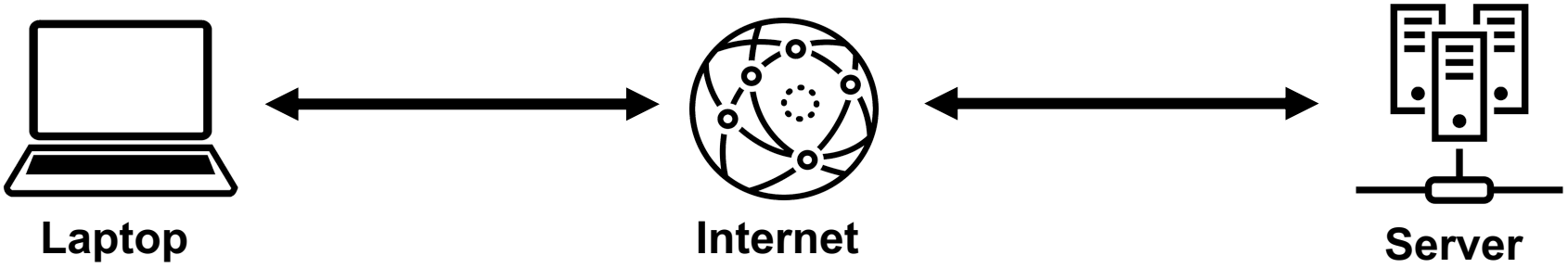
System Design Concepts

Enterprise Architectures for Big Data


I didn't invent
the Internet
I did not
invent the Web

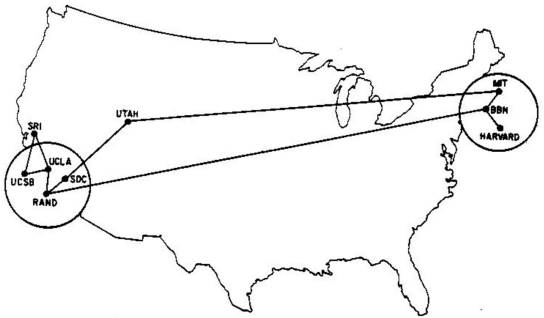

I invented
the Web
I invented
the Internet

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

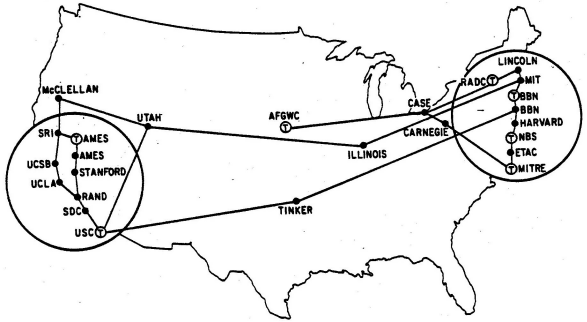


ARPANET (starting 1969)

