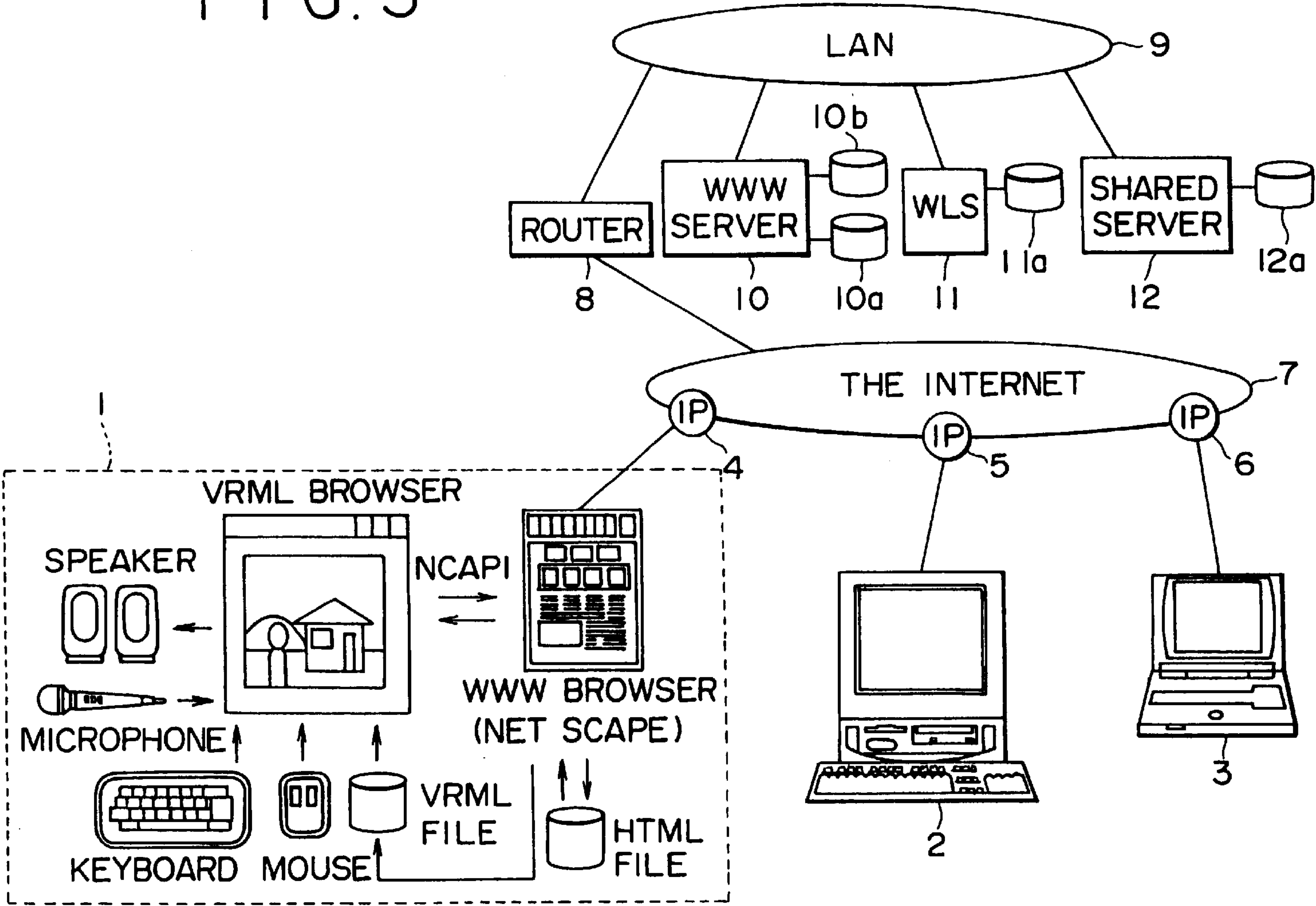


FIG. 3



COURSE OUTLINE

Topics

Basics

- What is the internet?
- What is the web?
- What is a website?

- DNS/Urls
- Localhost
- Folder structure
- Protocols
- Networks

Web related topics

- Accessibility
- Privacy
- Sustainability
- Darknet, TOR
- AI & Tools (partially)
- VPN security

Tools

- Github + Collaboration
- CI/CD Pipelines (?)
- Visual studio code & Extensions
- AI Agents

HTML

- HTML Basics

- Tags and attributes

CSS

- CSS Basics
- Responsive design/Breakpoints

- CSS specificity

JavaScript

- JavaScript (Programming) Basics
- Data fetching/API

- Object, functions and parameters, variables,
- Specific JavaScript tools?

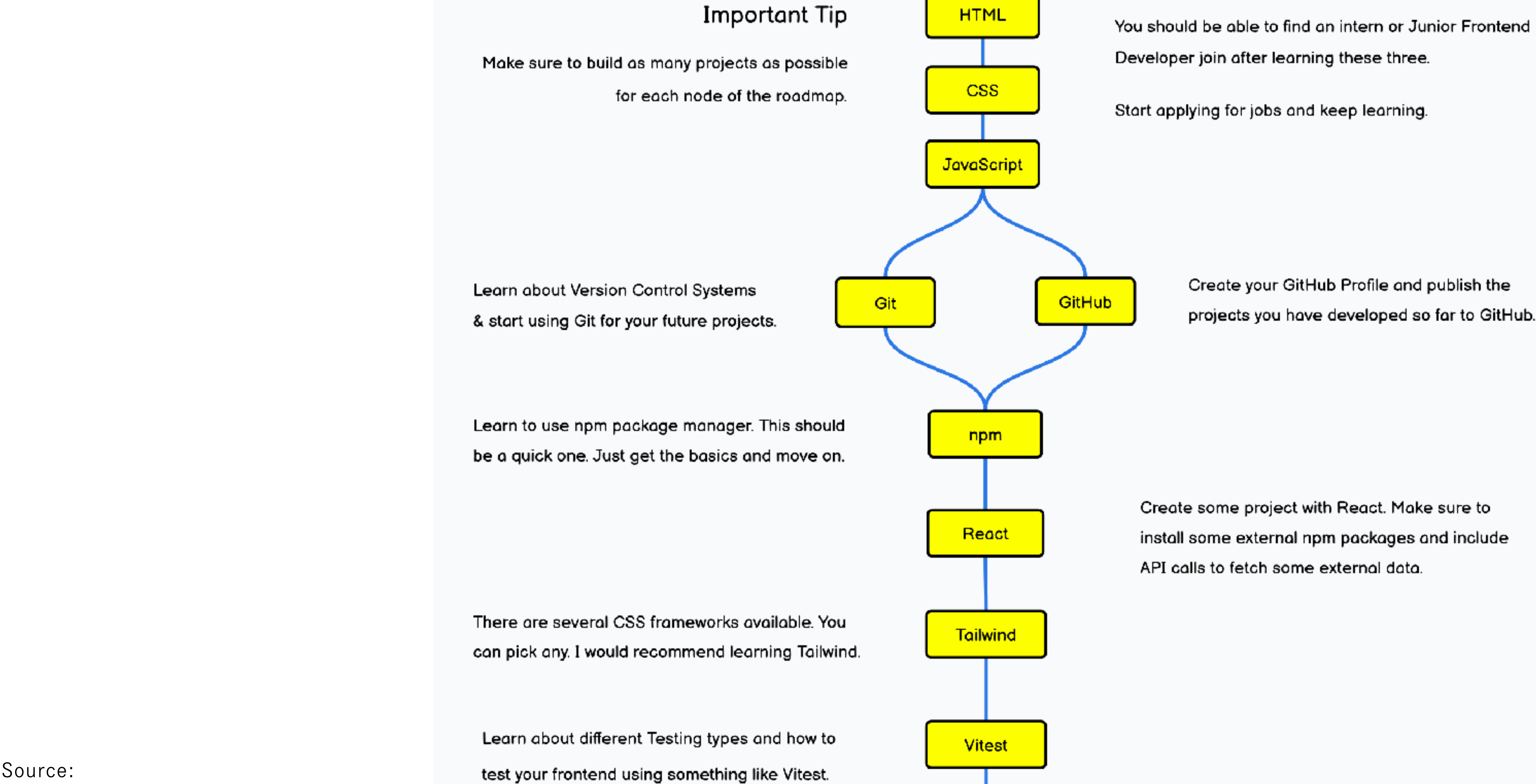
COURSE OUTLINE

This is the beginner friendly version of the Frontend roadmap. Make sure to

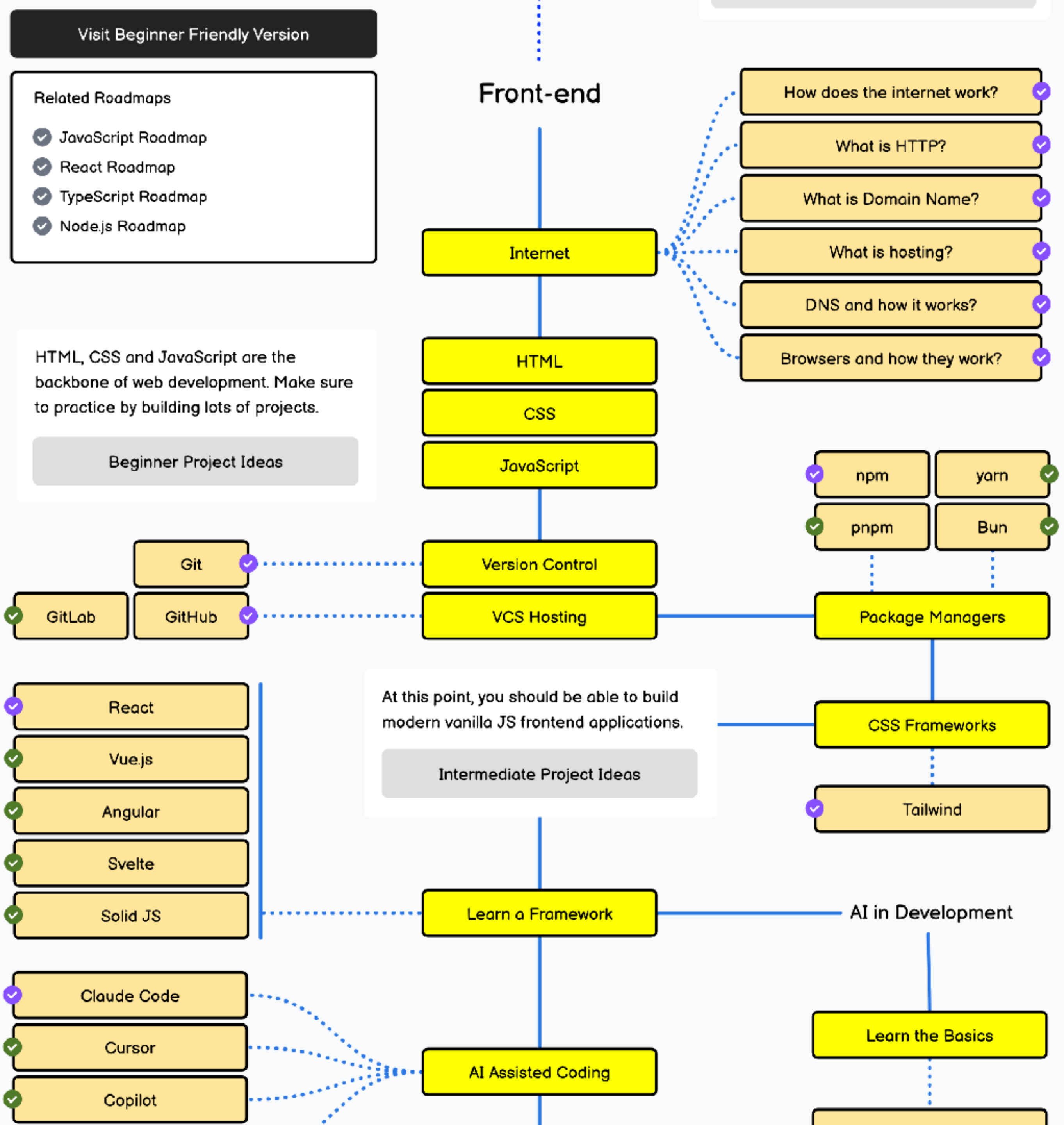
Visit the Detailed Version

Find the detailed version of this roadmap along with other similar roadmaps

roadmap.sh



COURSE OUTLINE



COURSE OUTLINE ROADMAP (2025)

✔ Personal Recommendation / Opinion

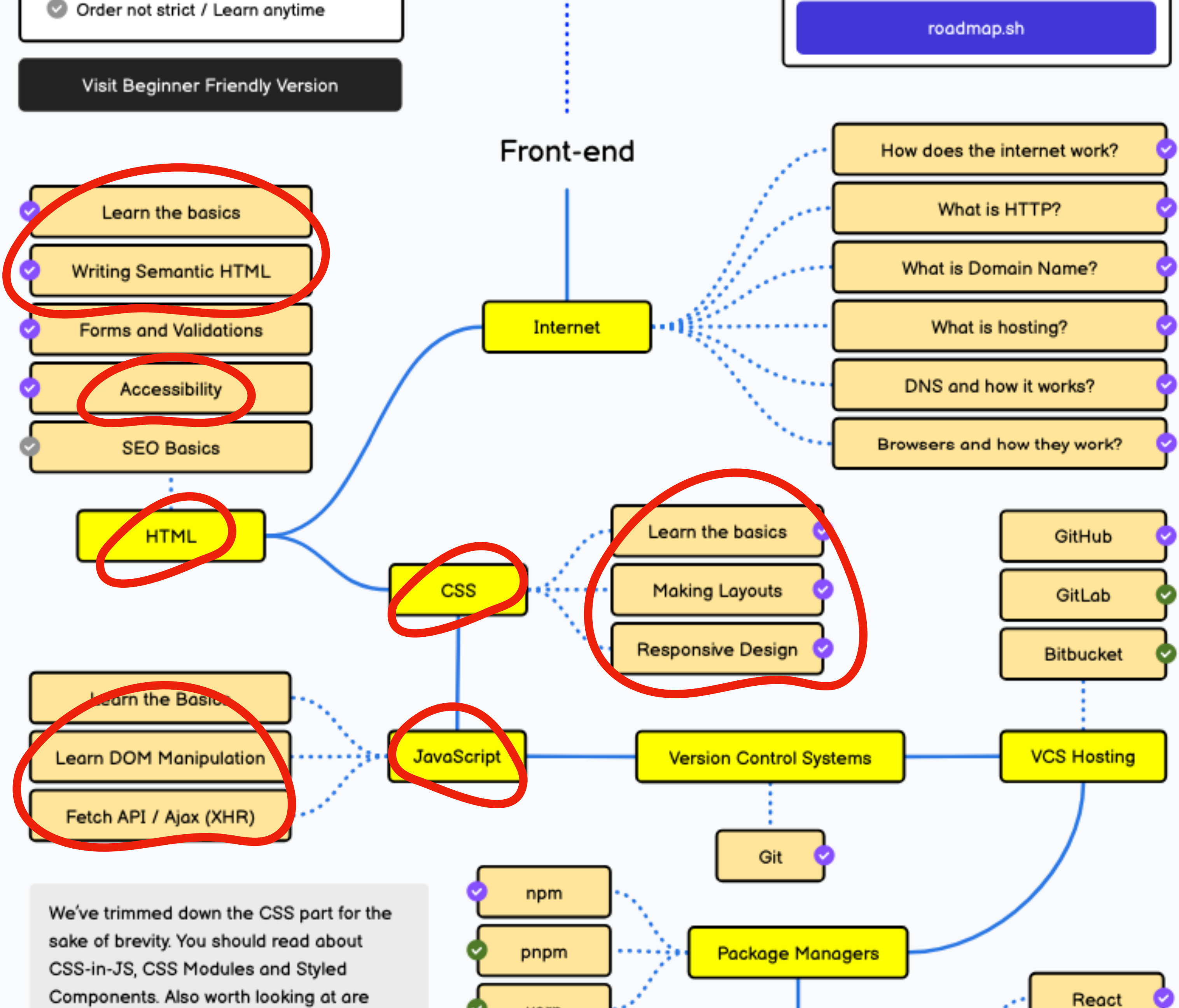
✔ Alternative Option / Pick this or purple

✔ Order not strict / Learn anytime

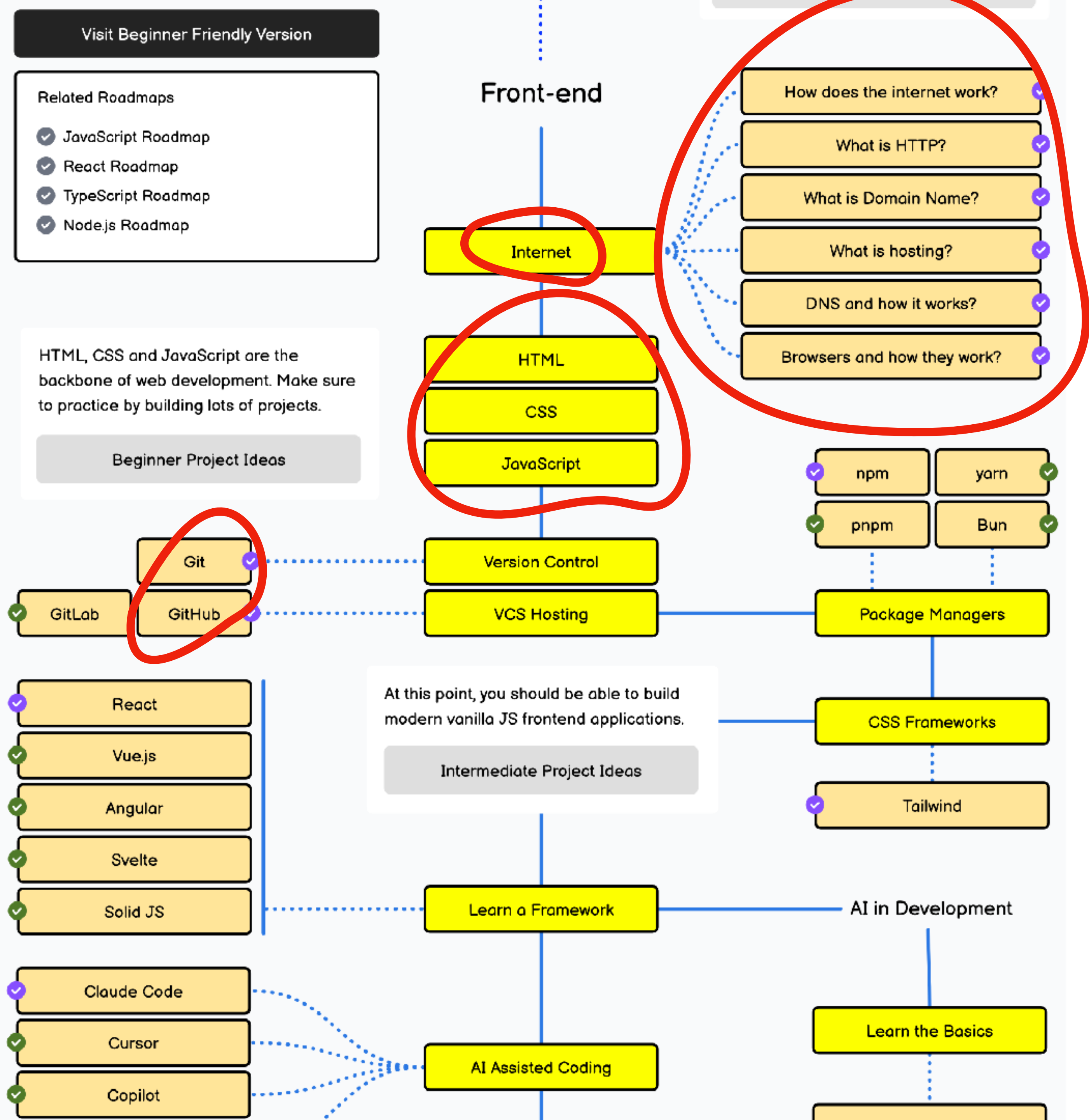
Visit Beginner Friendly Version

Find the detailed version of this roadmap along with other similar roadmaps

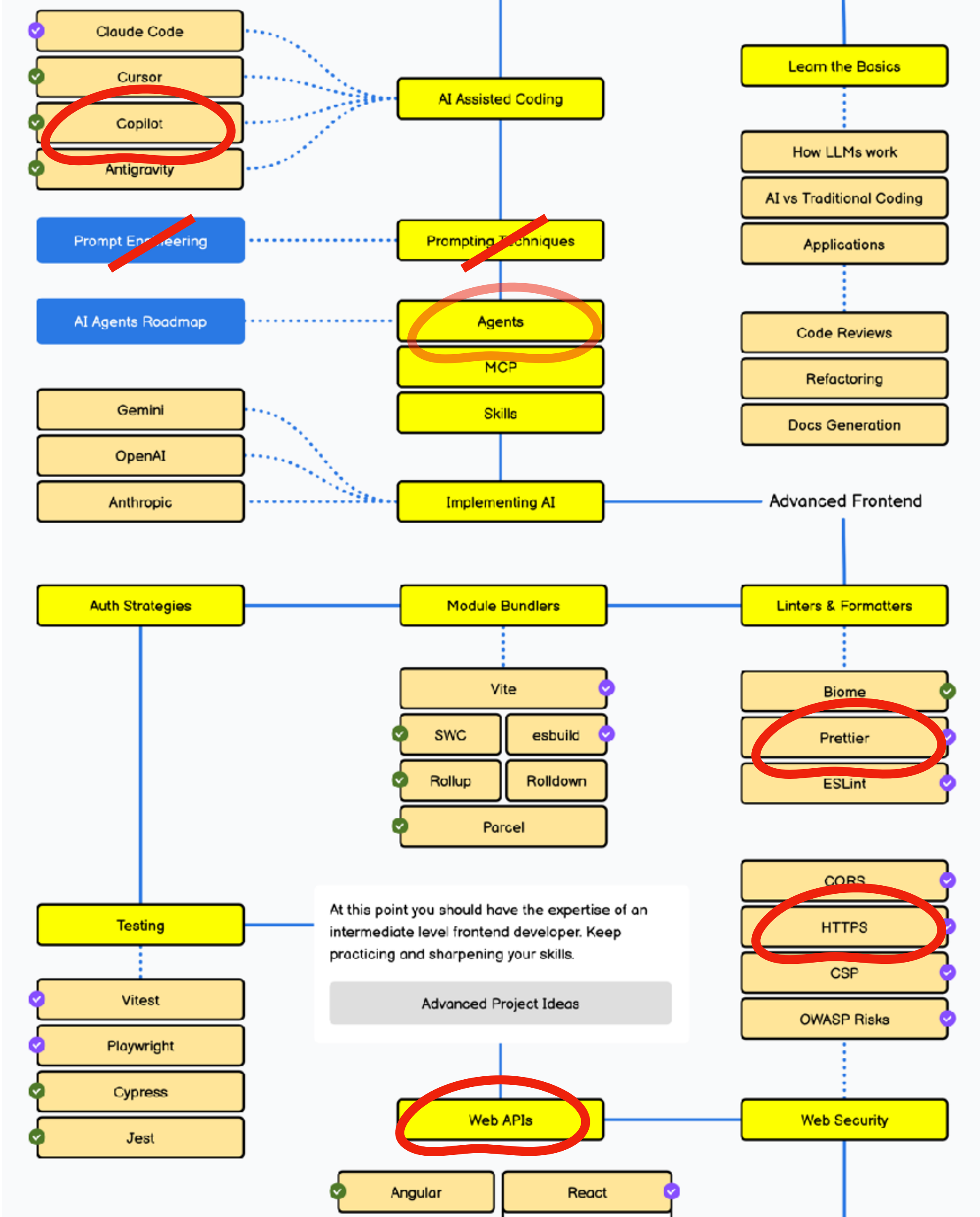
roadmap.sh



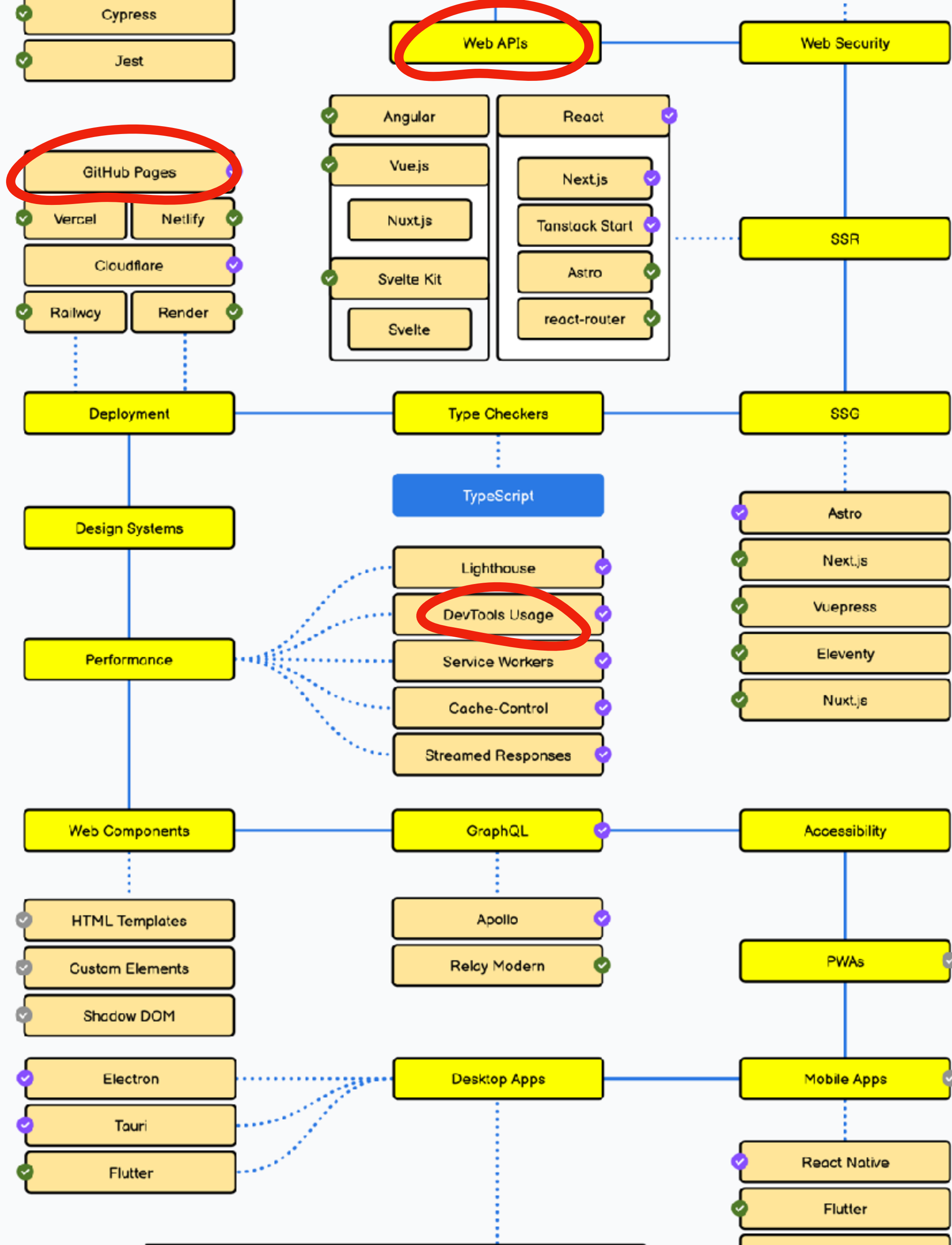
COURSE OUTLINE



COURSE OUTLINE



COURSE OUTLINE



COURSE OUTLINE



It's going to be a lot!

ADMIN

- 80% Attendance
- Grading
 - Exercise
 - Presentation
 - Participation in class

GRADING

- Grade your own work. Assign yourself a public grade and write a brief note explaining your reasoning — both are visible to everyone.
- Your final grade combines this self-assessment with your presentation and participation grade.
- → Active engagement, curiosity, and presence in class carry real weight.

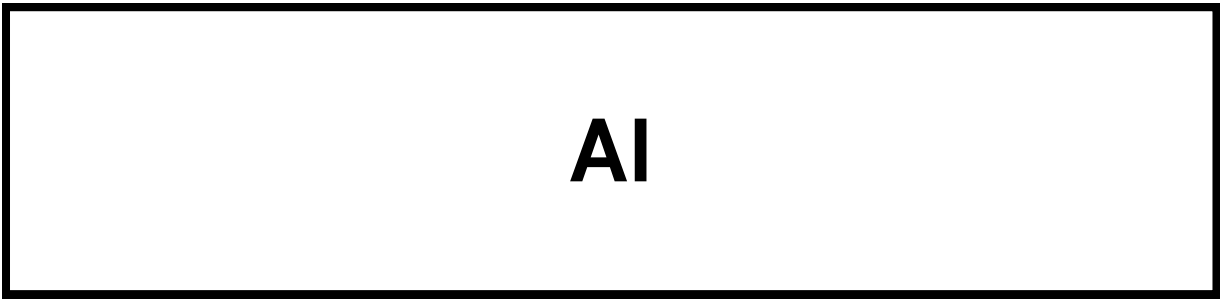
If highly unlikely, I will correct it

PROGRAMMING IN THE AGE OF AI?

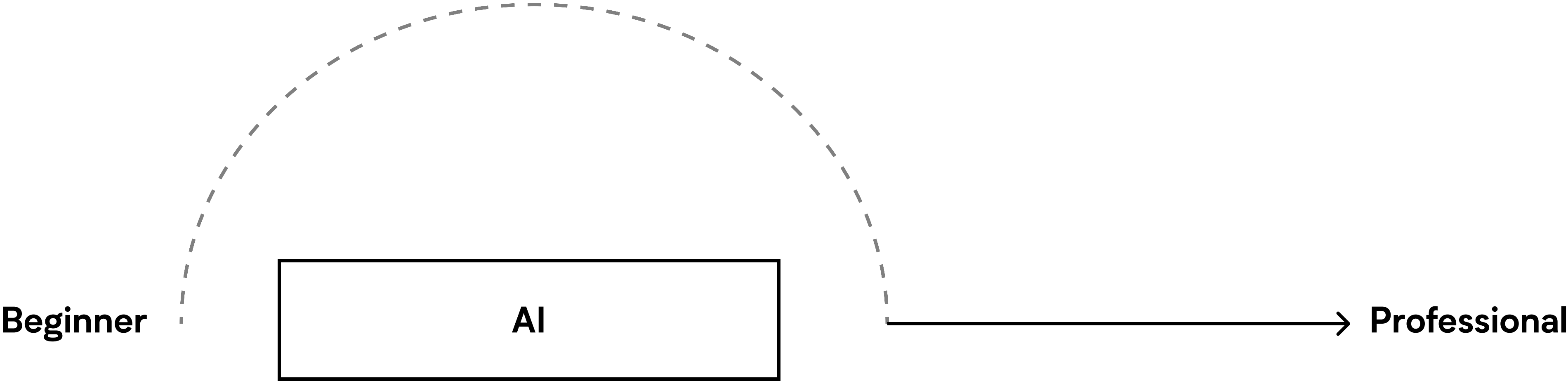


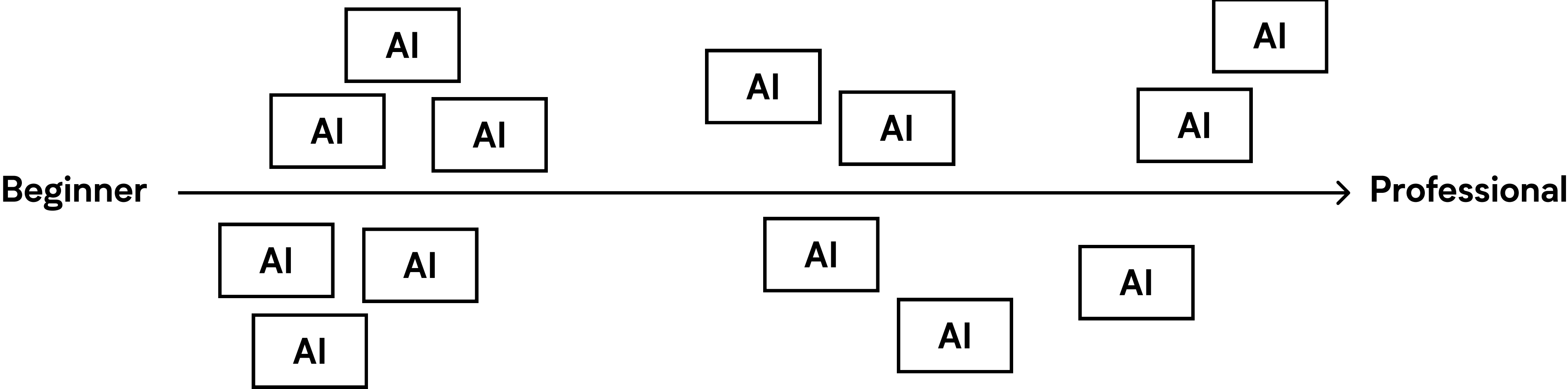
PROGRAMMING IN THE AGE OF AI?

