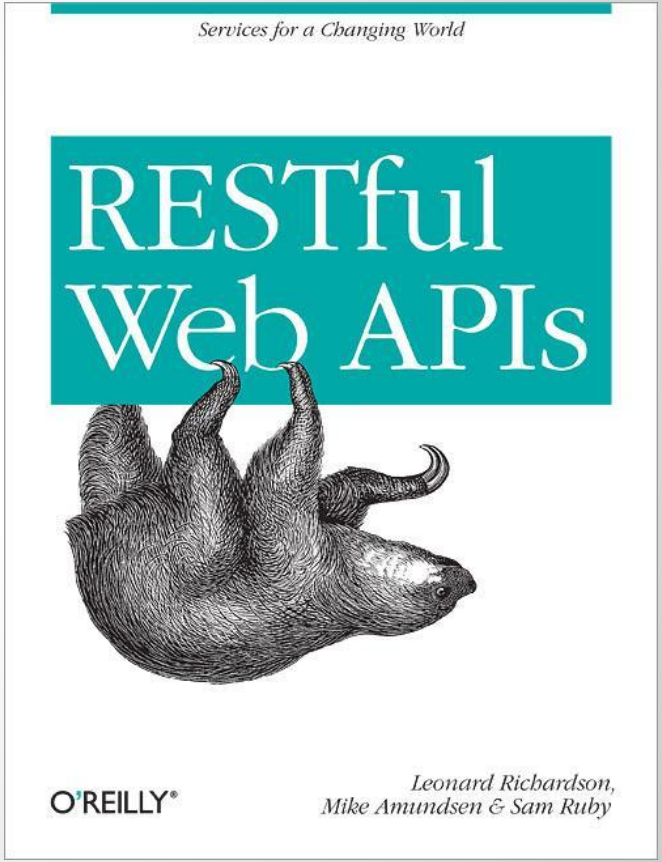
REST, Hypermedia and the Semantic Gap

Why "RMM Level-3" is not good enough.

+MikeAmundsen
@mamund

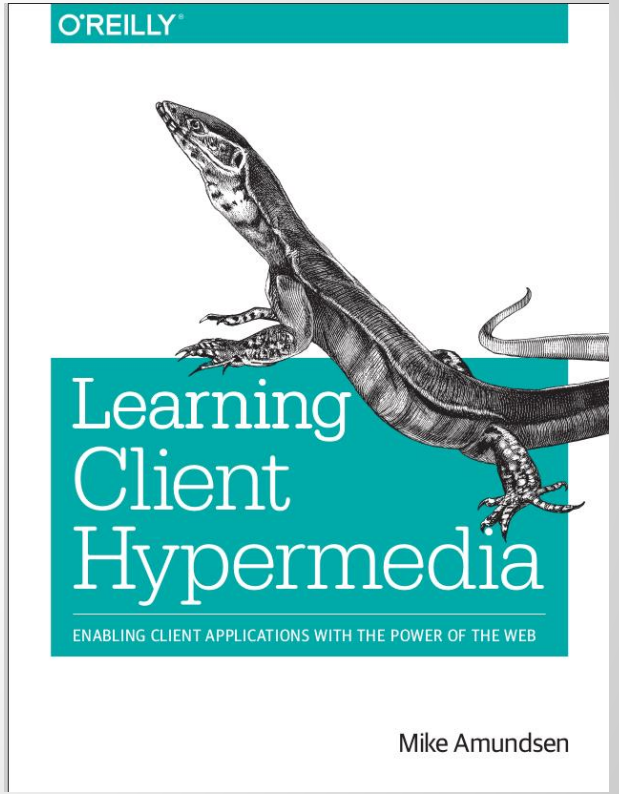
Introductions

- Mike Amundsen
- 25 years in computing
- 14 books on programming
- 10 years w/ "REST"
- "RESTful Web APIs"
  w/ Leonard Richardson



Learning Hypermedia Clients

- Focus on the client side code
- Covers human-driven & M2M
- Lots of code!
- Due in fall 2015
- @LCHBook #LCHProject



Hallway Conversations Podcast

Hosted by Phil Japikse, Steve Bohlen, Lee Brandt, James Bender

Website: www.hallwayconversations.com

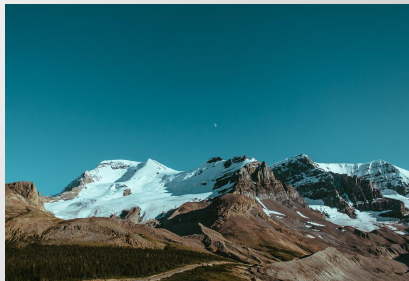
iTunes: http://bit.ly/hallway_convo_itunes

Feed Burner: http://bit.ly/hallway_convo_feed

Also available through Windows Store

Let's talk about...

