

Justice Will Take Us Millions Of Intricate Moves

A talk by Leonard Richardson
<http://www.crummy.com/>



Credit: Flickr user Sakurako Kitsa

Predicates



Launchpad



Leonard D. Richardson • Log Out

This site is running pre-release code. [Please report all bugs.](#)

[Launchpad Home](#)

[Code](#)

[Bugs](#)

[Blueprints](#)

[Translations](#)

[Answers](#)



What is Launchpad?

Launchpad is a hosting service for open source projects that's big on collaboration. [Take a tour... \(new!\)](#)

Start here:

- [Open Source Projects](#)
- [Distributions](#)
- [People and Teams](#)

What's new?

- ★ **Branch uploads: fast as lightning**
If Launchpad already hosts a project's trunk, pushing up another branch is now significantly faster!
- ★ **This bug is hot!**
Help projects understand how important a bug is with the new "me too" feature.
- ★ **Help test Launchpad's new API!**
Our new internet-services API is now in beta test.

Featured projects

- Anewt
- Awn
- Bauble
- Bazaar
- GNOME Do
- Drizzle
- Extreme Tux Racer
- Exaile
- GASP
- Get-You
- LottaNZB
- Me TV
- MySQL
- PyRoom
- SchoolTool
- Silva CMS
- TangoCMS
- Terminator
- Ubuntu
- Zope.org

Tonight's Episode

Act One:

The Intergalactic Computer Network
versus the Information Superhighway

Act Two:

You Can't Design Web Services In Here,
This is The War Room!

Act Three:

The Maturity Heuristic

Conclusion:

Meet George Jetson

Act One

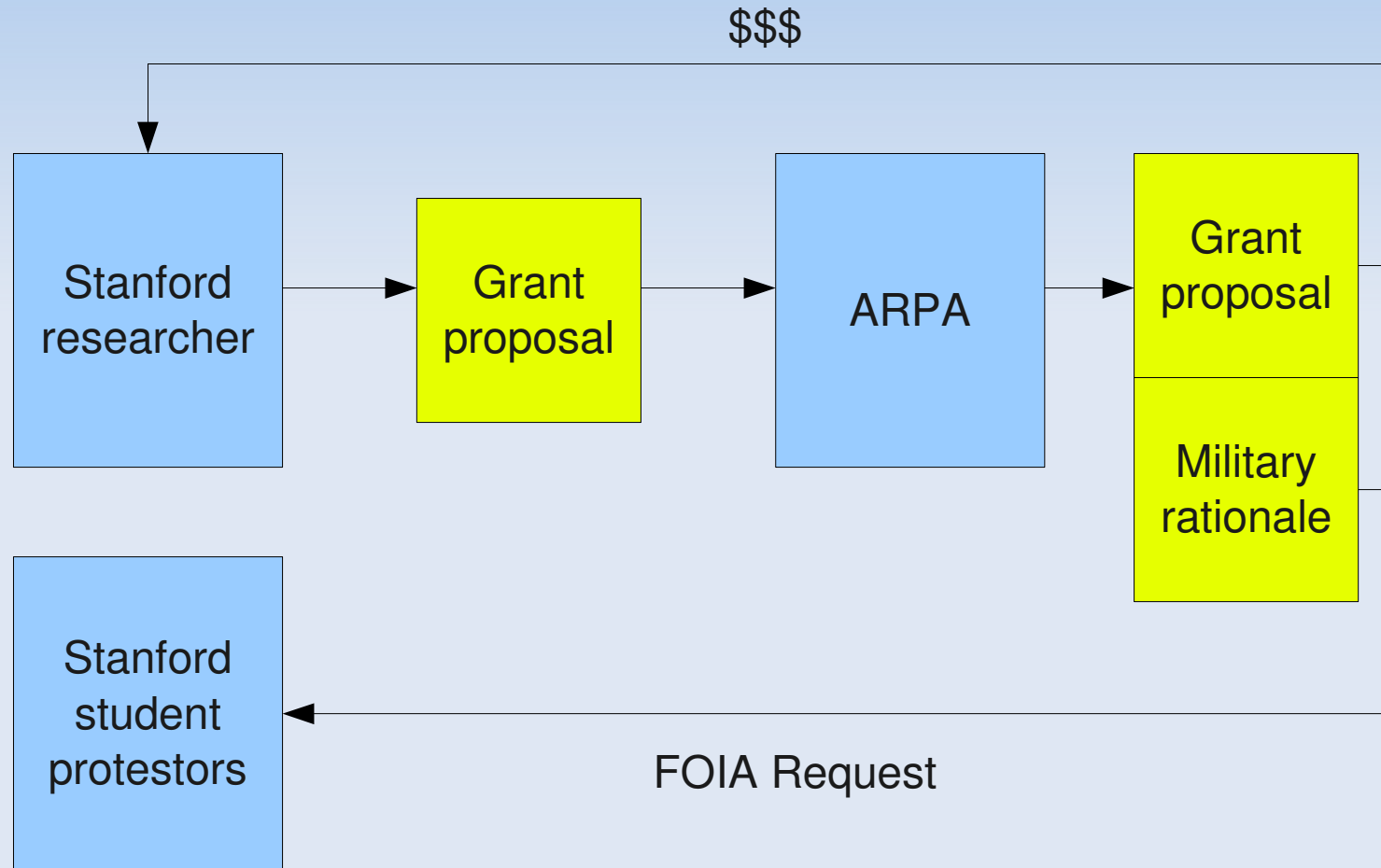
The Intergalactic Computer Network
versus the Information Superhighway



J.C.R. Licklider



CS research funding, c. 1969



(Waldrop, "The Dream Machine", p. 396-397)

The Intergalactic Computer Network



Albert Gore, Jr.



JANUARY 1994 • \$1.95 PRINTED IN U.S.A.

Popular Mechanics

1994 DESIGN & ENGINEERING AWARDS
Best Products, Designs And Ideas Of The Year

UNDERSTANDING THE INFORMATION SUPERHIGHWAY THE GREATEST SOCIAL REVOLUTION SINCE THE AUTOMOBILE

How You'll Shop, Bank, Learn, Be Entertained And More Via Interactive TV

NEW LIFE FOR DEAD BATTERIES
Revolutionary System Revives All Ordinary Alkaline Cells

BEAUTIFUL CONCRETE?
Yes! Stamped Concrete Adds High Class Without High Cash



1. Computers
2. Weather
3. Football

JANUARY 1994 • \$1.95 PRINTED IN U.S.A.

