jax 2015

Business Technology Days

BIG DATA CON

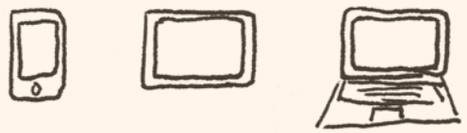
Oliver Wolf | innoQ

Hypermedia APIs für das Internet of Things

Oliver Wolf
@owolf

www.innoq.com
@innoQ

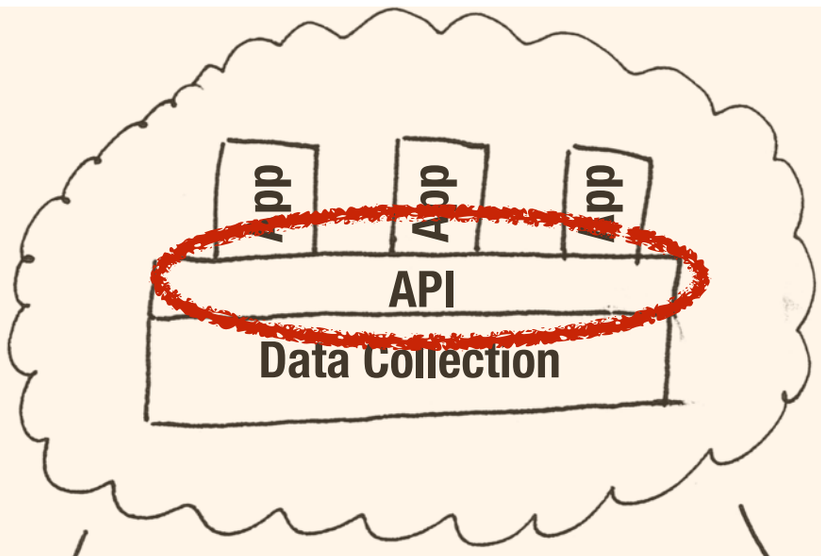
The Internet Of Things Architectural Tiers



HTTP JSON HTML

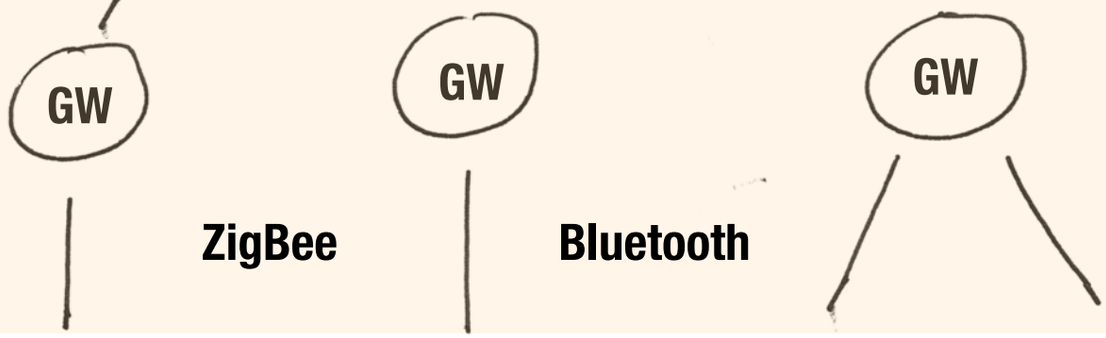
User

User Interface



Cloud

Application/Service Platform



Gateway

Personal Area Networks

Things

Sensors, Actors
Mesh Networks

Characteristics of a Good API
(according to Joshua Bloch)

- Easy to learn
- Easy to use, even without documentation
- Hard to misuse
- Easy to read and maintain code that uses it
- Sufficiently powerful to satisfy requirements
- Easy to extend
- Appropriate to the audience

REST

The “Uniform Interface” constraint revisited

Identification of resources

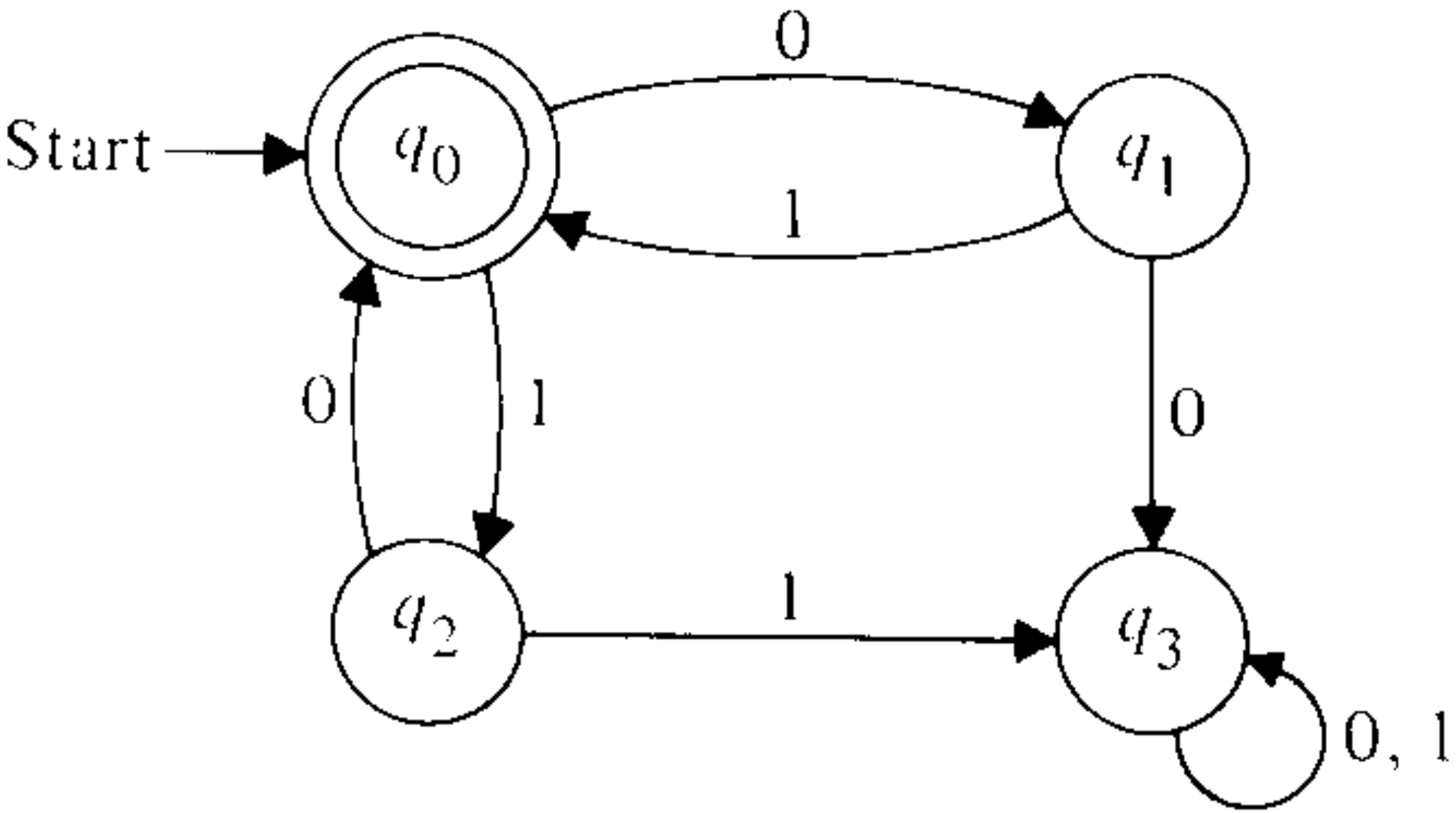
Manipulation of resources through representations

Self-descriptive messages

Hypermedia as the engine of application state

HATEOAS?

A resource is just a state machine, basically.



