RESTing in the ALPS

Generic Hypermedia and Domain-Specific APIs

@mamund
Mike Amundsen
Principal API Architect,
Layer 7 Technologies

A simple story

In three parts...

The bona fides

How we got here...

The confession

I left some stuff out...

Where I make amends

The payoff

Ok, let's begin...

How we got here...

The bona fides

Let's start with a quote from 2002

Fielding and Taylor, 2002

"REST provides [this] by
focusing on a shared understanding
of data types with metadata..."

That phrase struck me.

It has become a prime motivator for me.

focusing

focusing
on

focusing
on
shared understanding

shared understanding

How do we share understanding on the Web?

protocols

http

ftp

smtp

protocols

dns

ws

xmpp

But that's not all...

Fielding & Taylor, 2002

"REST components communicate by transferring a representation of the data in a format matching one of an evolving set of standard data types..."

html

csv

hal

standard data types

atom

json

Cj

html

csv

hal

registered media types

atom

json

Cj

html

csv

hal

messages

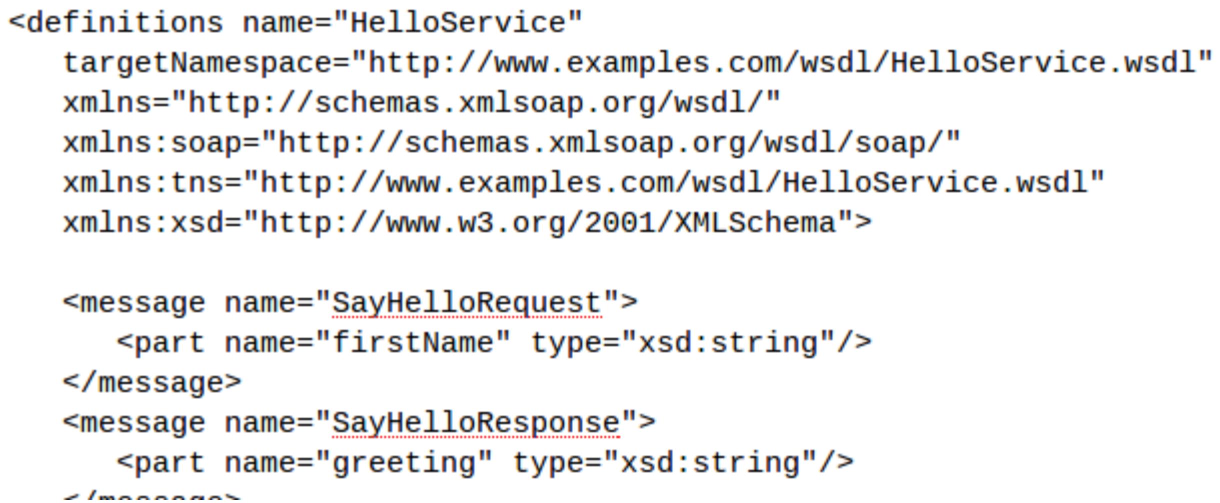
atom

json

Cj

We share understanding via messages.

Description

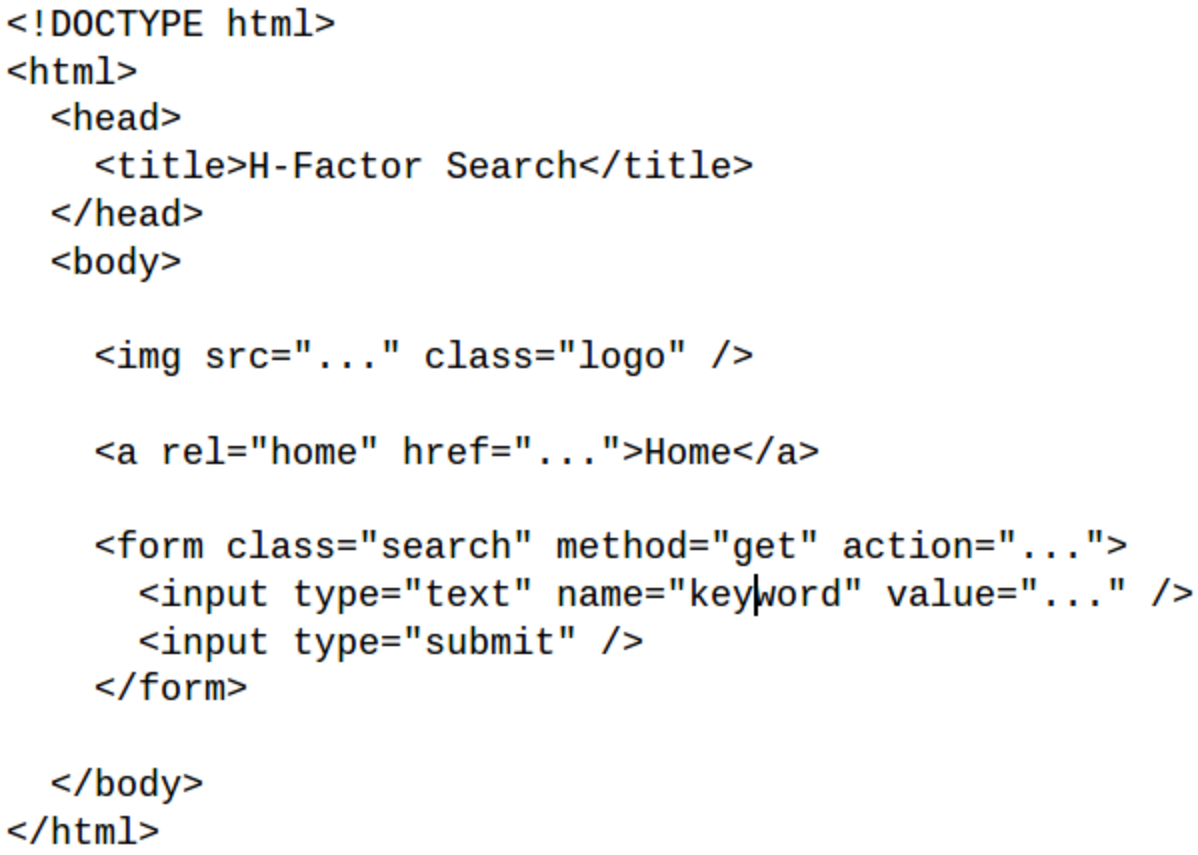


```
<definitions name="HelloService"
   targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <message name="SayHelloRequest">
      <part name="firstName" type="xsd:string"/>
   </message>
   <message name="SayHelloResponse">
      <part name="greeting" type="xsd:string"/>
   </message>
```

Discovery

```
{
  "resources": {
    "http://example.org/rel/widgets": {
      "href": "/widgets/"
    },
    "http://example.org/rel/widget": {
      "href-template": "/widgets/{widget_id}",
      "href-vars": {
        "widget_id": "http://example.org/param/widget"
      },
      "hints": {
        "allow": ["GET", "PUT", "DELETE", "PATCH"],
        "representations": ["application/json"],
        "accept-patch": ["application/json-patch"],
        "accept-post": ["application/xml"],
        "accept-ranges": ["bytes"]
      }
    }
  }
}
```

Hypermedia



```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <img src="..." class="logo" />

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

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

  </body>
</html>
```

These messages tell us what protocol actions are possible.

These messages tell us what **protocol actions** are possible.

How is this done in a message?

affordances

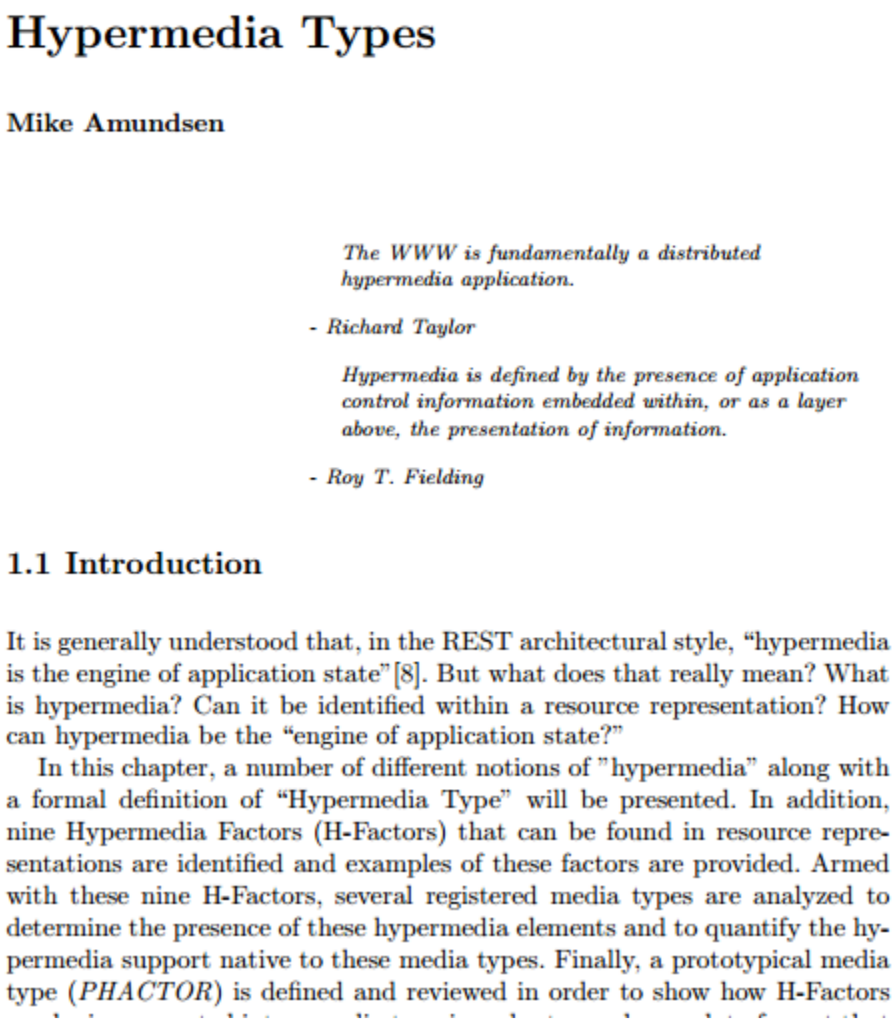
hypermedia affordances

protocol
hypermedia
affordances

Back in 2010, I called those...

H-Factors

H-Factors



Hypermedia Types

Mike Amundsen

> *The WWW is fundamentally a distributed hypermedia application.*
>
> *- Richard Taylor*
>
> *Hypermedia is defined by the presence of application control information embedded within, or as a layer above, the presentation of information.*
>
> *- Roy T. Fielding*

1.1 Introduction

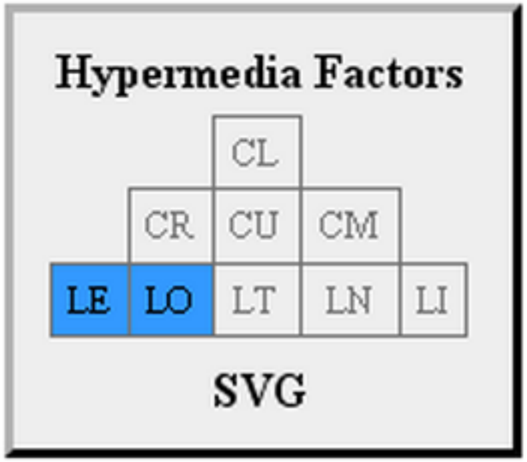
It is generally understood that, in the REST architectural style, “hypermedia is the engine of application state”[8]. But what does that really mean? What is hypermedia? Can it be identified within a resource representation? How can hypermedia be the “engine of application state?”

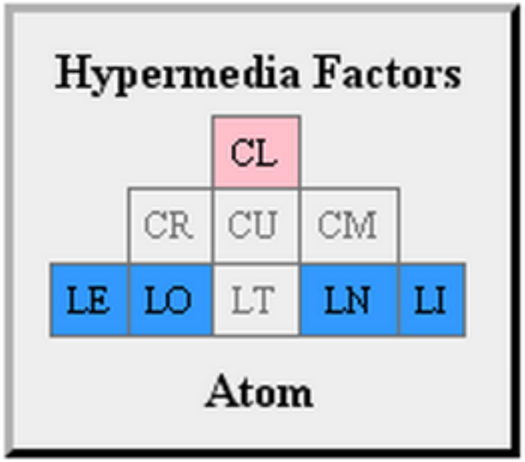
In this chapter, a number of different notions of ”hypermedia” along with a formal definition of “Hypermedia Type” will be presented. In addition, nine Hypermedia Factors (H-Factors) that can be found in resource representations are identified and examples of these factors are provided. Armed with these nine H-Factors, several registered media types are analyzed to determine the presence of these hypermedia elements and to quantify the hypermedia support native to these media types. Finally, a prototypical media type (*PHACTOR*) is defined and reviewed in order to show how H-Factors

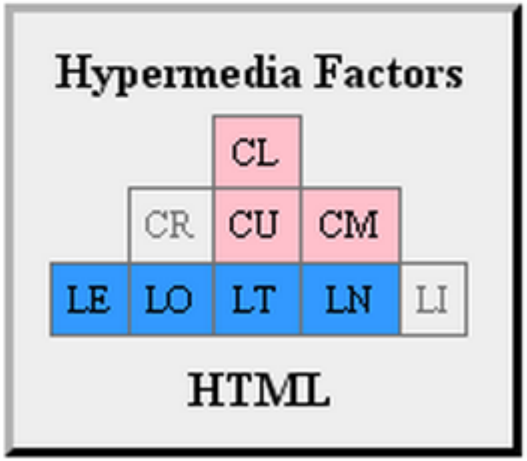
H-Factors

Identify nine features for sharing understanding about protocol actions.

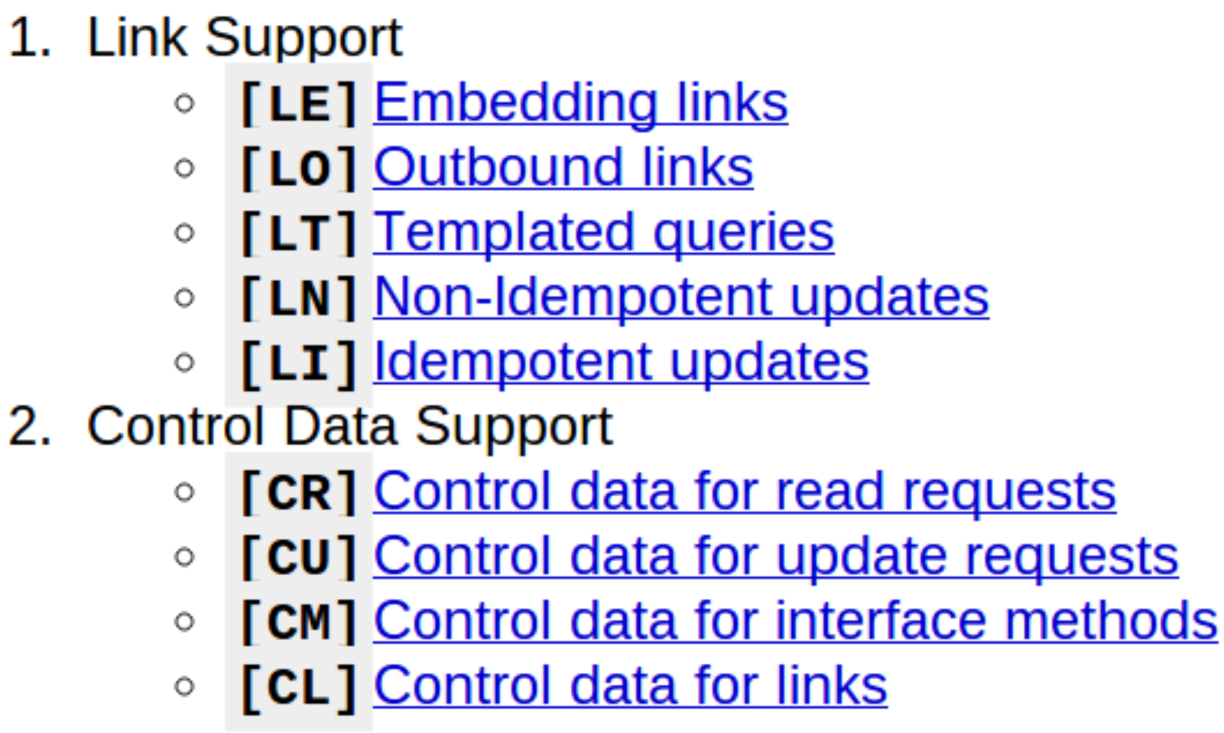
H-Factors


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


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

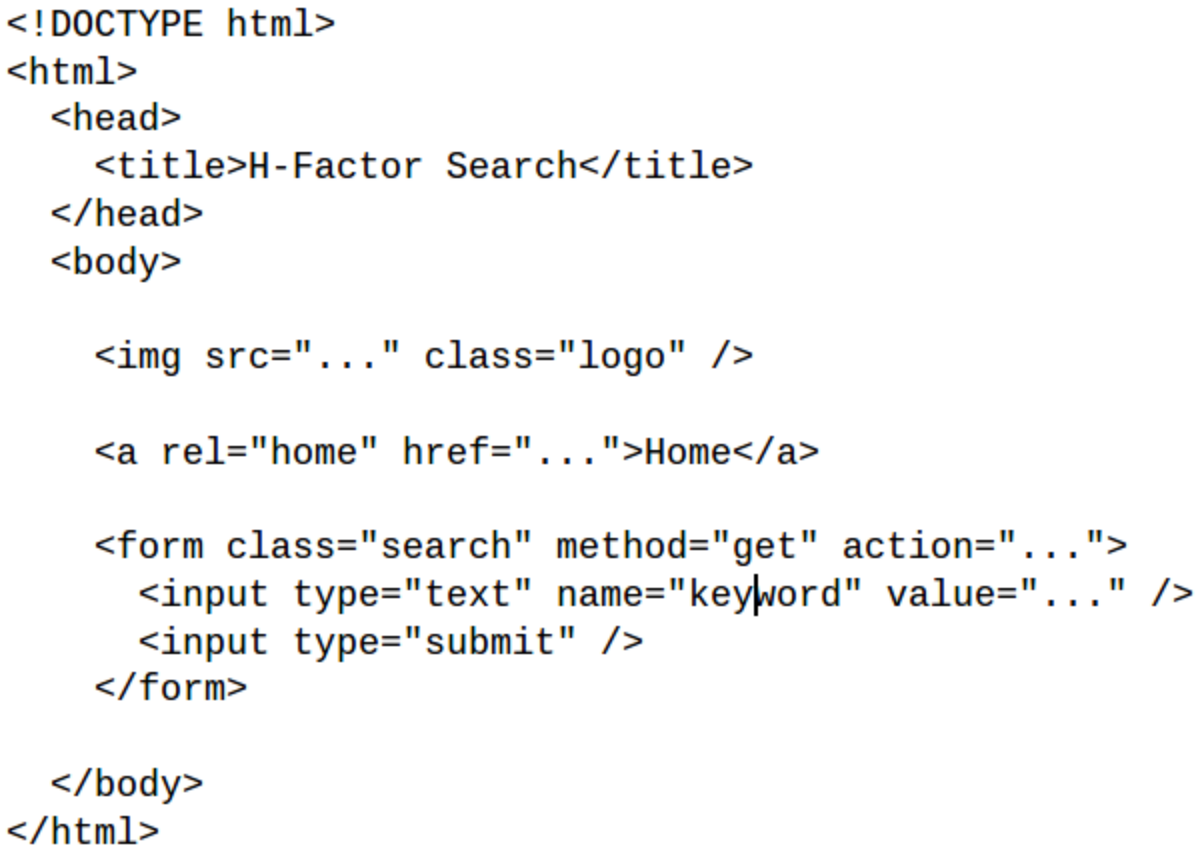

Hypermedia Factors
CL
CR
CU
CM
LE
LO
LT
LN
LI
HTML

H-Factors



1. Link Support
   - [LE] Embedding links
   - [LO] Outbound links
   - [LT] Templated queries
   - [LN] Non-Idempotent updates
   - [LI] Idempotent updates
2. Control Data Support
   - [CR] Control data for read requests
   - [CU] Control data for update requests
   - [CM] Control data for interface methods
   - [CL] Control data for links

H-Factors



```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <img src="..." class="logo" />

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

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

  </body>
</html>
```

H-Factors

Identify the affordances

H-Factors

```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <img src="..." class="logo" />

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

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

  </body>
</html>
```

H-Factors

Categorize them.

