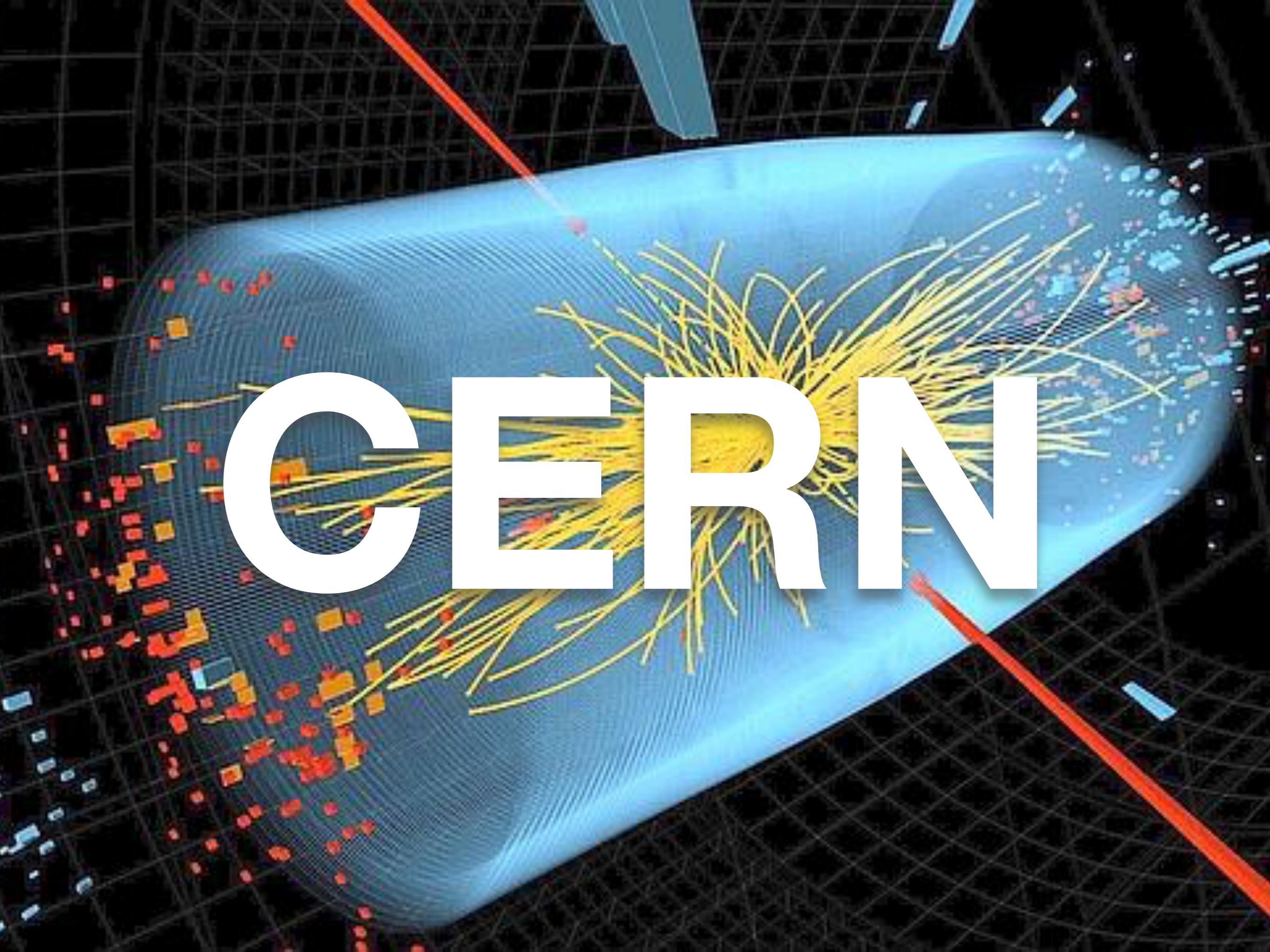


Talking Machines

A story of avoiding chaos



CERN

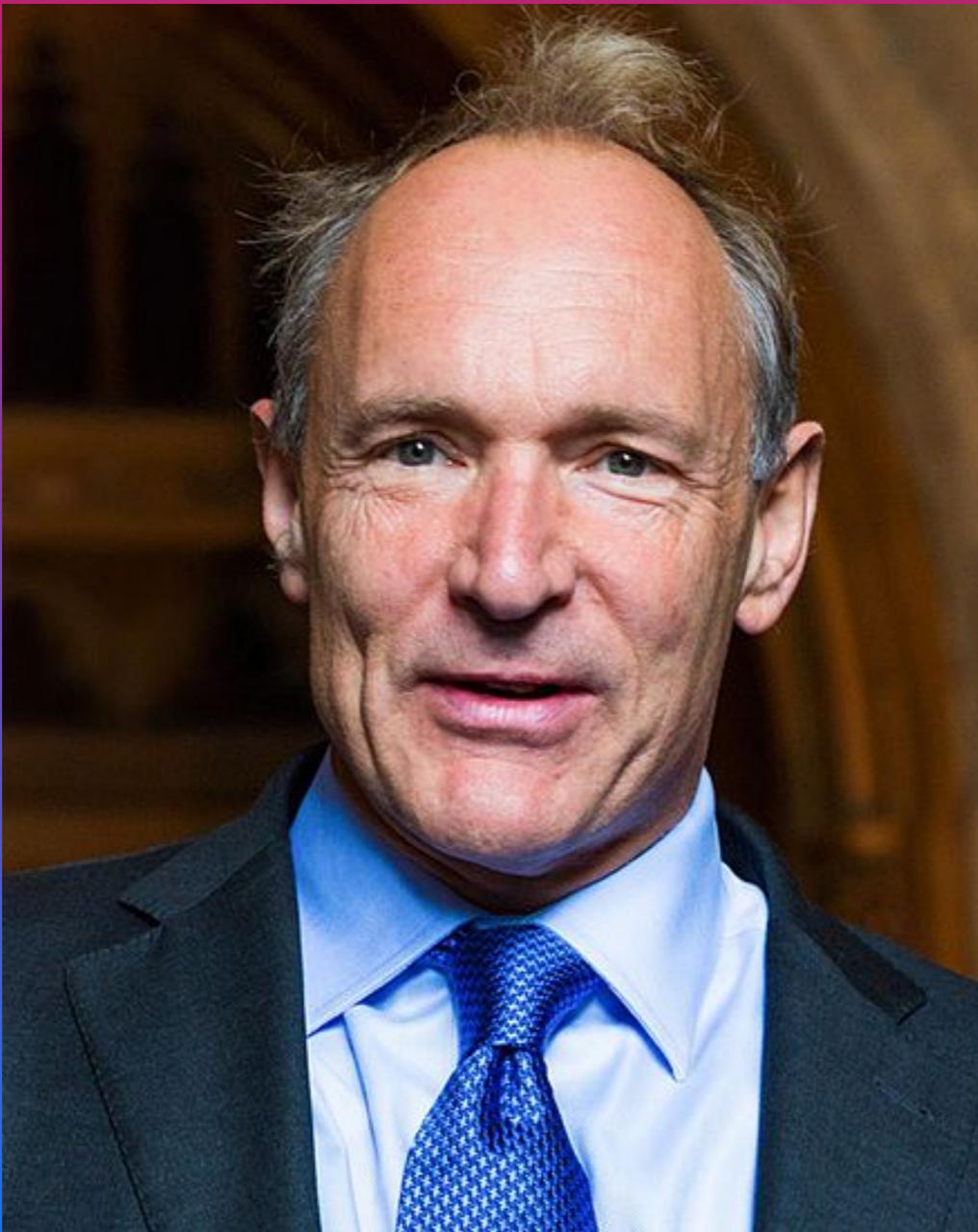
CERN

- The European Organization for Nuclear Research is a European research organization that operates the largest particle physics laboratory in the world.