Popular Mechanics

1994 DESIGN & ENGINEERING AWARDS
Best Products, Designs And Ideas Of The Year

UNDERSTANDING THE INFORMATION SUPERHIGHWAY

THE GREATEST SOCIAL REVOLUTION SINCE THE AUTOMOBILE

How You'll Shop, Bank, Learn, Be Entertained And More Via Interactive TV



BEAUTIFUL CONCRETE?
Yes! Stamped Concrete Adds High Class Without High Cash

01438
0 754745 1

1. Durability
2. Weather
3. Flexibility

Early Internet Protocols

- FTP: file sharing
- SMTP: email
- NNTP: discussion forums
- Gopher, WWW: hypertext document serving
- IRC: real-time chat
- Telnet and rlogin: remote shells
- WAIS: database search engine
- Archie: FTP search engine
- Veronica, Jughead: Gopher search engines
- Prospero: directory services

Early Internet Protocols

- FTP: file sharing
- SMTP: email
- NNTP: discussion forums
- Gopher, **WWW**: hypertext document serving
- IRC: real-time chat
- Telnet and rlogin: remote shells
- WAIS: database search engine
- Archie: FTP search engine
- Veronica, Jughead: Gopher search engines
- Prospero: directory services

From Internet to Interweb

- SMTP: spam
- BitTorrent: copyright infringement
- IRC, AIM, Skype: real-time chat
- SSH: remote shells
- WWW: everything else

ICN Protocols

- FTP: file sharing
- SMTP: email
- NNTP: discussion forums
- Gopher, WWW: hypertext document serving
- IRC: real-time chat
- Telnet and rlogin: remote shells
- WAIS: database search engine
- Archie: FTP search engine
- Veronica, Jughead: Gopher search engines
- Prospero: directory services



Hippie stuff.



And the government entity that'll bring it to you...



Shop, Bank, Learn, Be Entertained...

- Shopping: e.g., Comp-U-Store
- Banking: e.g., BANK-AT-HOME
- Stock trading: e.g., Desk Top Broker
- Pay-to-read news archives: e.g. Nexis

Shop, Bank, Learn, Be Entertained...

- Shopping → WWW
- Banking → WWW
- Stock trading → WWW
- Pay-to-read news archives → WAIS → WWW