Beginner



Professional

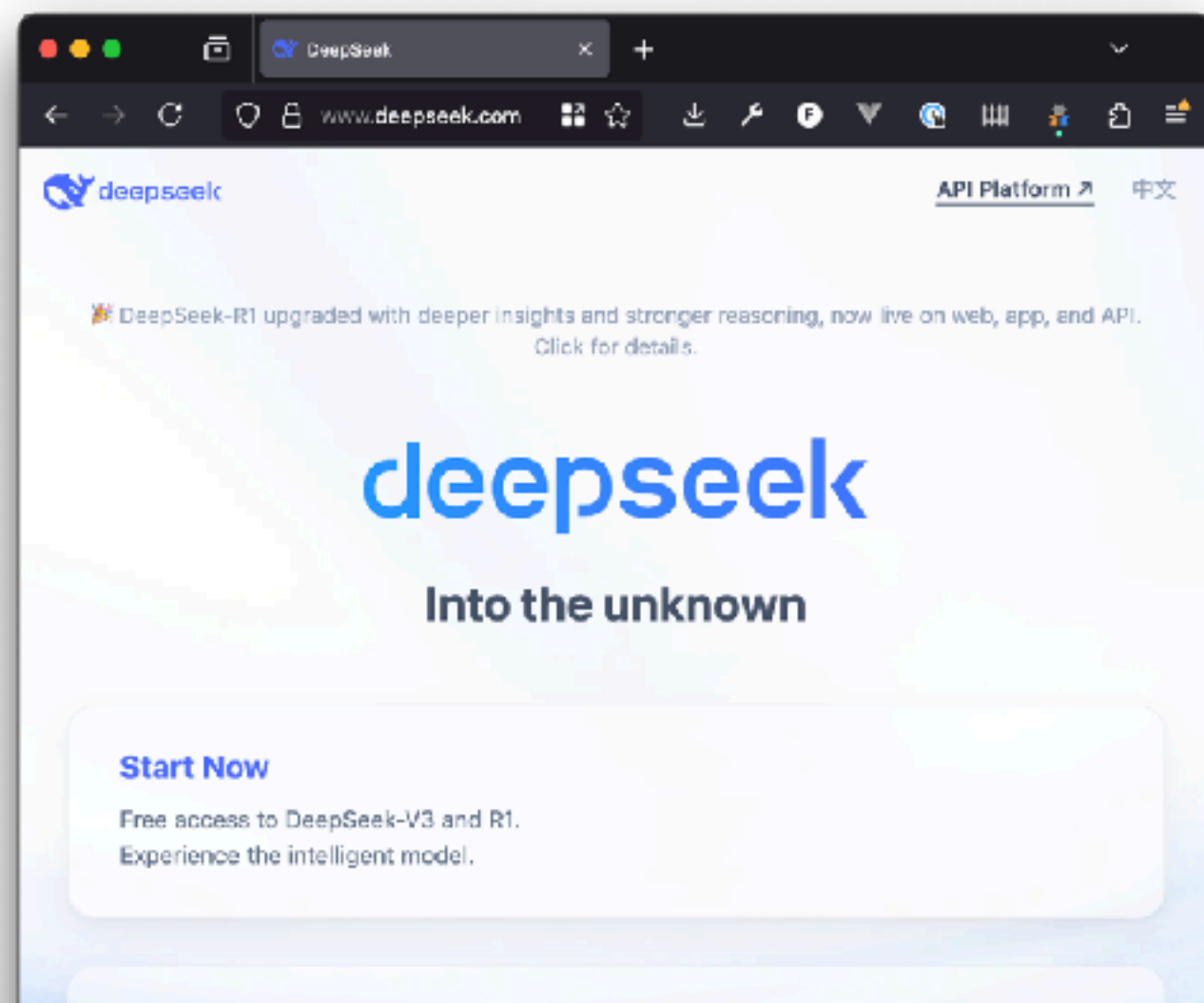
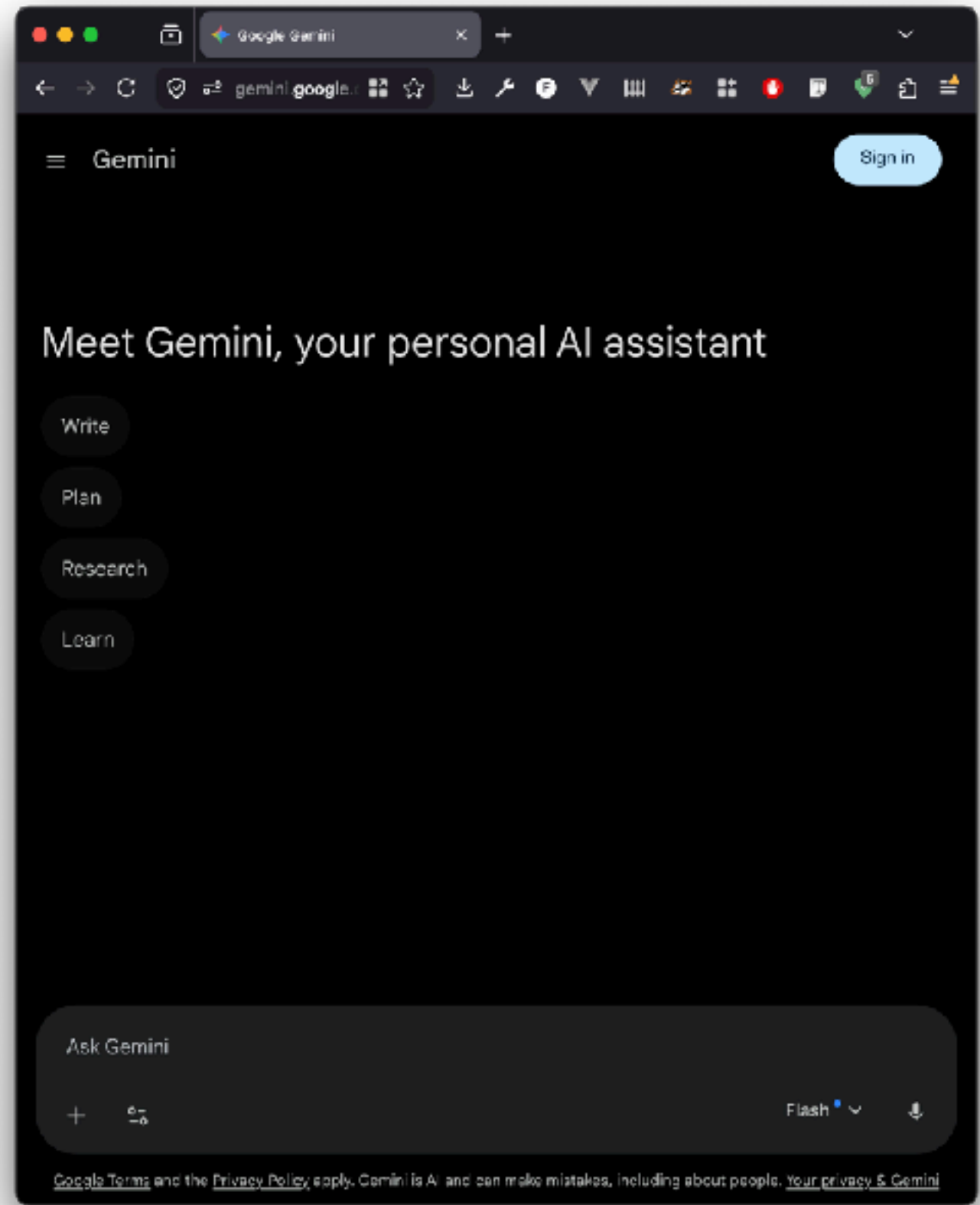




PROGRAMMING IN THE AGE OF AI?

- «You cannot direct what you can't read.»
- Way of thinking
- Problem solving / ecosystem knowledge
- It's fun!

AI

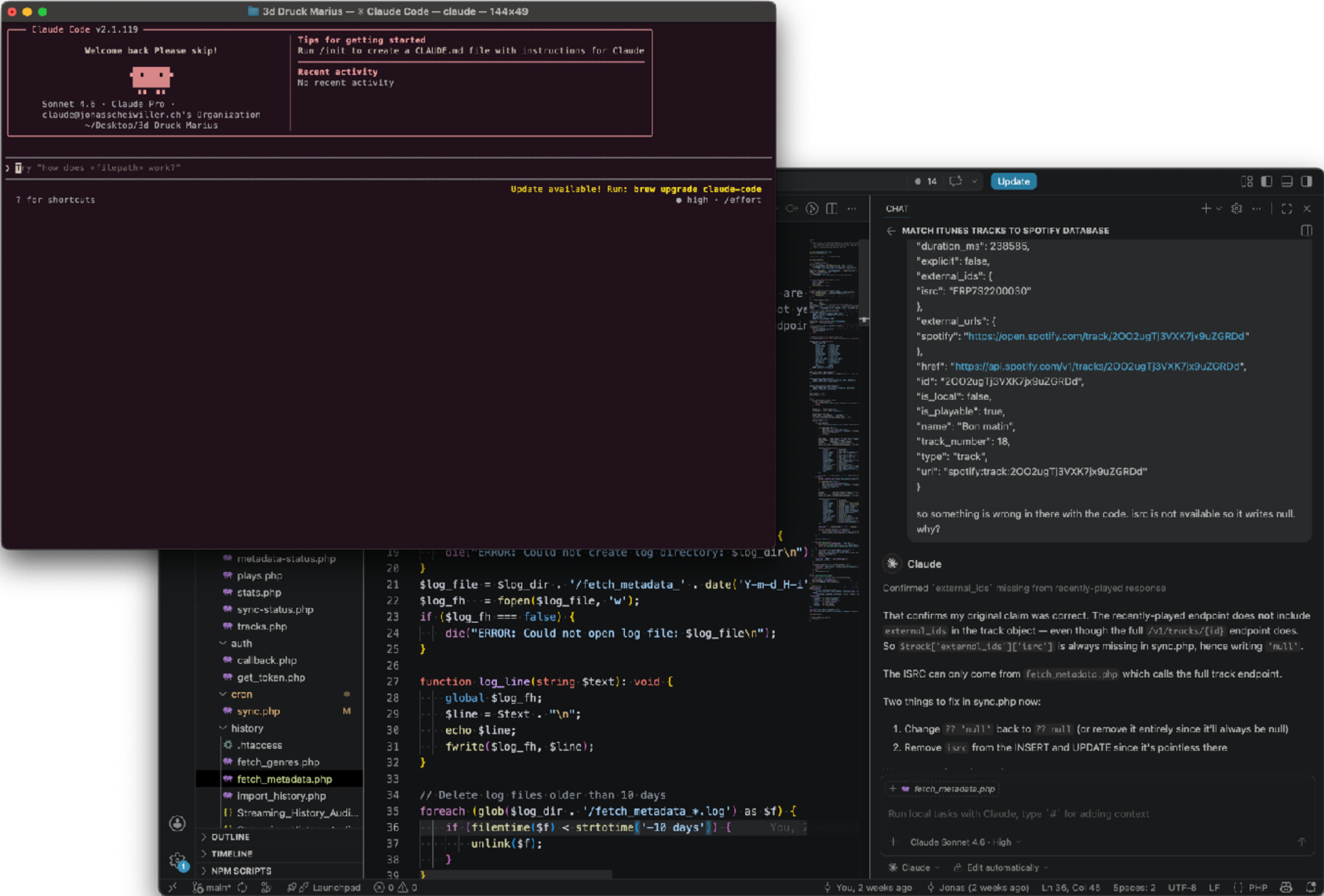


... etc.

Mistral – Le Chat
Anthropic – Claude
Gemini
replit.com
Deepseek

AI (AGENTS)

Claude Code
Github Copilot in VS Code



CALENDAR

June 2026

W24	Mon 8	Tue 9	Wed 10	Thu 11	Fri 12	
all-day						
08:00						
09:00	Webtechnologies - ZT 4.K16 ☉ 09:00 – 12:00	Webtechnologies - ZT 4.K16 ☉ 09:00 – 12:00	Self study – Webtechnologies - ZT 4.K16 ☉ 09:00 – 12:00	Mentorings on Demand – Webtechnologies - ZT 4.K16 ☉ 09:00 – 12:00		
10:00						
11:00						
12:00						
13:00	Webtechnologies - ZT 4.K16 ☉ 13:00 – 17:00	Webtechnologies - ZT 4.K16 ☉ 13:00 – 17:00	Self study – Webtechnologies - ZT 4.K16 ☉ 13:00 – 17:00	Webtechnologies - ZT 4.K16 ☉ 13:00 – 17:00	Presentations – Webtechnologies - ZT 4.K16 ☉ 13:00 – 17:00	
14:00						
15:00						
16:00						
17:00						

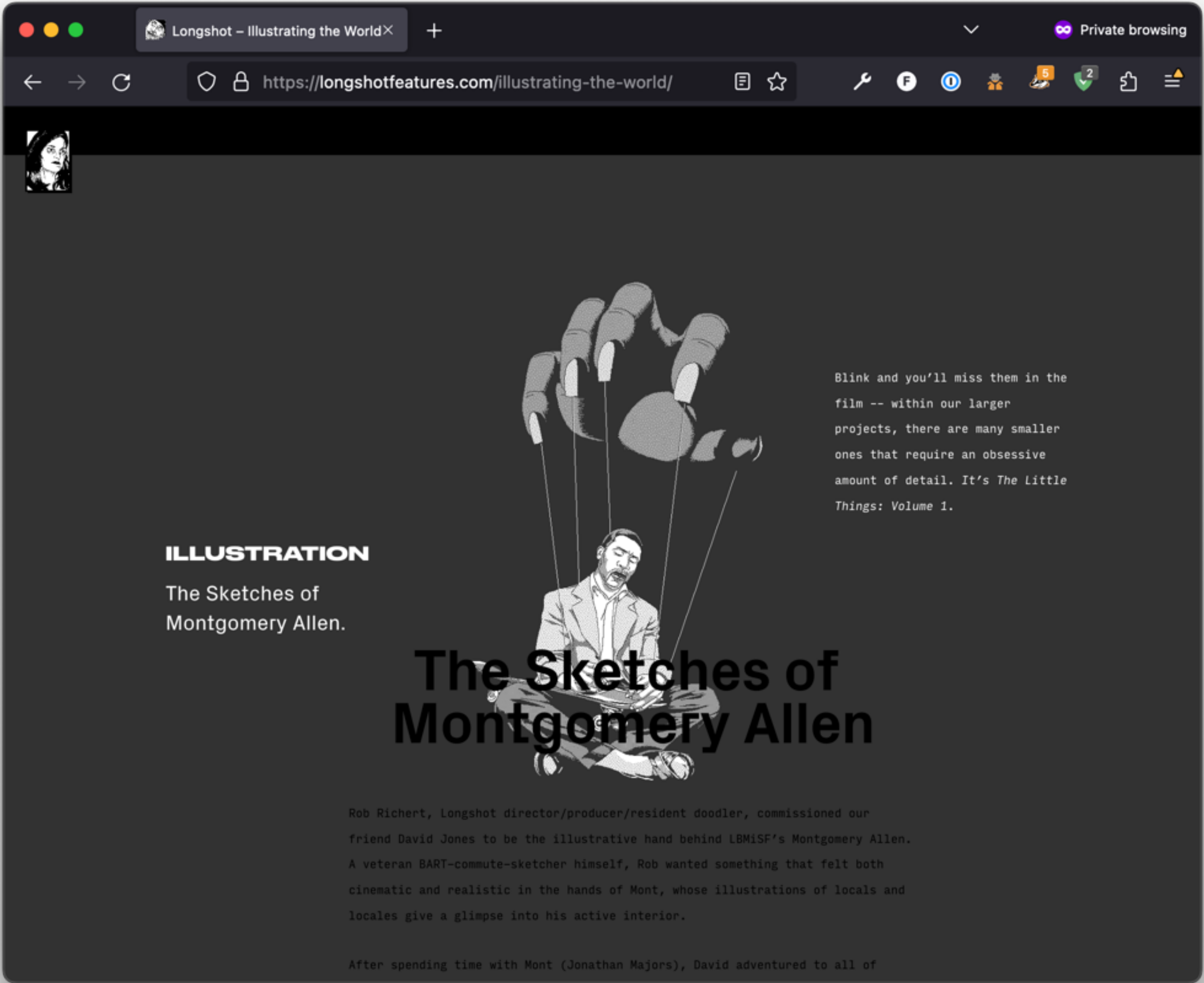
RESOURCES

- IAD Calendar
- Intranet
- ~~Paul~~
- Github repo: <https://github.com/Jones-S/zhdk-web-technologies-2026>

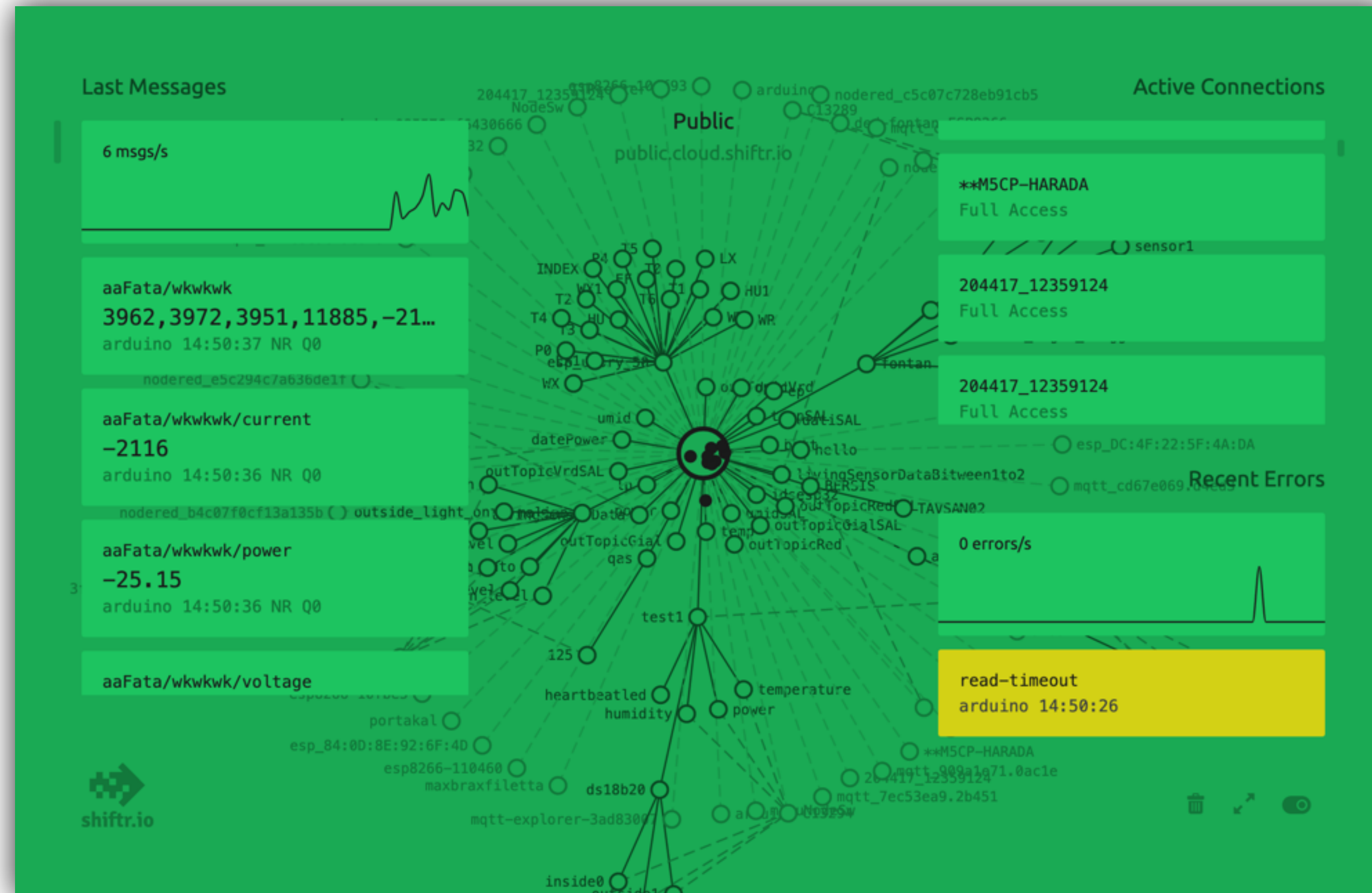
WEBTECHNOLOGIES – WHY?



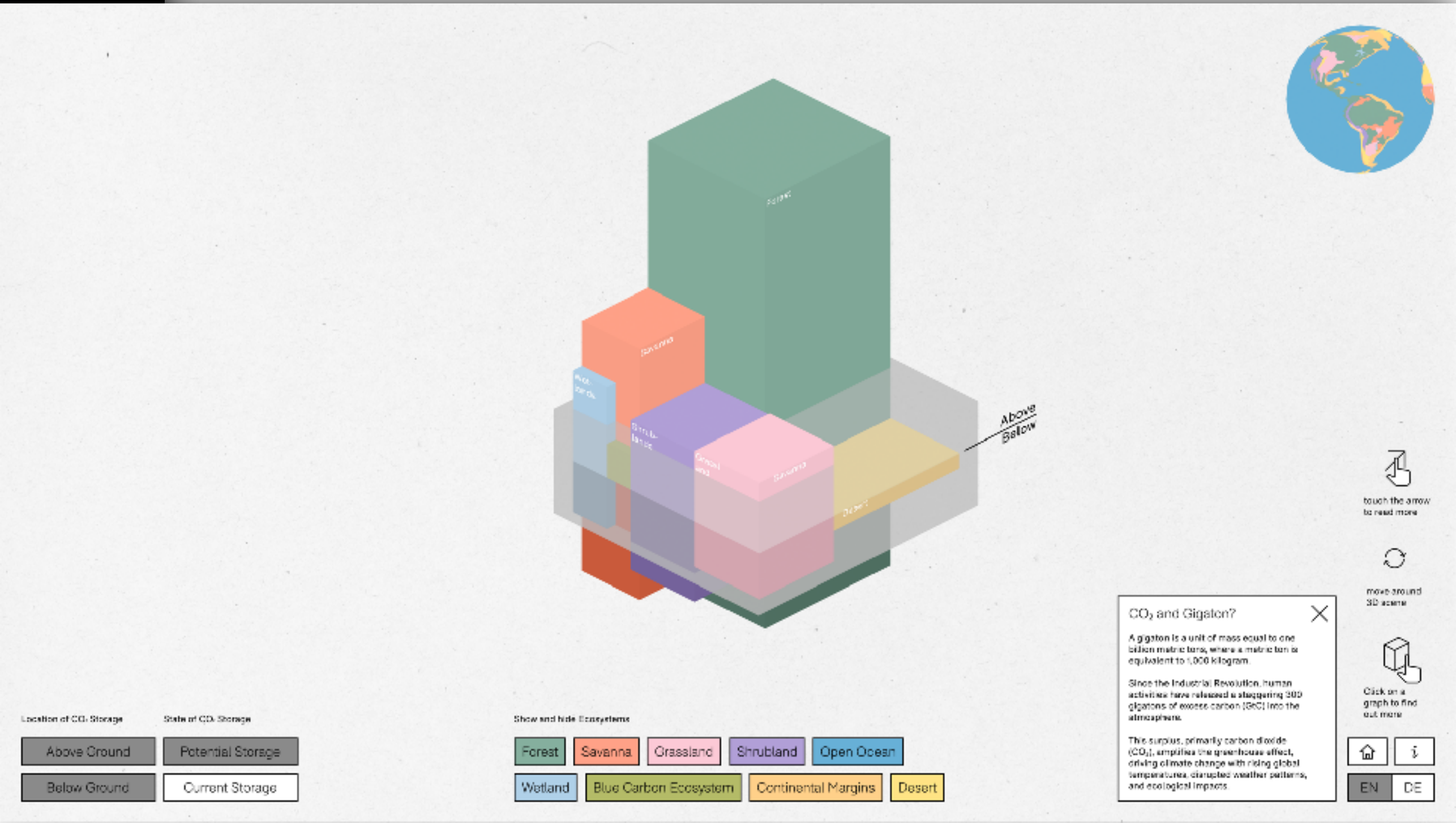
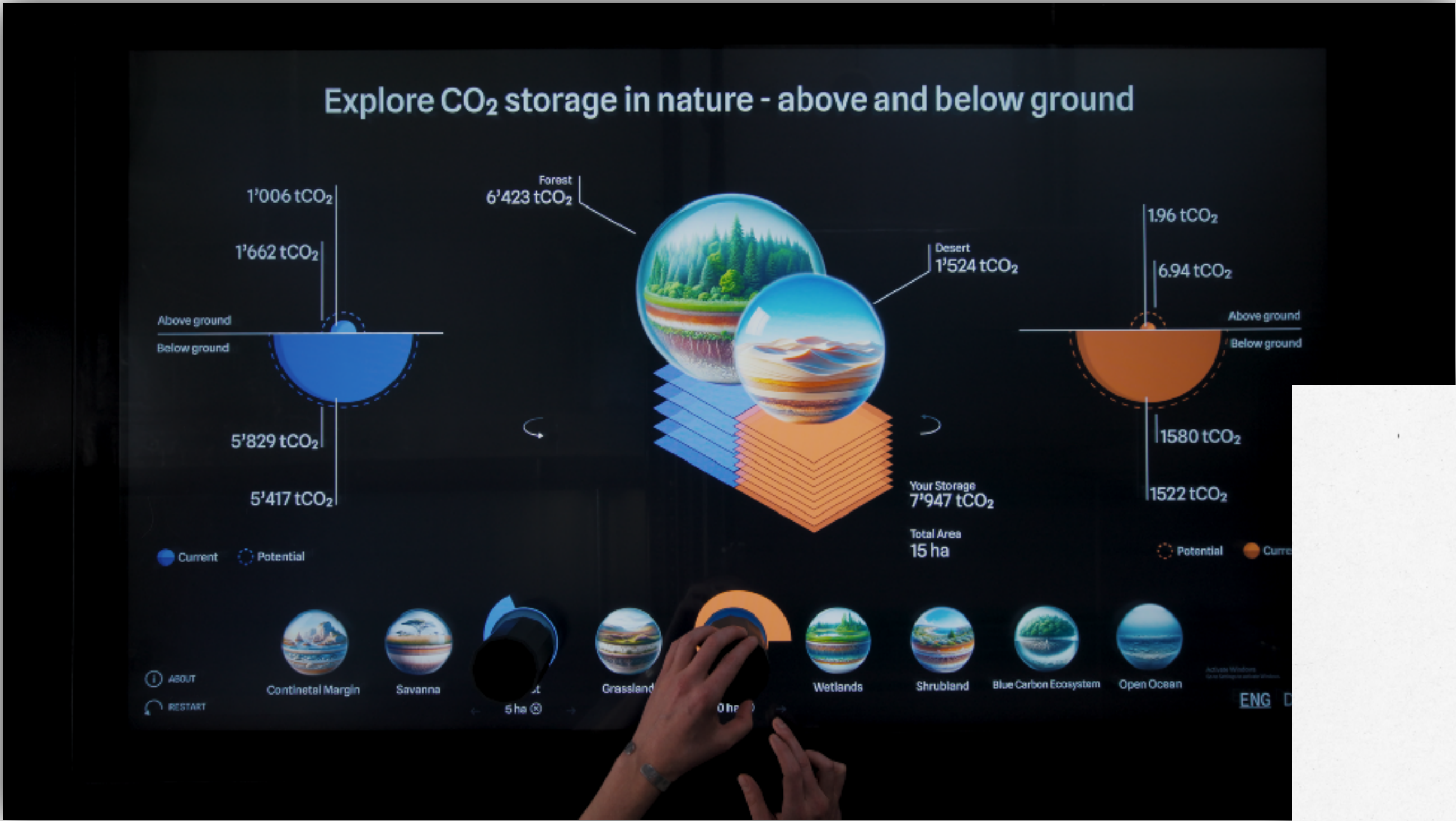
WEBTECHNOLOGIES – WHY?



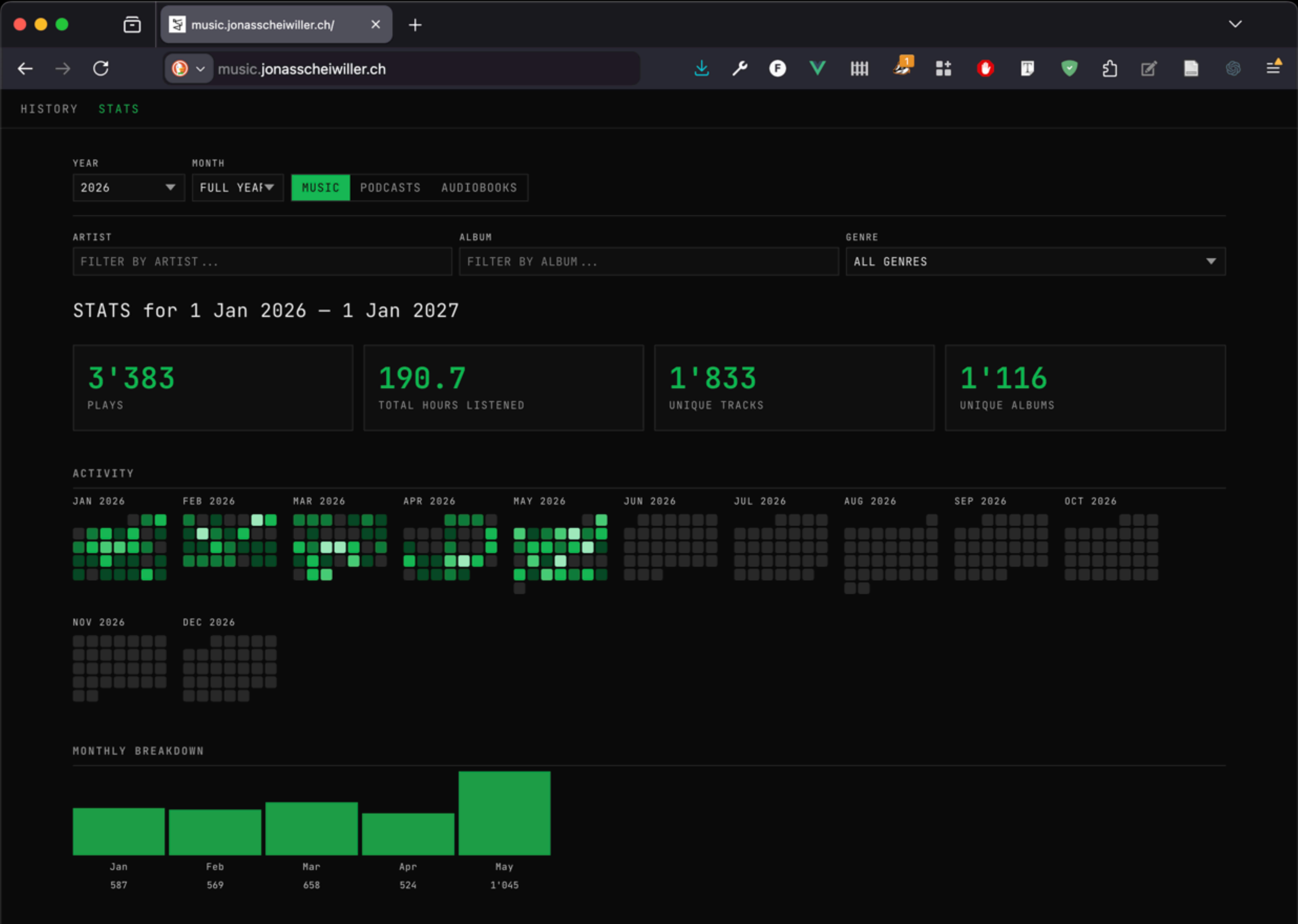
WEBTECHNOLOGIES – WHY?



