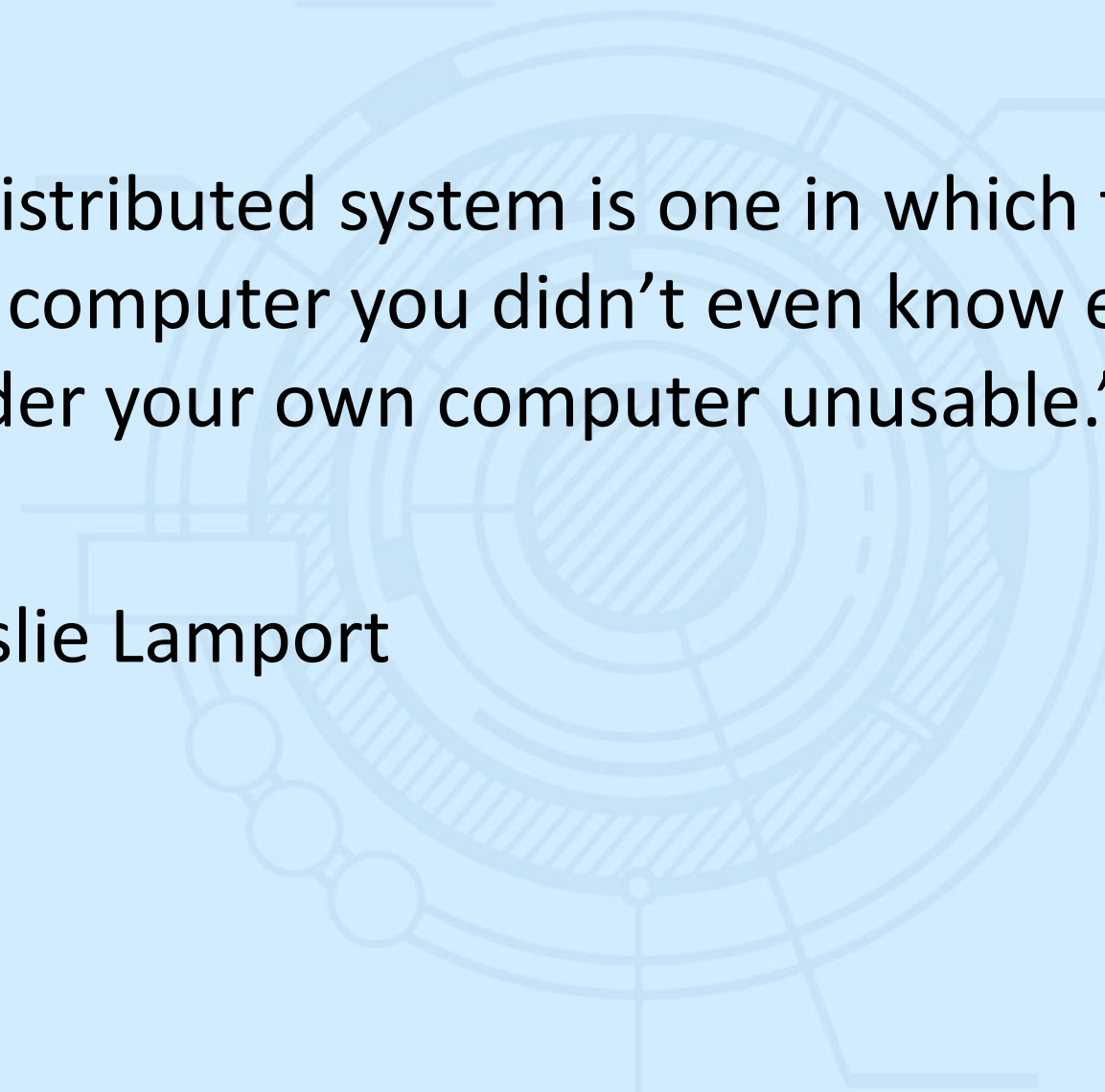




# REST Services: HTTP caching



“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.”

