

System Design Concepts

Enterprise Architectures for Big Data



I didn't invent
the Internet

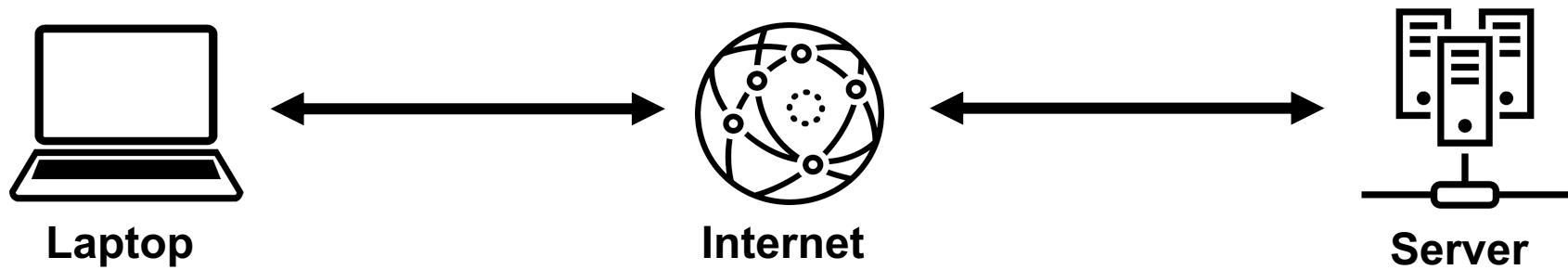
I did not
invent the Web



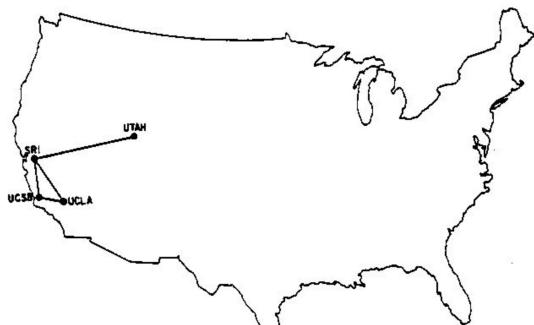
I invented
the Web

I invented
the Internet

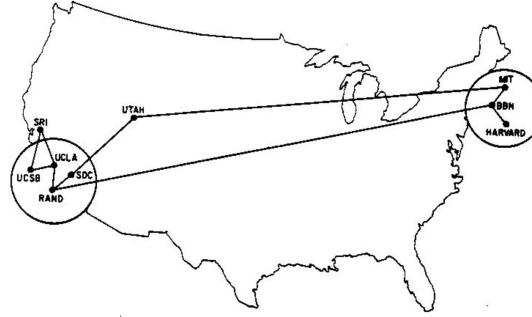
How does my computer communicate with other computers over the internet?



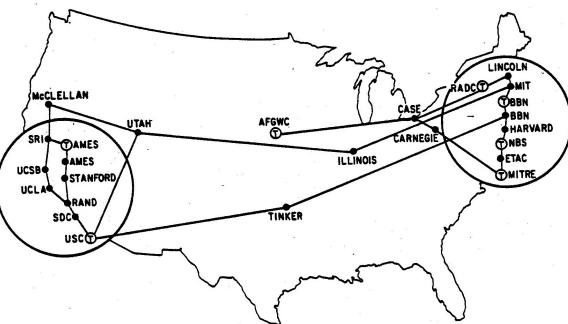
ARPANET (starting 1969)



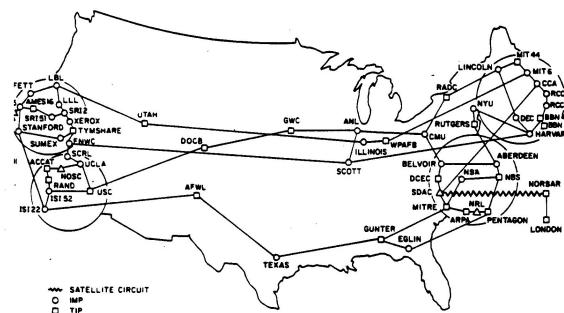
Dezember 1969



Juni 1970

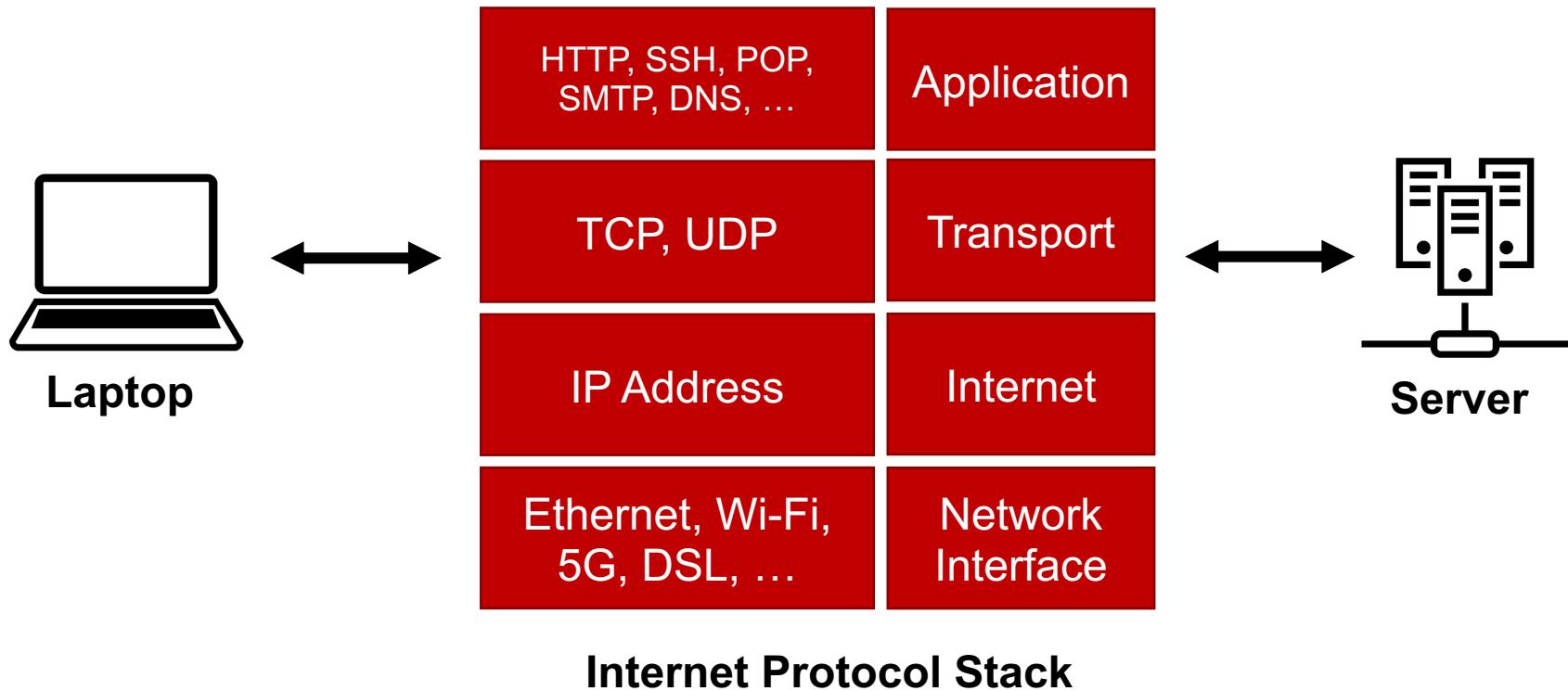


März 1972



Juli 1977

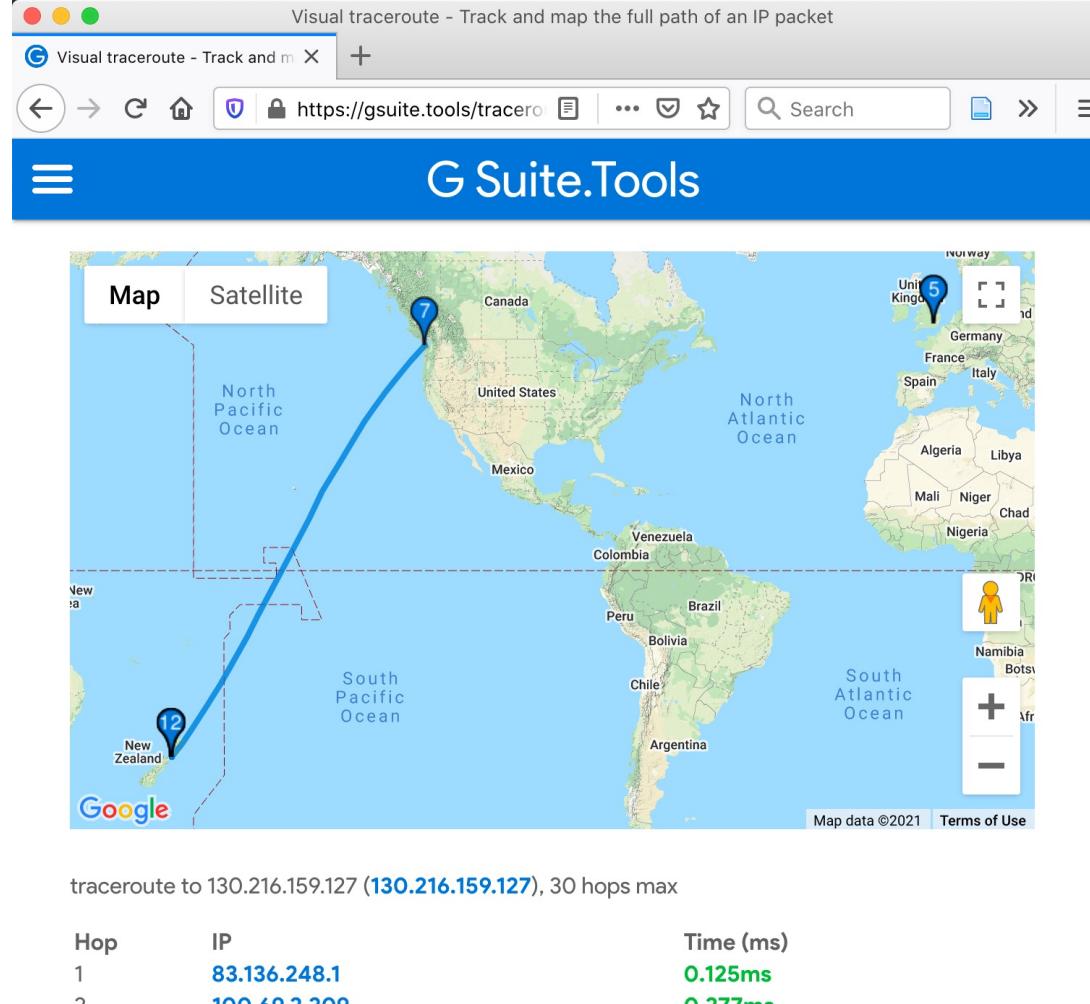
How does my computer communicate with other computers over the internet?