Heralds



Heralds



The Silver Surfer



Galactus, Eater of Worlds

Two responses, c. 2000



Roy T. Fielding

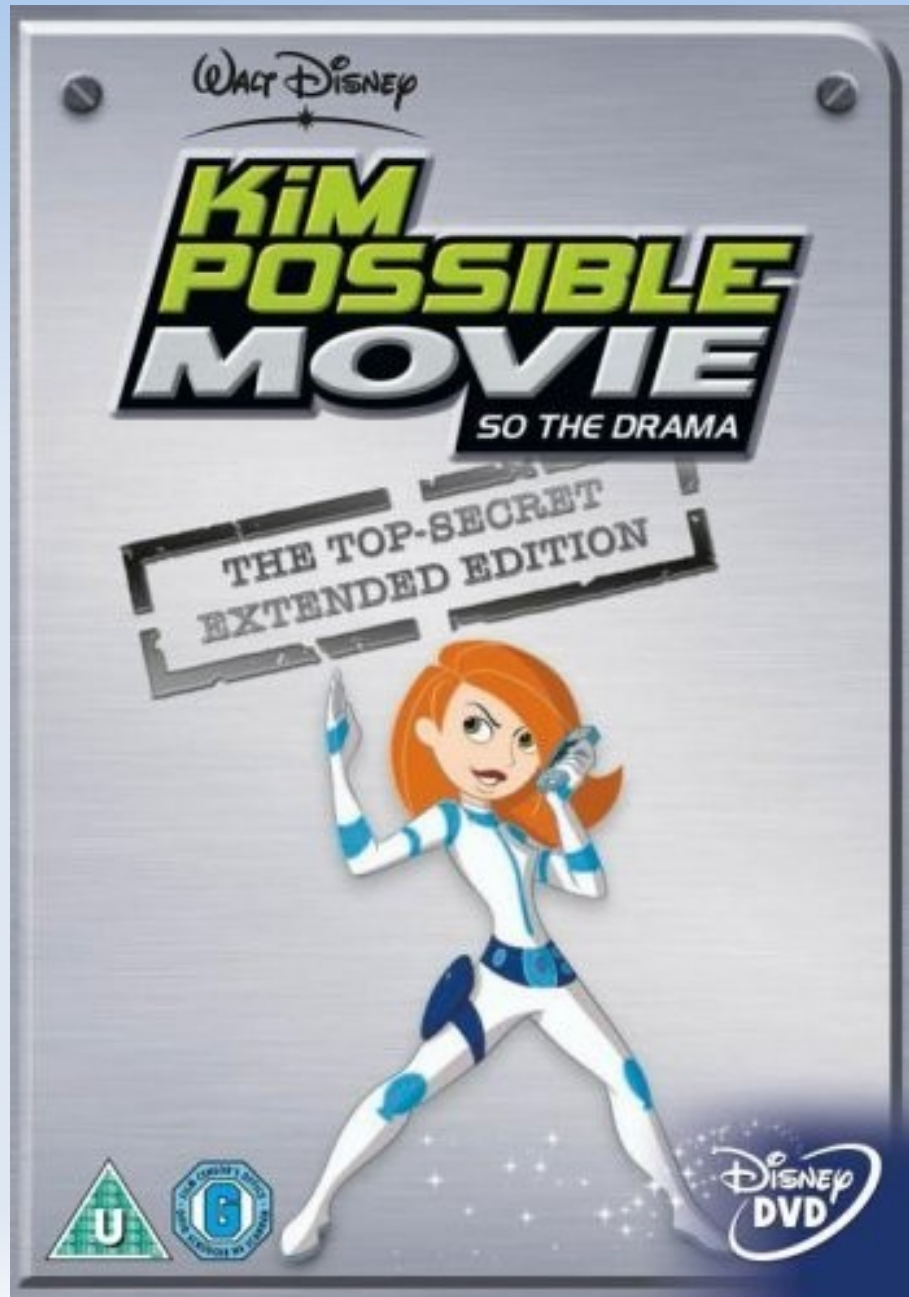


Dave Winer



Microsoft

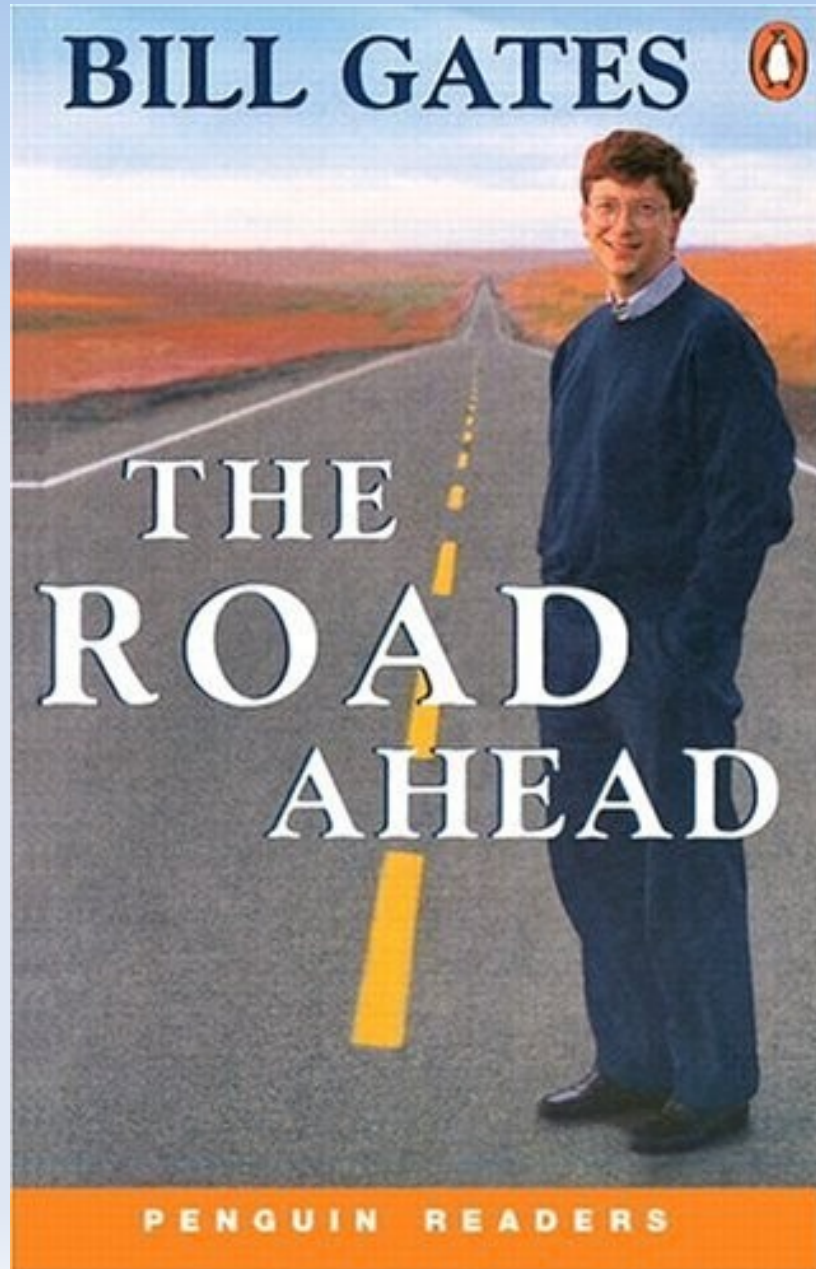
Drama



The wrong bandwagon

HTTP < WWW

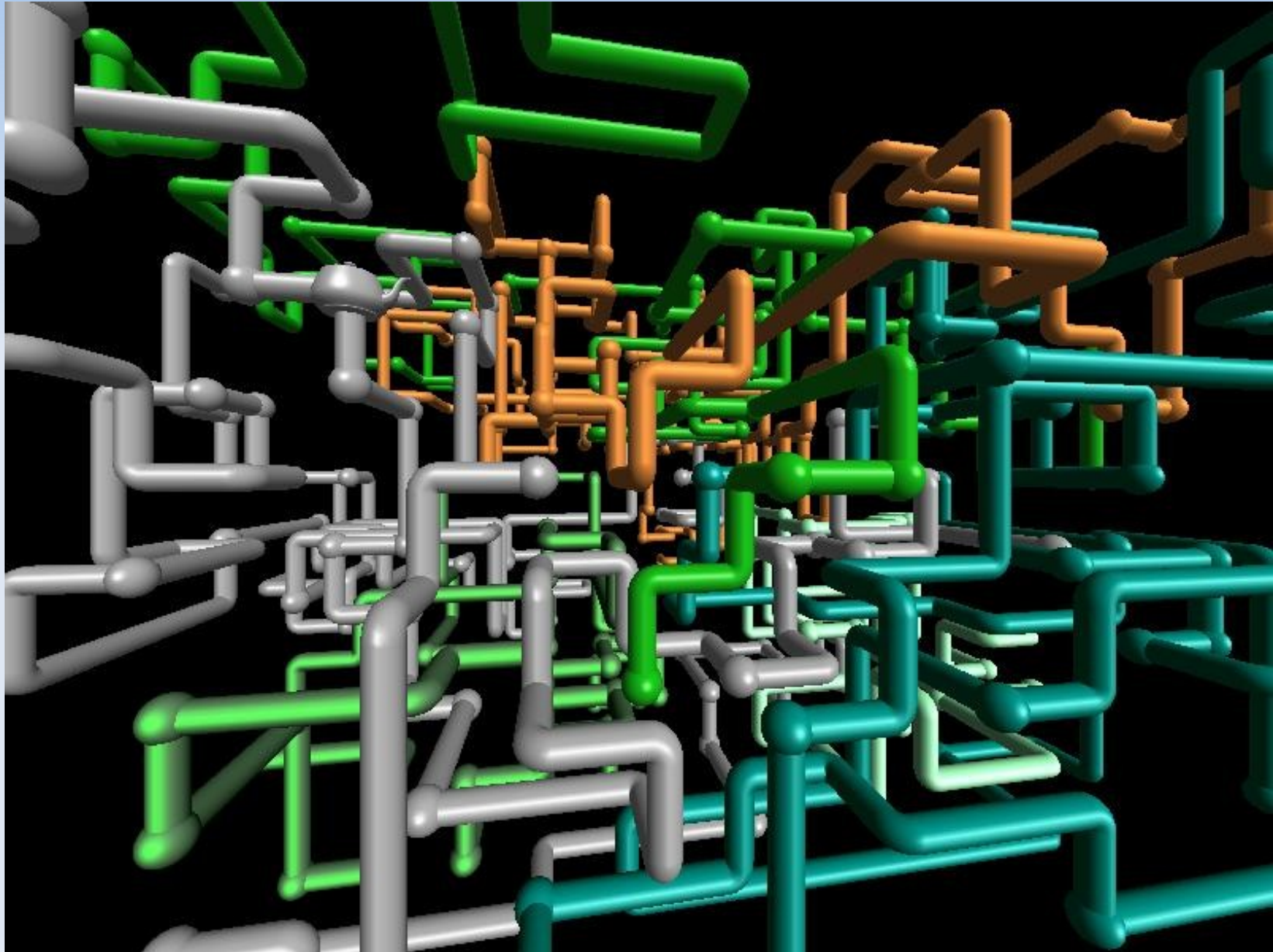
HTTP as truck



Through the firewall



Not a truck.



Use the whole web

- URI
- HTTP
- HTML

Act One conclusion

- The Web Is One Big Technology.
- It Is Made Of Three Smaller Technologies.
- And There Are Emergent Properties.

(For more retrotechnical details, see my essay,
"The System of the World Wide Web")

Act Two

You Can't Design Web Services In Here,
This Is The War Room!