- Leslie Lamport

- **REST** stands for **RE**presentational **S**tate **T**ransfer
- Representation is just a very simple rendering of the object
- Has benefits when leverages existing HTTP protocol

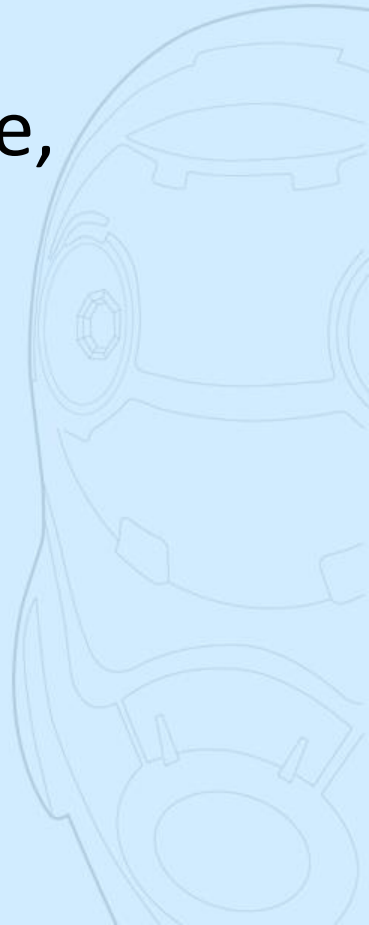
- Up-to-date
- Fresh
- Stale



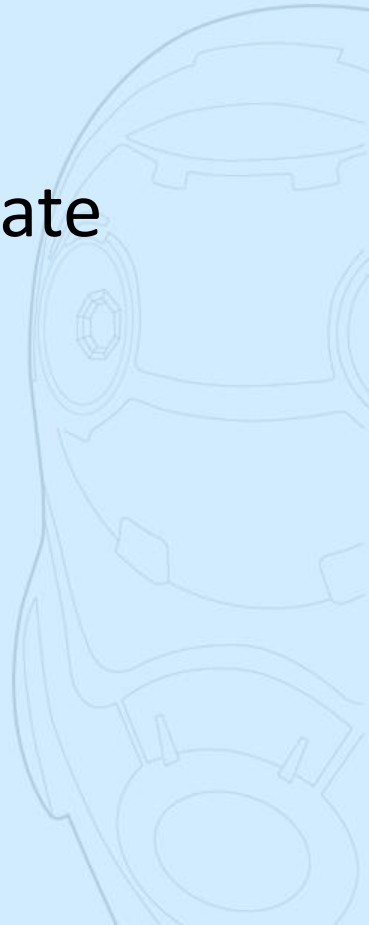
Up-to-Date state means the Representation is current **according to the origin server**

Representation is being sent back with no expiry and no caching instructions means the client is allowed to cache for however long or not cache at all

Fresh State considered as long as the Representation hasn't expired at client side, proxy or browser



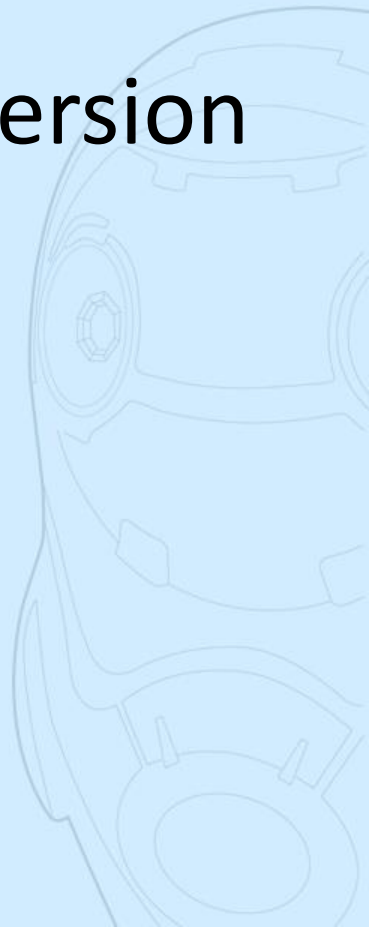
- Sending must-revalidate means the client must revalidate once the representation is Stale
- Stale means it's gone over it's expiration date



- Doesn't actually mean do not cache
- Means the client must revalidate both Fresh and Stale entries and then may use the cached copy if it's up to date



- In plain English text format
- Starting line: Method URI HTTP/version
- Headers
- Body