How does my computer communicate with other computers over the internet?

- traceroute 130.216.159.127

```
[rolandmueller@Rolands-MacBook-Pro-3 ~ % traceroute 130.216.159.127
traceroute to 130.216.159.127 (130.216.159.127), 64 hops max, 52 byte packets
1  192.168.178.1 (192.168.178.1)  4.305 ms  2.049 ms  2.338 ms
2  192.0.0.1 (192.0.0.1)  10.787 ms  6.691 ms  6.410 ms
3  62.214.36.172 (62.214.36.172)  6.967 ms  7.244 ms  6.786 ms
4  62.214.39.200 (62.214.39.200)  8.518 ms  7.313 ms  7.465 ms
5  et-1-0-15.edge6.dusseldorf1.level3.net (212.162.40.37)  7.594 ms  8.971 ms  8.245 ms
6  * * *
7  research-an.ear2.seattle1.level3.net (4.53.158.194)  327.353 ms  169.923 ms  221.442 ms
8  et-0-0-500.grt.and33-pdx.reannz.co.nz (210.7.33.244)  217.902 ms  436.809 ms  216.684 ms
9  ae0-500.grt.and32-mgw.reannz.co.nz (210.7.33.246)  299.608 ms  324.407 ms  620.538 ms
10 xe-0-0-0-500.grt.and12-nsh.reannz.co.nz (210.7.33.242)  300.563 ms  334.707 ms  466.474 ms
11 210.7.39.177 (210.7.39.177)  301.931 ms  302.397 ms  368.228 ms
12 210.7.39.178 (210.7.39.178)  316.839 ms  338.774 ms  391.754 ms
13 * * *
14 * * *
```



TCP Port Numbers

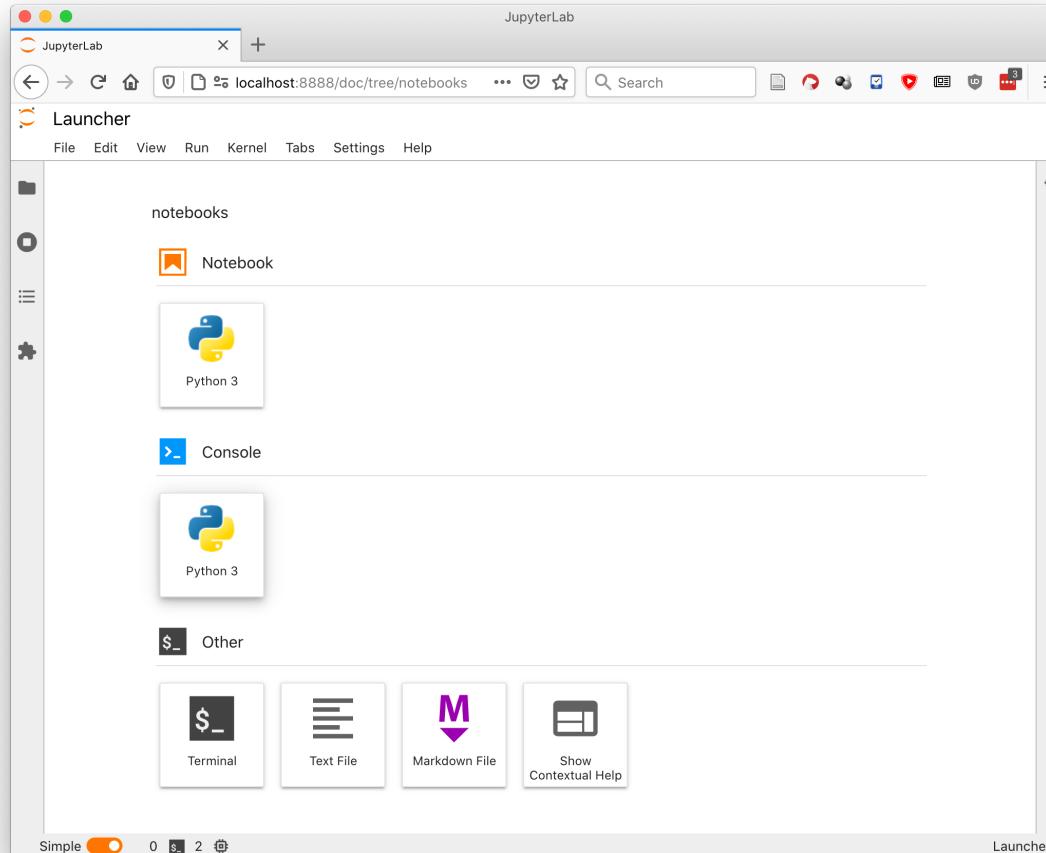
- Port numbers for matching services (protocols)
- Official assignments
- Unofficial use especially of higher port numbers
 - e.g. Jupyter, PostgreSQL, MySQL, Hadoop, ...
- Most important Protocols with Port Numbers:
 - Hypertext Transfer Protocol (**HTTP**): 80
 - Hypertext Transfer Protocol Secure (**HTTPS**): 443
 - Secure Shell Protocol (**SSH**): 22

Check what program is listing on what port number

```
[rolandmueller@Rolands-MacBook-Pro-3 ~ % sudo lsof -nP -iTCP -sTCP:LISTEN
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
postgres 198 postgres 7u IPv6 0x3d62c7414703d88d 0t0 TCP *:5432 (LISTEN)
postgres 198 postgres 8u IPv4 0x3d62c741464ba1d 0t0 TCP *:5432 (LISTEN)
coreaudio 280 _coreaudiod 3u IPv4 0x3d62c741464ba53d 0t0 TCP 127.0.0.1:49152 (LISTEN)
xartstora 414 root 3u IPv4 0x3d62c741464bb8fd 0t0 TCP *:61500 (LISTEN)
xartstora 414 root 4u IPv6 0x3d62c7414703dead 0t0 TCP *:61500 (LISTEN)
rapportd 471 rolandmueller 4u IPv4 0x3d62c7414bf569d 0t0 TCP *:49274 (LISTEN)
rapportd 471 rolandmueller 5u IPv6 0x3d62c7414703c00d 0t0 TCP *:49274 (LISTEN)
Dropbox 561 rolandmueller 115u IPv4 0x3d62c7414a24ab5d 0t0 TCP *:17500 (LISTEN)
Dropbox 561 rolandmueller 116u IPv6 0x3d62c7415b4b89ed 0t0 TCP *:17500 (LISTEN)
Dropbox 561 rolandmueller 165u IPv4 0x3d62c74149cba8fd 0t0 TCP 127.0.0.1:17600 (LISTEN)
Dropbox 561 rolandmueller 167u IPv4 0x3d62c7414d9c469d 0t0 TCP 127.0.0.1:17603 (LISTEN)
Adobe\20 761 rolandmueller 13u IPv4 0x3d62c7414b4f38fd 0t0 TCP 127.0.0.1:15292 (LISTEN)
zotero 1361 rolandmueller 36u IPv4 0x3d62c7414ceaa62dd 0t0 TCP 127.0.0.1:23119 (LISTEN)
zotero 1361 rolandmueller 37u IPv4 0x3d62c7414cea453d 0t0 TCP 127.0.0.1:23116 (LISTEN)
zotero 1361 rolandmueller 38u IPv4 0x3d62c74149cbc69d 0t0 TCP 127.0.0.1:19876 (LISTEN)
eddie-cli 1362 root 14u IPv4 0x3d62c74149cb91fd 0t0 TCP 127.0.0.1:34490 (LISTEN)
openvpn 1613 root 14u IPv4 0x3d62c74149cb9f1d 0t0 TCP 127.0.0.1:34490 (LISTEN)
rolandmueller@Rolands-MacBook-Pro-3 ~ % ]
```

```
[rolandmueller@Rolands-MacBook-Pro-3 ~ % netcat -z -v hwr-berlin.de 80
hwr-berlin.de [194.94.23.251] 80 (http) open
rolandmueller@Rolands-MacBook-Pro-3 ~ % ]
```