Home Documents for HTTP APIs (draft-nottingham-json-home-03)

Simple canonical JSON representation for HTTP API entry points (“home documents”)

```
GET / HTTP/1.1
Host: example.org
Accept: application/json-home

HTTP/1.1 200 OK
Cache-Control: max-age=3600
Content-Type: application/json-home
   {
     "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"]
         }
       }
     }
   }
```

Cacheable on client side

Special media type

Link relation type

Relative URI

Direct link (1 resource)

Web Linking
(RFC 5988)

Link Relation Types identify the semantics of a link

Two types of Relation Types:

Registered	**Identified by a registered token, currently one of:** `alternate, appendix, bookmark, chapter, contents, copyright, current, describedby, edit, edit-media, enclosure, first, glossary, help, hub, index, last, latest-version, license, next, next-archive, payment, prev, predecessor-version, previous, prev-archive, related, replies, section, self, service, start, stylesheet, subsection, successor-version, up, version-history, via, working-copy, working-copy-of`
Extension	**Uniquely identified by a URI which can (but doesn't have to) point to a document describing the link sematics**

URI Template
(RFC 6570)

URI Templates describe a range of URIs through variable expansion

```
http://example.com/~fred/
http://example.com/~mark/
```

Examples

Template `http://example.com/~{username}/`

```
http://example.com/dictionary/c/cat
http://example.com/dictionary/d/dog
```

Examples

Template `http://example.com/dictionary/{term:1}/{term}`

```
http://example.com/search?q=cat&lang=en
http://example.com/search?q=chien&lang=fr
```

Examples

Template `http://example.com/search{?q,lang}`

(The expression syntax is a little bit richer than this, actually.)

Simple canonical JSON representation for HTTP API entry points (“home documents”)

```
GET / HTTP/1.1
Host: example.org
Accept: application/json-home

HTTP/1.1 200 OK
Cache-Control: max-age=3600
Content-Type: application/json-home
   {
     "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"]
         }
       }
     }
}
```

Cacheable on client side

Special media type

Link relation type

Relative URI

Direct link (1 resource)

Link relation type

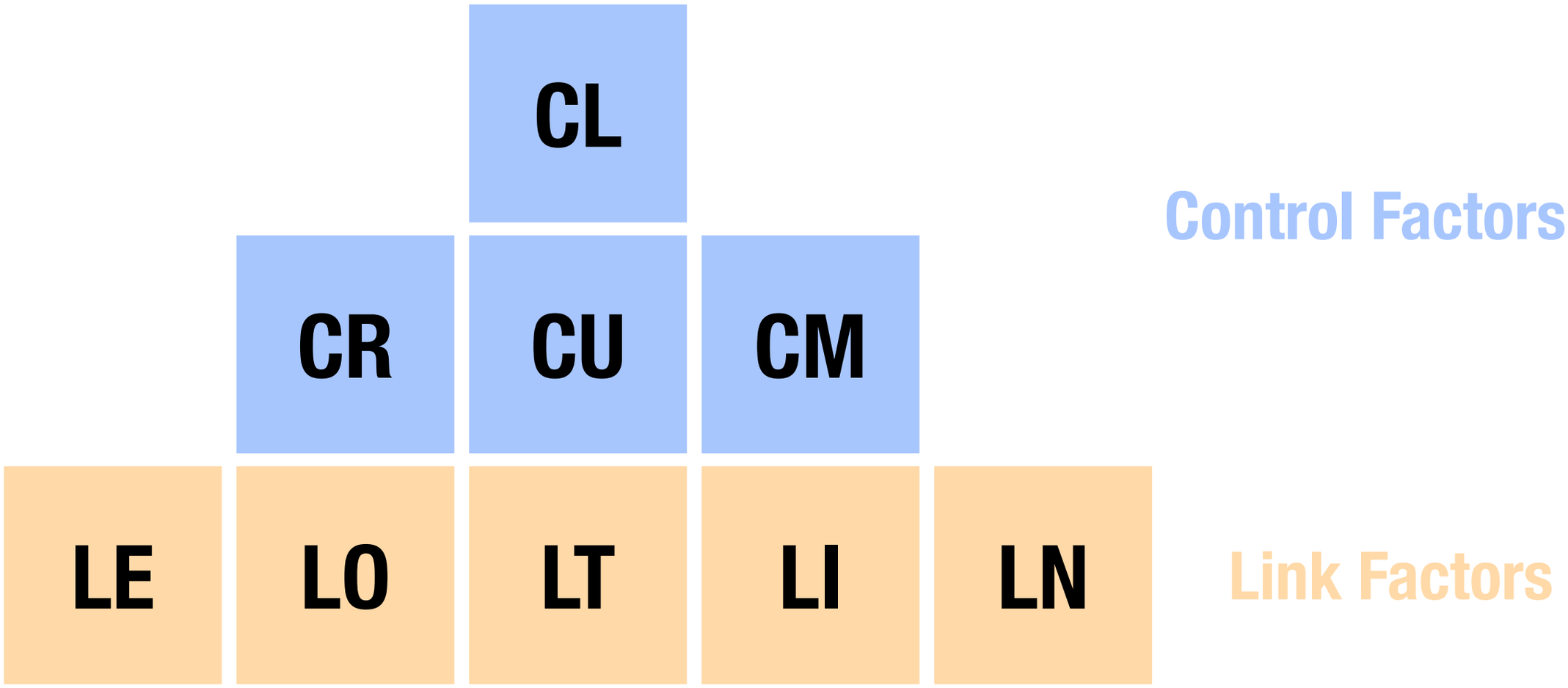
Link template

Template variables

Templated link (0..n resources)

Hints

Mike Amundsen's "H-Factors"



Source: M. Amundsen: Hypermedia APIs with HTML5 & Node, O'Reilly, 2011

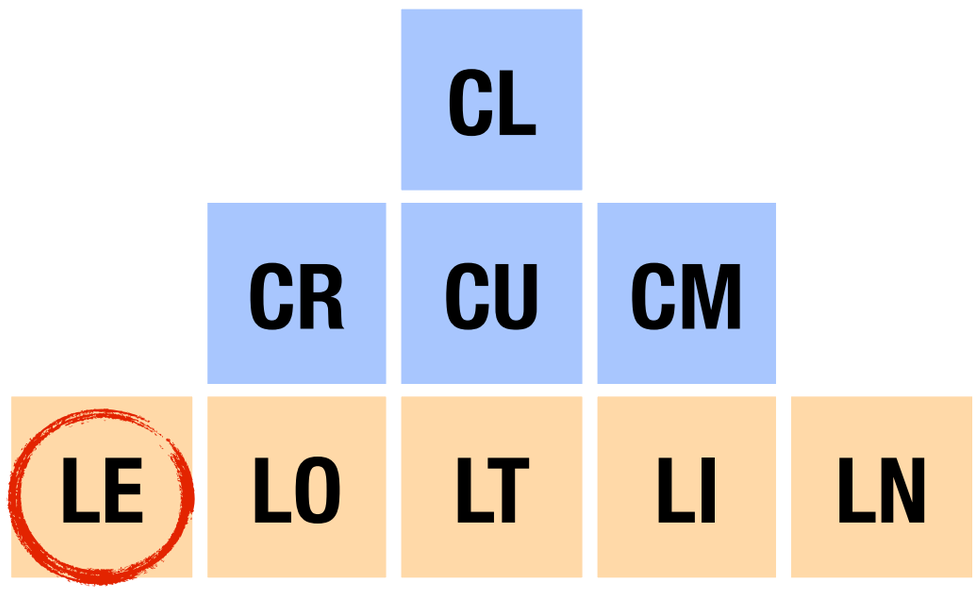
Mike Amundsen's "H-Factors"