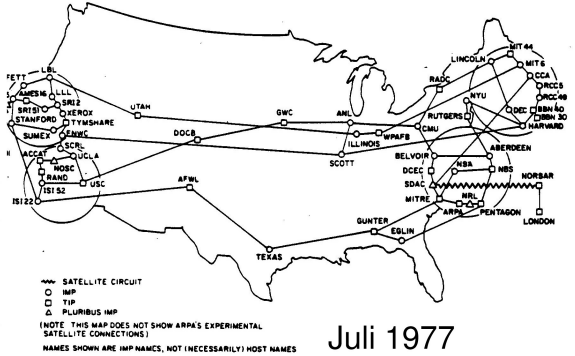
Dezember 1969



Juni 1970

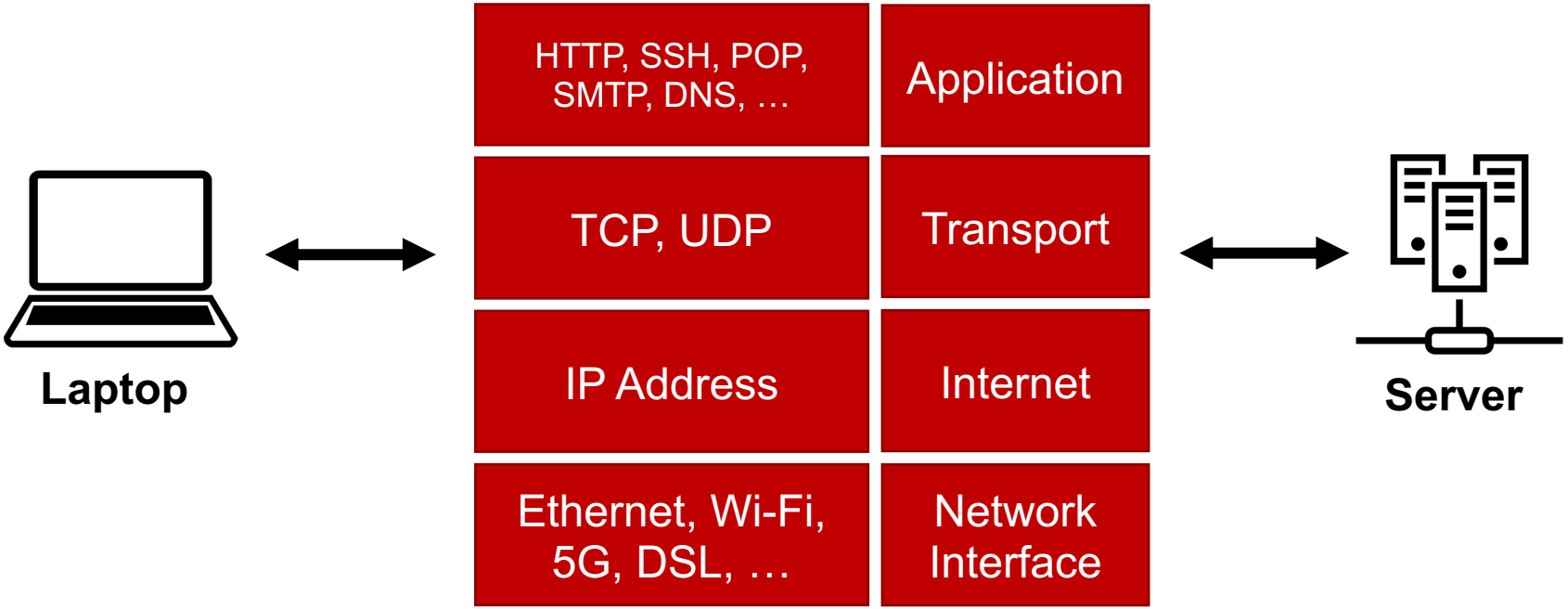


März 1972



Juli 1977

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



Internet Protocol Stack

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

- `traceroute 130.216.159.127`

rolandmueller — traceroute 130.216.159.127 — 105×36

```
[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-0-0-0-500.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  * * *
```

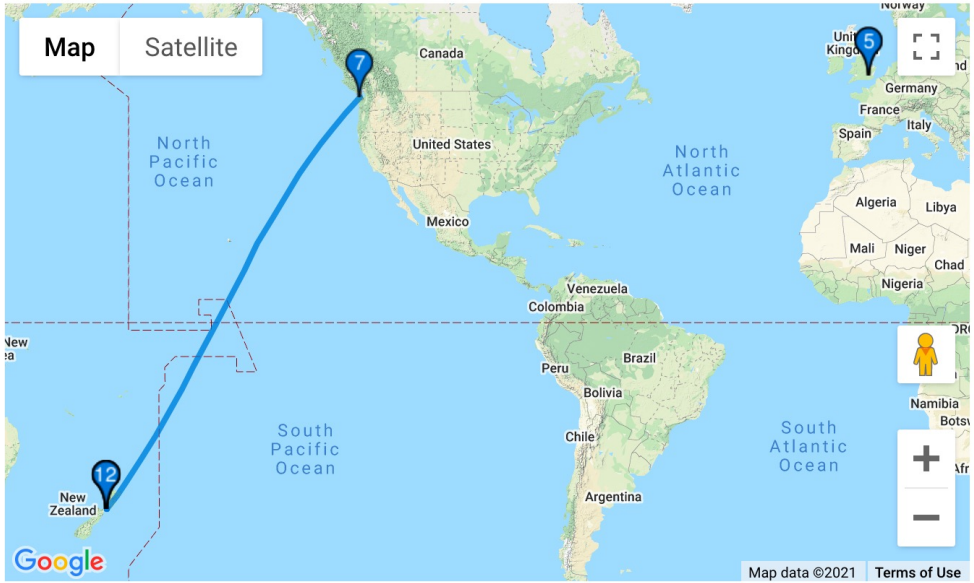
Visual traceroute - Track and map the full path of an IP packet

Visual traceroute - Track and m

https://gsuite.tools/tracero

Search

G Suite.Tools



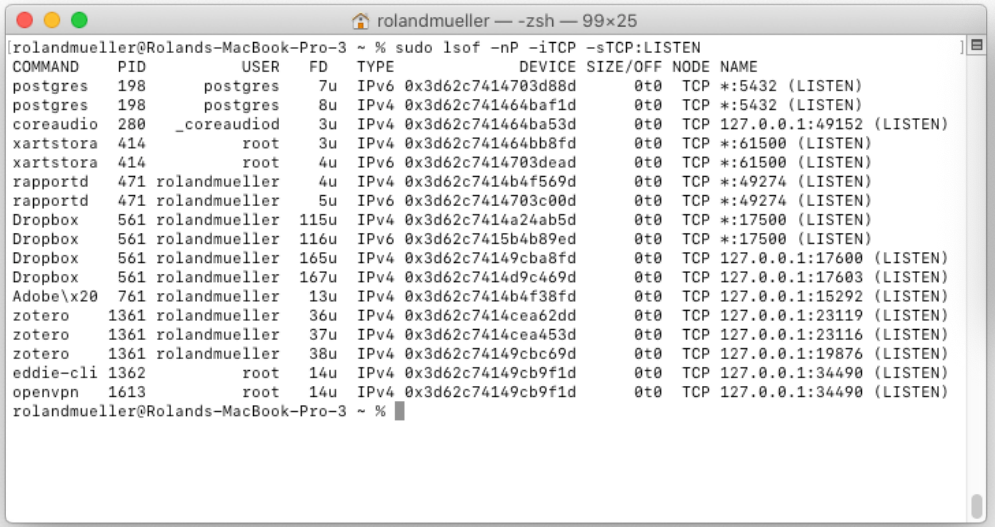
traceroute to 130.216.159.127 (130.216.159.127), 30 hops max

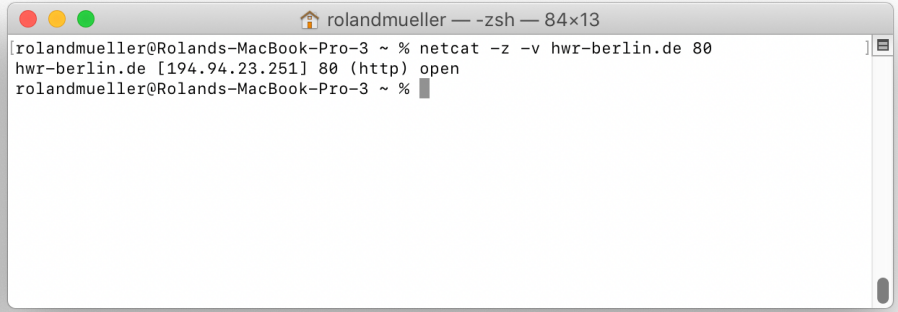
Hop	IP	Time (ms)
1	83.136.248.1	0.125ms
2	100.69.2.209	0.277ms

