

RESTful "Hypermedia-Services" auf der JVM

BEDCon 2015

Michael Vitz (@michaelvitz)

innoQ



RESTful?

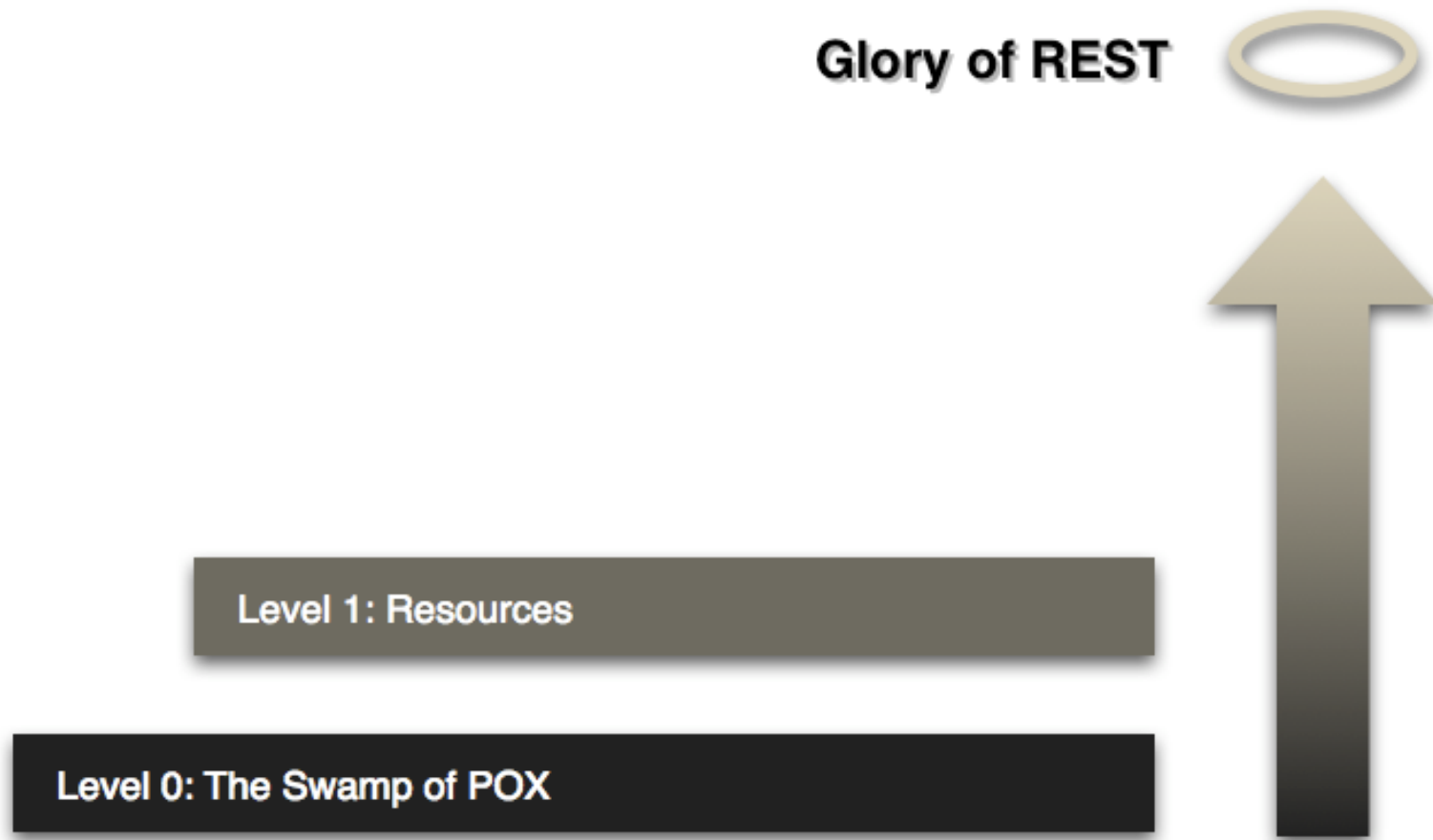
RPC over HTTP

Glory of REST

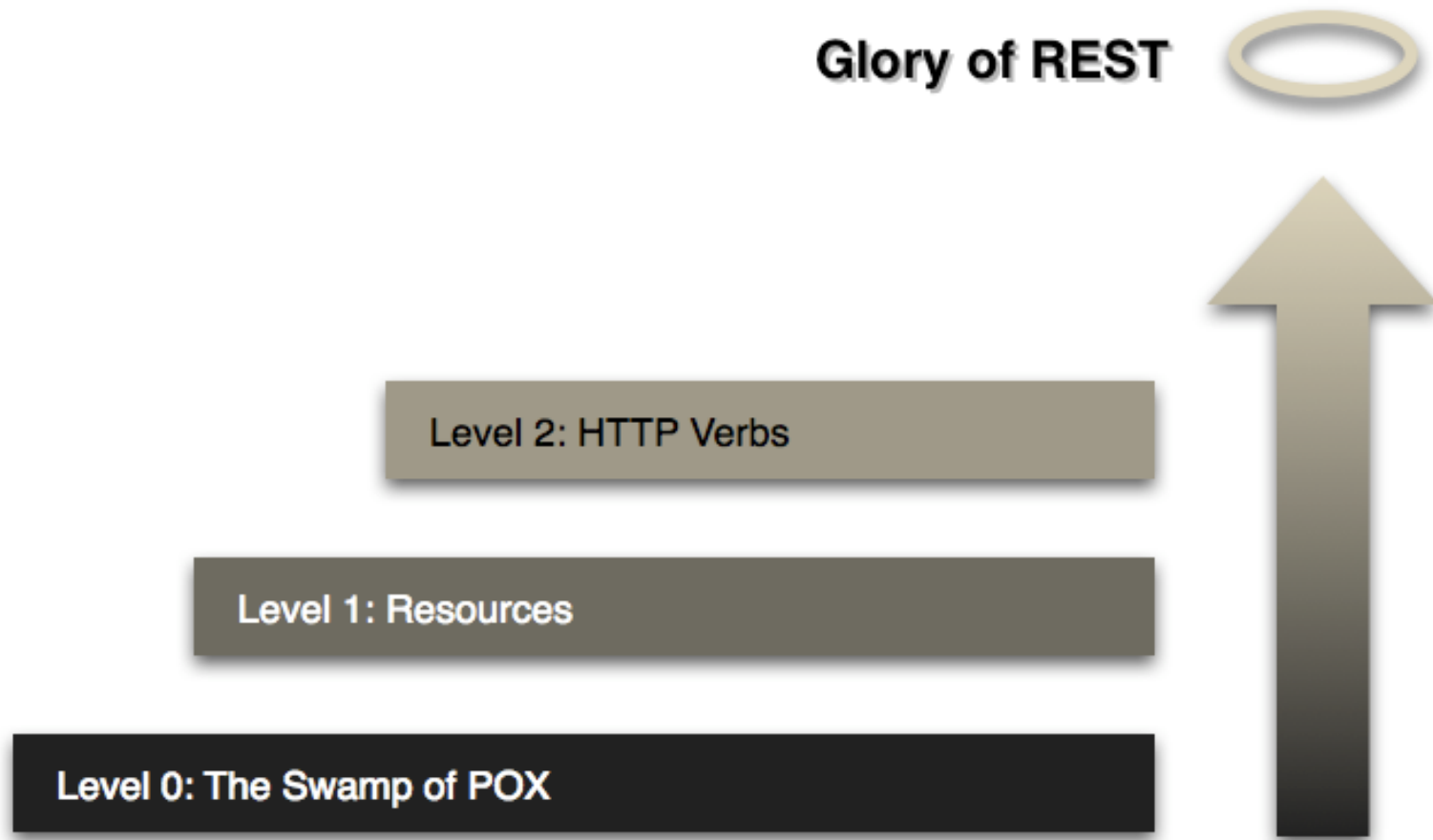


Level 0: The Swamp of POX

RPC over HTTP



RESTful



Methods?

GET, POST

PUT?

DELETE?

HEAD?

OPTIONS?

Methods!