Embedding Links

- dereference URI using HTTP GET and merge result with original document content

Example: IMG markup tag in HTML

```
<img src="..."/>
```



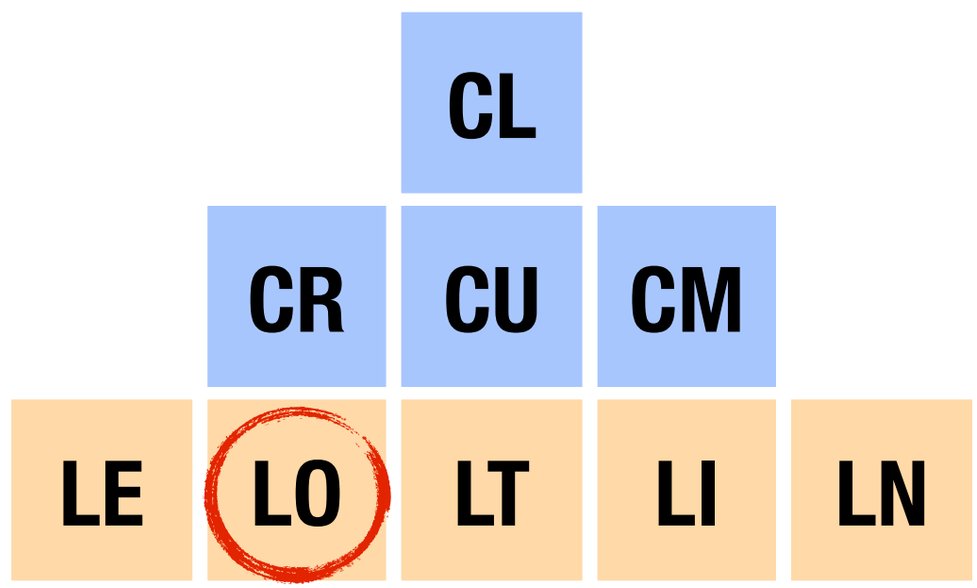
Mike Amundsen's "H-Factors"

Outbound Links

- dereference URI using HTTP GET and replace original document content with result (traversal/navigational link)

Example: A markup tag in HTML

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



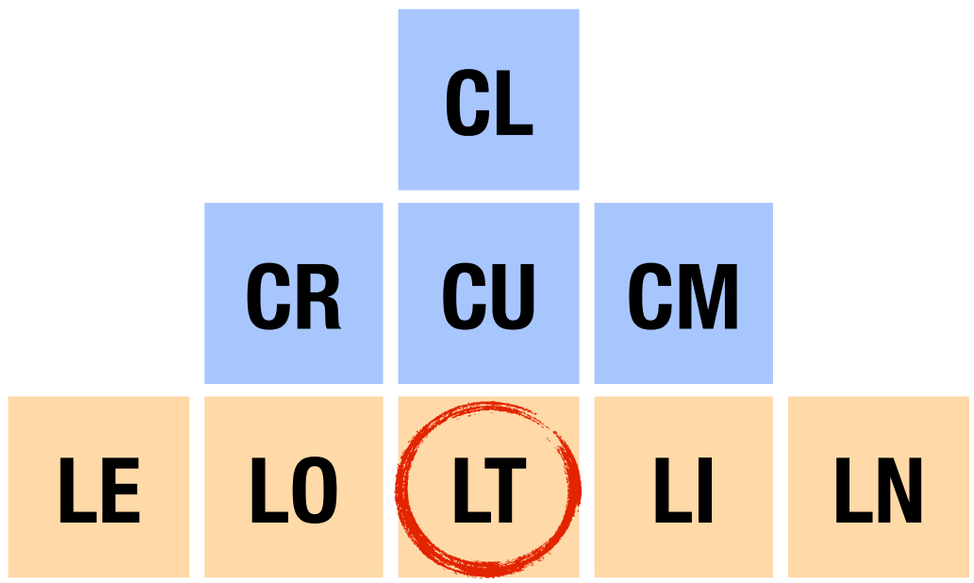
Mike Amundsen's "H-Factors"

Templated Links

- provide a way to indicate parameters that can be supplied when executing an HTTP GET

Example: URI Templates

```
<link href="http://example.org?search={search}">
```



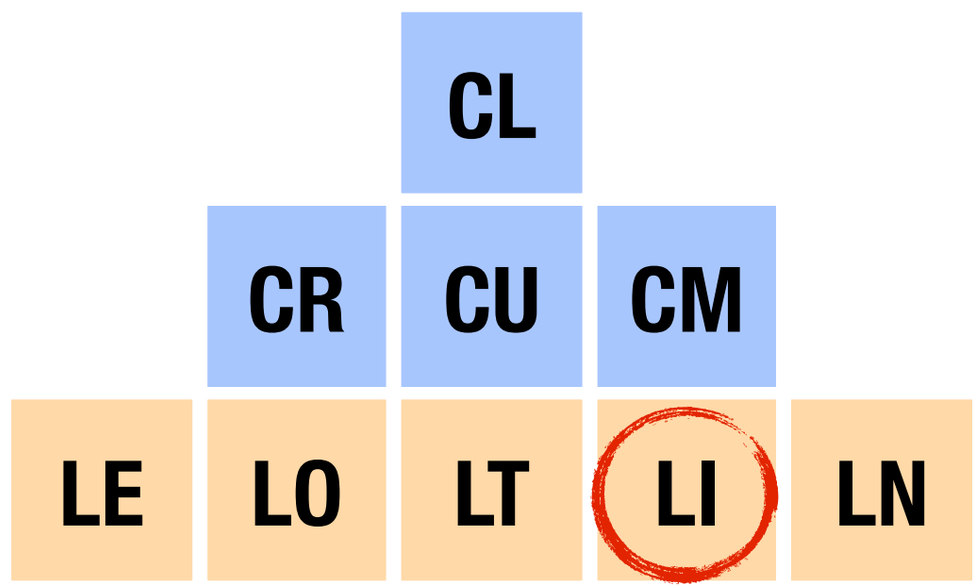
Mike Amundsen's "H-Factors"

Idempotent Links

- provides a way to define support for idempotent operations on server via HTTP PUT and DELETE methods

Example: Link with link relation type "edit" in the Atom media type

```
<link rel="edit" href="http://example.org/edit/1"/>
```



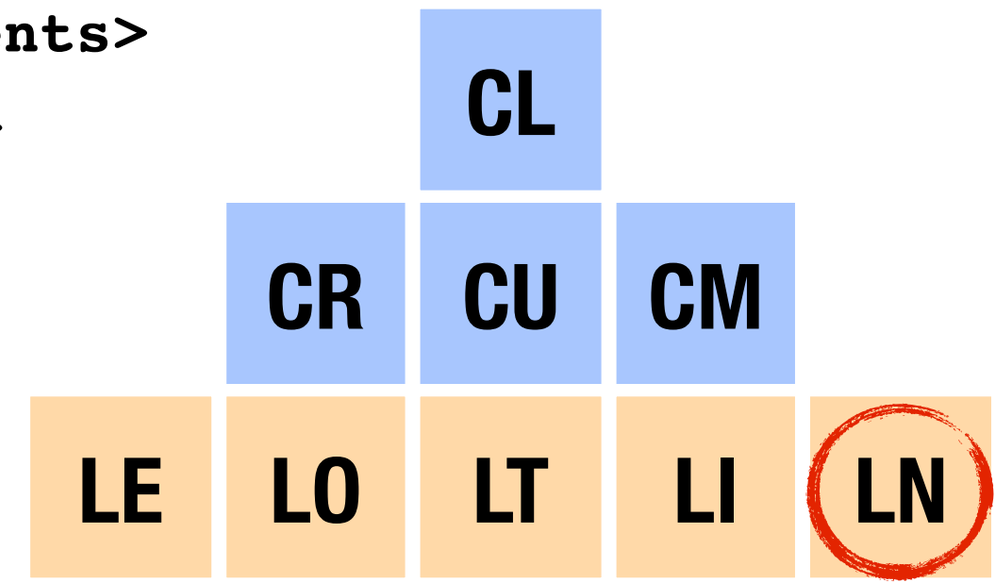
Mike Amundsen's "H-Factors"

Non-Idempotent Links

- provides a way to define support for non-idempotent operations on server via the HTTP POST method

