REST Constraints

Shawn Wildermuth

@shawnwildermuth | wilderminds.com

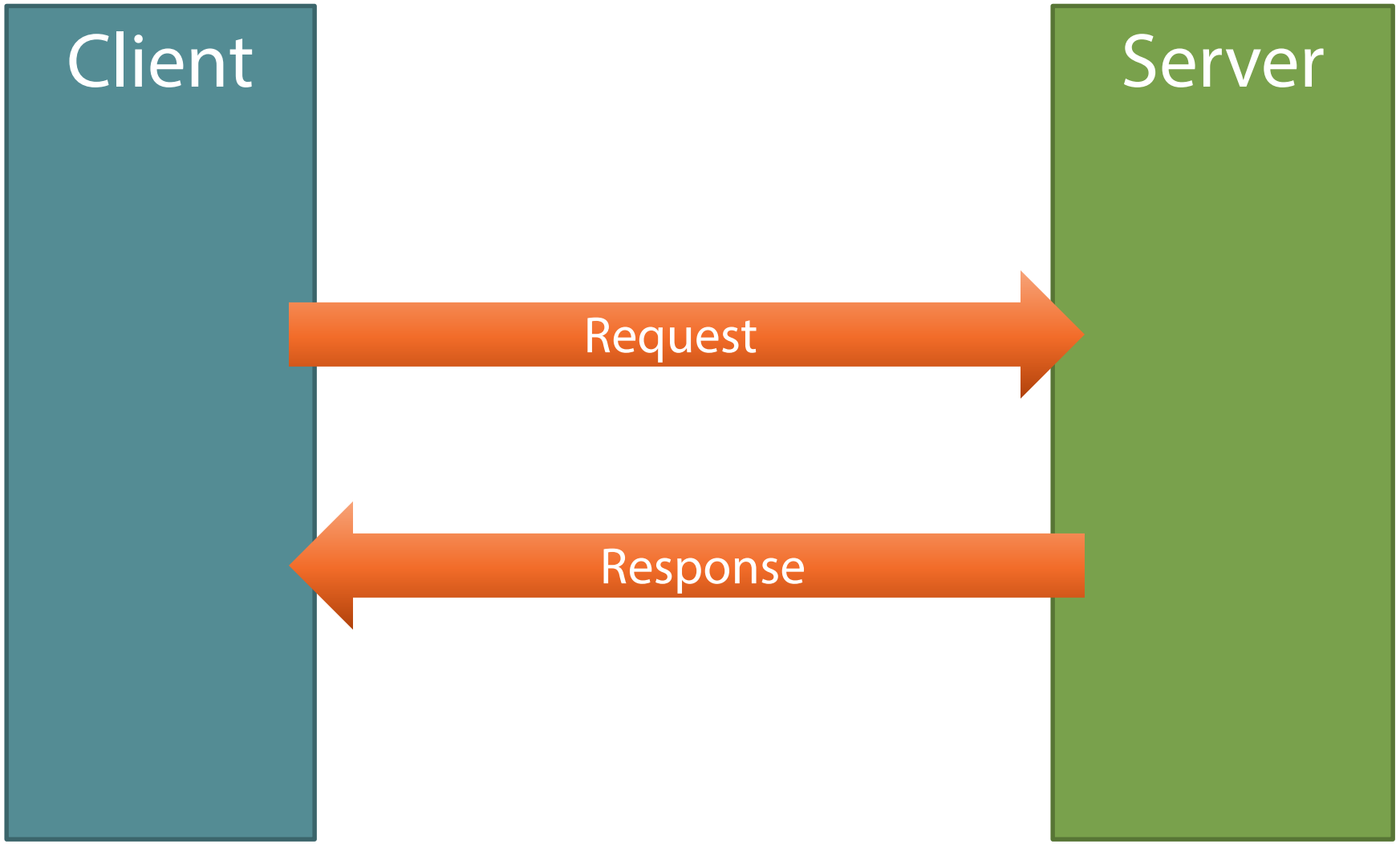