Launchpad Web Service



launchpad**help**

[Home](#) | [Page Info](#) | [Changes](#) | [Attachm](#)

API

Not logged in - [Log In](#) / [Register](#)

Launchpad web services API

The Launchpad website lets you learn about and manage bugs, projects, questions, and other artifacts of software development. These same objects are gradually being exposed through a web service, so that you can access them from scripts, applications, or other websites, in addition to accessing them through the Launchpad website. The service is designed around the principles of REST, with the goals of simplicity and transparency.

The Launchpad web service is currently in a limited beta, open to Launchpad's beta testers.

- Most users will be happy with [launchpadlib](#), the official Python client library for Launchpad's web service.

```
>>> me = launchpad.me
>>> me.display_name = 'My new display name'
>>> me.lp_save()
```

- Users who aren't Python programmers, or who are interested in the inner workings of the web service, should read [the hacking document](#).

```
PATCH /beta/+me HTTP/1.1
Host: api.edge.launchpad.net
Content-type: application/json
```

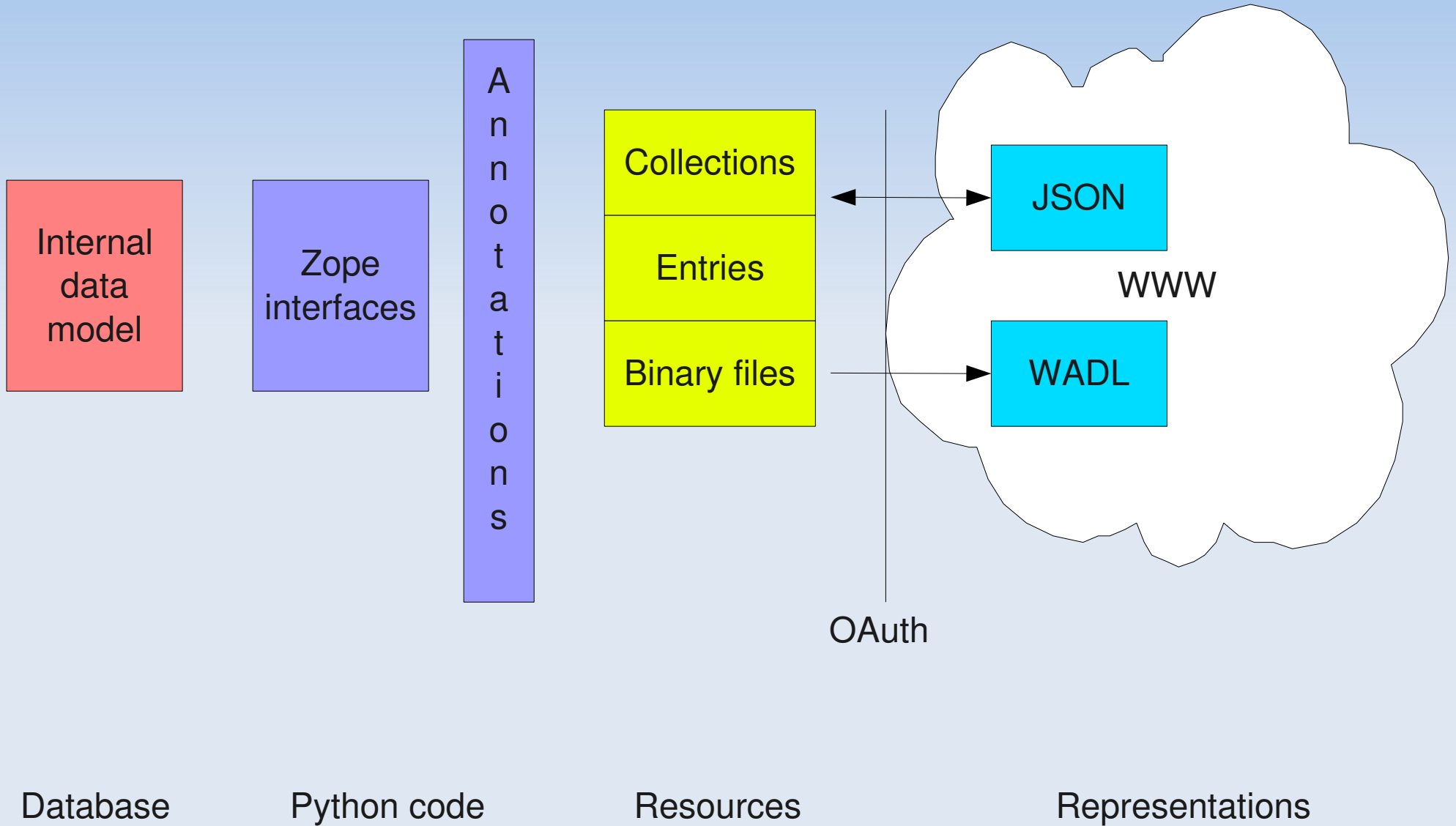
```
{ "display_name" : "My new display name" }
```

<https://help.launchpad.net/API>

The problem space

- Python and Zope
- Publish software development artifacts
 - Branches, bugs, blueprints, translations, etc.
- An average developer should be able to publish their part of Launchpad
- Also write a Python client that feels like a Python library

The design



A Zope interface

Zope
interfaces

```
class IPerson():
```

```
    id = Int(title=_('ID'), required=True, readonly=True)
```

```
    name = PersonNameField(
```

```
        title=_('Name'), required=True, readonly=False,
```

```
        constraint=name_validator,
```

```
        description=_(
```

```
            "A short unique name, beginning with a lower-case "
```

```
            "letter or number, and containing only letters, "
```

```
            "numbers, dots, hyphens, or plus signs."))
```

```
    displayname = StrippedTextLine(
```

```
        title=_('Display Name'), required=True, readonly=False,
```

```
        description=_(
```

```
            "Your name as you would like it displayed throughout "
```

```
            "Launchpad. Most people use their full name here.")),
```

```
    password = PasswordField(
```

```
        title=_('Password'), required=True, readonly=False)
```

```
    ...
```

A decorated Zope interface

```
class IPerson():
    id = Int(title=_('ID'), required=True, readonly=True)
    name = exported(PersonNameField(...))
    displayname = exported(
        StrippedTextLine(...)),
        exported_as='display_name')
    password = PasswordField(...)
    ...
```

Zope
interfaces

A
n
n
o
t
a
t
i
o
n
s

Resources

- Collections: GET, POST
- Entries: GET, PUT, PATCH, POST, [DELETE]
- Binary files: GET, PUT, DELETE

Collections

Entries

Binary files

A JSON representation

```
{u'bugs_collection_link':u'https://api.launchpad.net/beta/bugs',  
u'distributions_collection_link': u'https://api.launchpad.net/beta/distros',  
u'me_link': u'https://api.launchpad.net/beta/people/+me',  
u'people_collection_link': u'https://api.launchpad.net/beta/people',  
u'pillars_link': u'https://api.launchpad.net/beta/pillars',  
u'project_groups_collection_link': u'https://api.launchpad.net/beta/projectgroups',  
u'projects_collection_link': u'https://api.launchpad.net/beta/projects',  
u'resource_type_link': u'https://api.launchpad.net/beta/#service-root'}
```