Example: HTTP FORM element

```
<form method="post"
      action="http://example.org/comments>
  <textarea name="comment"></textarea>
  <input type="submit"/>
</form>
```



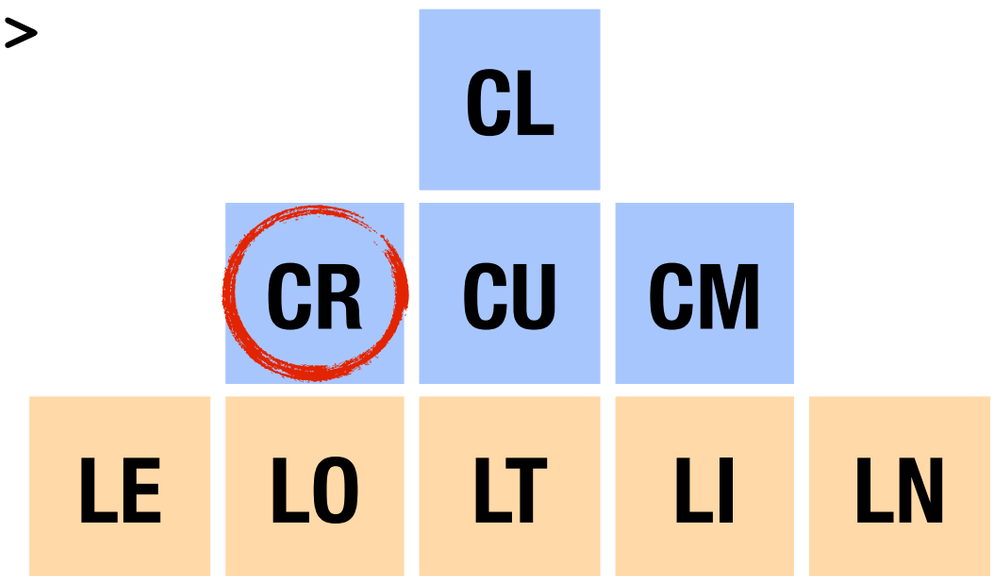
Mike Amundsen's "H-Factors"

Read Controls

- provides a way to support manipulation of control data for HTTP GET requests

Example: XInclude markup w/ accept-language attribute

```
<x:include href="http://example.org/newsfeed"
    accept-language="de, en-gb;q=0.8" />
```



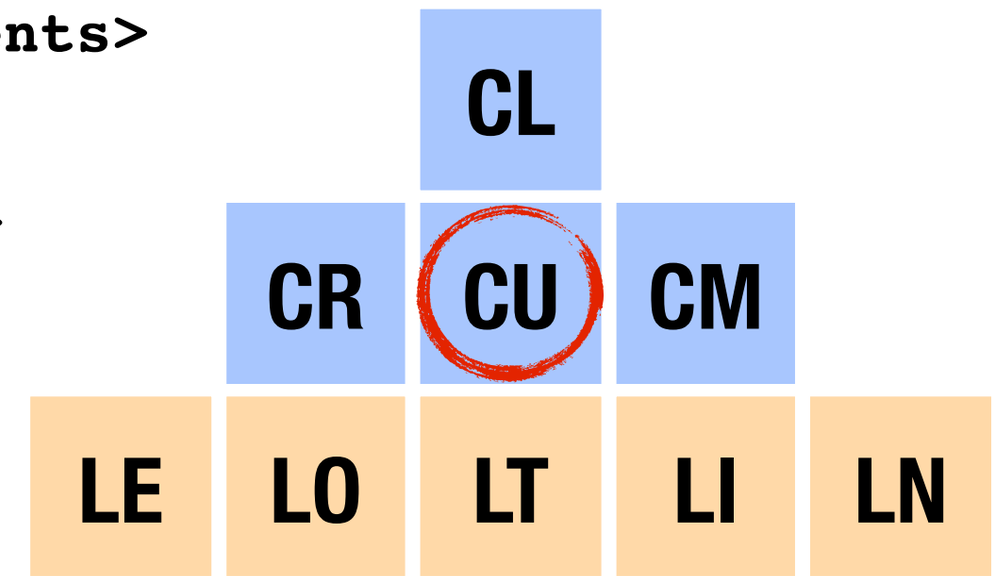
Mike Amundsen's "H-Factors"

Update Controls

- provides a way to support manipulation of control data for HTTP PUT/POST requests

Example: HTML FORM w/ enctype attribute

```
<form method="post"
      action="http://example.org/comments>
      enctype="text/plain"
  <textarea name="comment"></textarea>
  <input type="submit"/>
</form>
```



Mike Amundsen's "H-Factors"

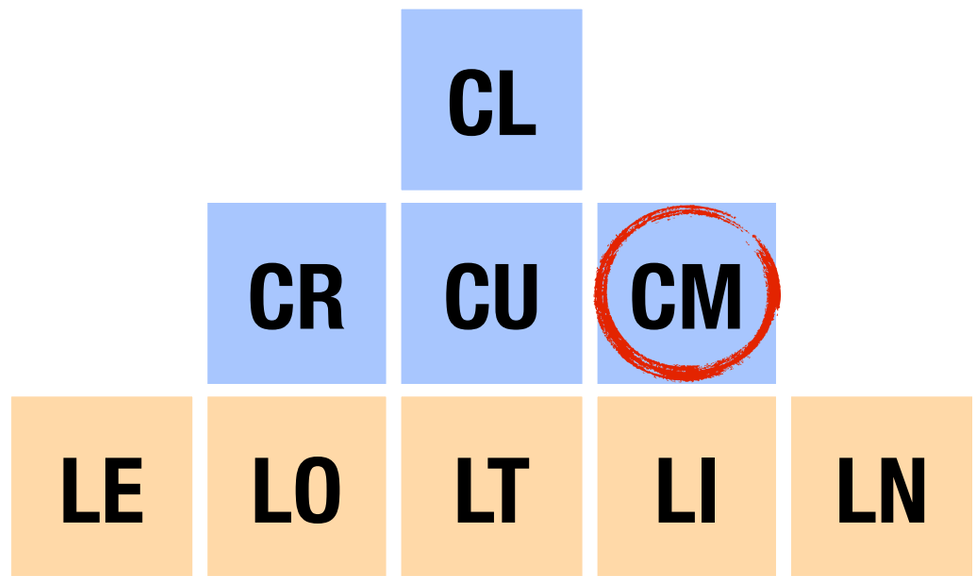
Method Controls

- support the ability to change the control data for the protocol method used

Example: HTML FORM's "method" element

```
<form method="post">
...
</form>

<form method="get">
...
</form>
```



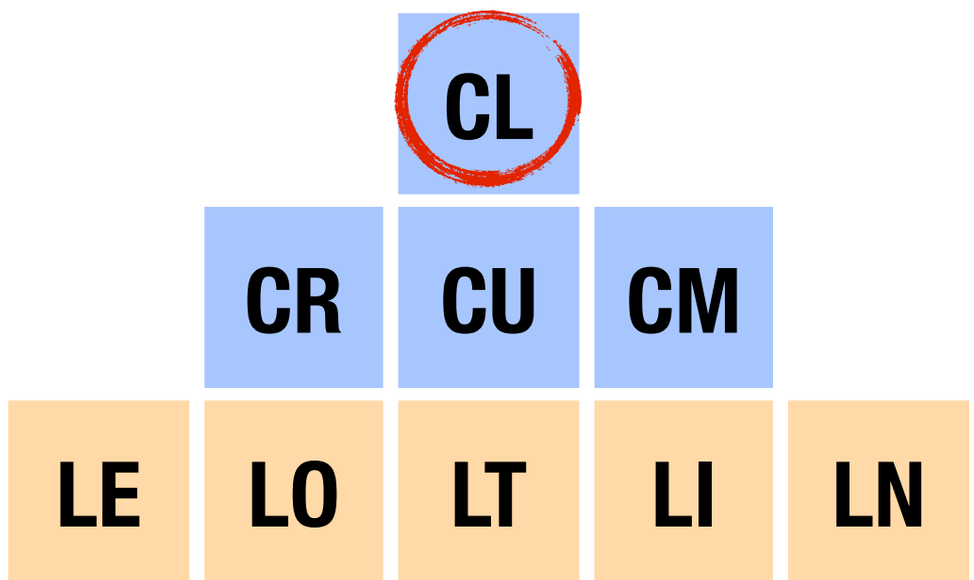
Mike Amundsen's "H-Factors"