## Request:

*GET /wiki HTTP/1.1*

*Host: wikipedia.org*

*Accept: text/html*

*Connection: close*

## Response:

*HTTP/1.1 200 OK*

*Content-Type: text/html; charset=utf-8*

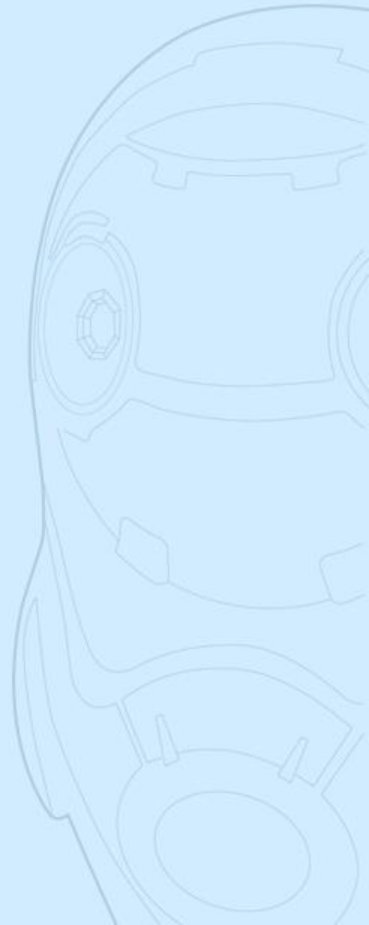
*Content-Language: en*

*Content-Length: 1234*



- Get
- Post
- Put
- Delete
- Options
- Head
- Patch
- Trace
- Link
- Unlink
- Connect

Cache-enabled methods are in Red



## Response:

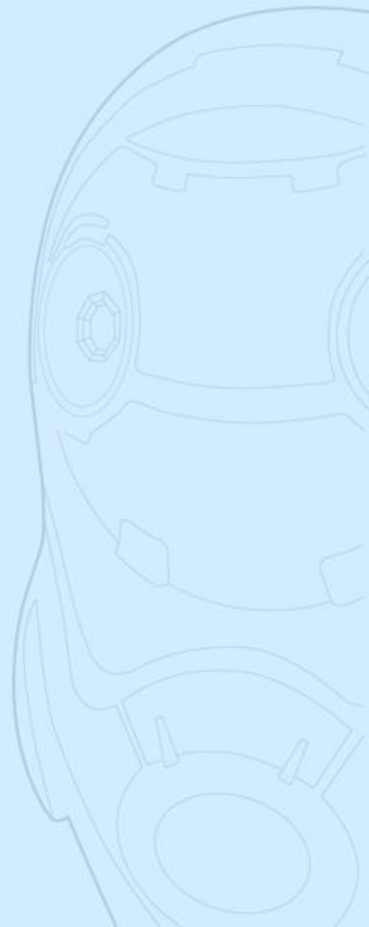
*HTTP/1.1 200 OK*

*Content-Type: text/html*

*Content-Length: 1234*

*Expires: **Fri, 27 Jan 2012 02:33:12 GMT***

- Allows only HTTP date
- Good for static resources
- Limited control



# Expires is Enough?

/Calculator/Multiply?x=2&y=2

Client

- Enough for static content
- For things that rarely change
- Or don't change at all like Calculator

- *public*
- *private*
- *no-cache*
- *no-store*
- *max-age=[seconds]*
- *s-maxage=[seconds]*
- *must-revalidate*
- *proxy-revalidate*
- *no-transform*



## Response:

*HTTP/1.1 200 OK*

*Content-Type: text/html*

*Content-Length: 1234*

*Cache-Control: max-age=3600, must-revalidate*

What if we need to **update** data?