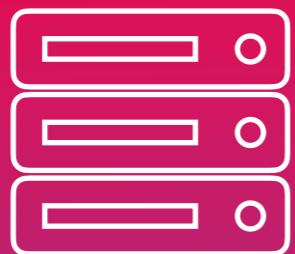
Physic Nerds



Computer Nerd



Tim Berners-Lee

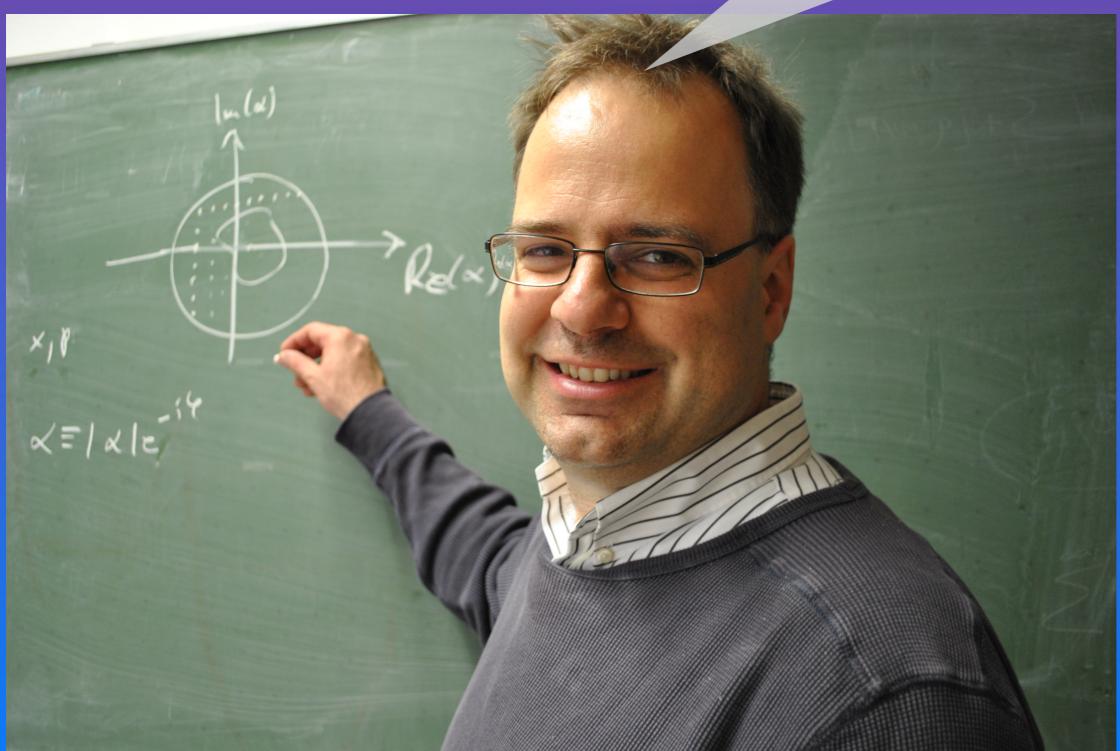


Server



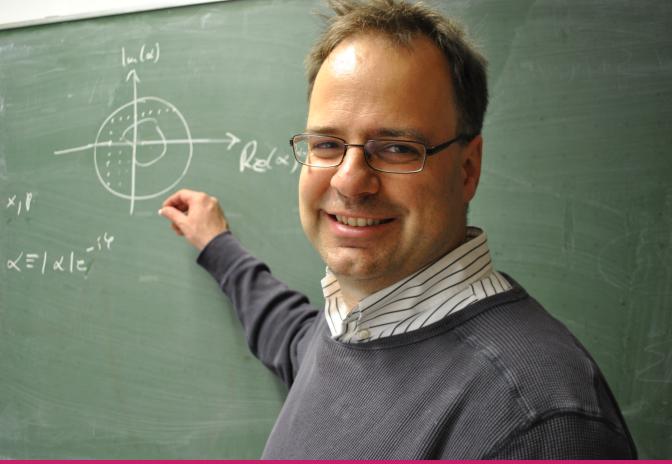
Client

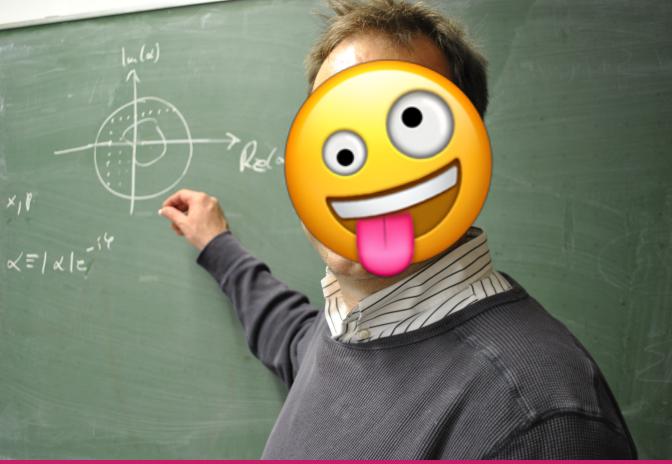