WEBTECHNOLOGIES – WHY?



WEBTECHNOLOGIES – WHY?



WEBTECHNOLOGIES – WHY?

- Interfaces/Controls
 - Storytelling & Information
 - (Functional) Prototypes
 - Easy Access (Smartphone)
 - Data Visualization
-
- Also: Portfolio
 - Customize webflow, WIX, cargo etc.
 - Use AI more productively
 - Deploy a website

GOAL OF THIS COURSE

- Understanding of possibilities
- Knowledge of tools and terms
- First experiences with web technologies
- Foundation for other modules

WHAT DO YOU ALREADY KNOW?

WHAT DO YOU EXPECT?

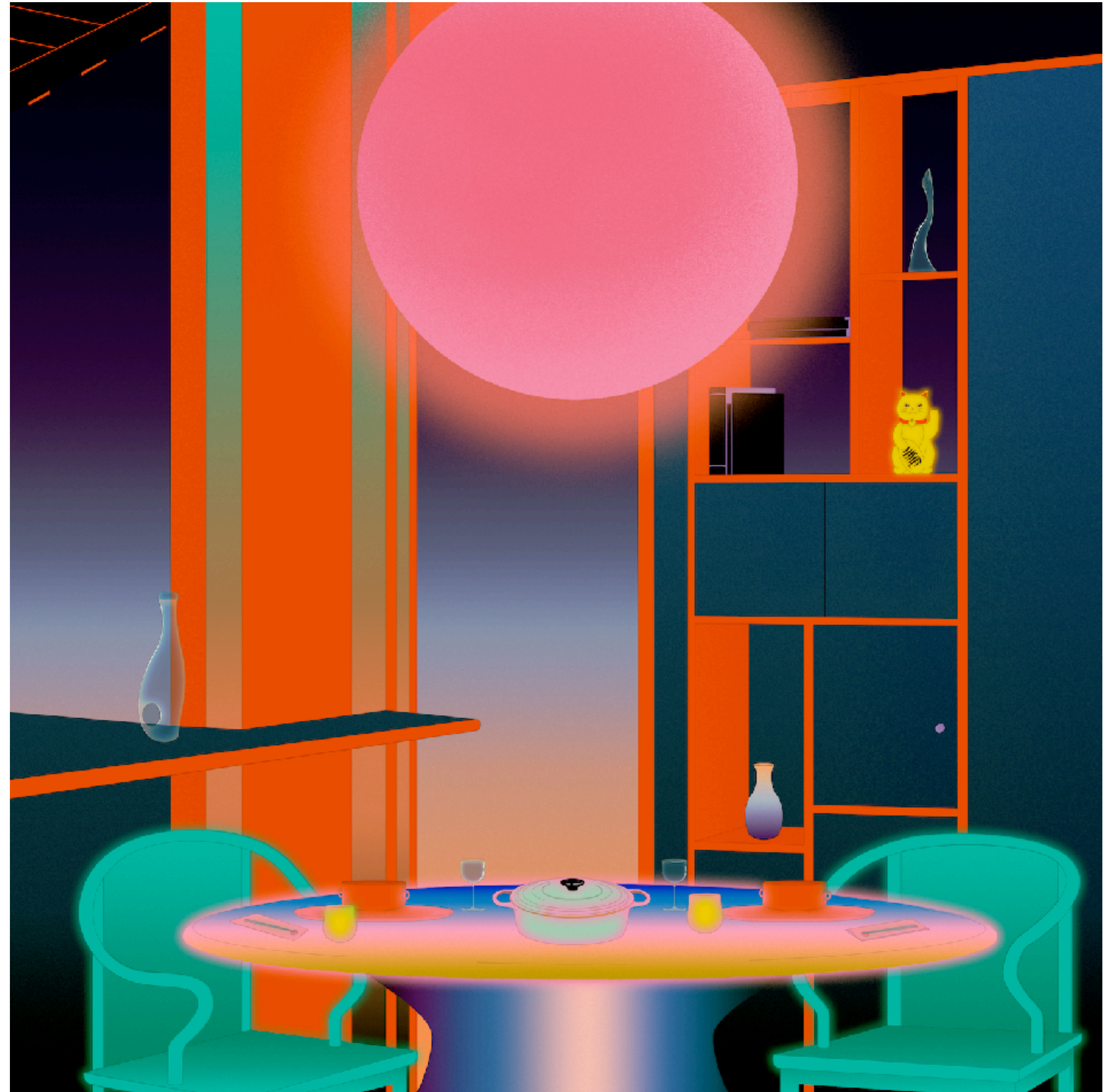
WHAT DO YOU ALREADY KNOW?

```
function greaterThanTen(x) {  
    return x > 10;  
}  
  
function sumAndCheck(a, b) {  
    let result = a + b;  
    return greaterThanTen(result);  
}  
  
console.log(sumAndCheck(6, 5));
```

```
let user = {  
    username: "Bob",  
    age: 22  
};
```

- ☐ a. true
- ☐ b. false
- ☐ c. 11
- ☐ d. 10

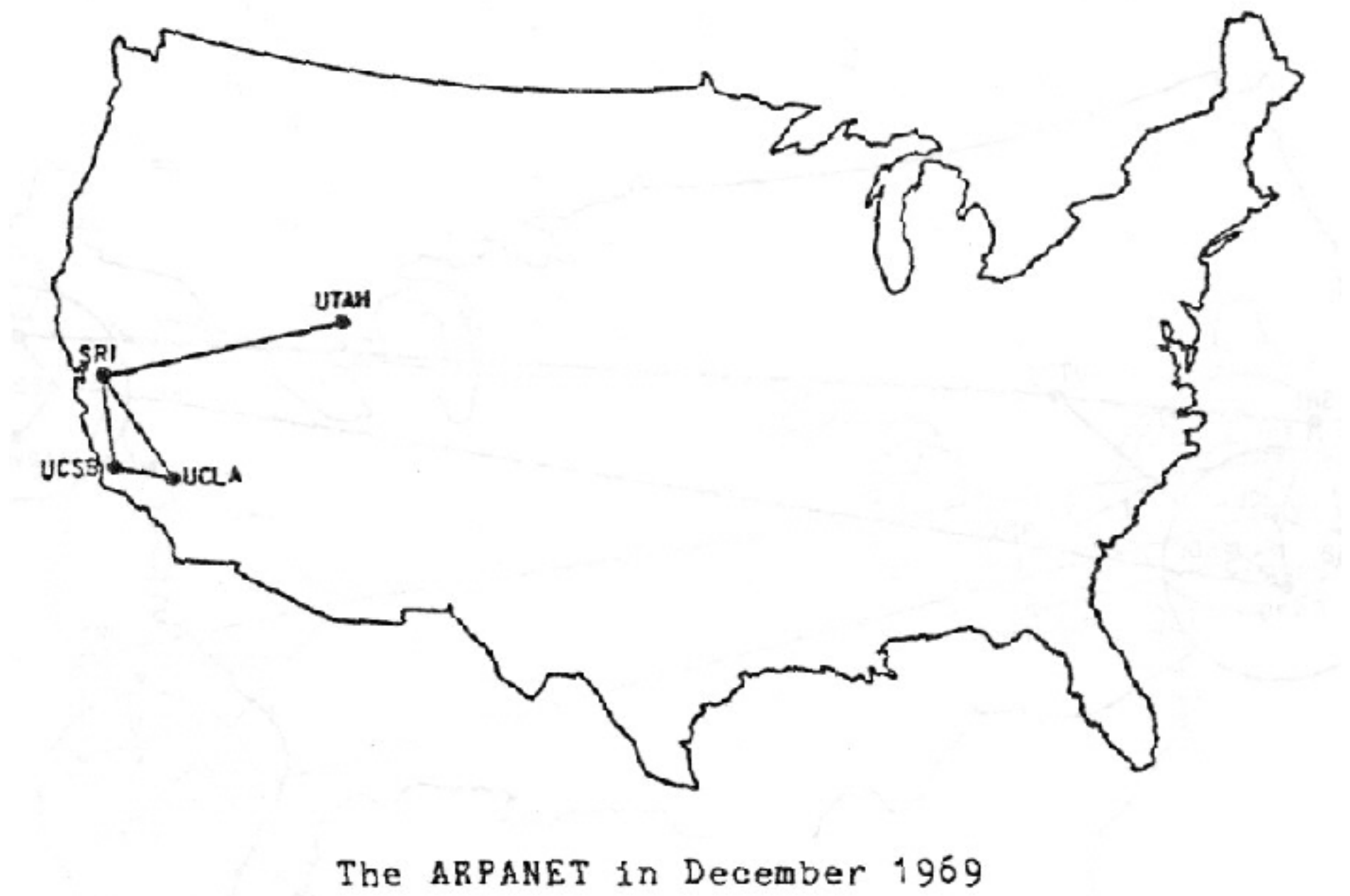
HISTORY OF THE INTERNET AND THE WWW



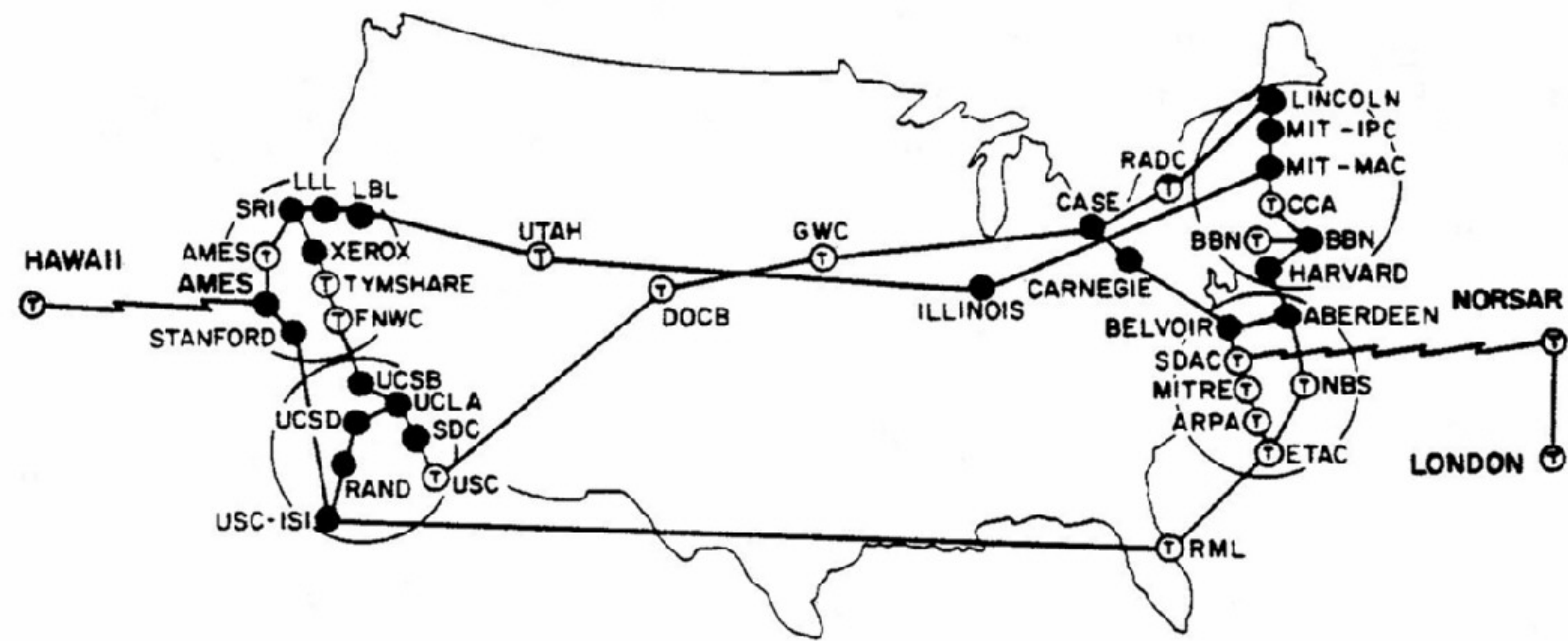
THE BEGINNING



THE BEGINNING

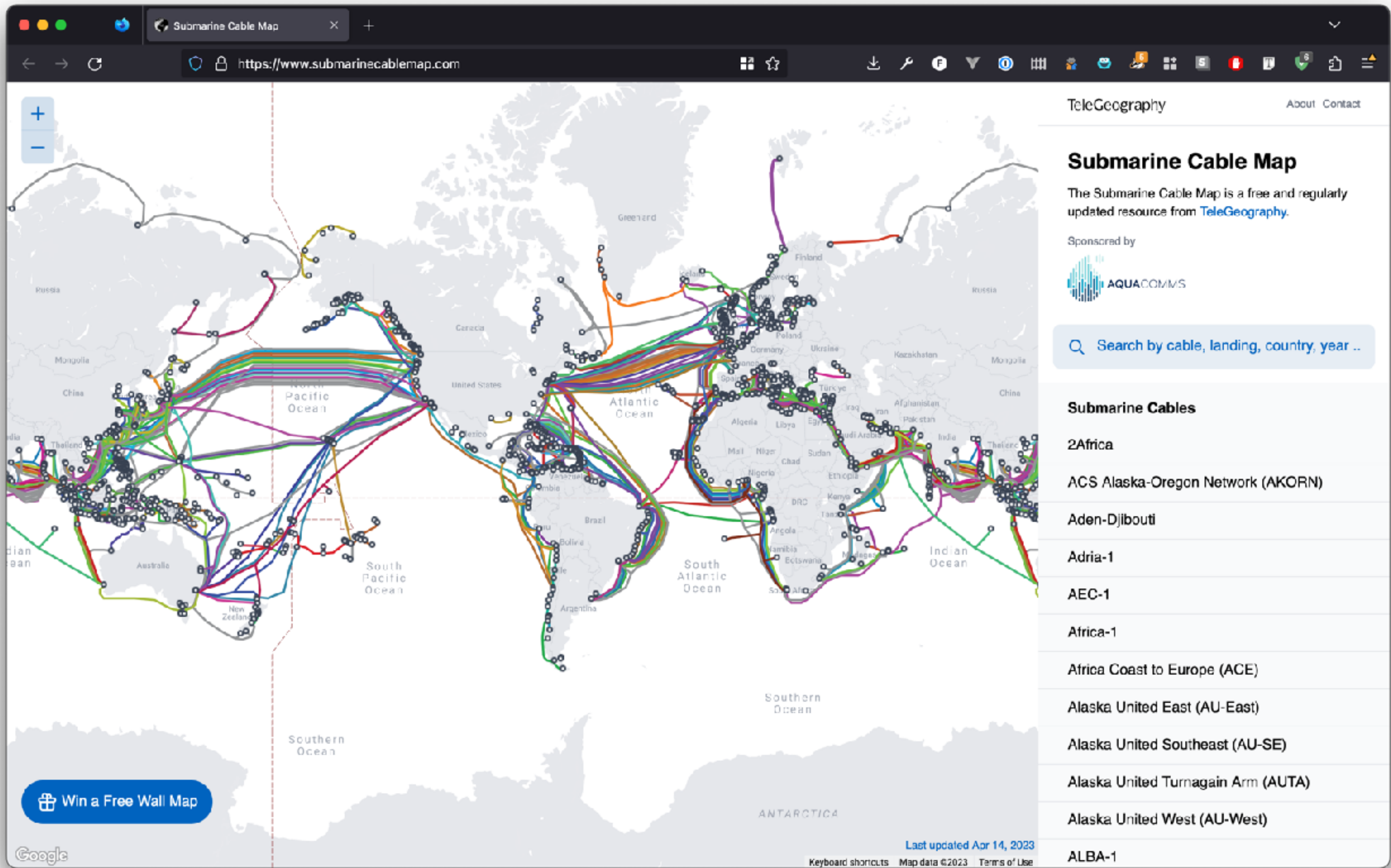


THE BEGINNING



1973: ARPANET goes international

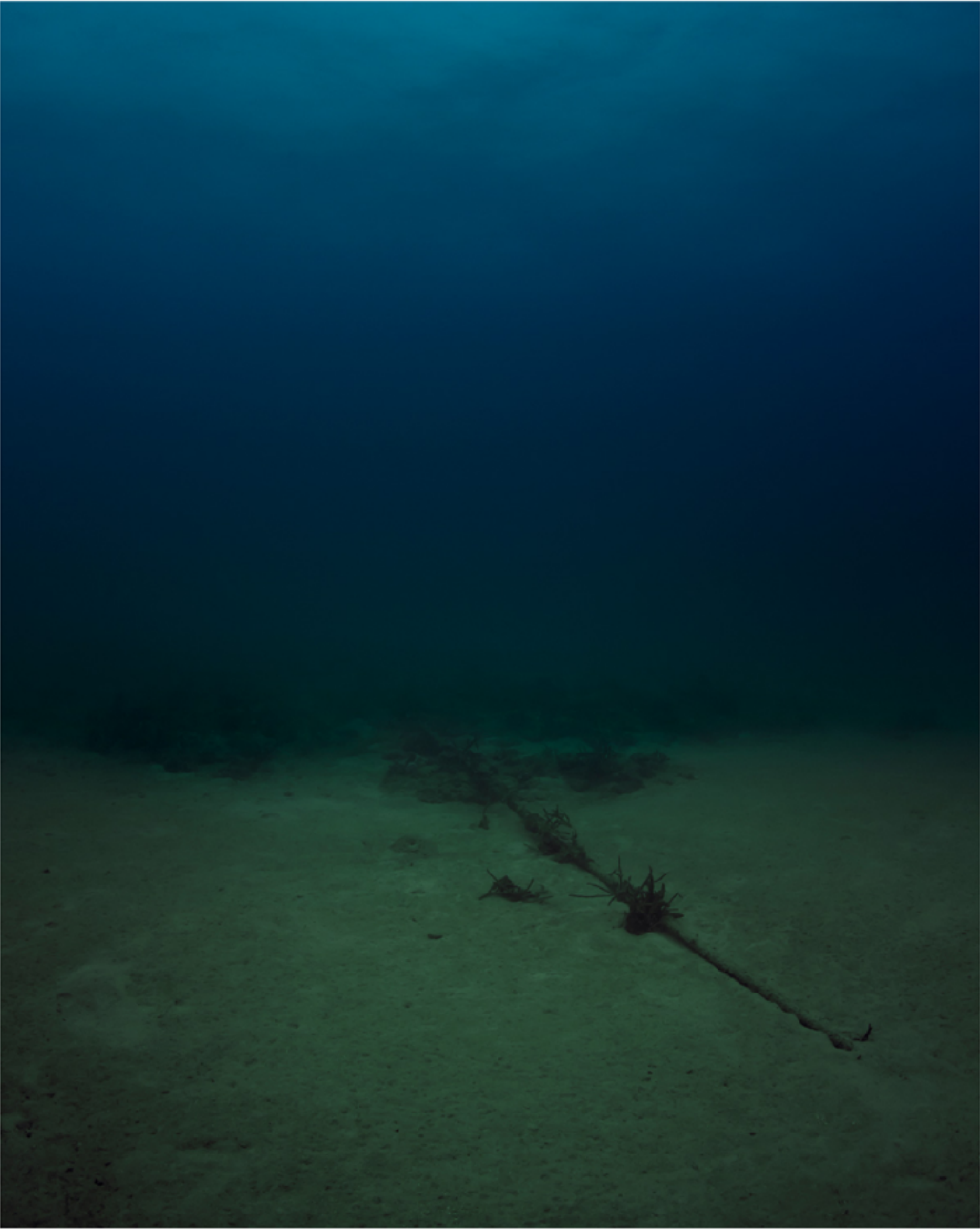
SUBMARINE CABLES → INTERNET



Source:
<https://www.submarinecablemap.com/>
(https://en.wikipedia.org/wiki/Submarine_communications_cable)

SUBMARINE CABLES

Source:
<https://paglen.studio/2020/05/22/undersea-cables/>
Artist: Trevor Paglen

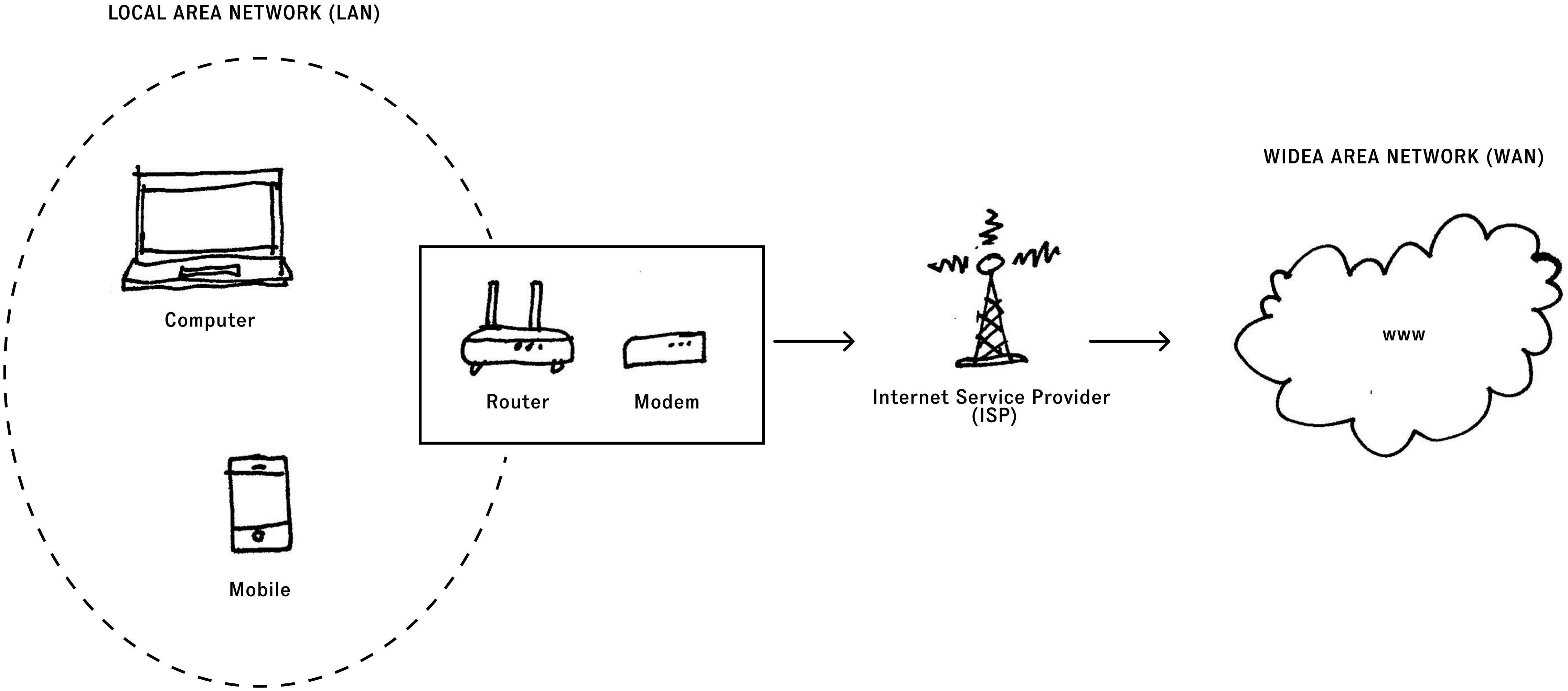


Bahamas Internet Cable System (BICS- 1)
NSA/GCHQ Tapped Undersea Cable
Atlantic Ocean, 2015
C-Print
60 x 48 in.

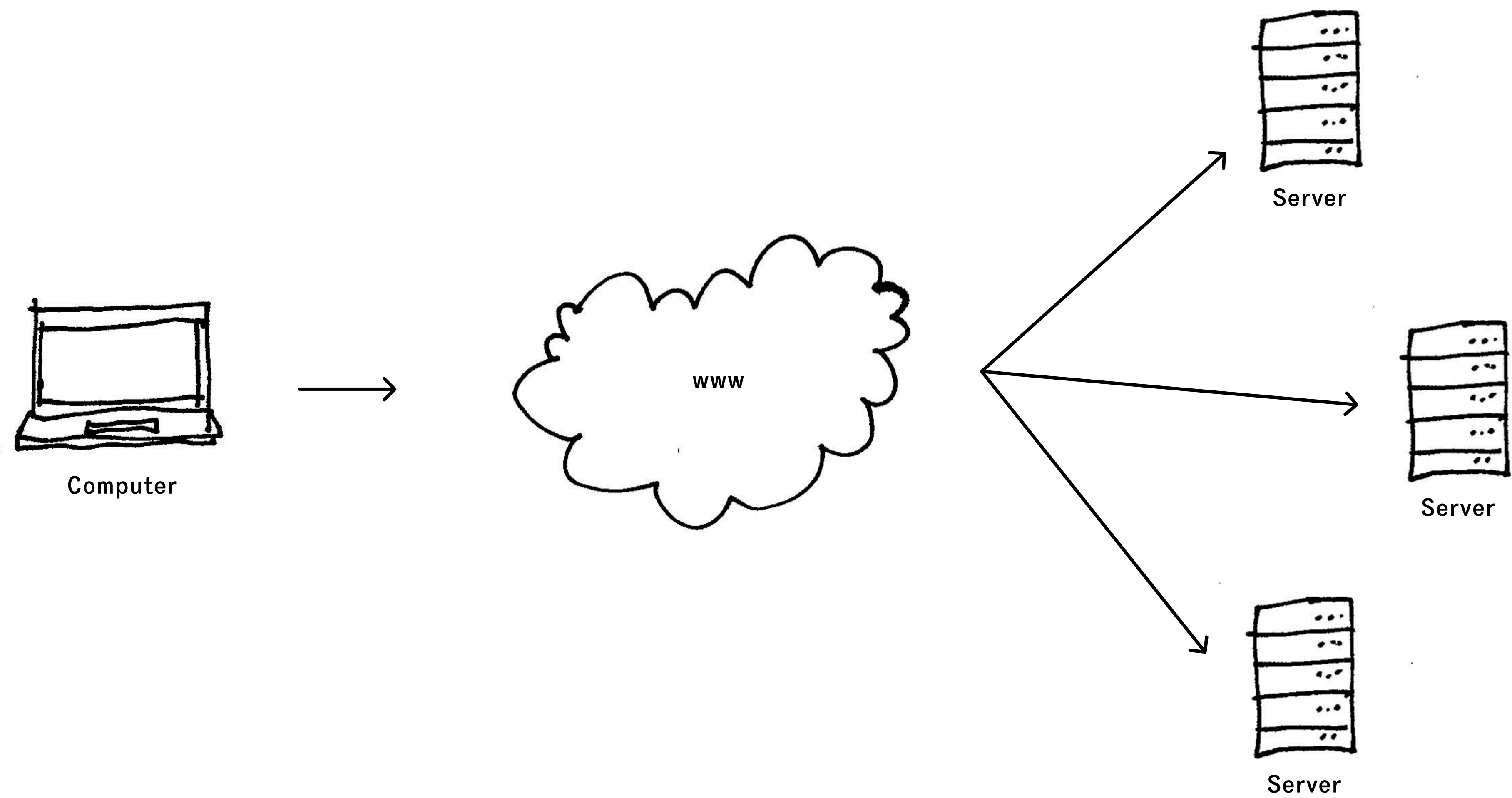
CRITICAL THINKING



CONNECTION



CONNECTION

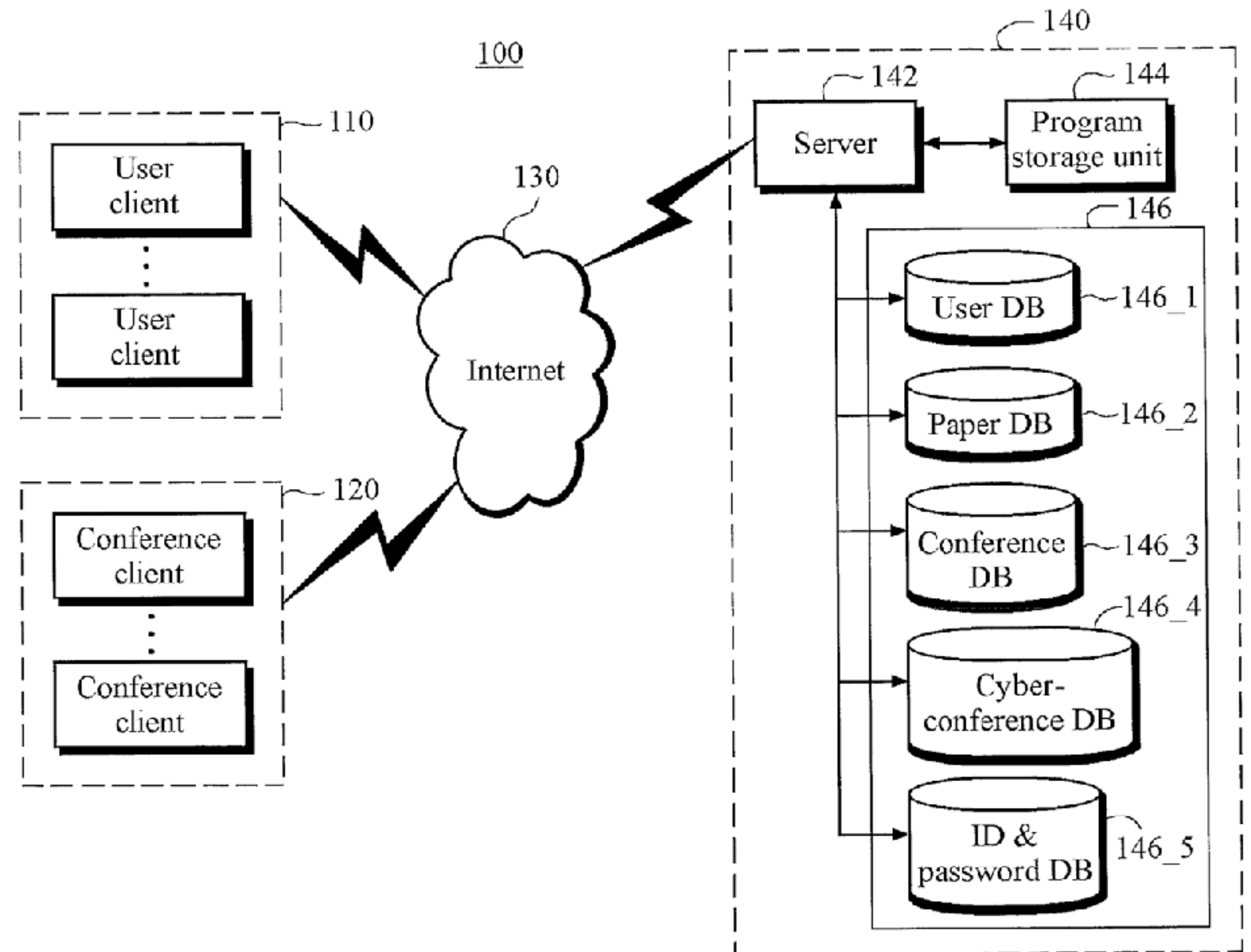


THE CLOUD

The cloud is just somebody else’s computer.