	PUT	DELETE	HEAD	OPTIONS
JSF	✗	✗	✗	✗
Play	✓	✓	✓	✗
Spring	✓	✓	✓	✓
Liberator	✓	✓	✓	✓

Status Codes?

```
$ curl -iL playframework.com
```

```
> HTTP/1.1 200 OK
```

```
> Server: Play/2.4.3 Scala/2.11.7
```


Status Codes?

```
$ curl -iL -XDELETE playframework.com
```

```
> HTTP/1.1 404 Not Found
```

```
> Server: Play/2.4.3 Scala/2.11.7
```

Status Codes!

100-101

200-206

300-305, 307

400-417, 426

500-505

Status Codes!

	Support
JSF	✘
Play	✔
Spring	✔
Liberator	✔

Content Negotiation?

```
$ curl -H 'Accept: text/plain' http://www.foaas.com/version
```

```
Version 0.1.8 FOAAS
```

```
$ curl -H 'Accept: application/json' http://www.foaas.com/version
```

```
{"message": "Version 0.1.8", "subtitle": "FOAAS"}
```

Content Negotiation!

	Support
JSF	✘
Play	✔
Spring	✔
Liberator	✔

Caching?

```
$ curl -i an-resource
```

```
> HTTP/1.1 200 OK
```

```
> ETag: 2
```

```
$ curl -i an-resource -H 'If-None-Match: 2'
```

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

```
$ curl -i an-resource -H 'If-Match: 1' --data foo=bar
```

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

Caching!

	Support
JSF	✘
Play	✔
Spring	✔
Liberator	✔

RESTful!

What about links???

Hypermedia?

Hypermedia

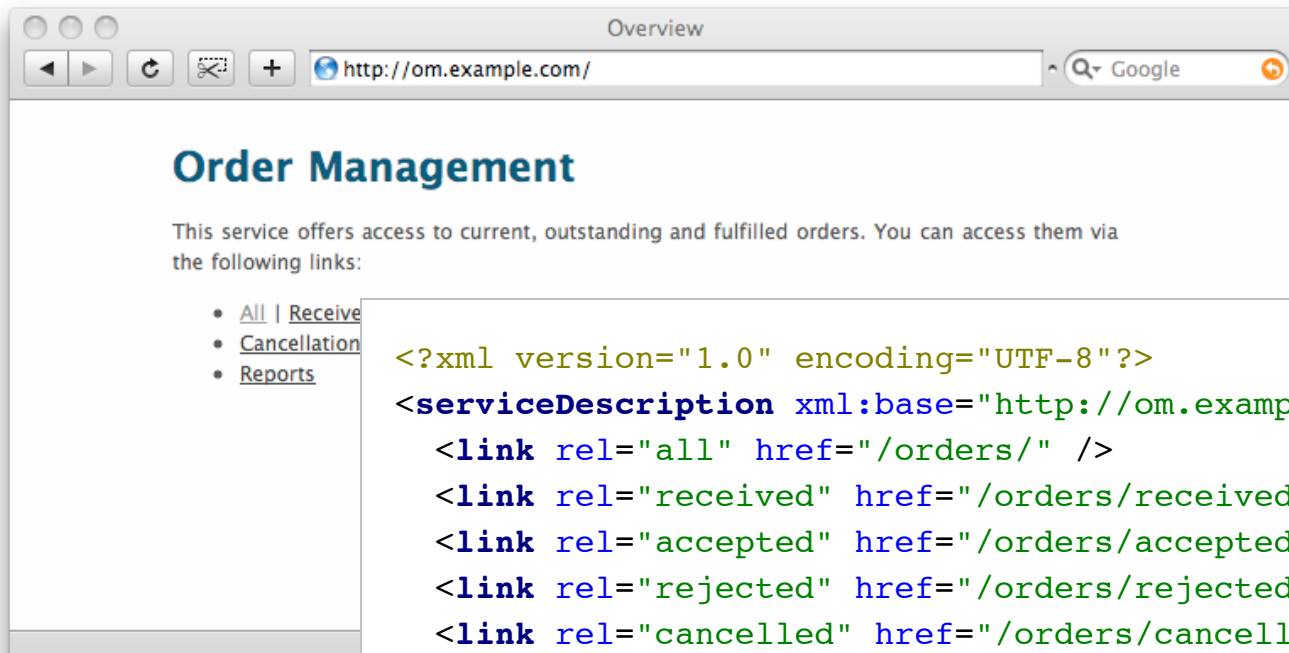
Server generates link

Client uses link

Nothing more???

Nothing more!!!!

Step 1: Service Documents



```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDescription xml:base="http://om.example.com">
  <link rel="all" href="/orders/" />
  <link rel="received" href="/orders/received/" />
  <link rel="accepted" href="/orders/accepted/" />
  <link rel="rejected" href="/orders/rejected/" />
  <link rel="cancelled" href="/orders/cancelled/" />
  <link rel="fulfilled" href="/orders/fulfilled/" />
  <link rel="cancellations" href="/cancellations/" />
  <link rel="reports" href="/reports/" />
</serviceDescription>
```

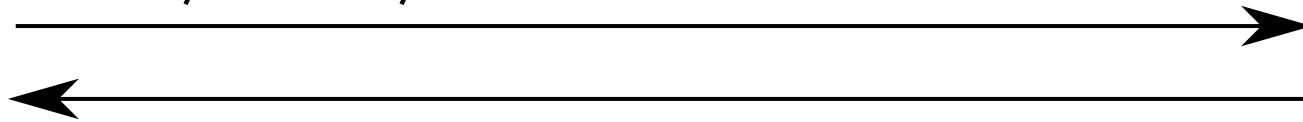
```
{ "serviceDescription" : {
  "base": "http://om.example.com",
  "links": [
    { "rel": "all", "href": "/orders/" },
    { "rel": "all", "href": "/orders/" },
    { "rel": "received", "href": "/orders/received/" },
    { "rel": "accepted", "href": "/orders/accepted/" },
    { "rel": "rejected", "href": "/orders/rejected/" },
    { "rel": "cancelled", "href": "/orders/cancelled/" },
    { "rel": "fulfilled", "href": "/orders/fulfilled/" },
    { "rel": "cancellations", "href": "/cancellations/" },
    { "rel": "reports", "href": "/reports/" }
  ]
}
```

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"]
      }
    }
  }
}
```


Step 2: Resource Links

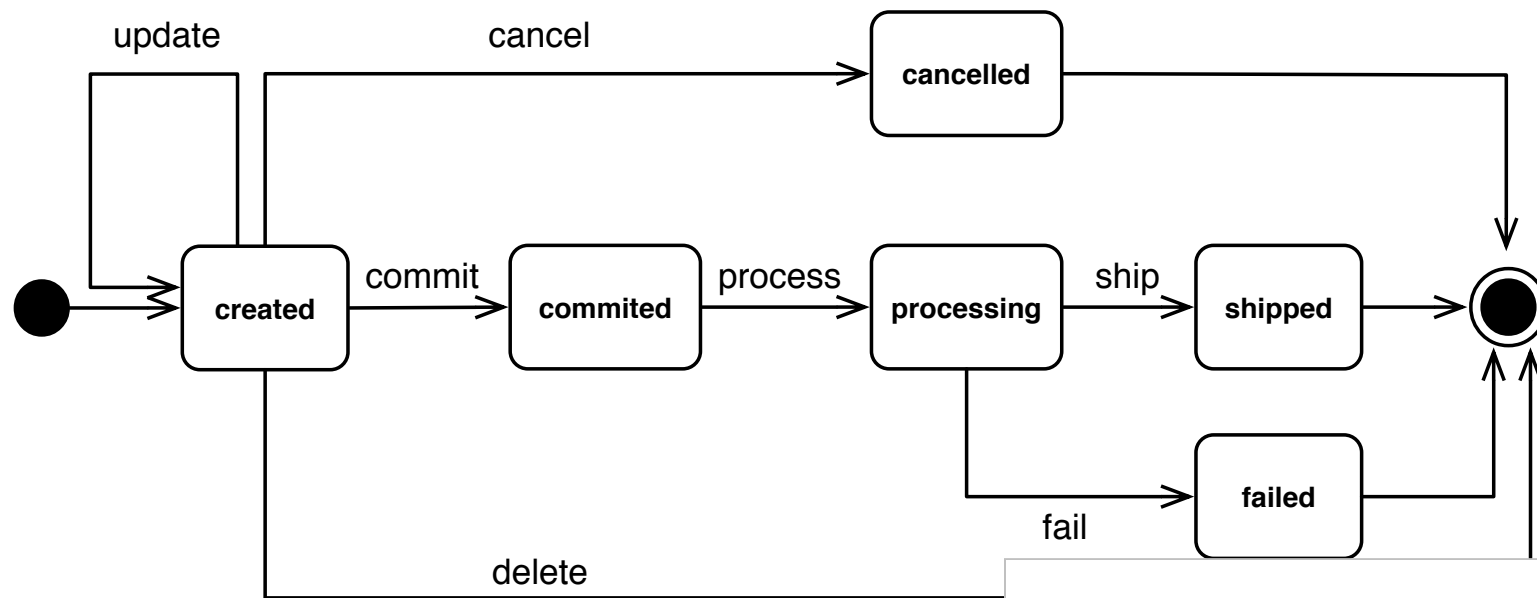
GET /order/123