TCP Port Numbers

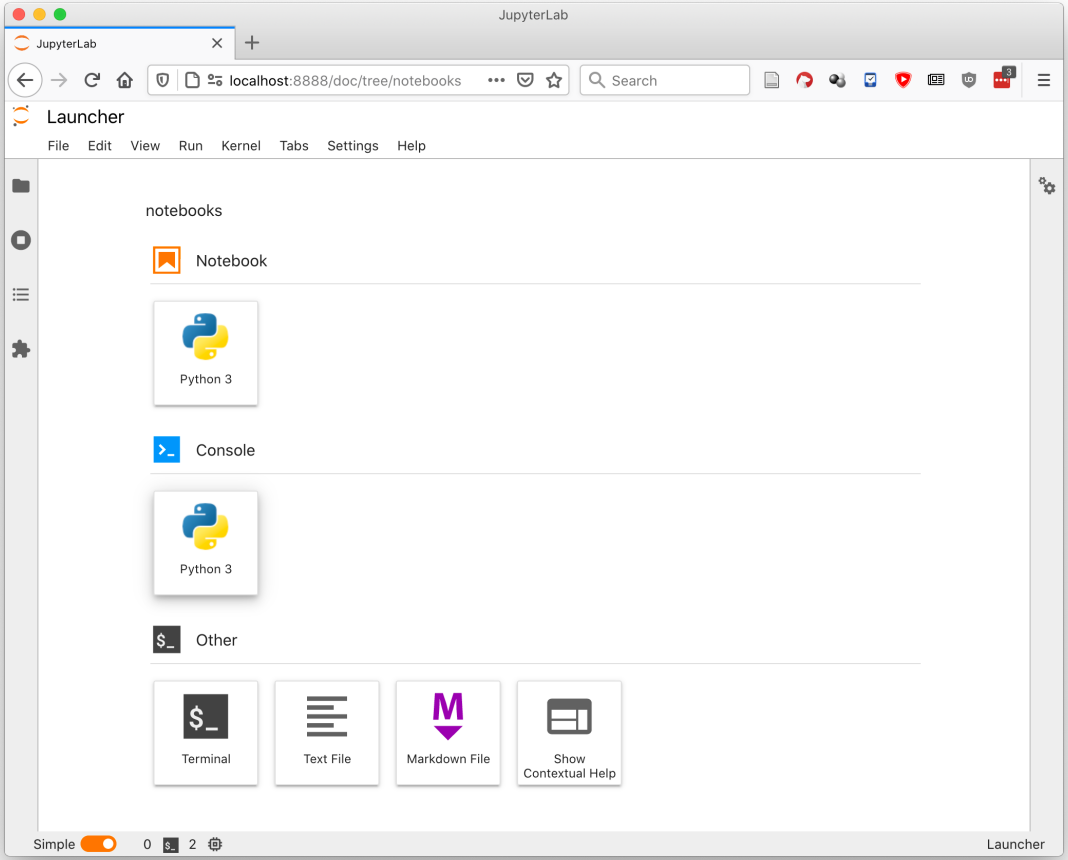
- Port numbers for matching services (protocols)
- Official assignments
- Unofficial use especially of higher port numbers
  - e.g. Jupyter, PostgreSQL, MySQL, Hadoop, …
- Most important Protocols with Port Numbers:
  - Hypertext Transfer Protocol (**HTTP**): 80
  - Hypertext Transfer Protocol Secure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