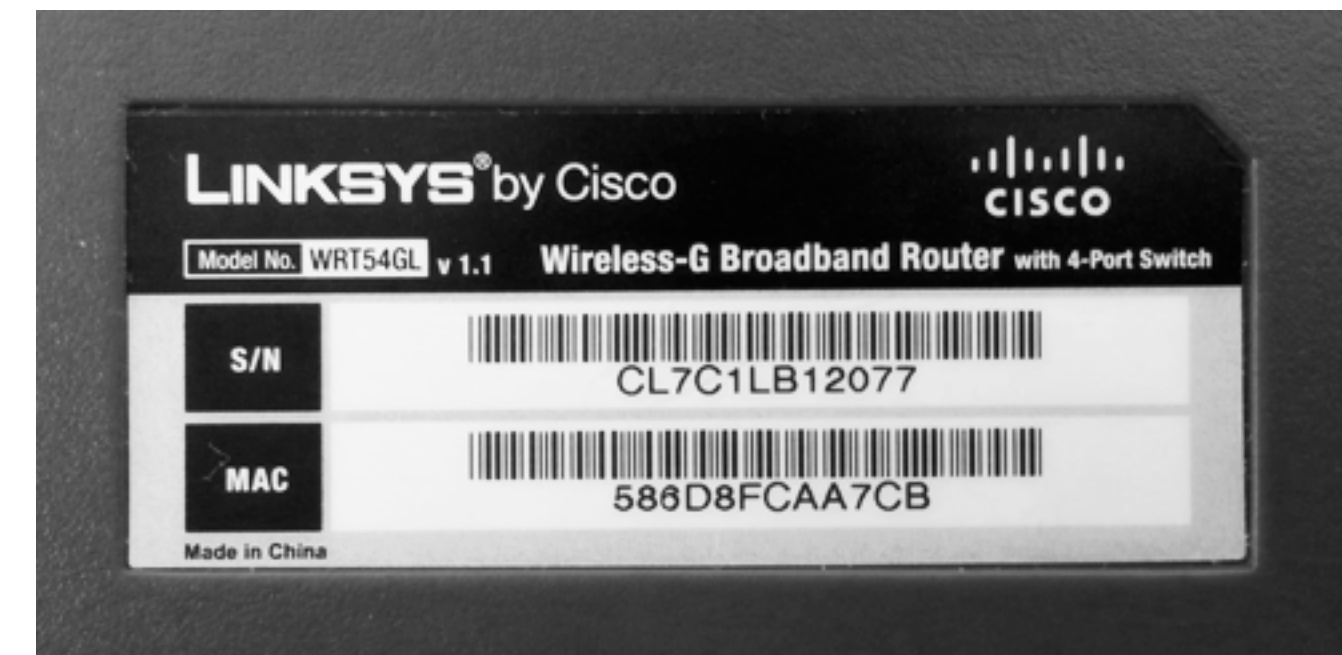
THE CLOUD

- The cloud: A common visual for the internet
- Internet is often depicted as an amorphous blob.

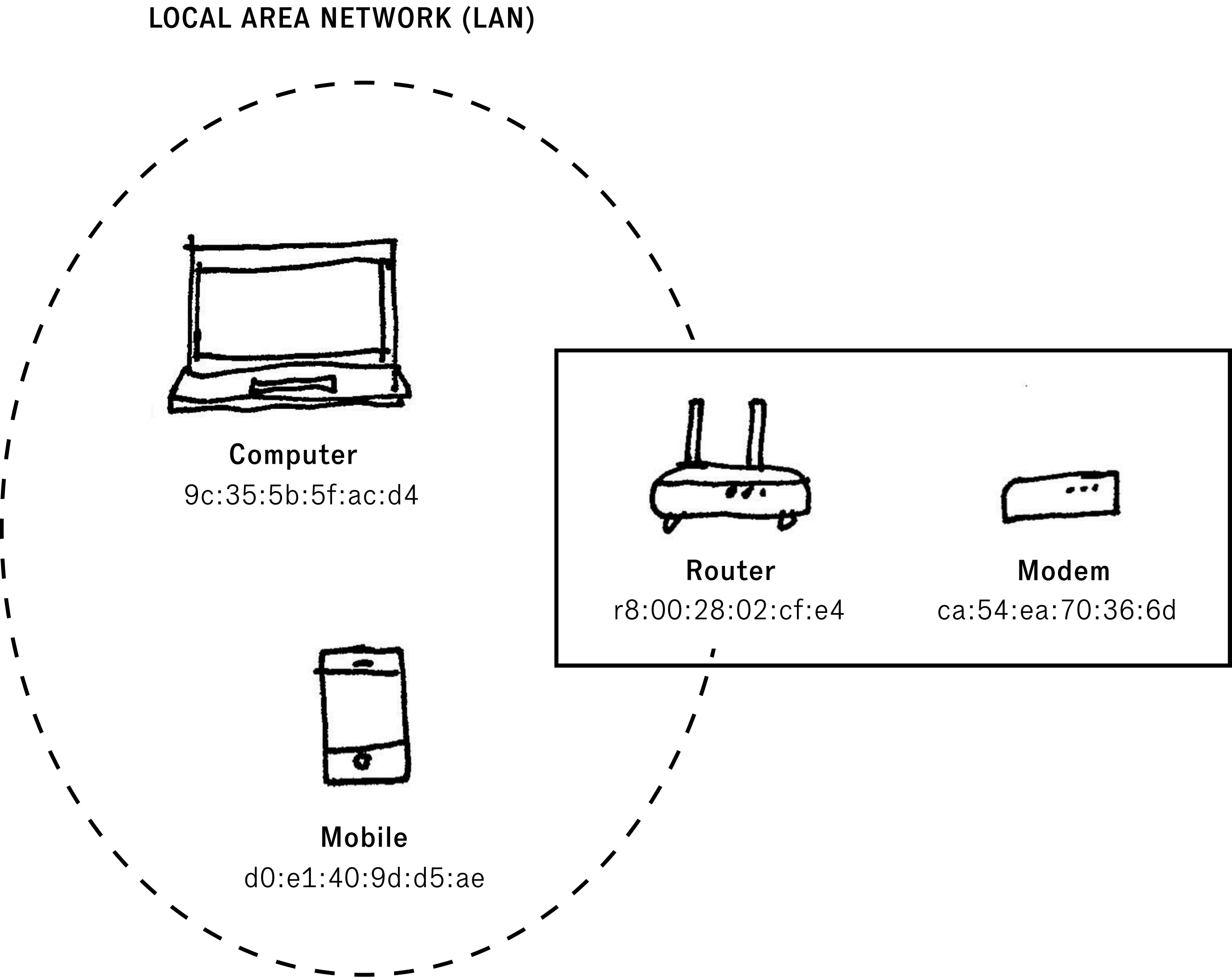


MAC ADDRESSES

- Every device that connects to a network has a MAC-address.
Example: d0:e1:40:9d:d5:ac
- Used to identify a device in a local network.
- $248 = 281,474,976,710,656$
possible MAC addresses
- Device manufacturers are given certain number ranges to assign to their devices.



MAC ADDRESSES



PROTOCOLS

- Set of rules that governs communication between two computers in a network.
- Different layers:
 - Data-Link – Protocols: e.g. Ethernet Protocol
 - Internet Protocols: e.g. IP4, IP6
 - Transport Protocols: e.g. TCP
 - Application Protocols: e.g. Email, FTP, HTTP

Layer architecture [\[edit \]](#)

The recommendation X.200 describes seven layers, labelled 1 to 7. Layer 1 is the lowest layer in this model.

OSI model				
Layer		Protocol data unit (PDU)	Function ^[26]	
Host layers	7	Application	Data	High-level protocols such as for resource sharing or remote file access, e.g. HTTP.
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

Further reading about layer architecture: https://en.wikipedia.org/wiki/OSI_model

IP (INTERNET PROTOCOL) ADDRESSES

- What is an IP address?
- Did you use one before?

IP (INTERNET PROTOCOL) ADDRESSES

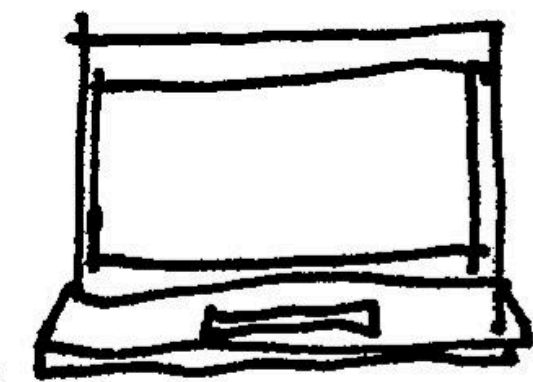
- What is an IP address?
- Did you use one before?

173.194.40.42

IP (INTERNET PROTOCOL) ADDRESSES

- Every internet connection has an IPv4 adress.
Example: 173.194.40.40
- Only 4'294'967'296 addresses world wide!
- Gradual transition of IPv4 to IPv6 addresses:
2001:0db8:0000:0042:0000:8a2e:0370:7334
- About 3.403×10^{38} IPv6 addresses world wide.
- An IP address is used to identify a device on the internet.

IP (INTERNET PROTOCOL) ADDRESSES

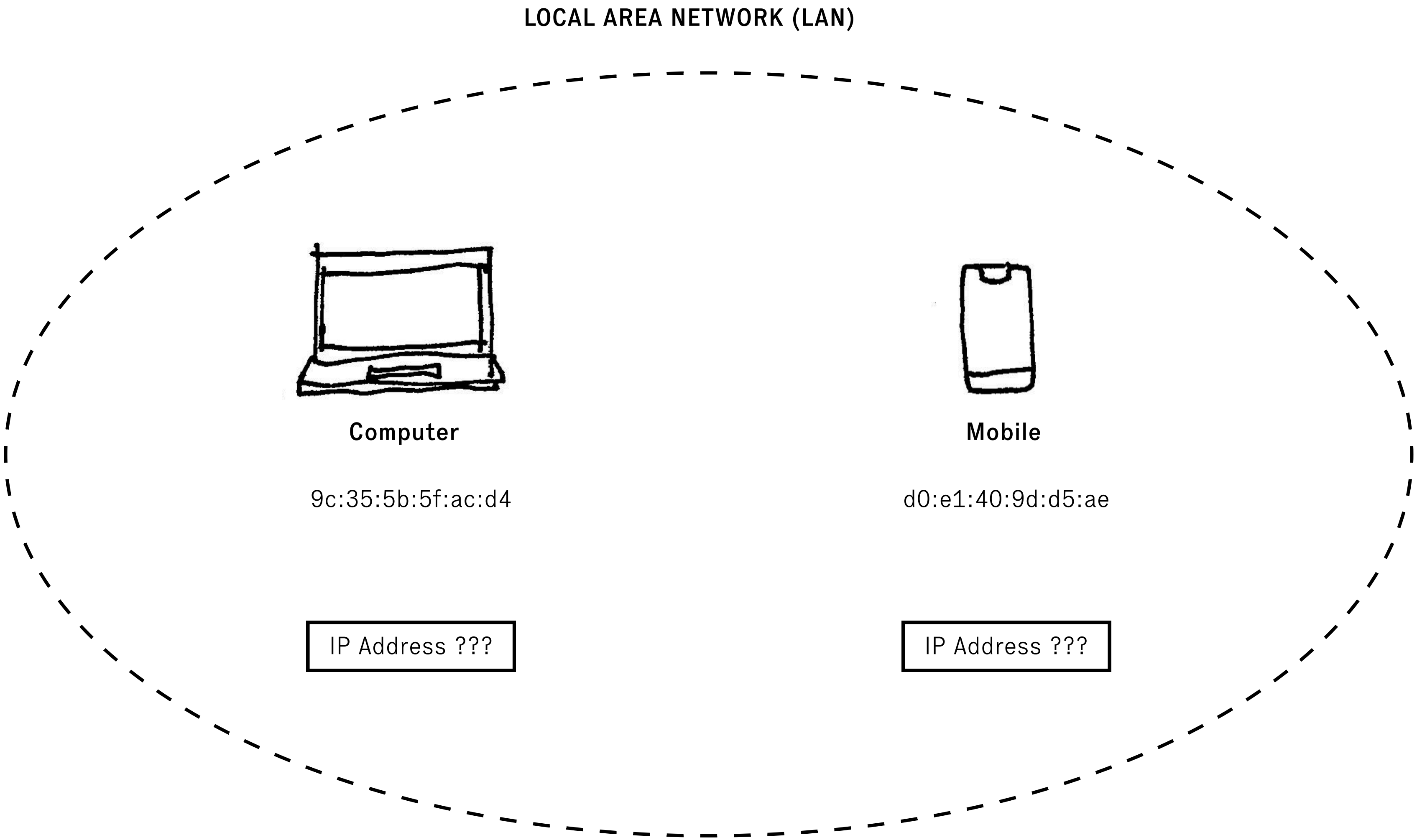


Computer

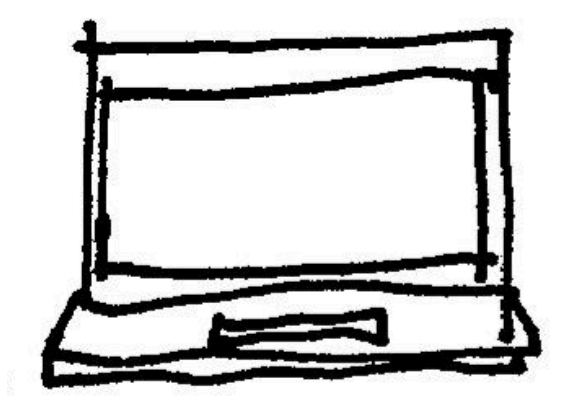
9c:35:5b:5f:ac:d4

IP Address ???

IP (INTERNET PROTOCOL) ADDRESSES



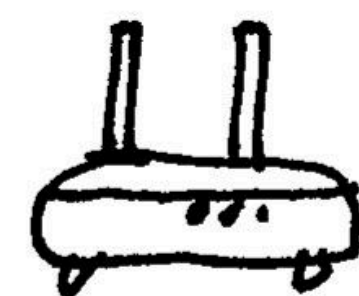
IP (INTERNET PROTOCOL) ADDRESSES



Computer

9c:35:5b:5f:ac:d4

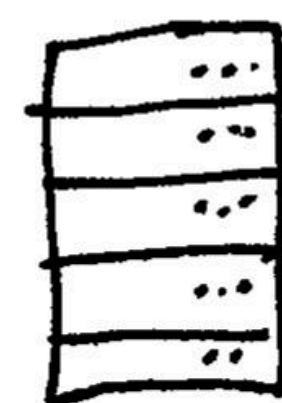
IP Address ???



Router/Modem

r8:00:28:02:cf:e4

Public IP: 66.234.164.22



Server (e.g. Netflix)

r8:00:28:02:cf:e4

54.57.125.166

RESERVED IP-ADDRESSES

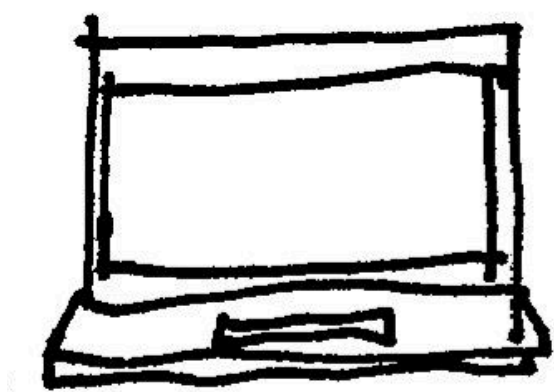
- Certain ranges are reserved for private networks.
(e.g. Local Area Networks (LAN))

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

IP (INTERNET PROTOCOL) ADDRESSES

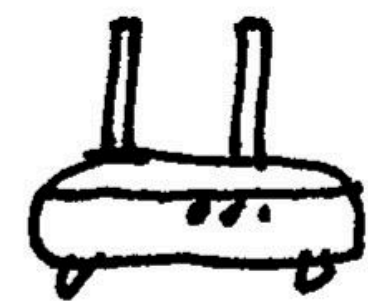


Computer

9c:35:5b:5f:ac:d4

Public IP: 66.234.164.22

Local IP: **192.168.1.23**

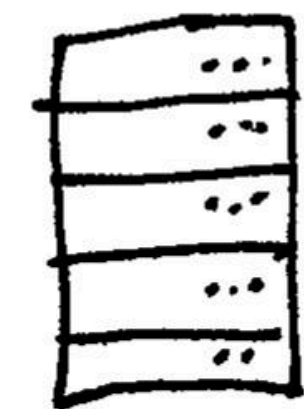


Router/Modem

r8:00:28:02:cf:e4

Public IP: 66.234.164.22

Local IP: **192.168.1.1**

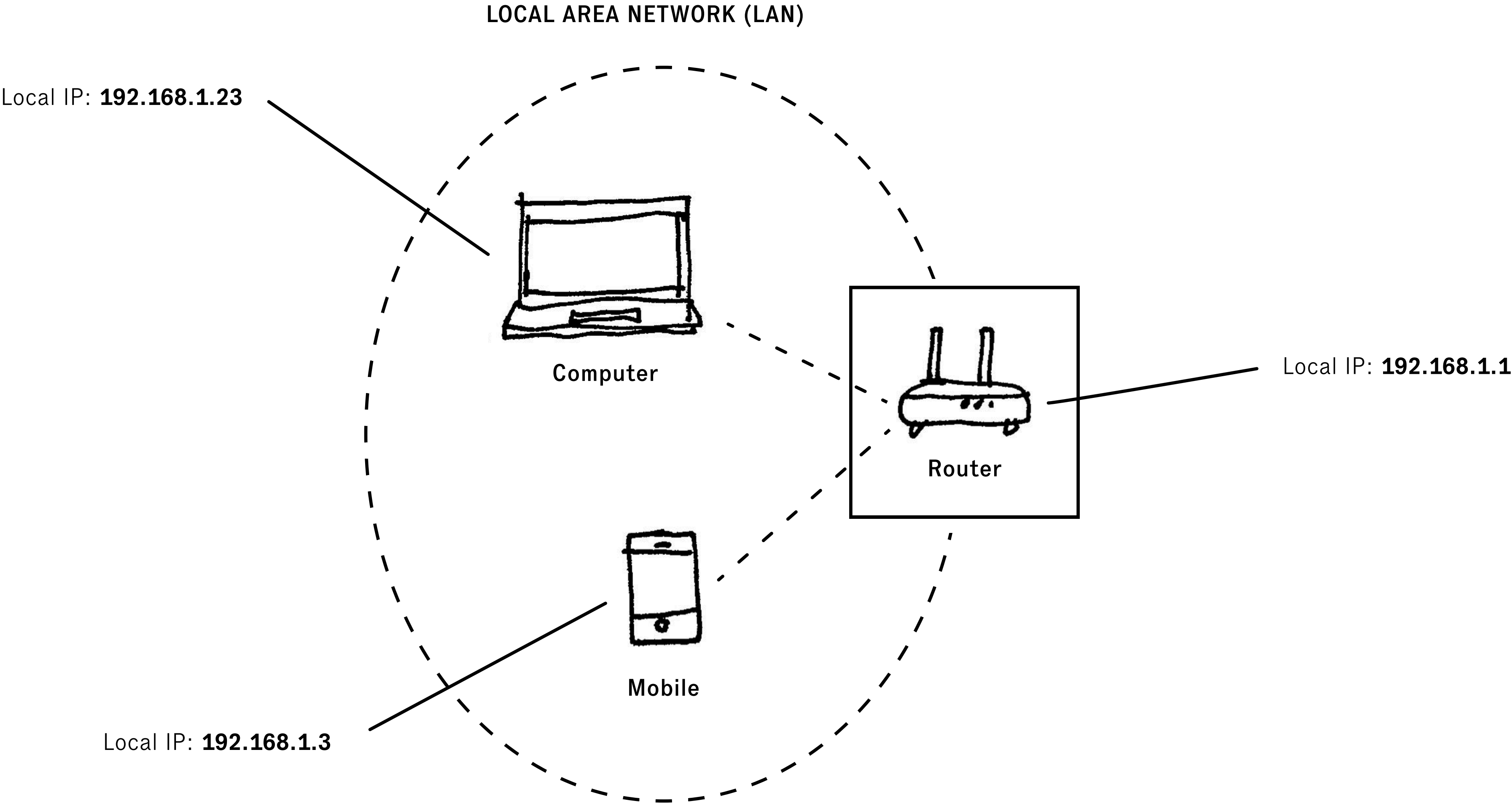


Server (e.g. Netflix)

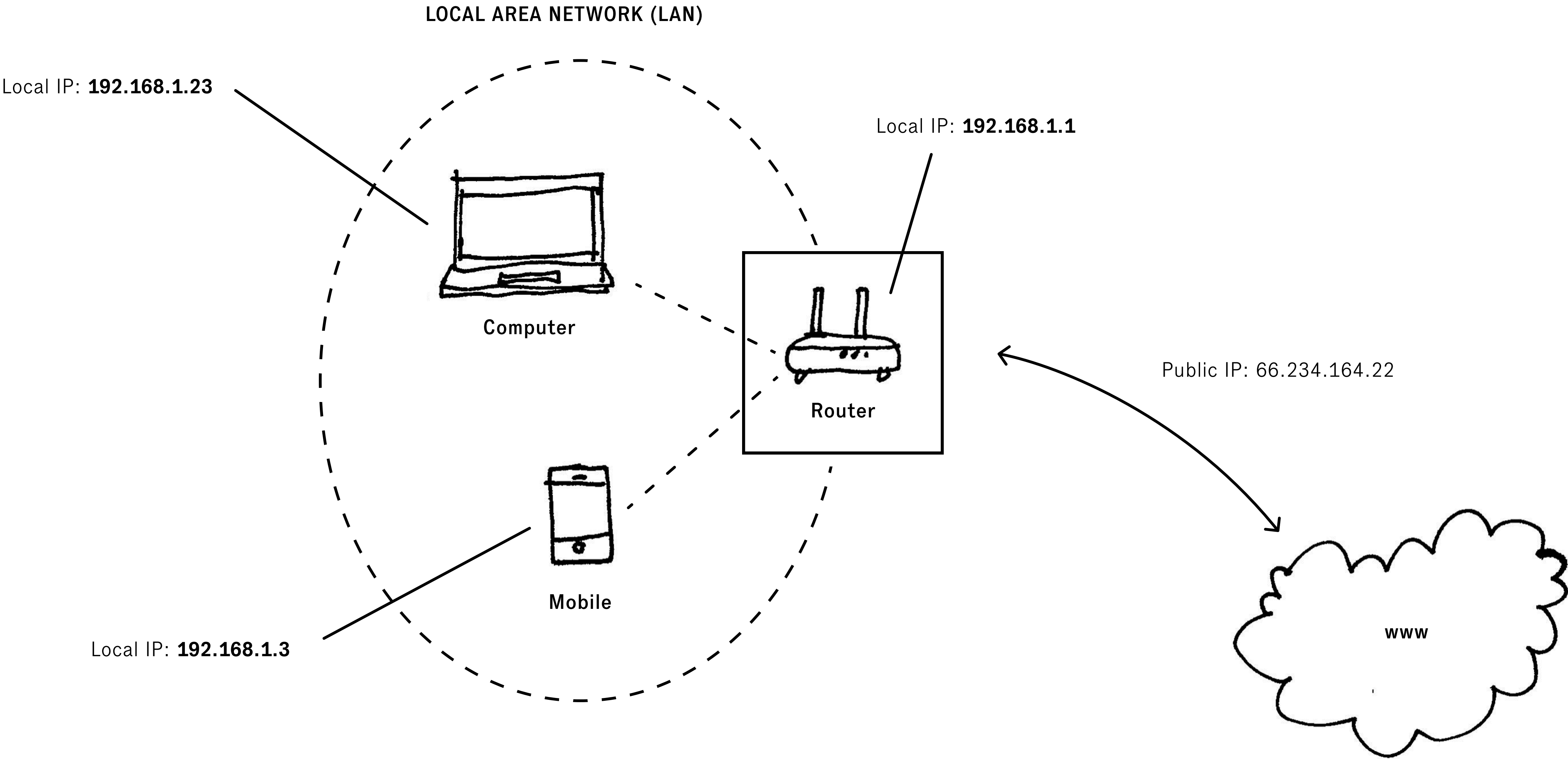
r8:00:28:02:cf:e4

54.57.125.166

LOCAL NETWORK



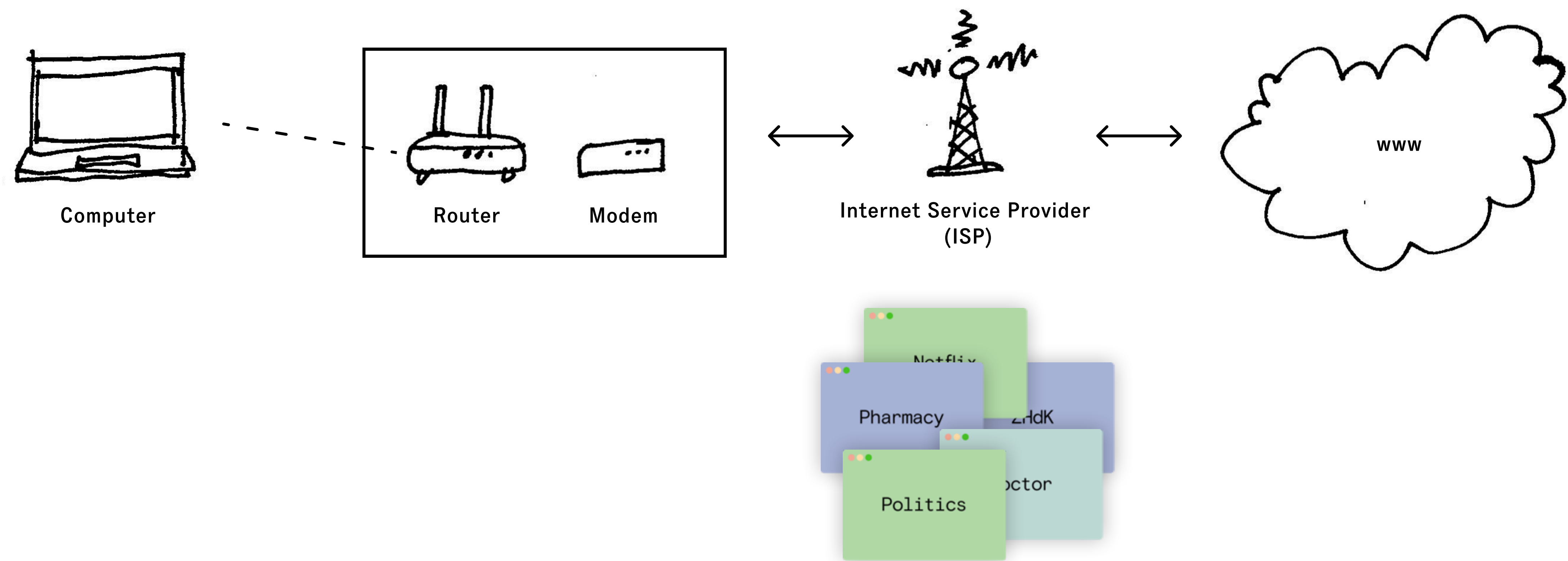
LOCAL NETWORK



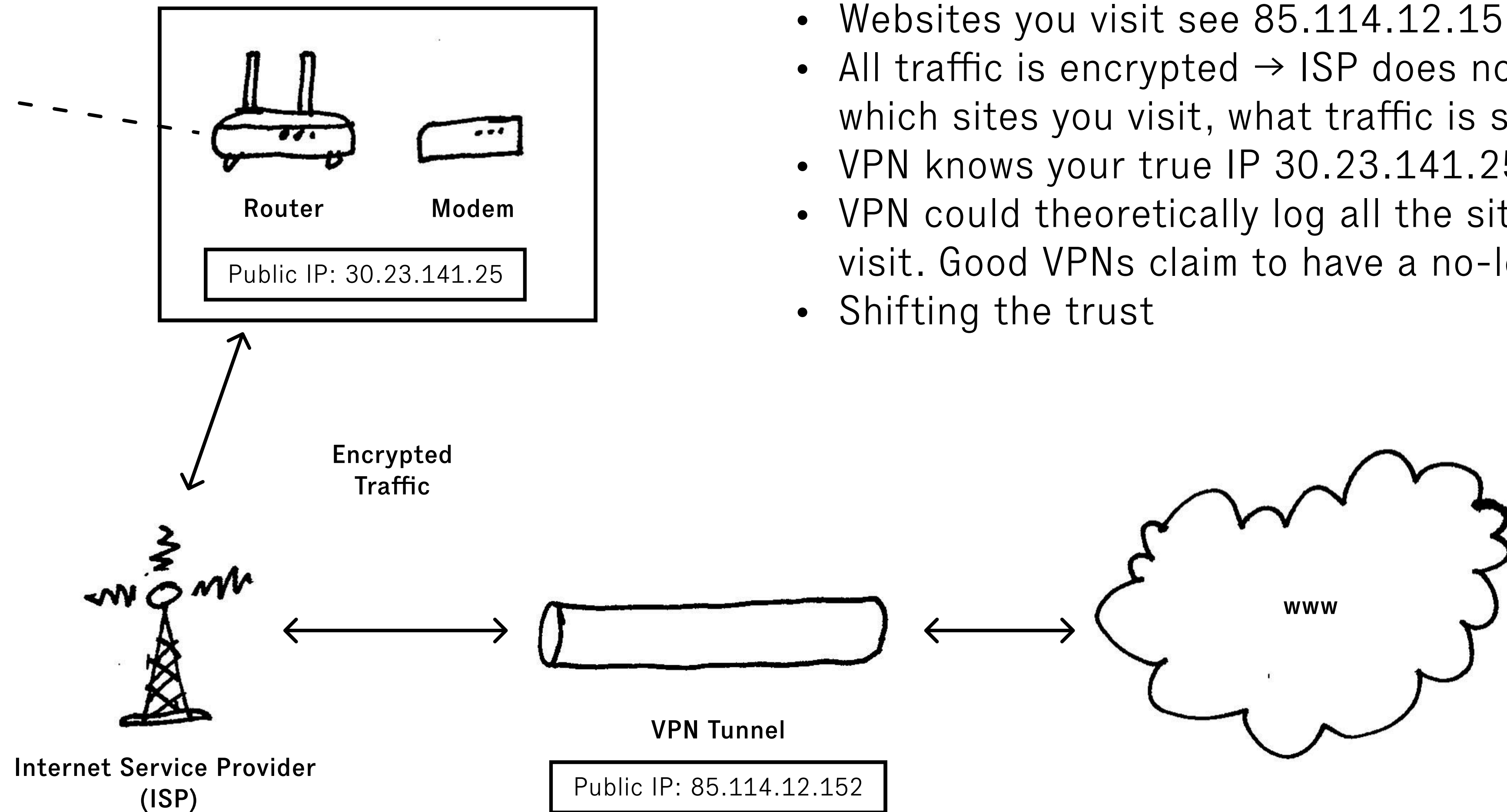
PING

- Switzerland: 66.234.146.88
- Angola: 74.118.126.9
- Japan: 194.114.136.218
- New Zealand: 62.93.164.71

PRIVACY



PRIVACY WITH VPN SERVICE



- Websites you visit see 85.114.12.152 as visitor
- All traffic is encrypted → ISP does not know which sites you visit, what traffic is sent etc.
- VPN knows your true IP 30.23.141.25
- VPN could theoretically log all the sites you visit. Good VPNs claim to have a no-log policy.
- Shifting the trust

Webtechnologies

127.0.0.1

LOCALHOST

- Your computer has a local address
`127.0.0.1` (=localhost)
- When building a webpage,
you often run a local server →
`http://localhost:3000`,
`http://localhost:5555`,
`http://localhost:8080` etc.



r/vibecoding



• 11 hr. ago



claude code is fucking insane

i know literally NOTHING about coding. ZERO. and i just built a fully functioning web app in minutes <http://localhost:3000/> check it out



980



95



3



Share

Webtechnologies

WORLD WIDE WEB



Source:

https://www.youtube.com/watch?v=sSqZ_hJu9zA&t=17s&ab_channel=CERN

HTML (HYPERTEXT MARKUP LANGUAGE)

HTML BASICS



```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <title>My First Page</title>
6    </head>
7    <body>
8
9      <h1>Hello, World!</h1>
10
11    </body>
12  </html>
13
```