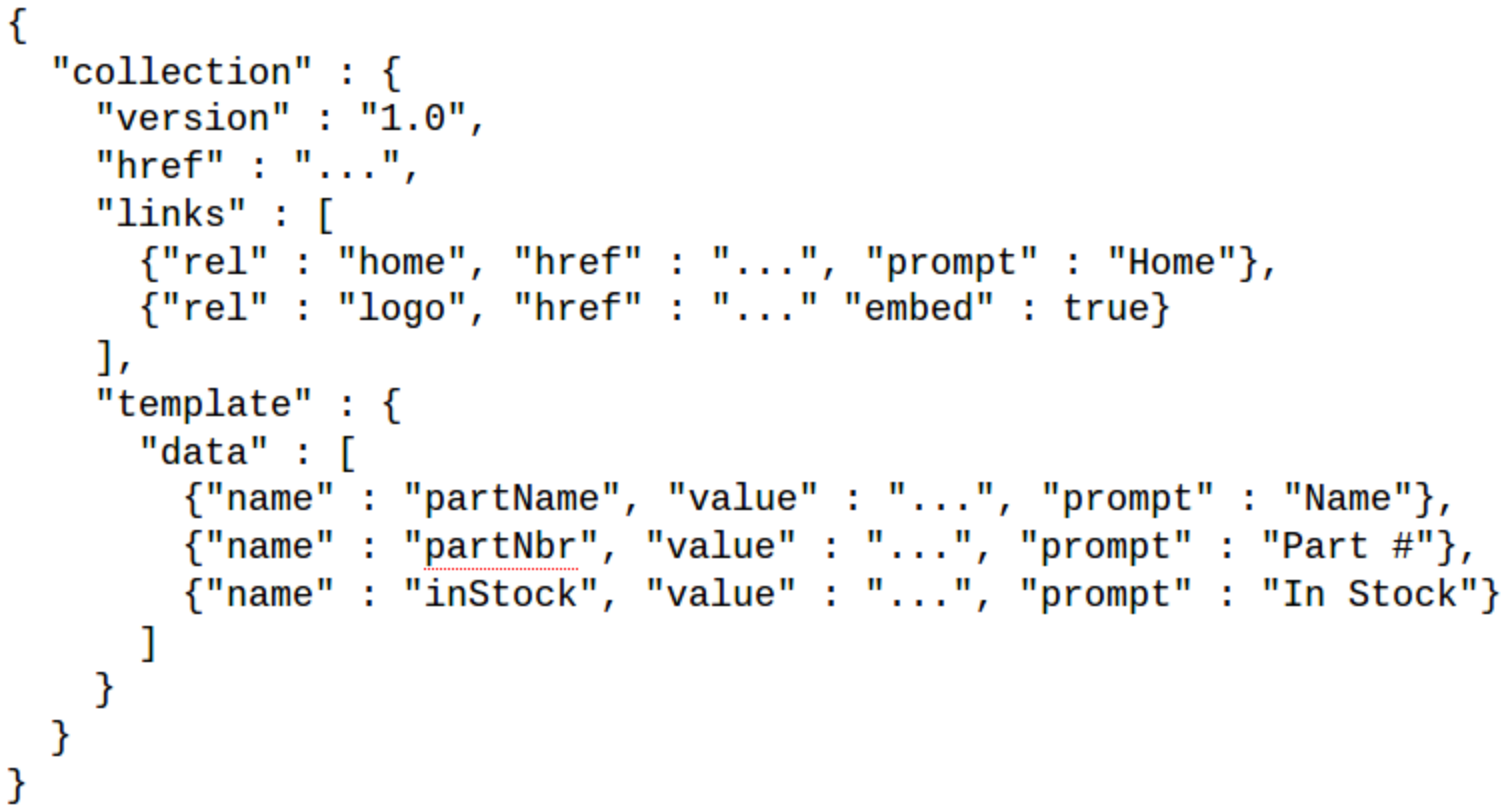
H-Factors

```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <LE src="..." class="logo" />

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

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

  </body>
</html>
```

H-Factors

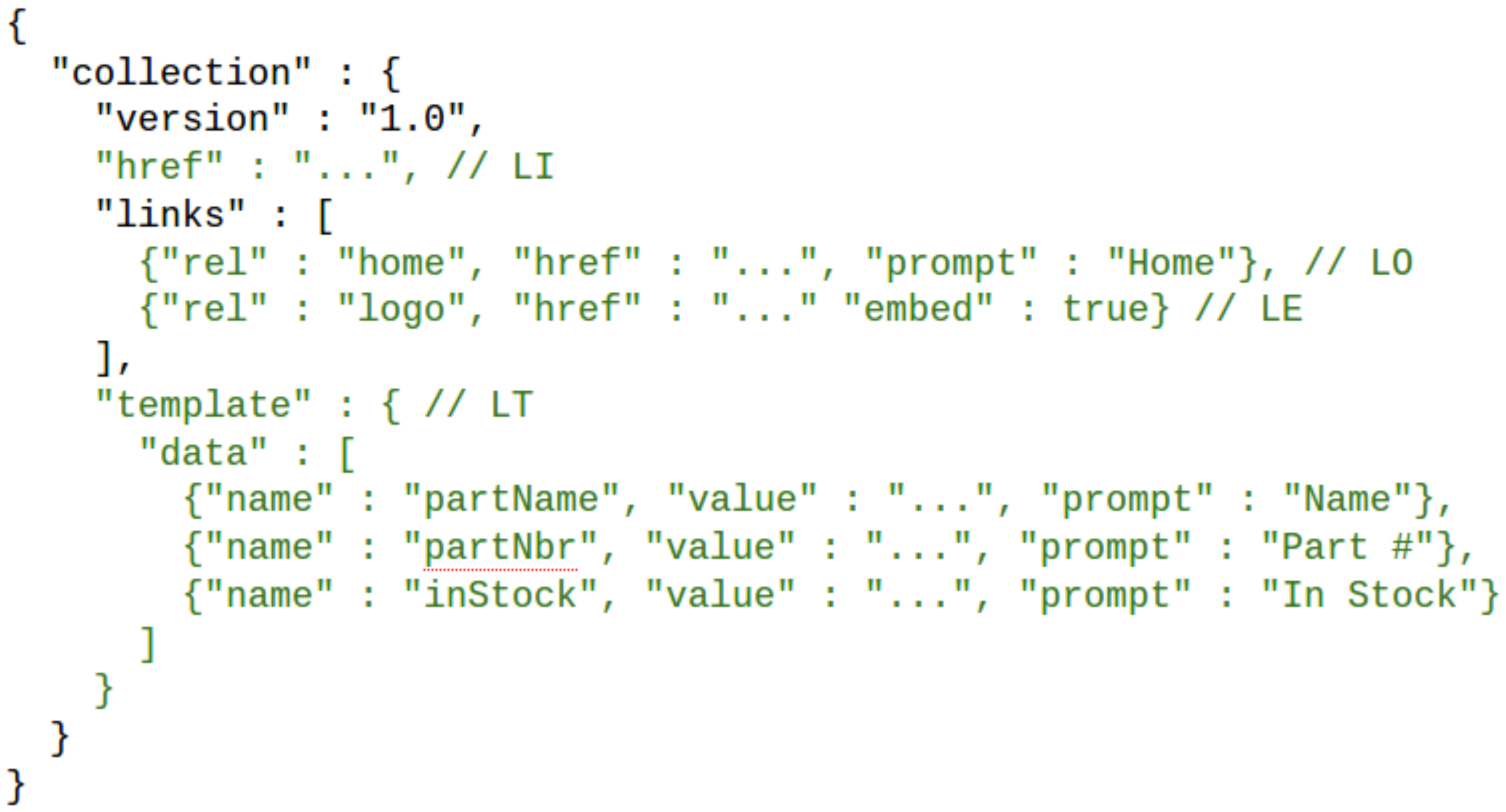
For lots of media type designs.

H-Factors



```
{
  "collection" : {
    "version" : "1.0",
    "href" : "...",
    "links" : [
      {"rel" : "home", "href" : "...", "prompt" : "Home"},
      {"rel" : "logo", "href" : "..." "embed" : true}
    ],
    "template" : {
      "data" : [
        {"name" : "partName", "value" : "...", "prompt" : "Name"},
        {"name" : "partNbr", "value" : "...", "prompt" : "Part #"},
        {"name" : "inStock", "value" : "...", "prompt" : "In Stock"}
      ]
    }
  }
}
```

H-Factors

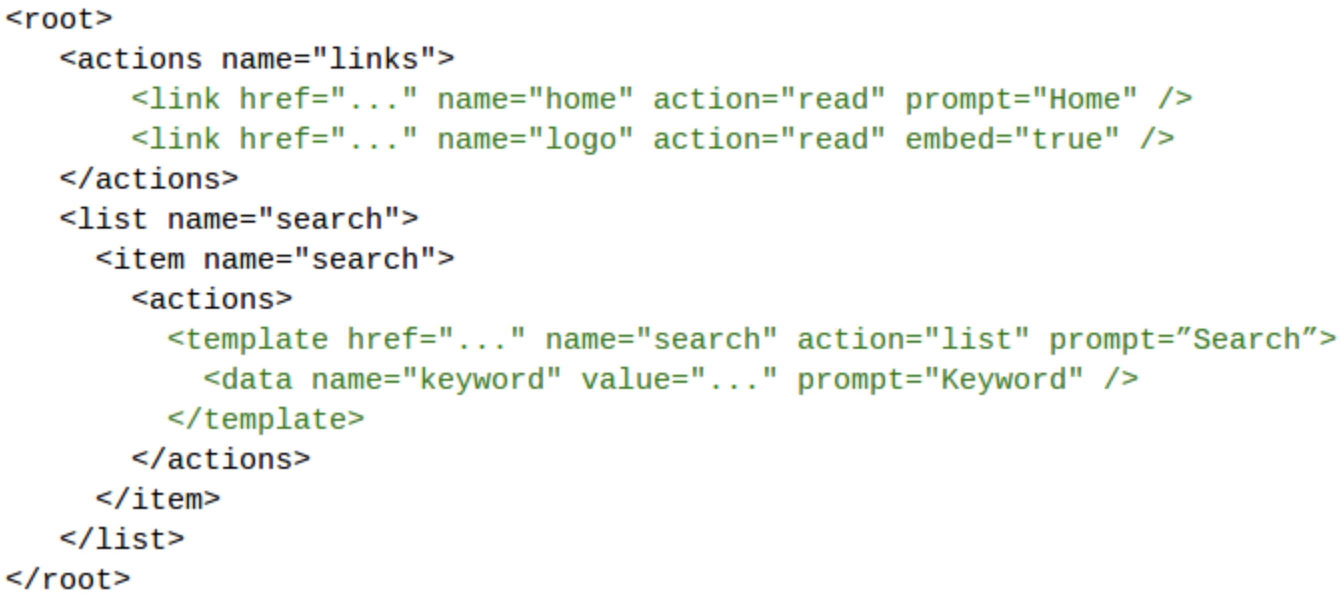


```
{
  "collection" : {
    "version" : "1.0",
    "href" : "...", // LI
    "links" : [
      {"rel" : "home", "href" : "...", "prompt" : "Home"}, // LO
      {"rel" : "logo", "href" : "..." "embed" : true} // LE
    ],
    "template" : { // LT
      "data" : [
        {"name" : "partName", "value" : "...", "prompt" : "Name"},
        {"name" : "partNbr", "value" : "...", "prompt" : "Part #"},
        {"name" : "inStock", "value" : "...", "prompt" : "In Stock"}
      ]
    }
  }
}
```

H-Factors

```
<root>
   <actions name="links">
       <link href="..." name="home" action="read" prompt="Home" />
       <link href="..." name="logo" action="read" embed="true" />
   </actions>
   <list name="search">
     <item name="search">
       <actions>
         <template href="..." name="search" action="list" prompt=”Search”>
           <data name="keyword" value="..." prompt="Keyword" />
         </template>
       </actions>
     </item>
   </list>
</root>
```

H-Factors

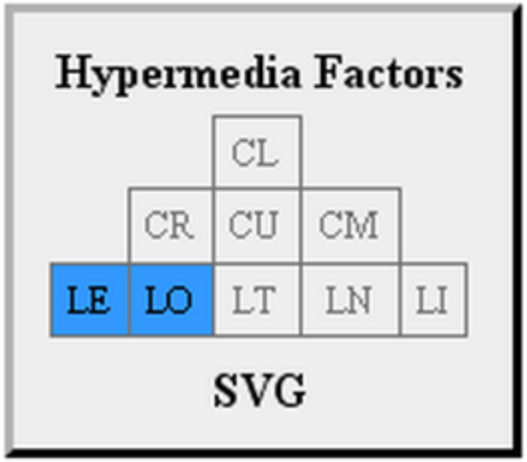


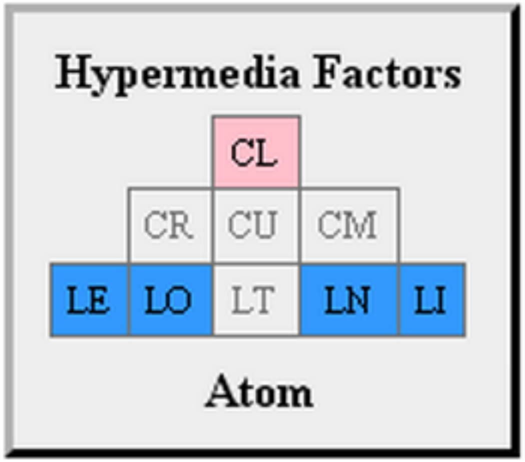
```
<root>
   <actions name="links">
       <link href="..." name="home" action="read" prompt="Home" />
       <link href="..." name="logo" action="read" embed="true" />
   </actions>
   <list name="search">
     <item name="search">
       <actions>
         <template href="..." name="search" action="list" prompt=”Search”>
           <data name="keyword" value="..." prompt="Keyword" />
         </template>
       </actions>
     </item>
   </list>
</root>
```

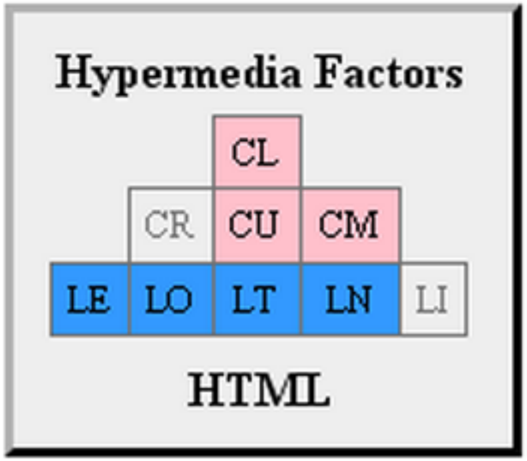
H-Factors

This gives us a tool for lots of tasks...