Link Annotation Controls

- decorate links with additional metadata communicating a link's semantics to a client

Example: well-known link relation type "stylesheet" in HTML

```
<link rel="stylesheet" href="..."/>
```



Hypermedia formats for resource representations

HTML

Atom/AtomPub

Collection+JSON

HAL

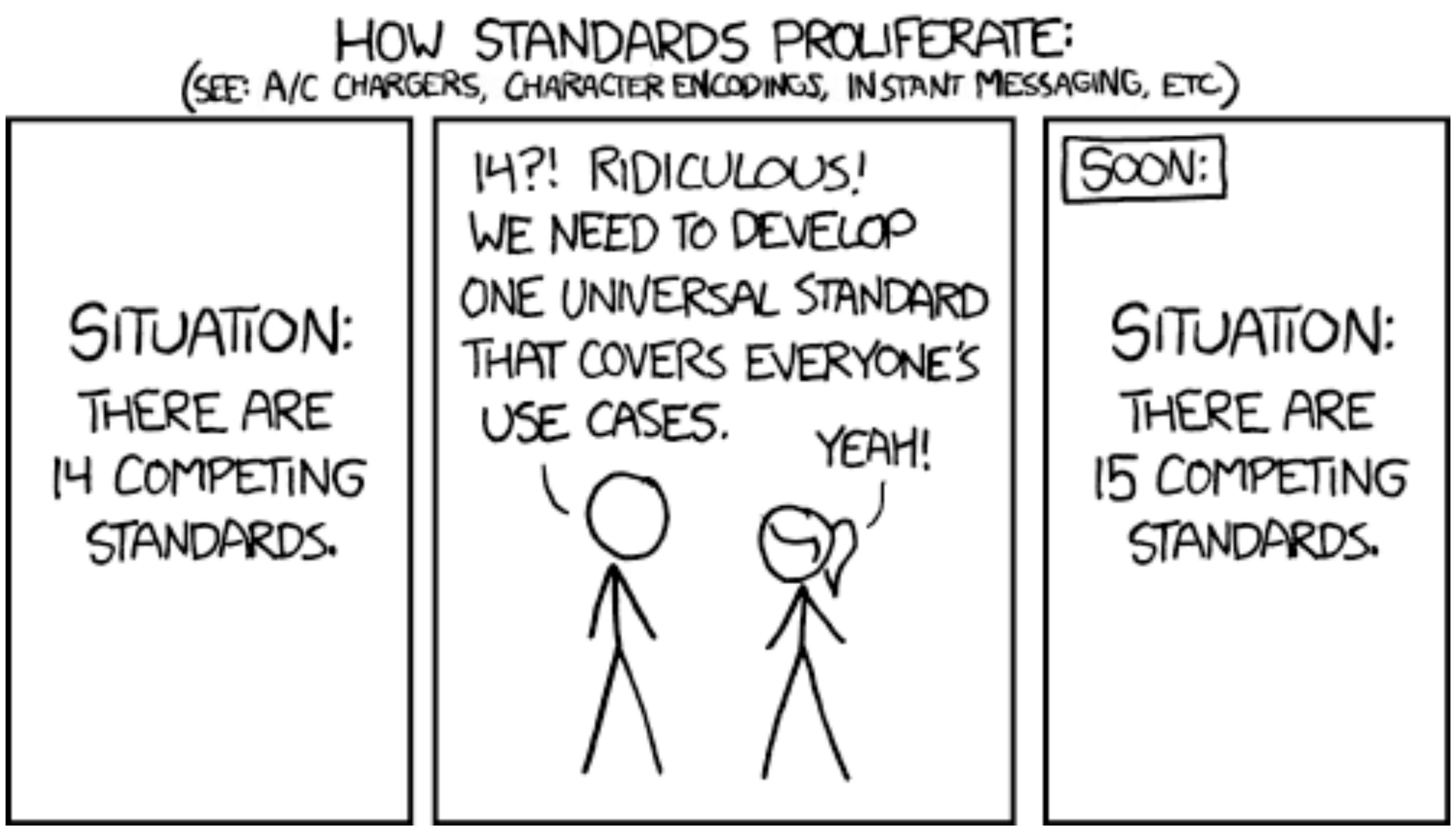
Hydra

Mason

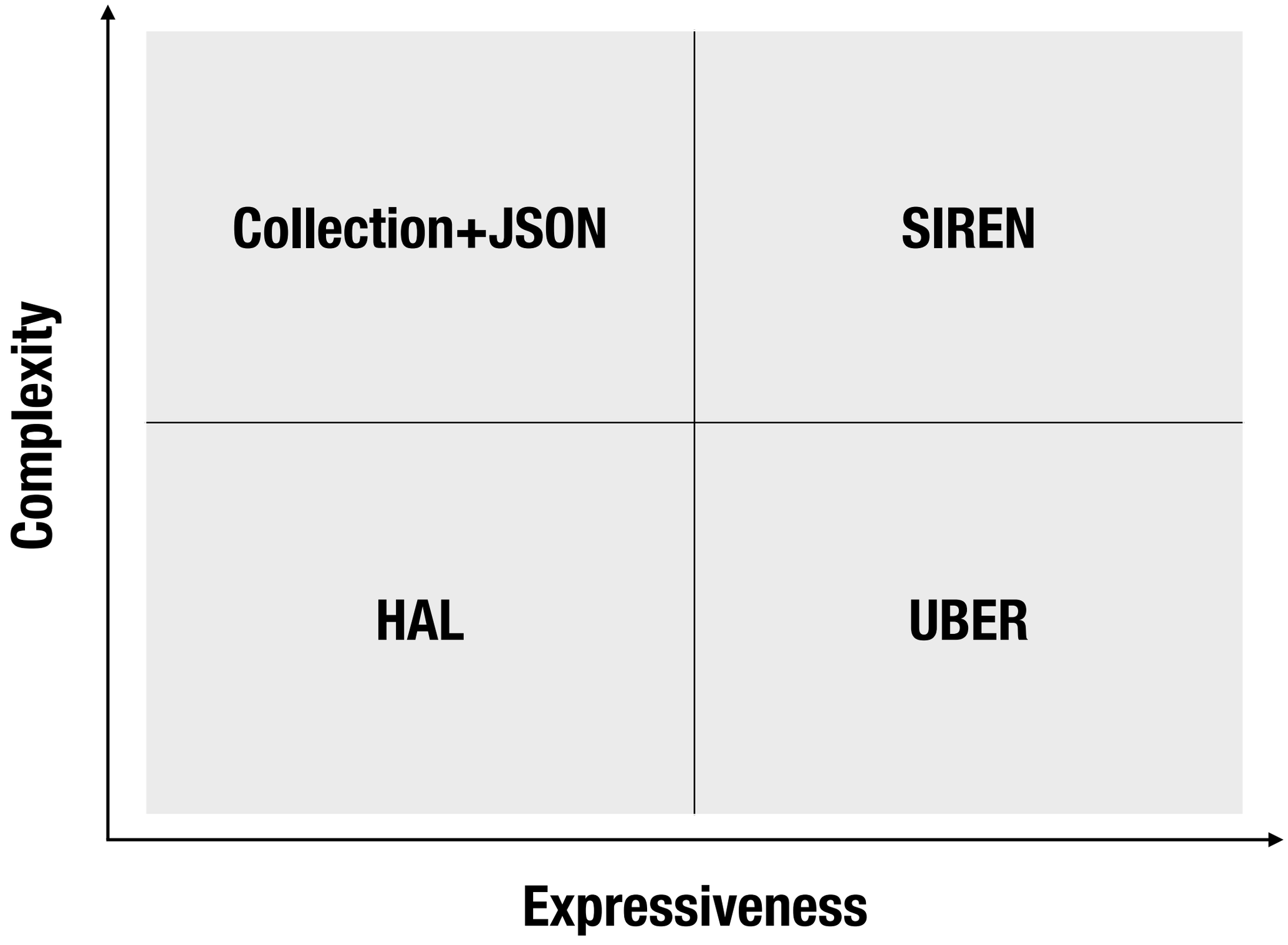
SIREN

UBER

...