TAGS

Tag

```
<html>...</html>
```

Opening tag

```
<p data-attribute="content" class="paragraph">...</p>
```

Closing tag

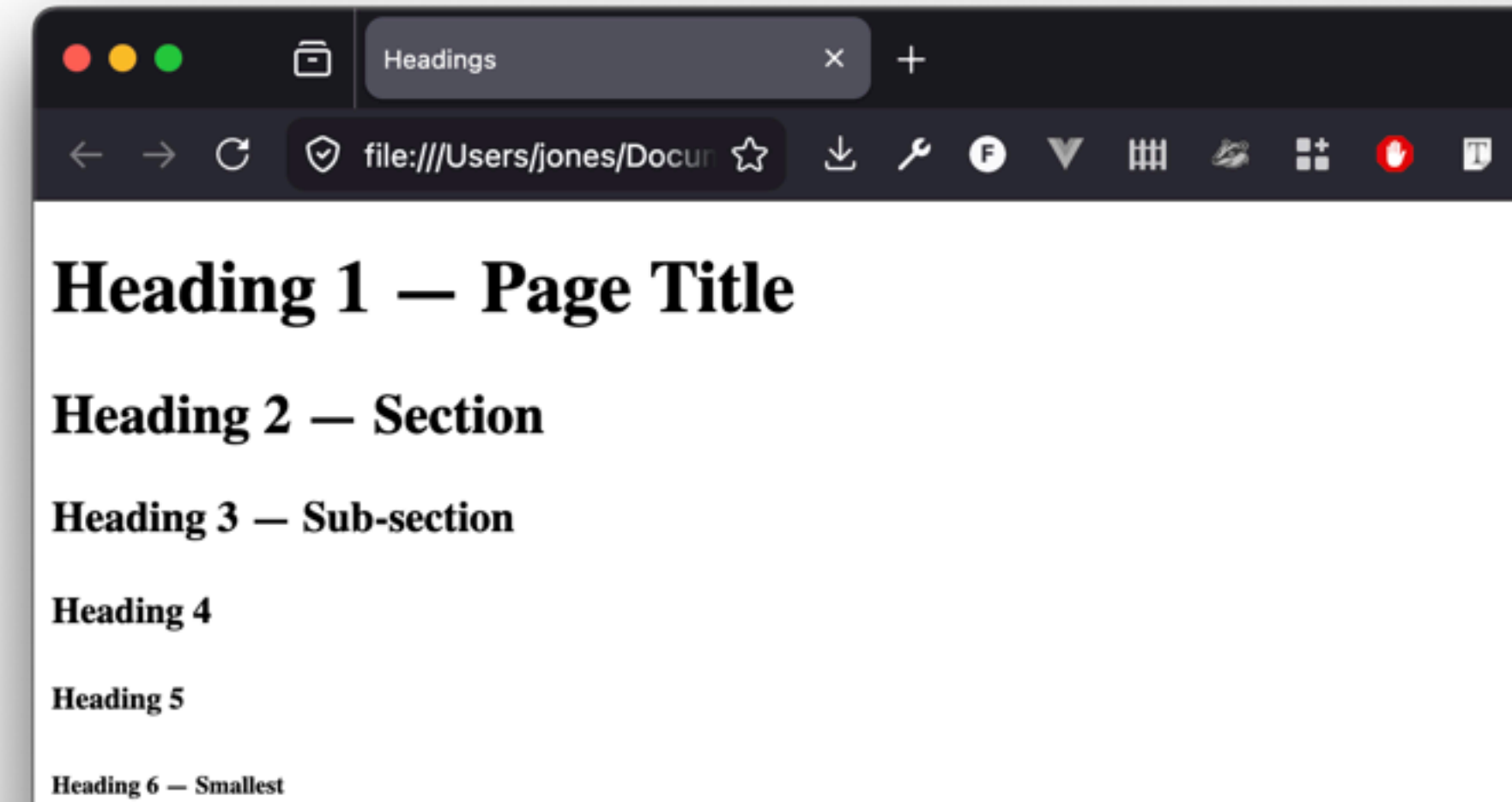
HTML BASICS



```
1  <body>
2
3  <!-- Opening + closing tag -->
4  <p>This is a paragraph.</p>
5
6  <!-- Self-closing tag (no content, no closing tag needed) -->
7  <br />
8  <hr />
9
10 </body>
```

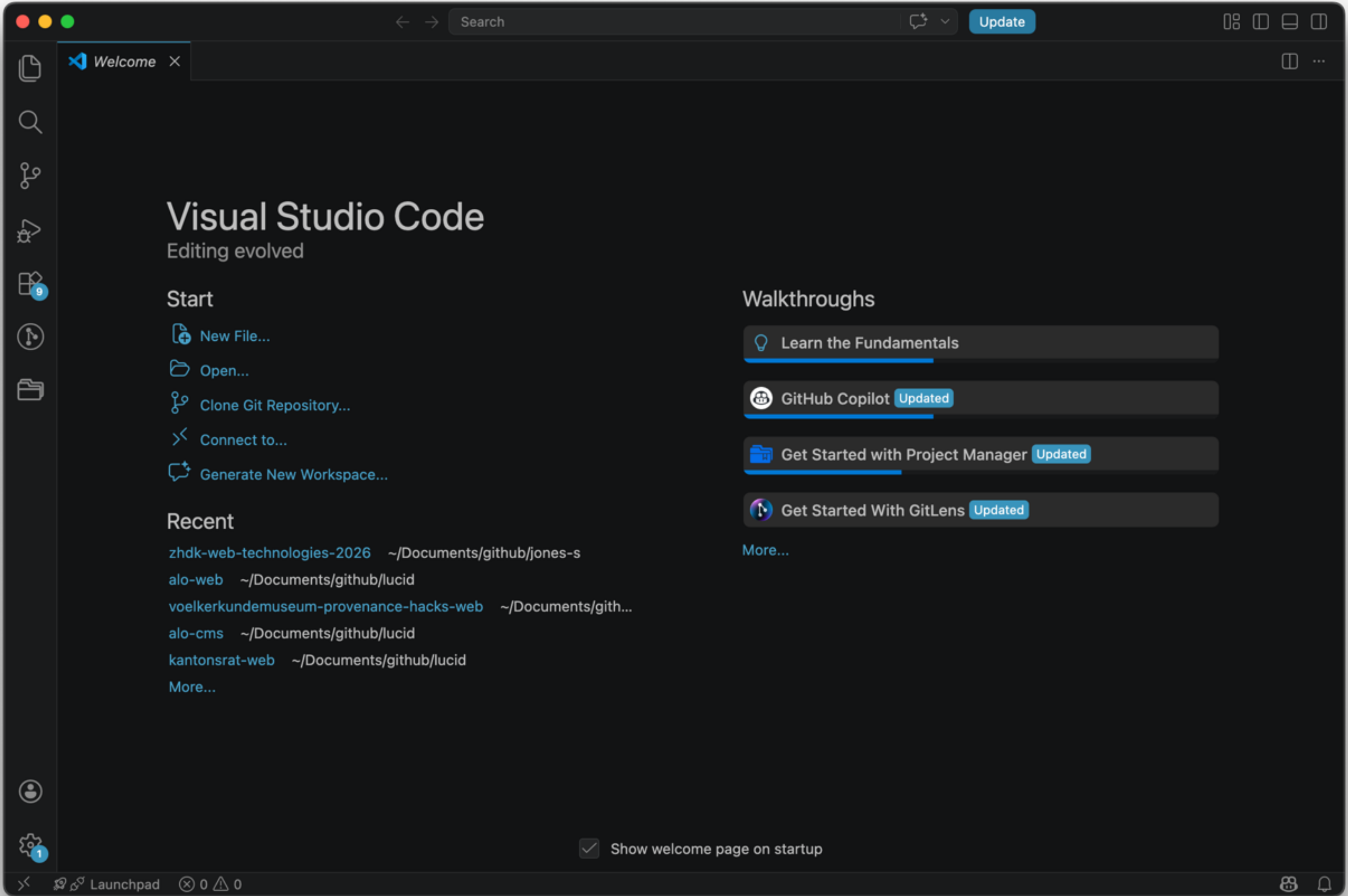

HTML BASICS

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Headings</title>
6    </head>
7    <body>
8
9      <h1>Heading 1 – Page Title</h1>
10     <h2>Heading 2 – Section</h2>
11     <h3>Heading 3 – Sub-section</h3>
12     <h4>Heading 4</h4>
13     <h5>Heading 5</h5>
14     <h6>Heading 6 – Smallest</h6>
15
16     <!-- Only one h1 per page. Use headings in order – don't skip levels. -->
17
18   </body>
19 </html>
20
```



PREREQUISITES

VISUAL STUDIO CODE



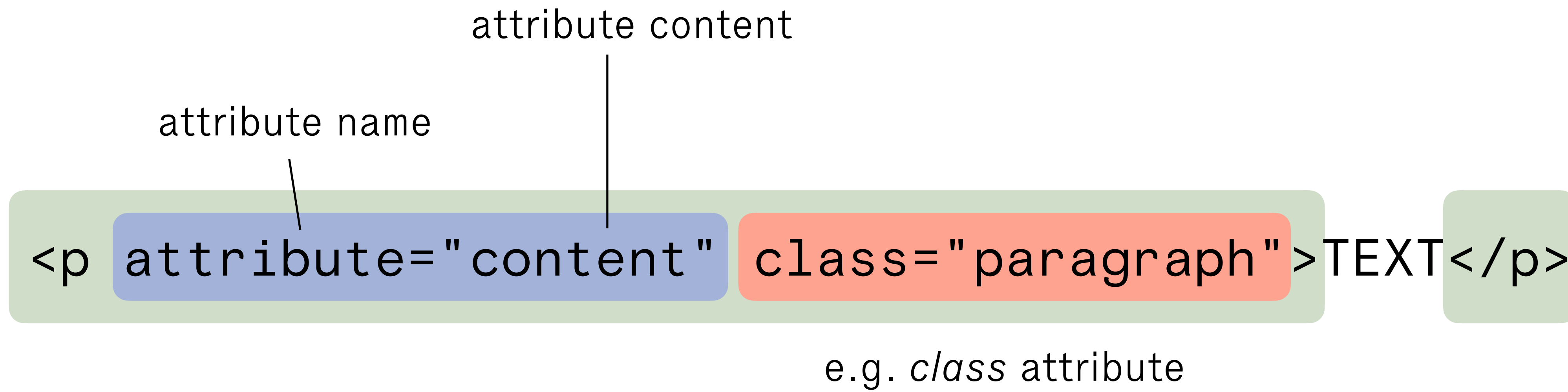
EXERCISE 1

- Mark the following text as a valid HTML page. The browsertab should show «Tomato Soup», and the hierarchy of the text should be clear.
- <https://short.jonasscheiwiller.ch/ex1>



```
1 Classic Tomato Soup
2
3 A warming soup for cold days. Ready in 30 minutes.
4
5 Ingredients
6 Olive oil
7 1 onion, chopped
8 2 cloves garlic
9 800g canned tomatoes
10 500ml vegetable stock
11 Salt and pepper
12
13 Steps
14 Heat olive oil in a large pot.
15 Add onion and cook for 5 minutes.
16 Add garlic and cook for 1 more minute.
17 Pour in tomatoes and stock.
18 Simmer for 20 minutes.
19 Blend until smooth.
20 Season with salt and pepper and serve.
```


ATTRIBUTES



HTML BASICS — IMG

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8" />
6    <title>Images</title>
7  </head>
8
9  <body>
10
11    <!-- src – where the image lives (URL or relative path) -->
12    <!-- alt – shown if image fails to load, required for accessibility -->
13    
14
15
16  </body>
17
18  </html>
```


EXERCISE 2

- Add an image to your recipe HTML. Add it above the recipe title. (Any tomato soup image)
- Add a line under the title: "Original recipe from" with a link to any cooking website.



```
1 Classic Tomato Soup
2
3 A warming soup for cold days. Ready in 30 minutes.
4
5 Ingredients
6 Olive oil
7 1 onion, chopped
8 2 cloves garlic
9 800g canned tomatoes
10 500ml vegetable stock
11 Salt and pepper
12
13 Steps
14 Heat olive oil in a large pot.
15 Add onion and cook for 5 minutes.
16 Add garlic and cook for 1 more minute.
17 Pour in tomatoes and stock.
18 Simmer for 20 minutes.
19 Blend until smooth.
20 Season with salt and pepper and serve.
```

PATHS

- When linking a file, for example an ``, we can use absolute or relative paths.

ABSOLUTE URLS

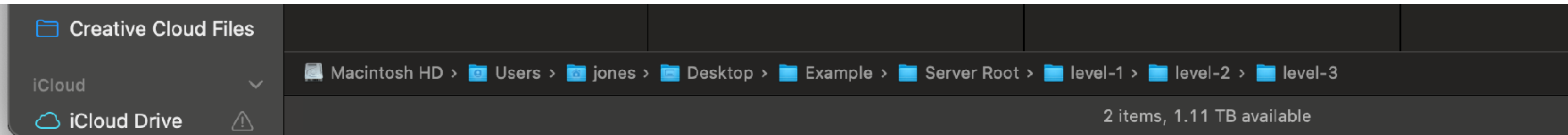


```
1  
```


ABSOLUTE PATHS

- An absolute path always begins from the start of your hard drive/server etc.
- It starts with a slash
- e.g.:

“/Users/jones/Desktop/Example/Server\ Root/level-1/level-2/level-3”



RELATIVE PATHS

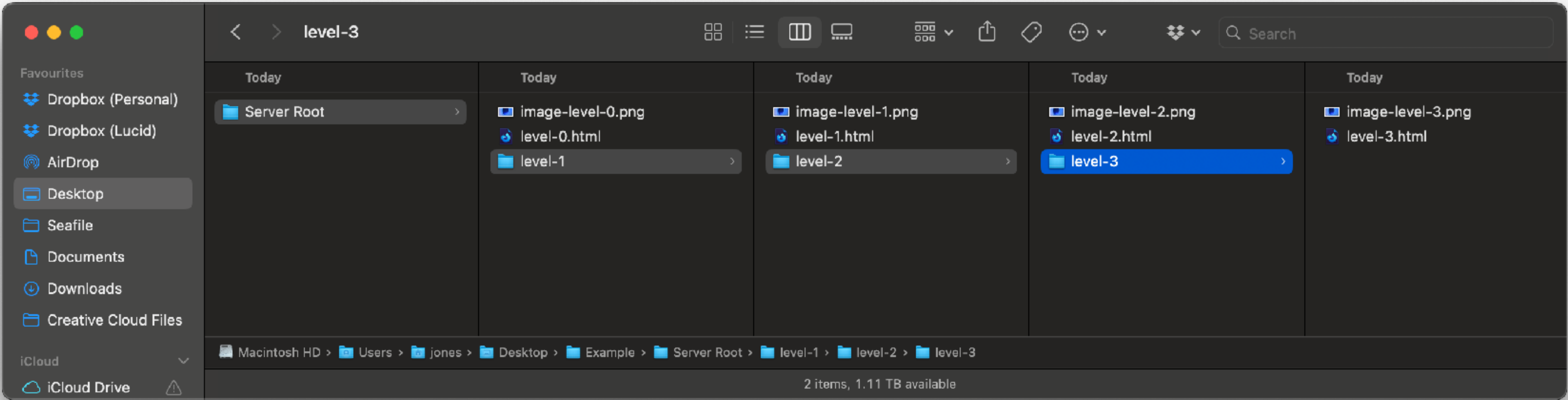
- `./` is the current directory
- `../` is the parent directory



```
1 <!-- Local image -->
2 
```


RELATIVE PATHS

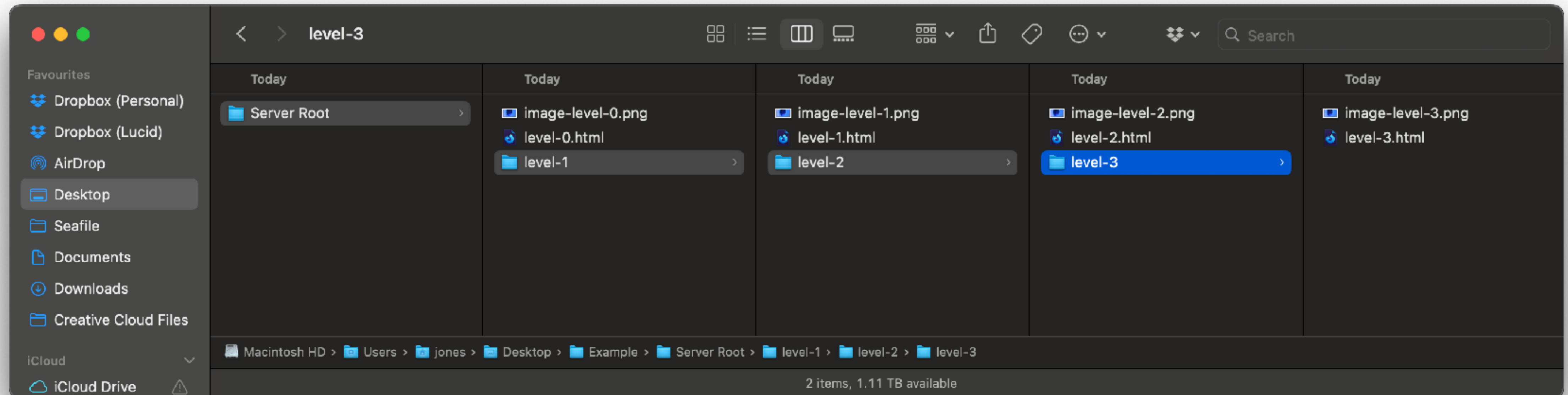
- How can I link in file *level-0.html* → to *image-level-3.png*?



RELATIVE PATHS

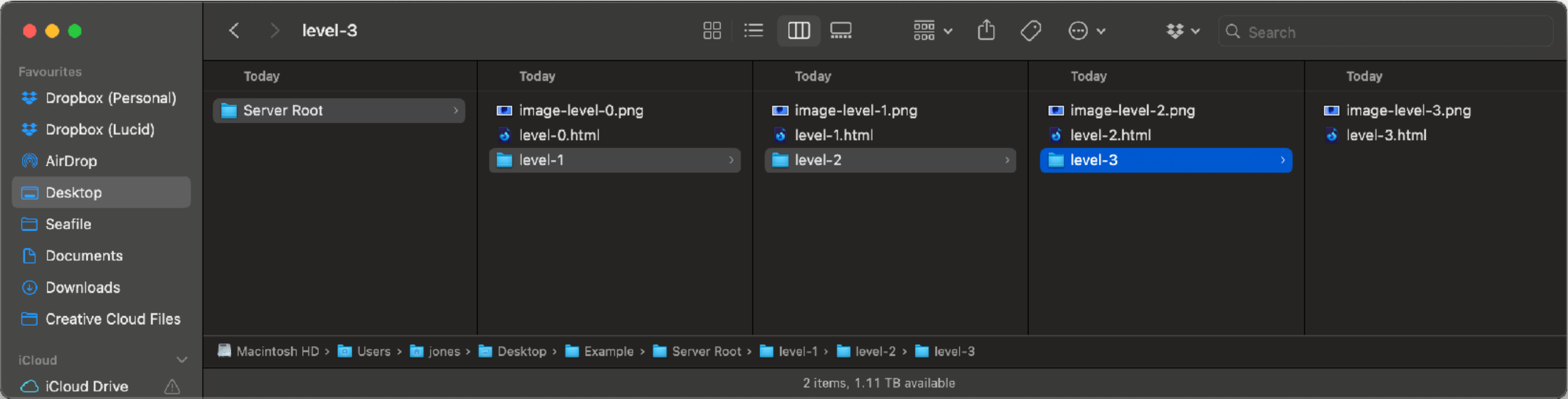
- How can I link in file *level-0.html* → to *image-level-3.png*?

```
src=" ./level-1/level-2/level-3/image-level-3.png"
```



RELATIVE PATHS

- How can I link the file *level-2.html* → Link to *image-level-1.png*?



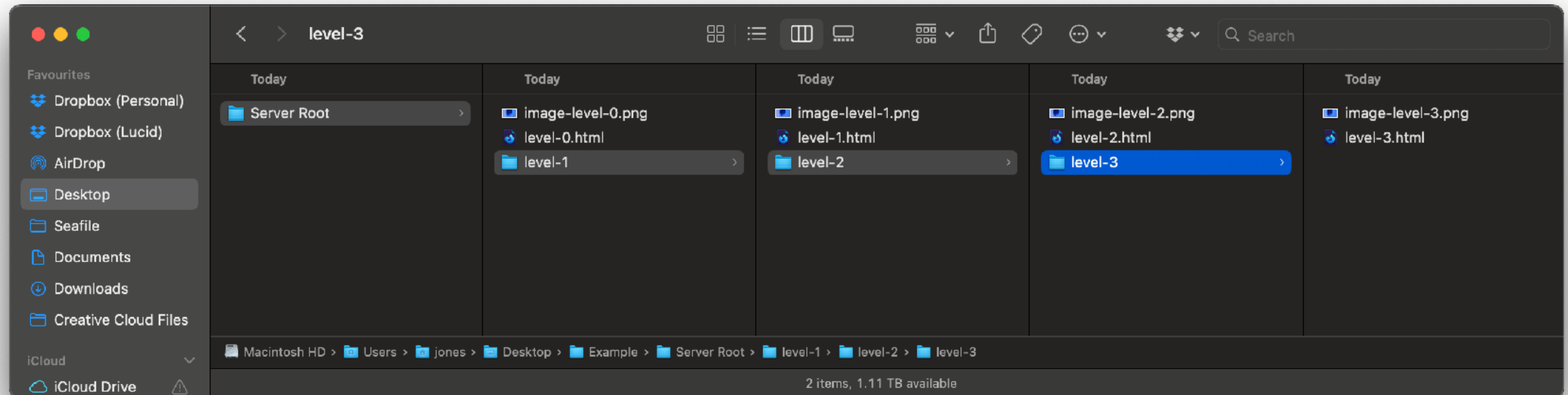
RELATIVE PATHS

- How can I link the file *level-2.html* → Link to *image-level-1.png*?

```
src="../../image-level-1.png"
```

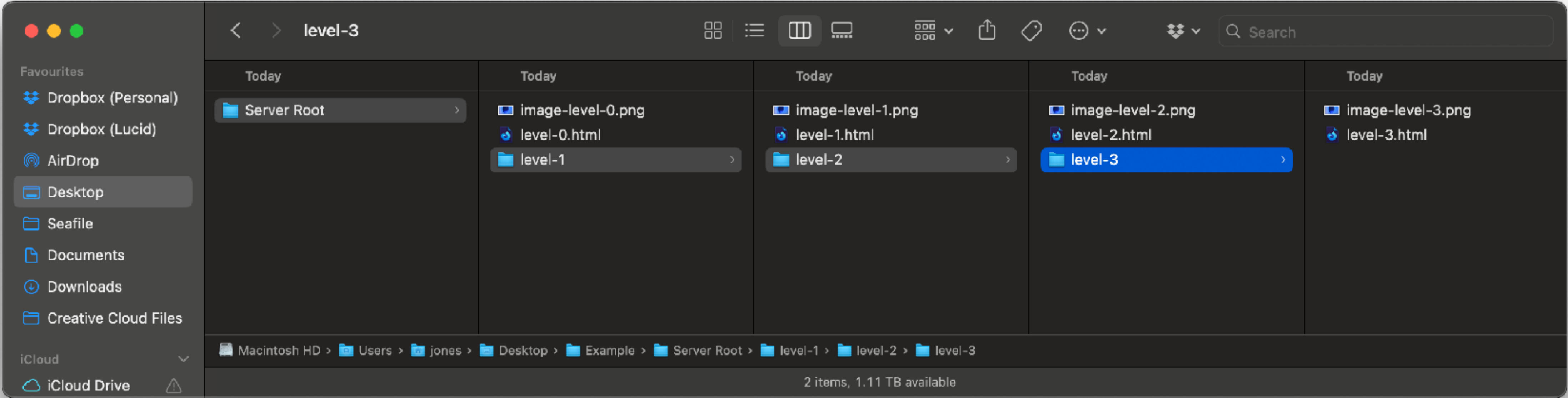
(possible, but useless, but only for demonstration)

```
src="../../../level-1/image-level-1.png"
```



RELATIVE PATHS

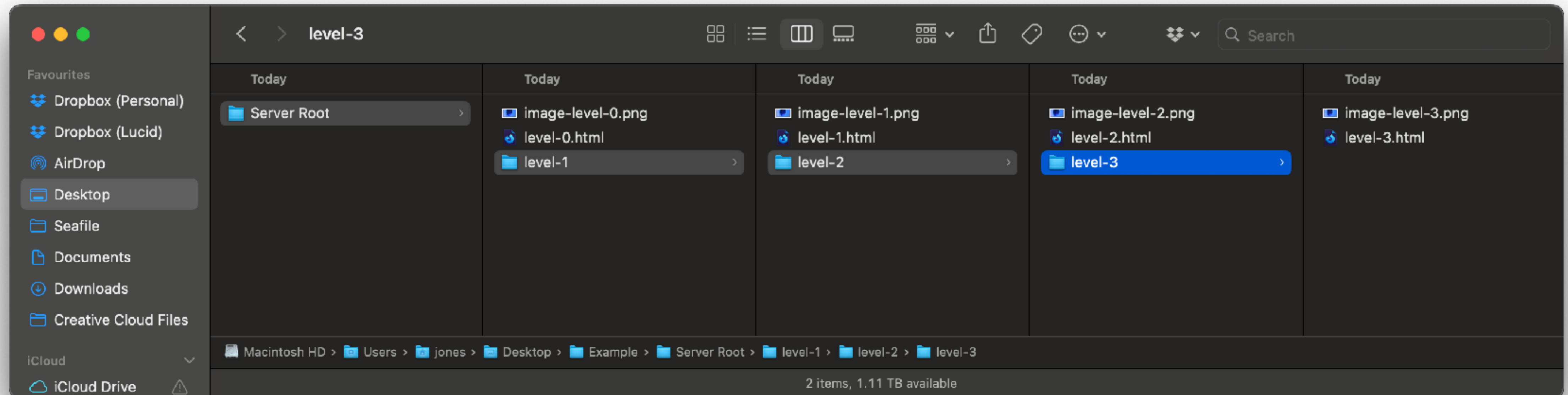
- How can I link the file *level-3.html* → Link to *image-level-1.png*?



RELATIVE PATHS

- How can I link the file *level-3.html* → Link to *image-level-1.png*?

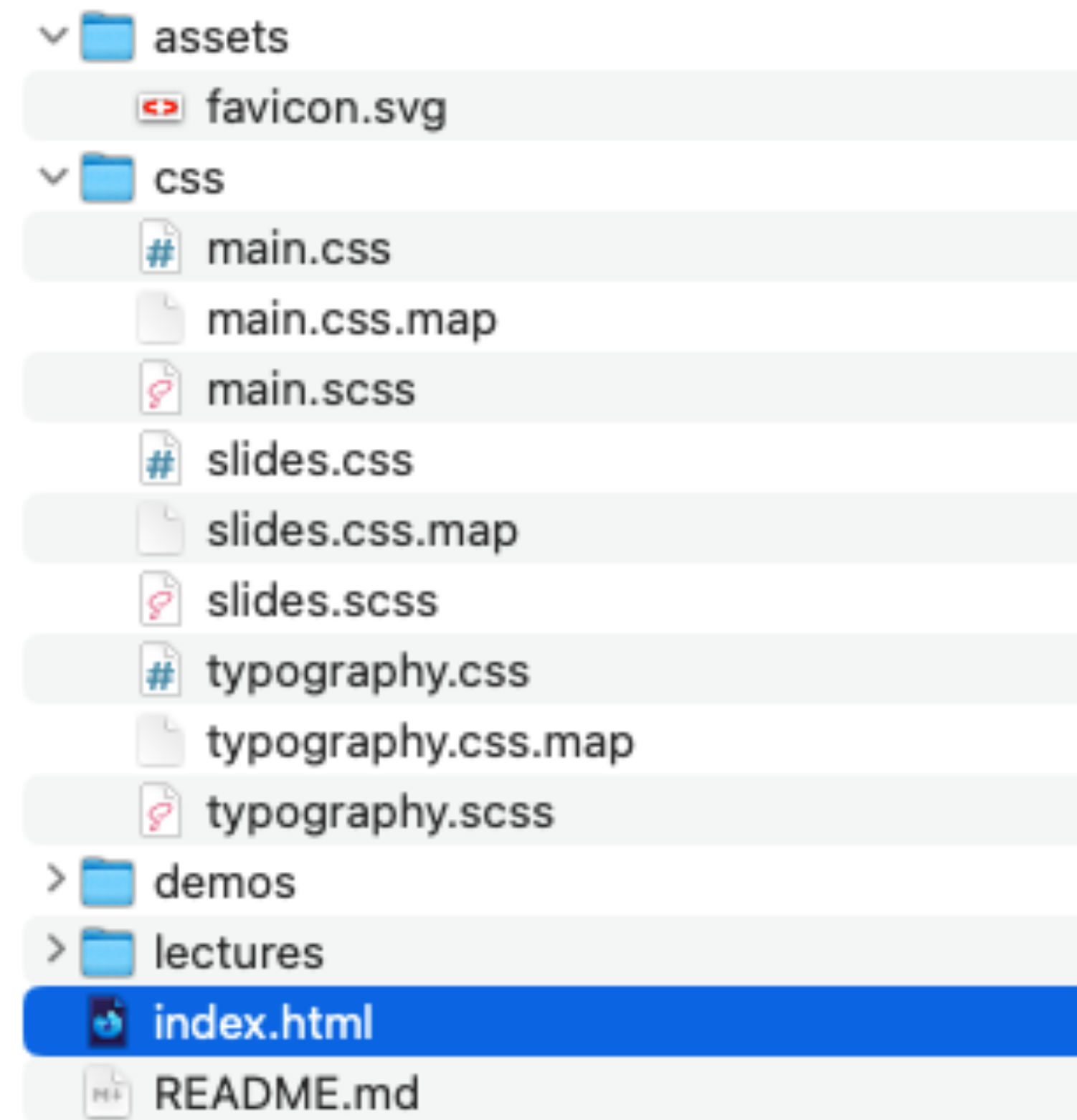
`src="../../image-level-1.png"`



WHAT IS A WEBSITE?

WHAT IS A WEBSITE?

- Collection of files interpreted by a browser
- If online, a website is accessible via the internet.



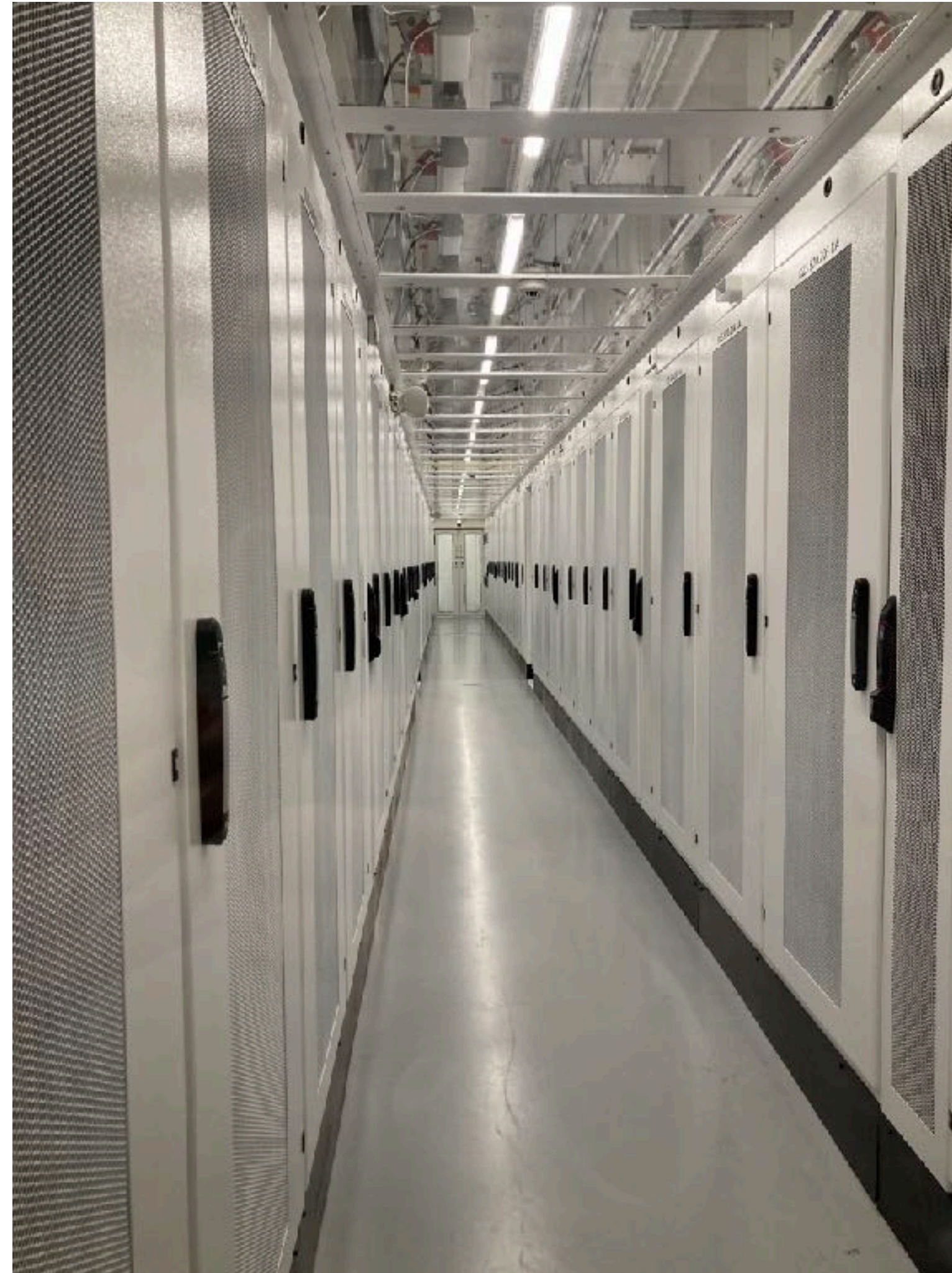
HOW IS A WEBSITE ONLINE?