This Module

REST Constraints

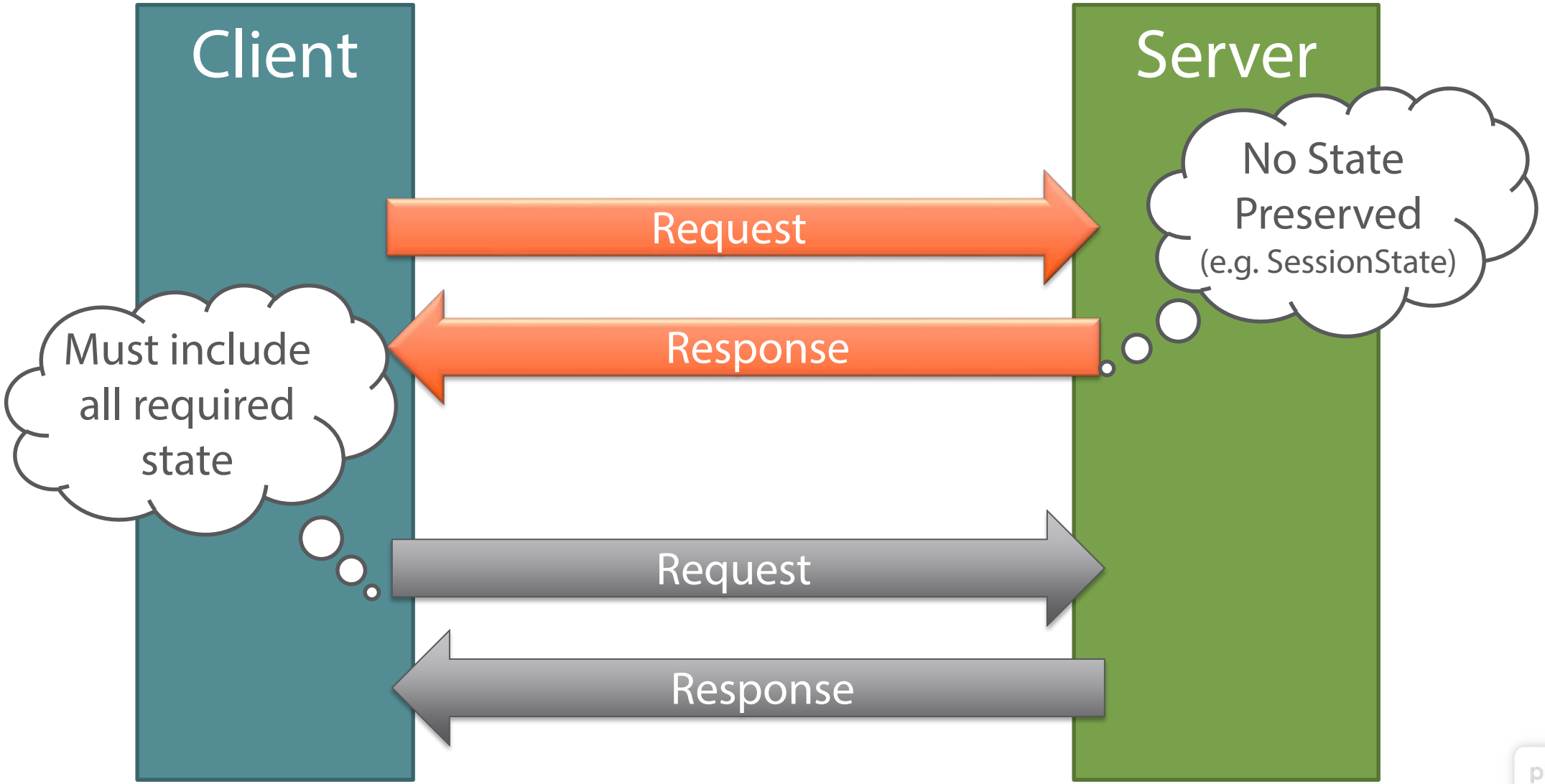
What Are the Constraints of REST?