We're making a bit of a mess on our server-client communications



K







Gimme this site ←

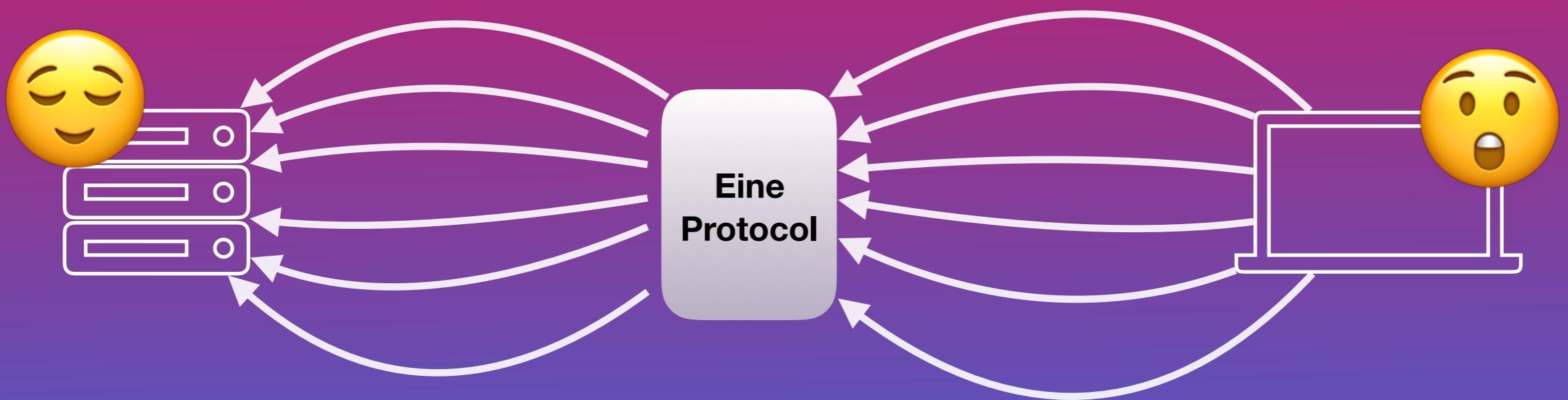
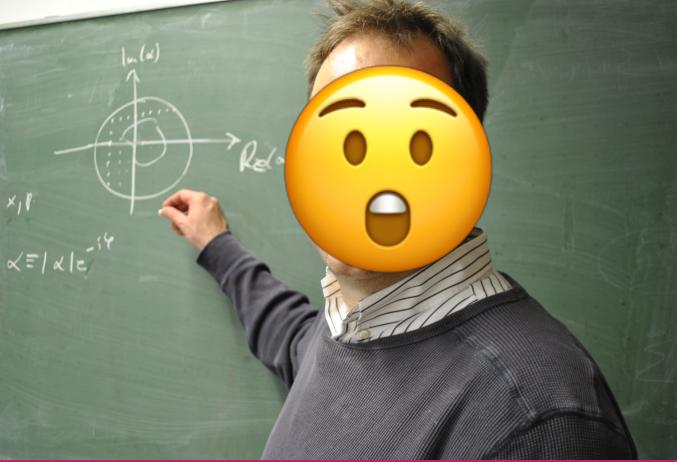
Lend me this website ←