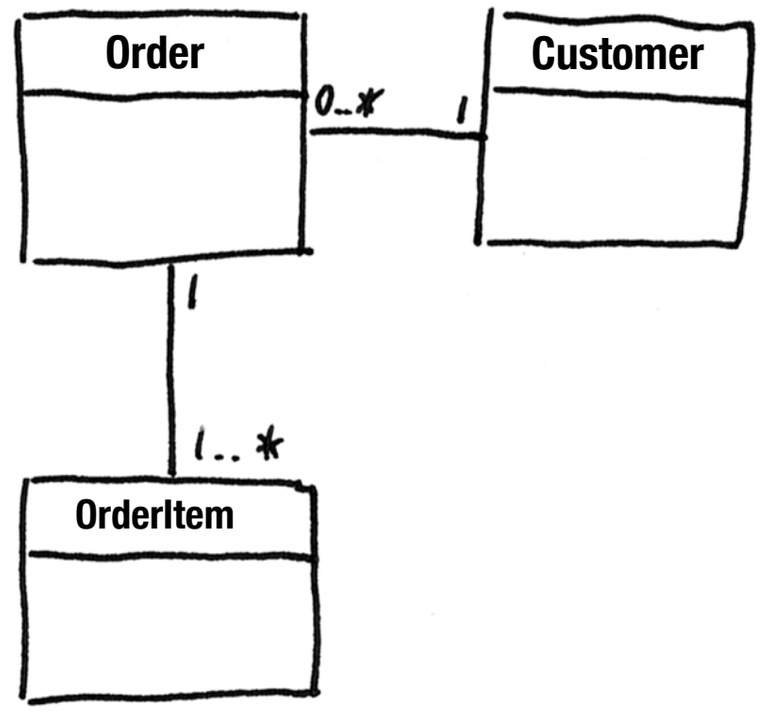
Source: http://xkcd.com/927/


Collection+JSON
SIREN
HAL
UBER
Complexity
Expressiveness

SIREN

SIREN example

```
{
  "class": [ "order" ],
  "properties": {
      "orderNumber": 42,
      "itemCount": 3,
      "status": "pending"
  },
  "entities": [
    {
      "class": [ "items", "collection" ],
      "rel": [ "http://x.io/rels/order-items" ],
      "href": "http://api.x.io/orders/42/items"
    },
    {
      "class": [ "info", "customer" ],
      "rel": [ "http://x.io/rels/customer" ],
      "properties": {
        "customerId": "pj123",
        "name": "Peter Joseph"
      },
      "links": [
        { "rel": [ "self" ], "href": "http://api.x.io/customers/pj123" }
      ]
    }
  ],
  "actions": [
    {
      "name": "add-item",
      "title": "Add Item",
      "method": "POST",
      "href": "http://api.x.io/orders/42/items",
      "type": "application/x-www-form-urlencoded",
      "fields": [
        { "name": "orderNumber", "type": "hidden", "value": "42" },
        { "name": "productCode", "type": "text" },
        { "name": "quantity", "type": "number" }
      ]
    }
  ],
  "links": [
    { "rel": [ "self" ], "href": "http://api.x.io/orders/42" },
    { "rel": [ "previous" ], "href": "http://api.x.io/orders/41" },
    { "rel": [ "next" ], "href": "http://api.x.io/orders/43" }
  ]
}
```



```
{
  "class": [ "order" ],
  "properties": {
      "orderNumber": 42,
      "itemCount": 3,
      "status": "pending"
  },
  "entities": [
    {
      "class": [ "items", "collection" ],
      "rel": [ "http://x.io/rels/order-items" ],
      "href": "http://api.x.io/orders/42/items"
    },
    {
      "class": [ "info", "customer" ],
      "rel": [ "http://x.io/rels/customer" ],
      "properties": {
        "customerId": "pj123",
        "name": "Peter Joseph"
      },
      "links": [
        { "rel": [ "self" ], "href": "http://api.x.io/customers/pj123" }
      ]
    }
  ],
  "actions": [
    {
      "name": "add-item",
      "title": "Add Item",
      "method": "POST",
      "href": "http://api.x.io/orders/42/items",
      "type": "application/x-www-form-urlencoded",
      "fields": [
        { "name": "orderNumber", "type": "hidden", "value": "42" },
        { "name": "productCode", "type": "text" },
        { "name": "quantity", "type": "number" }
      ]
    }
  ],
  "links": [
    { "rel": [ "self" ], "href": "http://api.x.io/orders/42" },
    { "rel": [ "previous" ], "href": "http://api.x.io/orders/41" },
    { "rel": [ "next" ], "href": "http://api.x.io/orders/43" }
  ]
}
```

An entity (“Order”)

The entity’s properties (state)

Collection of related (sub-)entities

First sub-entity:
collection of order items (linked)

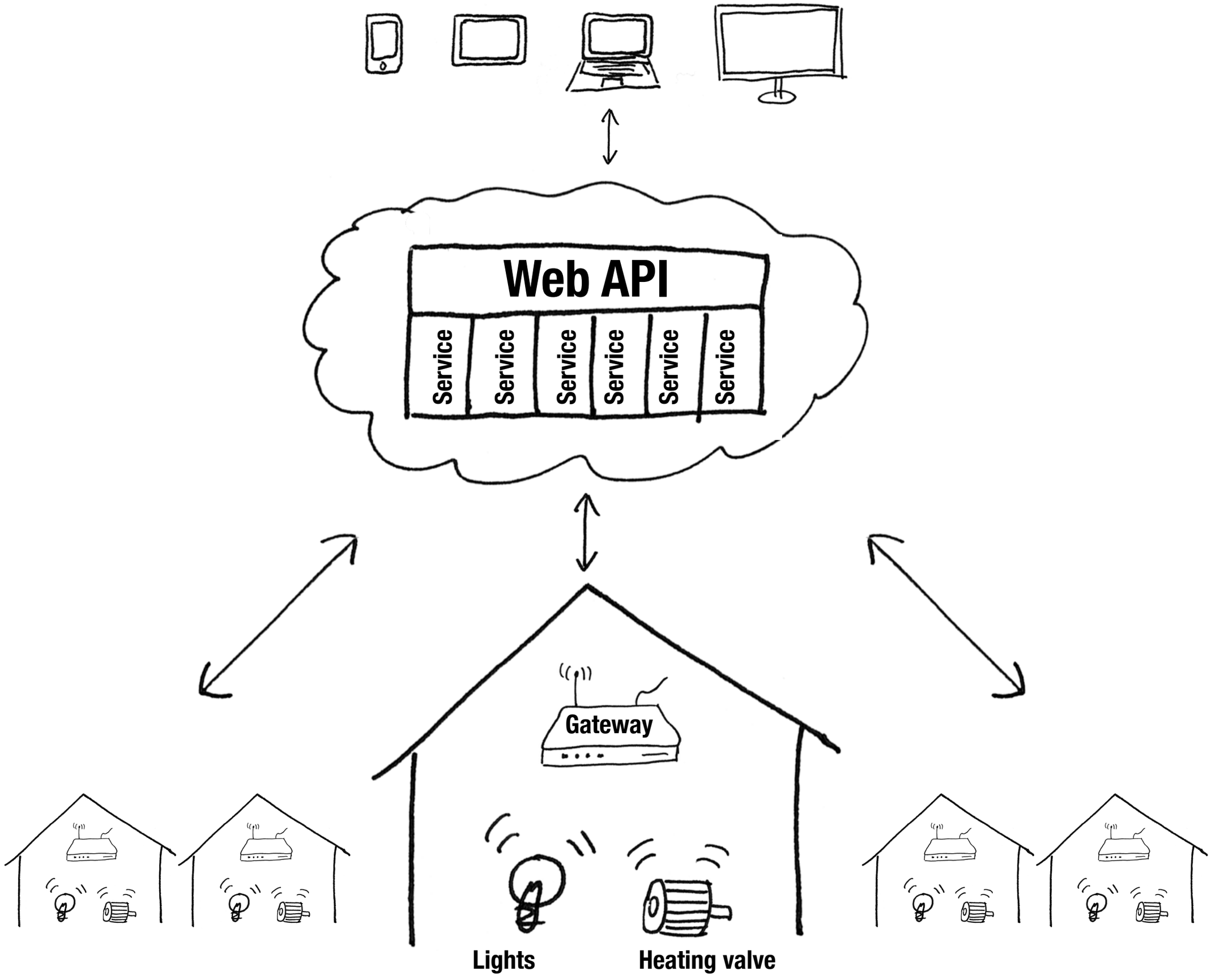
Second sub-entity:
customer (embedded)