```
{  
  "order": {  
    "date": "2013-07-02",  
    "total": 1240.02,  
    "...": "...",  
    "links": {  
      "self": "http://example.com/orders/123",  
      "contacts": "http://example.org/A3BEF1",  
      ...  
    }  
  }  
}
```

Your future extensions

Step 3: State Transition Links



```

{
  "order": {
    "state" : "created",
    "...": "...",
    "links" : {
      "cancel" : "http://...",
      "commit" : "http://...",
      "delete" : "http://...",
      ...
    }
  }
}

```

Step 4: Provide recipes for URIs

```
<form action="http://example.com/search" method="GET">
  Search for: <input type="text" name="query">
  <input type="submit">
</form>
```

http://example.com/search?query=xyz

```
{
  "links": {
    "search": {
      "template": "http://example.com/search?q={query}"
    }
  }
}
```

RFC 6570: URI Template [\[https://tools.ietf.org/html/rfc6570\]](https://tools.ietf.org/html/rfc6570)

```
<form action="http://example.com/add" method="POST">
  <label for="first">First:</label>
  <input type="text" name="first" id="first">
  <label for="last">Last:</label>
  <input type="text" name="last" id="last">
  <label for="birthday">Birthday:</label>
  <input type="date" name="birthday" id="birthday">
  <input type="submit">
</form>
```

```
{
  "links": {
    "add": {
      "href": "http://example.com/add",
      "template": {
        "first": "...",
        "last": "...",
        "birthday": "..."
      }
    }
  }
}
```

Hypermedia!

	Support
JSF	✗
Play	✓
Spring	✓
Liberator	✗

Hypermedia!



<http://clojure-liberator.github.io/>

Liberator

Version: 0.13

Inspired by: [Erlang's Webmachine](#)

Controller?

```
$ grep -ic controller rfc723{0..7}
```

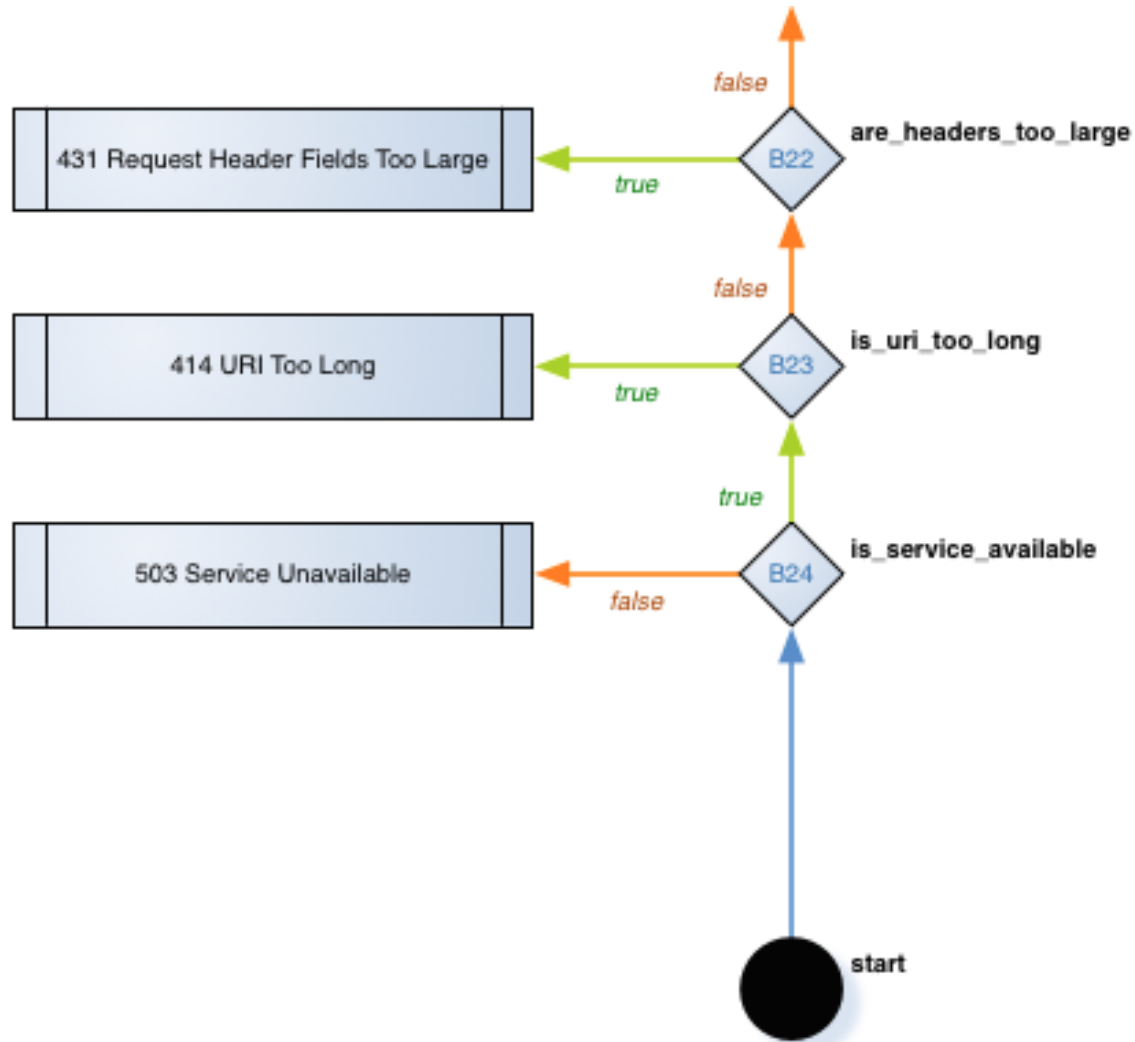
```
> 0
```



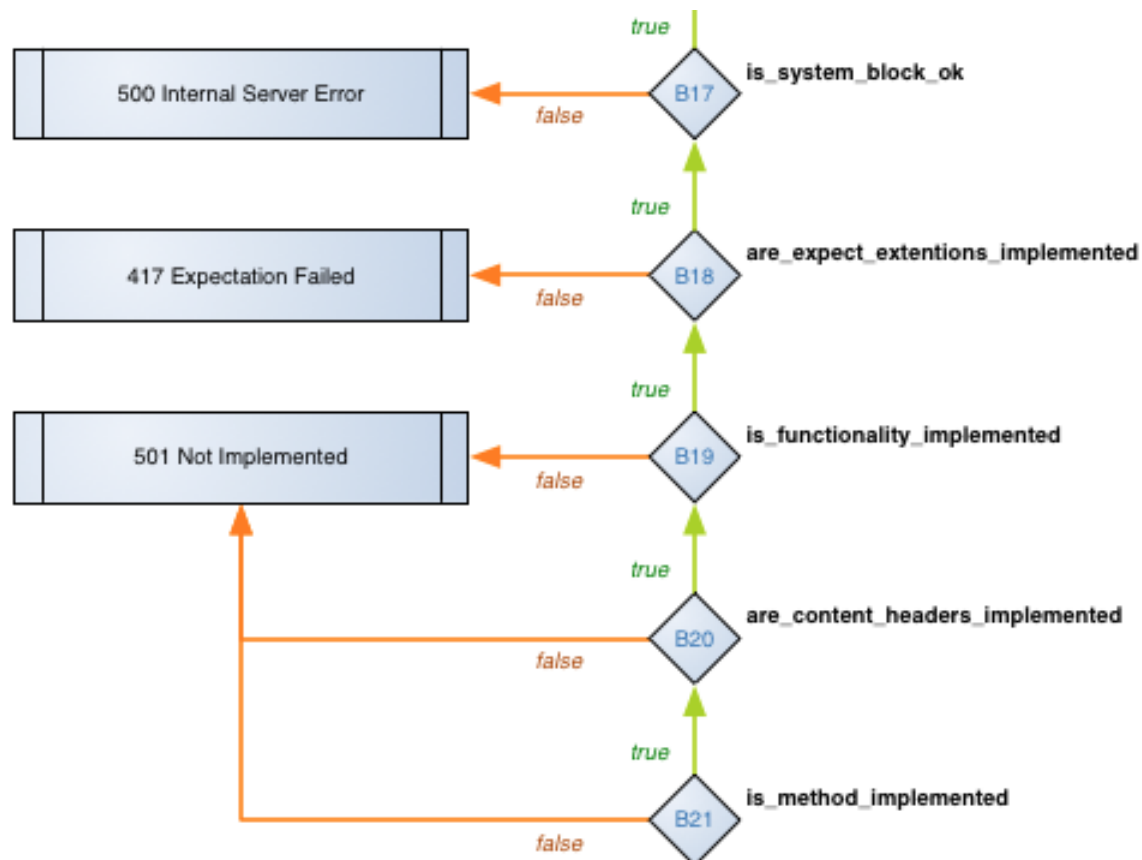
Resource!