H-Factors

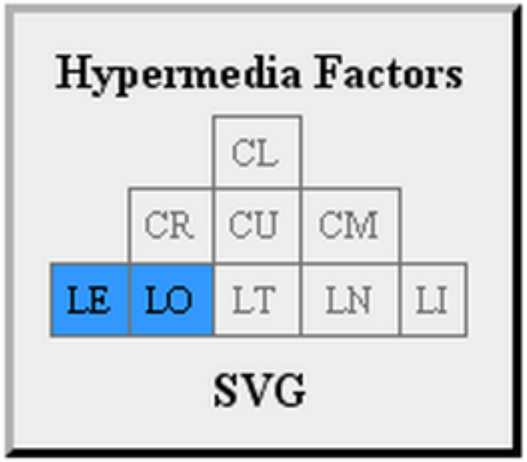


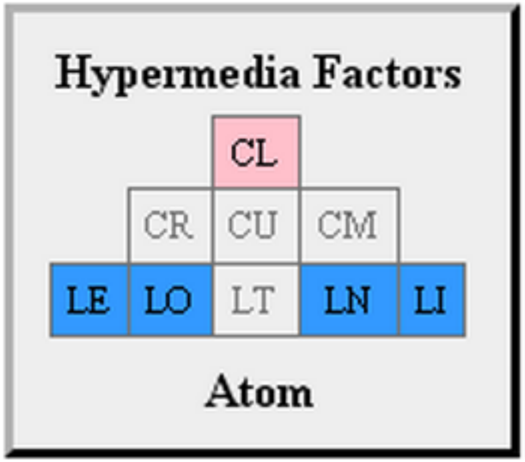


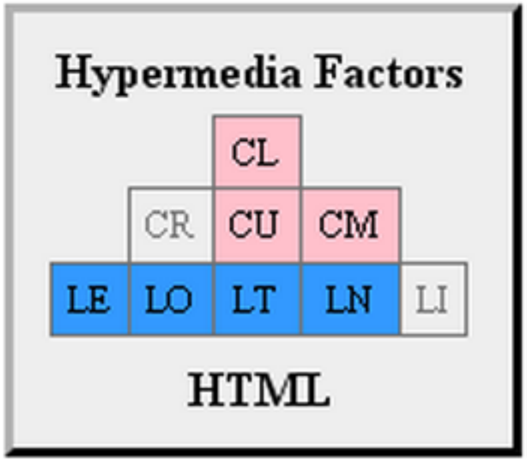


We can analyze existing designs

H-Factors

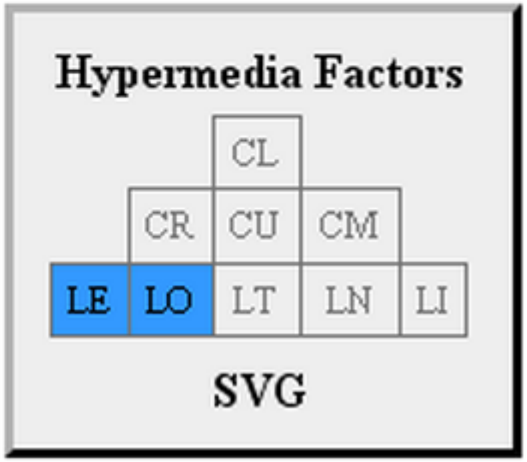






We can categorize types for 'best fit' implementations

H-Factors

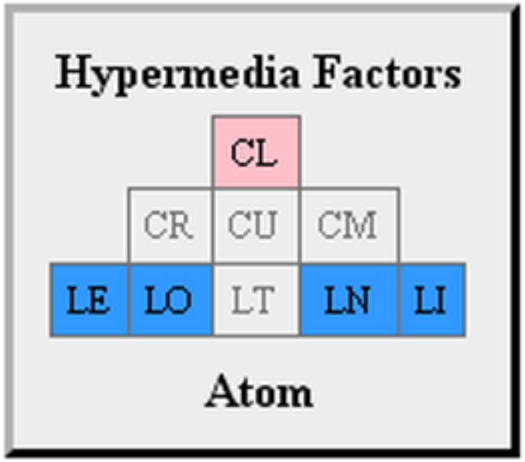

Hypermedia Factors

CL

CR CU CM

LE LO LT LN LI

SVG

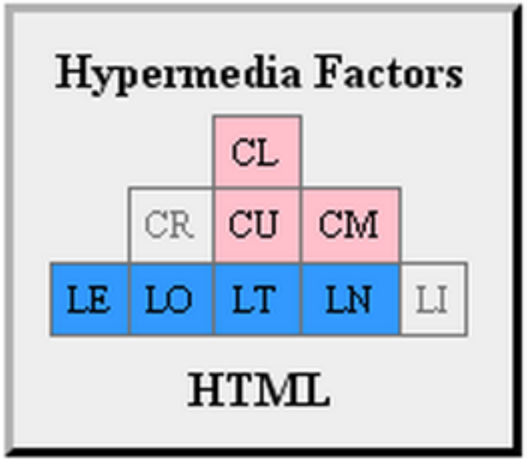

Hypermedia Factors

CL

CR CU CM

LE LO LT LN LI

Atom


Hypermedia Factors

CL

CR CU CM

LE LO LT LN LI

HTML

We can use H-Factors to model prototype designs

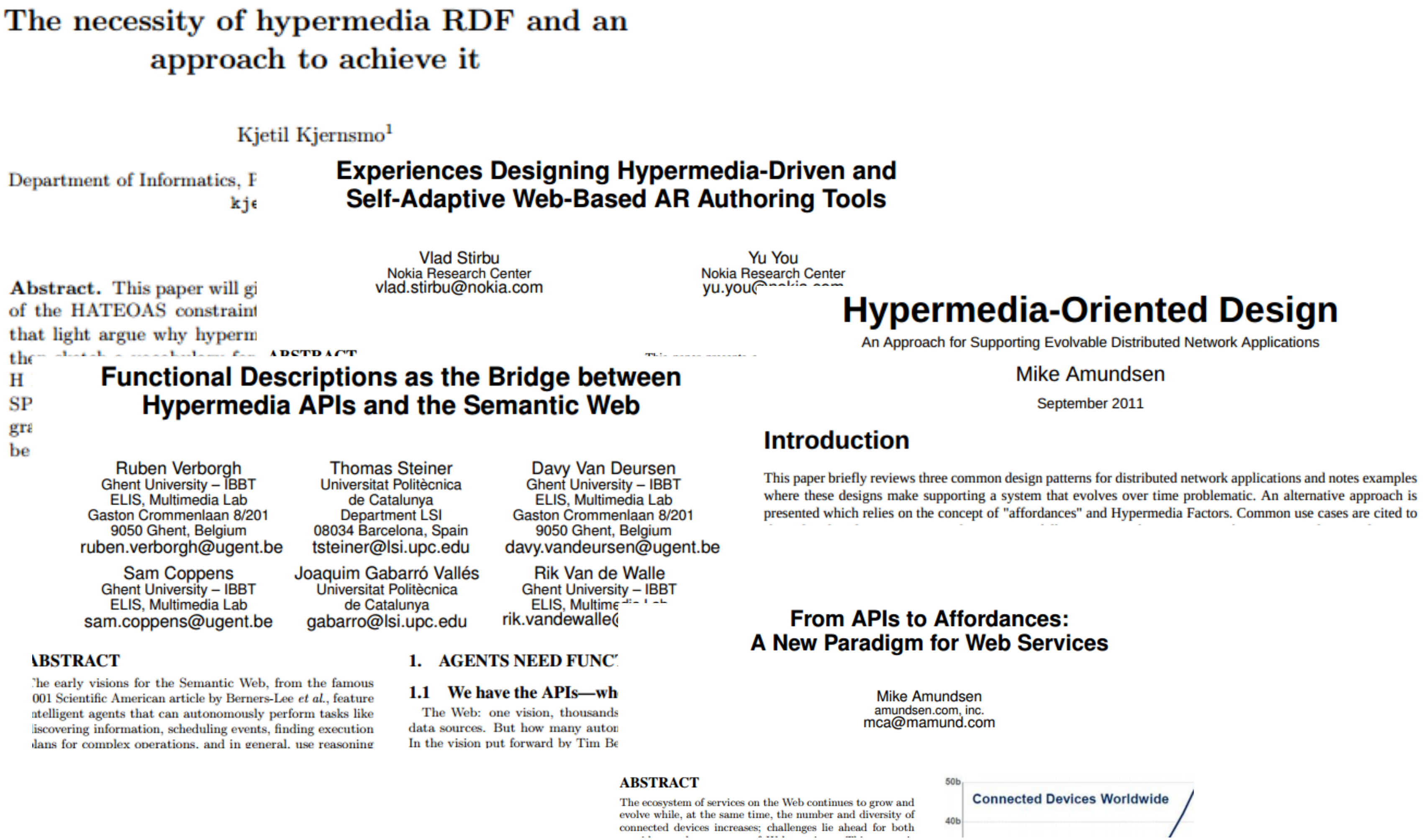
H-Factors

```
<document>
  <content>
    <section>
      <title>PHACTOR Media Type</title>
      <para>
        <text>
          This is a trivial hypermedia type implementation.
        </text>
        <LO CL="document" href="..." label="amundsen.com" />
      </para>
      <LE CL="document" href="..." label="mamund" />
      <LT CL="search" href="..." >
        <data name="p" label="Search Yahoo!" />
      </LT>
    </section>
  </content>
</document>
```

H-Factors

We now have a
shared understanding
about protocol affordances

H-Factors



The necessity of hypermedia RDF and an approach to achieve it

Kjetil Kjernsmo[1]

Department of Informatics, F
kje

Abstract. This paper will gi
of the HATEOAS constraint
that light argue why hyperm
the
H
SP
gra
be

Experiences Designing Hypermedia-Driven and Self-Adaptive Web-Based AR Authoring Tools

Vlad Stirbu
Nokia Research Center
vlad.stirbu@nokia.com

Yu You
Nokia Research Center
yu.you

Functional Descriptions as the Bridge between Hypermedia APIs and the Semantic Web

Ruben Verborgh
Ghent University – IBBT
ELIS, Multimedia Lab
Gaston Crommenlaan 8/201
9050 Ghent, Belgium
ruben.verborgh@ugent.be

Thomas Steiner
Universitat Politècnica
de Catalunya
Department LSI
08034 Barcelona, Spain
tsteiner@lsi.upc.edu

Davy Van Deursen
Ghent University – IBBT
ELIS, Multimedia Lab
Gaston Crommenlaan 8/201
9050 Ghent, Belgium
davy.vandeursen@ugent.be

Sam Coppens
Ghent University – IBBT
ELIS, Multimedia Lab
sam.coppens@ugent.be

Joaquim Gabarró Vallés
Universitat Politècnica
de Catalunya
gabarro@lsi.upc.edu

Rik Van de Walle
Ghent University – IBBT
ELIS, Multime
rik.vandewalle

ABSTRACT

he early visions for the Semantic Web, from the famous
001 Scientific American article by Berners-Lee *et al.*, feature
ntelligent agents that can autonomously perform tasks like
iscovering information, scheduling events, finding execution
lans for complex operations, and in general, use reasoning

1. AGENTS NEED FUNC

1.1 We have the APIs—wh

The Web: one vision, thousands
data sources. But how many auton
In the vision put forward by Tim Be

Hypermedia-Oriented Design

An Approach for Supporting Evolvable Distributed Network Applications

Mike Amundsen

September 2011

Introduction

This paper briefly reviews three common design patterns for distributed network applications and notes examples where these designs make supporting a system that evolves over time problematic. An alternative approach is presented which relies on the concept of "affordances" and Hypermedia Factors. Common use cases are cited to

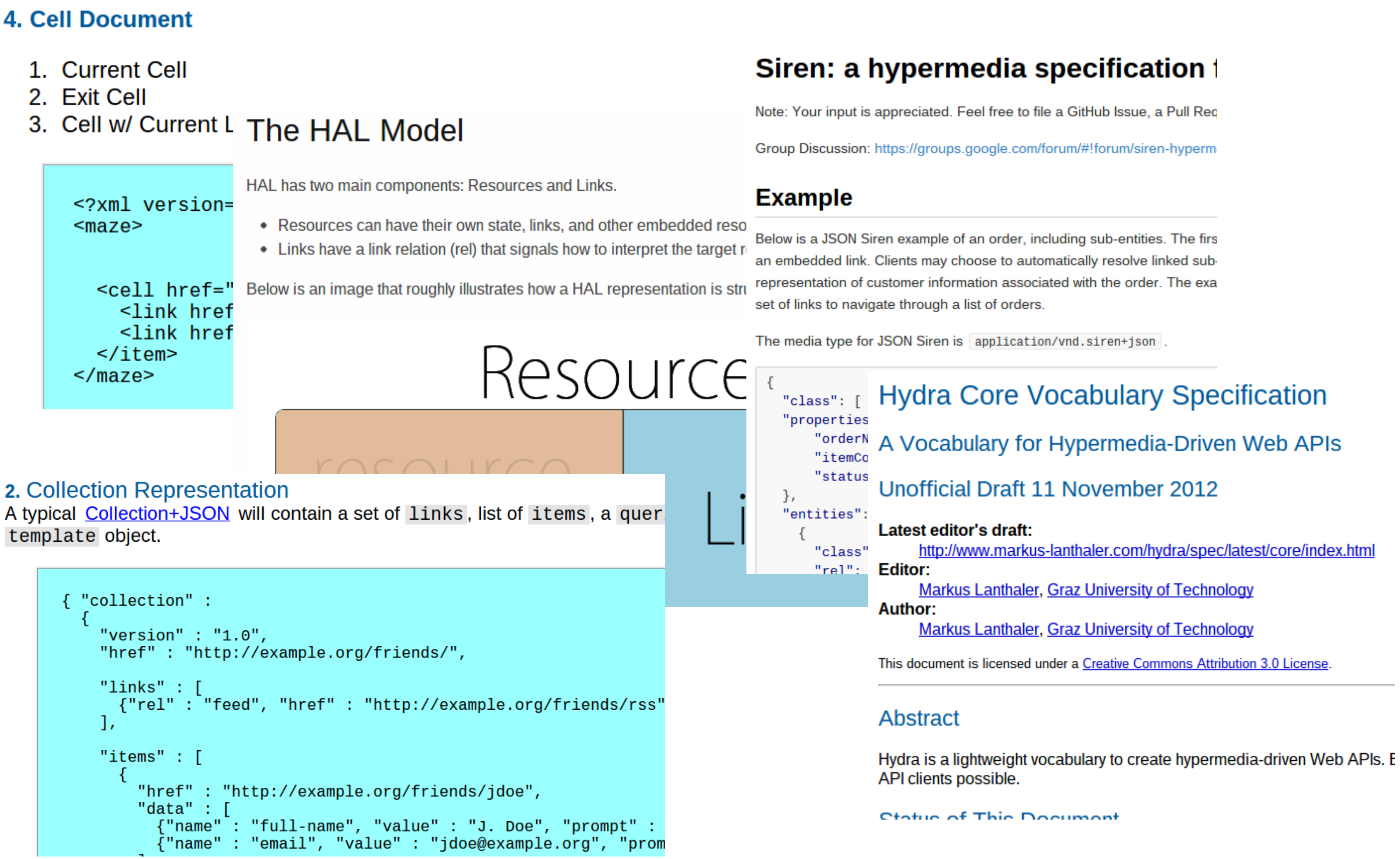
From APIs to Affordances: A New Paradigm for Web Services

Mike Amundsen
amundsen.com, inc.
mca@mamund.com

ABSTRACT

The ecosystem of services on the Web continues to grow and evolve while, at the same time, the number and diversity of connected devices increases; challenges lie ahead for both

H-Factors



4. Cell Document

1. Current Cell
2. Exit Cell
3. Cell w/ Current L

```
<?xml version=
<maze>


  <cell href="
    <link href
    <link href
  </item>
</maze>
```

The HAL Model

HAL has two main components: Resources and Links.

- Resources can have their own state, links, and other embedded reso
- Links have a link relation (rel) that signals how to interpret the target r

Below is an image that roughly illustrates how a HAL representation is stru

Siren: a hypermedia specification f

Note: Your input is appreciated. Feel free to file a GitHub Issue, a Pull Req

Group Discussion: https://groups.google.com/forum/#!forum/siren-hyperm

Example

Below is a JSON Siren example of an order, including sub-entities. The firs an embedded link. Clients may choose to automatically resolve linked sub- representation of customer information associated with the order. The exa set of links to navigate through a list of orders.

The media type for JSON Siren is `application/vnd.siren+json` .

```
{
  "class": [
  "properties
      "orderN
      "itemCo
      "status
  },
  "entities":
    {
      "class"
      "rel":
```

Hydra Core Vocabulary Specification

A Vocabulary for Hypermedia-Driven Web APIs

Unofficial Draft 11 November 2012

Latest editor's draft:
http://www.markus-lanthaler.com/hydra/spec/latest/core/index.html

Editor:
Markus Lanthaler, Graz University of Technology

Author:
Markus Lanthaler, Graz University of Technology

This document is licensed under a Creative Commons Attribution 3.0 License.

Abstract

Hydra is a lightweight vocabulary to create hypermedia-driven Web APIs. E API clients possible.

Status of This Document

2. Collection Representation

A typical Collection+JSON will contain a set of `links`, list of `items`, a `quer` `template` object.

```
{ "collection" :
  {
    "version" : "1.0",
    "href" : "http://example.org/friends/",

    "links" : [
      {"rel" : "feed", "href" : "http://example.org/friends/rss"
    ],

    "items" : [
      {
        "href" : "http://example.org/friends/jdoe",
        "data" : [
          {"name" : "full-name", "value" : "J. Doe", "prompt" :
          {"name" : "email", "value" : "jdoe@example.org", "prom
```

H-Factors

And this is all good.

H-Factors

But...

H-Factors

There's a problem here.

H-Factors

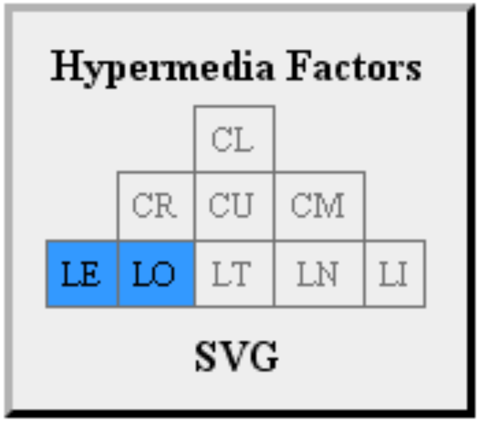
An important piece is missing.

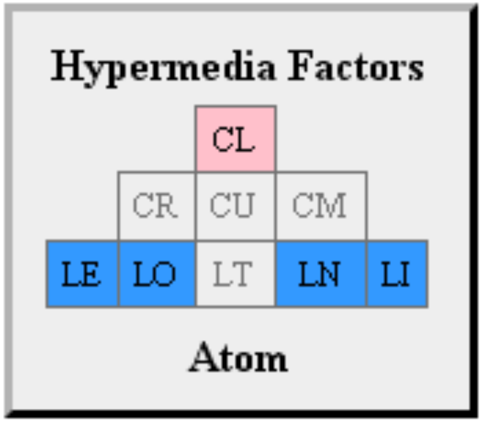
H-Factors

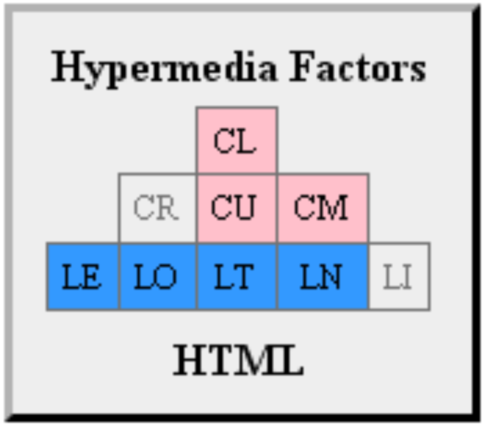
There is a "gap"
between theory and practice

H-Factors

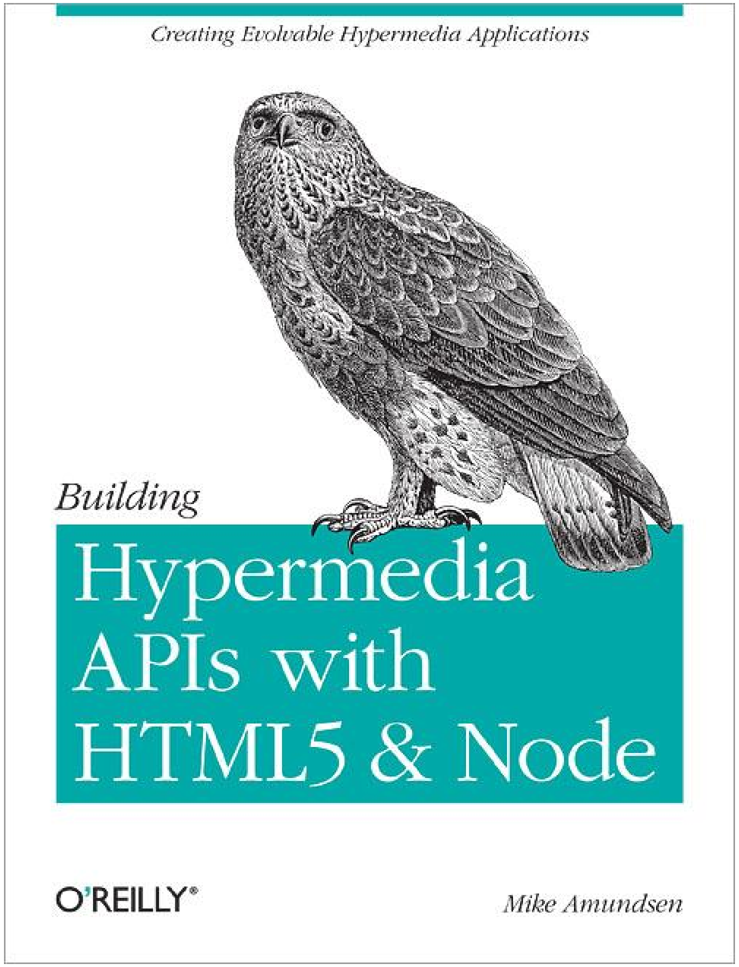
**There is a "gap"
between H-Factors and "the real world"**





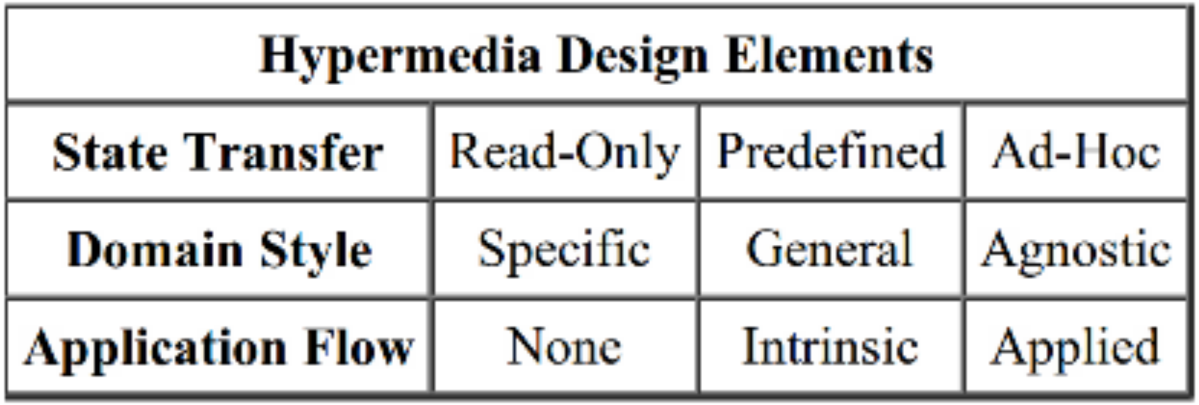


I ignored this "gap" when identifying H-Factors.

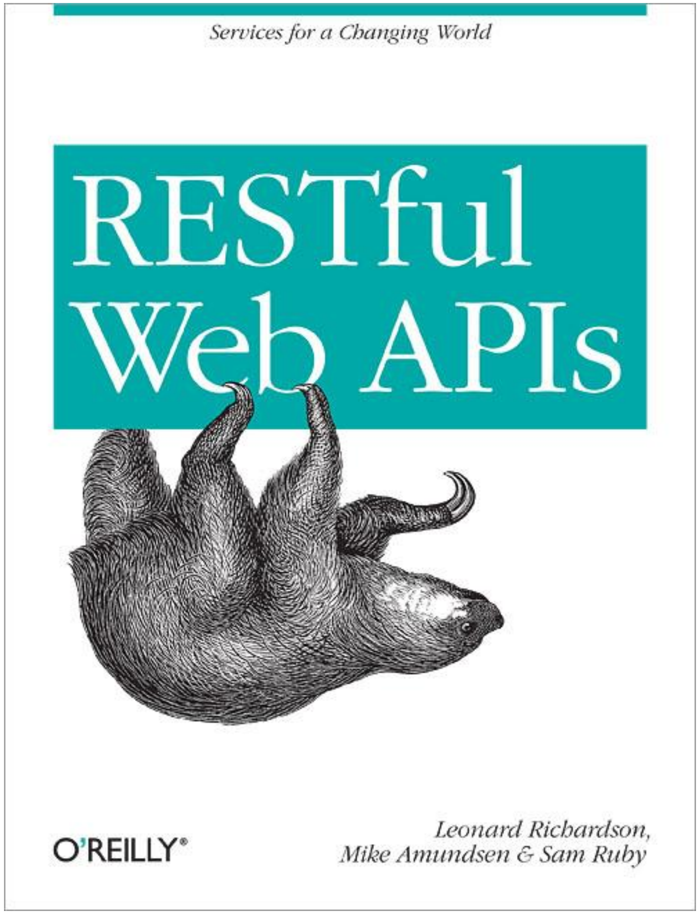

Creating Evolvable Hypermedia Applications
Building
Hypermedia APIs with HTML5 & Node
O'REILLY®
Mike Amundsen

I side-stepped the "gap" in the last book.