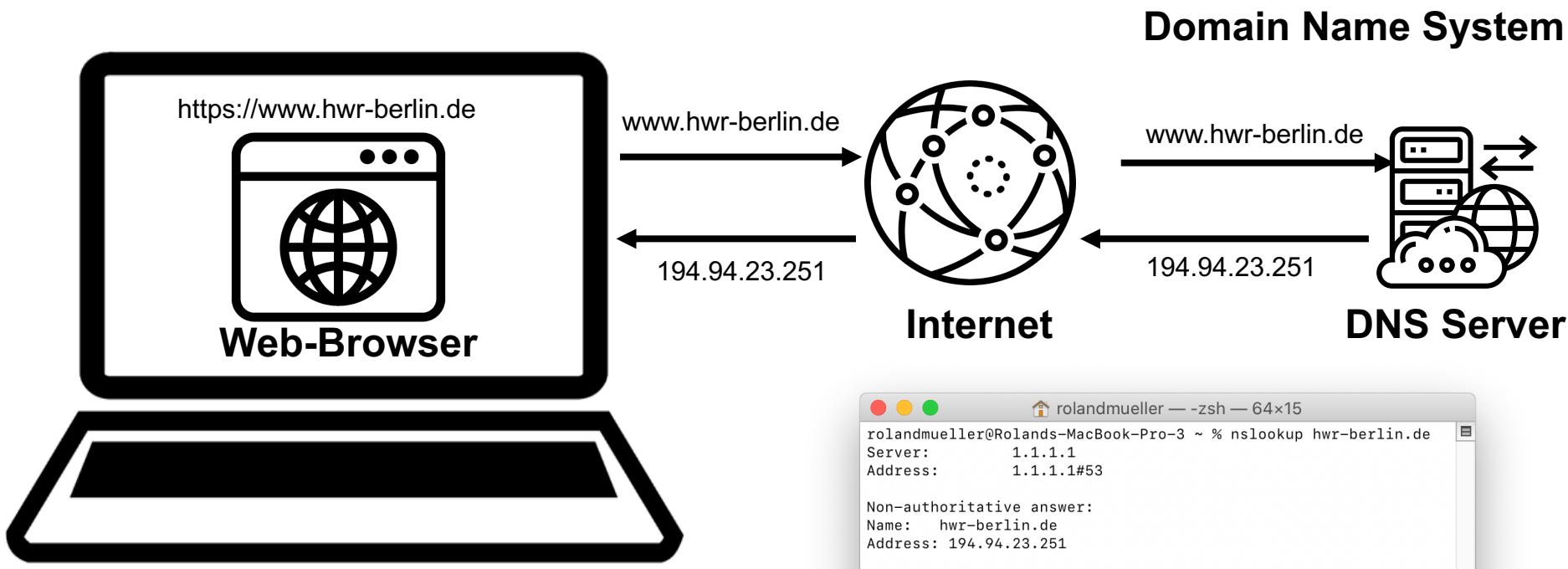
Port Numbers for distinguishing different Services



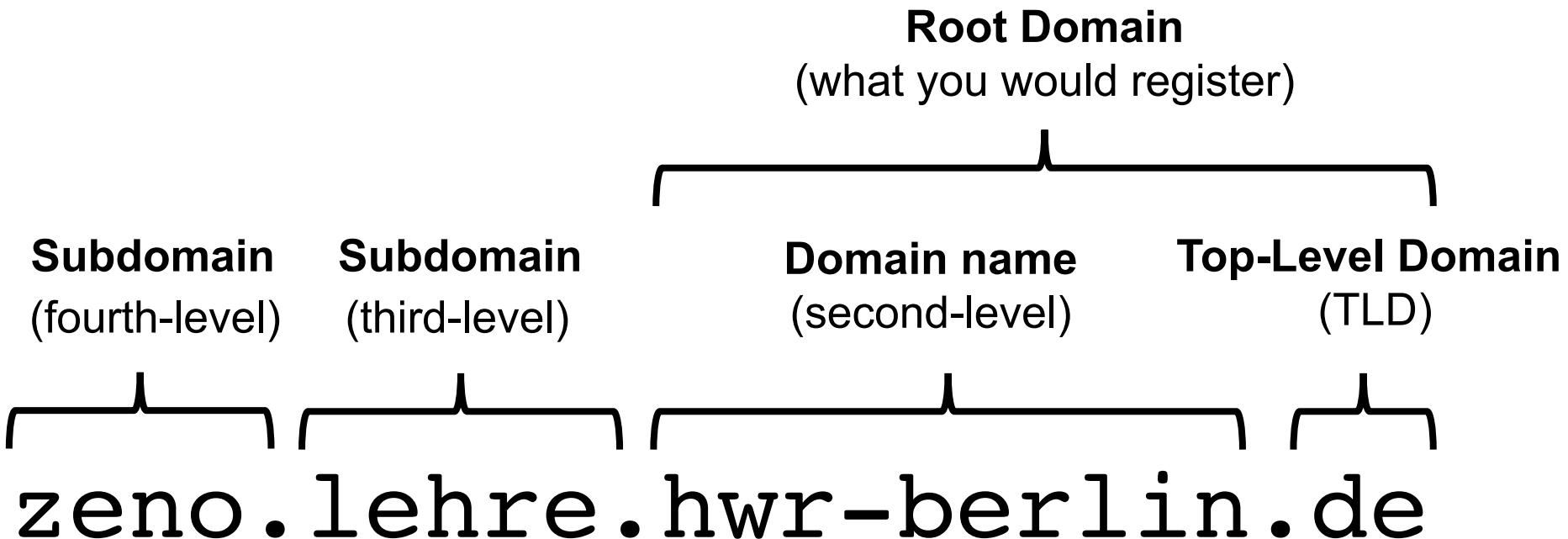
Firewall

- Monitors and controls network traffic
- Can run on and shield
 - a single computer (endpoint-based application firewalls)
 - or a subnetwork (demilitarized zone)
- Different types of firewalls:
 - Most common and most easy: Packet filter based on transport mechanism (TCP/IP) and Port number
 - More advanced: deep packet inspection or connection-based

How do I get the IP-address for a domain?

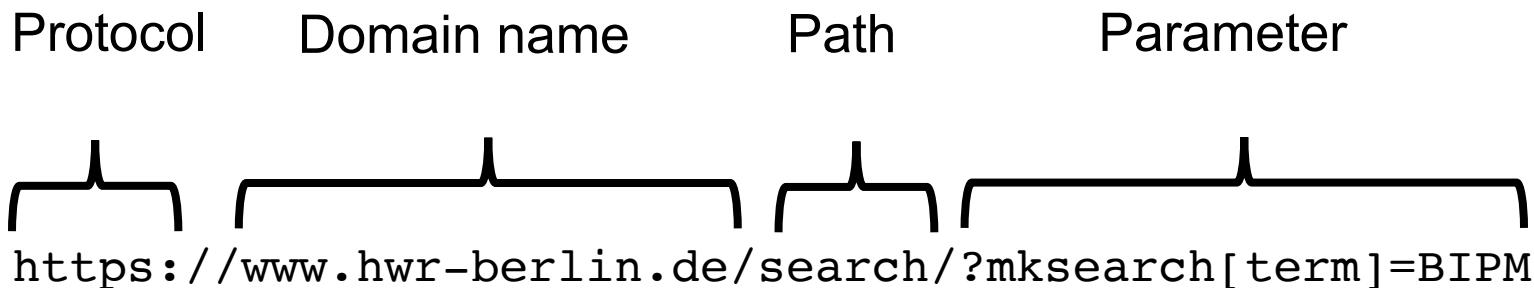


Domain name



URL

- Uniform Resource Locator
- Web address for a resource on the internet



HTTP: Hypertext Transfer Protocol

HTTP Get and Post Method

Get

- Click on a Link. URL with path and query string (name/value pairs)

`http://example.com/user/?name1=value1&name2=value2`

- Values: Only ASCII characters
- Length restrictions (maximum URL length is 2048 characters)
- Can be cached
- Can be bookmarked
- Reload is harmless (Idempotent)

Post

- Submit a form in the browser
- Values: No restriction
- No length restrictions
- Never cached
- Cannot be bookmarked
- Reload: Data will be resubmitted
- Form submit could also be Get (e.g. in a search engine)

HTTP: Hypertext Transfer Protocol

Web Developer Views (Firefox) or Developer Tools (Chrome)

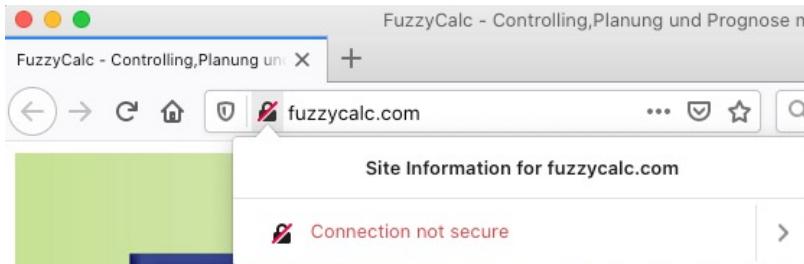
The screenshot shows the Firefox developer tools Network tab for the HWR Berlin website. The main content area displays the university's logo and name. Below the content, the developer tools interface is visible, featuring a toolbar with various developer tools like Inspector, Console, and Debugger, and a main pane showing network requests.

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	www.hwr-berli...	/	browsing-context...	html	20.28 KB	160.12 ...	170 ms
200	GET	www.hwr-berli...	merged-820b7d6f8cad6cc023a8159d99476a4d-f8...	script	js	914 B (raced)	930 B	31 ms
200	GET	www.hwr-berli...	merged-ee9317f04b2094fdcb8d1e32a4f27e00-7ea...	script	js	6.69 KB (raced)	24.21 KB	33 ms
200	GET	ajax.googleapis...	jquery.min.js	script	js	33.36 KB (raced)	93.54 ...	372 ms
200	GET	www.hwr-berli...	merged-9a31ffccb7021baf170118b86e081f14-004dc...	script	js	130 KB (raced)	438.4...	206 ms
200	GET	ajax.googleapis...	jquery-ui.min.js	script	js	67.07 KB (raced)	247.72...	354 ms

