identifiers for different formats
  - Entity tag information

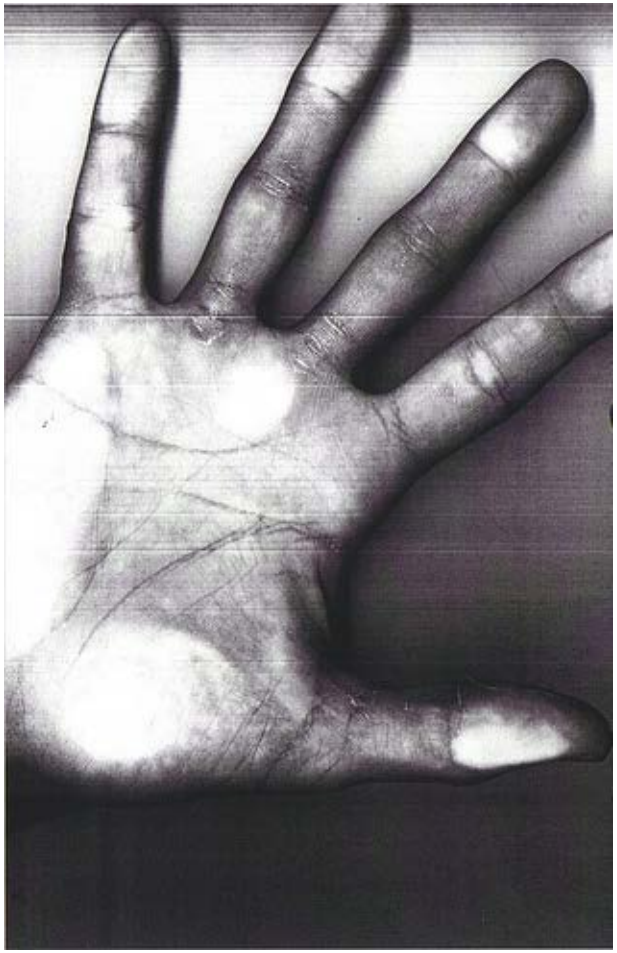
Resource Metadata in HTTP

```
POST http://localhost:8080/children HTTP/1.1
Host: localhost:8080
Content-Length: 39
content-type: application/x-www-form-urlencoded

firstname=Grace&lastname=Dierking&age=4
```

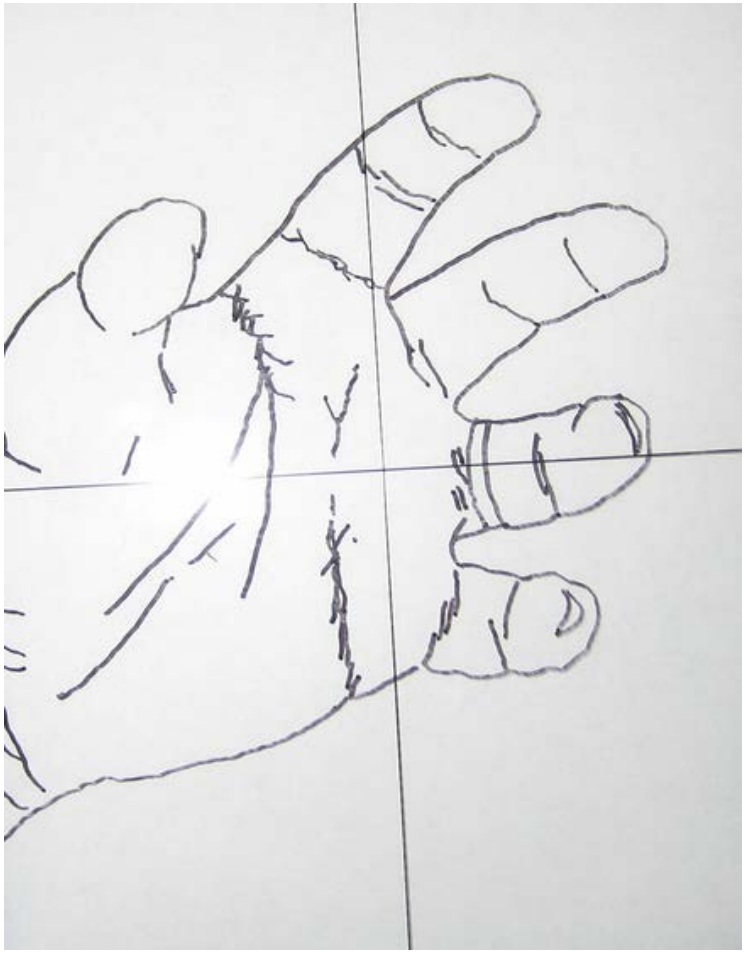
```
HTTP/1.1 201 Created
Content-Length: 0
Location: http://localhost:8080/children/grace
ETag: "18099fbe-ec69-428a-be49-1068c7faec8a"
Server: Microsoft-HTTPAPI/2.0
Date: Sat, 25 Feb 2012 22:41:33 GMT
```