JSON

Another JSON representation

JSON

```
{u'date_created': u'2007-10-09T14:53:16.682990+00:00',  
u'display_name': u'Leonard D. Richardson',  
u'hide_email_addresses': False,  
u'homepage_content': None,  
u'irc_nicknames_collection_link': u'https://api.launchpad.net/beta/~leonardr/...',  
u'is_team': False,  
u'is_valid': True,  
u'latitude': 40.766404069899998,  
u'longitude': -73.940906524699997,  
u'mugshot_link': u'https://api.launchpad.net/beta/~leonardr/mugshot',  
u'name': u'leonardr',  
u'resource_type_link': u'https://api.launchpad.net/beta/#person',  
u'self_link': u'https://api.launchpad.net/beta/~leonardr',  
u'time_zone': u'America/New_York'}
```

Modify me

```
PUT /~leonardr HTTP/1.1
Host: api.launchpad.net
Content-Type: application/json
```

```
{ ...
  'latitude' : '37.787132',
  'longitude' : '-122.402802'
  ...
}
```

Part of a WADL representation

WADL

```
<wadl:method id="people-findPerson" name="GET">
  <wadl:doc xmlns="http://www.w3.org/1999/xhtml">
    <p>Return all non-merged Persons with at least one email address
    whose name, displayname or email address match &lt;text&gt;.</p>
  </wadl:doc>

  <wadl:request>
    <wadl:param style="query" name="ws.op"
      required="true" fixed="findPerson">
    </wadl:param>
    <wadl:param style="query" required="true" name="text">
      <wadl:doc>Search text</wadl:doc>
    </wadl:param>
  </wadl:request>

  <wadl:response>
    <wadl:representation
      href="https://api.edge.launchpad.net/beta/#person-page"/>
    </wadl:response>
</wadl:method>
```


HTML approximation of WADL

```
<form id="people-findPerson" method="GET">  
  <p>Return all non-merged Persons with at least one email address  
    whose name, displayname or email address match &lt;text&gt;.</p>  
  
  <input type="hidden" name="ws.op" value="findPerson" />  
  <label for="text">Search text</label>  
  <input type="text" name="text" />  
</form>
```

You'd think...

```
status = exported(  
    Choice(title=_('Status'), vocabulary=BugTaskStatus,  
           default=BugTaskStatus.NEW, readonly=True))
```

```
PUT [bug task URI] HTTP/1.1  
Content-Type: application/json
```

```
{ ...  
  'status' : 'Won't Fix'  
  ...  
}
```

But no.

```
@rename_parameters_as(new_status='status')
@operation_parameters(
    new_status=copy_field(status))
@call_with(user=REQUEST_USER)
@export_write_operation()
def transitionToStatus(new_status, user):
    """Perform a workflow transition to the new status."""
```

```
POST [bug URI] HTTP/1.1
Content-Type: application/x-www-form-urlencoded

ws.op=transitionToStatus&new_status=Won't%20Fix
```

The solution: more annotations

```
@call_with(user=REQUEST_USER)
@call_on_write_to(new_status=status)
def transitionToStatus(new_status, user):
    """Perform a workflow transition to the new status."""
```

```
PUT [bug URI] HTTP/1.1
Content-Type: application/json
```

```
{ ...
  'status' : 'Won't Fix'
  ...
}
```

New object creation?

```
POST /bugs HTTP/1.1
Content-Type: application/json

{ 'title' : 'My new bug',
  ... }
```

New object creation

```
POST /bugs HTTP/1.1
```

```
Content-Type: application/x-www-form-urlencoded
```

```
ws.op=createBug&title=My%20new%20bug...
```

Does it matter?

- `launchpadlib`
- A hypermedia-based web service client
- with some Launchpad-specific assumptions and optimizations
- `sudo aptitude install python-launchpadlib` (Intrepid Ibex)

You'd think...

```
bug = launchpad.bugs[229625]  
bug_task = bug.bug_tasks[0]  
bug_task.status = "Won't Fix"  
bug_task.lp_save()
```


But no.

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "launchpadlib/resource.py", line 467, in lp_save

self._root._browser.patch(URI(self.self_link), representation)

File "launchpadlib/_browser.py", line 263, in patch

extra_headers=headers)

File "launchpadlib/_browser.py", line 211, in _request

raise HTTPError(response, content)

launchpadlib.errors.HTTPError: **HTTP Error 400: Bad Request**

Instead...

```
bug = launchpad.bugs[229625]  
bug_task = bug.bug_tasks[0]  
bug_task.transitionToStatus(status="Won't Fix")
```

When it's fixed

```
@call_with(user=REQUEST_USER)
@call_on_write_to(new_status=status)
def transitionToStatus(new_status, user):
    """Perform a workflow transition to the new status."""
```

```
bug = launchpad.bugs[229625]
bug_task = bug.bug_tasks[0]
bug_task.status = "Won't Fix"
bug_task.lp_save()
```

Human-readable reference docs

bug_task

A bug needing fixing in a particular product or package.

URL: `https://api.launchpad.net/beta/<target.name>/+bug/<bug.id>`

Default representation (application/json)

Key	Value	Description
assignee_link	(read-only) Link to a person .	Assigned to
bug_link	(read-only) Link to a bug .	Bug
bug_target_display_name	(read-only)	The short, descriptive name of the target
bug_target_name	(read-only)	The target as presented in mail notifications

Human-readable reference docs, cont.

Standard methods

GET

Response contains the default `application/json` representation for this entry.

PATCH

Entity body should contain a representation encoded using `application/json` of the entry fields to update. Any fields of the default representation marked as writeable can be included.

PUT

Entity body should contain a representation encoded using `application/json` of the entry. All fields of the default representation should be included. Only fields marked as writeable in the default representation should be modified.

Custom POST methods

transitionToAssignee

Perform a workflow transition to the given assignee.

When the bugtask assignee is changed from None to an `IPerson` object, the `date_assigned` is set on the task. If the assignee value is set to None, `date_assigned` is also set to None.