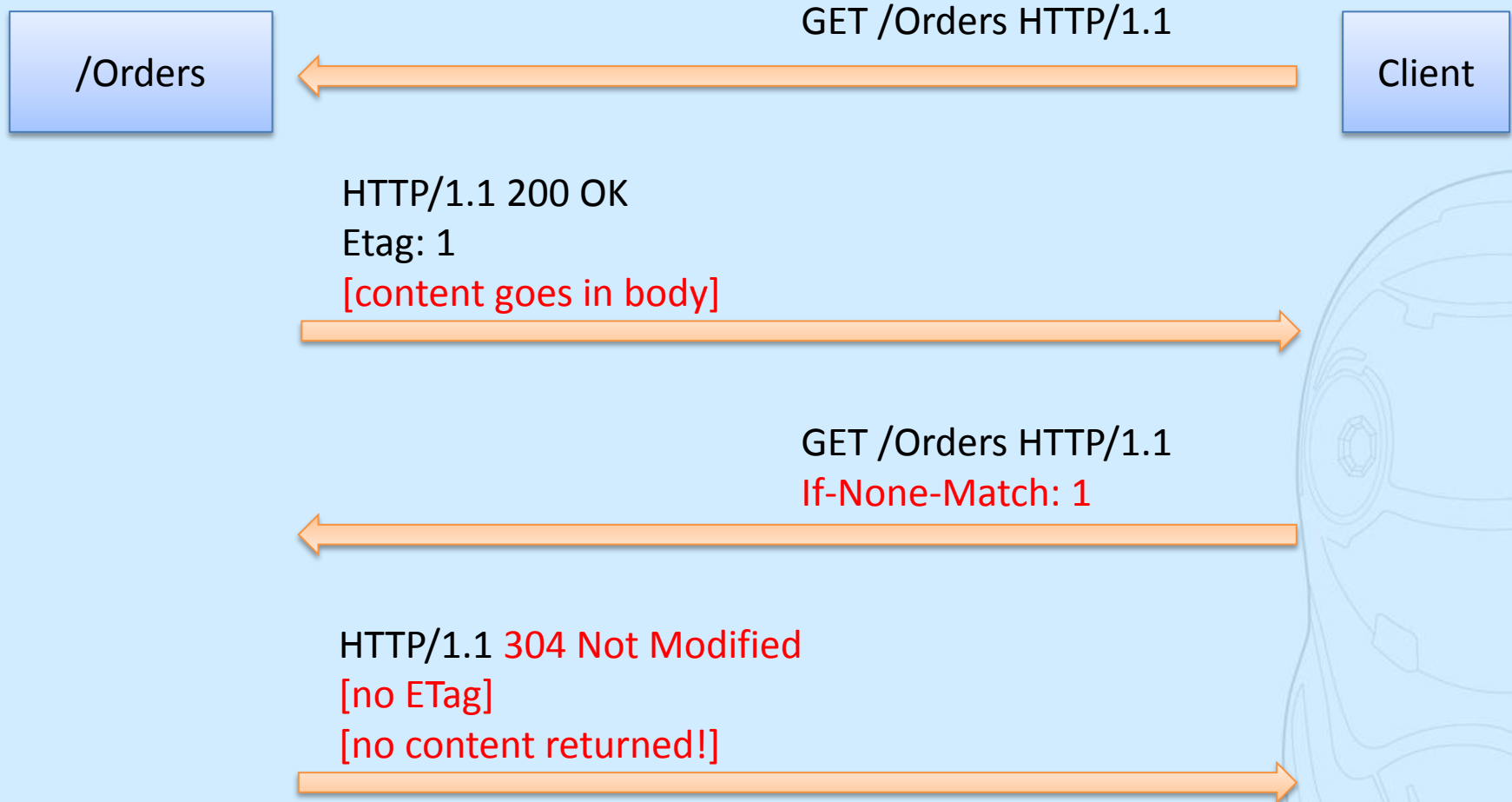




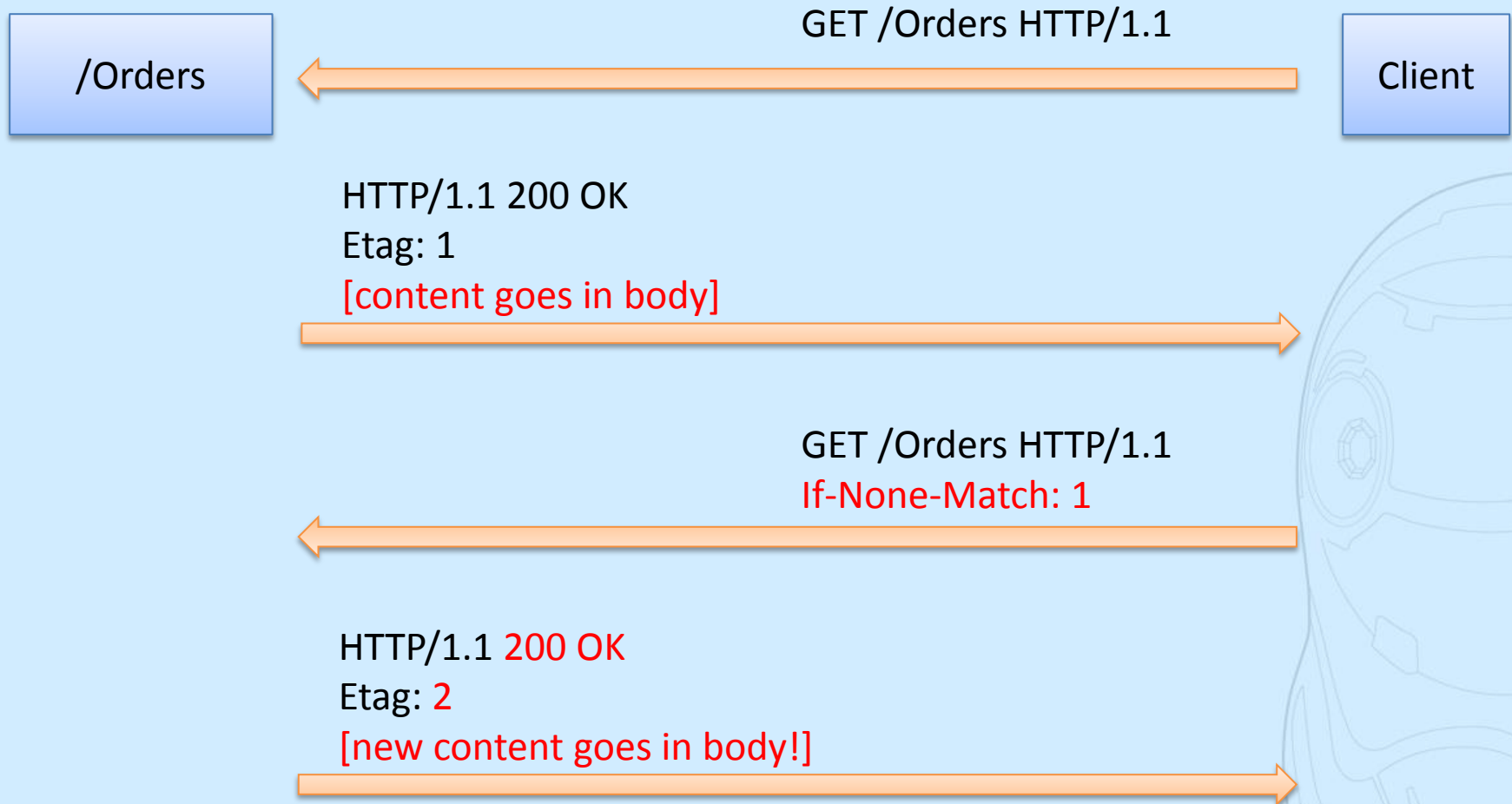
- Optimistic is when we version our data
- And do not have any transactions
- Make use of **ETag**, **If-Match** and **If-None-Match** headers