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

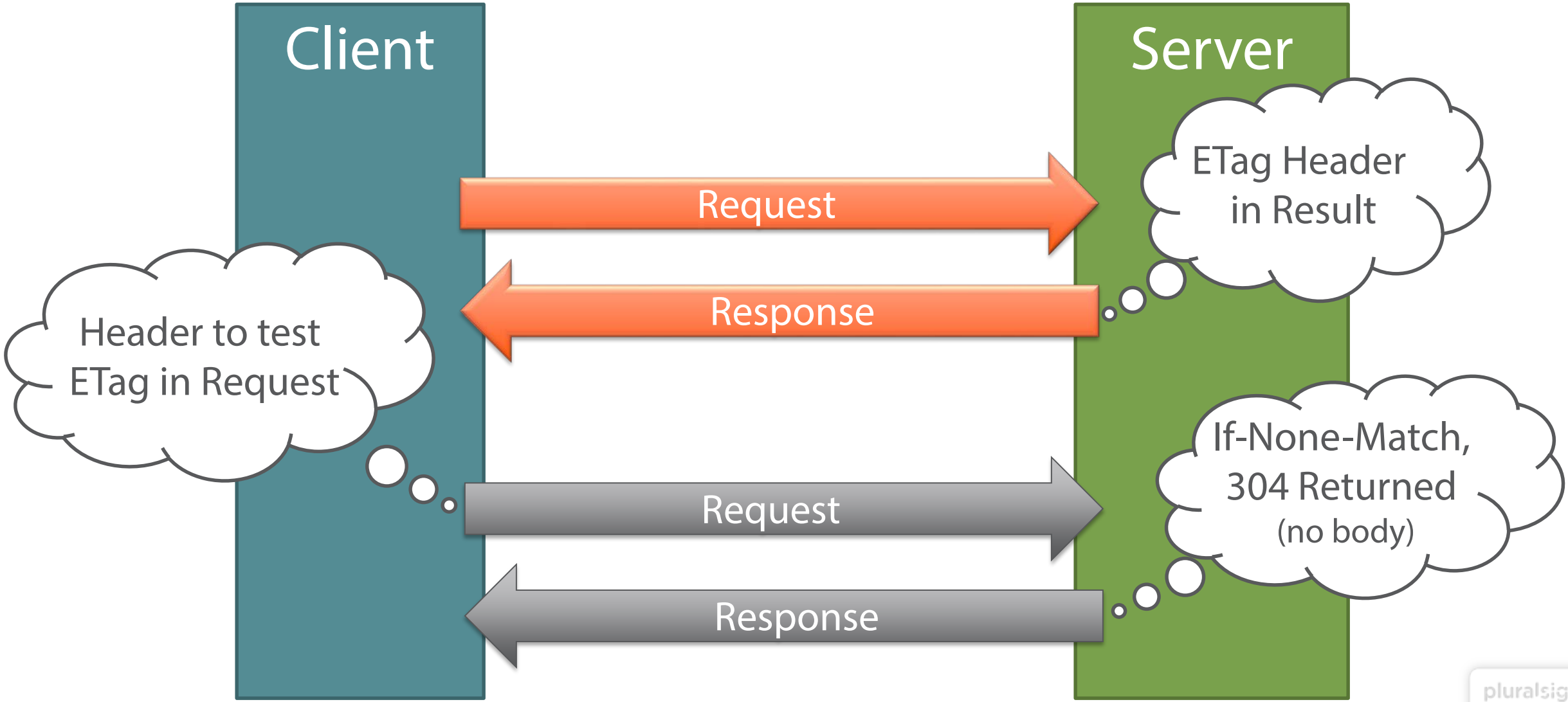
Client-Server



Stateless Server



Cache



What Are ETags?

- Header used as a unique key for resource

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Date: Thu, 23 May 2013 21:52:14 GMT
ETag: W/"4893023942098"
Content-Length: 639
```

- Requests should test using If-None-Match:

```
GET /api/nutrition/foods/2 HTTP/1.1
Accept: application/json, text/xml
Host: localhost:8863
If-None-Match: "4893023942098"
```

```
HTTP/1.1 304 Not Modified
```

What Are ETags?

- For PUT/PATCH it is different

```
PATCH /api/user/diaries/2013-5-12 HTTP/1.1
Accept: application/json, text/xml
Host: localhost:8863
If-Match: "4893023942098"
...
```

```
HTTP/1.1 412 Precondition Failed
```

Demo

Implementing etags on GET

Demo

Implementing etags on PUT/DELETE

Uniform Interface

- Defined in four parts:
  - Identification of resources