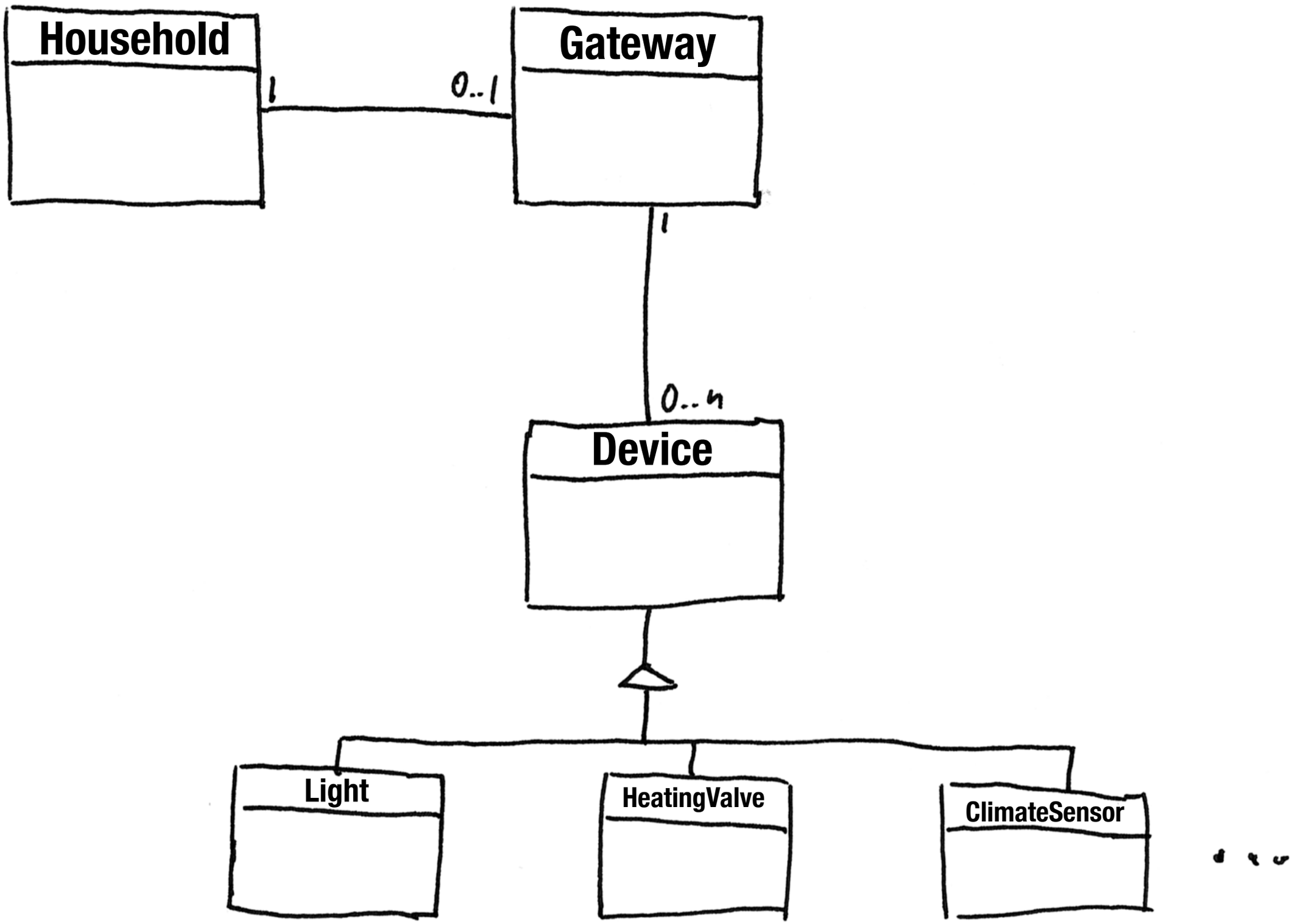
The actions that can be performed based on the current state of the entity

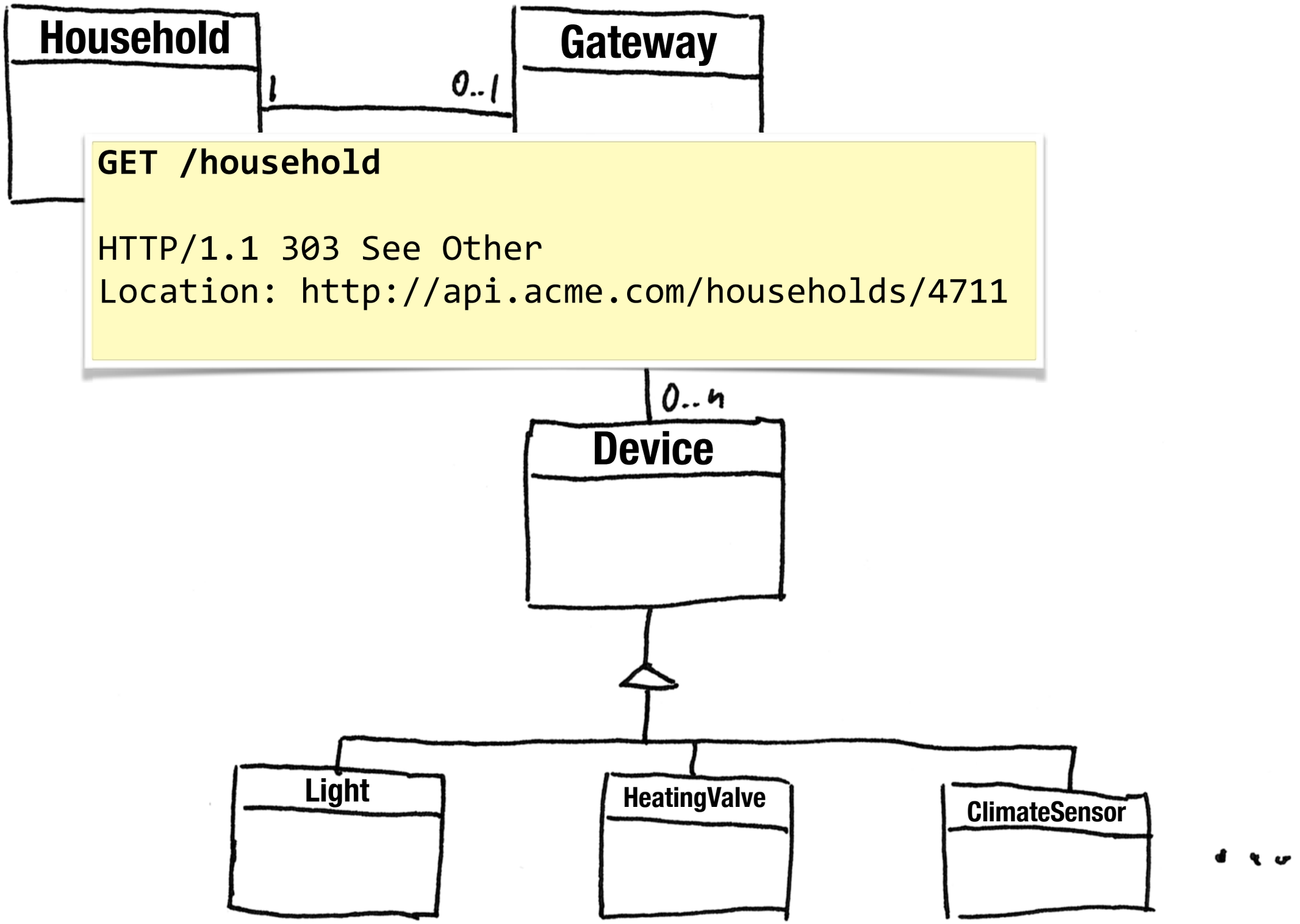
One action: “add-item”