Check what program is listing on what port number





Port Numbers for distinguishing different Services



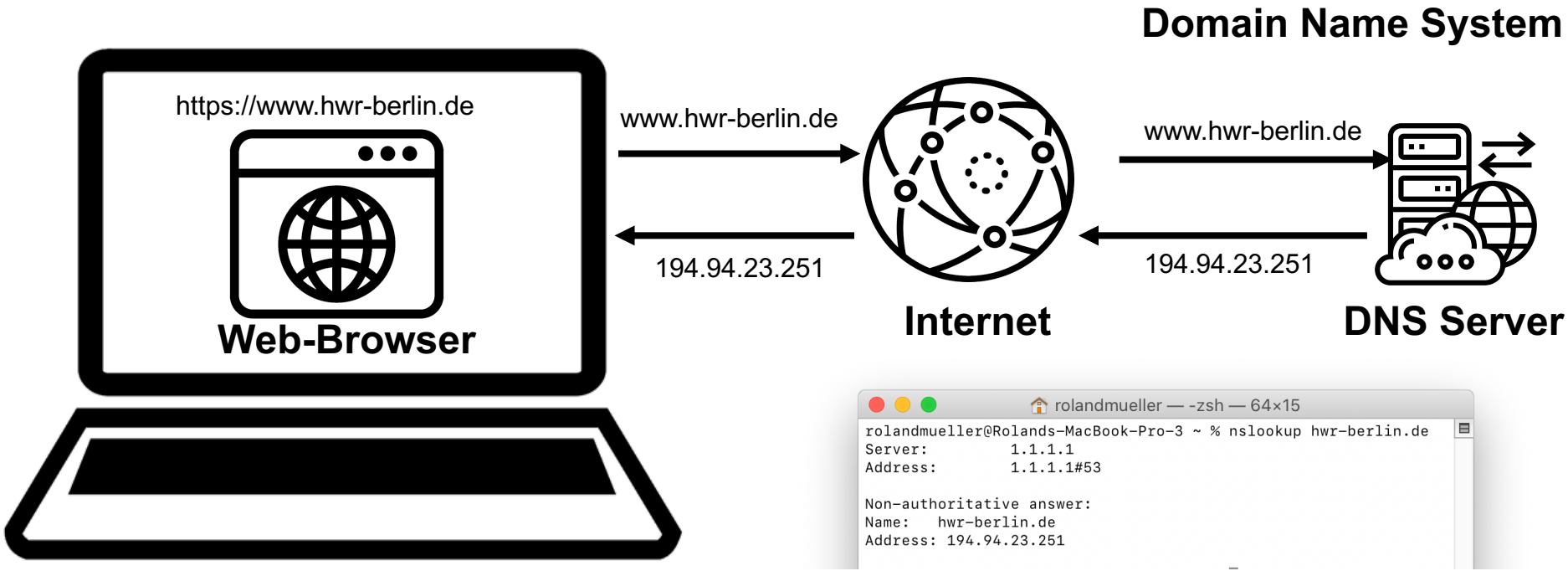
Firewall

- Monitors and controls network traffic

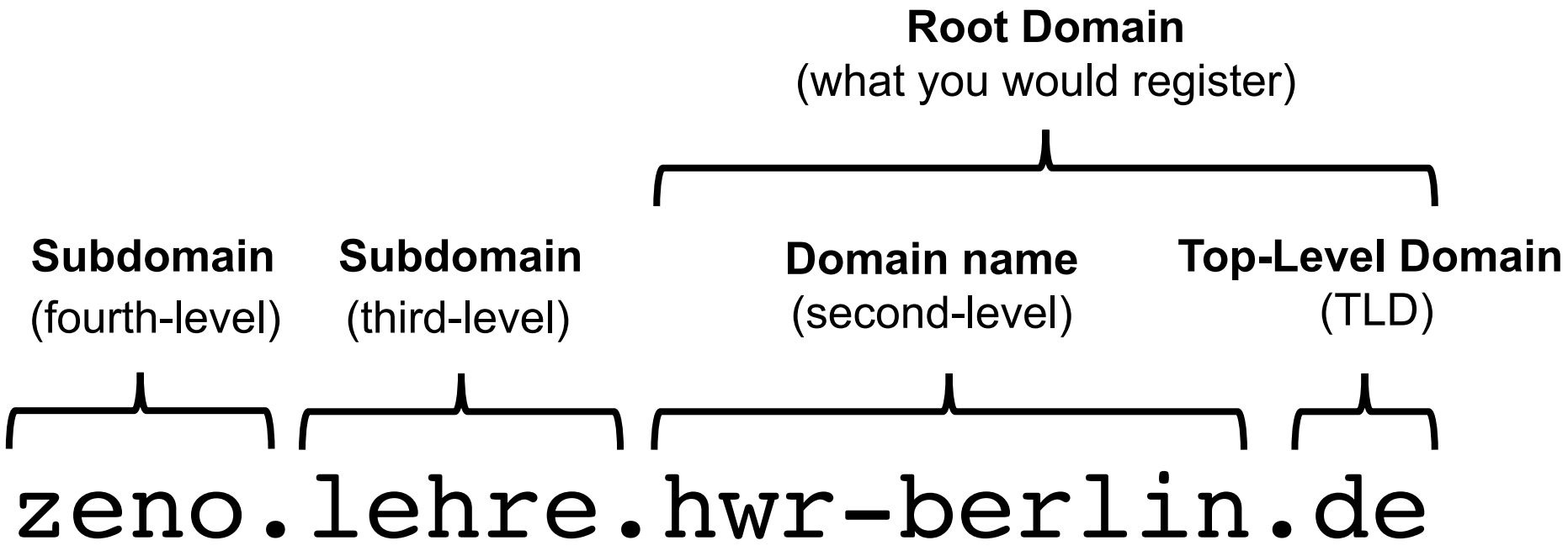
- Can run on and shield
  - a single computer (endpoint-based application firewalls)
  - or a subnetwork (demilitarized zone)

- 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?

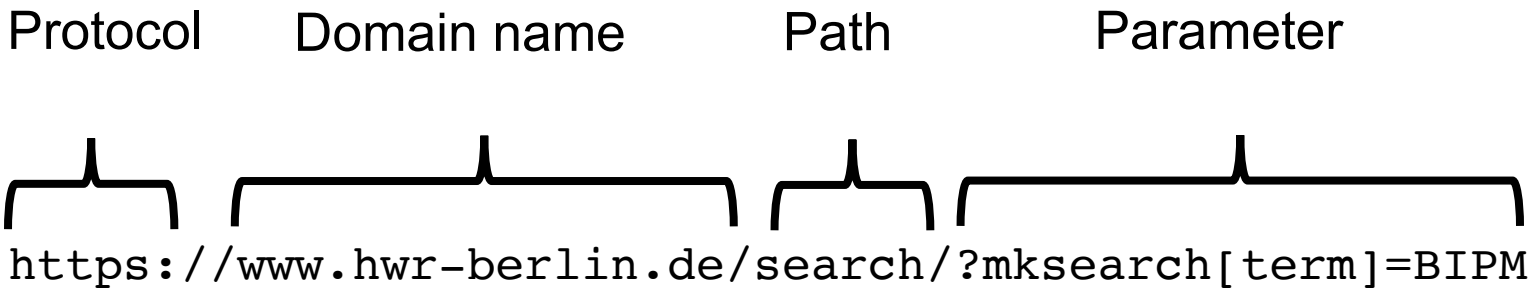


Domain name


Root Domain
(what you would register)
Subdomain
(fourth-level)
Subdomain
(third-level)
Domain name
(second-level)
Top-Level Domain
(TLD)
zeno.lehre.hwr-berlin.de