H-Factors



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


Services for a Changing World
RESTful Web APIs
O'REILLY
Leonard Richardson, Mike Amundsen & Sam Ruby

And this "gap" is a key theme in the next book.

H-Factors

Yep, you might say...

I left some stuff out.

The confession

The confession

I ignored the "hard" parts ...

The confession

```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <img src="..." class="logo" />

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

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

  </body>
</html>
```

The confession

What are those?

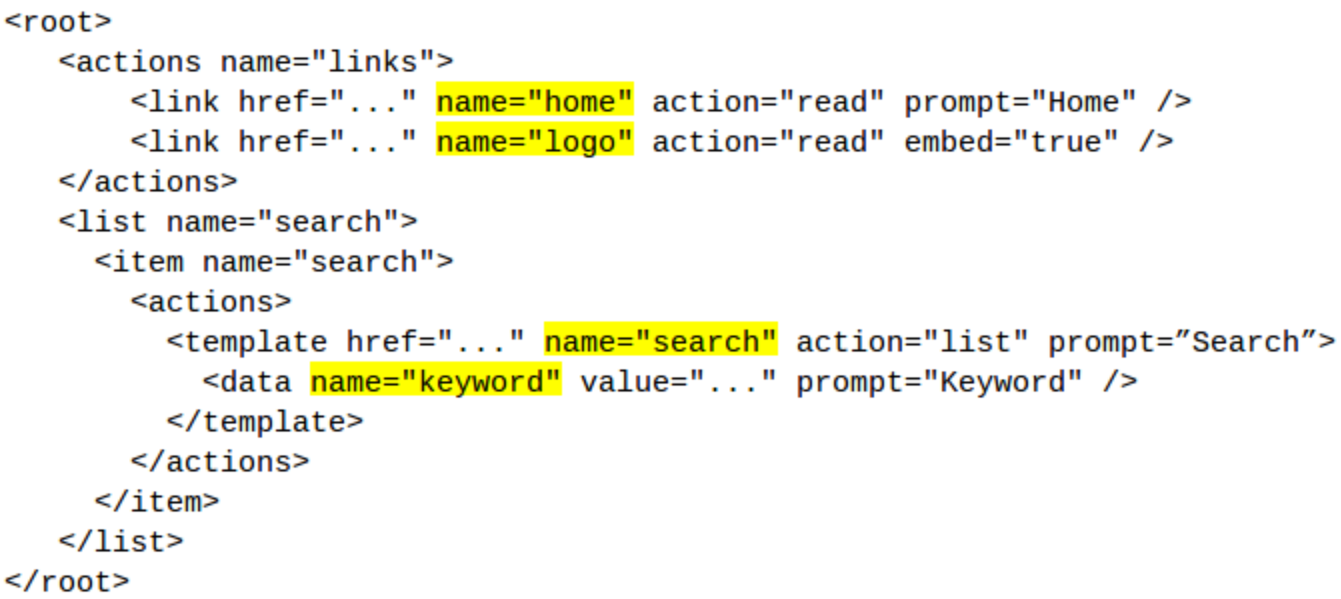
The confession

```
{
  "collection" : {
    "version" : "1.0",
    "href" : "...",

    "links" : [
      {"rel" : "home", "href" : "...", "prompt" : "Home"},
      {"rel" : "logo", "href" : "..." "embed" : true}
    ],

    "queries" : [
      {"rel" : "search", "href" : "...", "prompt" : "Search",
      "data" : [
        {"name" : "keyword", "value" : "...", "prompt" : "Keyword"}
      ]
    ]
  }
}
```

The confession

They're rather specific.

The confession

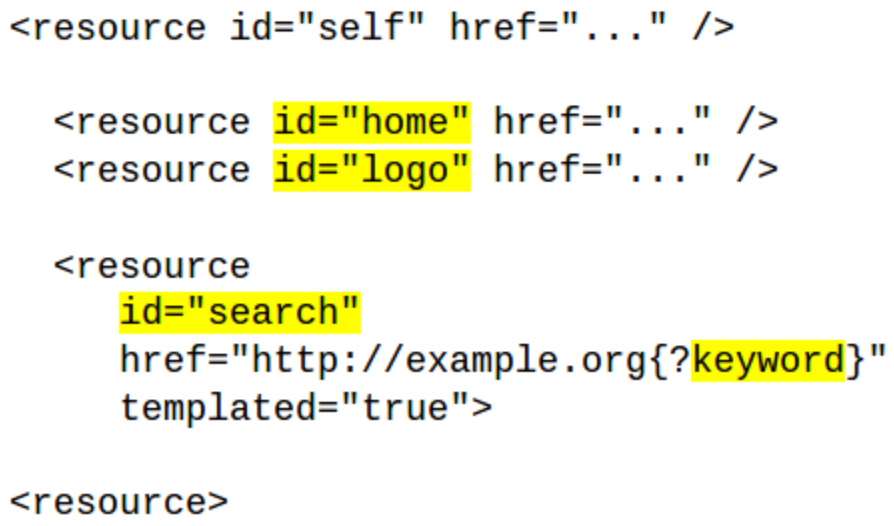


```
<root>
   <actions name="links">
       <link href="..." name="home" action="read" prompt="Home" />
       <link href="..." name="logo" action="read" embed="true" />
   </actions>
   <list name="search">
     <item name="search">
       <actions>
         <template href="..." name="search" action="list" prompt=”Search”>
           <data name="keyword" value="..." prompt="Keyword" />
         </template>
       </actions>
     </item>
   </list>
</root>
```

The confession

affordances?

The confession



```
<resource id="self" href="..." />

  <resource id="home" href="..." />
  <resource id="logo" href="..." />

  <resource
     id="search"
     href="http://example.org{?keyword}"
     templated="true">

<resource>
```

The confession

Well, they are not protocol affordances

The confession

I call them
application affordances

The confession

But there is another name for these...

Vocabularies

Srsly?

Vocabularies

On the Web,
representations contain
protocol AND application
affordances

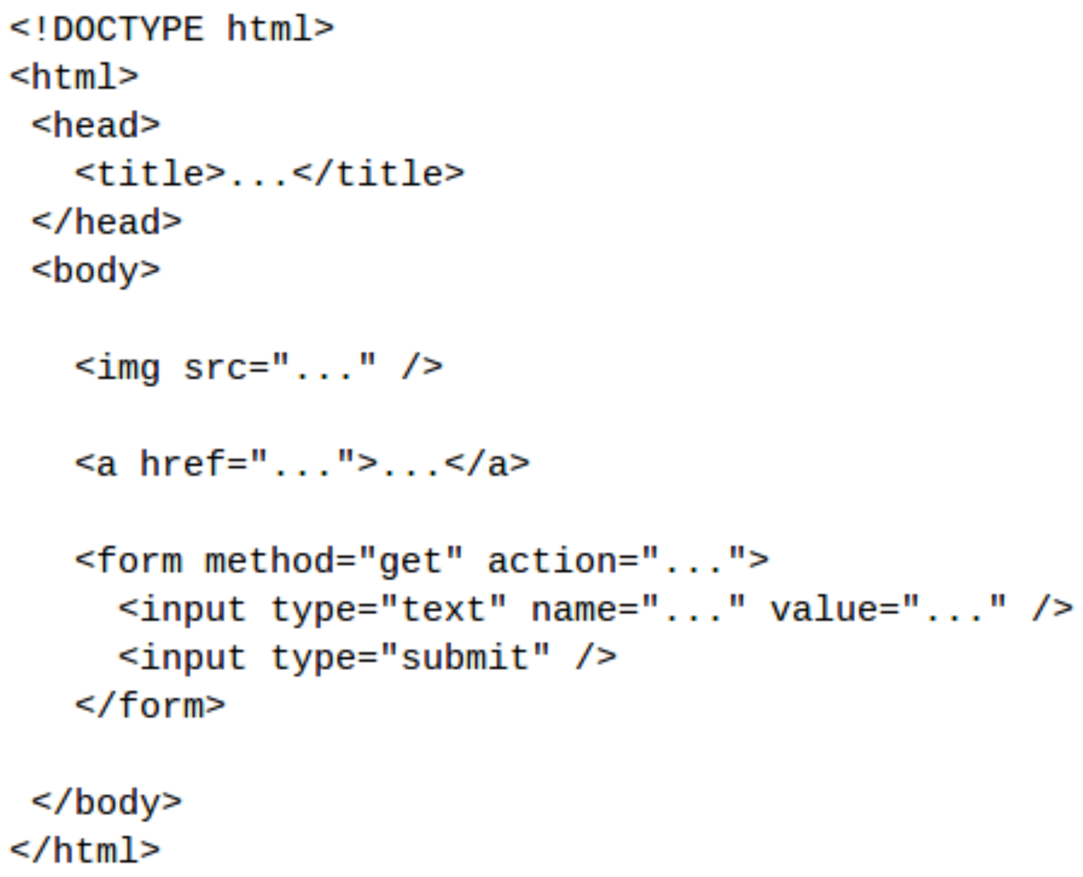
Vocabularies

We share understanding
at the application-level, too.

Vocabularies

Well, "we" means us humans.

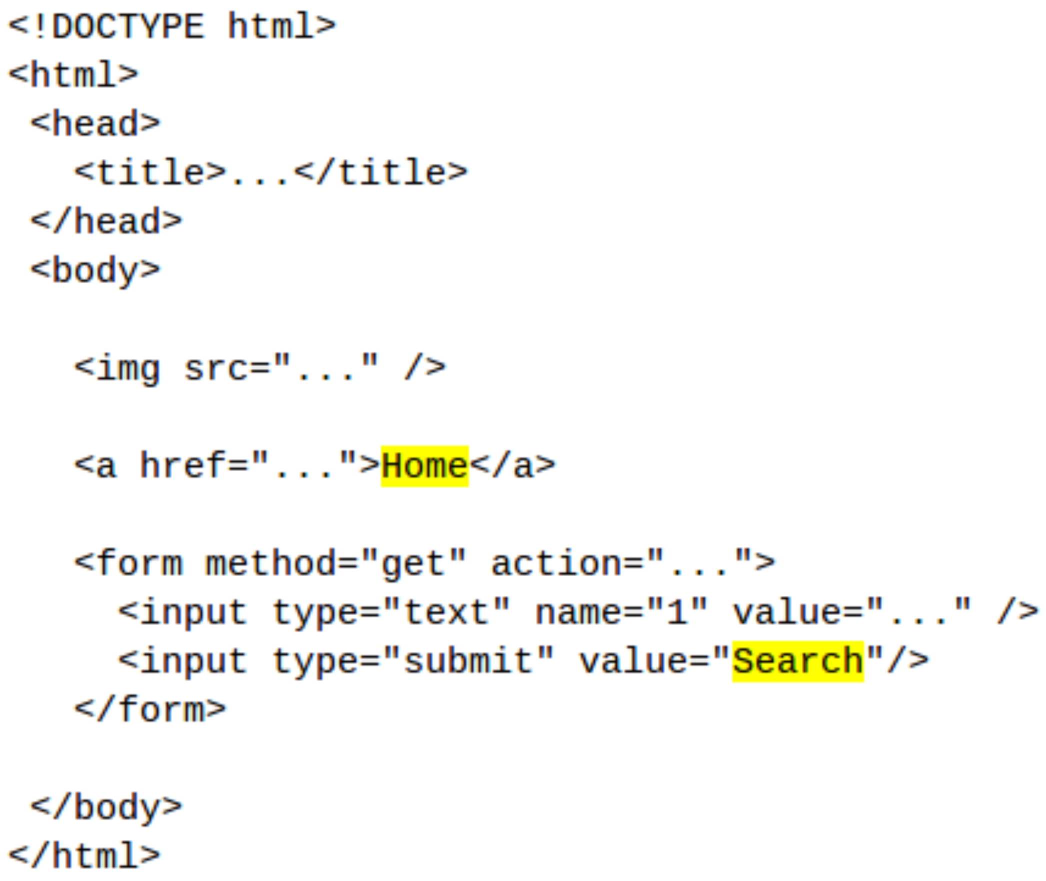
Vocabularies



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

   <img src="..." />

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

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

 </body>
</html>
```

Do we share app-level understanding?

Vocabularies



```
<!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>
```

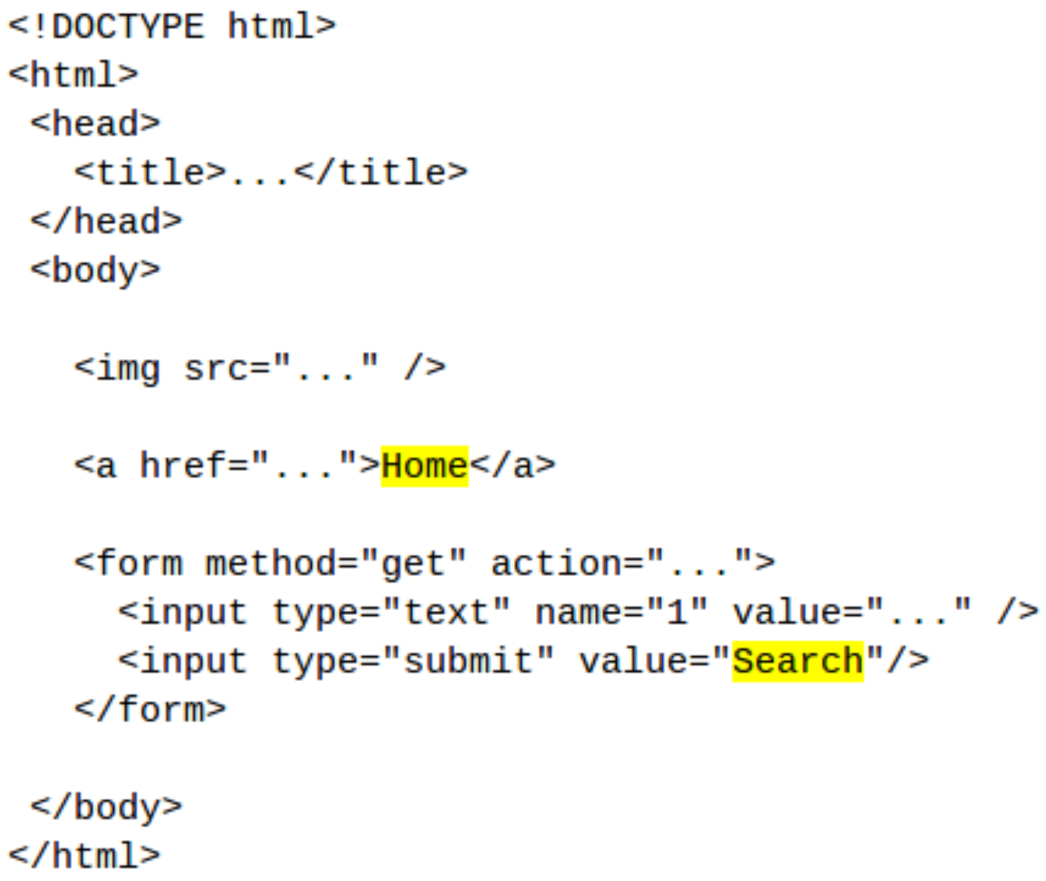
How about now?

Vocabularies

```
<!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>
```

How about now?

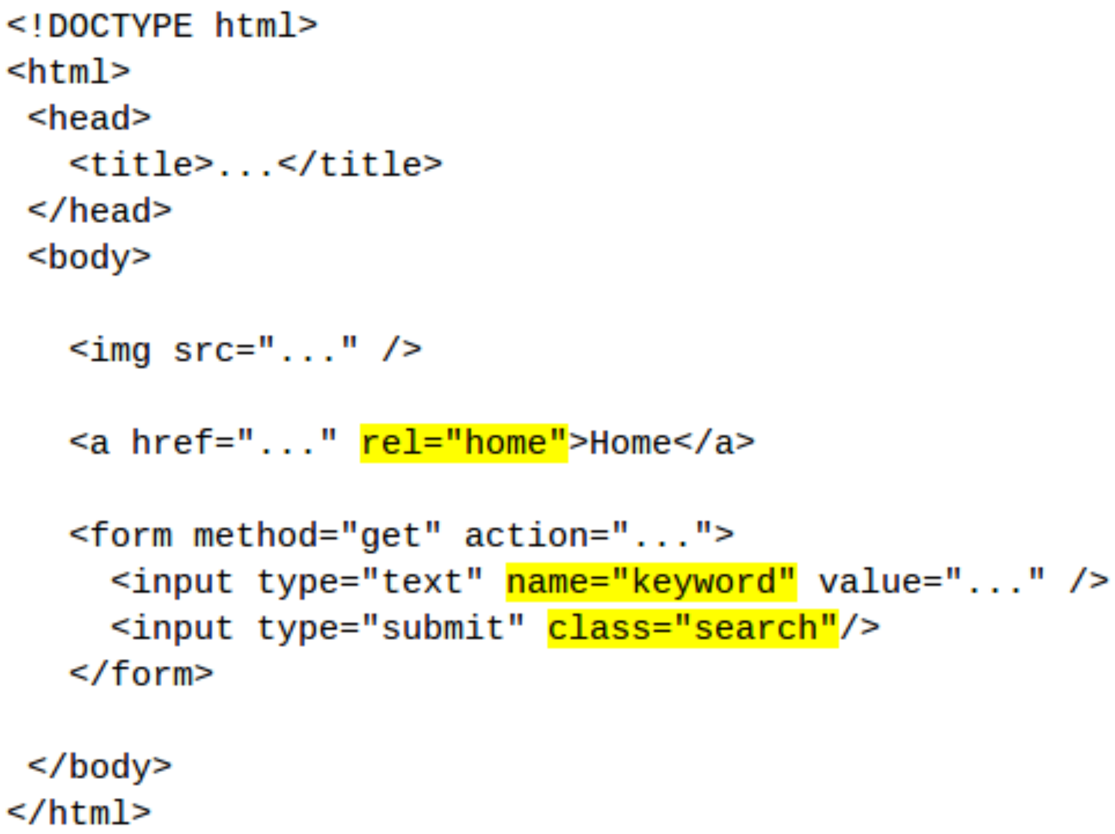
Vocabularies



```
<!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>
```

Human app-level affordances

Vocabularies