Dame este sitio ←

Muéstrame esta página ←

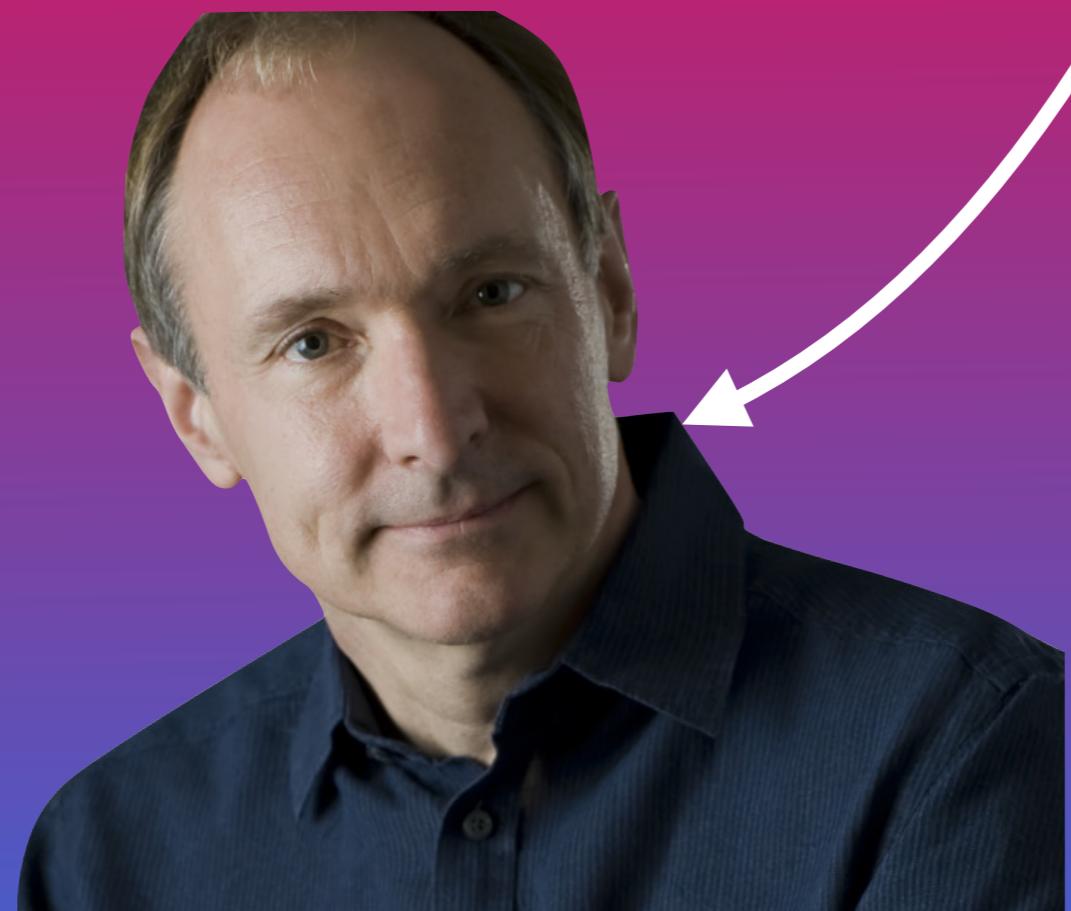
Gib mir diese Seite ←

donne moi cette page ←



HTTP & HTML

A man of protocols



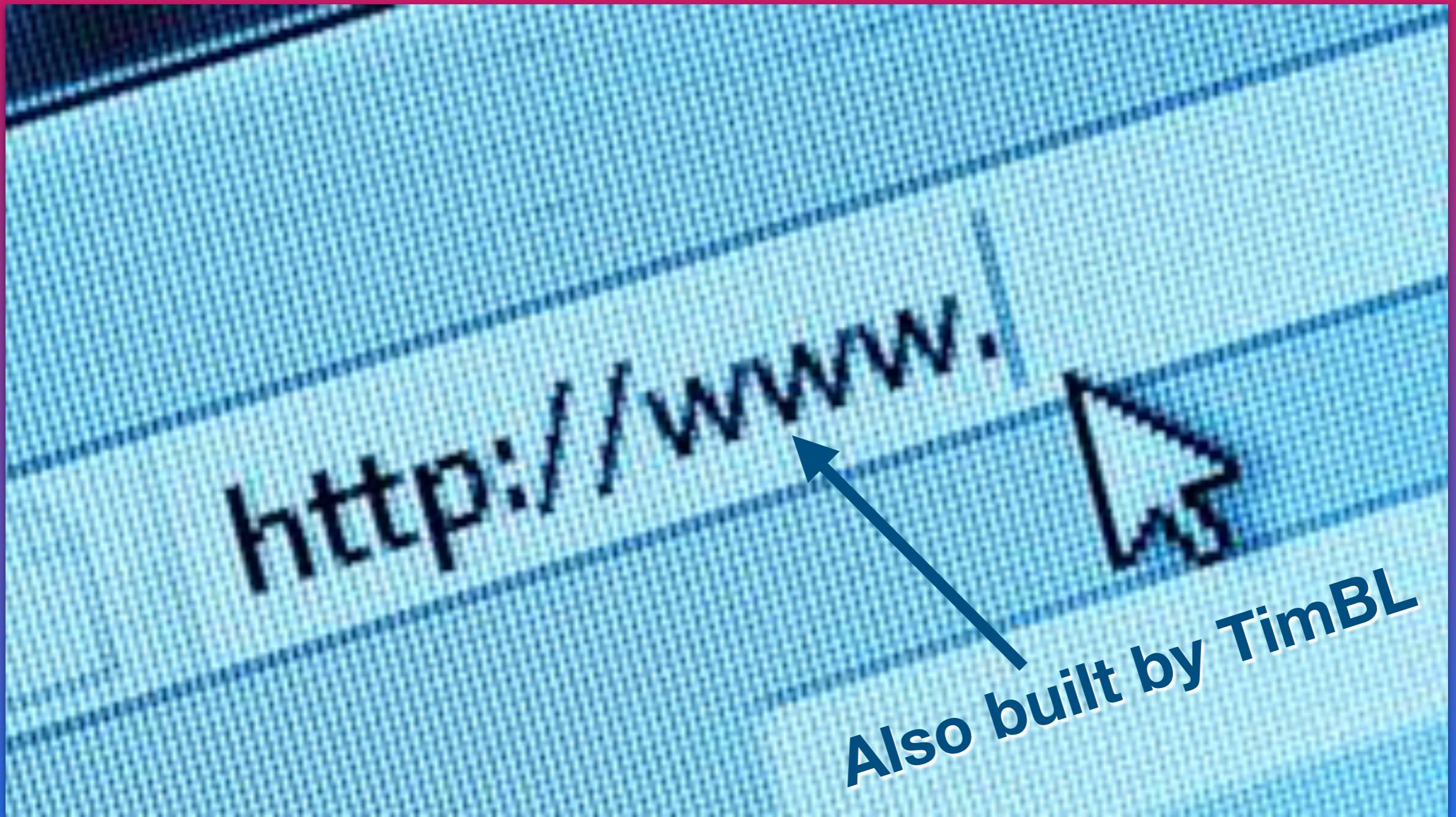
“Protocol: The **accepted or established code of procedure or behavior in any group, organization, or situation.”**