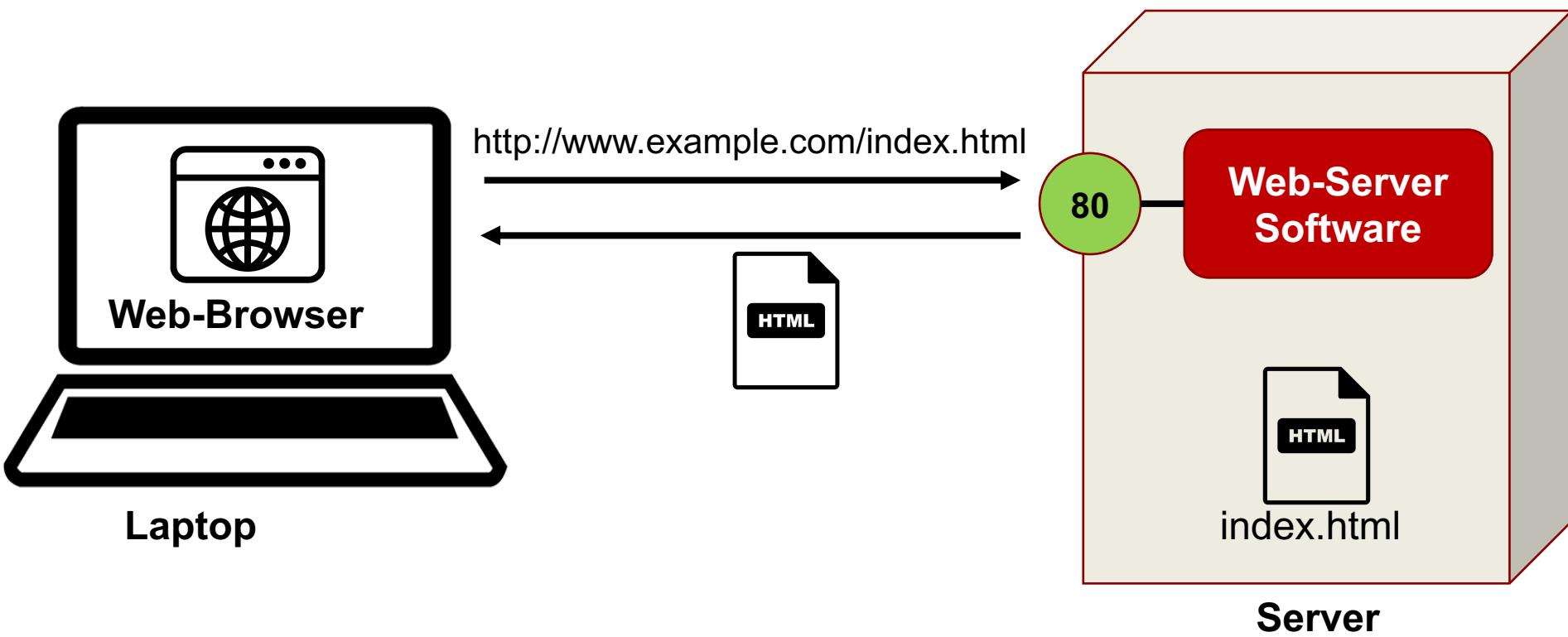
At the bottom, a summary bar indicates 34 requests, 2.13 MB transferred, a finish time of 2.91 s, and a DOMContentLoaded time of 1.51 s. The load time is highlighted in pink at 2.10 s.

HTTPS: Hypertext Transfer Protocol Secure

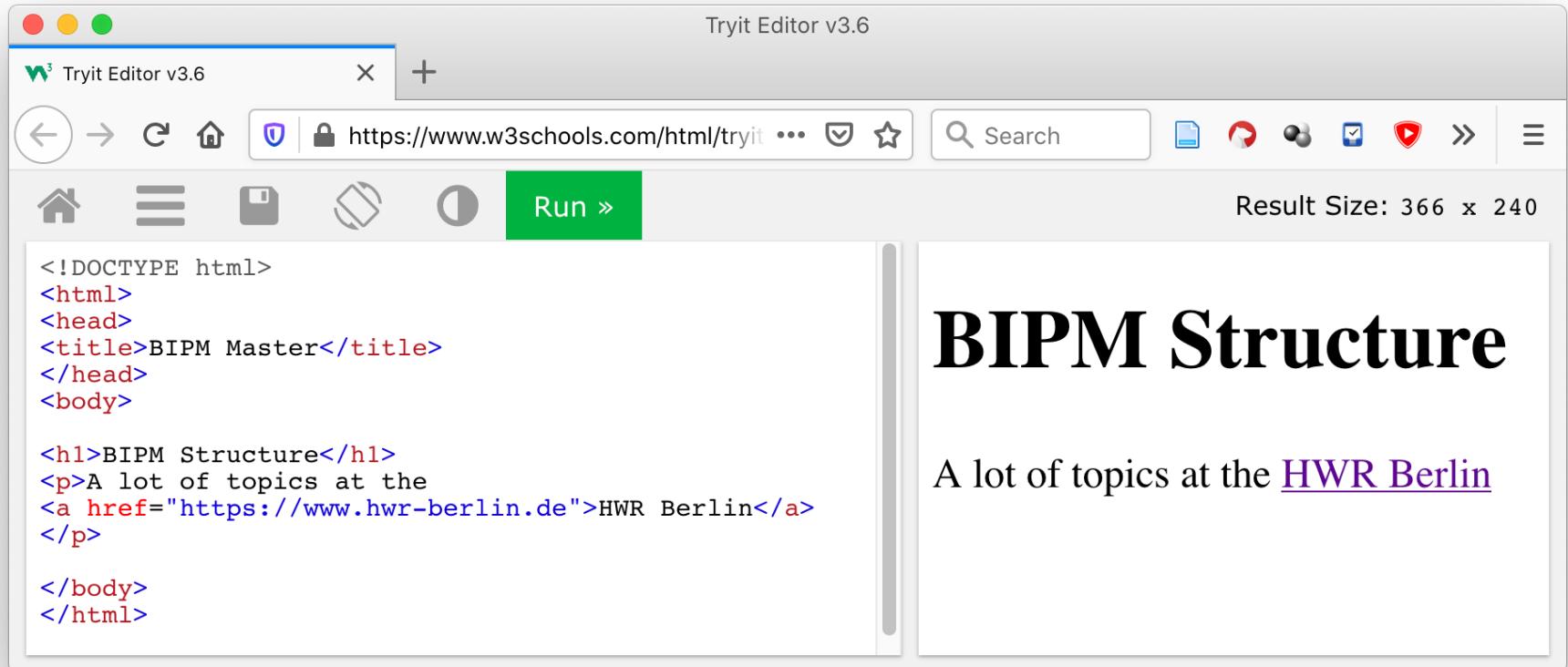
- Uses encryption for securing the transportation between server and browser
 - Uses public-key protocols (Diffie–Hellman key exchange) to exchange a common random and unique session key
 - Uses symmetric-key algorithm for encrypting the communication
- Transport Layer Security (TLS) / SSL (Secure Sockets Layer)
 - HTTPS = HTTP over TLS / HTTP over SSL
- Needs SSL/TLS certificate (e.g. from Let's Encrypt)



Most simple Web Page: Static Website



HTML: Hypertext Markup Language



The screenshot shows a web-based HTML editor interface. The top bar includes a title "Tryit Editor v3.6", a tab "Tryit Editor v3.6", a search bar, and various toolbar icons. The main area contains an HTML code editor with the following content:

```
<!DOCTYPE html>
<html>
<head>
<title>BIPM Master</title>
</head>
<body>

<h1>BIPM Structure</h1>
<p>A lot of topics at the
<a href="https://www.hwr-berlin.de">HWR Berlin</a>
</p>

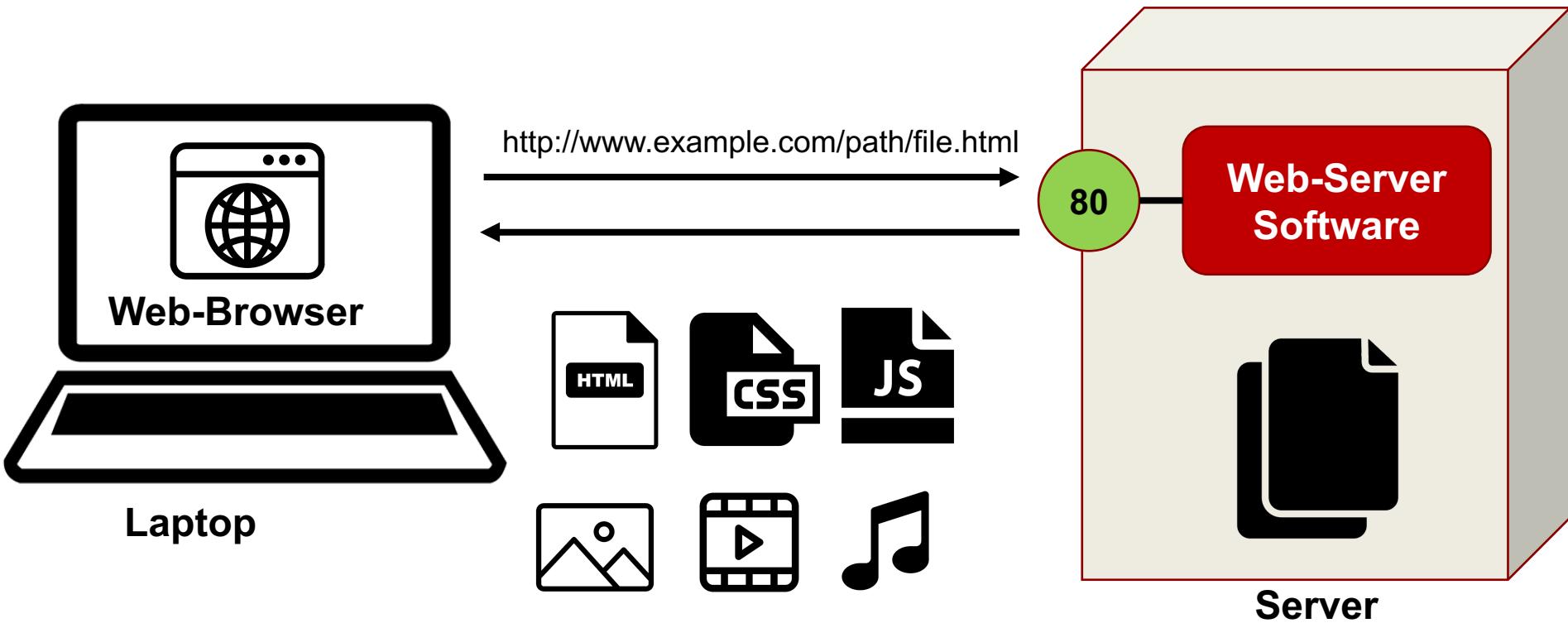
</body>
</html>
```

To the right of the editor is a preview pane displaying the rendered HTML output:

BIPM Structure

A lot of topics at the [HWR Berlin](https://www.hwr-berlin.de)

Most simple Web Page: Static Website



CSS: Cascading Style Sheets

The screenshot shows a web-based code editor window titled "Tryit Editor v3.6". The URL in the address bar is https://www.w3schools.com/css/tryit.asp?filename=trycss_default. The editor interface includes a toolbar with icons for file operations, a search bar, and a "Run" button. The code area contains the following CSS and HTML:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}
</style>
</head>
<body>