```
<!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>
```

Machine app-level affordances

Vocabularies

Of course, we've known this
for quite a while.

Vocabularies

schema.org

dublin core

VoID

There has been quite a bit of work on vocabularies

activity streams

microformats

Vocabularies

Vocabularies can provide
shared understanding
of the
application-specific affordances

Vocabularies

And this is all good.

Vocabularies

But...

Vocabularies

There's a problem here.

Vocabularies

An important piece is missing.

Vocabularies

There is a "gap"
between theory and practice

Vocabularies

**There is a "gap"
between Vocabularies and "the real world"**

Vocabularies

state

Vocabularies only model the "what"

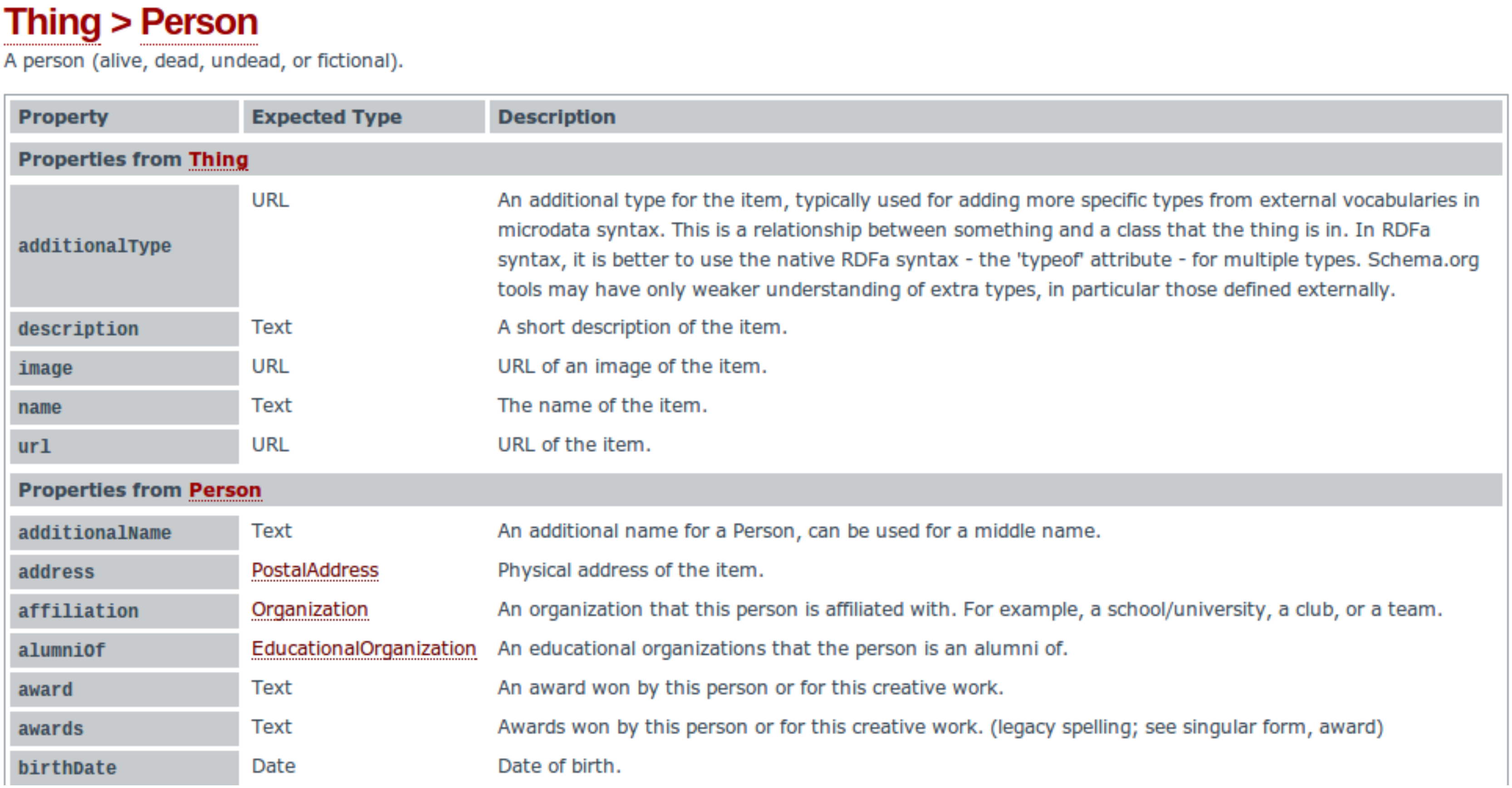
Vocabularies

state

**Vocabularies only model
the "what"
not the "how"**

transitions

Vocabularies



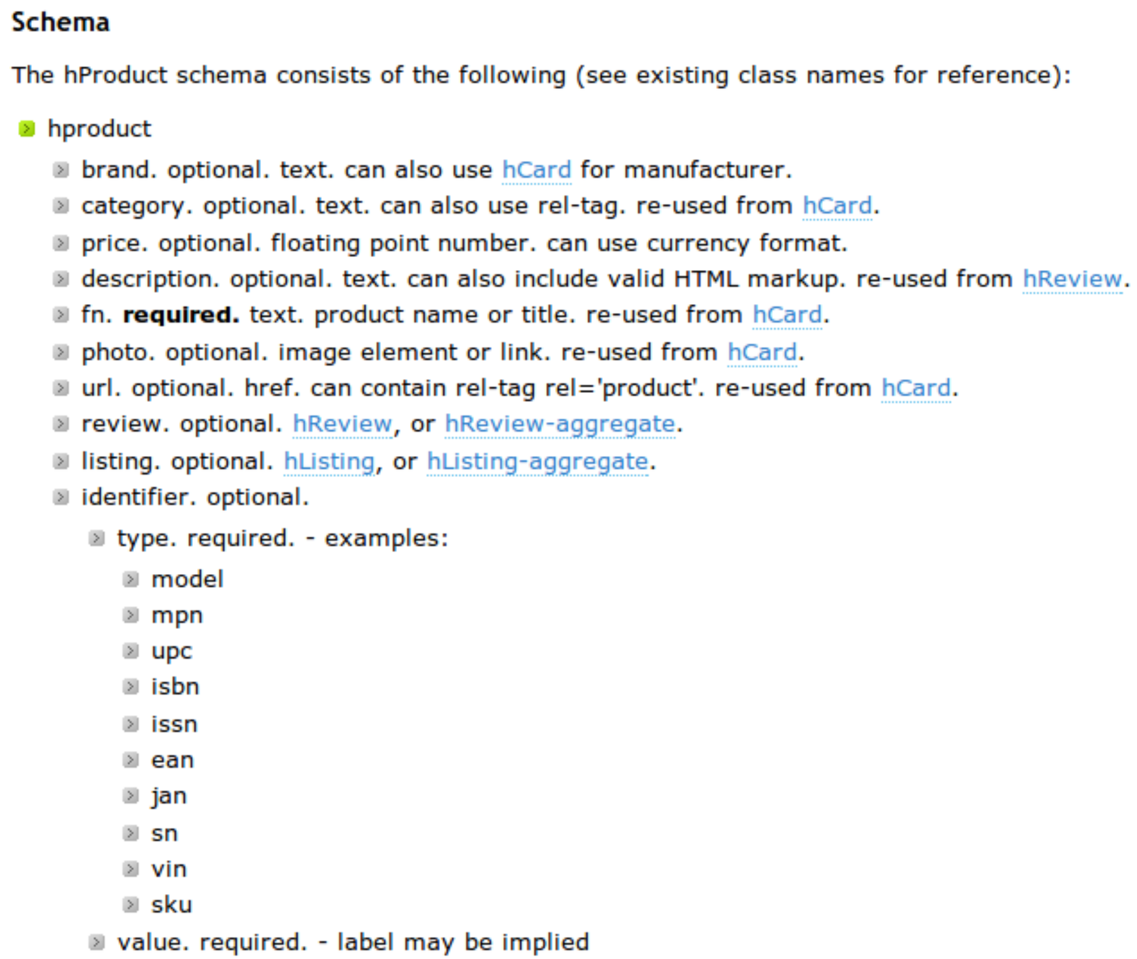
Thing > Person

A person (alive, dead, undead, or fictional).

Property	Expected Type	Description
Properties from Thing		
additionalType	URL	An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the native RDFa syntax - the 'typeof' attribute - for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	URL	URL of the item.
Properties from Person		
additionalName	Text	An additional name for a Person, can be used for a middle name.
address	PostalAddress	Physical address of the item.
affiliation	Organization	An organization that this person is affiliated with. For example, a school/university, a club, or a team.
alumniOf	EducationalOrganization	An educational organizations that the person is an alumni of.
award	Text	An award won by this person or for this creative work.
awards	Text	Awards won by this person or for this creative work. (legacy spelling; see singular form, award)
birthDate	Date	Date of birth.

I know what a Person is, but how can I interact with it?

Vocabularies



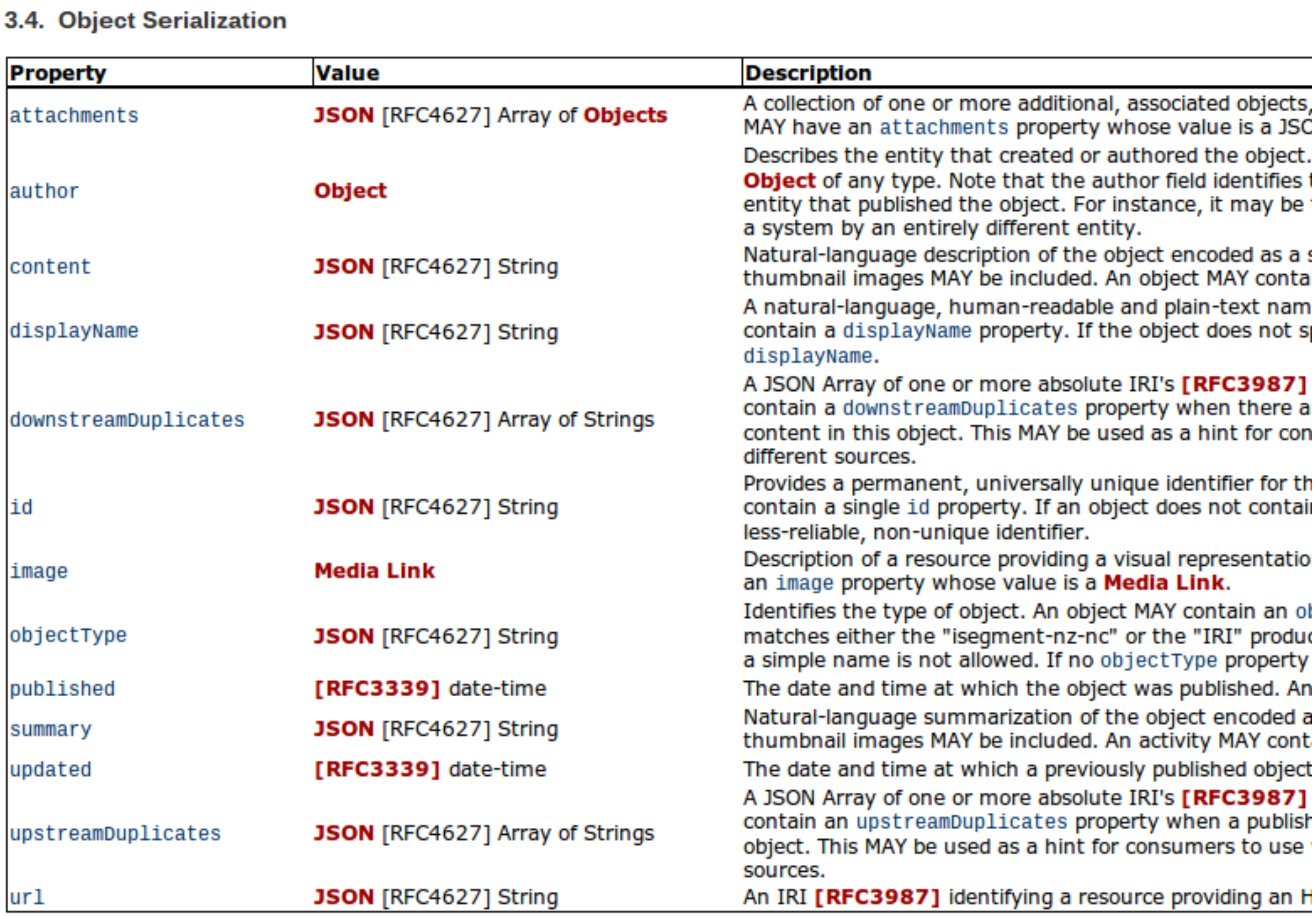
Schema

The hProduct schema consists of the following (see existing class names for reference):

- hproduct
  - brand. optional. text. can also use hCard for manufacturer.
  - category. optional. text. can also use rel-tag. re-used from hCard.
  - price. optional. floating point number. can use currency format.
  - description. optional. text. can also include valid HTML markup. re-used from hReview.
  - fn. **required.** text. product name or title. re-used from hCard.
  - photo. optional. image element or link. re-used from hCard.
  - url. optional. href. can contain rel-tag rel='product'. re-used from hCard.
  - review. optional. hReview, or hReview-aggregate.
  - listing. optional. hListing, or hListing-aggregate.
  - identifier. optional.
    - type. required. - examples:
      - model
      - mpn
      - upc
      - isbn
      - issn
      - ean
      - jan
      - sn
      - vin
      - sku
    - value. required. - label may be implied

I know what an hProduct is, but how can I interact with it?

Vocabularies



3.4. Object Serialization

Property	Value	Description
attachments	**JSON** [RFC4627] Array of **Objects**	A collection of one or more additional, associated objects, MAY have an attachments property whose value is a JSC
author	**Object**	Describes the entity that created or authored the object. **Object** of any type. Note that the author field identifies t entity that published the object. For instance, it may be a system by an entirely different entity.
content	**JSON** [RFC4627] String	Natural-language description of the object encoded as a s thumbnail images MAY be included. An object MAY conta
displayName	**JSON** [RFC4627] String	A natural-language, human-readable and plain-text nam contain a displayName property. If the object does not sp displayName.
downstreamDuplicates	**JSON** [RFC4627] Array of Strings	A JSON Array of one or more absolute IRI's **[RFC3987]** contain a downstreamDuplicates property when there a content in this object. This MAY be used as a hint for con different sources.
id	**JSON** [RFC4627] String	Provides a permanent, universally unique identifier for th contain a single id property. If an object does not contai less-reliable, non-unique identifier.
image	**Media Link**	Description of a resource providing a visual representatio an image property whose value is a **Media Link**.
objectType	**JSON** [RFC4627] String	Identifies the type of object. An object MAY contain an ob matches either the "isegment-nz-nc" or the "IRI" produc a simple name is not allowed. If no objectType property
published	**[RFC3339]** date-time	The date and time at which the object was published. An
summary	**JSON** [RFC4627] String	Natural-language summarization of the object encoded a thumbnail images MAY be included. An activity MAY cont
updated	**[RFC3339]** date-time	The date and time at which a previously published object
upstreamDuplicates	**JSON** [RFC4627] Array of Strings	A JSON Array of one or more absolute IRI's **[RFC3987]** contain an upstreamDuplicates property when a publish object. This MAY be used as a hint for consumers to use sources.
url	**JSON** [RFC4627] String	An IRI **[RFC3987]** identifying a resource providing an H

I know what an ActivityStream is, but how can I interact with it?

The payoff

Where I make amends

The payoff

state

What if we combined
the "what" of vocabularies

The payoff

state

What if we combined
the "what" of vocabularies
with
the "how" of protocols?

transitions

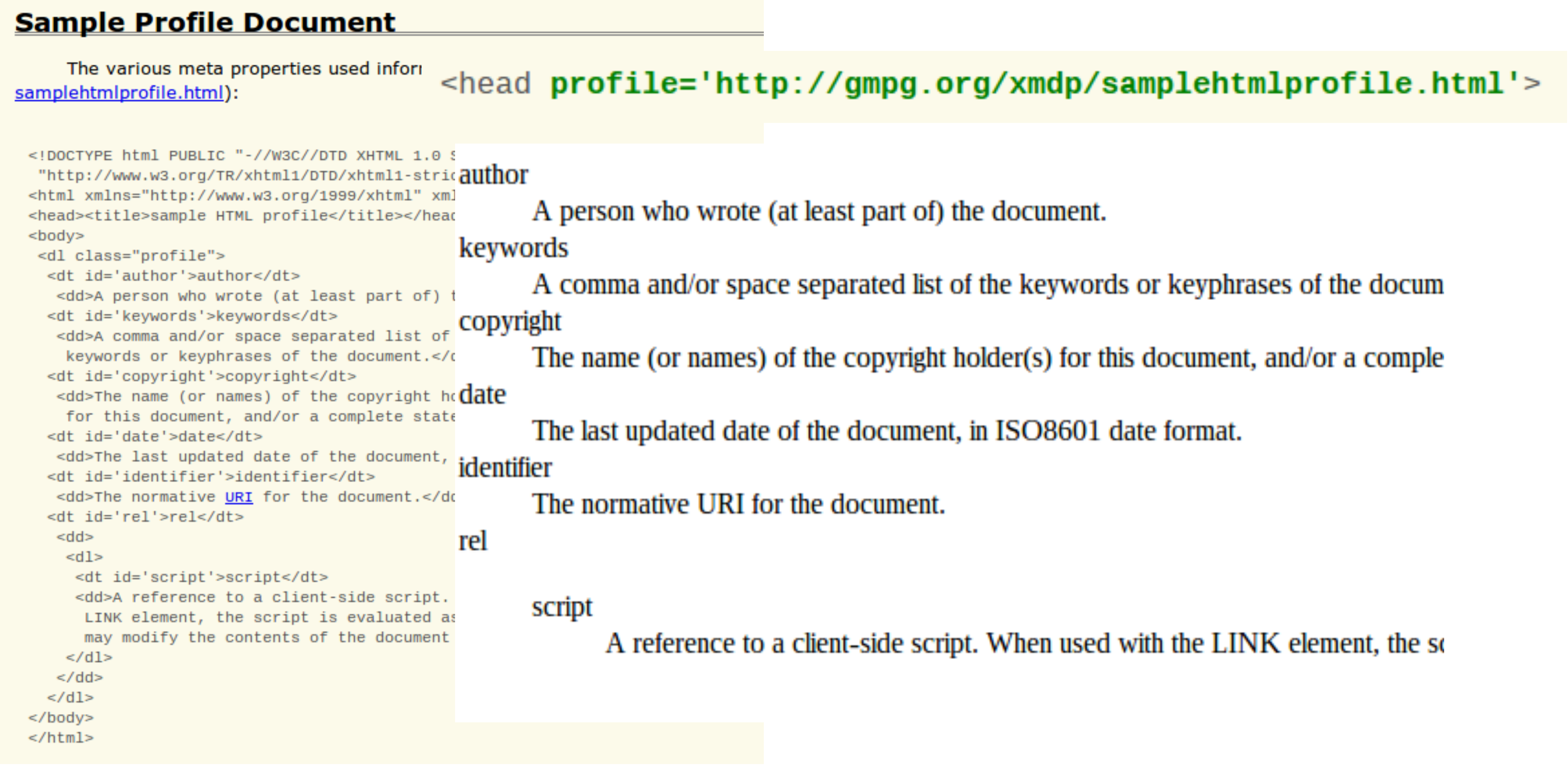
The payoff

state

What would that look like?

transitions

The payoff



Sample Profile Document

The various meta properties used infor
samplehtmlprofile.html):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 S
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-stric
<html xmlns="http://www.w3.org/1999/xhtml" xml
<head><title>sample HTML profile</title></head
<body>
 <dl class="profile">
  <dt id='author'>author</dt>
   <dd>A person who wrote (at least part of) t
  <dt id='keywords'>keywords</dt>
   <dd>A comma and/or space separated list of
    keywords or keyphrases of the document.</c
  <dt id='copyright'>copyright</dt>
   <dd>The name (or names) of the copyright hc
    for this document, and/or a complete state
  <dt id='date'>date</dt>
   <dd>The last updated date of the document,
  <dt id='identifier'>identifier</dt>
   <dd>The normative URI for the document.</dc
  <dt id='rel'>rel</dt>
   <dd>
    <dl>
     <dt id='script'>script</dt>
     <dd>A reference to a client-side script.
      LINK element, the script is evaluated as
      may modify the contents of the document
    </dl>
   </dd>
  </dl>
</body>
</html>
```

```
<head profile='http://gmpg.org/xmdp/samplehtmlprofile.html'>
```

author
: A person who wrote (at least part of) the document.

keywords
: A comma and/or space separated list of the keywords or keyphrases of the docum

copyright
: The name (or names) of the copyright holder(s) for this document, and/or a comple

date
: The last updated date of the document, in ISO8601 date format.

identifier
: The normative URI for the document.

rel

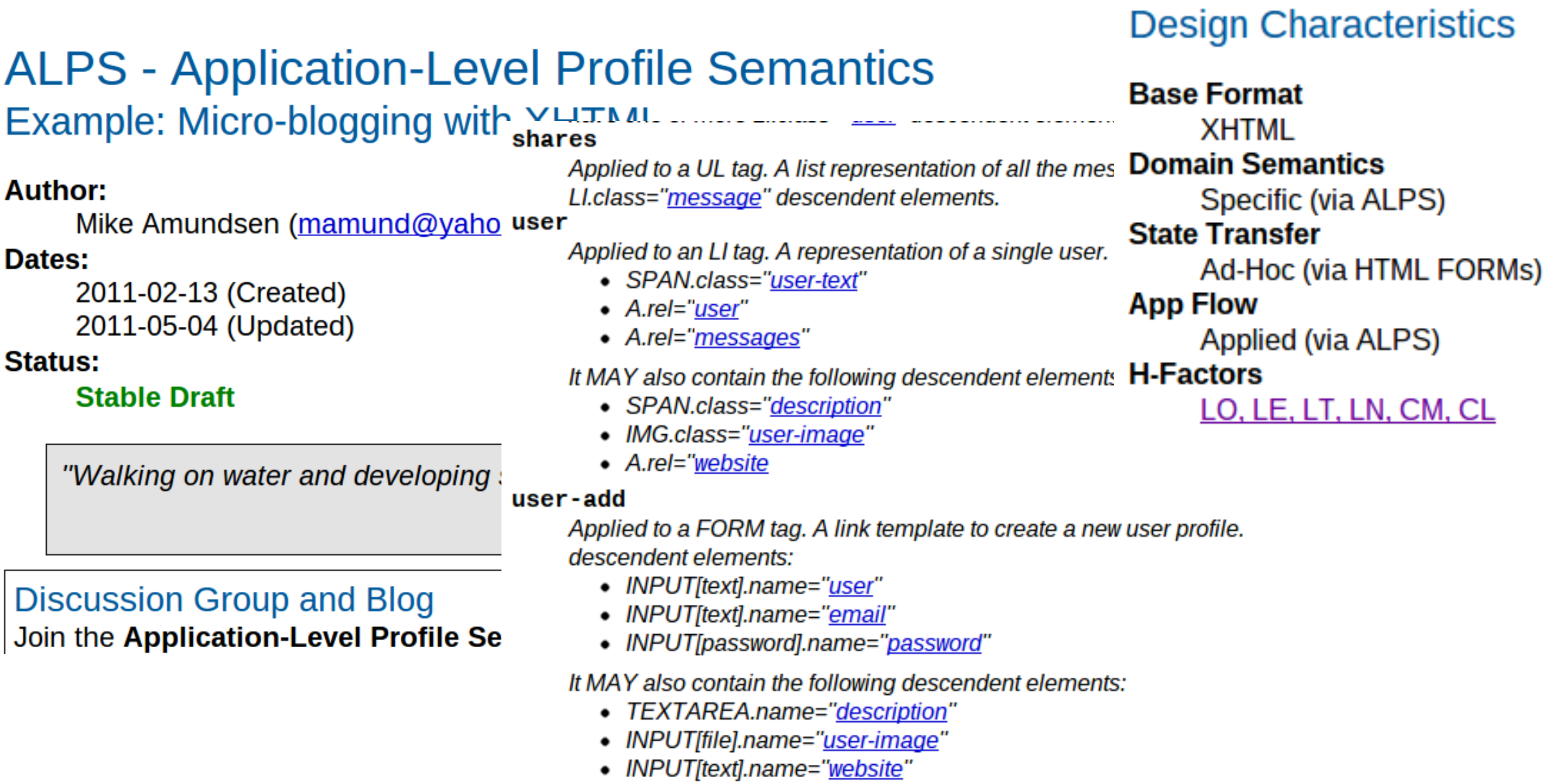
  script
  : A reference to a client-side script. When used with the LINK element, the sc

XHTML Meta Data Profiles (XMDP)
Tantek Celik, 2003

The payoff

A few years on, a similar approach...