- Fielding's REST
- HTTP APIs & CRUD
- Hypermedia APIs
- The Semantic Gap



REST - The Short Story

Fielding's Dissertation

"This dissertation defines a framework for understanding software architecture via architectural styles and demonstrates how styles can be used to guide the architectural design of network-based application software."
- Fielding, 2000

REST in one slide

Properties

- Performance
- Scalability
- Simplicity
- Modifiability
- Visibility
- Portability
- Reliability

REST in one slide

Properties

- Performance
- Scalability
- Simplicity
- Modifiability
- Visibility
- Portability
- Reliability

+ Requirements

- Low-Entry Barrier
- Extensibility
- Distributed Hypermedia
- Internet Scale

REST in one slide

Properties

- Performance
- Scalability
- Simplicity
- Modifiability
- Visibility
- Portability
- Reliability

+ Requirements

- Low-Entry Barrier
- Extensibility
- Distributed Hypermedia
- Internet Scale

= Constraints

- Client-Server
- Stateless
- Cache
- Uniform Interface
- Layered System
- Code on Demand

Affordances

"When I say hypertext, I mean the simultaneous presentation of information and controls such that the information becomes the affordance through which the user (or automaton) obtains choices and selects actions." - Fielding, 2008

Affordances

*"When I say hypertext, I mean the simultaneous presentation of information and controls such that **the information becomes the affordance** through which the user (or automaton) obtains choices and selects actions." - Fielding, 2008*

Affordances

HTTP APIs - The Shared Story

RESTful Web Services - 2007

"Our ultimate goal in this book is to reunite the programmable web with the human web. We envision a single interconnected network: a World Wide Web that runs on one set of servers, uses one set of protocols, and obeys one set of design principles."

- Richardson & Ruby, 2008

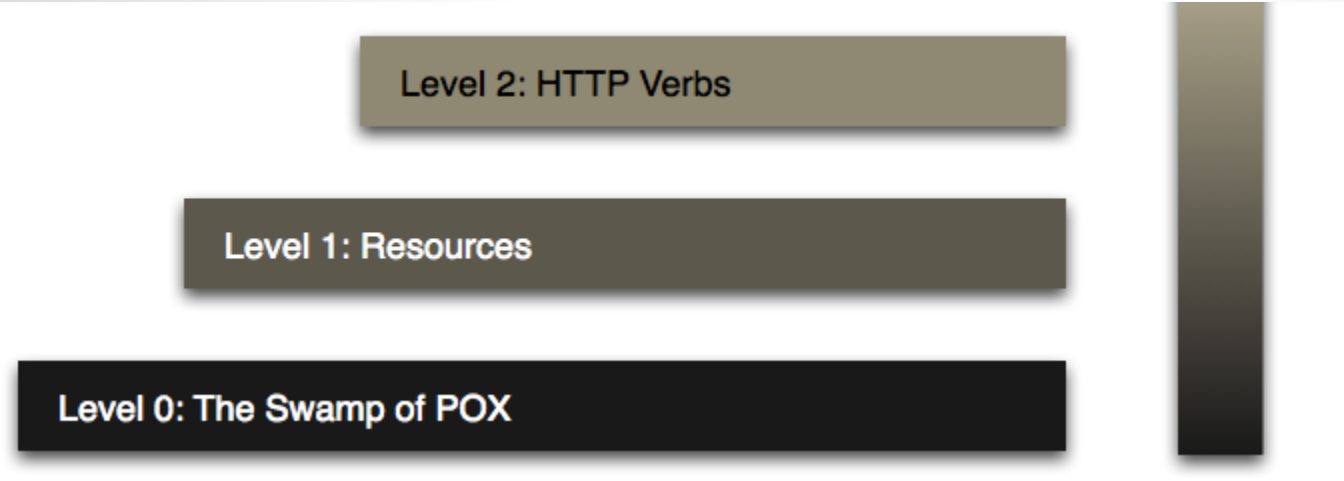
Richardson Maturity Model (RMM)