Representation



- Resource state at a point in time
- Can be any sequence of bytes
- A resource can have multiple available representations
- Content negotiation is the process of selecting the best representation of a resource

Representation Metadata



- **Data that describes the representation**
- **Helps the client or server know how to process the bytes**
  - How it's structure
  - What it means

Requesting a Representation

```
GET http://localhost:8080/children/grace HTTP/1.1
Host: localhost:8080
Accept: application/json
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
```

```
HTTP/1.1 200 OK
Date: Sat, 25 Feb 2012 08:06:14 GMT
Content-Type: application/json
Content-Length: 999

{...}
```

Requesting a Different Representation

```
GET http://localhost:8080/children/grace HTTP/1.1
Host: localhost:8080
Accept: image/png
```

```
HTTP/1.1 200 OK
Date: Sat, 25 Feb 2012 08:06:14 GMT
Content-Type: image/png
Content-Length: 999

01010101...
```

Control Data

- Data that describes a message sent between components
- Provides the semantics for the message exchange
- Enables overriding the default behavior of connectors

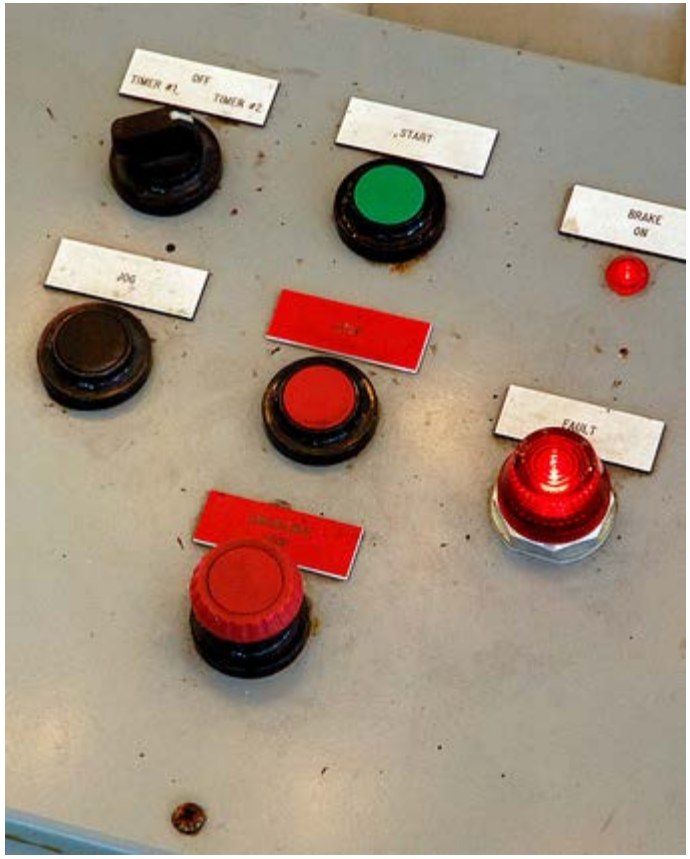
Control Data in HTTP

```
GET http://localhost:8080/children HTTP/1.1
Host: localhost:8080
if-none-match: "d621987e-9cbb-4c75-a4c4-365c67080cba"
```

```
HTTP/1.1 200 OK
Cache-Control: max-age=86400
Content-Length: 159
Content-Type: application/json; charset=utf-8
ETag: "fe7de8c3-066e-4346-b29a-8771309cfae3"
Date: Sat, 25 Feb 2012 23:05:55 GMT

{...}
```

Hypermedia



- This is the uniform interface constraint that people most often ignore
- Hypermedia as the way to initiate state transitions is arguably the most significant differentiator between REST and RPC
- The big idea is to reduce coupling between the client and server
- Coupling is reduced by reducing the number of URLs that the client needs to know about