```
(defresource hello-world  
  :available-media-types ["text/plain"]  
  :handle-ok "Hello, world!")
```

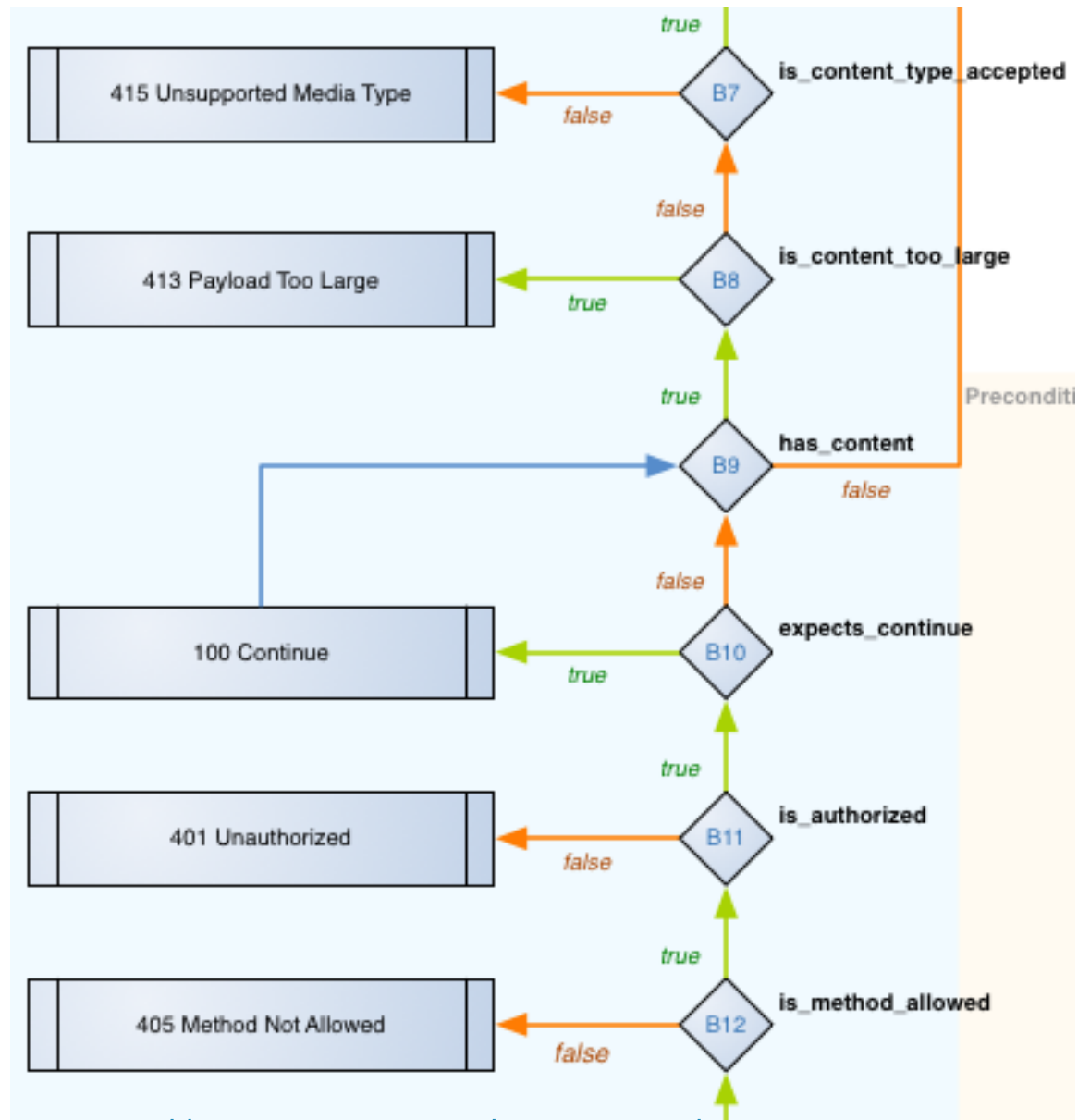
Decision Graph: System



Decision Graph: System

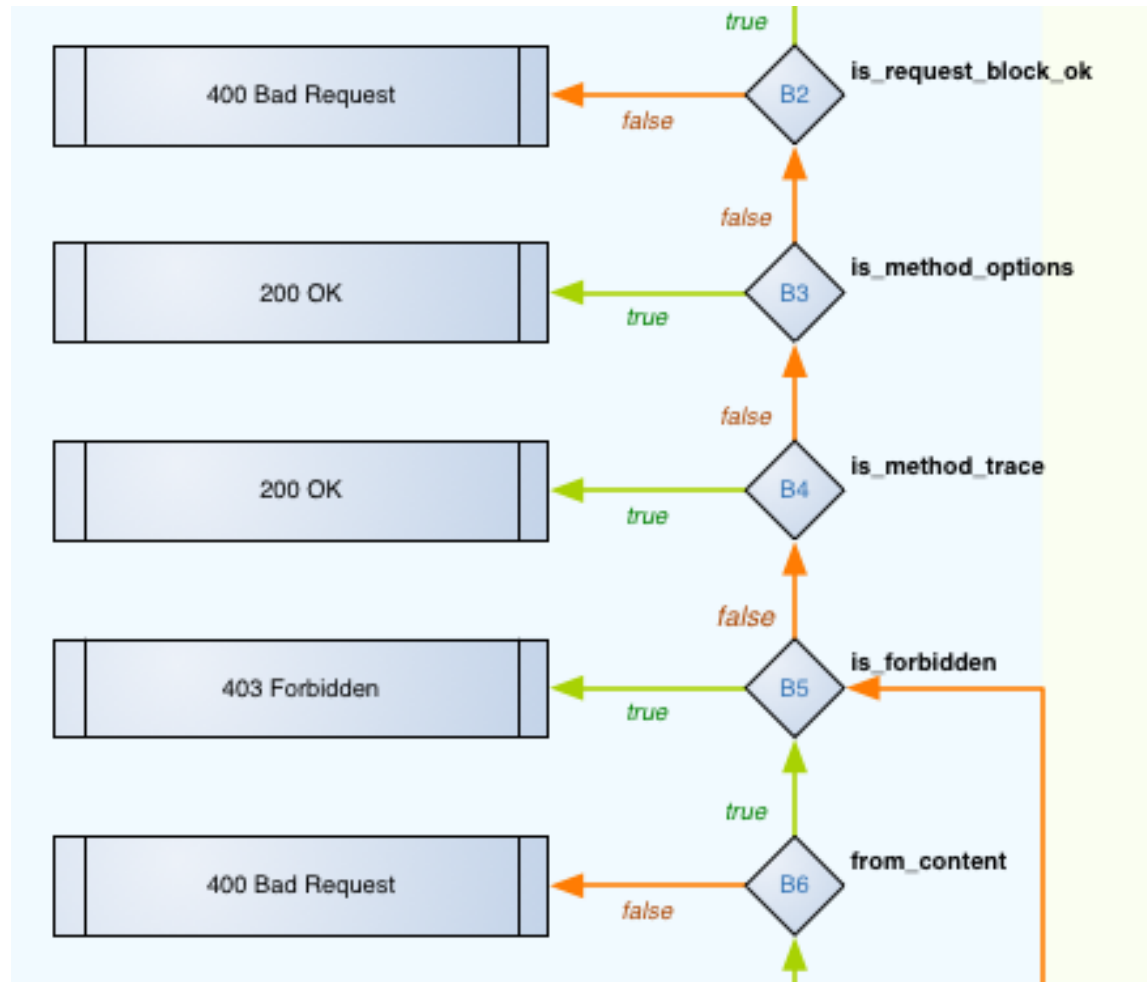


Decision Graph: Request

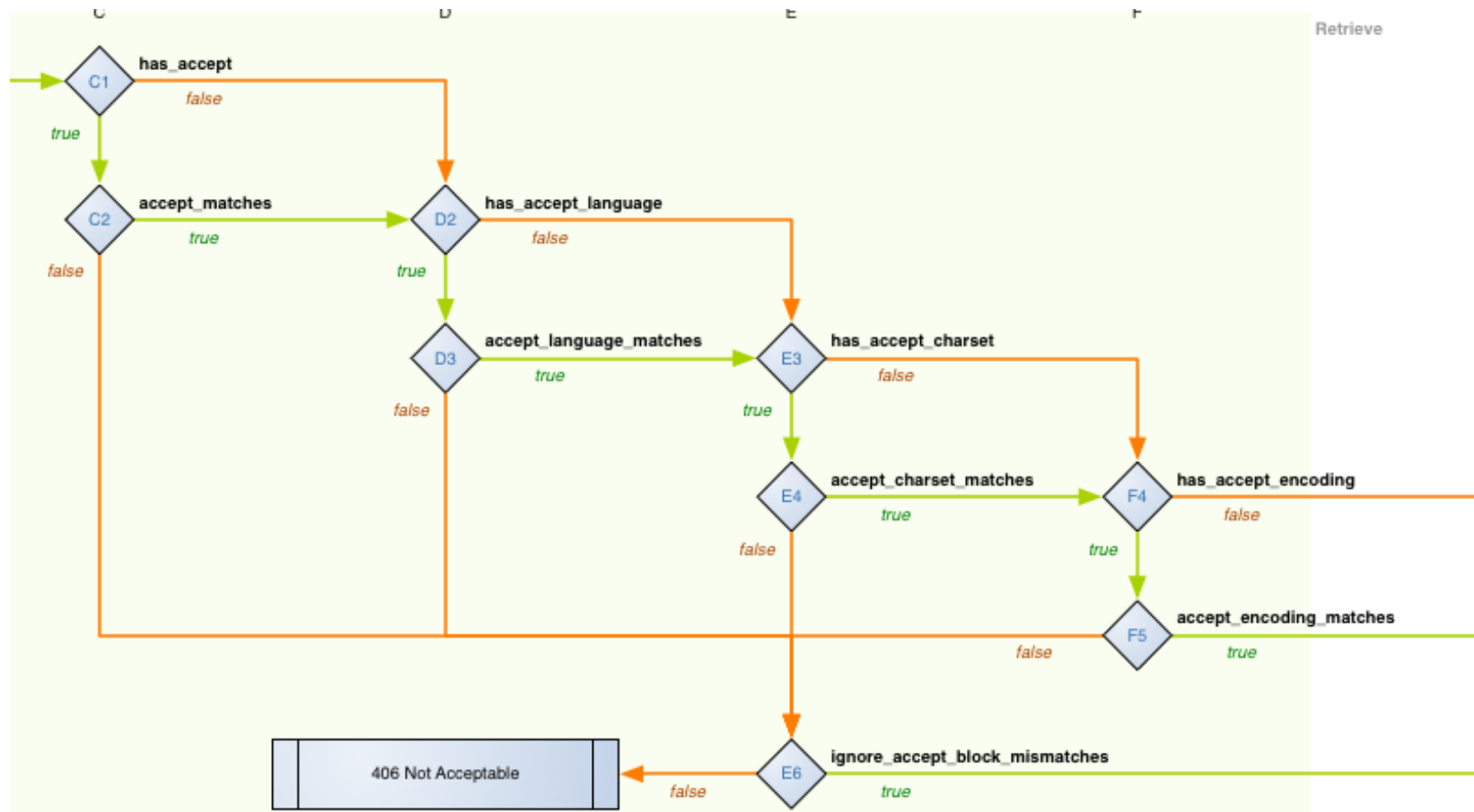


<https://github.com/for-GET/http-decision-diagram>