Parameters

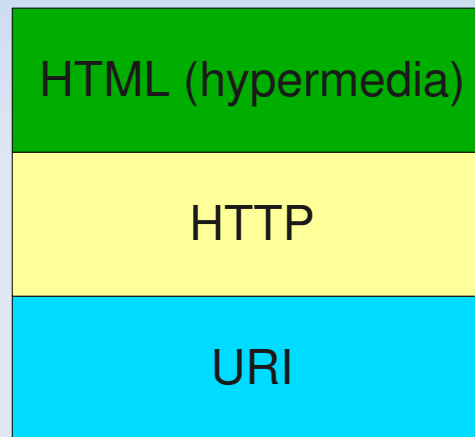
Parameter	Value	Description
ws.op	(required) Fixed: <code>transitionToAssignee</code>	
assignee	Link to a person .	Assigned to

Act Three

The Maturity Heuristic



An amazing technology stack



Level Zero



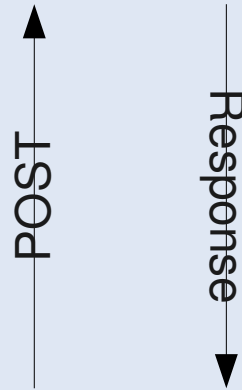
- One URI, one HTTP method
- XML-RPC and most SOAP services

Level One

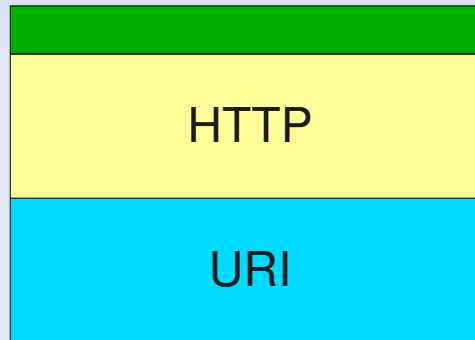


- Many URIs, one HTTP method
- Most "RESTful" services that aren't

Level Zero Service, Level One Mindset



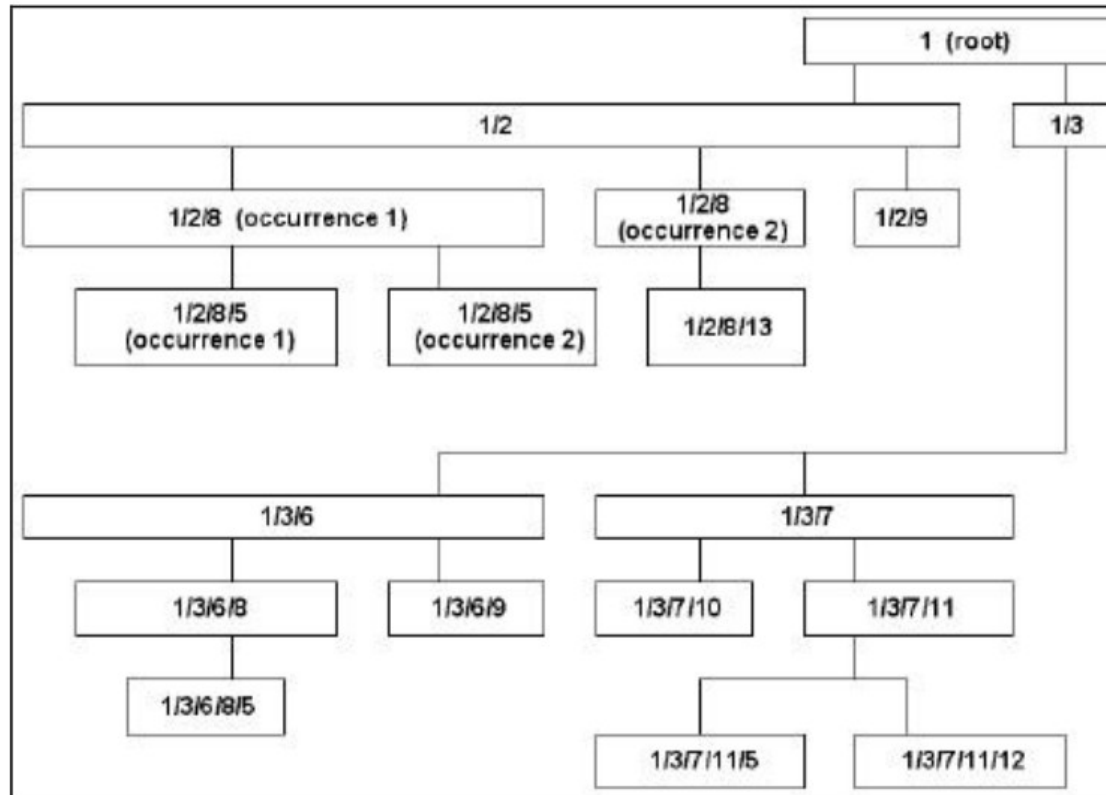
Level Two



- Many URIs, each supporting multiple HTTP methods
- The best example is Amazon S3

How we searched

ANSI/NISO Z39.50-2003



Each cell in the diagram represents an element whose tagPath is indicated within the cell. The numbers within the tagPath are tagValues; for simplicity, tagTypes are omitted, and assumed all to be the same. Leaf-nodes are highlighted by double-lined cells.

For example, the tagPath 1/3/7 represents the (non-leaf-node) element with tag 7 subordinate to the element with tag 3 subordinate to the element with tag 1. 1/3/7/11/12 represents the element whose (leaf-) node has tag 12.

RET.3.1.1.4.1 WildThing

A tagPath expression may include the wild card 'wildThing' in lieu of a tag. WildThing takes the form of an occurrence specification. For example, the tagPath expression '1/2/wildThing

How we search now

GET /search?q=Z39.50 HTTP/1.1

Host: google.com

What "GET" means to FTP

```
ftp://foo.com/RFCs/RFC2616.txt
```

```
220-
```

```
220- Welcome to my anonymous FTP server!
```

```
Name: anonymous
```

```
331 Anonymous login ok, send your complete email address as your password
```

```
Password:
```

```
230 Anonymous access granted, restrictions apply
```

```
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
```

```
ftp> cd RFCs
```

```
250 CWD command successful
```

```
ftp> get RFC2616.txt
```

```
local: RFC2616.txt remote: RFC2616.txt
```

```
200 PORT command successful
```

```
150 Opening BINARY mode data connection for RFC2616.txt (422,279 bytes)
```

```
226 Transfer complete
```

```
422279 bytes received in 1.01 secs (412.80 kB/s)
```

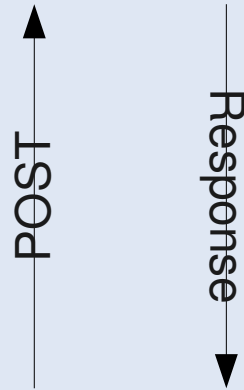
Billions of examples here

- Seriously
- But check out the Times Developer Network
- <http://developer.nytimes.com/>

Why GET matters

- GET has a specific meaning
- When meaning is constrained, you can optimize around the constraint

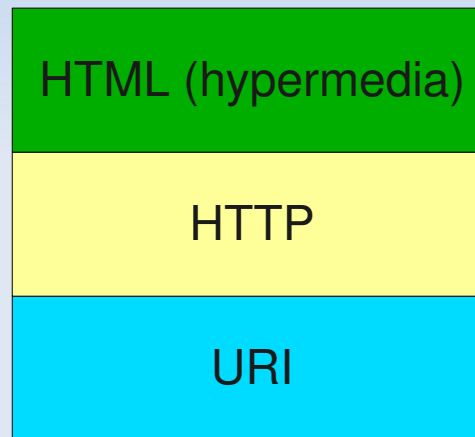
Whatever!