Richardson Maturity Model (RMM)

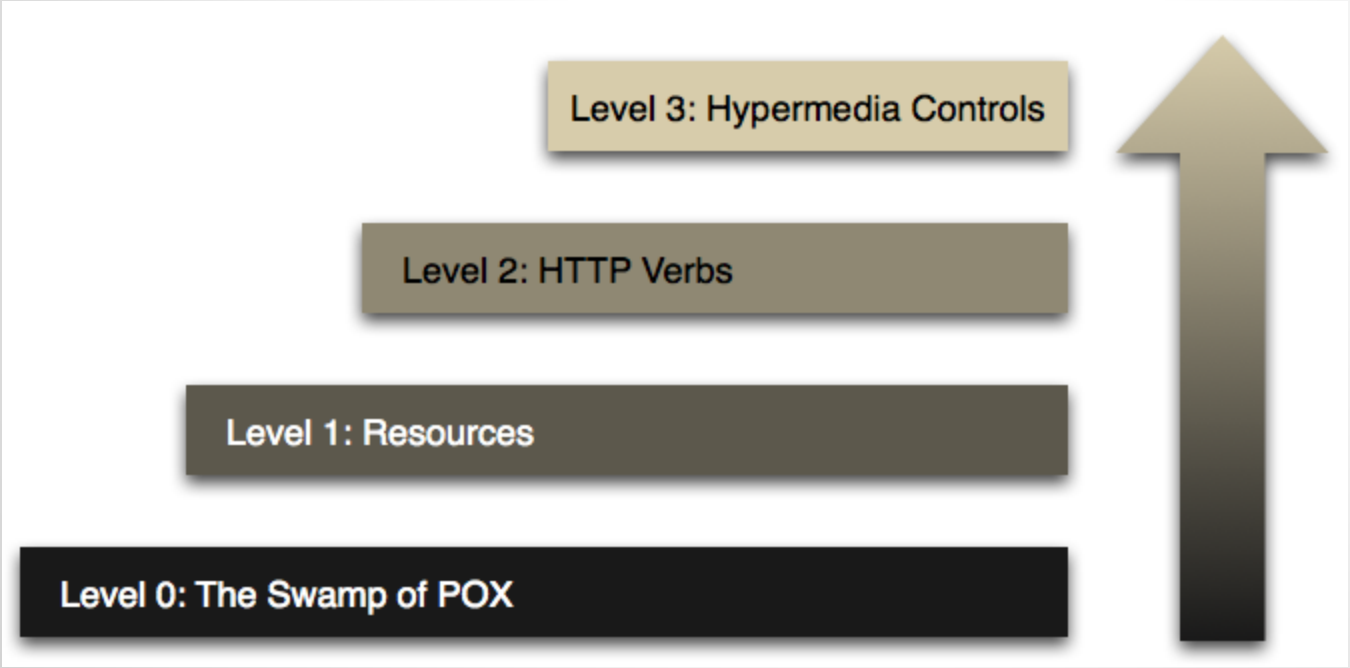
Level 0: The Swamp of POX

Richardson Maturity Model (RMM)