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)

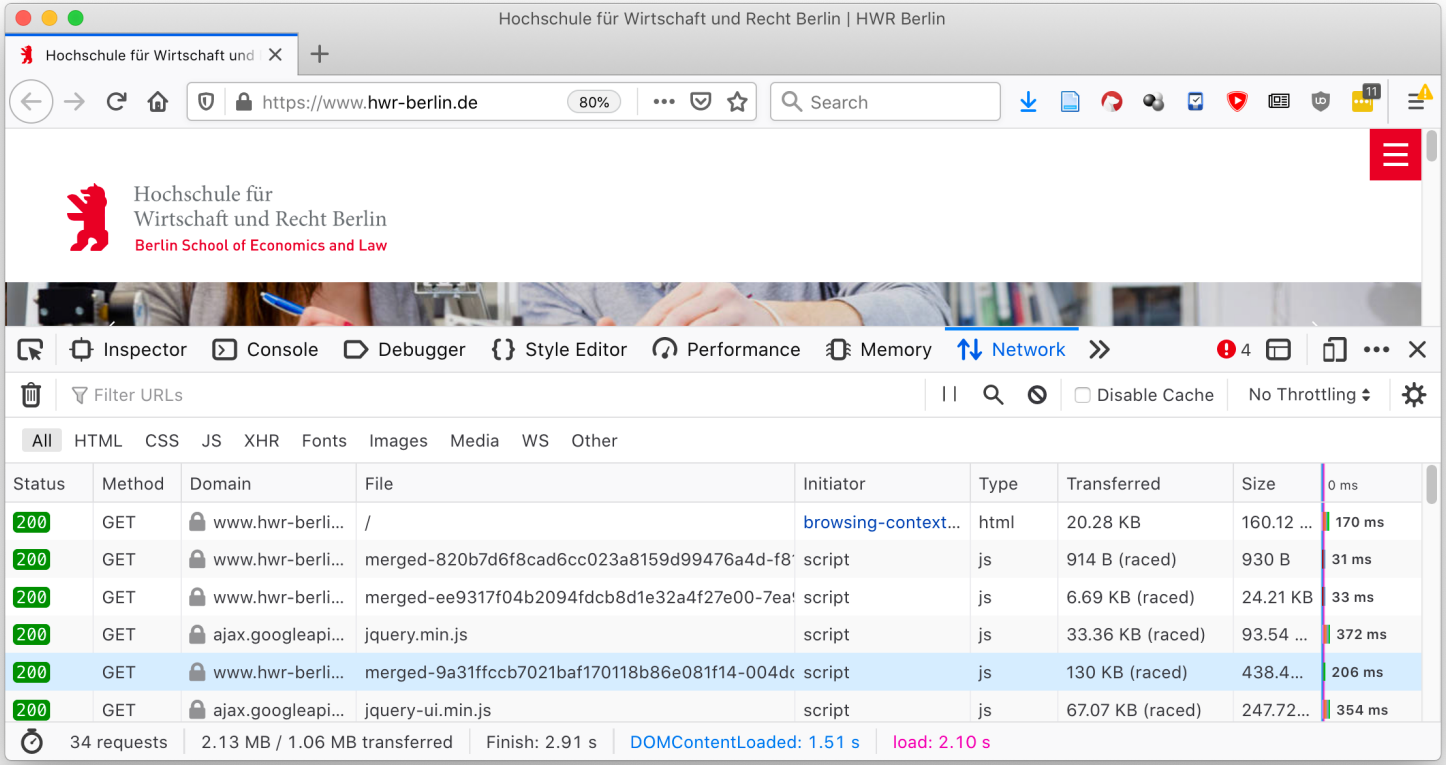
```
http://example.com/user/?name1=value1&name2=value2
```

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

Post

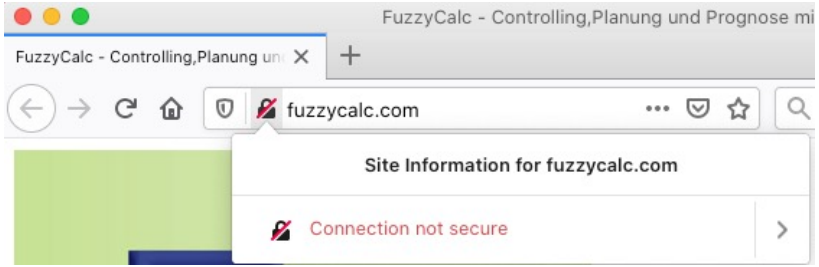
- Submit a form in the browser
- Values: No restriction
- No length restrictions
- Never cached
- Cannot be bookmarked
- Reload: Data will be resubmitted
- Form submit could also be Get (e.g. in a search engine)

HTTP: Hypertext Transfer Protocol
Web Developer Views (Firefox) or Developer Tools (Chrome)