- Files hosted on a server
- Server: Computers serving (web) documents via internet
- → Most websites live in a shelf in a data centre.

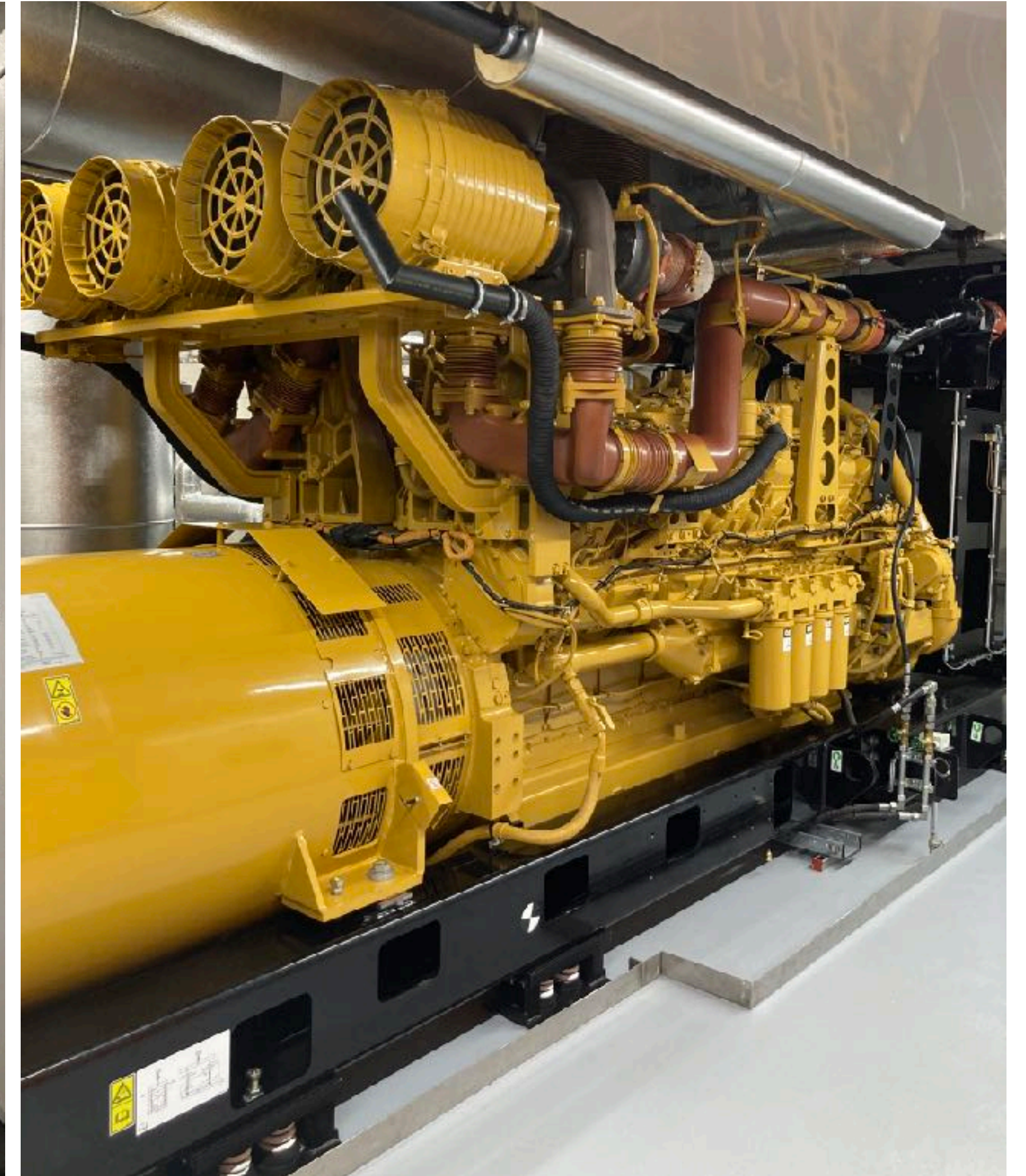


DATA CENTRES – ALWAYS ON

- 24/7, 99% uptime
- If power goes out → diesel generator or other sources of energy

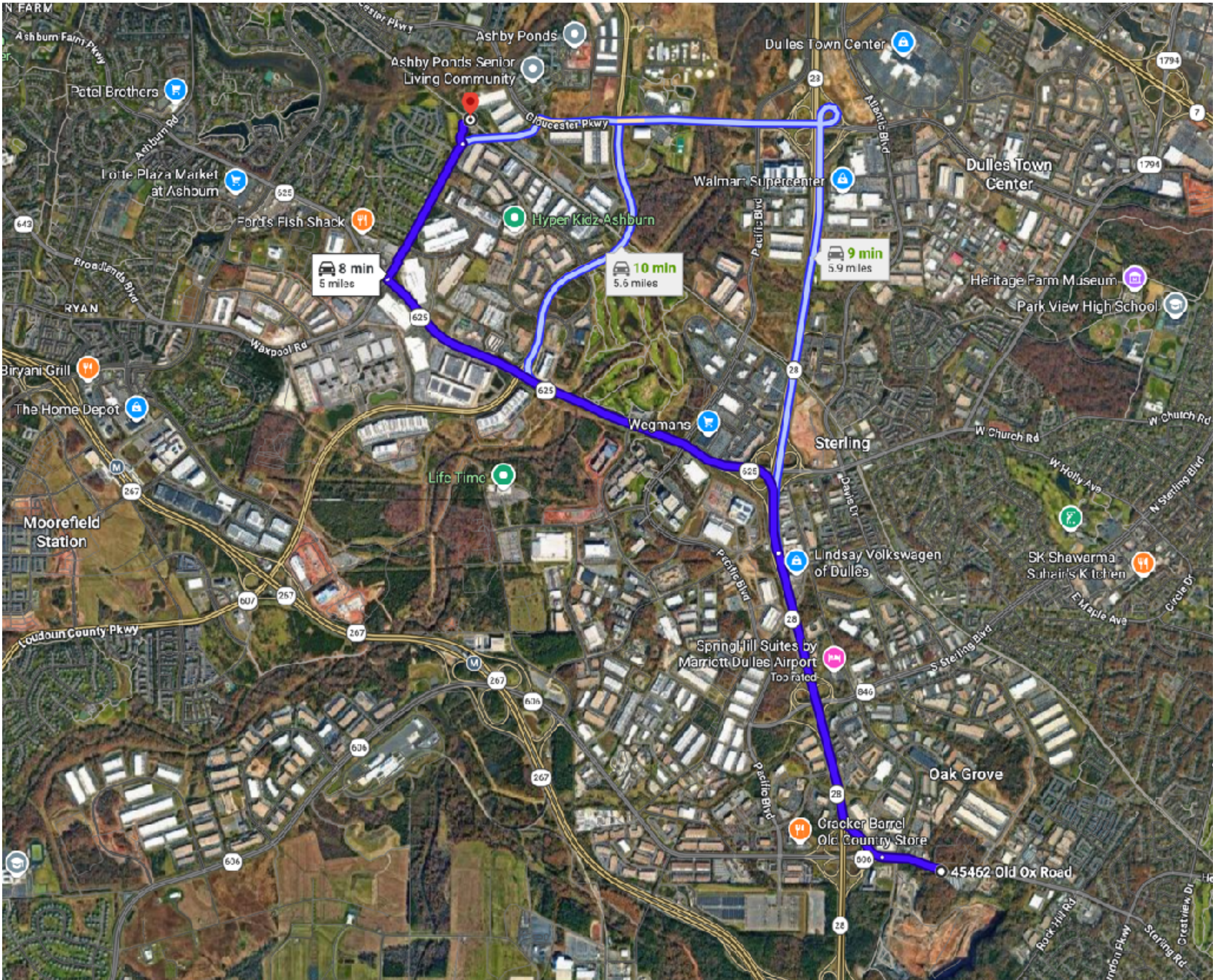


Lots of air or water needed for cooling



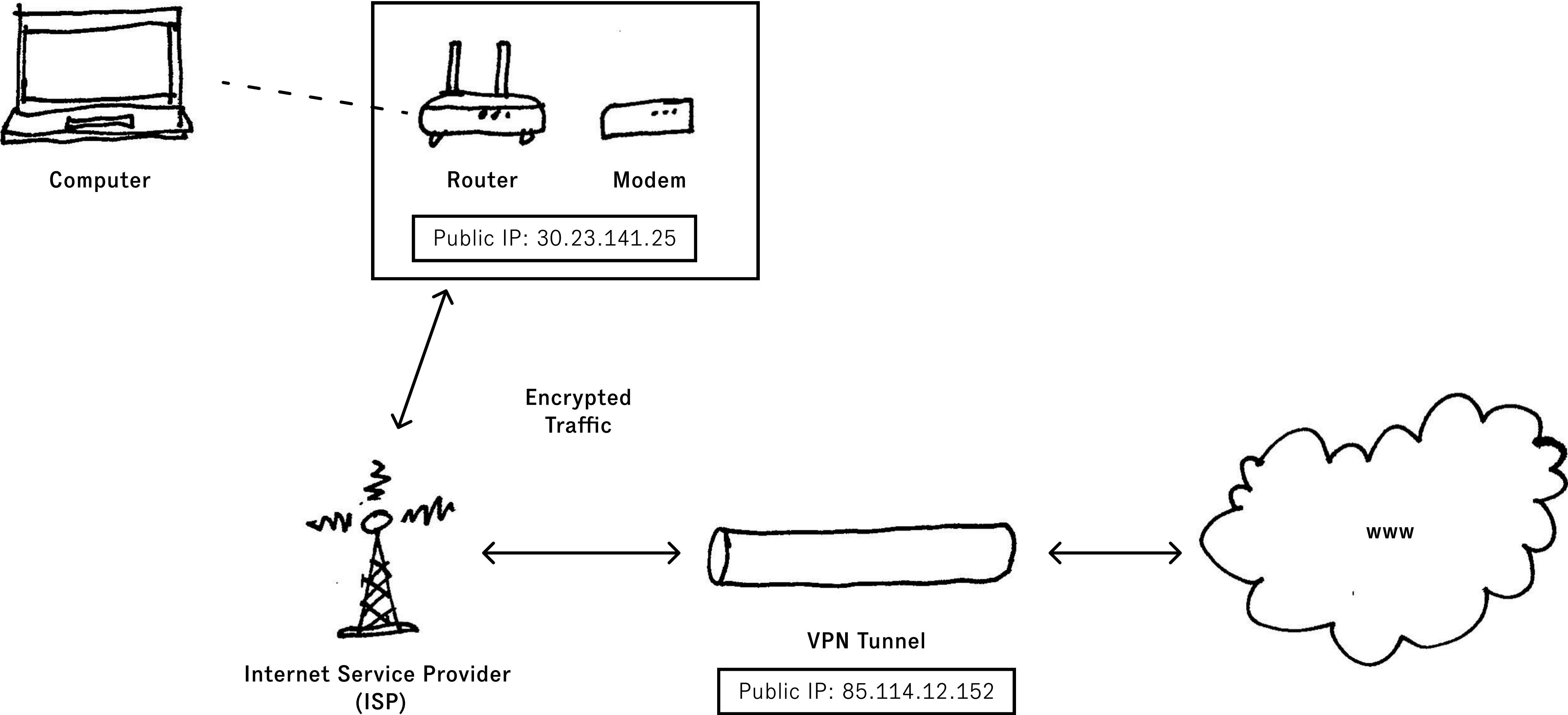
Tank of 60'000 litres of diesel, could run the (small) data centre for ~ 3 days.

DATA CENTRES – VIRGINIA

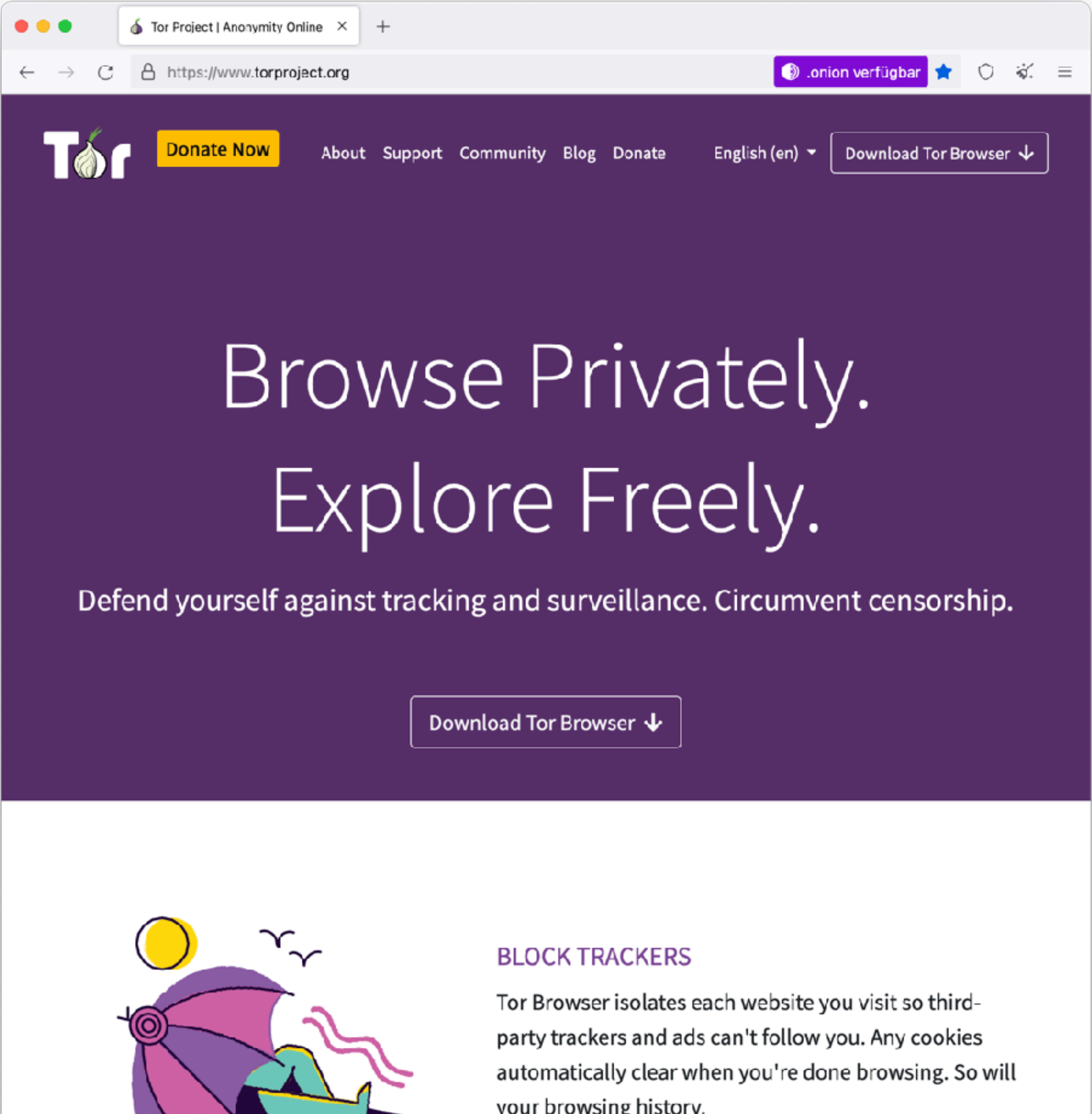


Virginia, where the cloud lives...

PRIVACY WITH VPN SERVICE

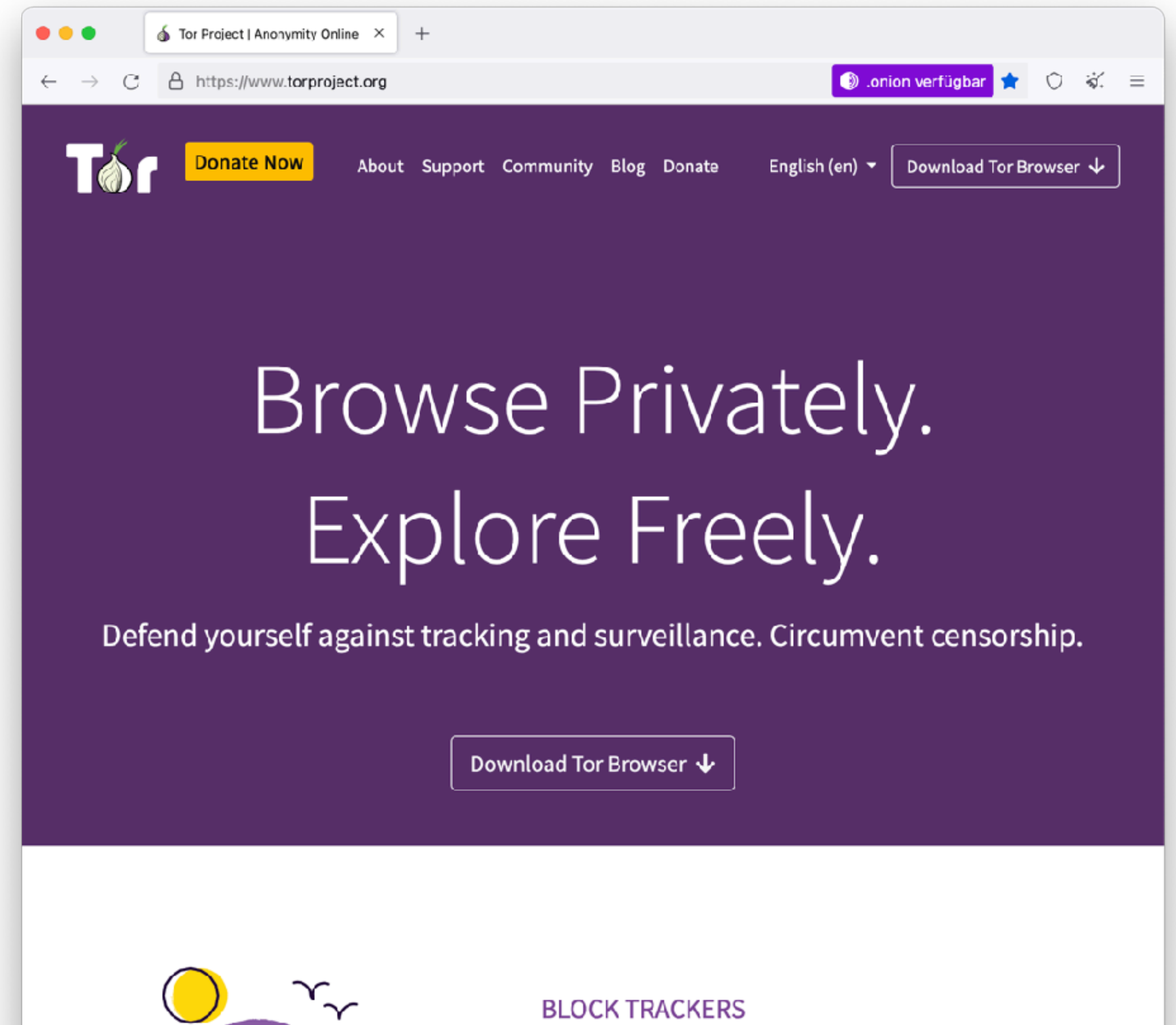


TOR BROWSER



TOR BROWSER (THE ONION ROUTER)

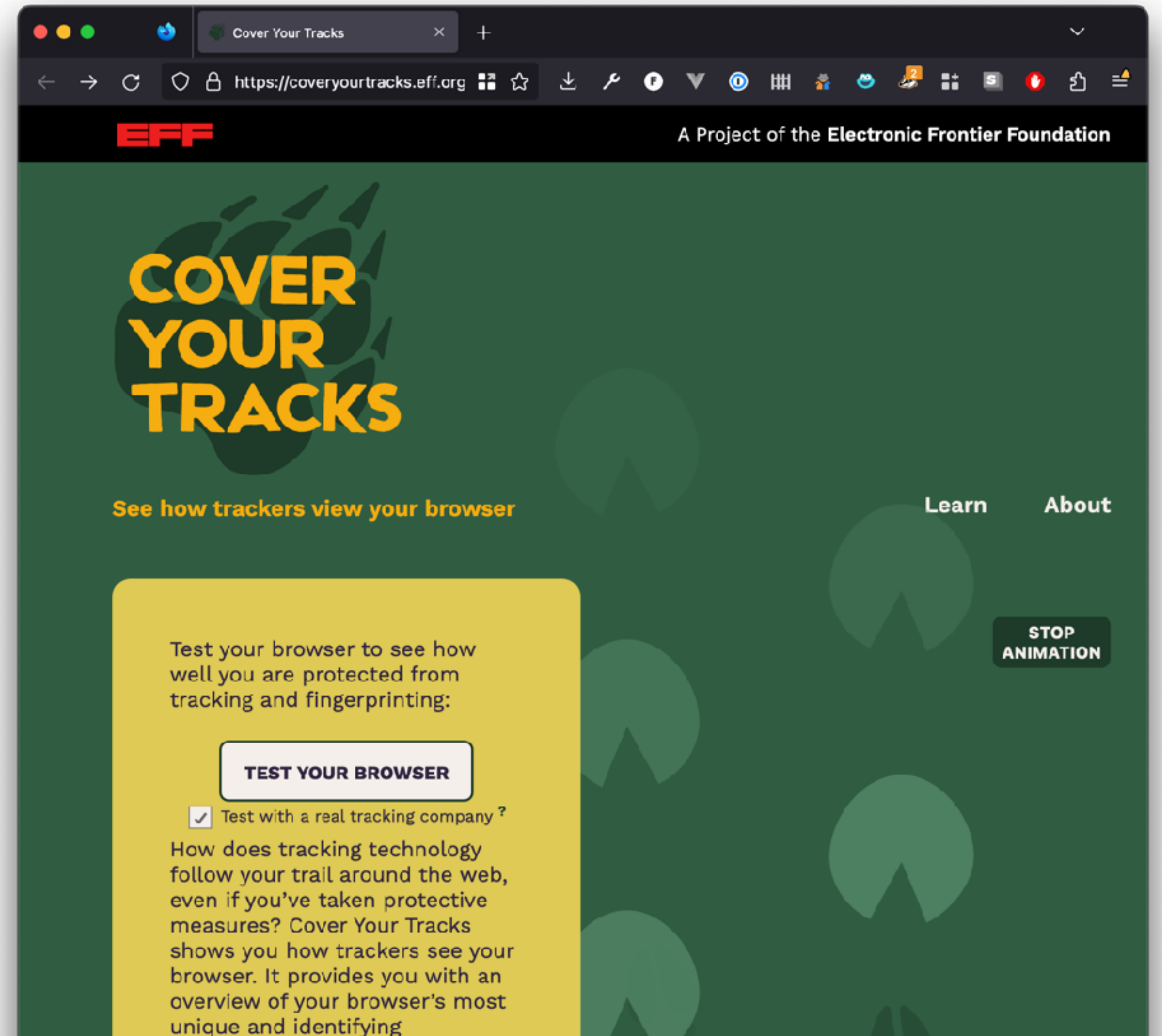
- Routes traffic through anonymous Tor network
- Allows access to .onion websites (darknet)
- Resist fingerprinting



WHAT IS FINGERPRINTING?

- A website can test a lot of your browser's features and its settings to create an almost unique ID for you.
- This allows to track your “fingerprint” across multiple sites.
- Especially with “free services” like Google Fonts, Facebook Like-buttons, Image-CDNs etc.

<https://coveryourtracks.eff.org/>



WHAT IS FINGERPRINTING?

- A website can test a lot of your browsers feature and its settings to create an almost unique id for you.
- This allows to track your “fingerprint” across multiple sites.
- Especially with “free services” like google fonts, Facebook Like-buttons, Image-CDNs etc.

Your Results

Your browser fingerprint **appears to be unique** among the 230,057 tested in the past 45 days.

Currently, we estimate that your browser has a fingerprint that conveys **at least 17.81 bits of identifying information.**

WEBGL VENDOR & RENDERER

Apple~Apple M1

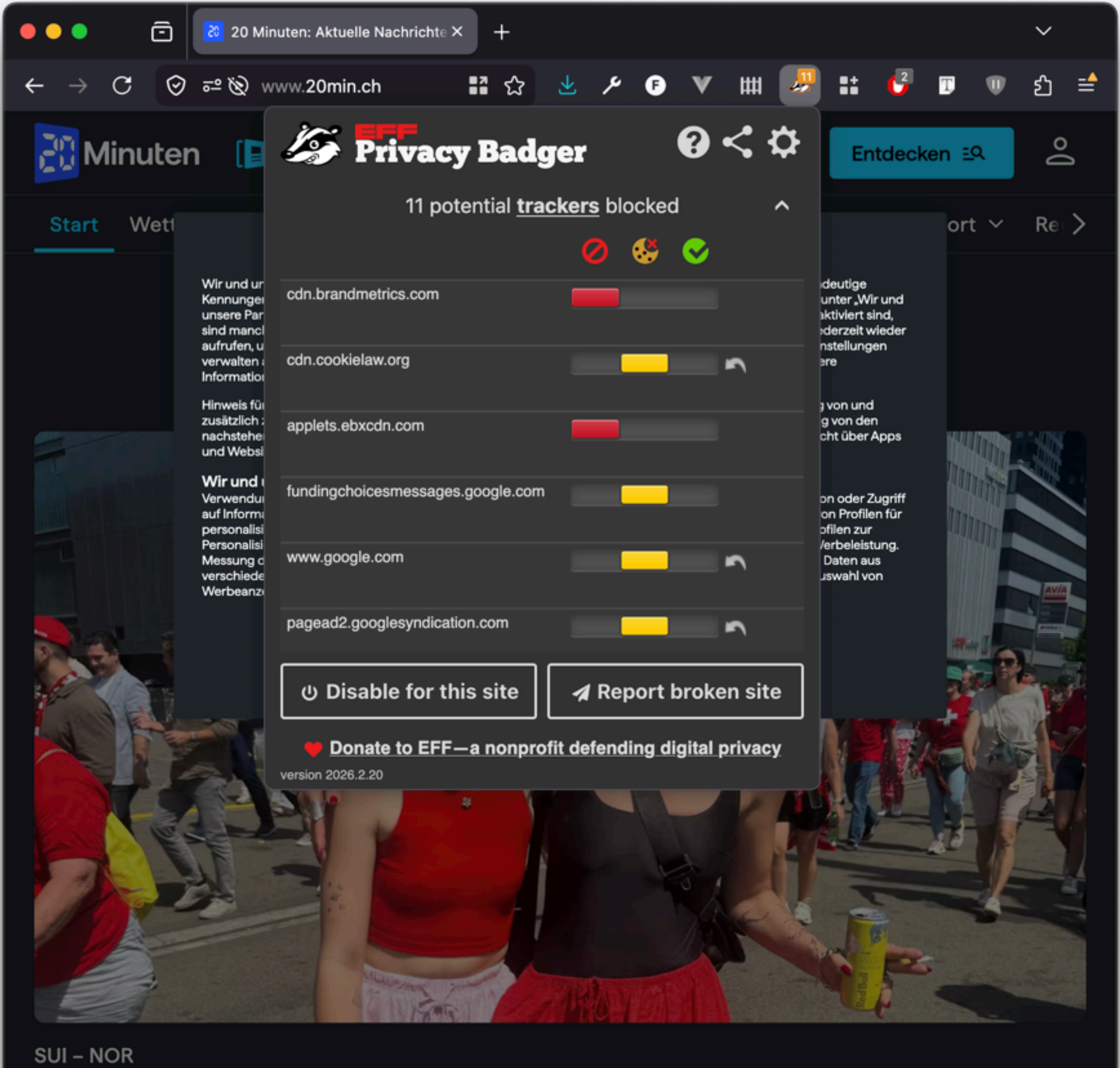
Bits of identifying information: 6.79
One in x browsers have this value: 110.45

USER AGENT

Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0)
Gecko/20100101 Firefox/112.0

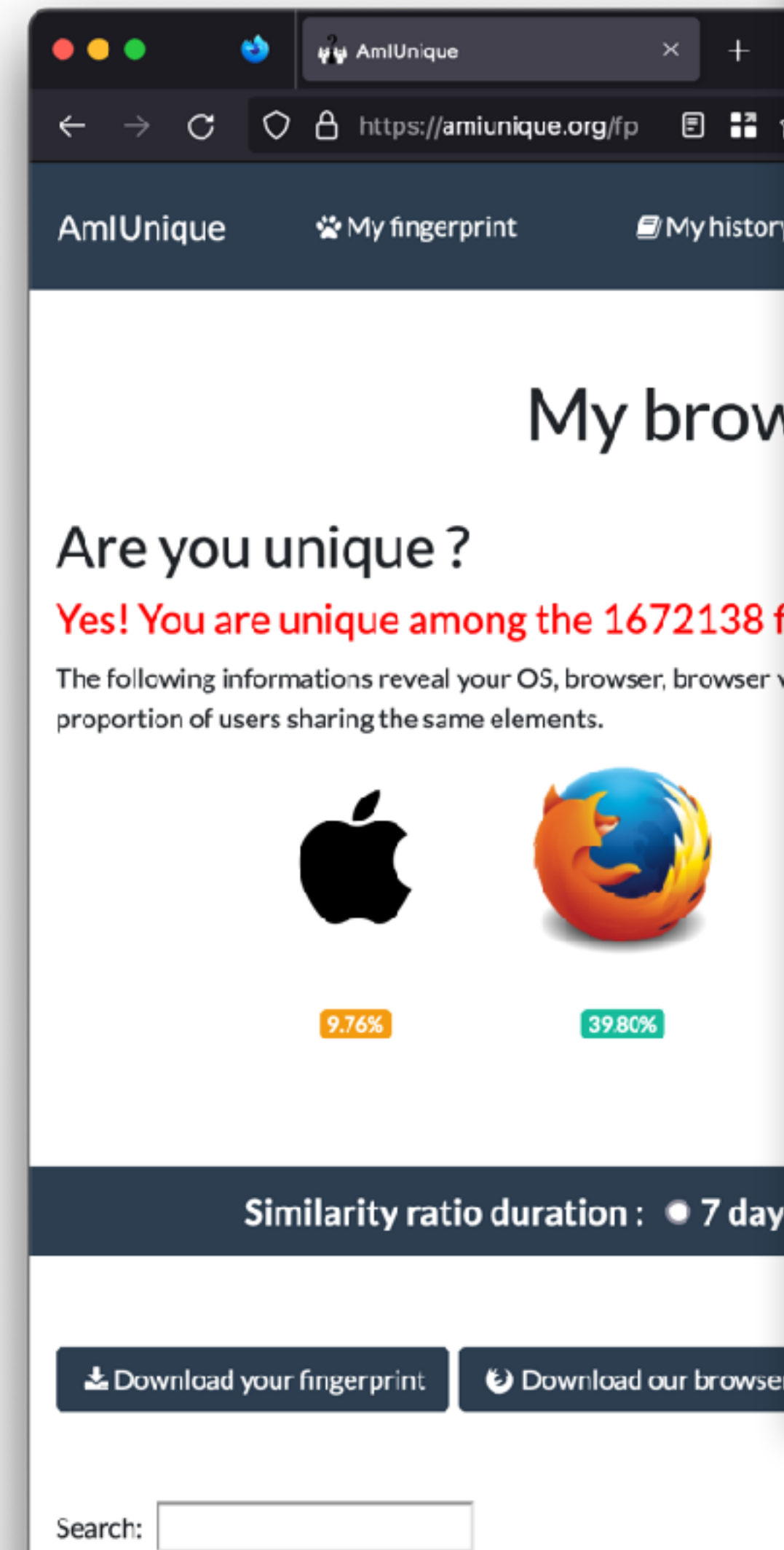
Bits of identifying information: 9.28
One in x browsers have this value: 623.46

WHAT IS FINGERPRINTING?