<h1>BIPM is awesome</h1>
<p>Yes it is.</p>

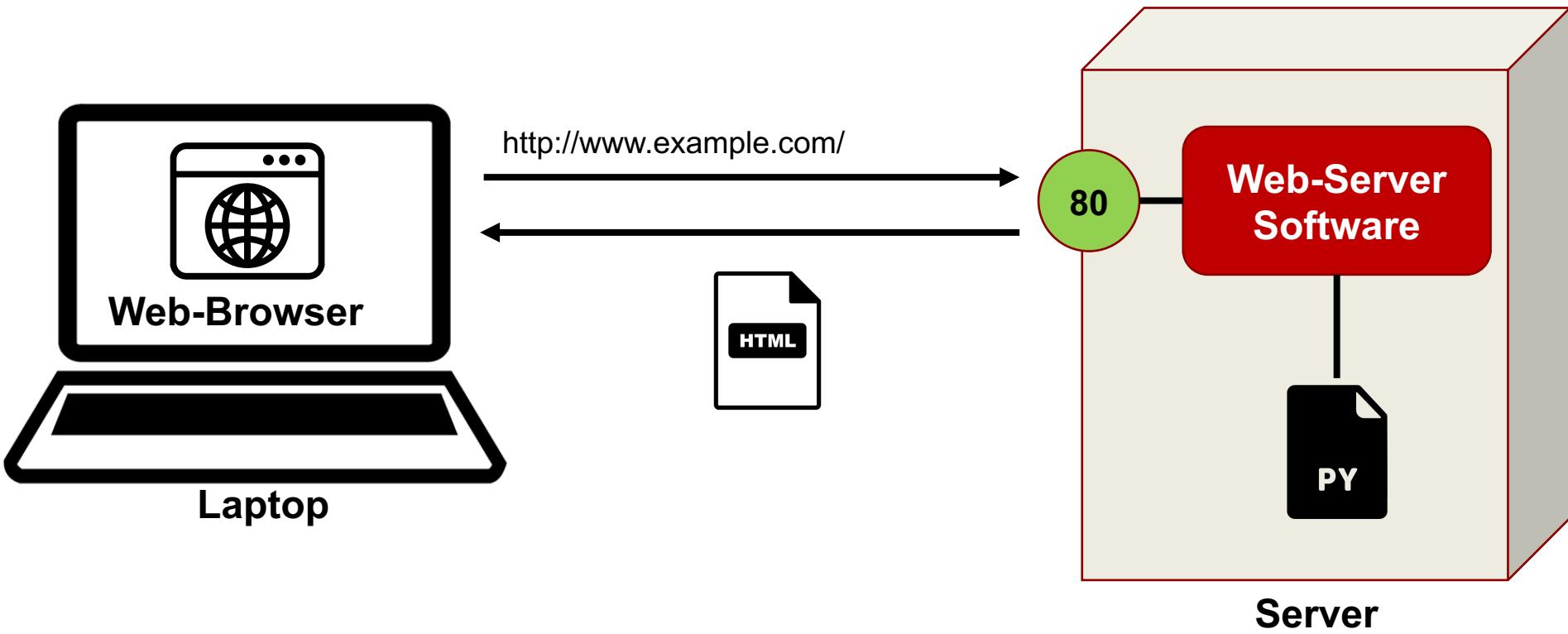
</body>
</html>
```

The result preview on the right shows a light blue background with a centered white **BIPM is awesome** heading and a black "Yes it is." paragraph below it.

Virtual Private Server (VPS)

- Multi-tenant cloud hosting
- Virtualized server resources via a cloud provider
- Each VPS is installed on a physical machine, operated by the cloud provider, that runs multiple VPSs
- Infrastructure as a Service (IaaS)
vs.
 - Platform as a Service (PaaS)
 - Software as a Service (SaaS)
- VPS hosting provider
 - Digital Ocean Droplets
 - Hetzner Cloud Server
 - Amazon Lightsail
 - Amazon EC2
 - Azure Virtual Machines
 - Google Compute Engine
 - and many more ...

Server-side Dynamic Website



Jinja: Python Template for dynamic HTML generation

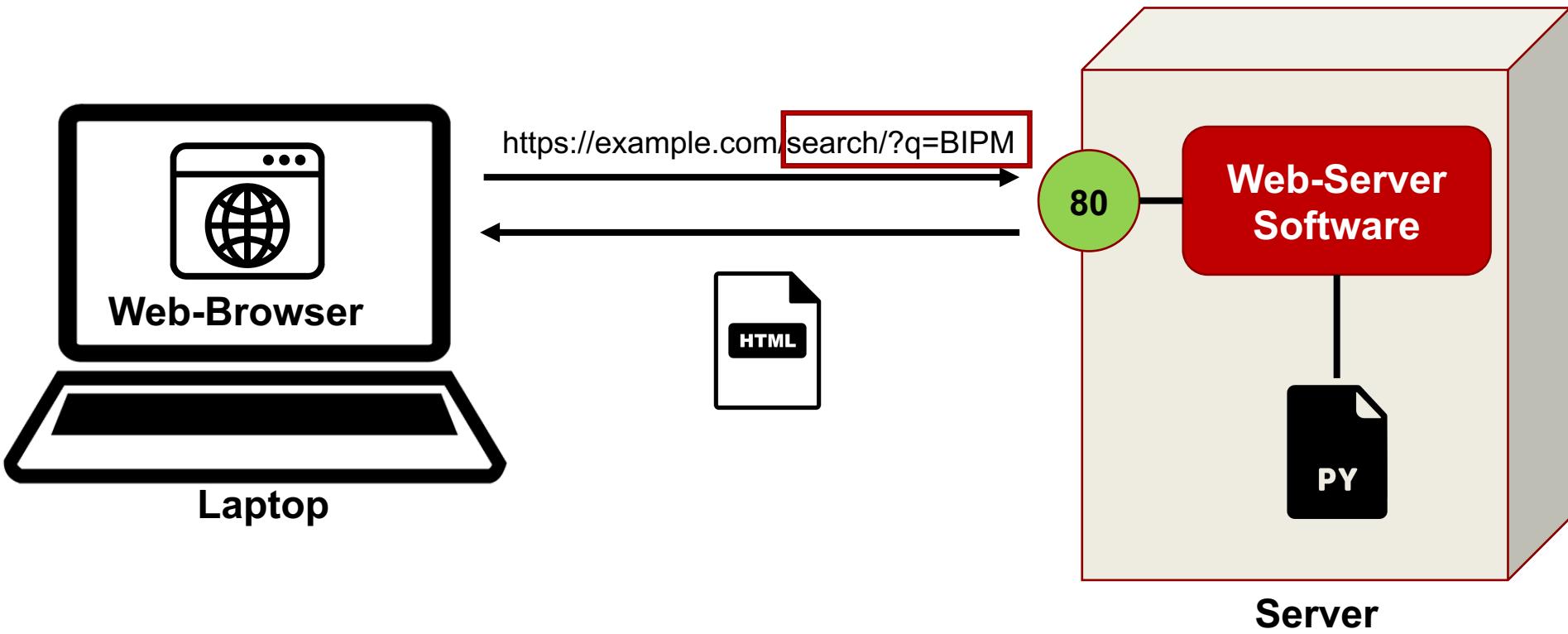
The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** docker - <<hello.html>>
- Toolbar:** Includes standard icons for file operations, Git integration, and search.
- Project Structure:** Shows the project tree under "docker":
 - app
 - static
 - templates
 - hello.html
 - app.py
 - computation.py
 - iwashere.txt
 - requirements.txt
 - docker-compose.yml
 - Dockerfile
 - requirements.txt
- External Libraries
- Scratches and Consoles

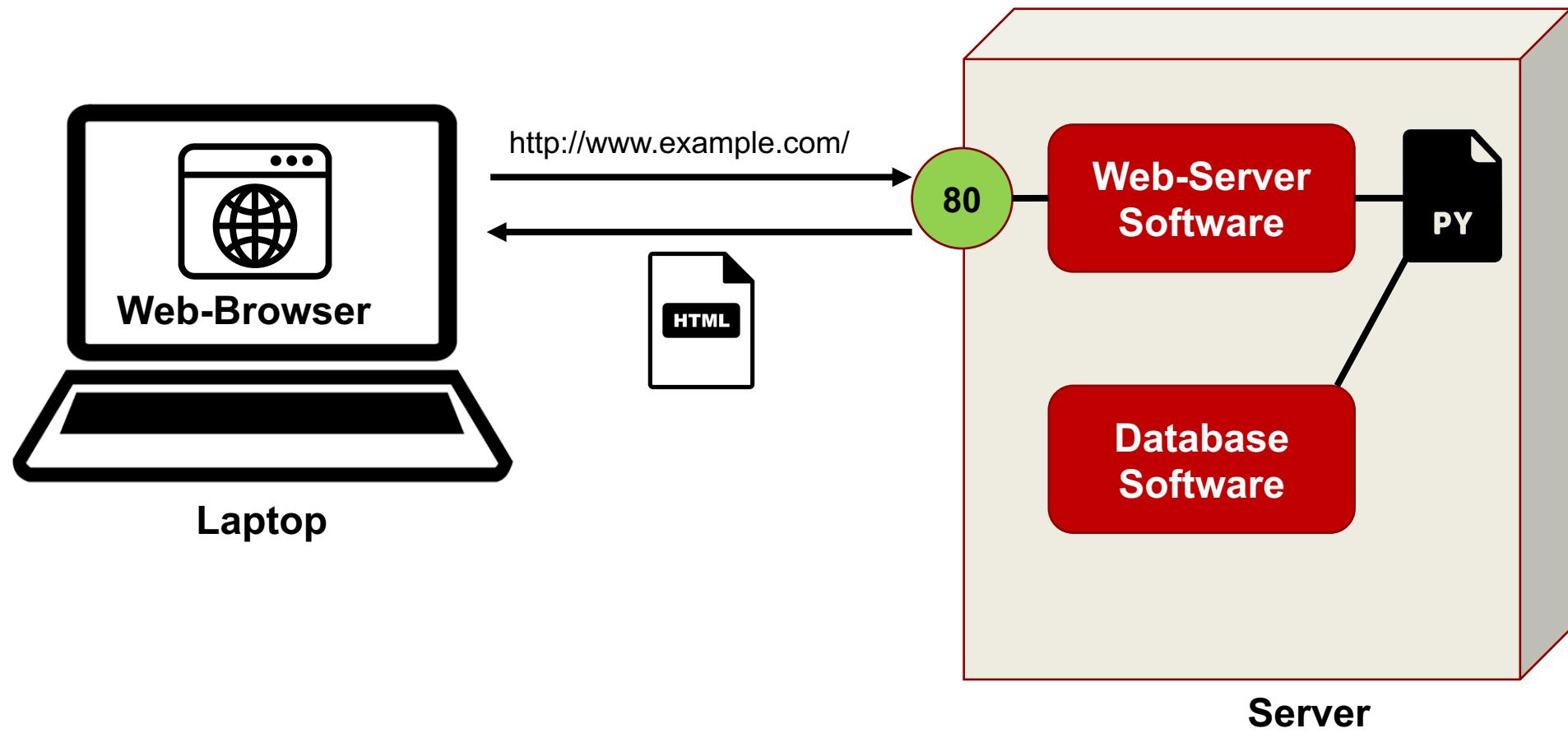
- Code Editor:** The file "hello.html" is open, displaying Jinja template code:

```
<!doctype html>
<html>
  <link rel=stylesheet type=text/css href="{{ url_for('static', filename='style.css') }}>
  <body>
    <h1>I have been seen {{ count }} times.</h1>
    <h2>Hallo</h2>
    <ul>
      {% for i in range(1, count+1) %}
        <li>{{ i }}: Hello {{ name }}! </li>
      {% endfor %}
    </ul>
  </body>
</html>
```
- Bottom Status Bar:** Shows Dockerfile detection, file encoding (UTF-8), and other system information.
- Right Sidebar:** Includes "Key Promoter X", "Database", "SciView", and "make" sections.
- Bottom Navigation:** Problems, Git, Terminal, TODO, Python Console, Event Log.

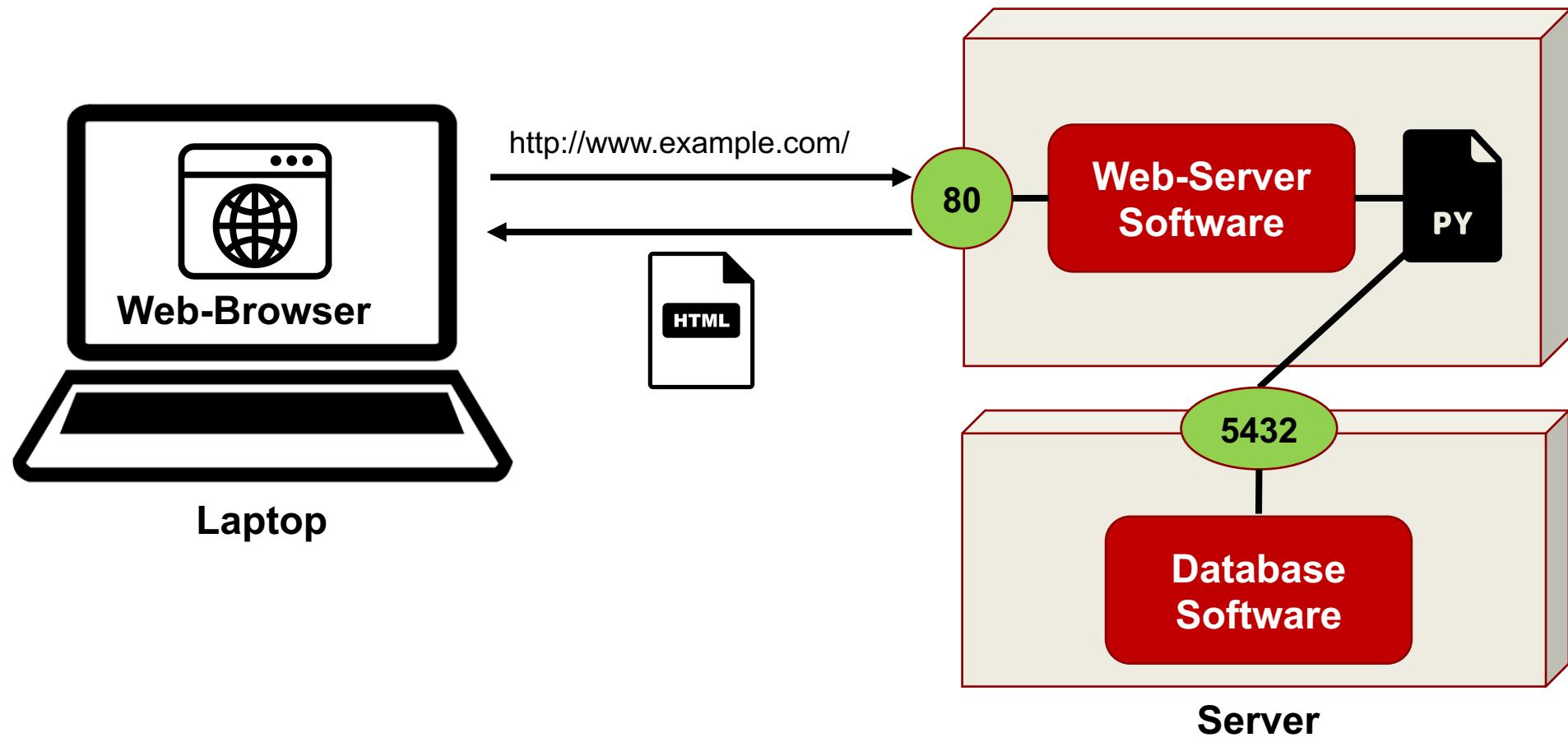
Server-side Dynamic Website



Server-side Dynamic Website



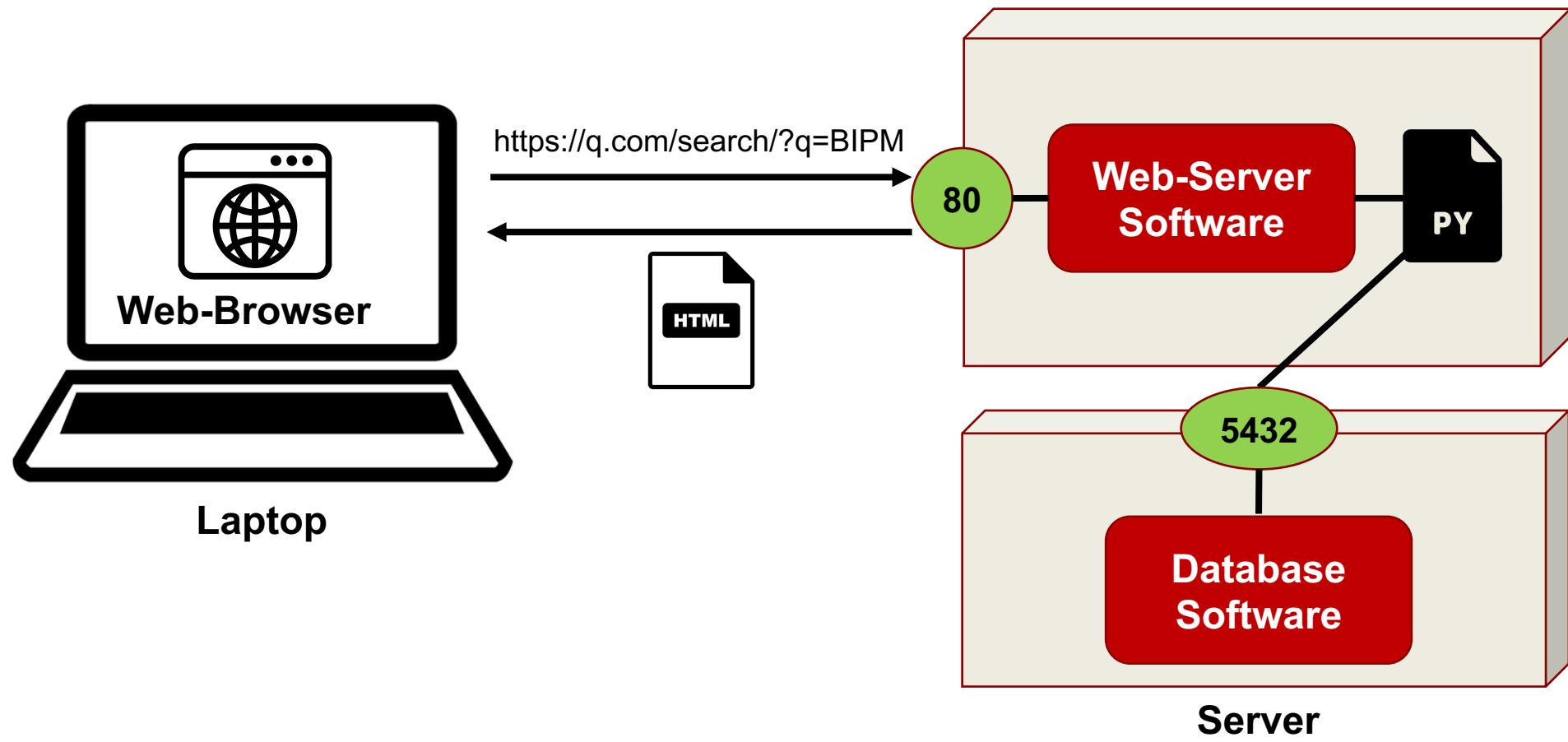
Server-side Dynamic Website



Popular Web Application Frameworks for different Programming Languages

- Python
 - Django
 - Flask
 - FastAPI
- Ruby
 - Ruby on Rails
- PHP
 - Laravel
- JavaScript
 - Node.js
- Java
 - Spring
 - Java Server Faces
- Examples of Websites that have been built with Python (at least partially):
 - Netflix
 - Google
 - YouTube
 - Instagram
 - Uber
 - Pinterest
 - Dropbox
 - Reddit
 - Quora
 - Spotify

Server-side Dynamic Website



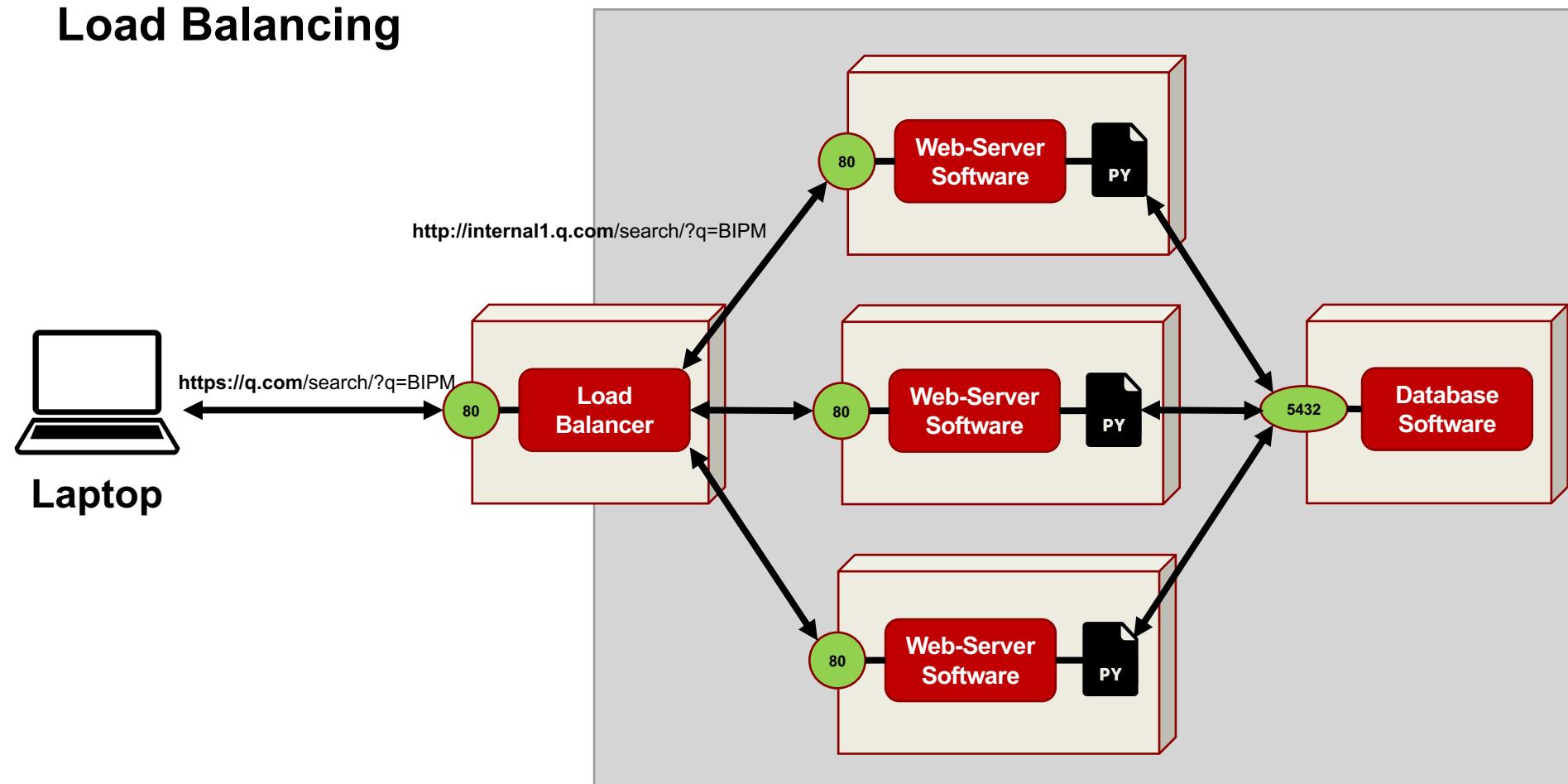
Cache

- Save computational expensive results
- Example: Front Page of New York Times
- “There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors.” — Leon Bambrick
- “There’s two hard problems in computer science: we only have one joke and it’s not funny.” — Phillip Scott Bowden
- Examples of caching systems
 - Memcached
 - Redis
 - Casandra

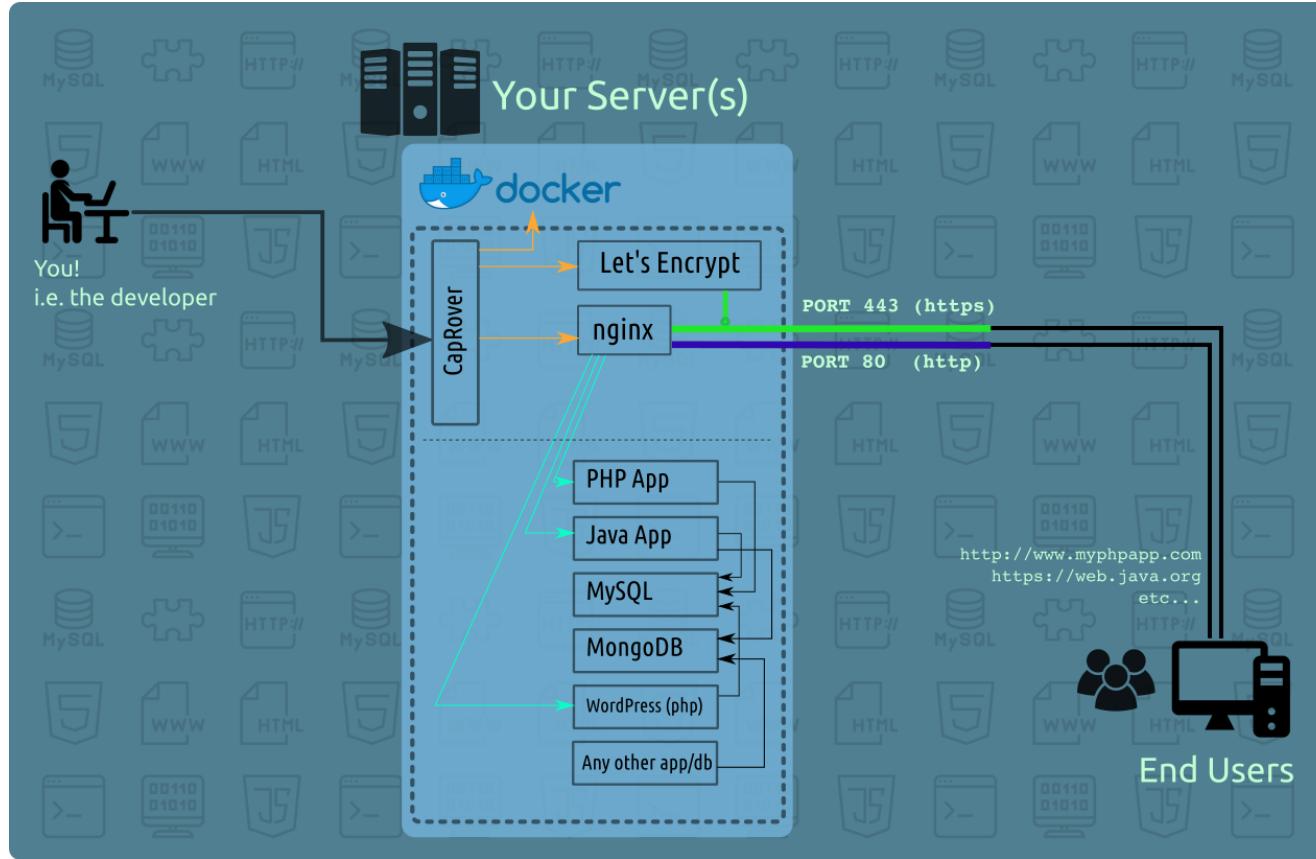
Proxy Server

- Intermediary for requests
- **Reverse Proxy**
 - Forwards to a server and returns the server's response to the client
- **Load Balancer**
 - Distributes requests among a group of servers in each case returning the response from the selected server to the appropriate client
 - Types of Load Balancing
 - Server-side Load Balancer (e.g. with an NGINX webserver)
 - DNS Load Balancing