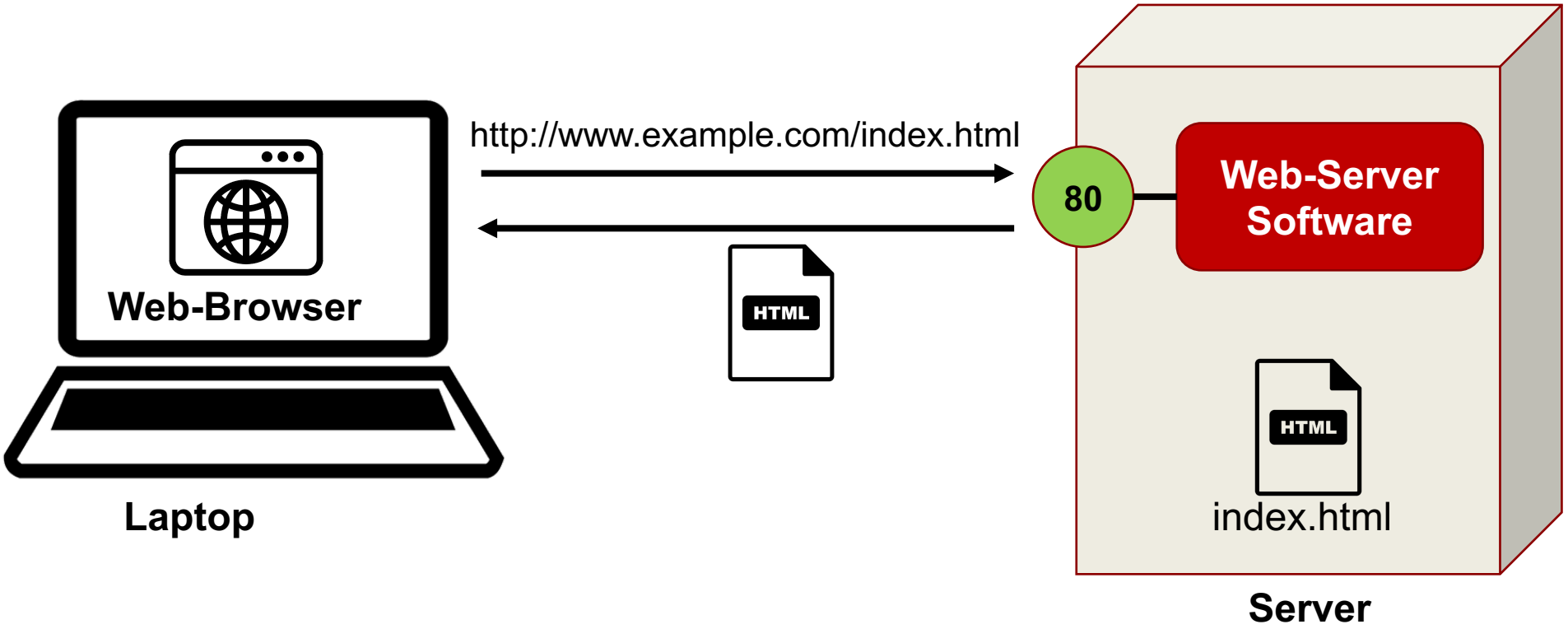


HTTPS: Hypertext Transfer Protocol Secure

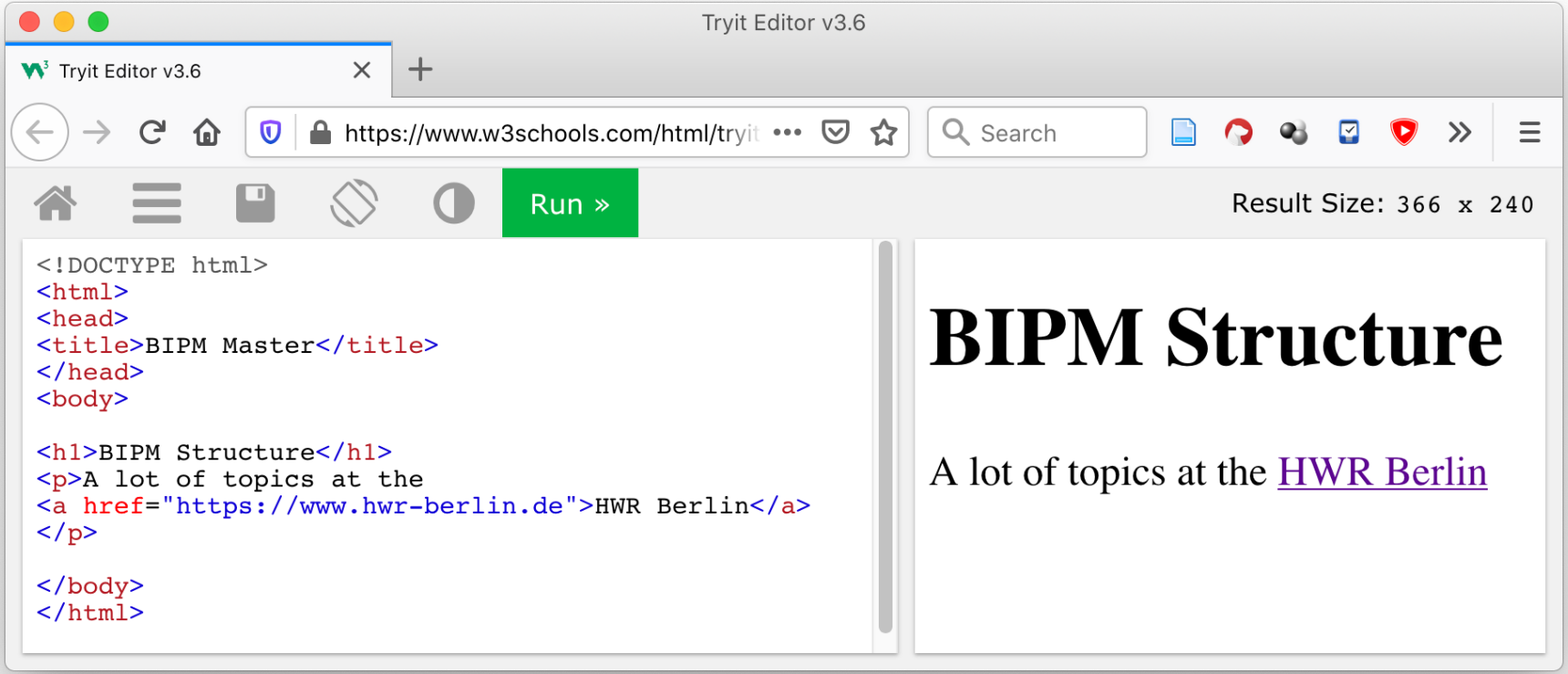
- Uses encryption for securing the transportation between server and browser
  - Uses public-key protocols (Diffie–Hellman key exchange) to exchange a common random and unique session key
  - Uses symmetric-key algorithm for encrypting the communication
- Transport Layer Security (TLS) / SSL (Secure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