Anatomy of a Hypermedia Control

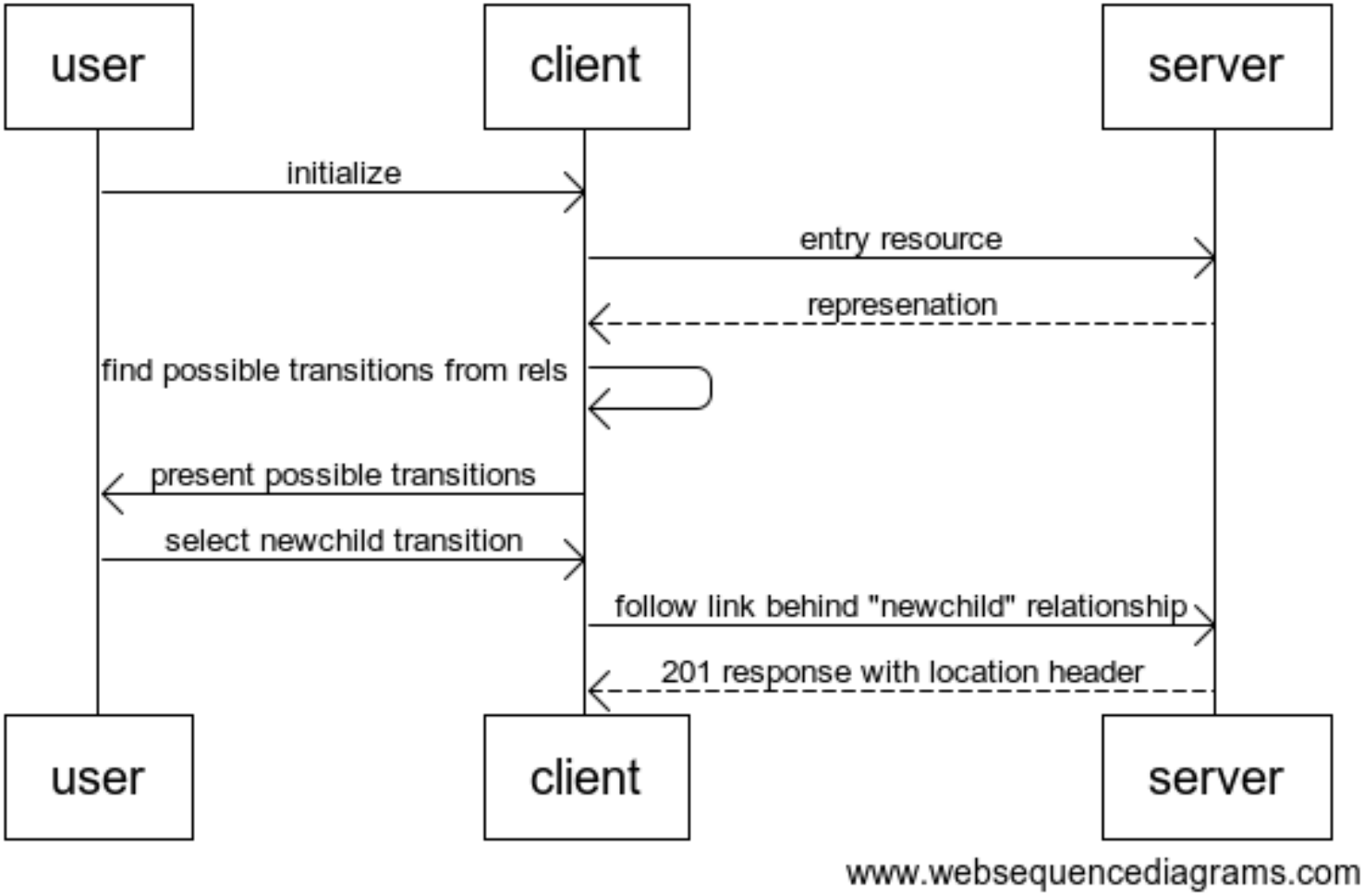
```
<a href="http://localhost:8080/children/grace" rel="child">
   Grace Dierking
</a>
```

```
<form class="newchild" action="children" method="post">
   First Name <input name="firstname" type="text" /><br />
   Last Name <input name="lastname" type="text" /><br />
   Age <input name="age" type="text" /><br />
   <input type="submit" value="New Child" />
</form>
```

Different Types of Links

Link Type	Example
Embedded	HTML <img> tag
Outbound	HTML <a> tag
Templated	HTML form with method="GET" URI Templates
Idempotent	HTML form with method="PUT"
Non-idempotent	HTML form with method="POST"

Example Hypermedia Workflow



Summary

- Connectors and Components
- Resources
- Representations
- Control Data
- Hypermedia

- Fielding, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- Amundsen, Mike. Building Hypermedia APIs with HTML5 and Node. O'Reilly Media, 2011.
- URI Template Specification Draft - http://tools.ietf.org/html/draft-gregorio-uritemplate-08
- Fiddler - http://www.fiddler2.com/fiddler2/
- cURL - http://curl.haxx.se/