BROWSER WINDOW SIZE

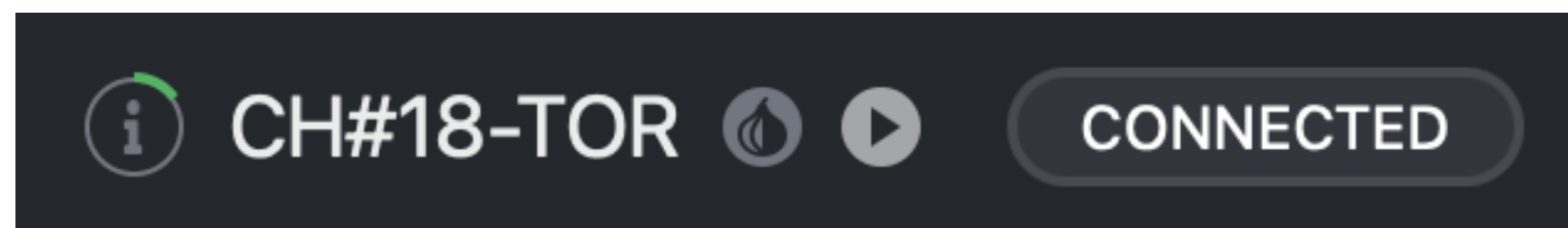
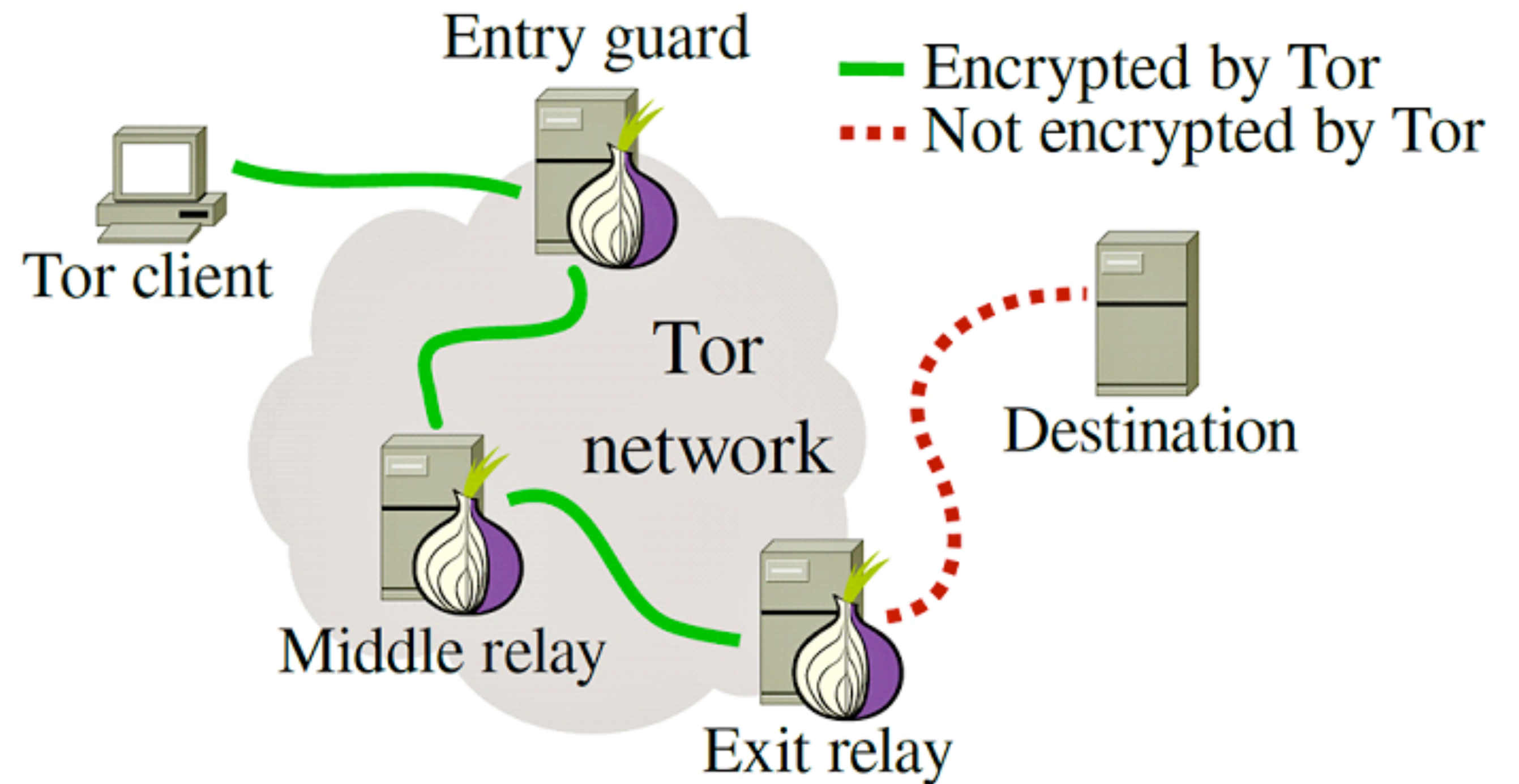
- Screen width and height can give away much information
- Tor has a default sized window.
(x * 200px, y * 100px)
→ in my case 1000px * 800px
- Not recommended to resize because of unique fingerprint.



Screen width ⓘ	21.46%	1920
Screen height ⓘ	20.22%	1080
Screen depth ⓘ	2.98%	30
Screen available top ⓘ	<0.01%	-111
Screen available Left ⓘ	0.58%	-1920
Screen available Height ⓘ	0.37%	1055
Screen available width ⓘ	20.77%	1920
Screen left ⓘ	0.17%	-1920
Screen top ⓘ	<0.01%	-136
Permissions ⓘ	1.94%	geolocation : prompt midi : prompt notifications : prompt persistent-storage : prompt push : prompt
WebGL Vendor ⓘ	0.66%	Apple
WebGL Renderer ⓘ	0.62%	Apple M1
WebGL Data ⓘ	0.13%	

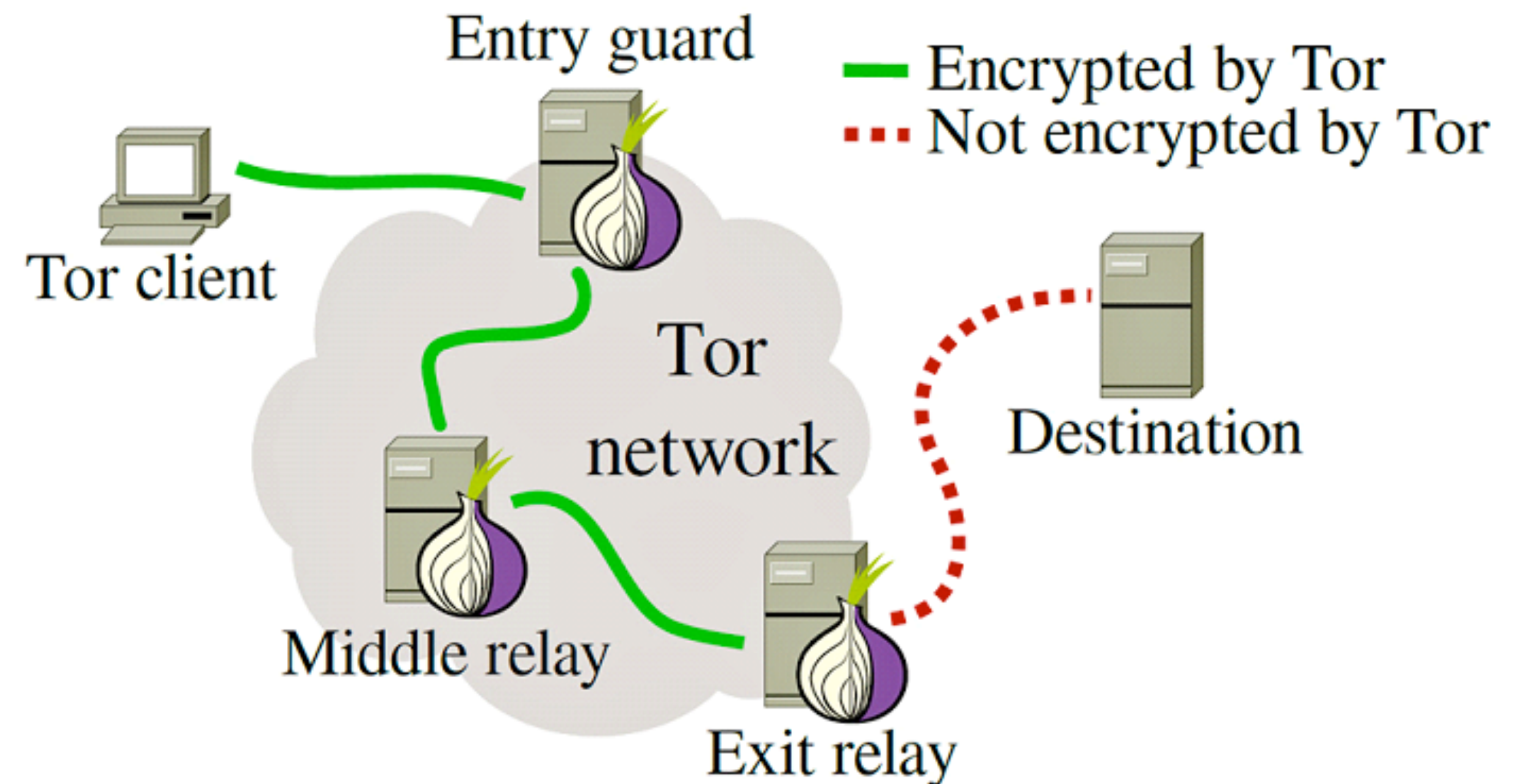
TOR BROWSER (THE ONION ROUTER)

- Each relay (= Tor node) decrypts the next “onion layer” to reveal only the next relay
- Some VPN Services offer to reroute traffic through Tor
- Spying on exit relay possible



.ONION

- 56 mostly random letters and numbers characters
- e.g. DuckDuckGo hidden service:
`https://duckduckgogg42×joc72×3sjasowoarfbgcmvfimaftt6twagswzczad.onion/`
- On server in Tor network (only accessible with Tor-browser)
- No spying possible on Tor exit node, because it is inside the Tor network



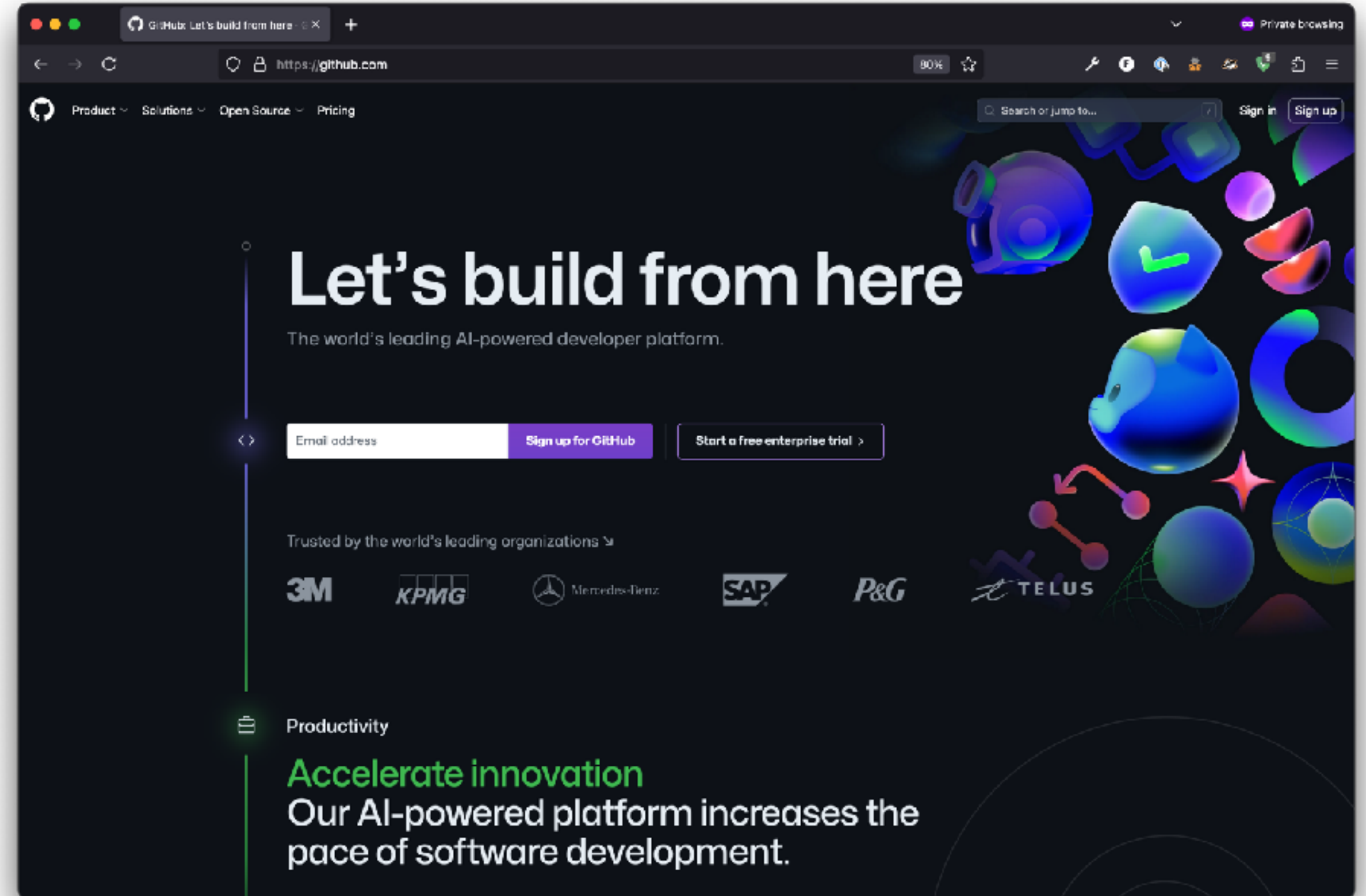
EXERCISE 3

- Go through provided markup:
<https://short.jonasscheiwiller.ch/ex3>
- Discuss in groups of 3 and write down new aspects and questions.
- Collect new tags and think about their semantic meaning.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <meta name="description" content="GeoLens Magazine – stories from the edges of the world." />
8   <title>GeoLens – Earth, Science & Exploration</title>
9 </head>
10
11 <body>
12
13   <header>
14     <a href="#main-content" class="skip-link">Skip to main content</a>
15     <div class="masthead">
16       <a href="/" aria-label="GeoLens home">
17         
18       </a>
19       <nav aria-label="Primary navigation">
20         <ul>
21           <li><a href="/science">Science</a></li>
22           <li><a href="/environment">Environment</a></li>
23           <li><a href="/exploration">Exploration</a></li>
24           <li><a href="/photography">Photography</a></li>
25           <li><a href="/archive">Archive</a></li>
26         </ul>
27       </nav>
28       <button type="button" aria-expanded="false" aria-controls="mobile-menu">
29         <span class="visually-hidden">Open menu</span>
30         <span aria-hidden="true">≡</span>
31       </button>
32     </div>
33     <nav id="mobile-menu" aria-label="Mobile navigation" hidden>
34       <ul>
35         <li><a href="/science">Science</a></li>
36         <li><a href="/environment">Environment</a></li>
37         <li><a href="/exploration">Exploration</a></li>
38         <li><a href="/photography">Photography</a></li>
39         <li><a href="/archive">Archive</a></li>
40       </ul>
41     </nav>
42   </header>
43
44   <main id="main-content">
45
```


GIT & GITHUB

- Most popular VCS (Version control system) is git.
- Github is a hosting platform for git repositories.
Alternatives: bitbucket.org, gitlab.com

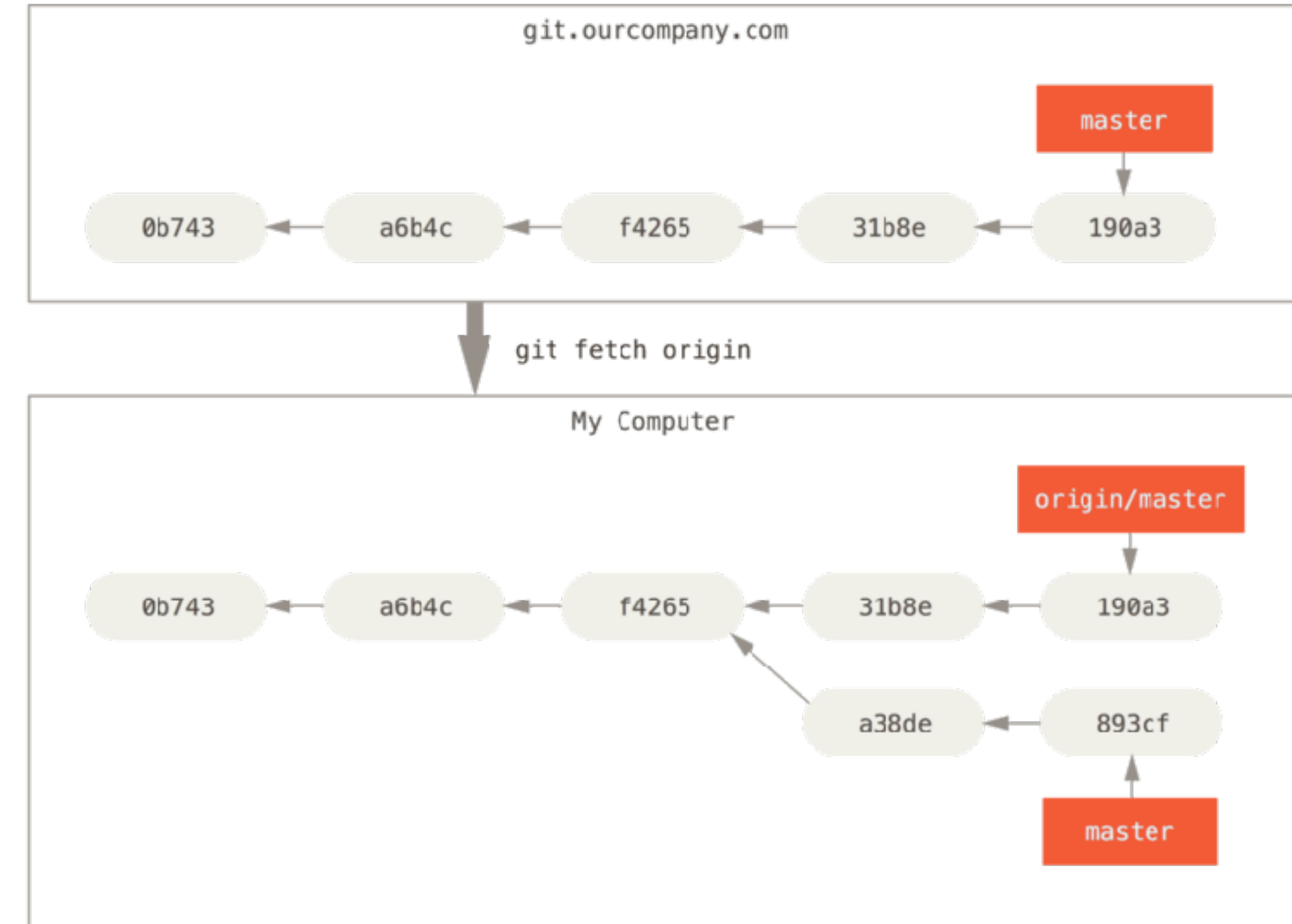


EXERCISE 4

- What are the advantages of using git?
- List the most common actions performed when working with git?

GIT

- Version control system
- Collaboration (Team/Open Source)



GIT

Git history



Branch

Commits

- feat: make more fields translatable, upgrade deps to fix problem with seomatic graphql caching
- feat: change translation method for nested entry links in image grid matrix
- Merge branch 'feature/delete-unused-assets'
- [↗ origin/feature/delete-unused-assets](#) [↗ feature/delete-unused-assets](#) feat: make some texts translatable
- feat: add plugin born05/craft-assetusage and make test
- fix: fix project yaml
- Merge branch 'main' into staging
- feat: update schema version
- feat: change craft version to redeploy
- feat: update deps and patch vulnerability
- feat: make teaser text translatable
- feat: update dependencies
- feat: upgrade craft (critical update)
- [↗ origin/feature/iframe](#) feat: add iframe field to page builder
- chore(deps): update

GIT → PREVIEW CHANGES

Lines deleted

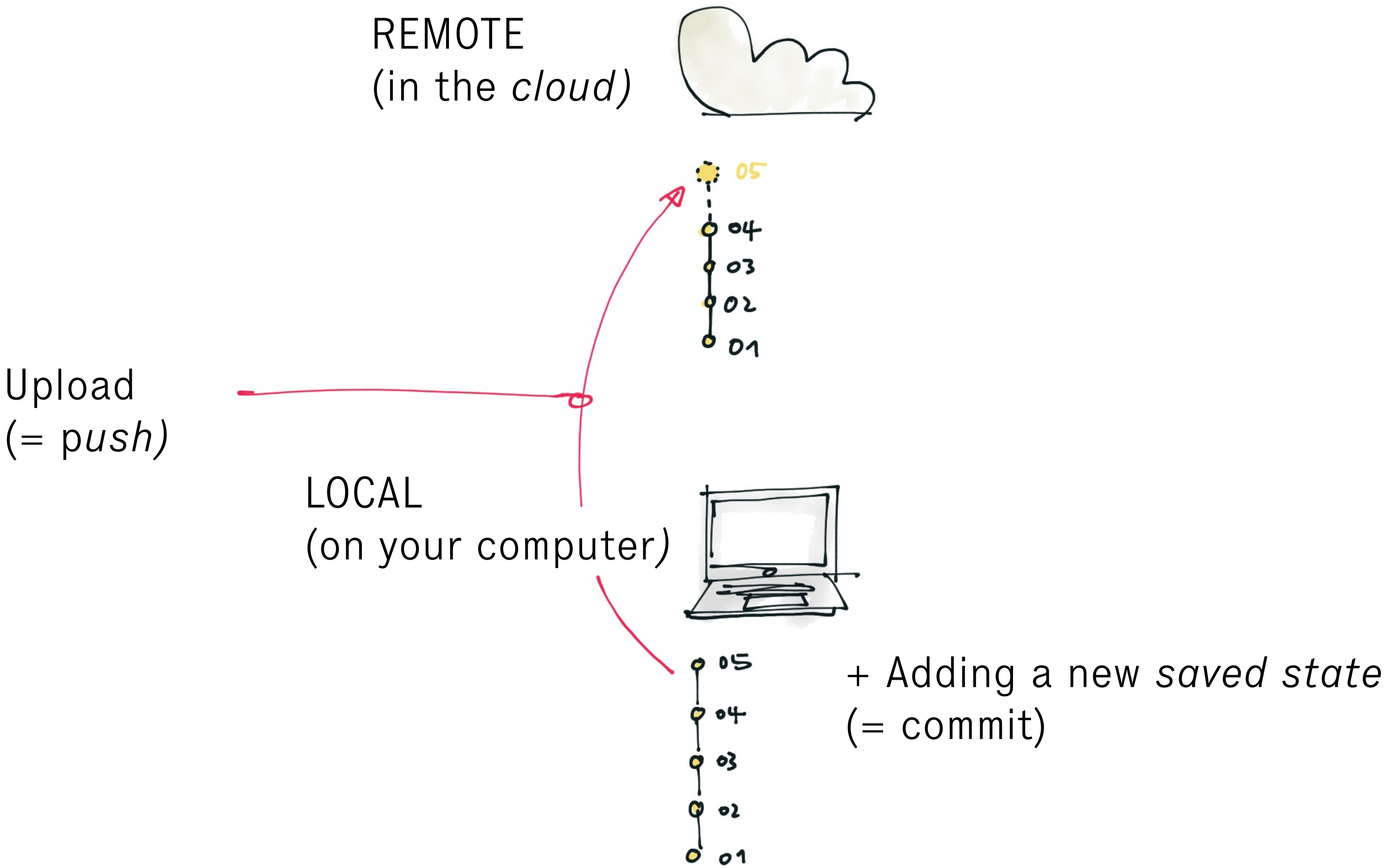
```

7      .#{ $c } {
8      -   align-items: center;
9      -   background-color: $color-secondary;
10         height: 100vh;
11      -   justify-content: center;
12         padding-top: $s-header-height-static;
13         width: 100vw;
14
15
16         padding-top: $s-header-height-dynamic;
17     }
18
19     +   &__text-wrapper {
20     +       align-items: center;
21     +       height: 100%;
22     +
23     +       // because of main-grid class we can use:
24     +       justify-content: center;
25     +   }
26
27     +