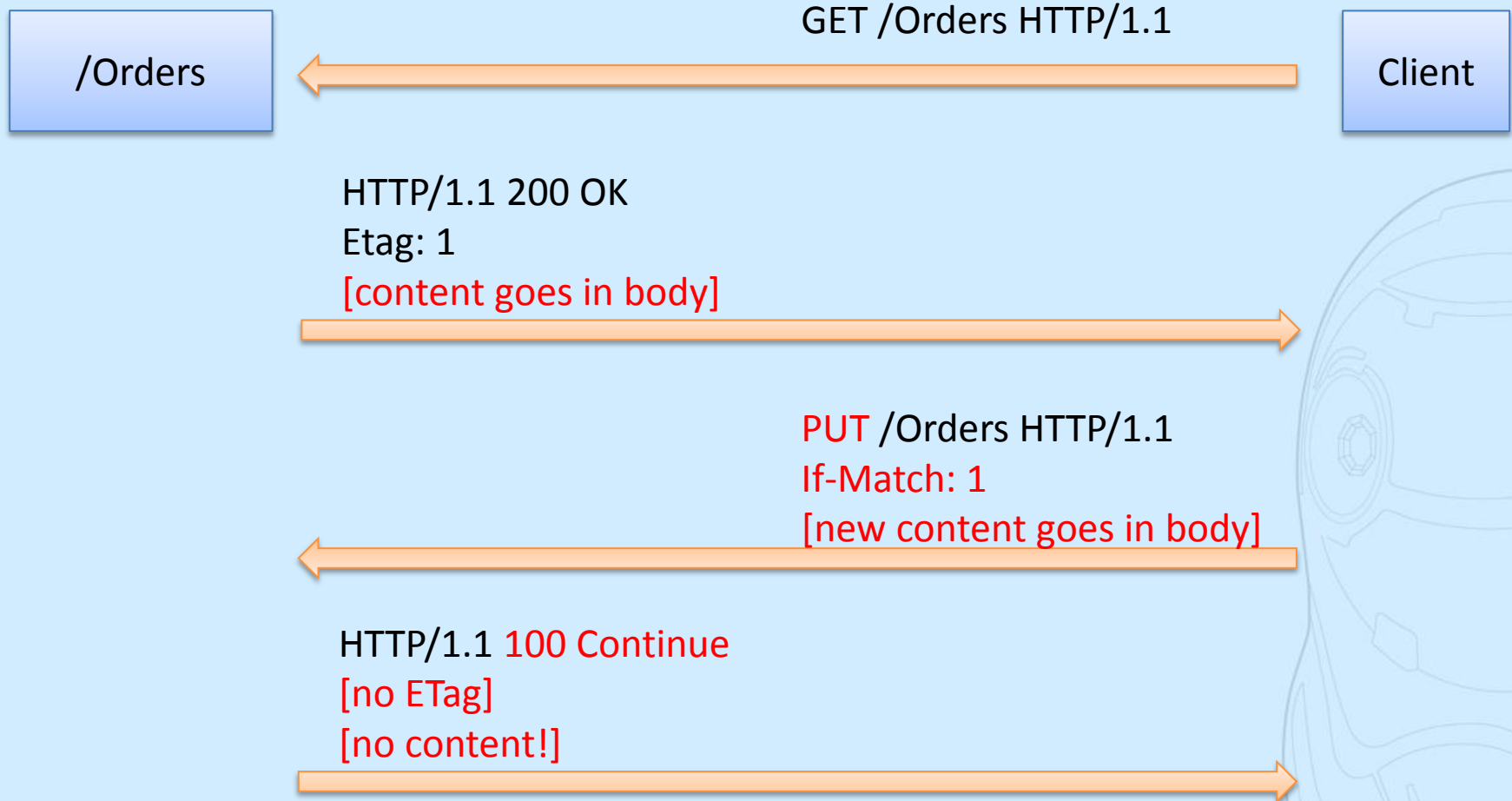
# ETag: GET Not Modified Resource



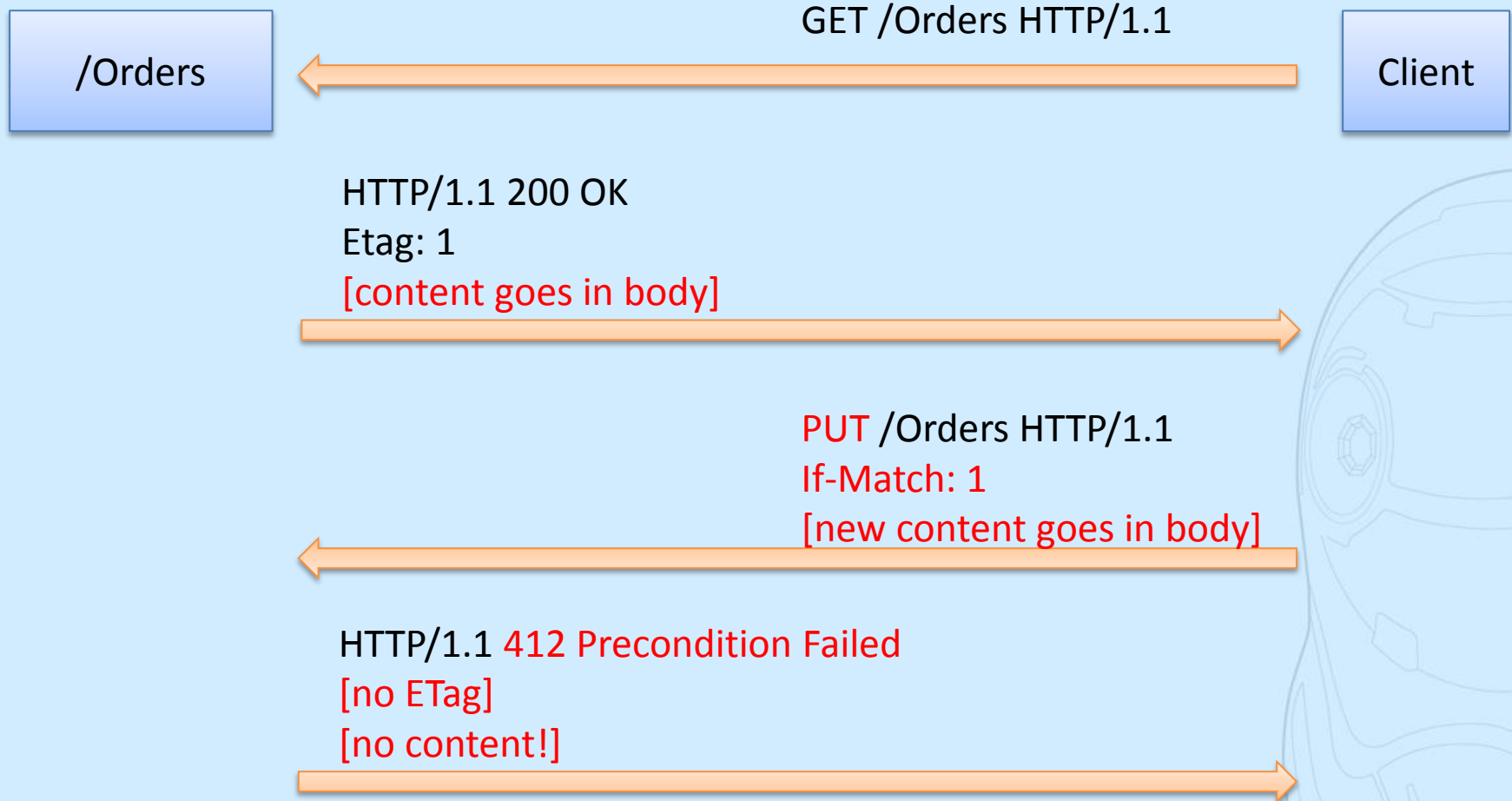
# ETag: GET modified resource



# ETag: Modify Resource (PUT)

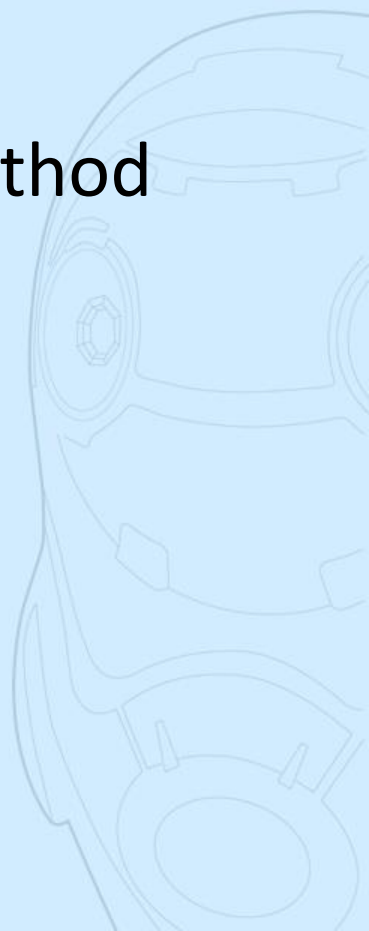


# ETag: Modify Changed Resource (PUT)

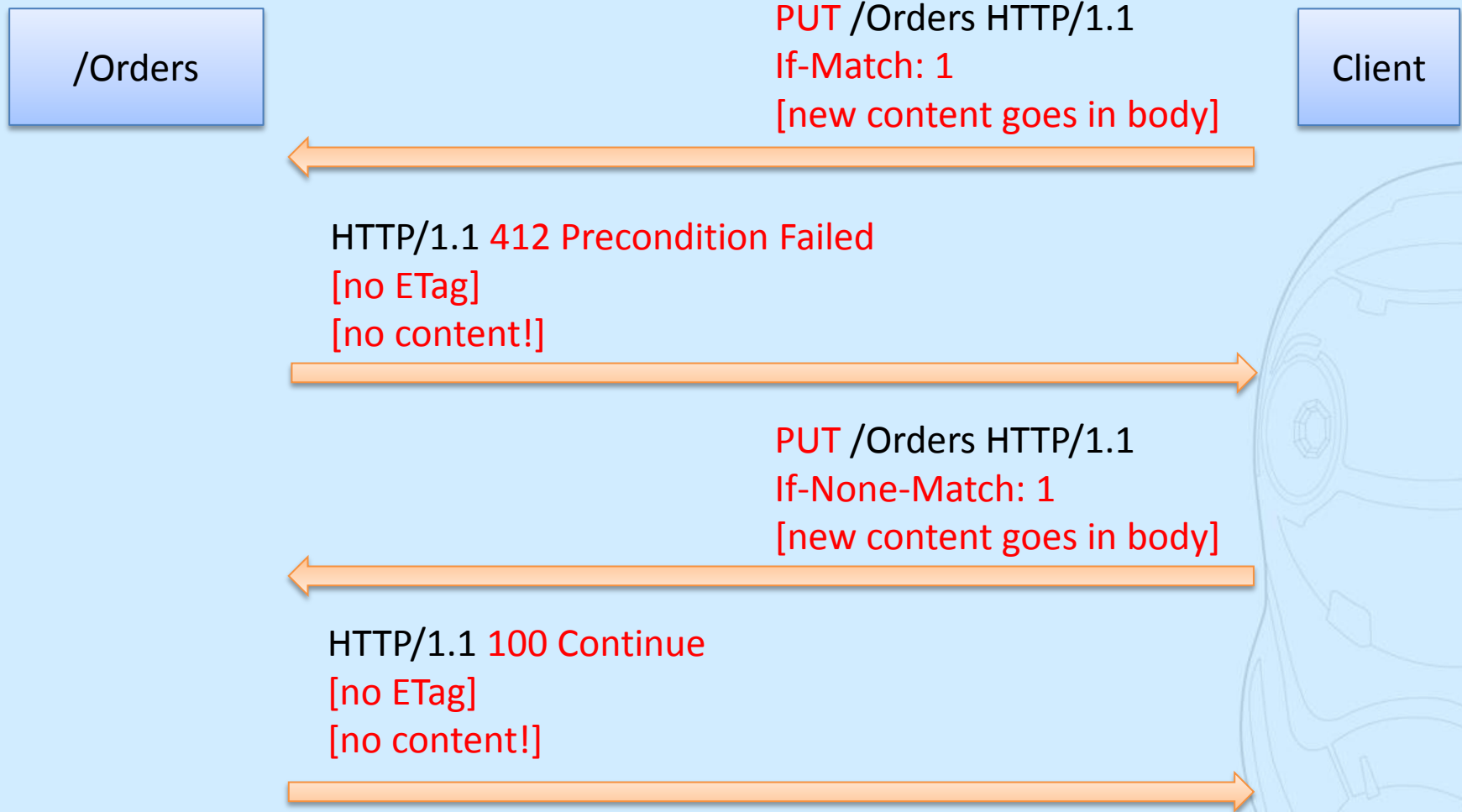