```
http://.../api/nutrition/foods/12345
```

```
http://.../api/user/diaries/2013-5-24
```

```
http://.../api/user/diaries/2013-5-24/entries/12
```

Uniform Interface

- Defined in four parts:
  - Manipulation of resources through representations
    - Same Structures can be sent back as are received through the API
    - Using the standard HTTP verbs to represent the operation

Uniform Interface

- Defined in four parts:
  - Self-descriptive messages
  - Hypermedia as the engine of application state
    - Represented by links in the results

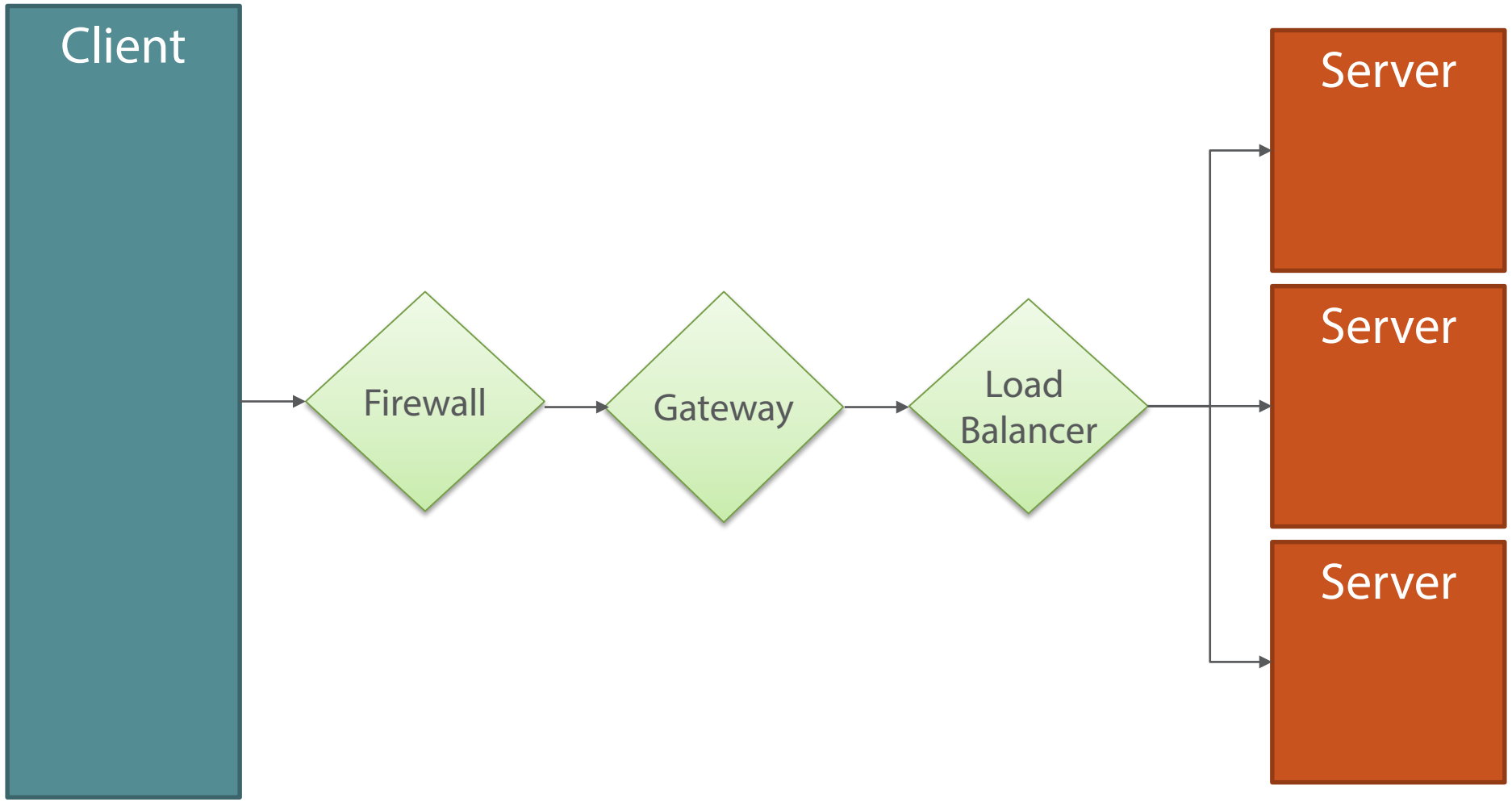
Links