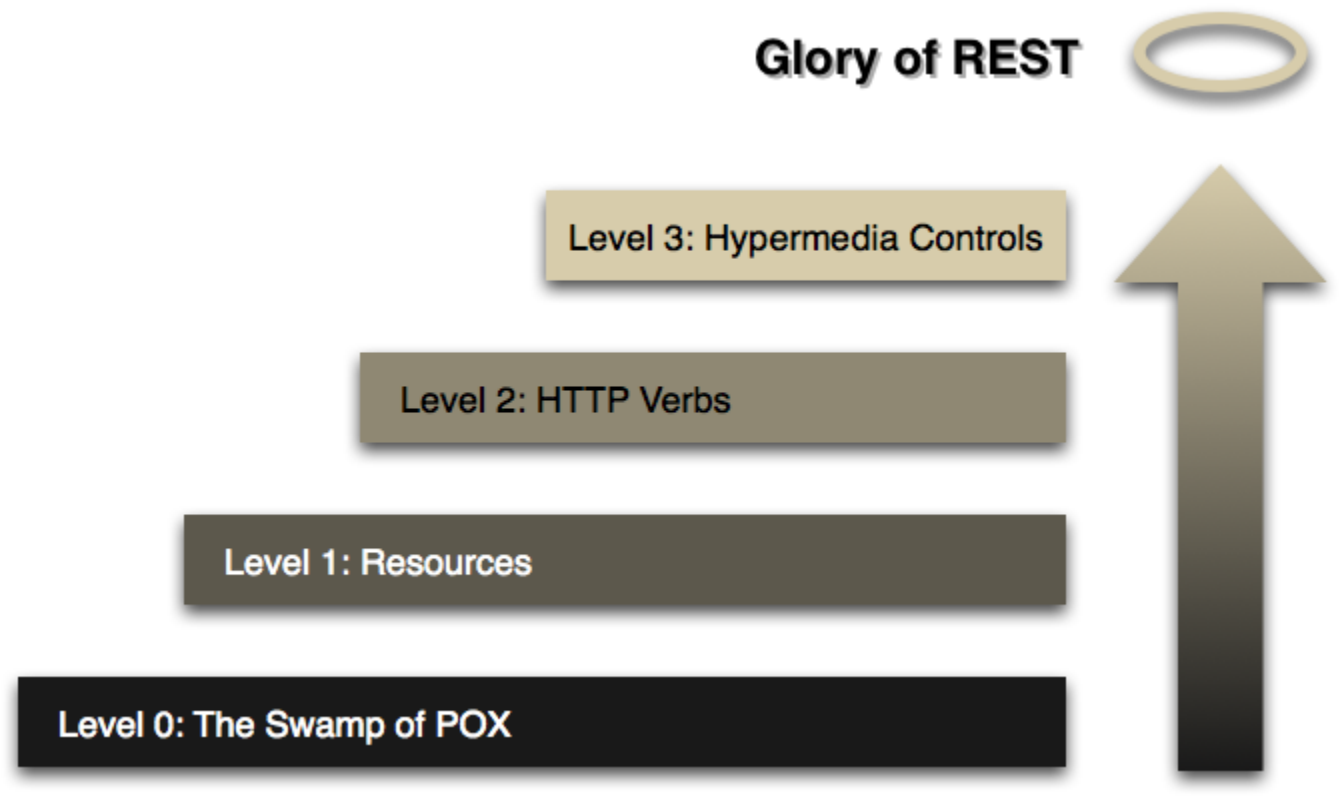
Richardson Maturity Model (RMM)



Richardson Maturity Model (RMM)



Richardson Maturity Model (RMM)



Richardson Maturity Model (RMM)

Level 2: HTTP Verbs

CRUD in one slide

Resource	Method	Representation	Status Codes
Employee	GET	Employee Format	200, 301, 410
Employee	PUT	Employee Format	200, 301, 400, 410
Employee	DELETE	N/A	200, 204
All Employees	GET	Employee List Format	200, 301
All Employees	POST	Employee Format	201, 400

Gregorio's Four Questions (2006)

1. What are the URIs?
2. What are the formats?
3. What methods are supported at each URI?
4. What status codes could be returned?

Common CRUD Guidance

- URI design is the primary task `http://{server}/{collection}/{id}`
- Focus on serializing domain objects `{customer: {name:"mike",...}}`
- Use URIs to express object relationships `http://example.com/users/abc/friends/xyz`
- Use controller URIs to handle service

To-Do CRUD App Demo

To-Do CRUD App Demo

```
{
  - tasks: [
      - {
            id: 0,
            text: "this is some item"
        },
      - {
            id: 1,
            text: "this is another item"
        },
      - {
            id: 2,
            text: "this is one more item"
        },
      - {
            id: 3,
            text: "this is possibly an item"
        }
    ]
}
```

To-Do CRUD App Demo

```
g.addUrl = '/tasks/';
g.listUrl = '/tasks/';
g.searchUrl = '/tasks/search?text={@text}';
g.completeUrl = '/tasks/complete/';

// prime system
function init() {⟺}

// handle "list"
function refreshList() {⟺}

// handle "search"
function searchList() {⟺}

// handle "add"
function addToList() {⟺}

// handle "complete"
function completeItem() {⟺}

/* parse the returned document */
function showList() {⟺}

function initButtons() {⟺}
function clickButton() {⟺}

// handle network request/response
function makeRequest(href, context, body) {⟺}
function processResponse(ajax, context) {⟺}
```

Hypermedia APIs - The Linked Story

RESTful Web APIs - 2013

*"**RESTful Web Services** covered hypermedia, but it wasn't central to the book. It was possible to skip the hypermedia parts of the book and still design a functioning API. By contrast, **RESTful Web APIs** is effectively a book about hypermedia." - Richardson & Amundsen*