–Oxford Dictionary

HTML = Standard for responding

```
<!DOCTYPE html>
<html class="osx x86 js loaded" dir="ltr" data-latest-
firefox="46.0" data-esr-versions="38.8.0 45.1.0" lang="en"> ev
  ▶ <head></head>
  ▼ <body id="home" class="html-ltr lang-en-US"> ev
    <div id="strings" data-global-close="Close" data-global-
      next="Next" data-global-previous="Previous" data-global-update-
      firefox="Update your Firefox" data-read-more="Read more"></div>
    ▼ <div id="outer-wrapper">
      ▼ <div id="wrapper">
        ▶ <header id="masthead"></header>
        ▼ <main role="main">
          ▼ <header class="main-header">
            ▼ <div class="container">
              ▶ <h1></h1>
              | ::after
            </div>
```

HTTP = Standard for asking



“Thou shalt request with a **GET** method on
a given server’s **path** and expect either an
error or an **HTML** file.”

-HTTP v.0.9

IMPLICIT: GET METHOD

`http://mysite.com/`

Protocol

Server

Path

`http://mysite.com/users/me`

“(...) expect either an **error** (...)"



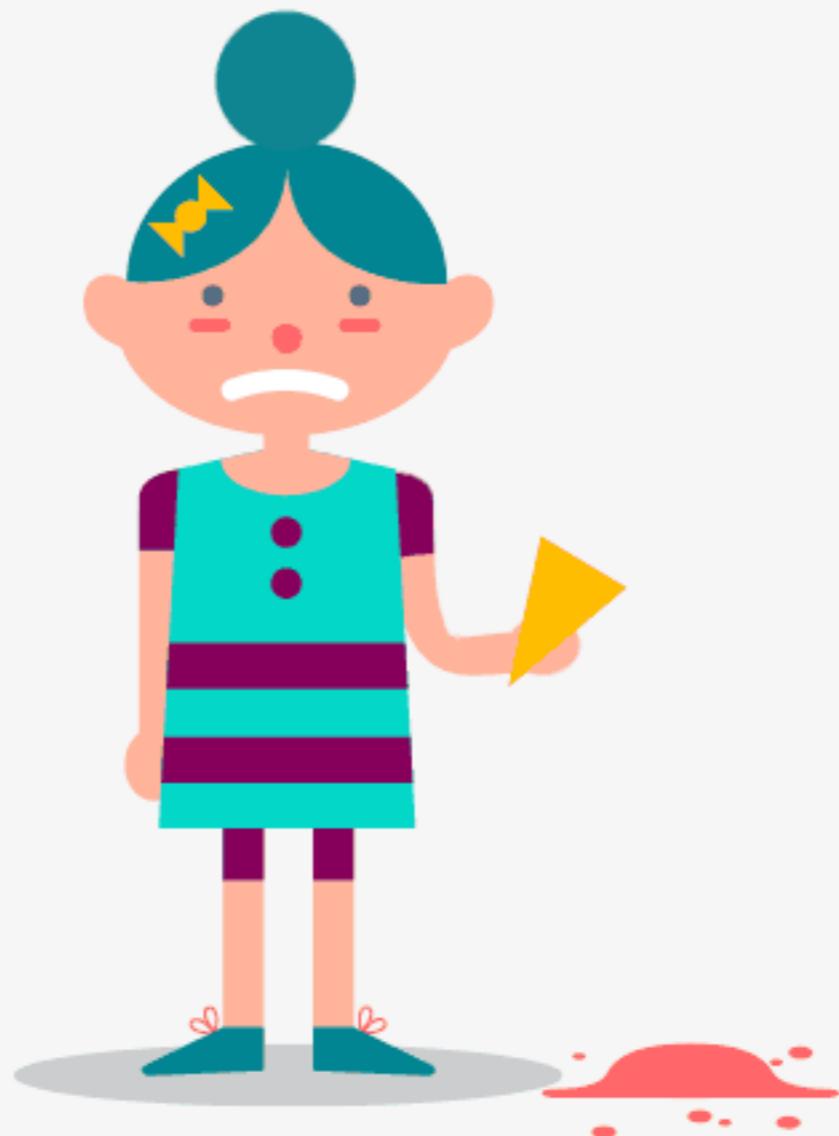
Oops!