 - Load Balancing Strategies
 - Round-robin
 - Hashing on IP
 - Least load

Load Balancing



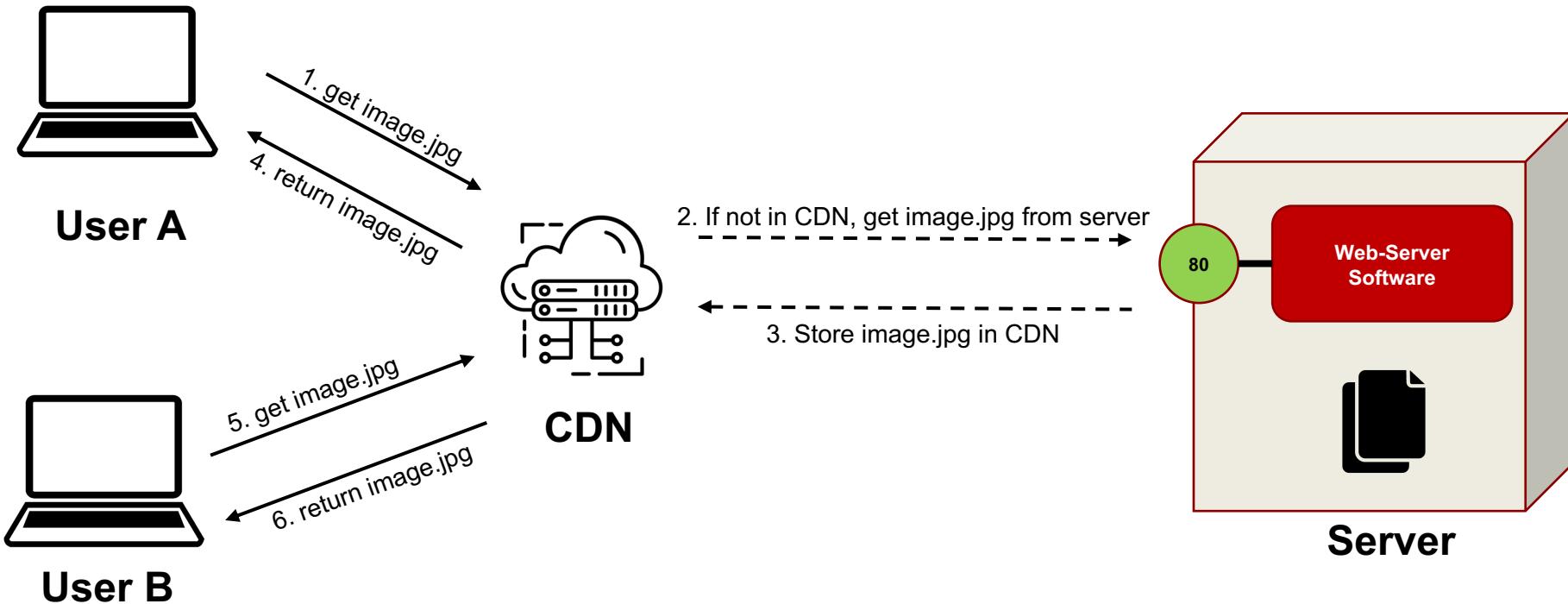
CapRover: Reverse Proxy for Docker Containers with easy HTTPS



Content Delivery Network (CDN)

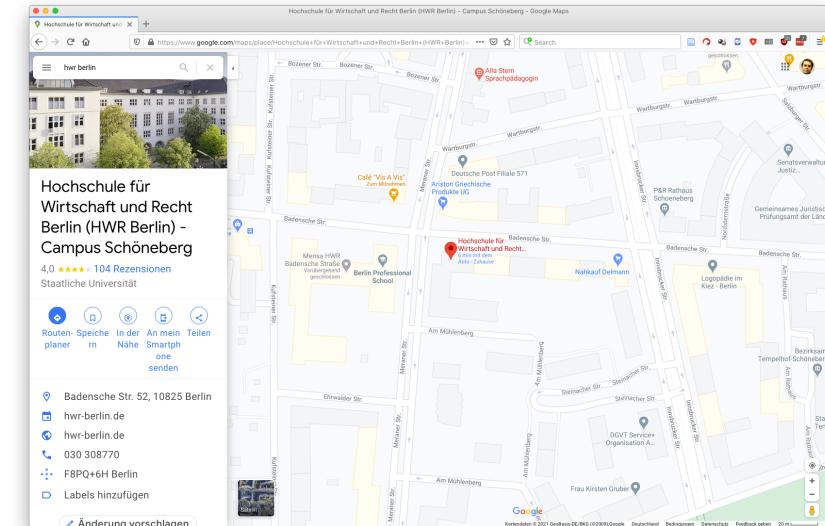
- Geographically distributed network of proxy servers and data centers for delivering static assets
- Goals
 - Reduction of load to original servers
 - Increasing availability
 - Geographically nearer located to end user
- Static Assets: any non-dynamic file like Images, Videos, CSS, JavaScript, Static HTML
- Example CDN Providers
 - Cloudflare
 - Akamai
 - Amazon CloudFront
 - BunnyCDN

Content Delivery Network (CDN)



Examples of highly interactive websites

Google Maps



Miro

A screenshot of a Miro board titled "Big Data Use Cases, Online Whiteboard for Visual Collaboration". The board features several cards and diagrams. One card in the center says "How might we use big data to increase access to affordable, reliable, and sustainable energy?". Another card shows a scatter plot with axes "High Performance" and "Low Cost". A third card features a globe with the text "Save the planet". The interface includes a toolbar on the left with various drawing and selection tools, and a navigation bar at the bottom.

JavaScript

- Programming Language
 - Supports imperative, object-oriented and function paradigm
 - Weakly typed
 - Syntax looks like Java and C (a lot of curly brackets and semicolons)
- Can run
 - Inside any web browser
 - On a server (via Node.js)
- Inside Web Browser
 - Can change the website without a reload from web server
 - Combination of HTML, CSS and JS

Client-side Dynamics with JavaScript

The screenshot shows a web-based code editor interface. At the top, there's a toolbar with standard browser icons like back, forward, search, and refresh. Below the toolbar, the title bar says "Tryit Editor v3.6". The main area has a "Run" button on the left. On the right, it displays "Result Size: 428 x 267". The code editor contains the following HTML and JavaScript:

```
<!DOCTYPE html>
<html>
<body>

<h3>BIPM JavaScript</h3>

<button type="button"
onclick="document.getElementById('bipm').innerHTML = 'BIPM is awesome!'">
How is BIPM?</button>

<p id="bipm"></p>

</body>
</html>
```

To the right, a preview window shows the resulting page with the heading "BIPM JavaScript" and a button that, when clicked, changes the text inside a paragraph to "BIPM is awesome!". There is also a "How is BIPM?" link.

Client-side Dynamics with JavaScript

The screenshot shows a web-based code editor interface. At the top, there's a toolbar with standard browser icons like back, forward, search, and refresh. Below the toolbar, the title bar says "Tryit Editor v3.6". The main area has a "Run" button highlighted in green. To the right, it displays "Result Size: 428 x 267". The code editor contains the following HTML and JavaScript:

```
<!DOCTYPE html>
<html>
<body>

<h3>BIPM JavaScript</h3>

<button type="button"
onclick="document.getElementById('bipm').innerHTML = 'BIPM is awesome!'">
How is BIPM?</button>

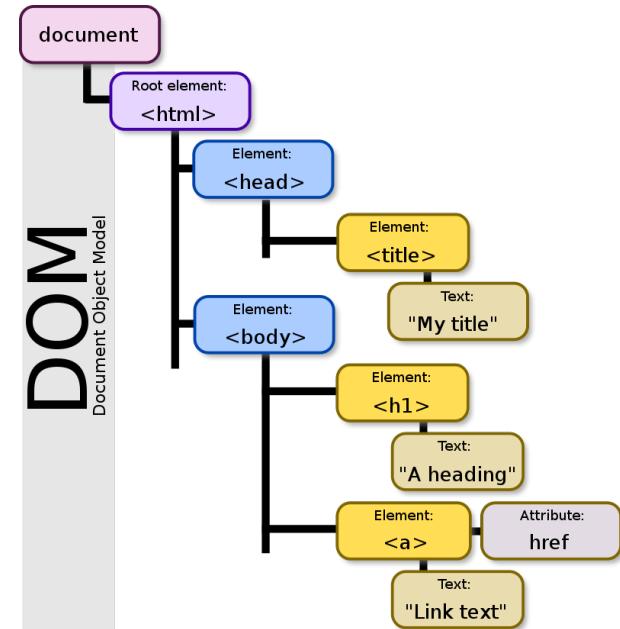
<p id="bipm"></p>

</body>
</html>
```

The result panel on the right shows the output of the code. It features a large heading "BIPM JavaScript" and a button labeled "How is BIPM?". Below the button, the text "BIPM is awesome!" is displayed.

Document Object Model (DOM)

- DOM represent the tree of HTML tags
- Browser provides a programming interface for the DOM
- Attach event handlers to trigger user events
- Manipulate DOM



JavaScript Front-end Libraries

- Simplifies the programming of highly interactive client-side websites
- Single-Page Application (SPA)
 - Dynamically rewriting the current DOM with new data from the web server
 - Communication with the server through Web APIs and JSON (JavaScript Object Notation)
 - Advantages: no full-page reload, highly interactive websites
 - Disadvantages: high complexity and SEO problems
- Examples of JavaScript Front-end Libraries
 - React
 - Angular
 - Svelte
 - Vue.js

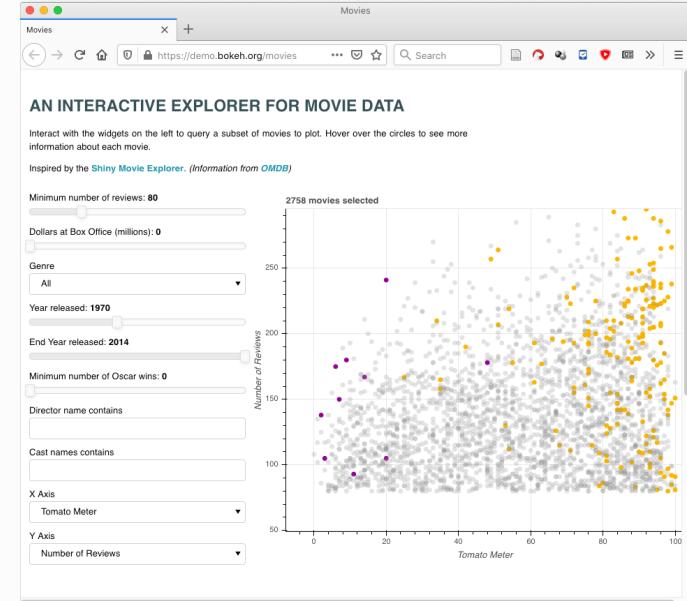
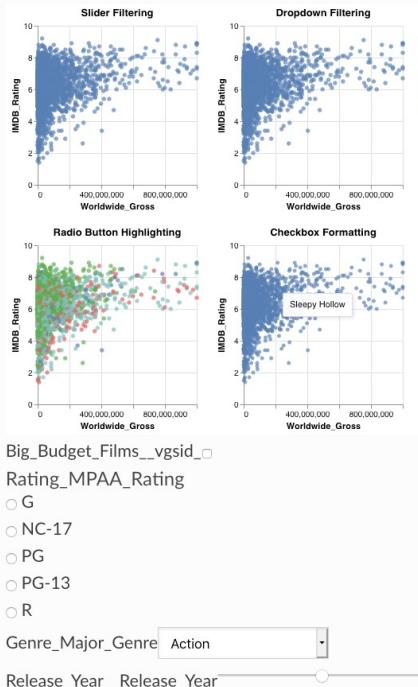
Interactive Python Data Visualization

■ Python Libraries that generate JavaScript

■ Examples

- Altair
- Bokeh
- Plotly Express
- Plotly Dash
- Panel

■ No need to learn JavaScript



Mobile Apps

- **Native app**
 - Run natively on smartphone, tablet, or watch
 - Can run offline
 - Easier access of all capabilities like camera, GPS, etc.
 - Programming Languages
 - iOS Apps: Swift, Objective-C
 - Android Apps: Java, Kotlin
- **Hybrid app / Cross platform**
 - Often a mix of native and web app
 - Example Technologies: React Native, Flutter, Svelte Native, NativeScript, Apache Cordova
- **Web-based app**
 - Coded in HTML, CSS or JavaScript
 - Sharable with a link (do not have to be installed)
- Native Apps and Hybrid Apps are distributed app stores like Apple App Store or Google Play
- Communication with the server through Web APIs and JSON (JavaScript Object Notation)