H-Factors (2010)

Link Factors

- LO
- LE
- LT
- LN
- LI

```
<a href="http://www.example.org/search" title="view search page">Search</a>
```

```
<img src="http://www.example.org/images/logo" title="company logo" />
```

```
<form method="get">
  <label>Search term:</label>
  <input name="query" type="text" value="" />
  <input type="submit" />
</form>
```

```
<form method="post" action="http://www.example.org/my-keywords"/>

  <label>Keywords:</label>
  <input name="keywords" type="text" value="" />
  <input type="submit" />
</form>
```

```
function delete(id)
{
  var client = new XMLHttpRequest();
  client.open("DELETE", "/records/"+id);
}
```

H-Factors

Control Factors

- CR
- CU
- CM
- CL

```
<xsl:include href="http://www.exmaple.org/newsfeed" accept="application/rss" />
```

```
<form method="post" action="http://www.example.org/my-keywords" enctype="application/x-www-form-urlencoded" />

  <label>Keywords:</label>
  <input name="keywords" type="text" value="" />
  <input type="submit" />
</form>
```

```
<form method="post" action="http://www.example.org/my-keywords" />
  <label>Keywords:</label>

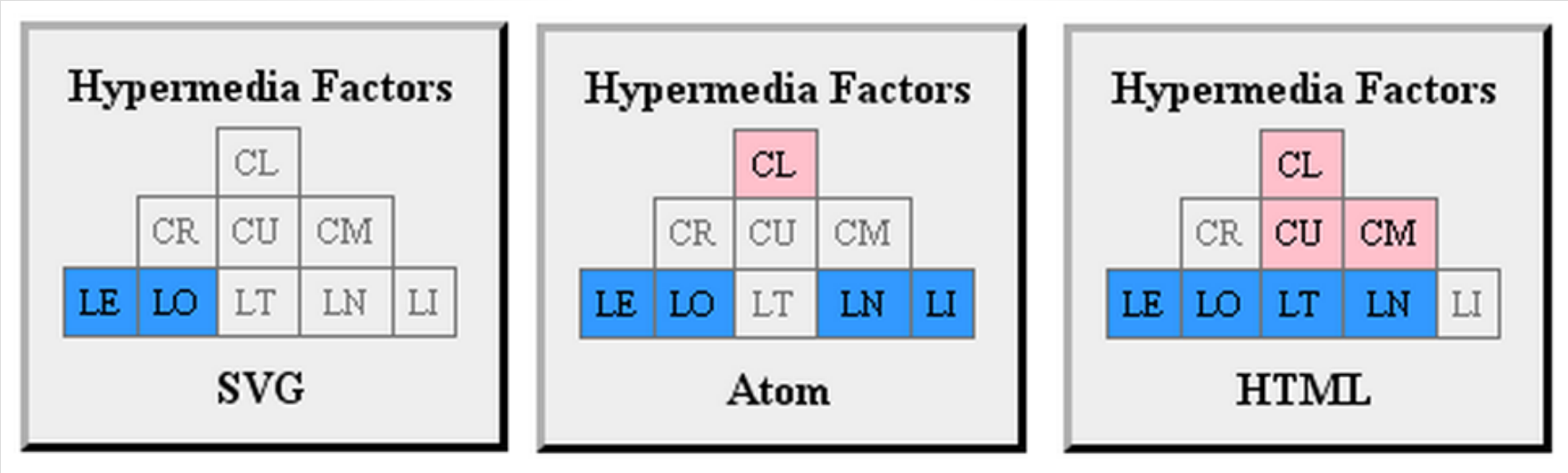
  <input name="keywords" type="text" value="" />
  <input type="submit" />
</form>
```

```
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>Atom-Powered Robots Run Amok</title>

  <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
  <updated>2003-12-13T18:30:02Z</updated>
  <author><name>John Doe</name></author>

  <content>Some text.</content>
  <link rel="edit" href="http://example.org/edit/first-post.atom"/>
</entry>
```

H-Factors


Hypermedia Factors
CL
CR
CU
CM
LE
LO
LT
LN
LI
SVG
Hypermedia Factors
CL
CR
CU
CM
LE
LO
LT
LN
LI
Atom
Hypermedia Factors
CL
CR
CU
CM
LE
LO
LT
LN
LI
HTML