We can't seem to find the page you're looking for.

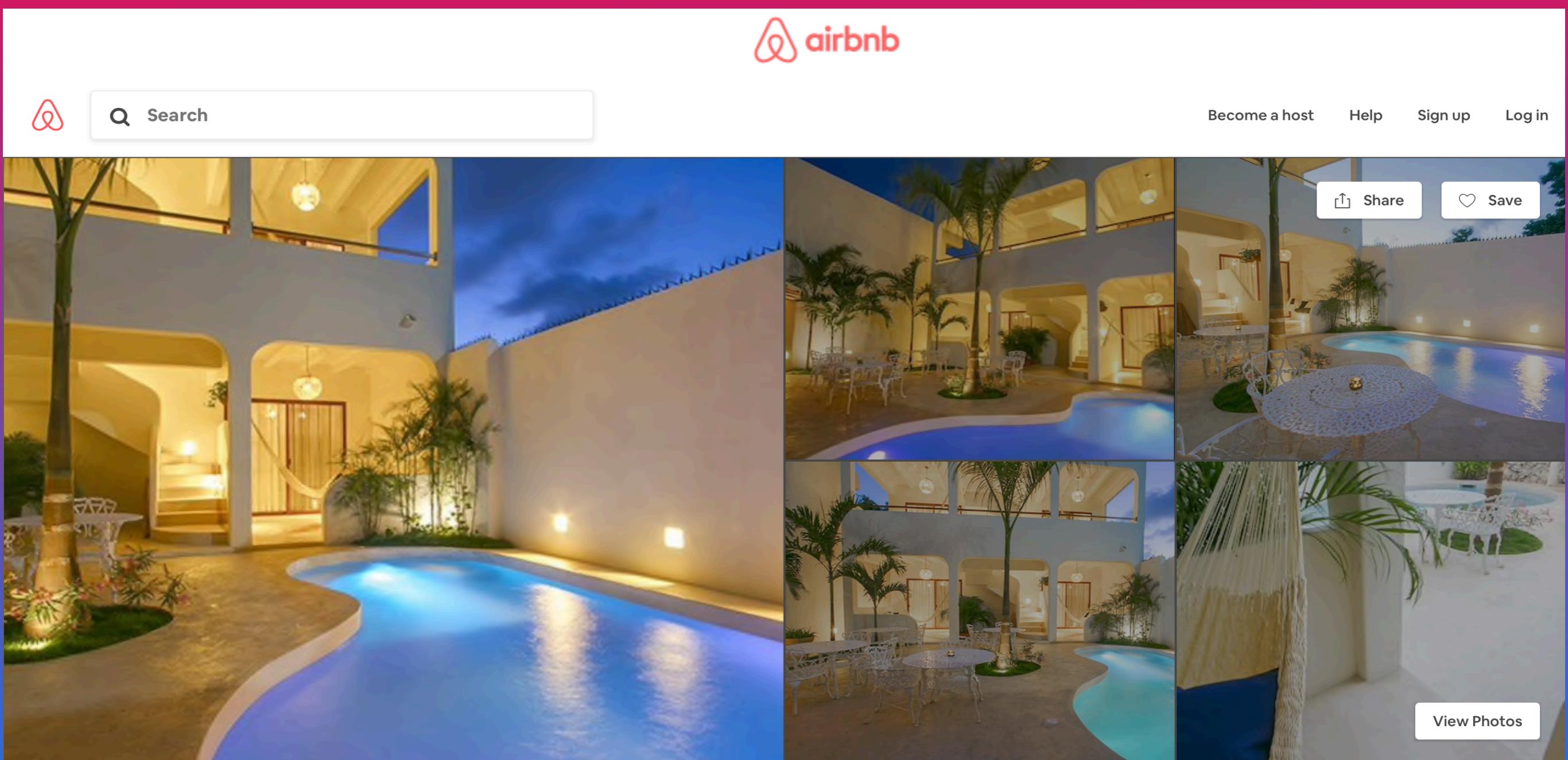
Error code: 404

Here are some helpful links instead:

[Home](#)
[Search](#)
[Help](#)
[Traveling on Airbnb](#)
[Hosting on Airbnb](#)
[Trust & Safety](#)
[Sitemap](#)



“(...) or an **HTML** file.”



The screenshot shows a Airbnb listing for an entire apartment named "Villas del Mar Azul department". The main image displays a large, modern house with a curved swimming pool illuminated at night. To the right, there is a grid of smaller images showing different angles of the property, including the interior and exterior during the day. A "View Photos" button is located in the bottom right corner of the photo grid.

ENTIRE APARTMENT

Villas del Mar Azul department

Playa del Carmen

2 guests | 1 bedroom | 1 bed | 1 bath

 Stefano

\$1,018 per night
★★★★★ 243

Dates

Check in → Check out

Newer, Better.