The payoff



ALPS - Application-Level Profile Semantics

Example: Micro-blogging with XHTML

Author:
Mike Amundsen (mamund@yaho

Dates:
2011-02-13 (Created)
2011-05-04 (Updated)

Status:
Stable Draft

> *"Walking on water and developing*

Discussion Group and Blog
Join the **Application-Level Profile Se**

`shares`
Applied to a UL tag. A list representation of all the mes LI.class="message" descendent elements.

`user`
Applied to an LI tag. A representation of a single user.

- *SPAN.class="user-text"*
- *A.rel="user"*
- *A.rel="messages"*

It MAY also contain the following descendent elements

- *SPAN.class="description"*
- *IMG.class="user-image"*
- *A.rel="website*

`user-add`
Applied to a FORM tag. A link template to create a new user profile. descendent elements:

- *INPUT[text].name="user"*
- *INPUT[text].name="email"*
- *INPUT[password].name="password"*

It MAY also contain the following descendent elements:

- *TEXTAREA.name="description"*
- *INPUT[file].name="user-image"*
- *INPUT[text].name="website"*

Design Characteristics

Base Format
XHTML

Domain Semantics
Specific (via ALPS)

State Transfer
Ad-Hoc (via HTML FORMs)

App Flow
Applied (via ALPS)

H-Factors
LO, LE, LT, LN, CM, CL

ALPS - Microblogging with XHTML
Mike Amundsen, 2011

The payoff

ALPS includes both state and transitions

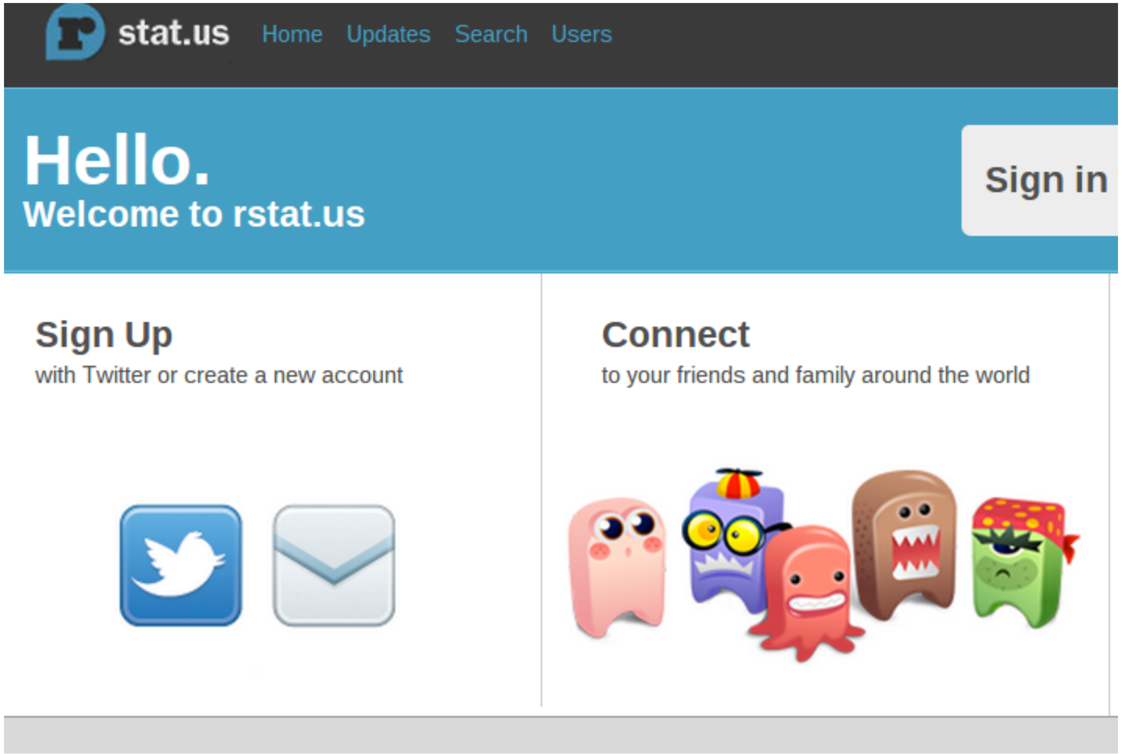
The payoff

ALPS(mb) hackathon
at 2011 RESTFest.

Initial experiment was a success.

Independently built apps could interop.

The payoff



stat.us Home Updates Search Users

Hello.
Welcome to rstat.us

Sign in

Sign Up
with Twitter or create a new account

Connect
to your friends and family around the world

Microblogging site **rstatus** implements ALPS(mb) in 2012

The payoff

**ALPS(mb) was a good idea
but we can do better.**

The payoff

**How about a profile spec
that includes both
state and transitions
that works for
a wide range
of media types?**

Application-Level Profile Semantics

The sequel

ALPS

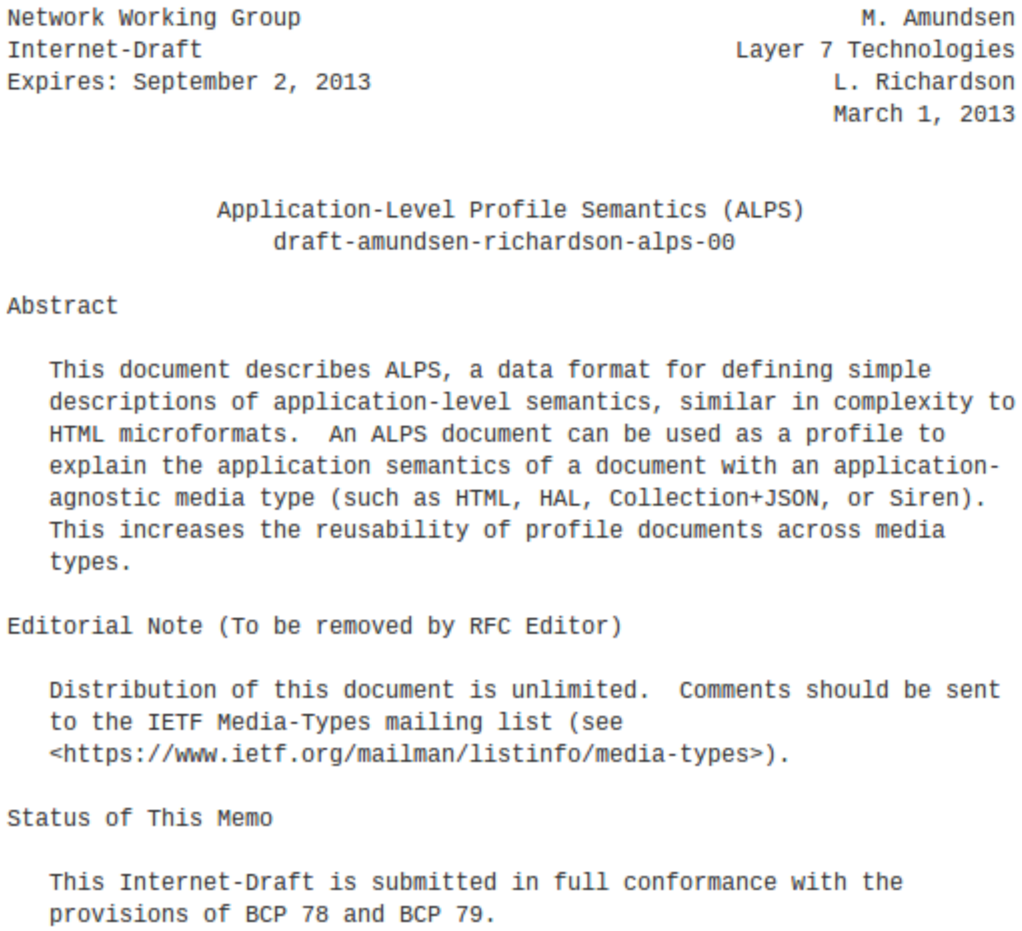
Warning!
What follows is early-stage, tentative design

ALPS

Design and register two media types for describing problem domains.

application/alps+xml (or +json)

ALPS



Network Working Group
M. Amundsen

Internet-Draft
Expires: September 2, 2013

M. Amundsen
Layer 7 Technologies
L. Richardson
March 1, 2013

Application-Level Profile Semantics (ALPS)
draft-amundsen-richardson-alps-00

Abstract

This document describes ALPS, a data format for defining simple descriptions of application-level semantics, similar in complexity to HTML microformats. An ALPS document can be used as a profile to explain the application semantics of a document with an application-agnostic media type (such as HTML, HAL, Collection+JSON, or Siren). This increases the reusability of profile documents across media types.

Editorial Note (To be removed by RFC Editor)

Distribution of this document is unlimited. Comments should be sent to the IETF Media-Types mailing list (see <https://www.ietf.org/mailman/listinfo/media-types>).

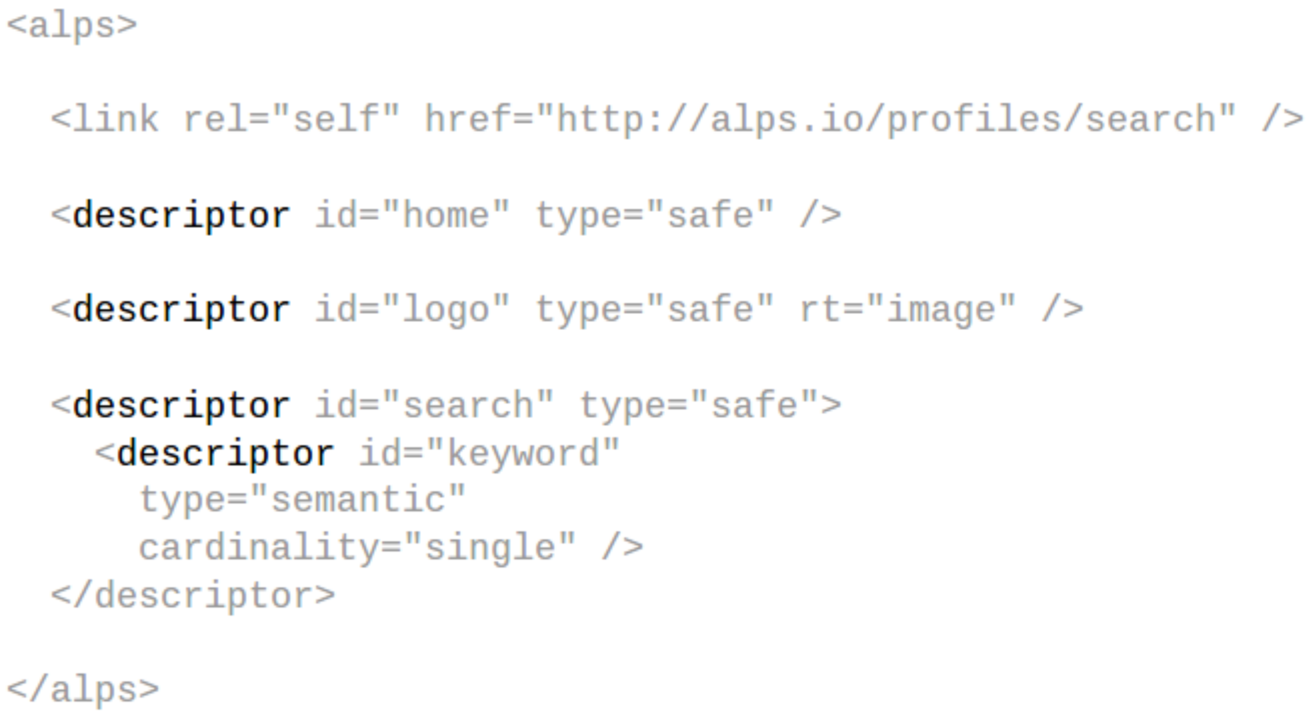
Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

ALPS

```
<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>
```

ALPS



```
<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>
```

ALPS

```
<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>
```

ALPS

Four possible descriptor types:

1. semantic (data)
2. safe (HTTP.GET)
3. unsafe (HTTP.POST)
4. idempotent (HTTP.PUT & HTTP.DELETE)

ALPS

```
<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>
```

ALPS

ALPS 'id' properties become representation identifiers

- @class, @rel, @name (HTML)
- @rel, @name (Cj, HAL)
- @rel (Atom)

ALPS

Apply these semantics to a wide range of existing media types

ALPS

```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <img src="..." class="logo" />

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

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

  </body>
</html>
```

ALPS

There will be a set of rules for applying ALPS semantics to each media type.

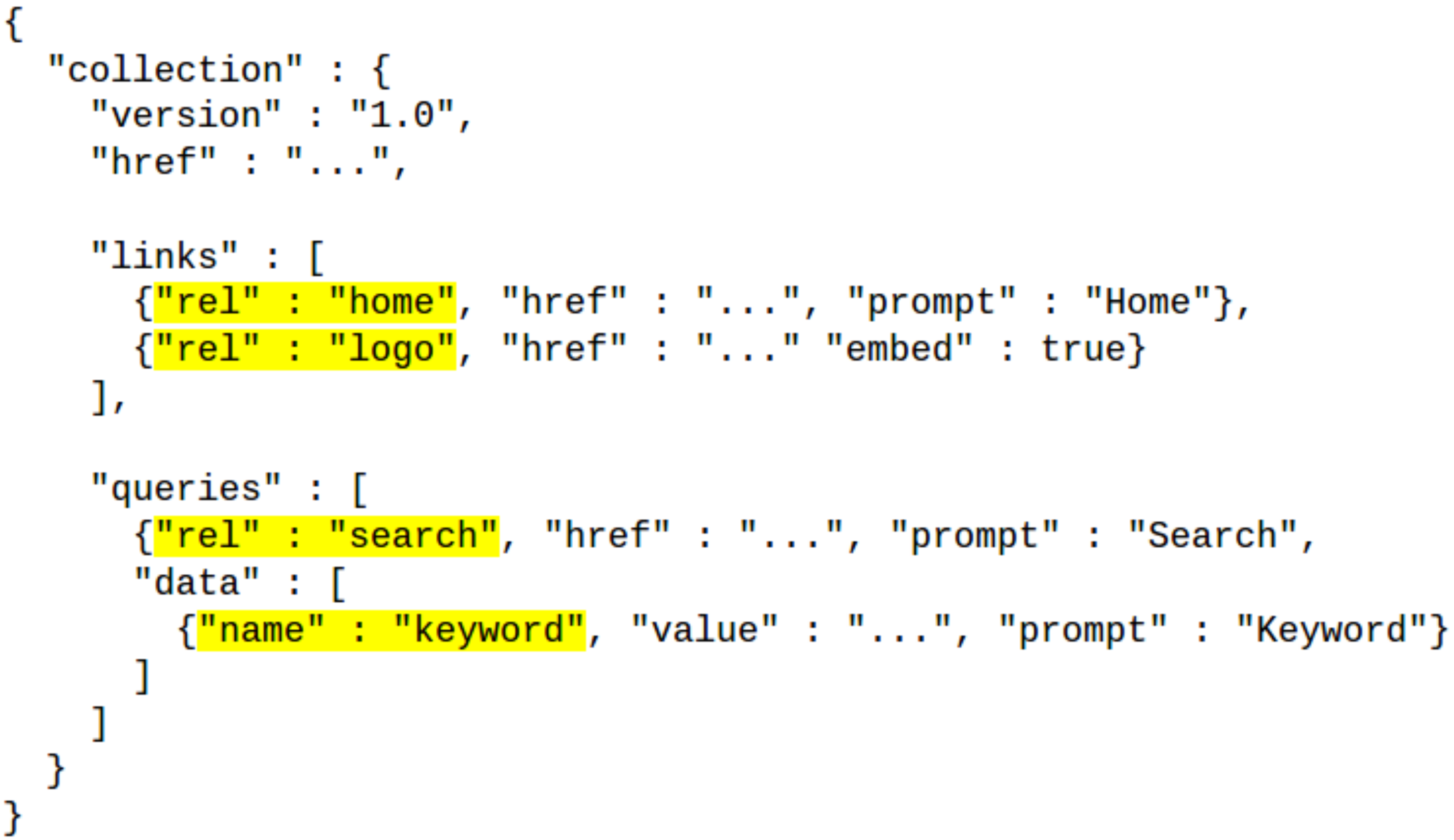
ALPS

```
<!DOCTYPE html>
<html>
  <head>
    <title>H-Factor Search</title>
  </head>
  <body>

    <img src="..." class="logo" />

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

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

  </body>
</html>
```

ALPS

By applying semantic descriptors consistently...

ALPS



```
{
  "collection" : {
    "version" : "1.0",
    "href" : "...",

    "links" : [
      {"rel" : "home", "href" : "...", "prompt" : "Home"},
      {"rel" : "logo", "href" : "..." "embed" : true}
    ],

    "queries" : [
      {"rel" : "search", "href" : "...", "prompt" : "Search",
      "data" : [
        {"name" : "keyword", "value" : "...", "prompt" : "Keyword"}
      ]
    ]
  }
}
```

ALPS

selecting
an implementation media type
can be independent of the
state and transition
semantics.