Hypermedia in one slide

- ```
  {"link" :
    {"rel" : "help", "href" : "..."}
  }
  ```
- ```
  {"image" :
    {"rel" : "logo", "href" : "..."}
  }
  ```
- ```
  {"form" :
    {"rel" : "edit",
     "href" : "...",
     "method" : "put",
     "data" : [{"name":"...","value":"..."}]
    }
  }
  ```


  ```

To-Do Hypermedia Demo App

To-Do Hypermedia Demo App


```
{
  - links: [
    - {
          rel: "add",
          href: "/tasks/",
          method: "post",
        - data: [
          - {
                name: "text"
            }
          ]
      },
    - {
          rel: "list",
          href: "/tasks/",
          method: "get"
      },
    - {
          rel: "search",
          href: "/tasks/search",
          method: "get",
        - data: [
          - {
                name: "text"
            }
          ]
      }
  ],
```

```
- collection: [
  - {
        id: 0,
        text: "this is some item",
      - link: {
            rel: "complete",
            href: "/tasks/complete/",
            method: "post",
          - data: [
            - {
                  name: "id"
              }
            ]
        }
    },
  - {
        id: 1,
        text: "this is another item",
      - link: {
            rel: "complete",
            href: "/tasks/complete/",
            method: "post",
          - data: [
            - {
                  name: "id"
              }
            ]
        }
    },
  + {…},
  + {…}
]
```


To-Do Hypermedia Demo App

```
var thisPage = function() {

  var g = {};
  g.msg = {};
  g.listUrl = '/tasks/';

  // prime the system
  function init() {...}

  /* parse the response */
  function showResponse() {...}

  // handle possible hypermedia controls
  function showControls() {...}
  function clickButton() {...}

  // handle network request/response
  function makeRequest(href, context, body) {...}
  function processResponse(ajax, context) {...}

  var that = {};
  that.init = init;
  return that;
};
```

The Semantic Gap - The Future Story

Profiles - from XMDP to ALPS

- Dublin Core (2000)
- XMDP for HTML (2003)
- Microformats for HTML (2005)
- Activity Streams (2011)
- Schema.org (2011)
- ALPS for HTML (2011)
- Profile Link Relation (2013)

Shared Understanding

- Focus on the domain vocabulary
- Independent of protocol (HTTP, XMPP, etc.)
- Independent of format (HTML, Cj, HAL, etc.)

Shared Understanding

```
<!DOCTYPE html>
<html>
 <head>
   <title>...</title>
 </head>
 <body>

   <img src="..." />

   <a href="...">Home</a>

   <form method="get" action="...">
     <input type="text" name="1" value="..." />
     <input type="submit" value="Search"/>
   </form>

 </body>
</html>
```

Shared Understanding

```
<!DOCTYPE html>
<html>
 <head>
   <title>...</title>
 </head>
 <body>

   <img src="..." />

   <a href="..." rel="home">Home</a>

   <form method="get" action="...">
     <input type="text" name="keyword" value="..." />
     <input type="submit" class="search"/>
   </form>

 </body>
</html>
```

Shared Understanding

```
<alps>

  <link rel="self" href="http://alps.io/profiles/search" />

  <descriptor id="home" type="safe" />

  <descriptor id="logo" type="safe" rt="image" />

  <descriptor id="search" type="safe">
    <descriptor id="keyword"
      type="semantic"
      cardinality="single" />
  </descriptor>