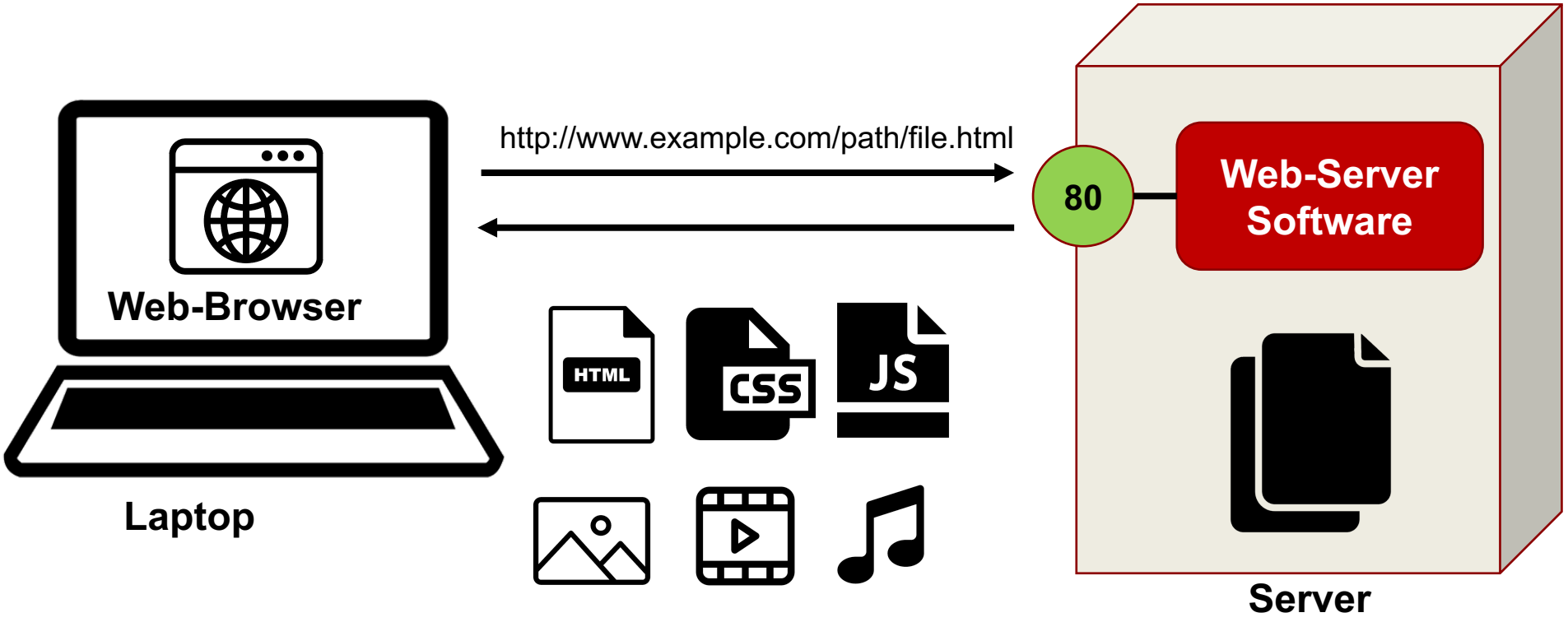
Most simple Web Page: Static Website



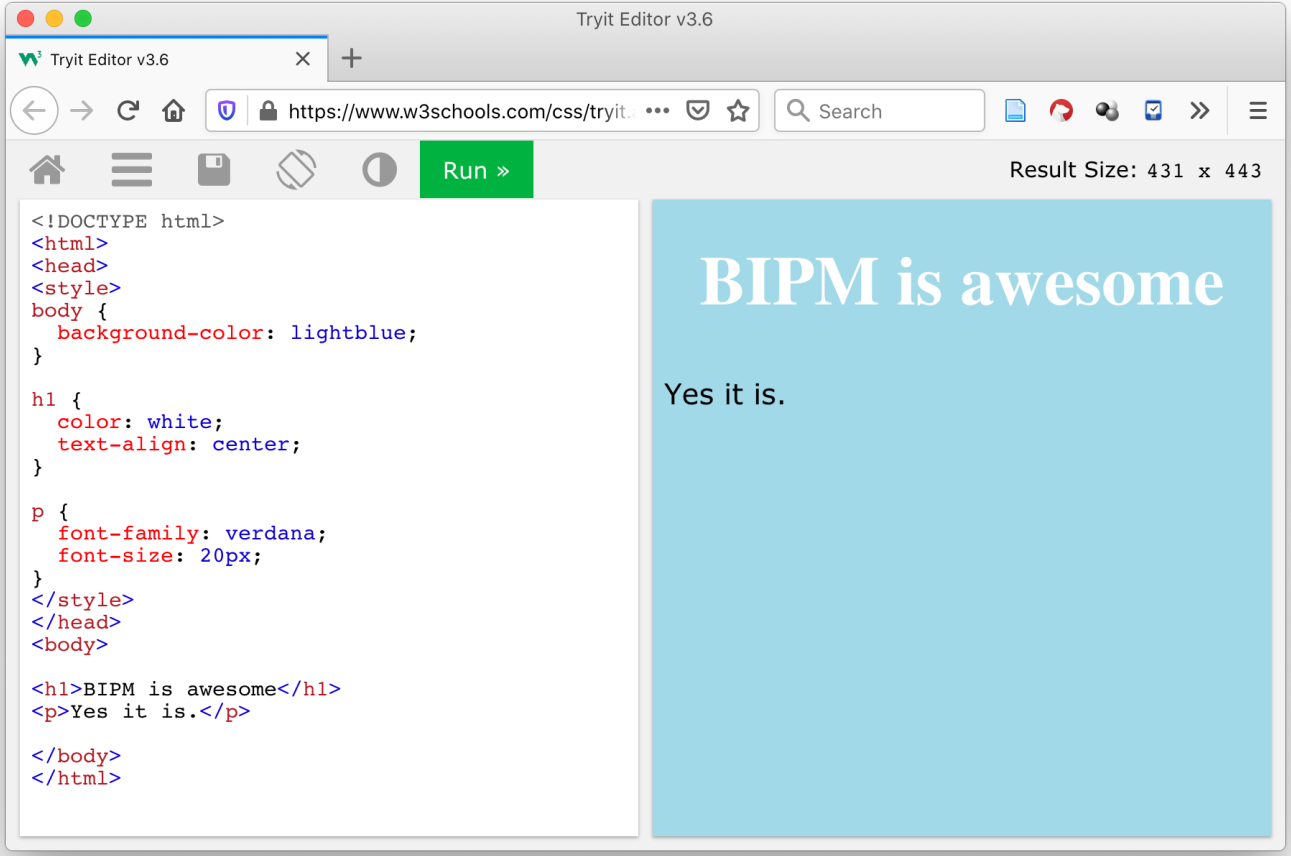
HTML: Hypertext Markup Language



Most simple Web Page: Static Website