Should we go further?

- Split PUT, DELETE, PATCH... out from POST?
- Constraints give more specific meaning
- Which means more opportunities to optimize
- But not all clients know what those words mean

Level Three



- Resources describe their own capabilities and interconnections
- WWW, AtomPub, Netflix, Launchpad...

An Amazon S3 document

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>crummy.com</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>mydocument</Key>
    <LastModified>2006-10-27T16:01:19.000Z</LastModified>
    <ETag>"93bede57fd3818f93eedce0def329cc7"</ETag>
    <Size>22</Size>
    <Owner>
      <ID>
        c0363f7260f2f5fcf38d48039f4fb5cab21b060577817310be5170e7774aad70</ID>
      <DisplayName>leonardr28</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

An Amazon S3 document

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>crummy.com</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>mydocument</Key>
    <LastModified>2006-10-27T16:01:19.000Z</LastModified>
    <ETag>"93bede57fd3818f93eedce0def329cc7"</ETag>
    <Size>22</Size>
    <Owner>
      <ID>
        c0363f7260f2f5fcf38d48039f4fb5cab21b060577817310be5170e7774aad70</ID>
      <DisplayName>leonardr28</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

Not an Amazon S3 document

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>crummy.com</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>http://s3.amazonaws.com/crummy.com/mydocument</Key>
    <LastModified>2006-10-27T16:01:19.000Z</LastModified>
    <ETag>"93bede57fd3818f93eedce0def329cc7"</ETag>
    <Size>22</Size>
    <Owner>
      <ID>
        c0363f7260f2f5fcf38d48039f4fb5cab21b060577817310be5170e7774aad70</ID>
      <DisplayName>leonardr28</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

Part of a Netflix document

```
<link
href="http://api.netflix.com/catalog/titles/series/70023522/cast"
  rel="http://schemas.netflix.com/catalog/people" title="cast">
<cast>
  <link href="http://api.netflix.com/catalog/people/30011713"
    rel="http://schemas.netflix.com/catalog/person"
    title="Steve Carell"/>
  <link href="http://api.netflix.com/catalog/people/30014922"
    rel="http://schemas.netflix.com/catalog/person"
    title="John Krasinski"/>
  <link href="http://api.netflix.com/catalog/people/20047634"
    rel="http://schemas.netflix.com/catalog/person"
    title="Jenna Fischer"/>
  <link href="http://api.netflix.com/catalog/people/20028105"
    rel="http://schemas.netflix.com/catalog/person"
    title="Rainn Wilson"/>
  ...
</cast>
</link>
```

Hypermedia as the source of client flexibility

From this:

```
bug = launchpad.bugs[229625]  
bug_task = bug.bug_tasks[0]  
bug_task.transitionToStatus(status="Won't Fix")
```

To this:

```
bug = launchpad.bugs[229625]  
bug_task = bug.bug_tasks[0]  
bug_task.status = "Won't Fix"  
bug_task.lp_save()
```

Without changing the client.

Links are not the whole story

- Media types describe how to process a representation
- Embedded links describe relationships to other resources (`rel="prev"`)
- Embedded forms describe how to manipulate resources (or how to choose one from an infinite set of resources)

What's the difference between a stockbroker website and a wiki?

```
<form action="/purchases" id="new-purchase" method="POST">  
  <input id="symbol" type="text" />  
  <input id="quantity" type="text" />  
  <input type="button" value="Purchase" />  
</form>
```

```
POST /purchases HTTP/1.1  
Content-type: application/x-www-form-urlencoded  
...
```

```
<form action="/ThisWikiPage" id="edit" method="POST">  
  <textarea id="contents">  
    The page's current contents.  
  </textarea>  
  <input type="button" value="Edit">  
</form>
```

```
POST /ThisWikiPage HTTP/1.1  
Content-type: application/x-www-form-urlencoded  
...
```

The alternative to hypermedia is media

Methods

Update

- [posts/update](#) - Check to see when a user last posted an item.

Posts

- [posts/add](#) - add a new bookmark
- [posts/delete](#) - delete an existing bookmark
- [posts/get](#) - get bookmark for a single date, or fetch specific items
- [posts/dates](#) - list dates on which bookmarks were posted
- [posts/recent](#) - fetch recent bookmarks
- [posts/all](#) - fetch all bookmarks by date or index range
- [posts/all?hashes](#) - fetch a change detection manifest of all items

Tags

- [tags/get](#) - fetch all tags
- [tags/delete](#) - delete a tag from all posts
- [tags/rename](#) - rename a tag on all posts

Tag Bundles

- [tags/bundles/all](#) - fetch tag bundles
- [tags/bundles/set](#) - assign a set of tags to a bundle
- [tags/bundles/delete](#) - delete a tag bundle

Update

<https://api.del.icio.us/v1/posts/update>

Returns the last update time for the user, as well as the number of new items in the user's inbox since it was last visited.