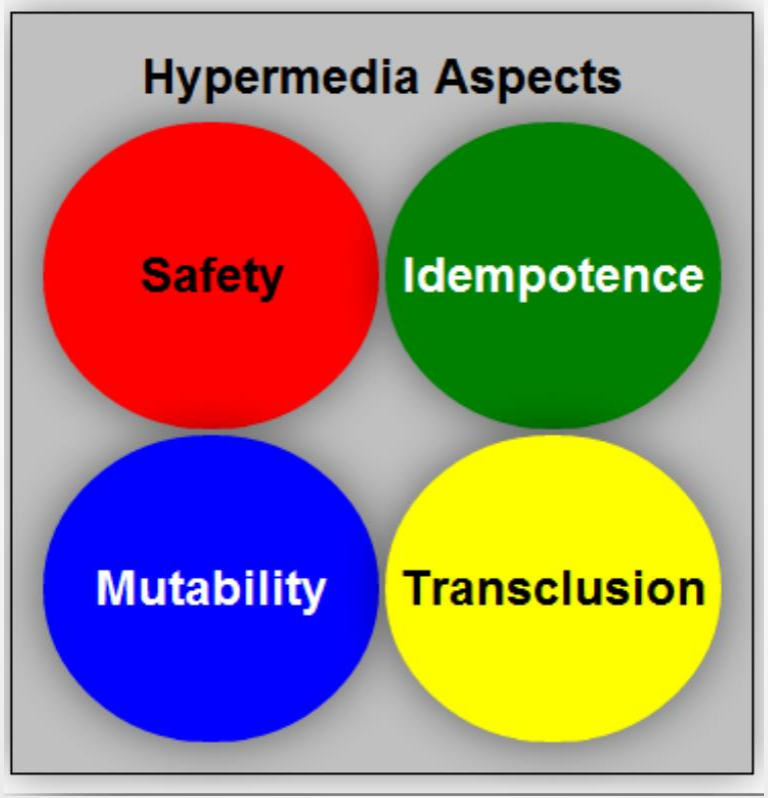
</alps>
```

Affordance Aspects

For the purposes of applying affordances to hypermedia, there are four aspects to consider:

```
<proto
  safe="true|false"
   idempotent="true|false"
   mutable="true|false"
   transclusion="true|false" />
```

Affordance Aspects



Linking

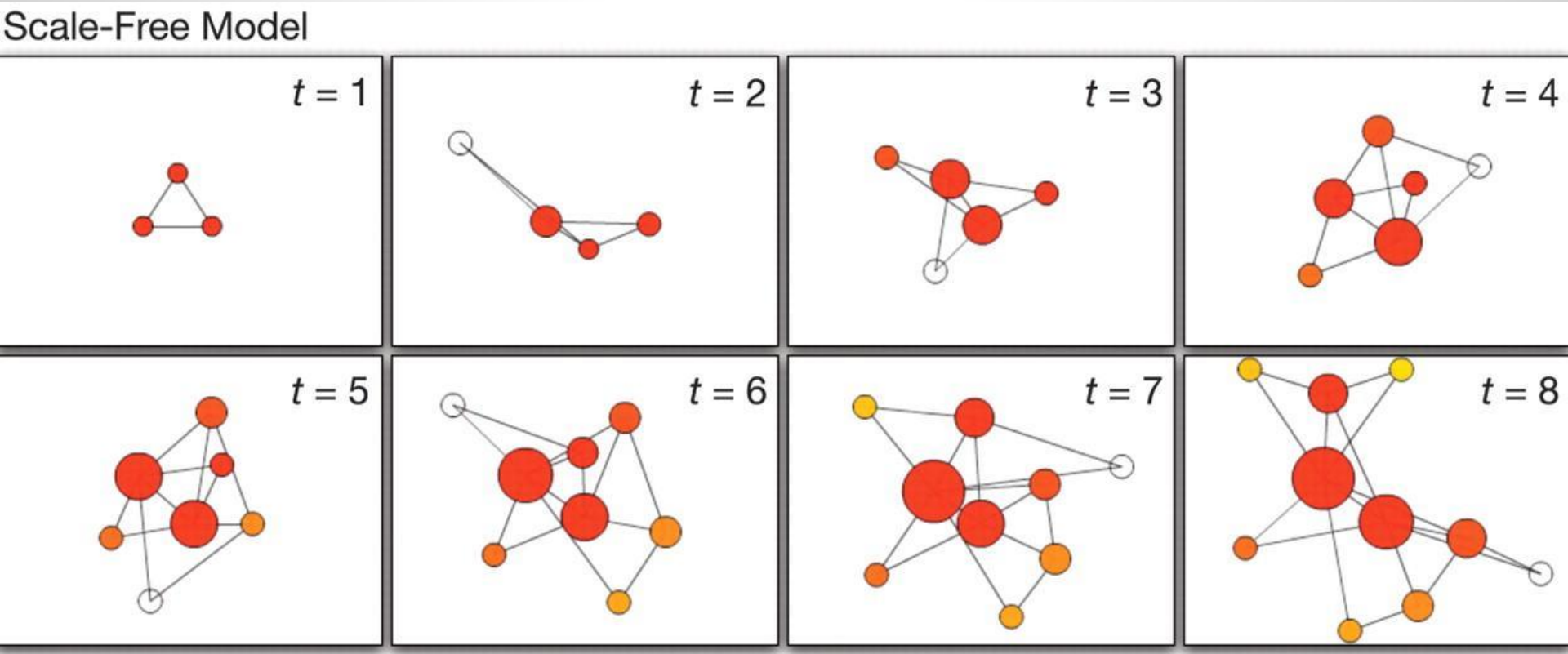
"[Links are] necessary to connect the data we have into a web, a serious, unbounded web in which one can find all kinds of things." - Tim Berners-Lee, 2006-09

Linking

"[Links are] necessary to connect the data we have into a web, a serious, unbounded web in which one can find all kinds of things." - Tim Berners-Lee, 2006-09

Linking

Linking



ALPS in one slide

SoC is key

- Format
- Protocol
- Domain
- Workflow