God damned HTTP!  
What we gonna do?

- **Download** the latest version using GET method
- We will know the new Etag then
- **Override changes!**
- Yeah! It's possible!

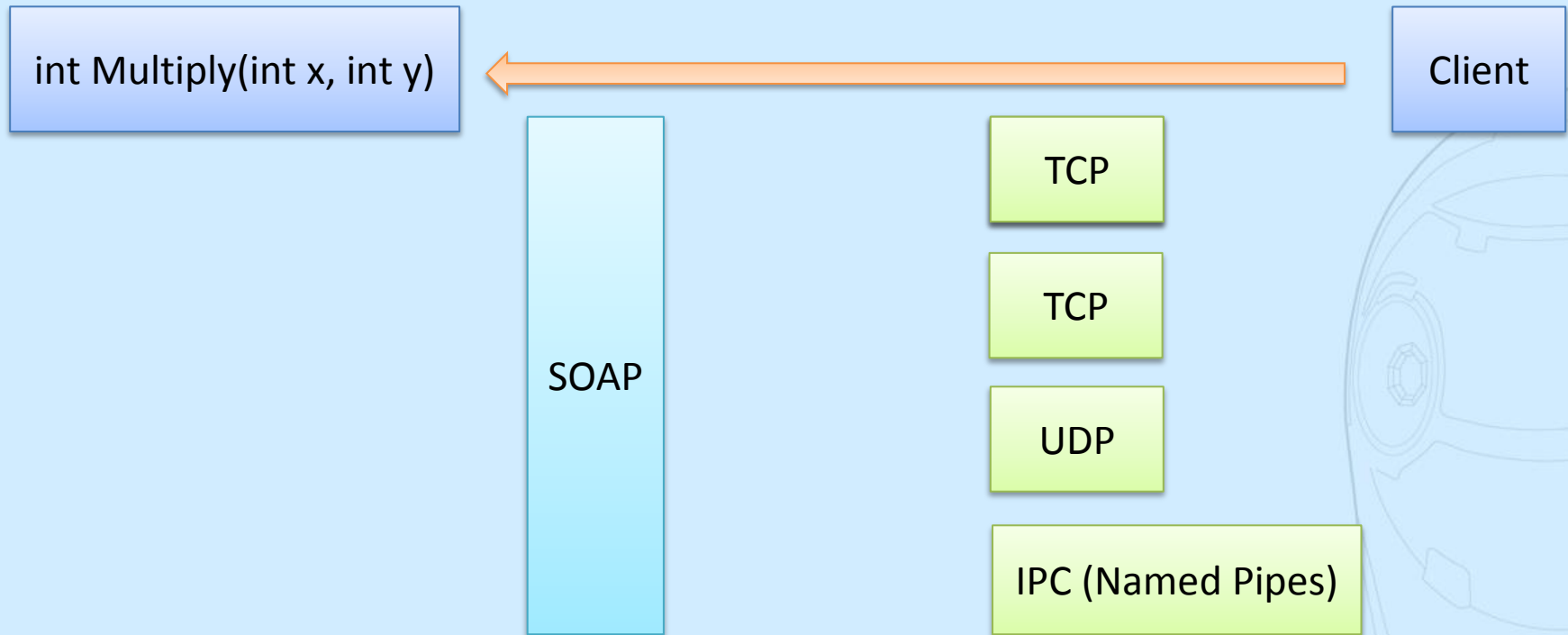


# Etag: Override Changes When PUT



- ETag and If-\* headers came with HTTP 1.1
- In HTTP 1.0 there are another set of headers:  
Last-Modified: Sat, 29 Oct 2011 19:43:31 GMT  
If-Modified-Since: Sat, 29 Oct 2011 19:43:31 GMT





POST /search HTTP/1.1

Host: www.domain.com

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

term=something

HTTP/1.1 303 See Other

Location: /search/something/1