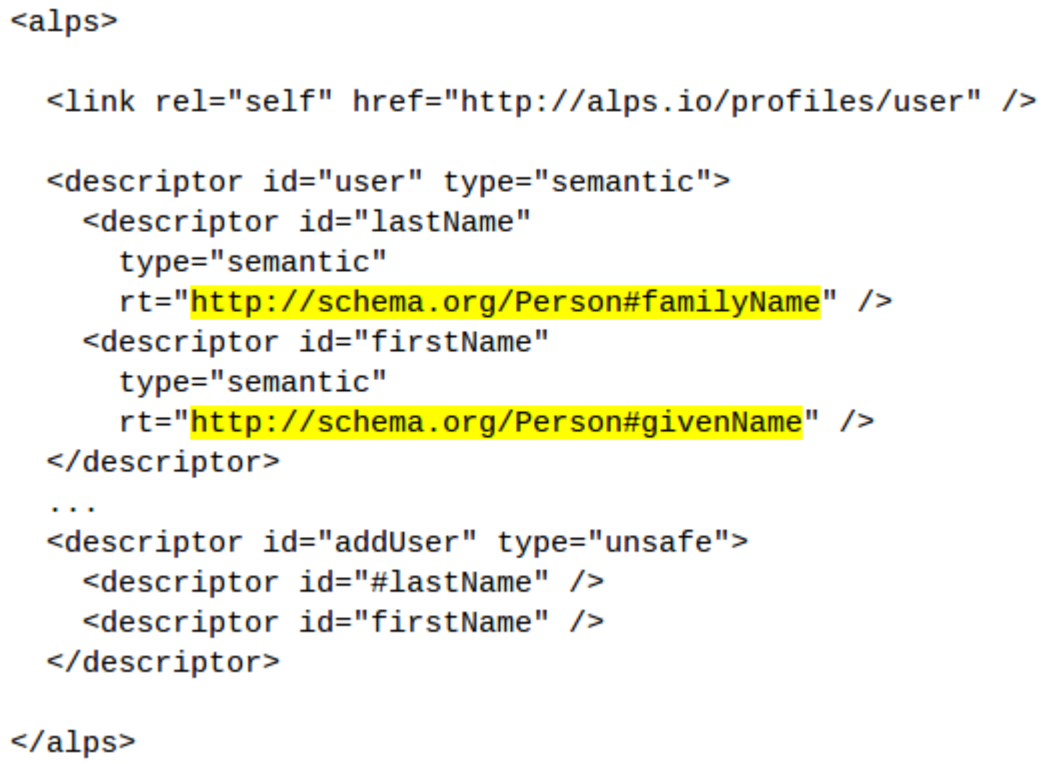
ALPS

```
<resource id="self" href="..." />

  <resource id="home" href="..." />
  <resource id="logo" href="..." />

  <resource
     id="search"
     href="http://example.org{?keyword}"
     templated="true">

<resource>
```

ALPS

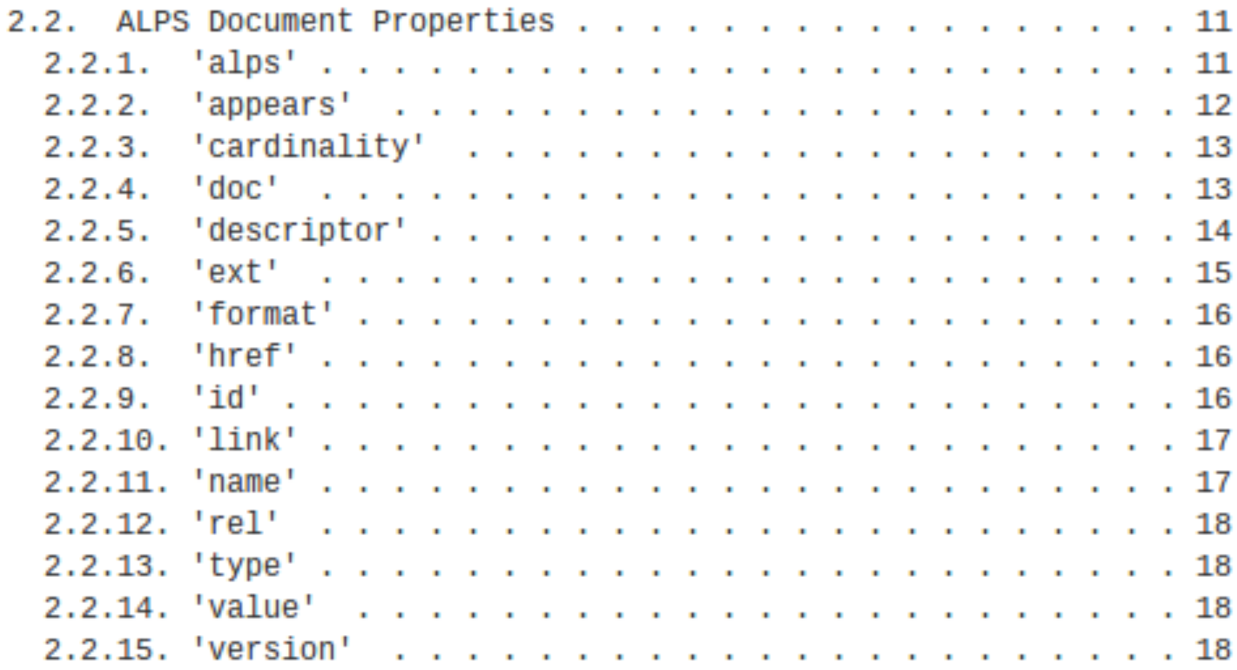
selecting a reference vocabulary can be independent of the protocol semantics

ALPS



```
<alps>

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

  <descriptor id="user" type="semantic">
    <descriptor id="lastName"
      type="semantic"
      rt="http://schema.org/Person#familyName" />
    <descriptor id="firstName"
      type="semantic"
      rt="http://schema.org/Person#givenName" />
  </descriptor>
  ...
  <descriptor id="addUser" type="unsafe">
    <descriptor id="#lastName" />
    <descriptor id="firstName" />
  </descriptor>

</alps>
```

ALPS



ALPS spec will provide several markup elements

ALPS

**Now we have a way
to share understanding**

ALPS

Now we have a way to share understanding about a problem domain

ALPS

Now we have a way to share understanding about a problem domain

Not a service document (WSDL)

ALPS

Now we have a way to share understanding about a problem domain

Not a discovery document (Google, @mnot)

ALPS

Now we have a way to share understanding about a problem domain

Not a hypermedia type (HAL, Cj, Siren, etc.)

ALPS

Now we have a way to share understanding about a problem domain

Not a vocabulary repository

ALPS

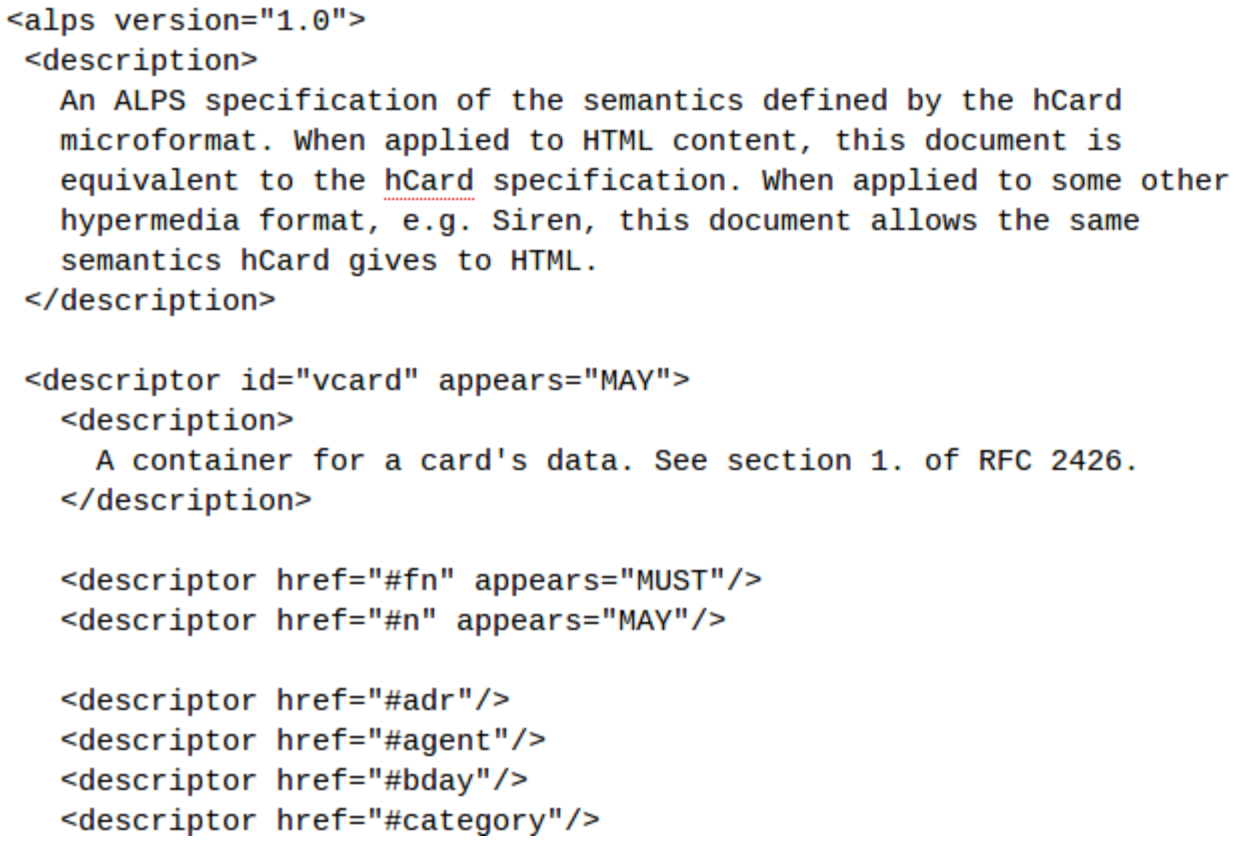
Now we have a way to share understanding about a problem domain

Not an object graph

ALPS

This could give us tools for lots of tasks...

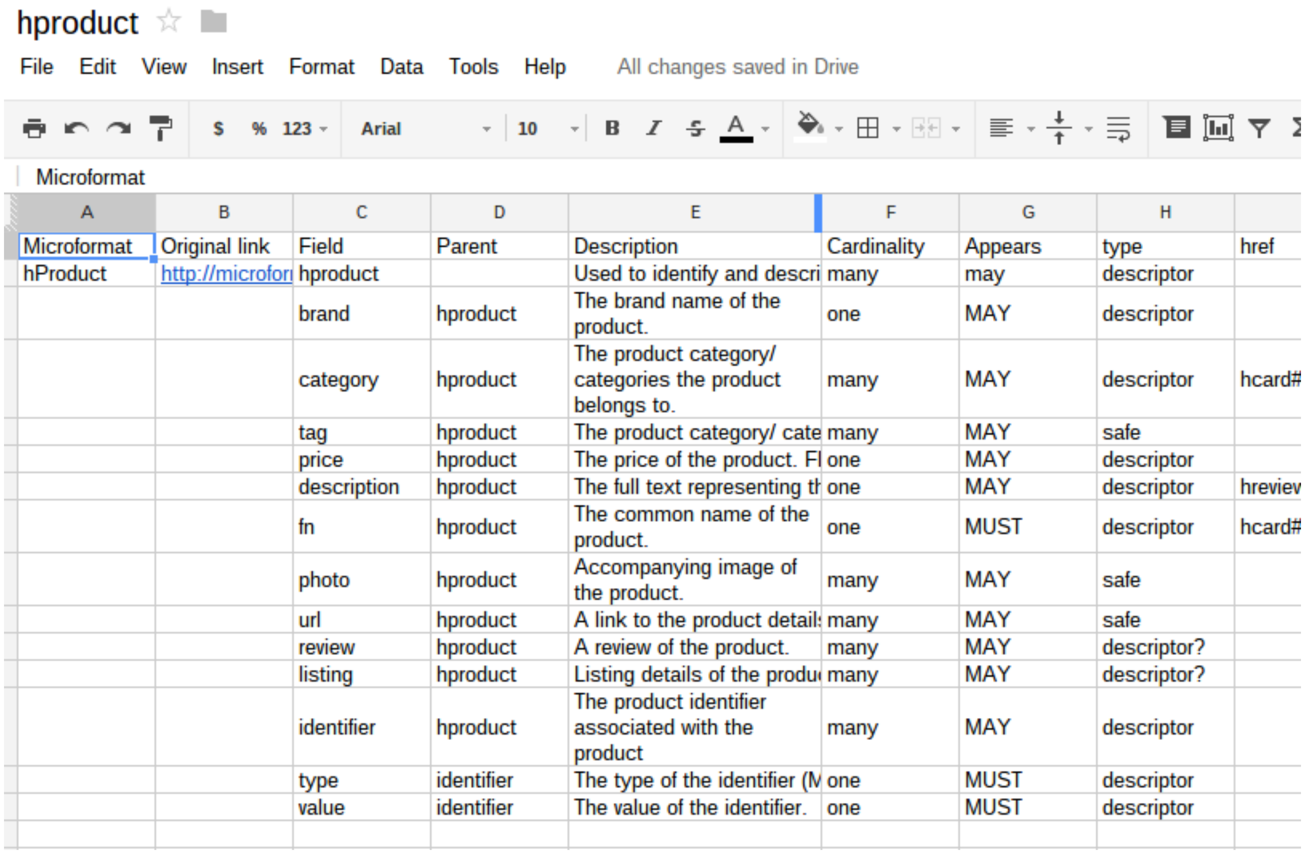
ALPS



```
<alps version="1.0">
 <description>
   An ALPS specification of the semantics defined by the hCard
   microformat. When applied to HTML content, this document is
   equivalent to the hCard specification. When applied to some other
   hypermedia format, e.g. Siren, this document allows the same
   semantics hCard gives to HTML.
 </description>

 <descriptor id="vcard" appears="MAY">
   <description>
     A container for a card's data. See section 1. of RFC 2426.
   </description>

   <descriptor href="#fn" appears="MUST"/>
   <descriptor href="#n" appears="MAY"/>

   <descriptor href="#adr"/>
   <descriptor href="#agent"/>
   <descriptor href="#bday"/>
   <descriptor href="#category"/>
```

Describing a domain profile

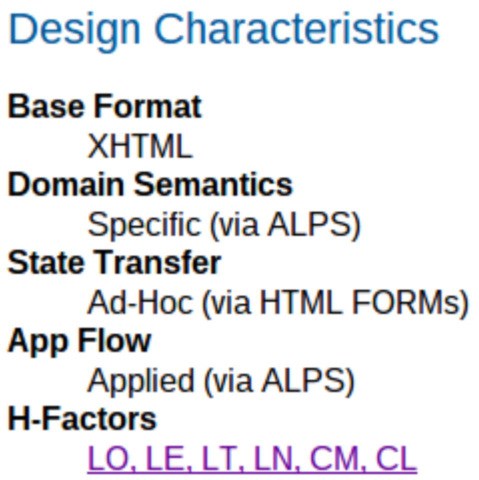
ALPS



Microformat	Original link	Field	Parent	Description	Cardinality	Appears	type	href
hProduct	http://microfor	hproduct		Used to identify and descri	many	may	descriptor	
		brand	hproduct	The brand name of the product.	one	MAY	descriptor	
		category	hproduct	The product category/ categories the product belongs to.	many	MAY	descriptor	hcard#
		tag	hproduct	The product category/ cate	many	MAY	safe	
		price	hproduct	The price of the product. Fl	one	MAY	descriptor	
		description	hproduct	The full text representing th	one	MAY	descriptor	hreviev
		fn	hproduct	The common name of the product.	one	MUST	descriptor	hcard#
		photo	hproduct	Accompanying image of the product.	many	MAY	safe	
		url	hproduct	A link to the product detail:	many	MAY	safe	
		review	hproduct	A review of the product.	many	MAY	descriptor?	
		listing	hproduct	Listing details of the produ	many	MAY	descriptor?	
		identifier	hproduct	The product identifier associated with the product	many	MAY	descriptor	
		type	identifier	The type of the identifier (M	one	MUST	descriptor	
		value	identifier	The value of the identifier.	one	MUST	descriptor	

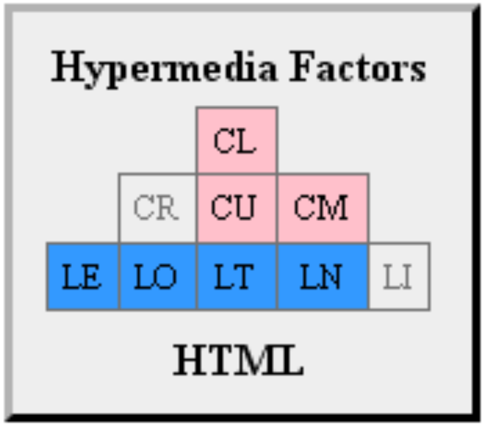
Analyzing domain profiles

ALPS



Design Characteristics

Base Format
XHTML
Domain Semantics
Specific (via ALPS)
State Transfer
Ad-Hoc (via HTML FORMs)
App Flow
Applied (via ALPS)
H-Factors
LO, LE, LT, LN, CM, CL



Matching domain profiles to media types

ALPS

contacts

gift lists

school pals

schema.org/Person

photos

matchmaker

ancestry tree

Sharing selected state with multiple services

ALPS

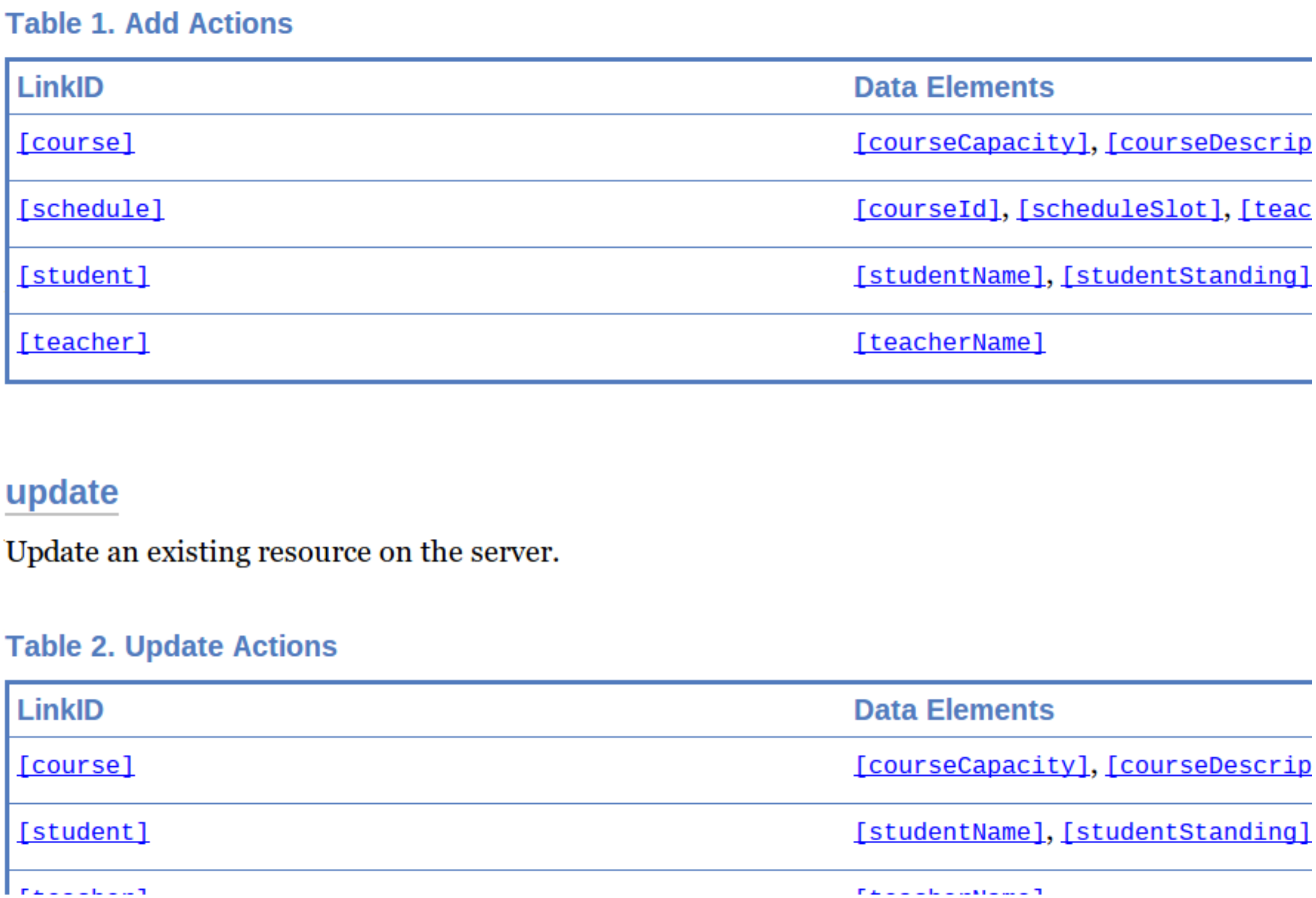


Table 1. Add Actions

LinkID	Data Elements
[course]	[courseCapacity], [courseDescrip
[schedule]	[courseId], [scheduleSlot], [teac
[student]	[studentName], [studentStanding]
[teacher]	[teacherName]

update

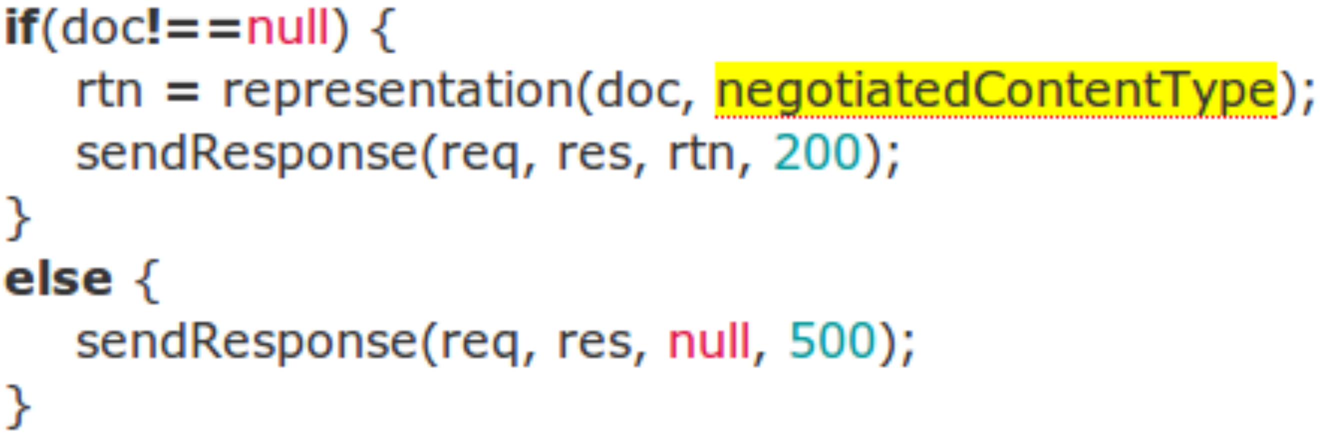
Update an existing resource on the server.