More Methods

HTTP METHOD	CRUD	ENTIRE COLLECTION (E.G. /USERS)	SPECIFIC ITEM (E.G. /USERS/123)
POST	Create	201 (Created), 'Location' header with link to /users/	Avoid using POST on single resource
GET	Read	200 (OK), list of users. Use pagination, sorting	200 (OK), single user. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	404 (Not Found), unless you want to update every	200 (OK) or 204 (No Content). Use 404 (Not Found), if ID not
PATCH	Partial Update/ Modify	404 (Not Found), unless you want to modify the collection	200 (OK) or 204 (No Content). Use 404 (Not Found), if ID not
DELETE	Delete	404 (Not Found), unless you want to delete the whole	200 (OK). 404 (Not Found), if ID not found or invalid.

Codes

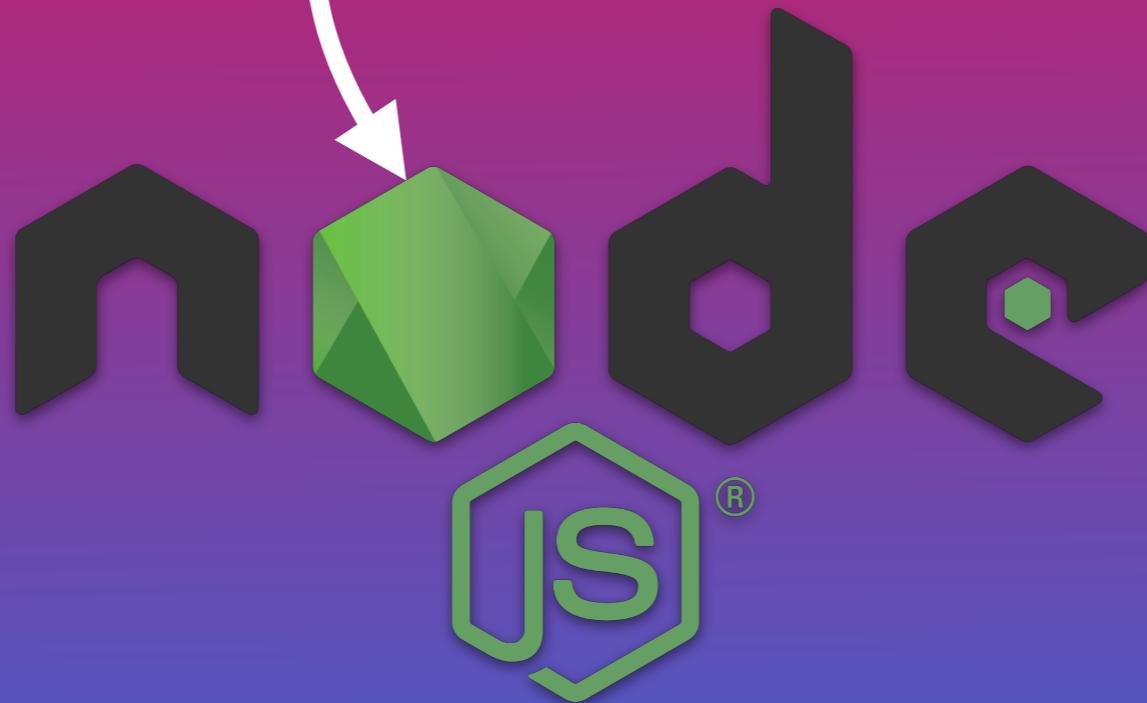
HTTP CODE	STATUS
200	Ok
201	Created
204	No Content
304	Not Modified
400	Bad Request
401	Not Authorized
403	Forbidden
404	Not Found
409	Conflict
500	Internal Server Error

Architecture

- **URI:** Request path, may include query variables.
- **Body:** In request, used to pass object data, in response is the data returned. XML or JSON.
- **Header:** Included in request and response, used for bidirectional server-client communication. XML or JSON.

How to use it?