```
<alps>
  <doc/>
  <descriptor
    type="semantic"/>
  <descriptor
    type="safe|idempotent|unsafe"/>
</alps>
```

To-Do ALPS-driven Demo App

To-Do ALPS-driven Demo App


```
{
  - alps: {
        version: "1.0",
      + doc: {…},
      - descriptor: [
          + {…},
          - {
                id: "contacts",
                type: "semantic",
              - doc: {
                    format: "text",
                    value: "contact item"
                },
              - descriptor: [
                  - {
                        id: "link",
                        type: "safe",
                      - doc: {
                            format: "text",
                            value: "link to contact"
                        }
                    },
                  - {
                        id: "givenName",
                        type: "semantic",
                        href: "http://schema.org/givenName"
                    },
                  - {
                        id: "familyName",
                        type: "semantic",
                        href: "http://schema.org/familyName"
                    },
                  - {
                        id: "email",
                        type: "semantic",
                        href: "http://schema.org/email"
                    },
                  - {
                        id: "telephone",
                        type: "semantic",
                        href: "http://schema.org/telephone"
                    }
                ]
            }
        ]
    }
}
```


To-Do ALPS-driven Demo App

```
{
  - people: [
      - {
            firstName: "Mike",
            lastName: "Amundsen",
            primaryEmail: "mamund@yahoo.com",
            voicePhone: "123-456-7890"
        },
      - {
            firstName: "Mark",
            lastName: "Gunderson",
            primaryEmail: "mgunder@example.com",
            voicePhone: "234-567-8901"
        }
    ]
}
```

To-Do ALPS-driven Demo App


```
{
  - collection: {
      - links: [
          - {
              href: "https://rawgit.com/alps-io/alps-contacts/master/contact-alps.js",
              rel: "profile"
            },
          - {
              href: "https://rawgit.com/alps-io/alps-contacts/master/contact-doc.html",
              rel: "help"
            }
        ],
      - items: [
          + {…},
          - {
              href: "http://localhost:1337/contacts/p0o9i8u7",
            - data: [
                - {
                    name: "givenName",
                    value: "Mark"
                  },
                - {
                    name: "familyName",
                    value: "Gunderson"
                  },
                - {
                    name: "email",
                    value: "mgunder@example.com"
                  },
                - {
                    name: "telephone",
                    value: "234-567-8901"
                  }
              ]
            }
        ],
      - queries: [
          - {
              rt: "contacts",
              rel: "search",
              href: "http://localhost:1337/search",
            - data: [
                - {
                    name: "name",
                    value: ""
                  }
              ]
            }
        ]
    }
}
```


Summary

Summary

- REST
- HTTP APIs
- Hypermedia APIs
- The Semantic Gap

Summary - REST

- Network Architecture
- Properties + Requirements = Constraints
- Data is an architectural element
- Uniform API for all
- Use hypermedia to change state
- Information is the affordance

Summary - CRUD

- Resource Oriented Architecture
- URI + Object = Resource
- Focus on URIs
- Manipulate Resources w/ HTTP Methods
- Use objects to change state
- URI is the affordance

Summary - Hypermedia

- Task Oriented Architecture
- Data + Instructions = Representation
- Focus on Tasks
- Manipulate state via hypermedia controls
- Message is the affordance

Summary - Semantic Gap

- Problem Domain Modeling
- Data + Transitions = Model
- Focus on open world (power law)
- Separate domain model from
  - protocol
  - format
  - workflow

REST, Hypermedia and the Semantic Gap

Why "RMM Level-3" is not good enough.

+MikeAmundsen
@mamund