Decision Graph: Request

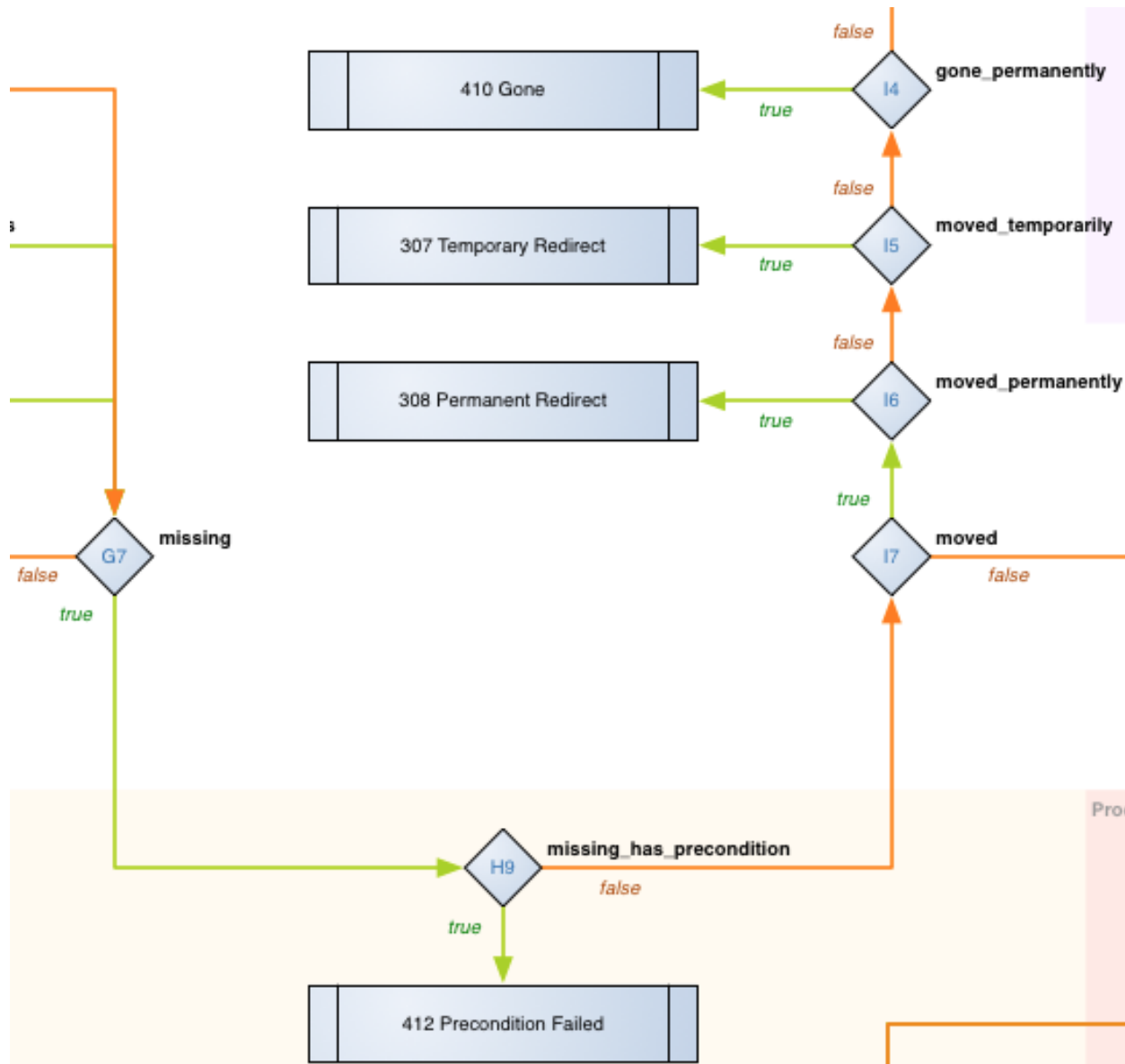


Decision Graph: Accept



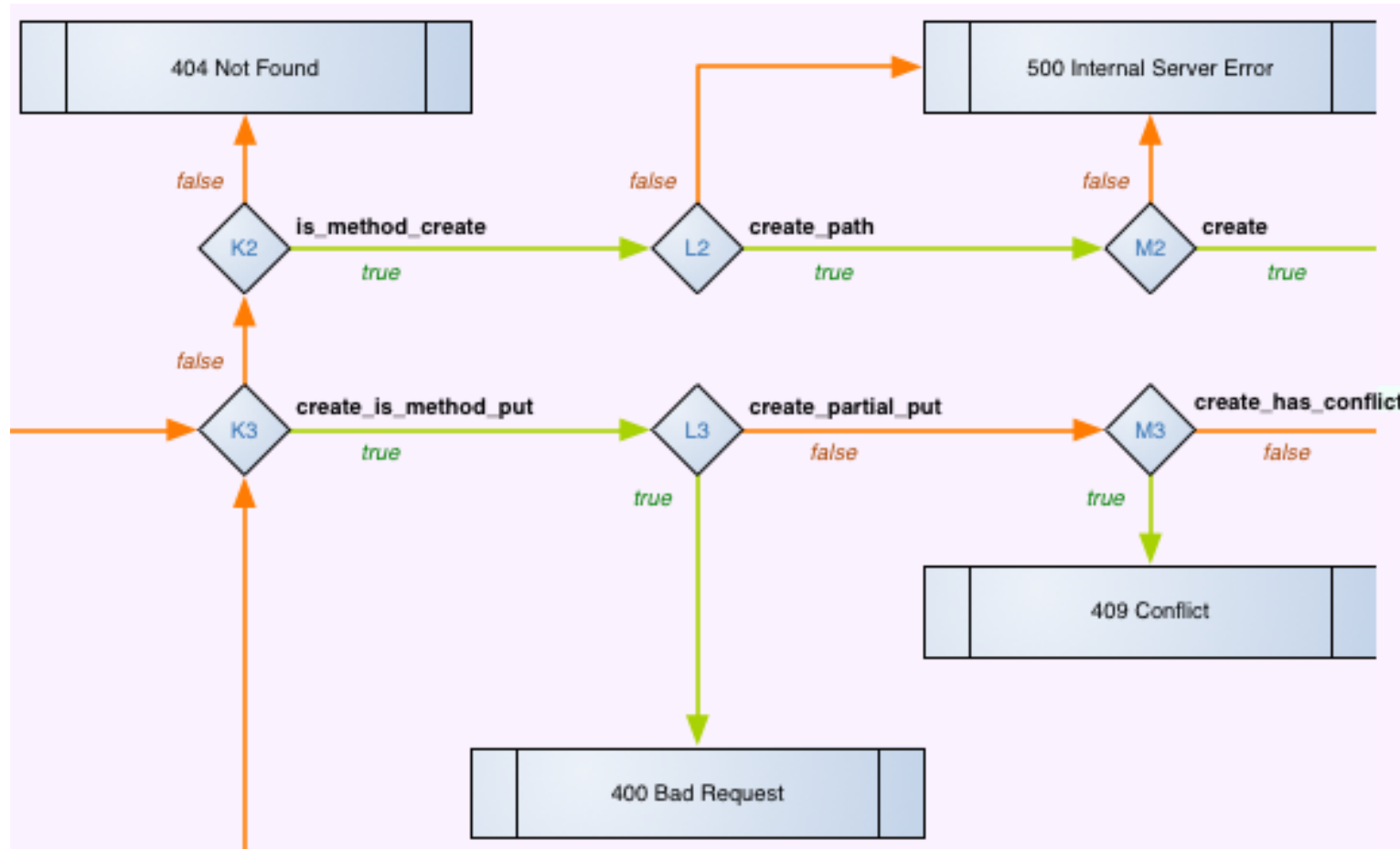
<https://github.com/for-GET/http-decision-diagram>

Decision Graph: Retrieve

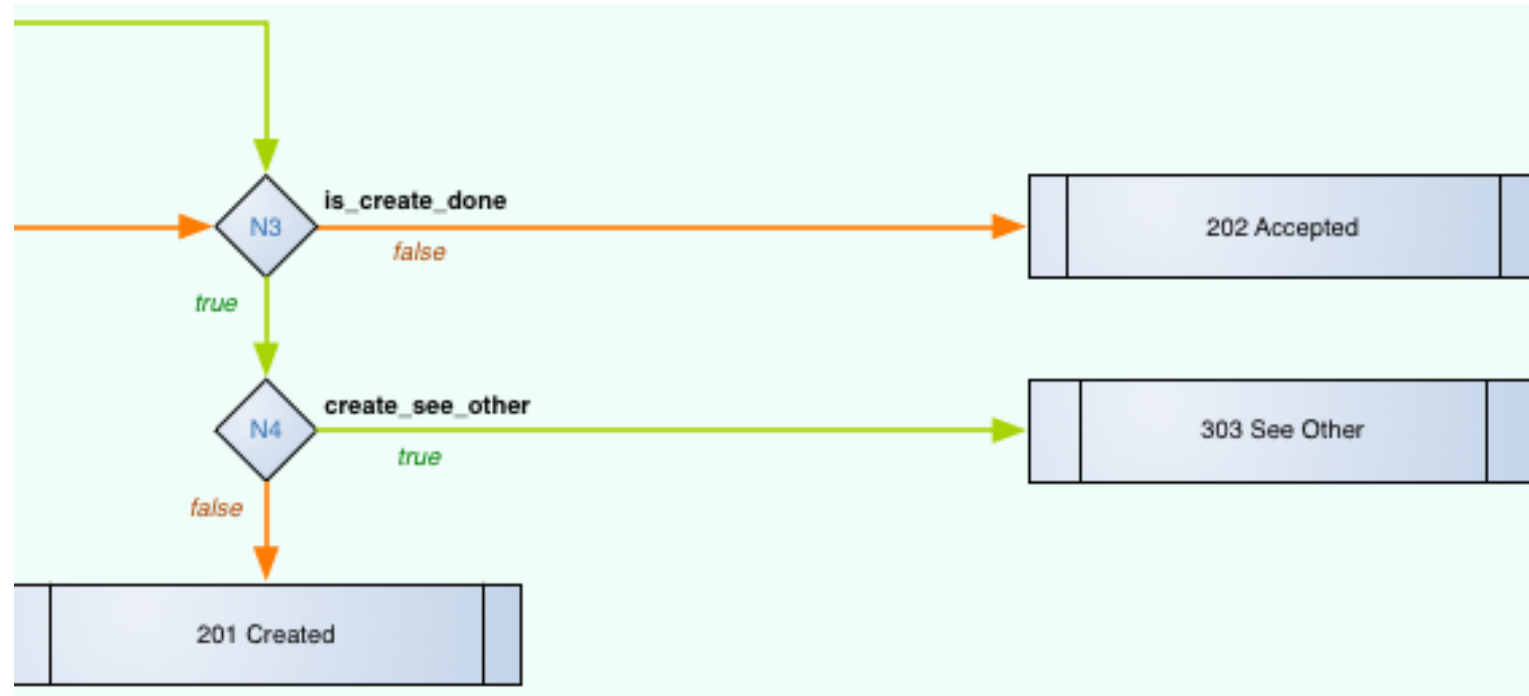


<https://github.com/for-GET/http-decision-diagram>

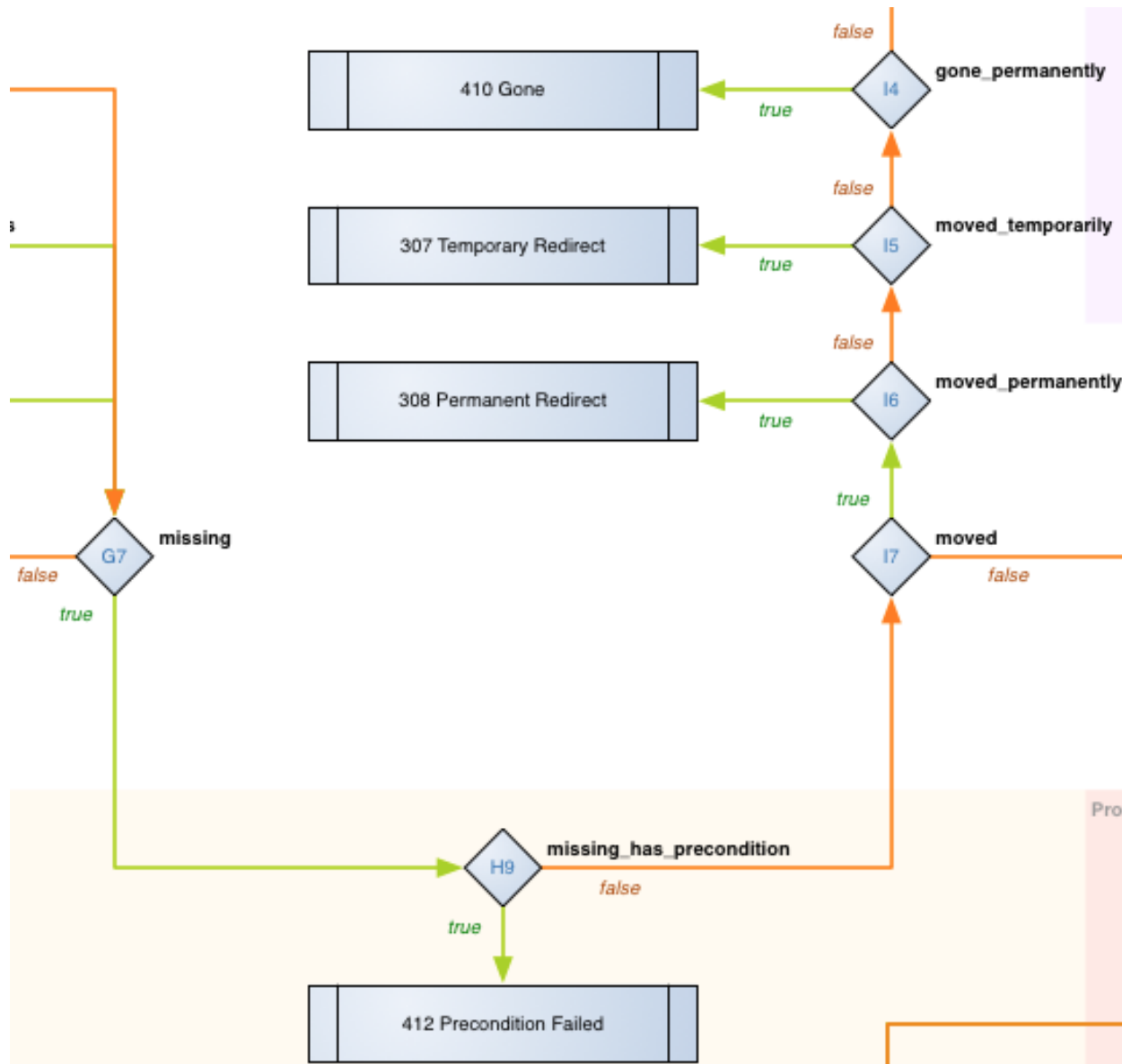
Decision Graph: Create