Has HTTP built in

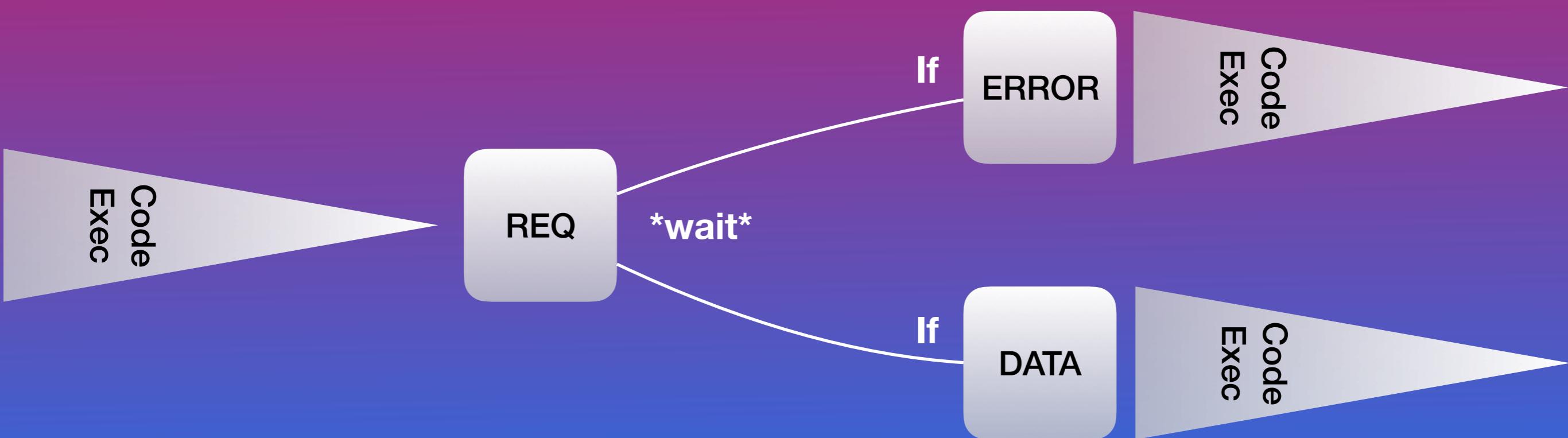


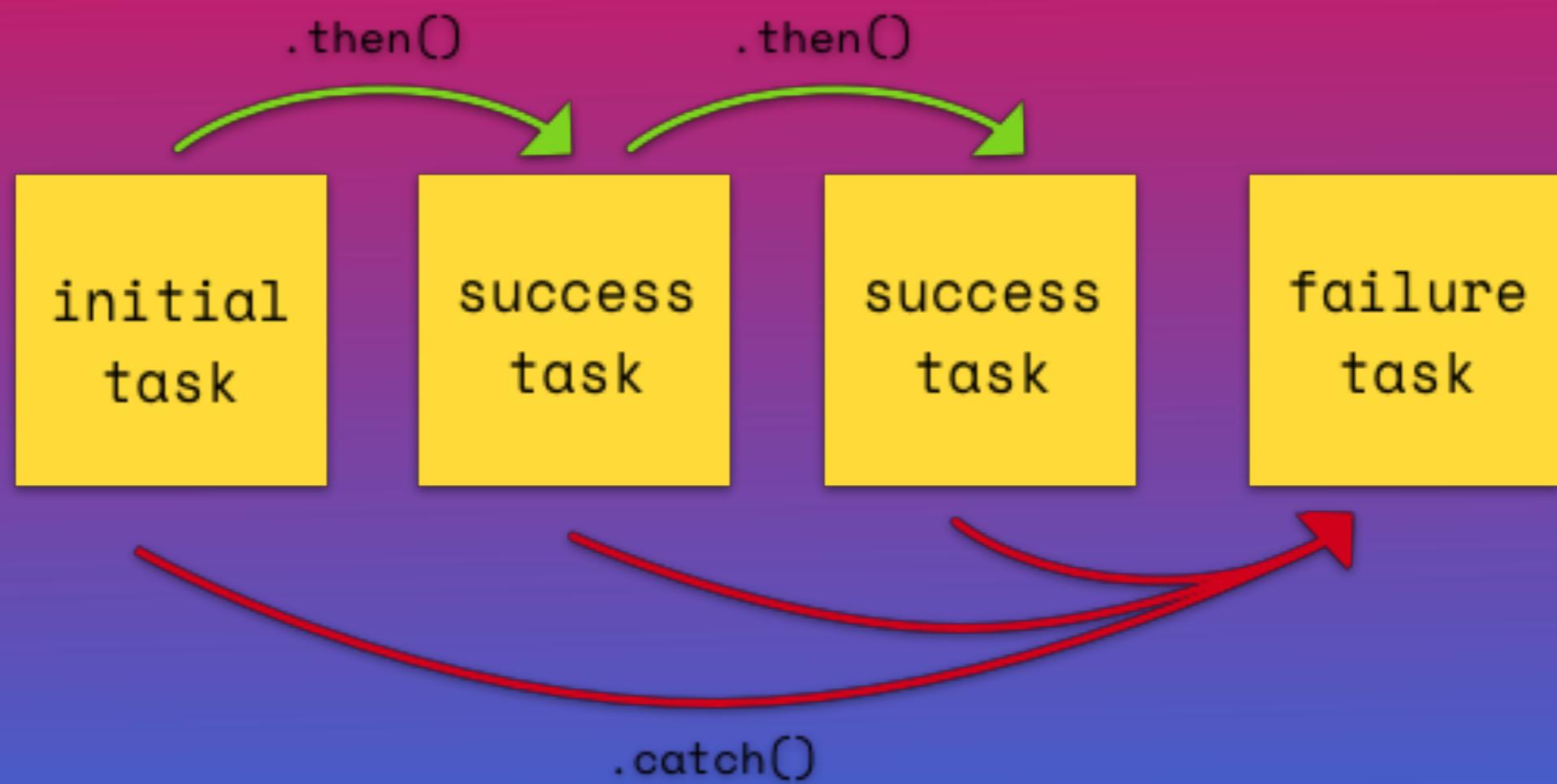
```
$ yarn add axios
```

Benefits

- Elegant
- Easy to use
- Promise based

JS Promises in a nutshell





```
import axios from 'axios';
const myUri = 'https://mysite.com/user/666?name=hello&lastName=world';

axios.get(myUri)
  .then(r => console.log('RES', r))
  .catch(e => console.log('ERR', e));

const getDataAsync = async (uri) => {
  let data, err;
  try {
    data = await axios.get(uri);
  } catch (e) {
    console.log('ERR', e);
    throw new Error(e);
  } finally {
    return data;
  };
};

getDataAsync(myUri);
```

Activity!

Let us get our hands dirty

bit.ly/http-playground-hmc

Takeaways_mod6

- The internet is built atop standards
- HTTP is very useful for server-client and server-server communication
- CRUD operations = Create, Read, Update, Delete