Use this before calling `posts/all` to see if the data has changed

Link technologies

- URI

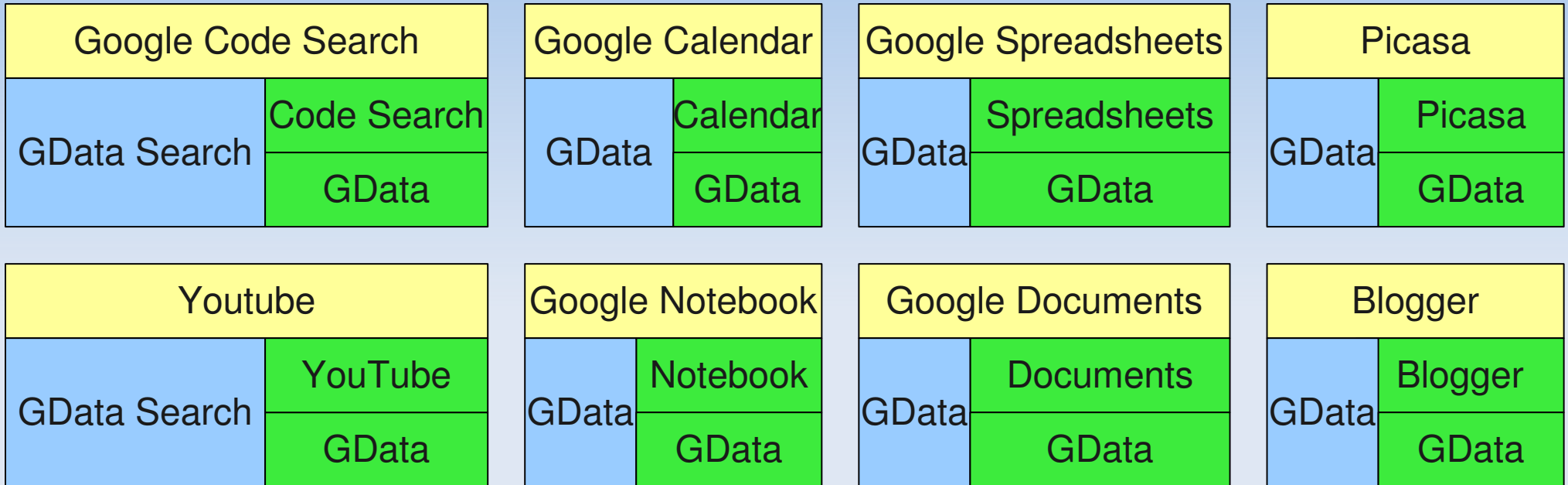
Form technologies

- HTML
- AtomPub descriptors
- WADL
- XForms
- OpenSearch
- RDF Forms
- URI Templates
- ...

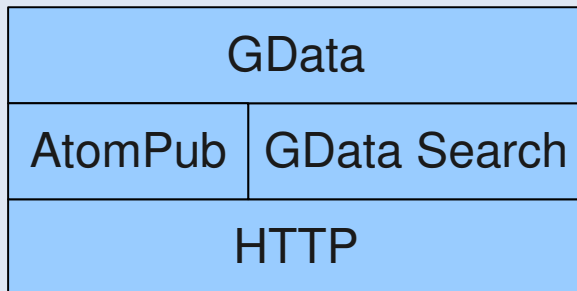
The Web technologies are extensible

- HTTP (AtomPub, ActiveResource)
- XML
- HTML (Microformats)
- URI: http://, irc://, ftp://, ...

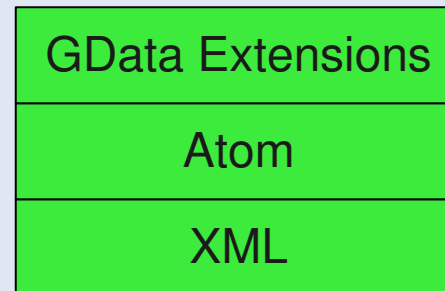
Technologies stack



Resource behavior



Data formats



Conclusion

Meet George Jetson



Robots are complex



Credit: Flickr user abux_77

Robot psychotherapy

- I AM THE PSYCHOTHERAPIST.
- PLEASE, DESCRIBE YOUR PROBLEMS.
- EACH TIME YOU ARE FINISHED TALKING,
TYPE RET TWICE

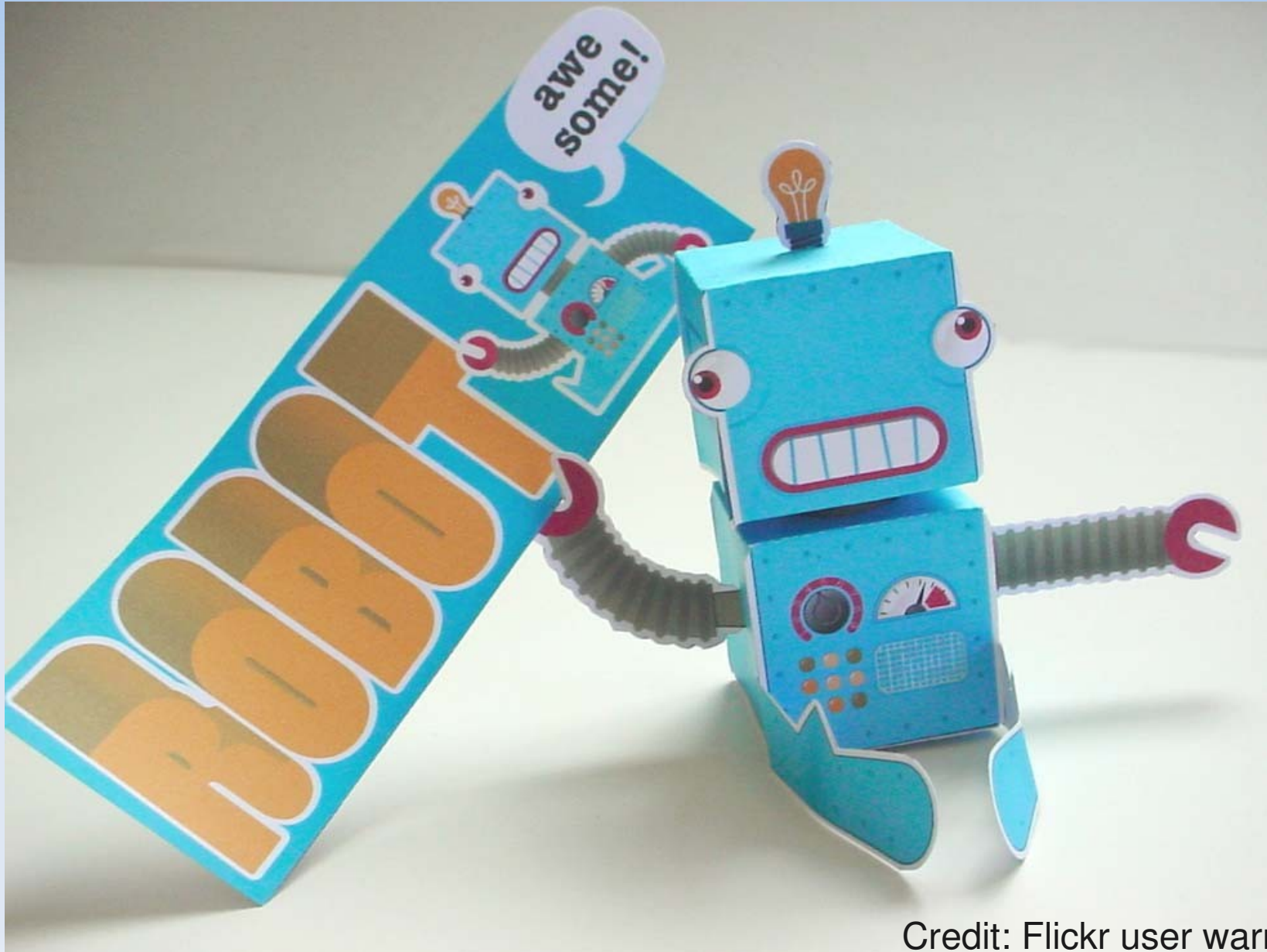
Dealing with complexity

- Divide and conquer: *spreads complexity around*
- Refactor: *reduces complexity*
- Describe special behavior in a standard way: *makes complexity learnable*

Dealing with complexity

- Divide and conquer: **URI**
- Refactor: **HTTP**
- Describe special behavior in a standard way:
Hypermedia

END



Credit: Flickr user warmnfuzzy