Navigational links

The Use Case


Web API
Service
Service
Service
Service
Service
Service
Gateway
Lights
Heating valve


Household
1
0..1
Gateway
1
0..n
Device
Light
HeatingValve
ClimateSensor
...


Household
Gateway
1
0..1
GET /household
HTTP/1.1 303 See Other
Location: http://api.acme.com/households/4711
0..n
Device
Light
HeatingValve
ClimateSensor

GET /households/4711

```
{
    "class": ["household"],
    "properties": {
        "household_id": 4711,
        "location": "Berlin, Germany"
    },
    "entities": [
        {
            "class": ["gateway"],
            "rel": ["http://acme.com/rels/gateway"],
            "href": "http://api.acme.com/households/4711/gateway"
        },
        {
            "class": ["devices", "collection"],
            "rel": ["http://acme.com/rels/devices"],
            "href": "http://api.acme.com/households/4711/devices"
        }
    ],
    "links": [
        { "rel": ["self"], "href": "http://api.acme.com/households/4711" }
    ]

}
```

Entity “Household”

Related entity “Gateway” (linked)

Collection of “Device” (linked)

Self link

GET /households/4711/gateway

```
{
    "class": ["gateway"],
    "properties": {
        "gateway_id": 0815,
        "mode": "NORMAL_OPERATION"
    },
    "actions": [
        {
            "name": "set-mode",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/gateway/mode",
            "type": "application/json",
            "fields": [
                { "name": "mode", "type": "radio" }
            ]
        }
    ],
    "links": [
        { "rel": ["self"],
          "href": "http://api.acme.com/households/4711/gateway" }
    ]
}
```

Entity “Gateway”

Valid actions on “Gateway” based on current state

Self link

GET /households/4711/devices

```
{
    "class": ["devices", "collection"],
    "entities": [
        {
            "class": ["device", "collection-item"],
            "rel": ["http://acme.com/rels/device"],
            "href": "http://api.acme.com/households/4711/devices/1234"
        },
        {
            "class": ["device", "collection-item"],
            "rel": ["http://acme.com/rels/device"],
            "href": "http://api.acme.com/households/4711/devices/5678"
        },
        {
            "class": ["device", "collection-item"],
            "rel": ["http://acme.com/rels/device"],
            "href": "http://api.acme.com/households/4711/devices/9012"
        }
    ],
    "links": [
        { "rel": ["self"],
          "href": "http://api.acme.com/households/4711/devices" }
    ]
}
```

Collection of “Device” entities

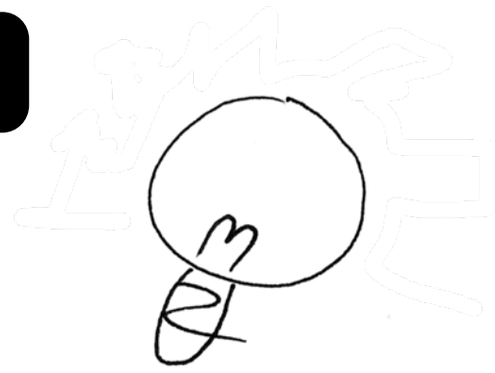
Entity “Device” (linked)

Self link

```
GET /households/4711/devices/5678
```

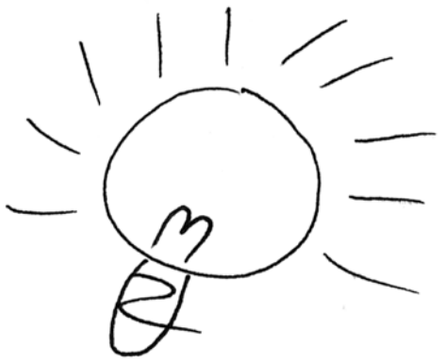
```
{
    "class": ["device"],
    "properties": {
        "device_id": 5678,
        "device_type": "SWITCHABLE_LIGHT",
        "display_name": "Living room ambient lamp"
        "power_status": "off"
    },
    "actions": [
        {
            "name": "power-on",
            "method": "POST",
            "href": "http://api.acme.com/households/4711/devices/5678/power_on",
        },
        {
            "name": "set-display_name",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/devices/5678/display_name",
            "type": "application/json",
            "fields": [{ "name": "display_name", "type": "text"}]
        }
    ],
    "links": [
        { "rel": ["self"], "href": “http://api.acme.com/households/4711/devices/5678" }
    ]
}
```

Entity “Device”



Valid actions on “Device” based on current state

GET /households/4711/devices/5678




```
{
    "class": ["device"],
    "properties": {
        "device_id": 5678,
        "device_type": "SWITCHABLE_LIGHT",
        "display_name": "Living room ambient lamp"
        "power_status": "on"
    },
    "actions": [
        {
            "name": "power-off",
            "method": "POST",
            "href": "http://api.acme.com/households/4711/devices/5678/power_off",
        },
        {
            "name": "set-display_name",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/devices/5678/display_name",
            "type": "application/json",
            "fields": [{ "name": "display_name", "type": "text"}]
        }
    ],
    "links": [
        { "rel": ["self"], "href": "http://api.acme.com/households/4711/devices/5678" }
    ]
}
```

GET /households/4711/devices/5678

API change!

```
{
    "class": ["device"],
    "properties": {
        "device_id": 5678,
        "device_type": "SWITCHABLE_LIGHT",
        "display_name": "Living room ambient lamp"
        "power_status": "on"
    },
    "actions": [
        {
            "name": "power-off",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/devices/5678/power_status",
            "type": "application/json",
            "fields": [{ "name": "power_status", "type": "text", "value": "off"}]
        },
        {
            "name": "set-display_name",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/devices/5678/display_name",
            "type": "application/json",
            "fields": [{ "name": "display_name", "type": "text"}]
        }
    ],
    "links": [
        { "rel": ["self"], "href": “http://api.acme.com/households/4711/devices/5678" }
    ]
}
```

GET /households/4711/devices/1234

```
{
    "class": ["device"],
    "properties": {
        "device_id": 1234,
        "device_type": "HEATING_VALVE",
        "display_name": "Heating Valve Living Room”,
        "current_temperature": 20,
        "set_temperature": 22
    },
    "actions": [
        {
            "name": "set-set_temperature",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/devices/1234/set_temperature",
            "type": "application/json",
            "fields": [{ "name": "set_temperature", "type": "number"}]
        },
        {
            "name": "set-display_name",
            "method": "PUT",
            "href": "http://api.acme.com/households/4711/devices/1234/display_name",
            "type": "application/json",
            "fields": [{ "name": "display_name", "type": "text"}]
        }
    ],
    "links": [
        { "rel": ["self"], "href": “http://api.acme.com/households/4711/devices/1234" }
    ]
}
```

See why hypermedia makes sense?

Oh – one more thing...

Didn’t we just reinvent HTML?

That’s all I have.

Feel free to ask me anything!

@owolf

Meet us at our booth in the exhibition hall!

innoQ