Decision Graph: Response

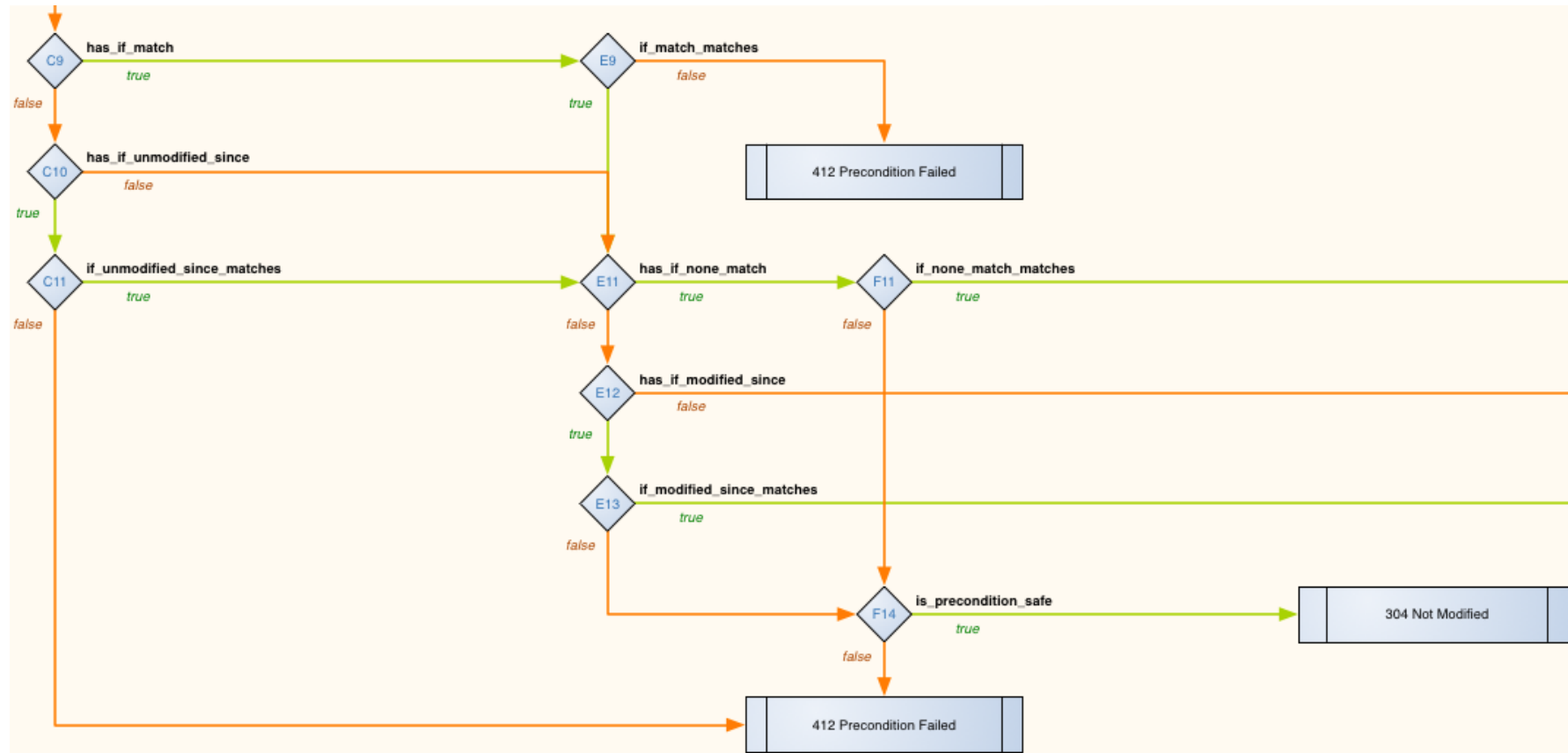


Decision Graph: Retrieve

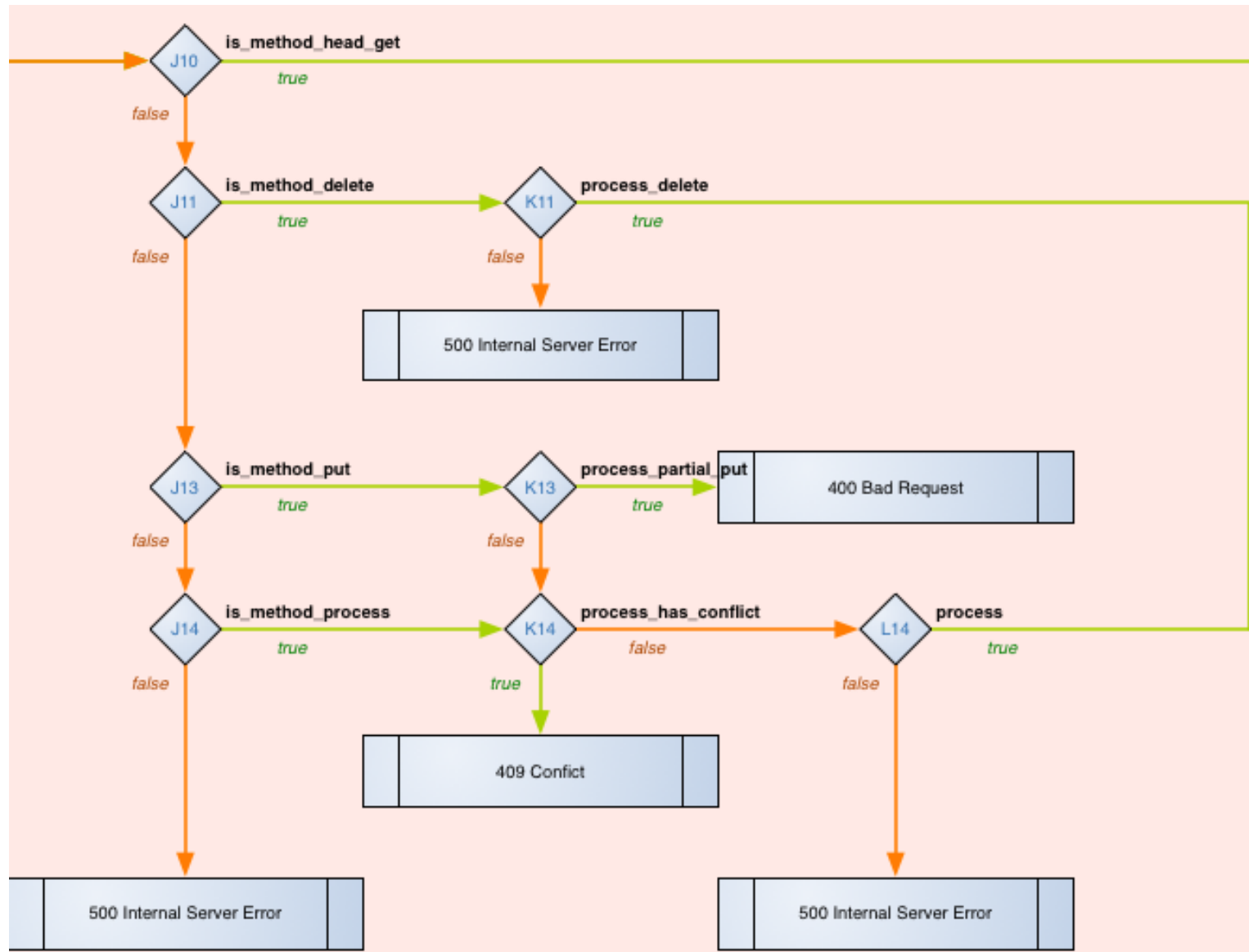


<https://github.com/for-GET/http-decision-diagram>

Decision Graph: Precondition

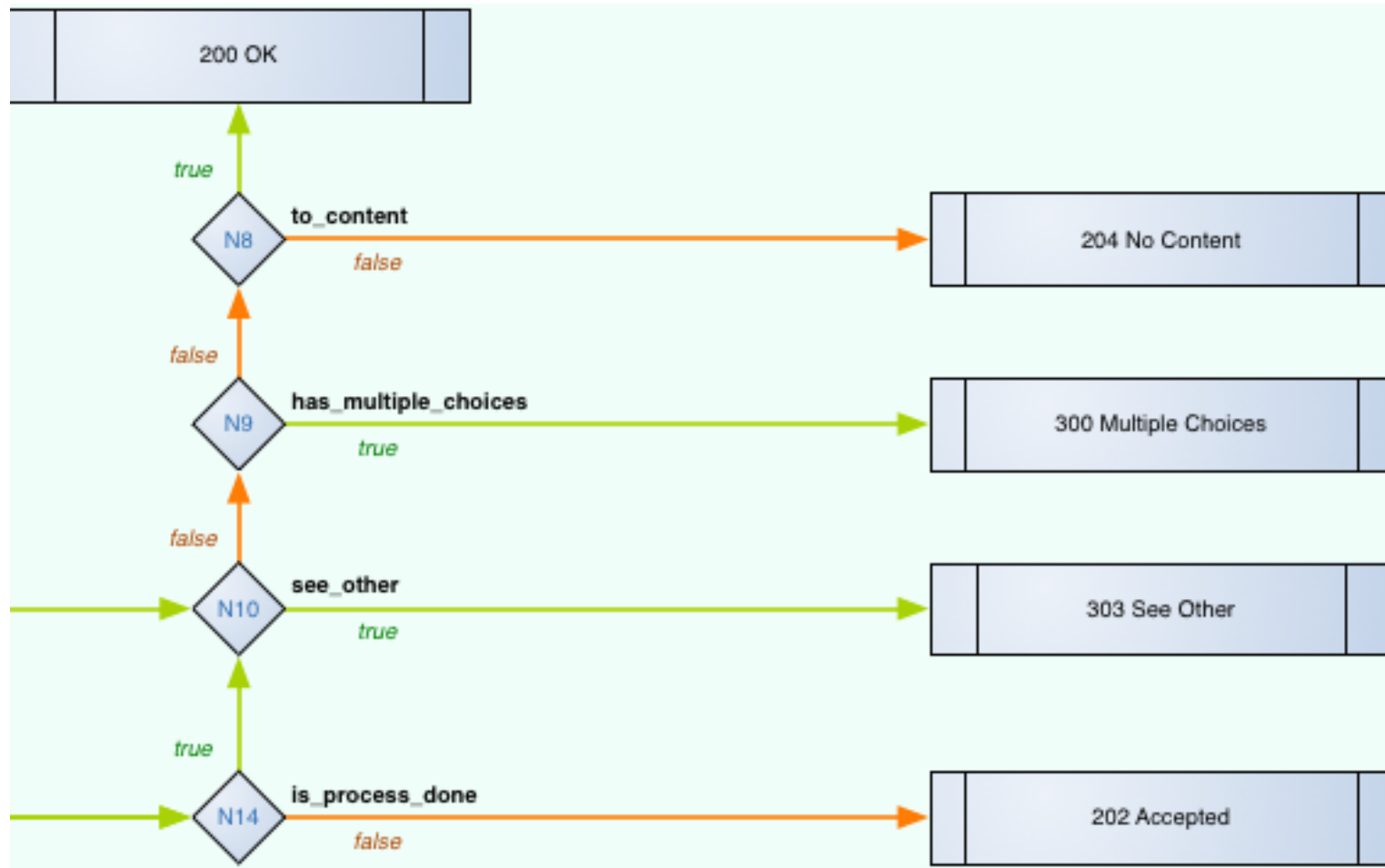


Decision Graph: Process



<https://github.com/for-GET/http-decision-diagram>

Decision Graph: Response



Liberator: Example

```
(ANY "/choice" []
  (resource
    :available-media-types ["text/html"]
    :exists? (fn [ctx]
      (if-let [choice
        (get {"1" "stone" "2" "paper" "3" "scissors"}
          (get-in ctx [:request :params "choice"]))]
        {:choice choice}))
    :handle-ok (fn [ctx]
      (format "<html>Your choice: %s"
        (get ctx :choice)))
    :handle-not-found (fn [ctx]
      (format "<html>No value for option: %s"
        (get-in ctx [:request :params "choice"] ""))))))
```

Similar Frameworks

webmachine

webmachine-ruby

Webster

Summary

RESTful

Hypermedia

HTTP decision graph

Frameworks/Libraries

Thank you! Questions?

Michael Vitz

michael.vitz@innoq.com

[@michaelvitz](#)

<https://speakerdeck.com/michaelvitz>