Table 2. Update Actions

LinkID	Data Elements
[course]	[courseCapacity], [courseDescrip
[student]	[studentName], [studentStanding]
[teacher]	[teacherName]

Documenting domain profiles

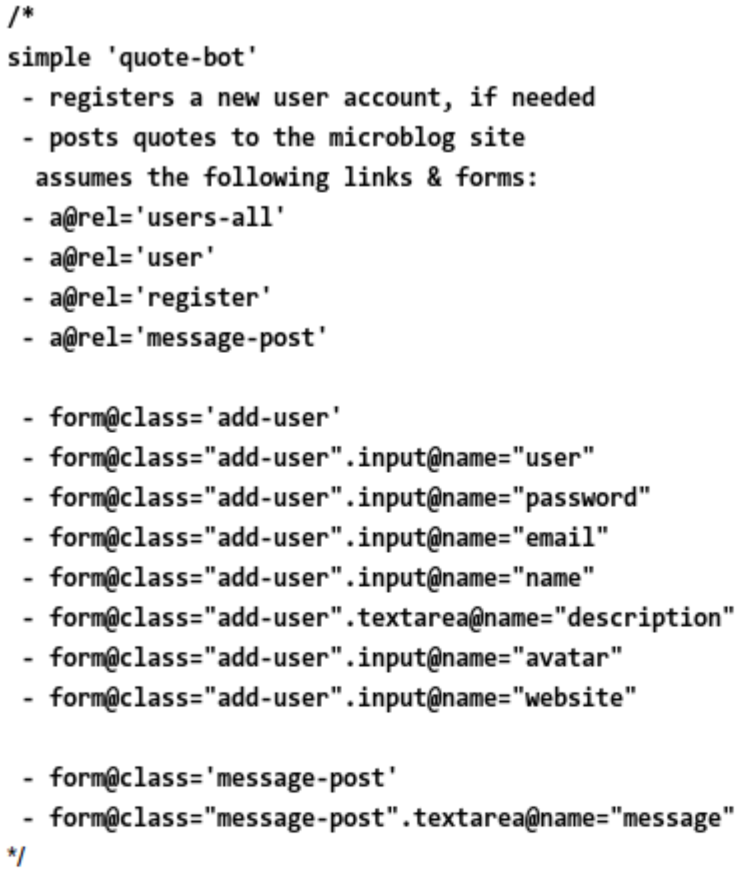
ALPS



```
if(doc!==null) {
    rtn = representation(doc, negotiatedContentType);
    sendResponse(req, res, rtn, 200);
}
else {
    sendResponse(req, res, null, 500);
}
```

Driving server-side representation engines

ALPS



```
/*
simple 'quote-bot'
 - registers a new user account, if needed
 - posts quotes to the microblog site
  assumes the following links & forms:
 - a@rel='users-all'
 - a@rel='user'
 - a@rel='register'
 - a@rel='message-post'

 - form@class='add-user'
 - form@class="add-user".input@name="user"
 - form@class="add-user".input@name="password"
 - form@class="add-user".input@name="email"
 - form@class="add-user".input@name="name"
 - form@class="add-user".textarea@name="description"
 - form@class="add-user".input@name="avatar"
 - form@class="add-user".input@name="website"

 - form@class='message-post'
 - form@class="message-post".textarea@name="message"
*/
```

Driving client-side processing engines

ALPS

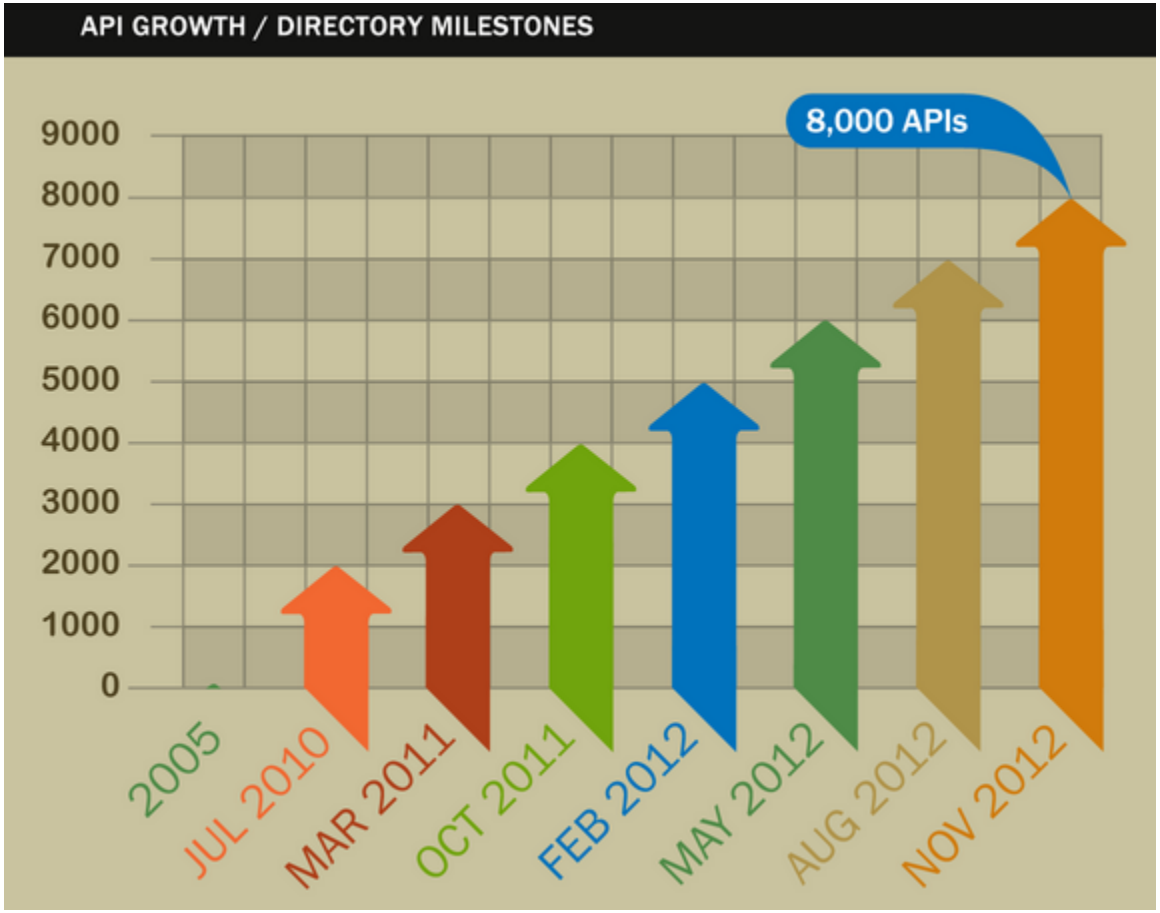
Possible benefits of ALPS Repositories:

- Share your domain profiles
- Search for profiles by topic
- Match abstracts w/ favored vocabularies
- Reflect aggregate profile use/reference data

Summary

So, where are we headed now?

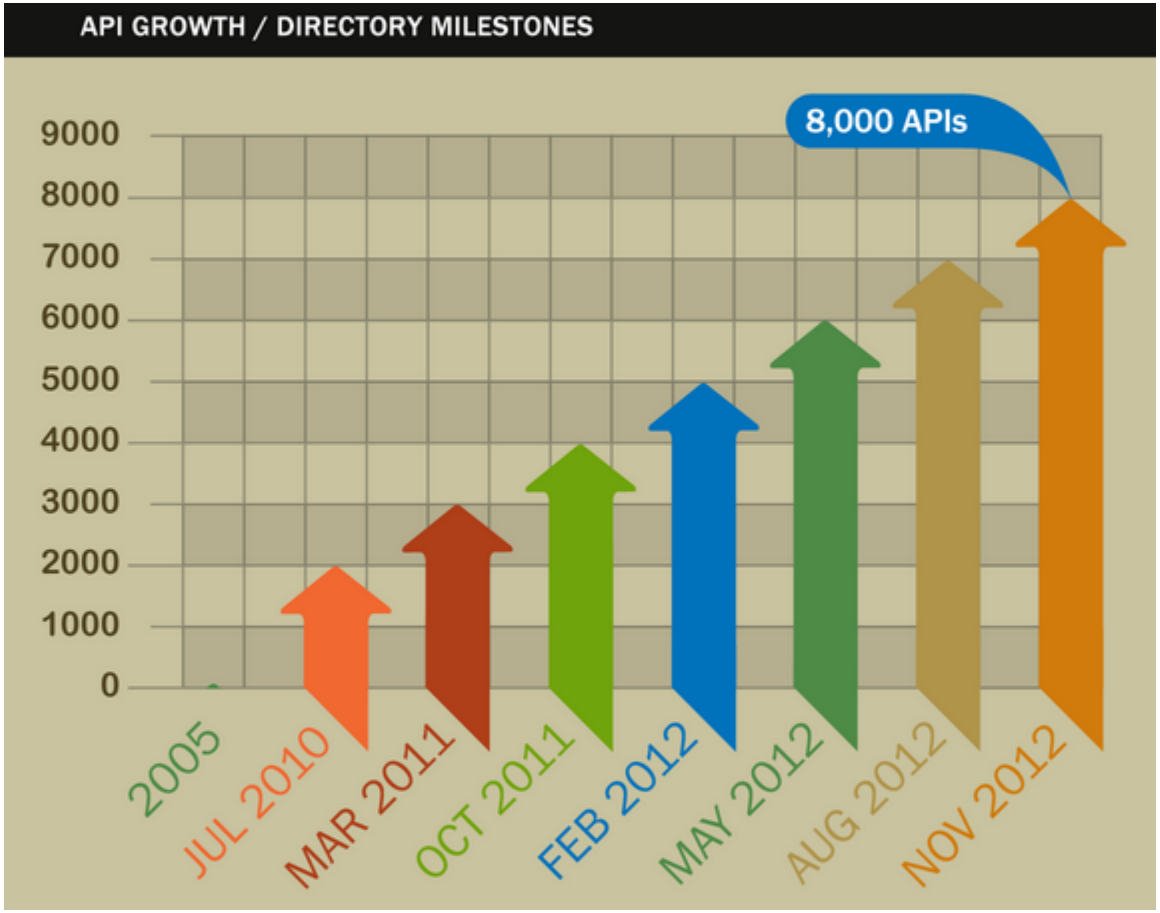
Summary



Snowflakes still common

programmableWeb, 2012

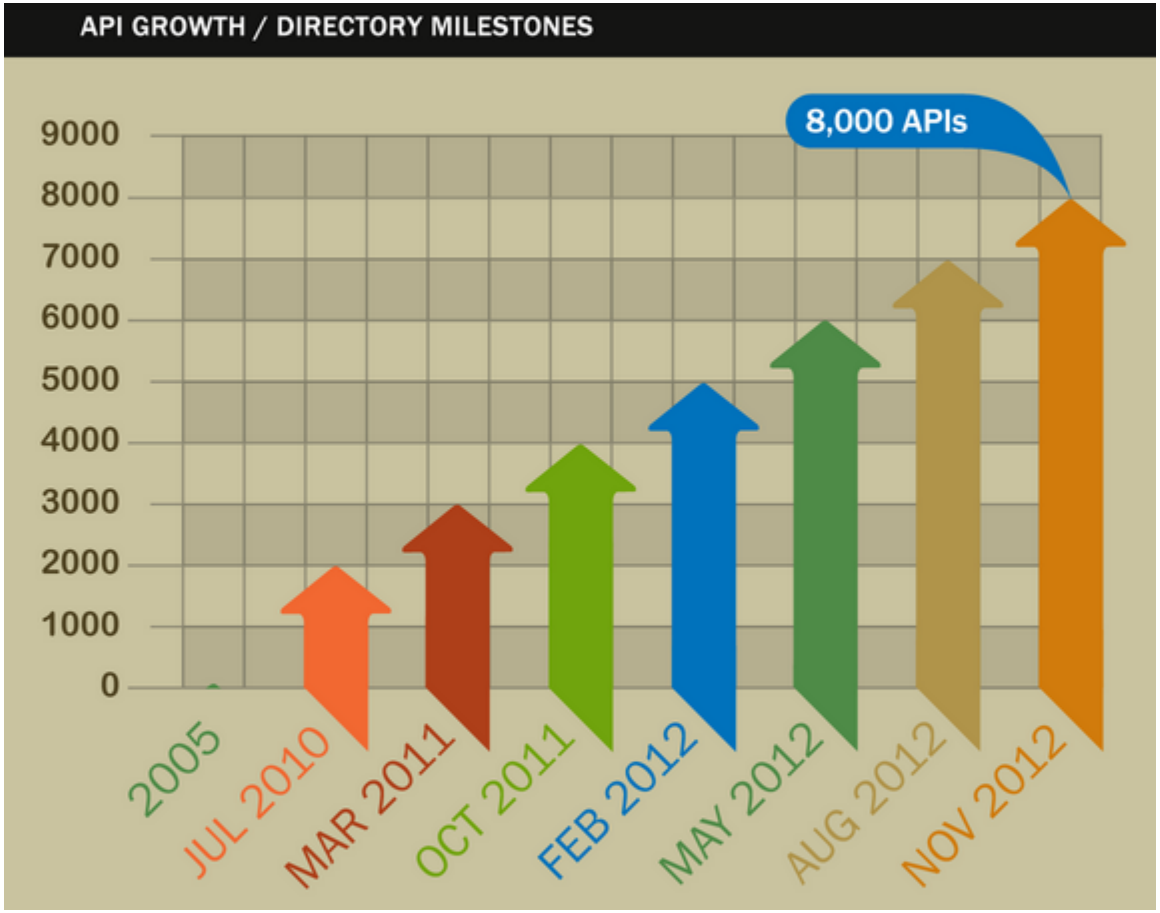
Summary



How many data types here?

programmableWeb, 2012

Summary



How much shared understanding here?

programmableWeb, 2012

Fielding & Taylor, 2002

"REST components communicate by transferring a representation of the data in a format matching one of ***an evolving set of standard data types****..."*

Summary

standard data types
includes both
state and transition types

Summary

shared understanding
includes both
state and transition types

Summary

because the Web
includes both
state and transition types

http

ftp

smtp

protocols

dns

ws

xmpp

html

csv

hal

media types

atom

json

Cj

schema.org

dublin core

VoID

vocabularies

activity streams

microformats

vCard

micro-blogging

domain profiles

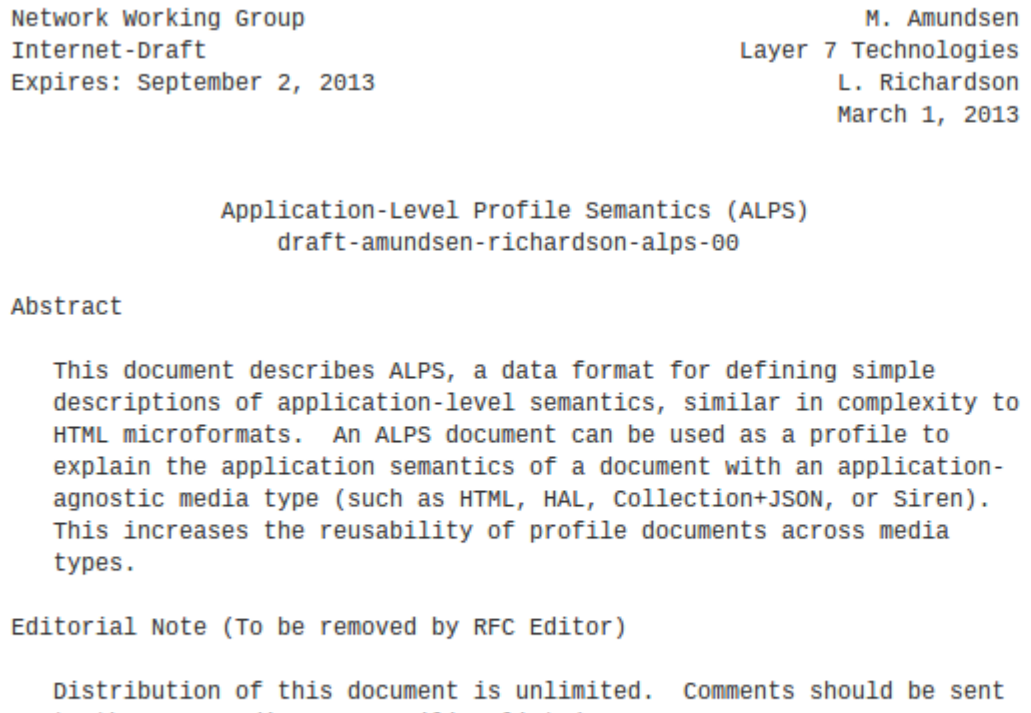
Person

hProduct

accounting

Summary

- ALPS IETF I-D will be posted in March
- Watch my blog/twitter for details
- Looking for feedback/contributions



```
Network Working Group                                       M. Amundsen
Internet-Draft                                   Layer 7 Technologies
Expires: September 2, 2013                                L. Richardson
                                                          March 1, 2013


              Application-Level Profile Semantics (ALPS)
                   draft-amundsen-richardson-alps-00

Abstract

   This document describes ALPS, a data format for defining simple
   descriptions of application-level semantics, similar in complexity to
   HTML microformats.  An ALPS document can be used as a profile to
   explain the application semantics of a document with an application-
   agnostic media type (such as HTML, HAL, Collection+JSON, or Siren).
   This increases the reusability of profile documents across media
   types.

Editorial Note (To be removed by RFC Editor)

   Distribution of this document is unlimited.  Comments should be sent
```

Let's talk!

RESTing in the ALPS

Generic Hypermedia and Domain-Specific APIs

@mamund
Mike Amundsen
Principal API Architect,
Layer 7 Technologies