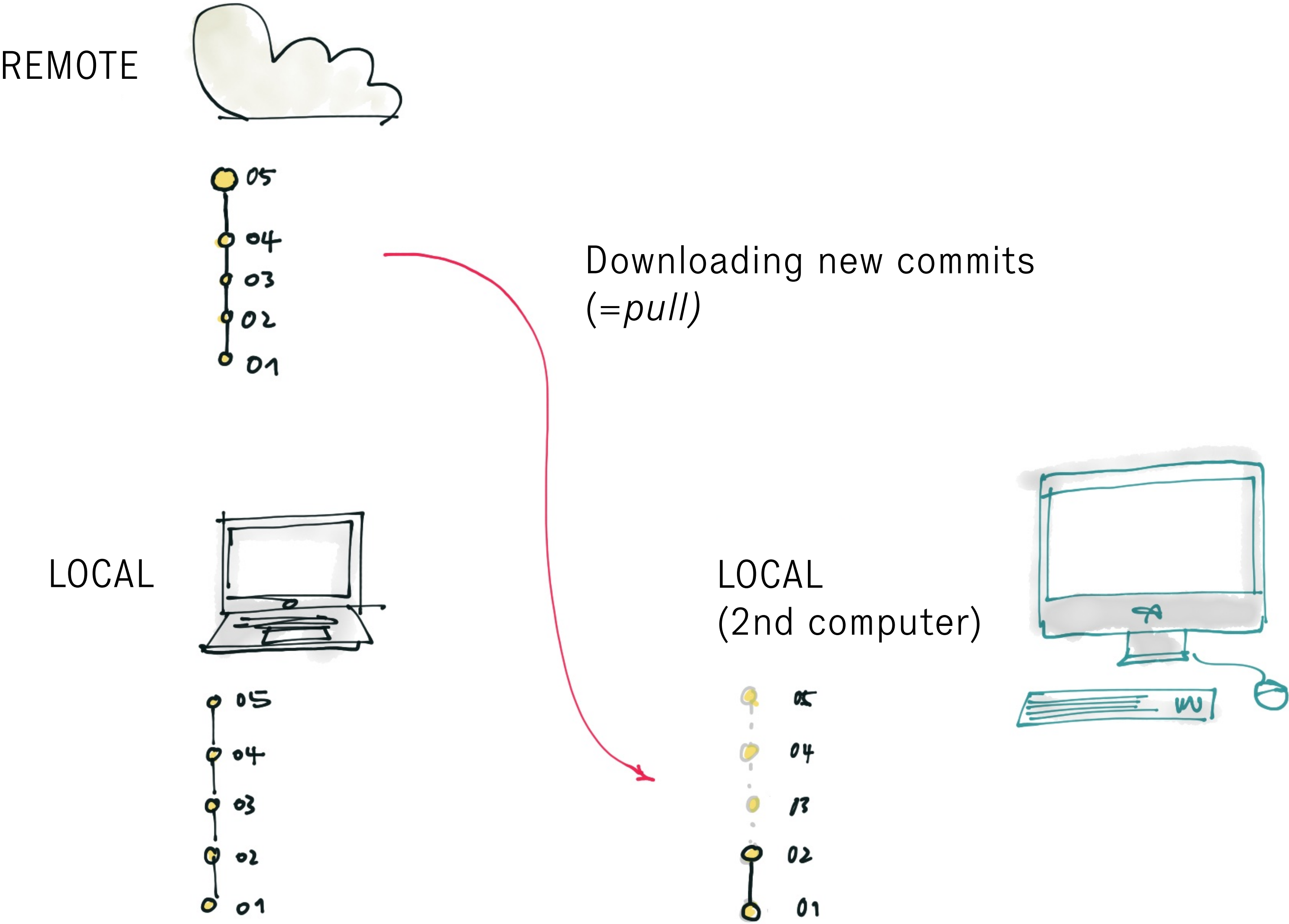
```

Lines
added

GIT → REMOTE VS. LOCAL

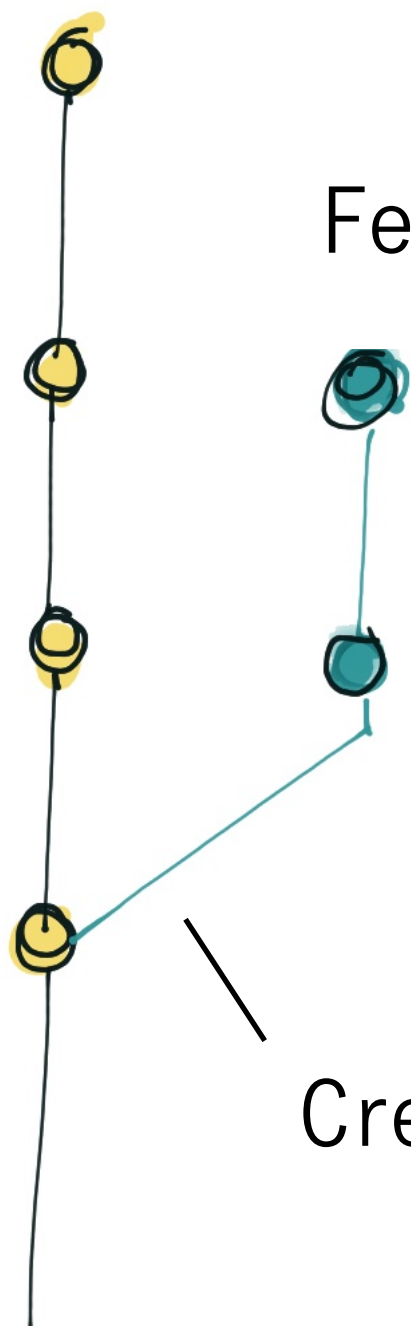


GIT → PUSH & PULL



GIT → BRANCHES

Main Branch



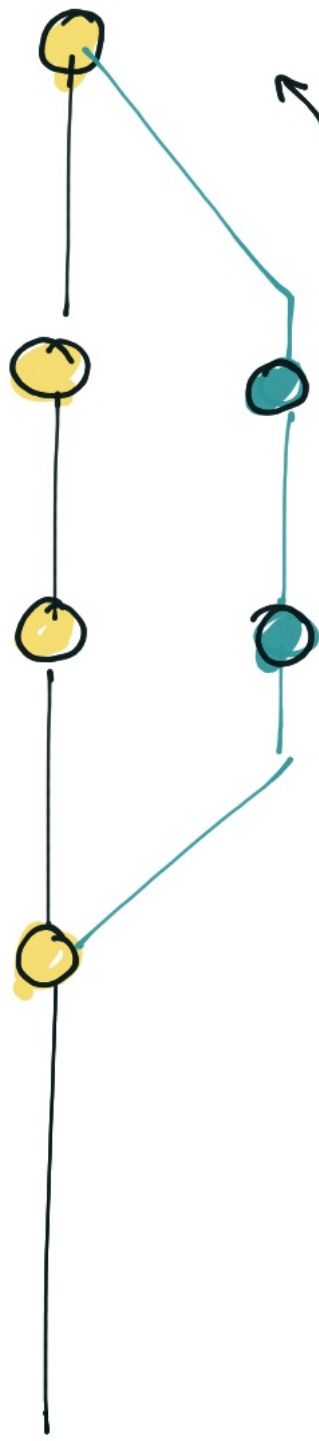
Feature Branch



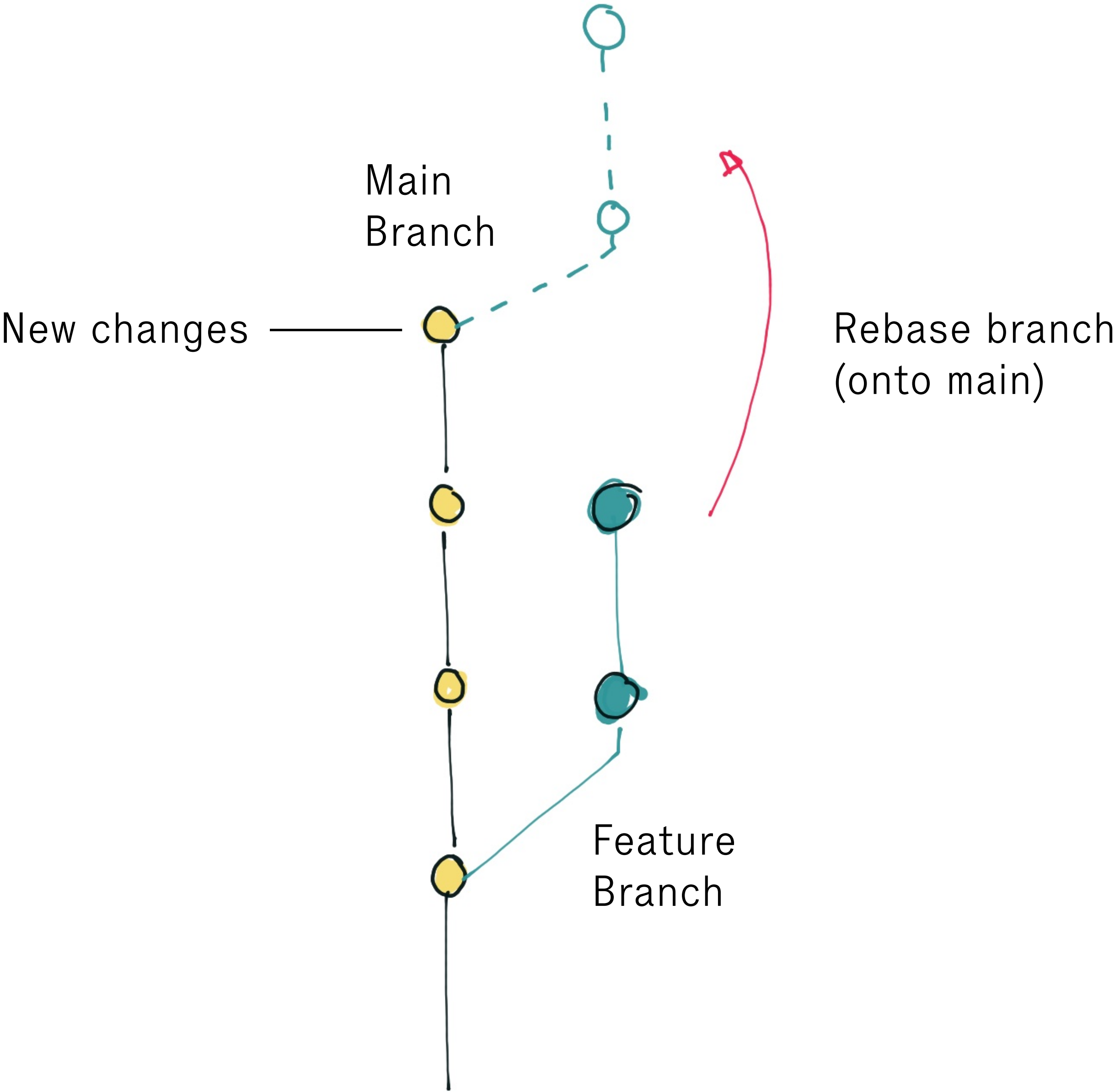
Create new branch



Merge branch



GIT → REBASE

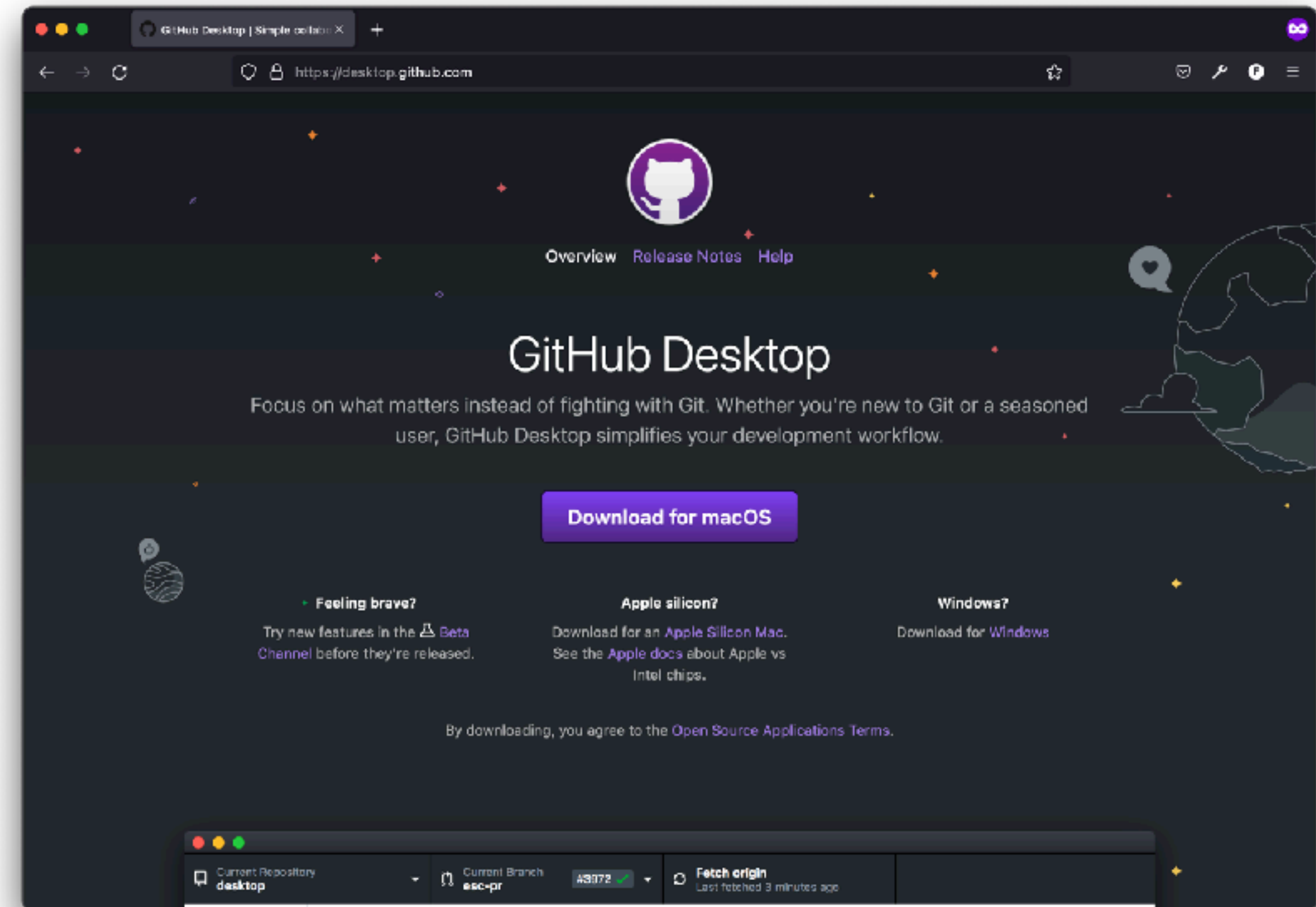


GIT → LINGO

- Repository: Codebase (Project)
- Commit: Single entry in history
- Origin/Remote: Code on server (e.g. github)
- Local: Code on your machine
- Push/Pull/Fetch: Actions to get code from or upload to remote
- Master/Main: Main branch
- Merge: Bringing to branches with different code together
- Merge conflicts: When both branches during a merge have edited the same file merge conflicts appear.
- Rebase: Reapply commits on top of another branch

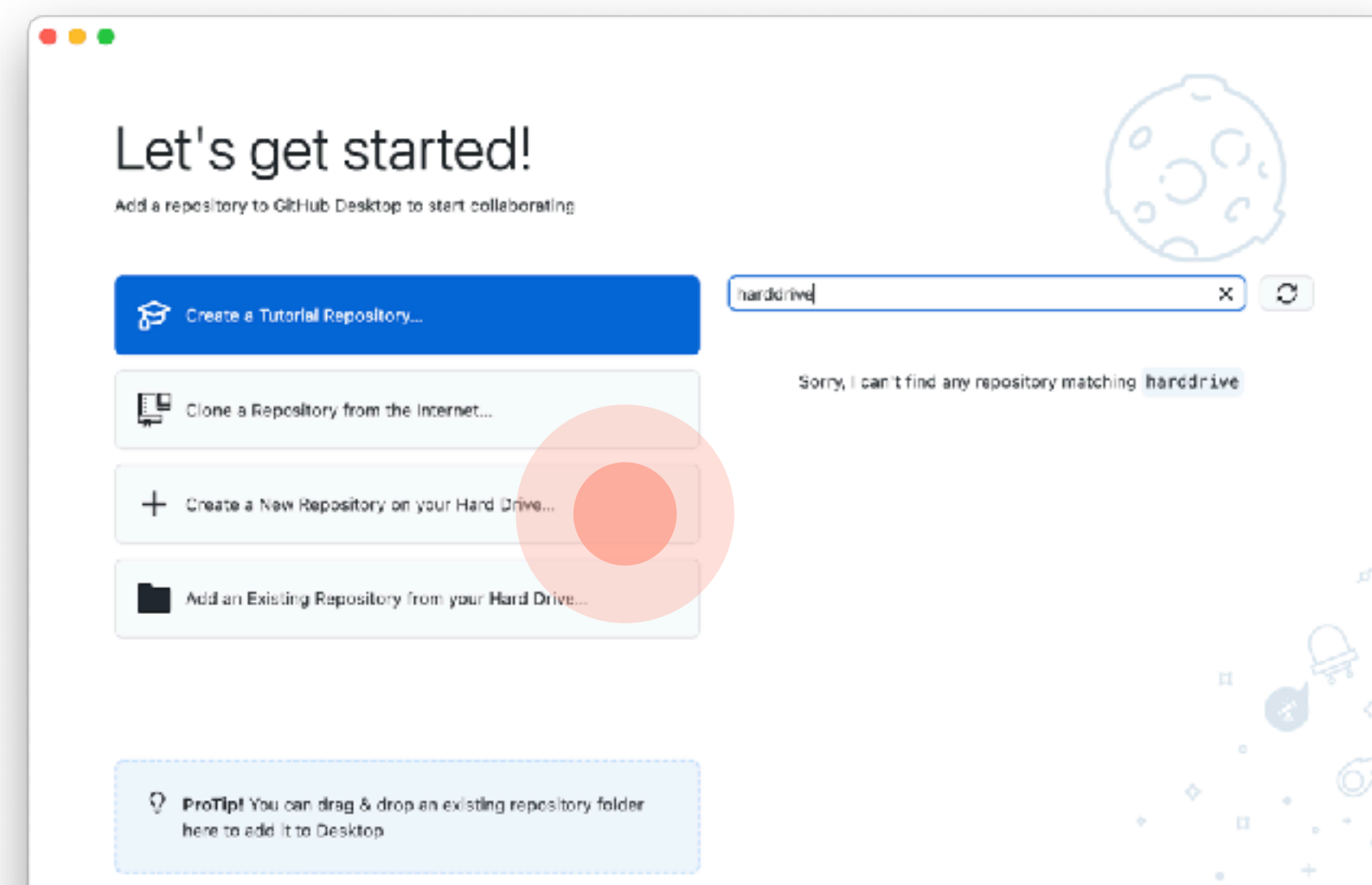
GITHUB CLIENT

- Github Desktop
<https://desktop.github.com/>
- Alternatives (if you know what you're doing)
 - Sourcetree (<https://www.sourcetreeapp.com/>)
 - GitKraken (<https://www.gitkraken.com/git-client>)
 - CLI (command line interface)
 - ...



EXERCISE 4

- Create GitHub repository
 - Follow best naming practices
- Install *Github Desktop*
- Create local folder structure
- Move your files into the created folder and rename your files (following best practices).
- ⚠ The folder should be outside of any cloud folders (OneDrive, dropbox, google drive etc.)!



General Best Practices for Naming Repositories

Keep Names Descriptive

Names should be descriptive enough to convey the repository's purpose or functionality. A good repository name equips contributors with instant context.

Example:

- **Good:** ``ecommerce-product-catalog``
- **Bad:** ``project123``

In the above example, ``ecommerce-product-catalog`` clearly indicates what the repository contains, while ``project123`` fails to provide any meaningful information.

Use Hyphens or Underscores for Separation

Utilizing hyphens or underscores in naming ensures better readability. A clear separation between words makes it easier for others to understand the repository name at a glance.

Example:

- Preferred: ``user-authentication-module``
- Not preferred: ``userauthenticationmodule``

Using separators improves visual clarity and comprehension.

Favor Lowercase Letters

Many operating systems (e.g., Linux) are case-sensitive, meaning that ``ProjectA`` and ``projecta`` would be treated as two distinct entities. Adopting lowercase for your repository names ensures consistency and avoids potential issues.

Example:

- Prefer: ``data-analytics``

FOLDER STRUCTURE

- Don't use uppercase letters!
- Don't use special characters!
- Don't use spaces! Use dashes or underscores instead
- Keep everything in one place (one folder)

```
1 ~/Developer/
2 |— my-portfolio/
3 |— side-project/
4 |— company-app/
5 |— zhdk-webtechnologies/
6
7 <!-- Alternative, structured by github user -->
8 ~/Developer/
9 |— wearelucid/
10 |   |— client-project-name/
11 |   |— zoozh-web/
12 |— jones-s/
13 |   |— zhdk-webtechnologies-2026/
14
15 <!-- Alternative, structured by topic -->
16 ~/dev/
17 |— personal/
18 |   |— my-portfolio/
19 |   |— side-project/
20 |— work/
21 |   |— company-app/
22 |   |— internal-tools/
23 |— learning/
24 |— webtechnologies/
```


EXERCISE 4

- Create GitHub repository
 - Follow best naming practices
- Install *Github Desktop*
- Create local folder structure
- Move your files into the created folder and rename your files (following best practices).
- ⚠ The folder should be outside of any cloud folders (OneDrive, dropbox, google drive etc.)!

```
1 ~/Developer/  
2 |— my-portfolio/  
3 |— side-project/  
4 |— company-app/  
5 |— zhdk-webtechnologies/  
6  
7 <!-- Alternative, structured by github user -->  
8 ~/Developer/  
9 |— wearelucid/  
10 |   |— client-project-name/  
11 |   |— zoozh-web/  
12 |— jones-s/  
13 |   |— zhdk-webtechnologies-2026/  
14  
15 <!-- Alternative, structured by topic -->  
16 ~/dev/  
17 |— personal/  
18 |   |— my-portfolio/  
19 |   |— side-project/  
20 |— work/  
21 |   |— company-app/  
22 |   |— internal-tools/  
23 |— learning/  
24 |   |— webtechnologies/
```

General Best Practices for Naming Repositories

Keep Names Descriptive

Names should be descriptive enough to convey the repository's purpose or functionality. A good repository name equips contributors with instant context.

Example:

- **Good:** `ecommerce-product-catalog`
- **Bad:** `project123`

In the above example, `ecommerce-product-catalog` clearly indicates what the repository contains, while `project123` fails to provide any meaningful information.

Use Hyphens or Underscores for Separation

Utilizing hyphens or underscores in naming ensures better readability. A clear separation between words makes it easier for others to understand the repository name at a glance.

Example:

- Preferred: `user-authentication-module`
- Not preferred: `userauthenticationmodule`

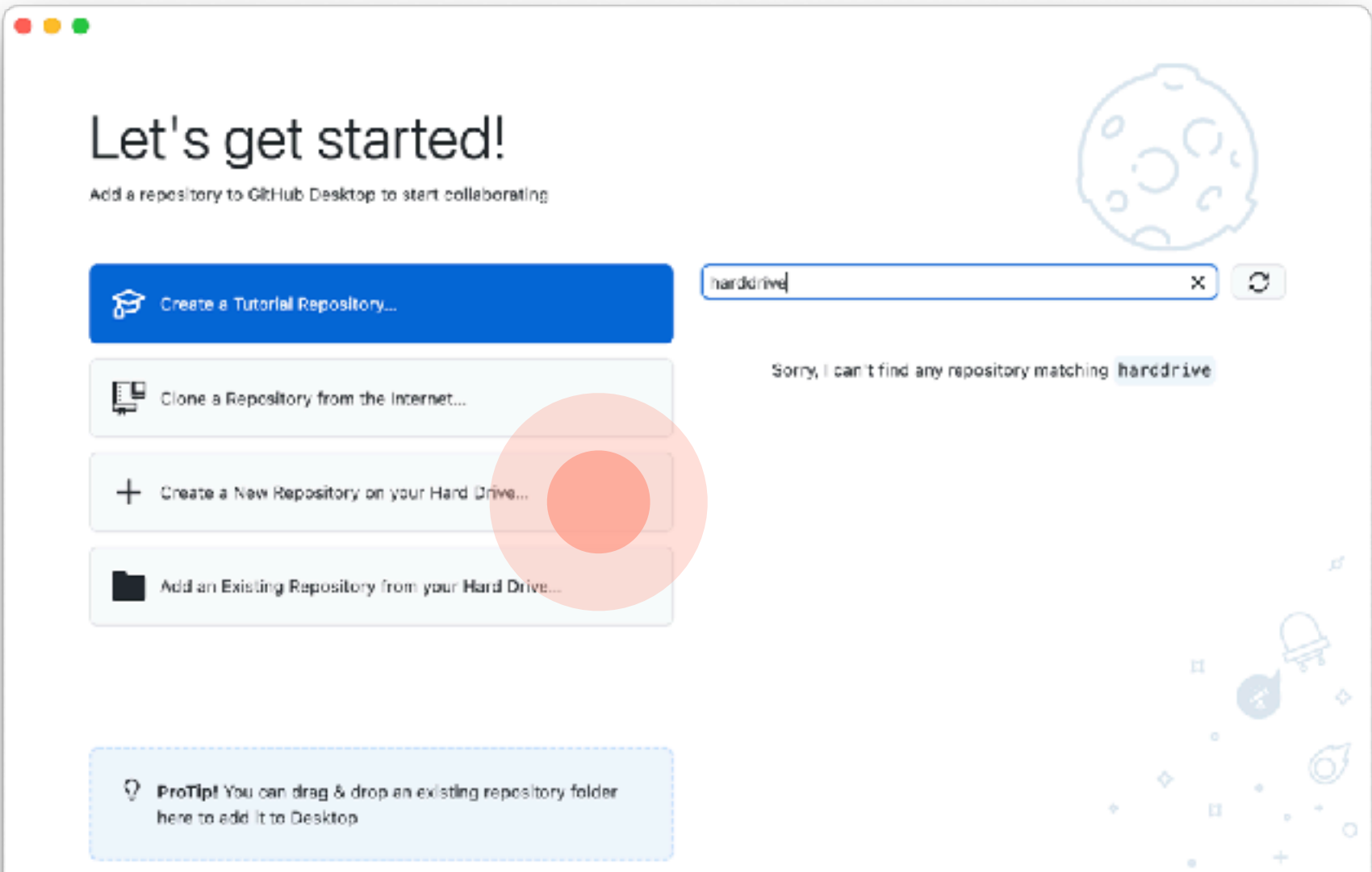
Using separators improves visual clarity and comprehension.

Favor Lowercase Letters

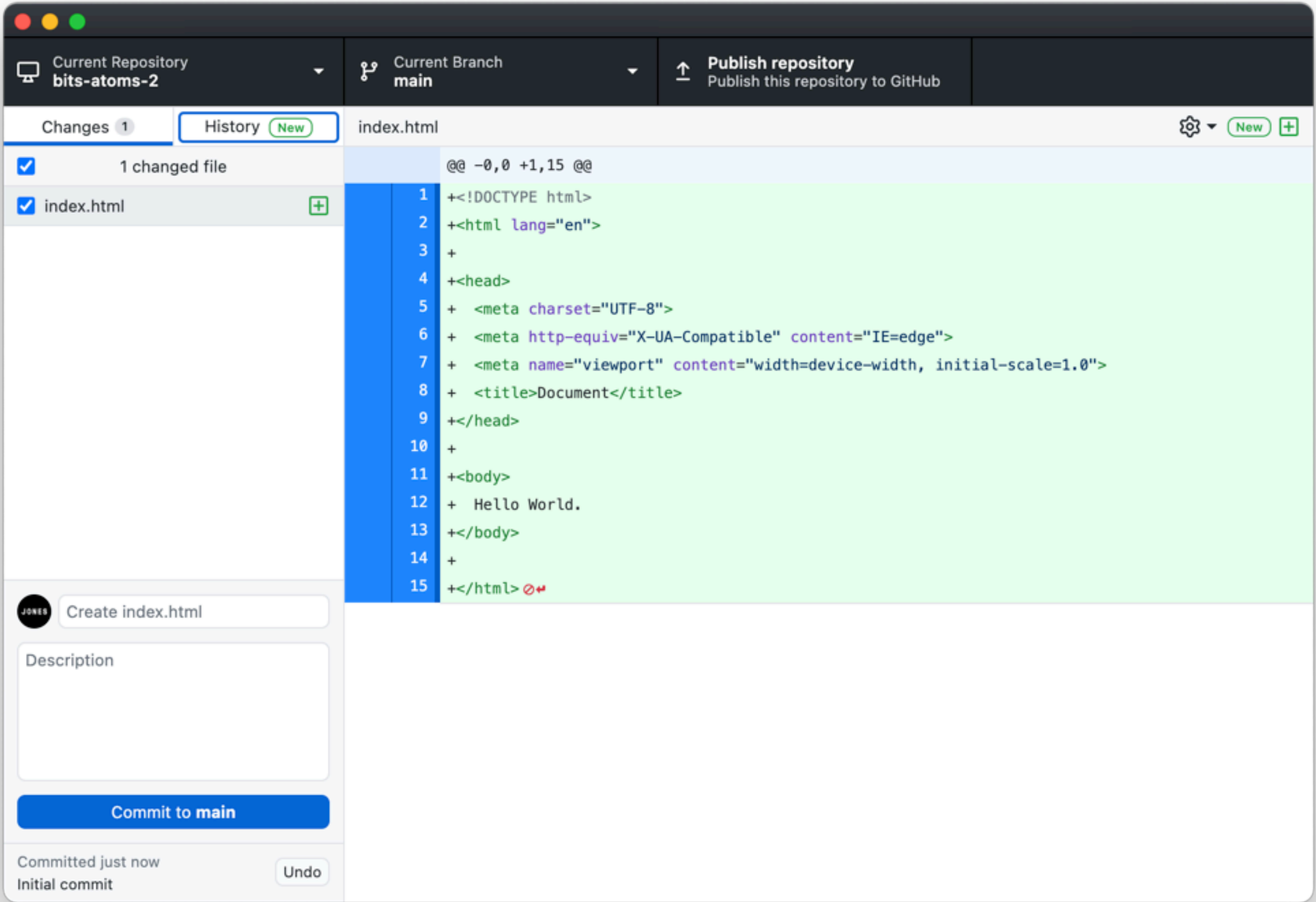
Many operating systems (e.g., Linux) are case-sensitive, meaning that `ProjectA` and `projecta` would be treated as two distinct entities. Adopting lowercase for your repository names ensures consistency and avoids potential issues.

Example:

- Prefer: `data-analytics`

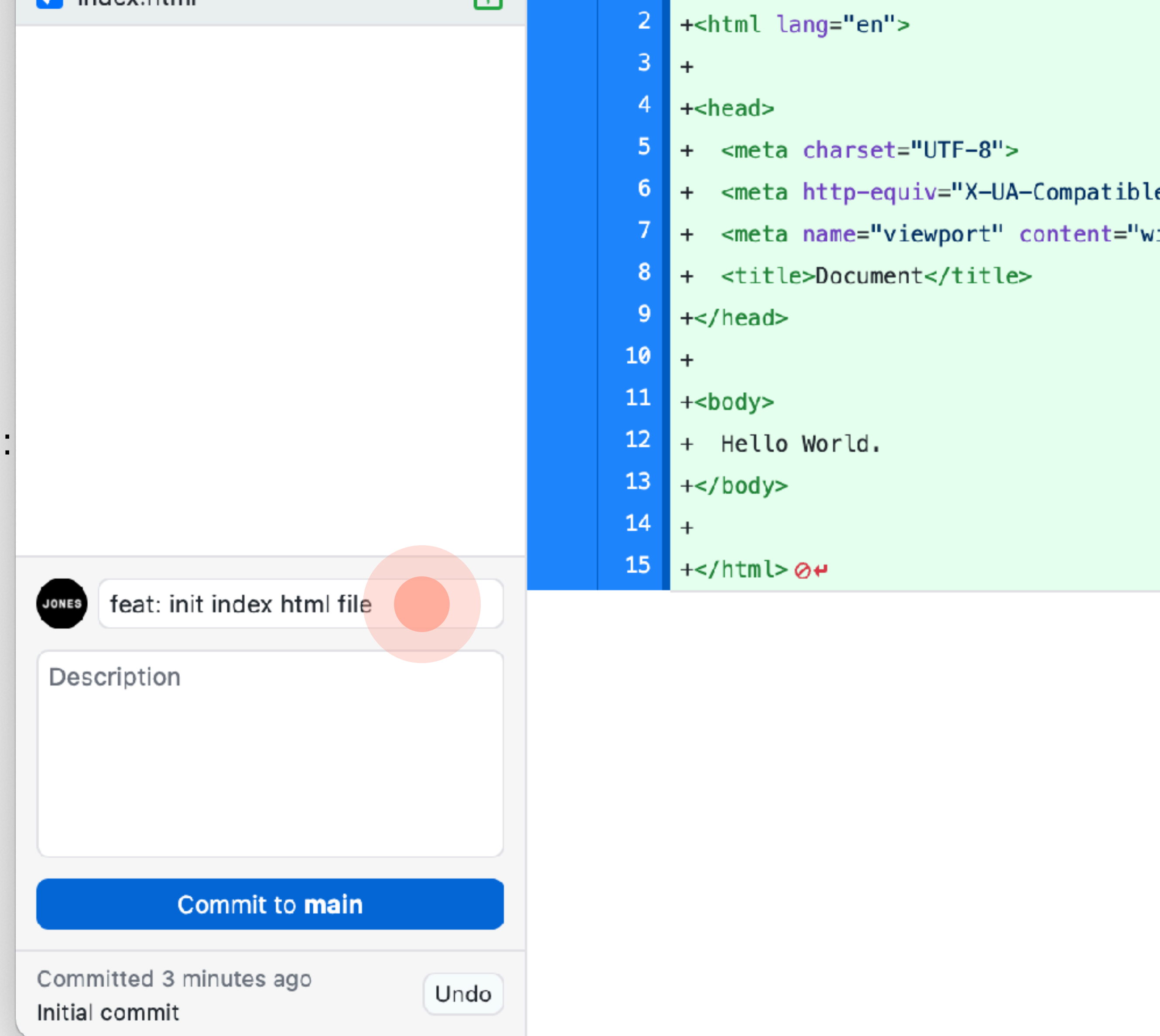
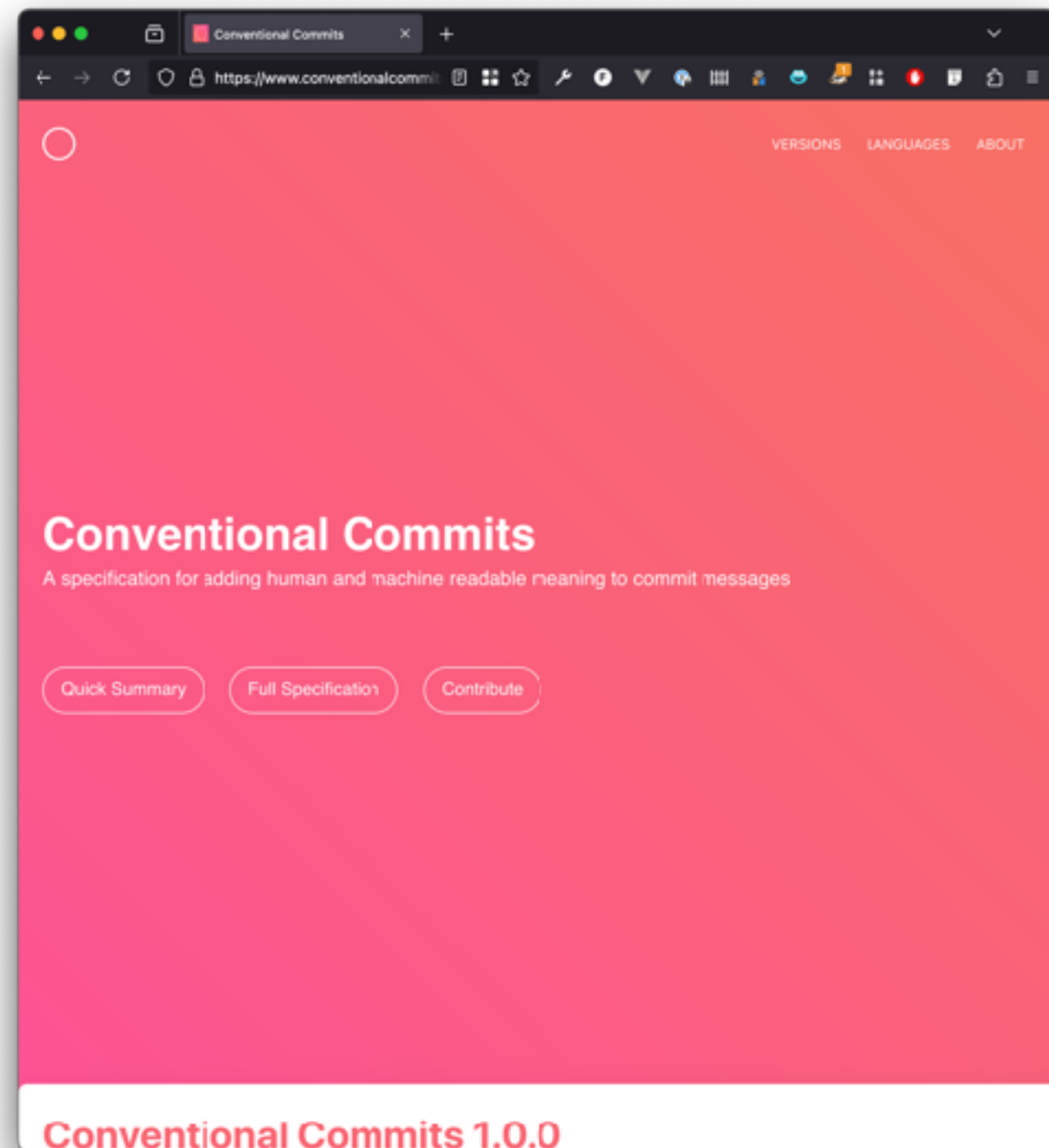


CHECK YOUR CHANGES



FIRST COMMIT

- Describe your changes and commit them.
 - ⚠ Description is rarely used
- You might want to start using some commit guidelines:
<https://www.conventionalcommits.org/en/v1.0.0/>



PUBLISH (& PUSH)