```
{
  "links":[ {
      "href":"http://.../api/camps/ATL2016",
      "rel":"self",
      "method”: "GET"
    },{
      "href":"http://.../api/camps/ATL2016/speaker/1 ",
      "rel":"createSpeaker",
      "method": "POST"
    }],
  ...
}
```

Demo

Implementing Links

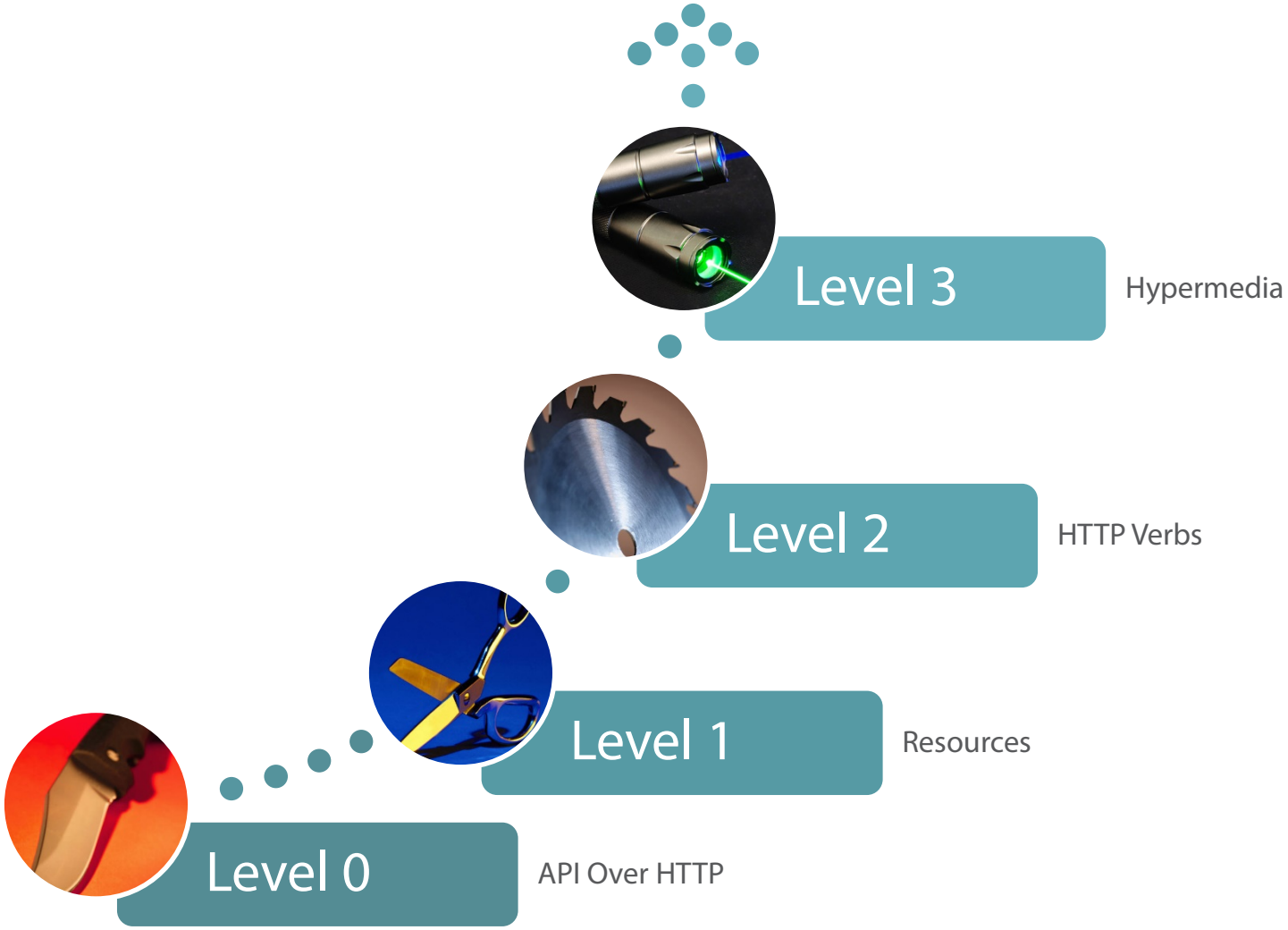
Layered System



Code On-Demand

- Mentioned as Optional by Original REST Paper
  - The ability to deliver code from the server for the client to execute
  - Very rarely adopted
  - Possible in the JavaScript world, but rarely would clients trust the server

A REST Maturity Model*



* Leonard Richardson's Model of REST 'Completeness'
http://shawnw.me/restrmm

Pragmatism

- Richardson Maturity Model is used as a Cudgel
  - Build "Enough" maturity for your API
  - Over-Building is worse than Under-Building
  - Remember your users
  - Don't get caught up in the Dogma Trap

What We've Learned

REST Constraints

Build enough REST for your needs

Caching and Links can help make your API easier to use

Don't get bogged down in the dogma