CSS: Cascading Style Sheets



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)

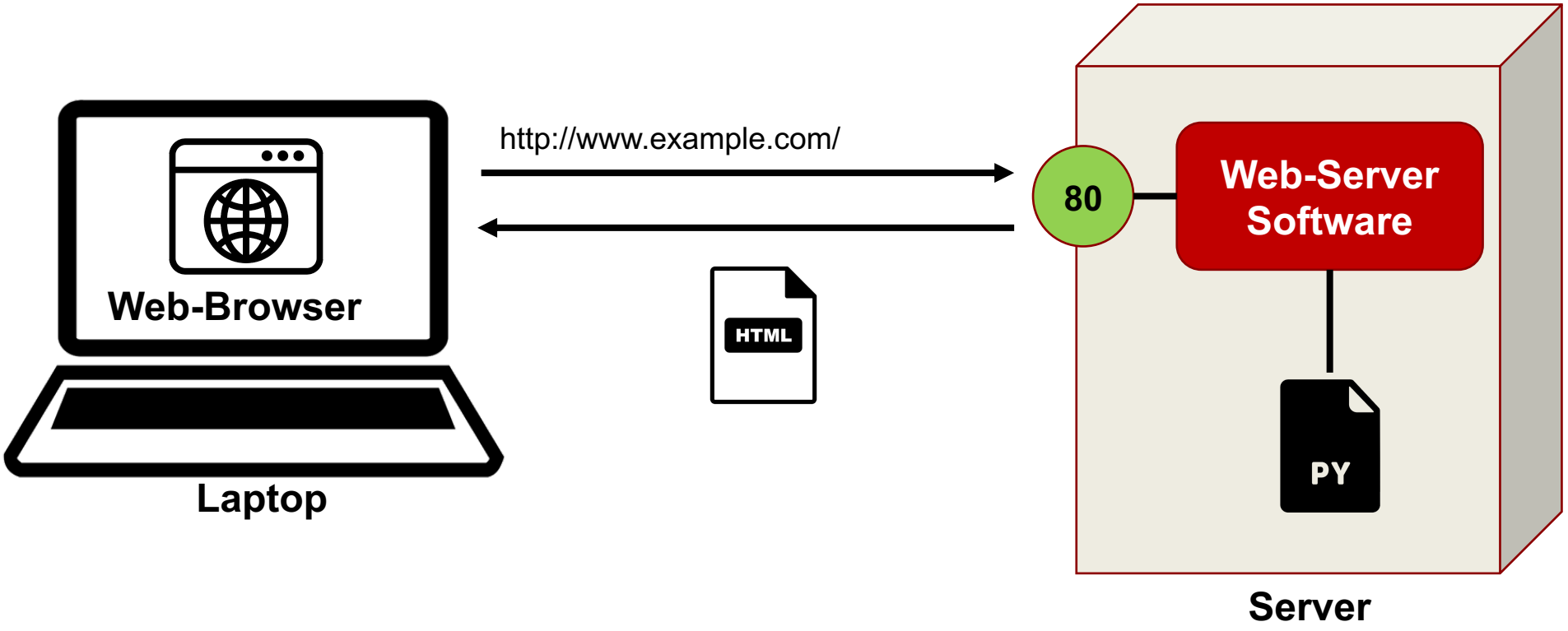
vs.

- Platform as a Service (PaaS)
- Software as a Service (SaaS)

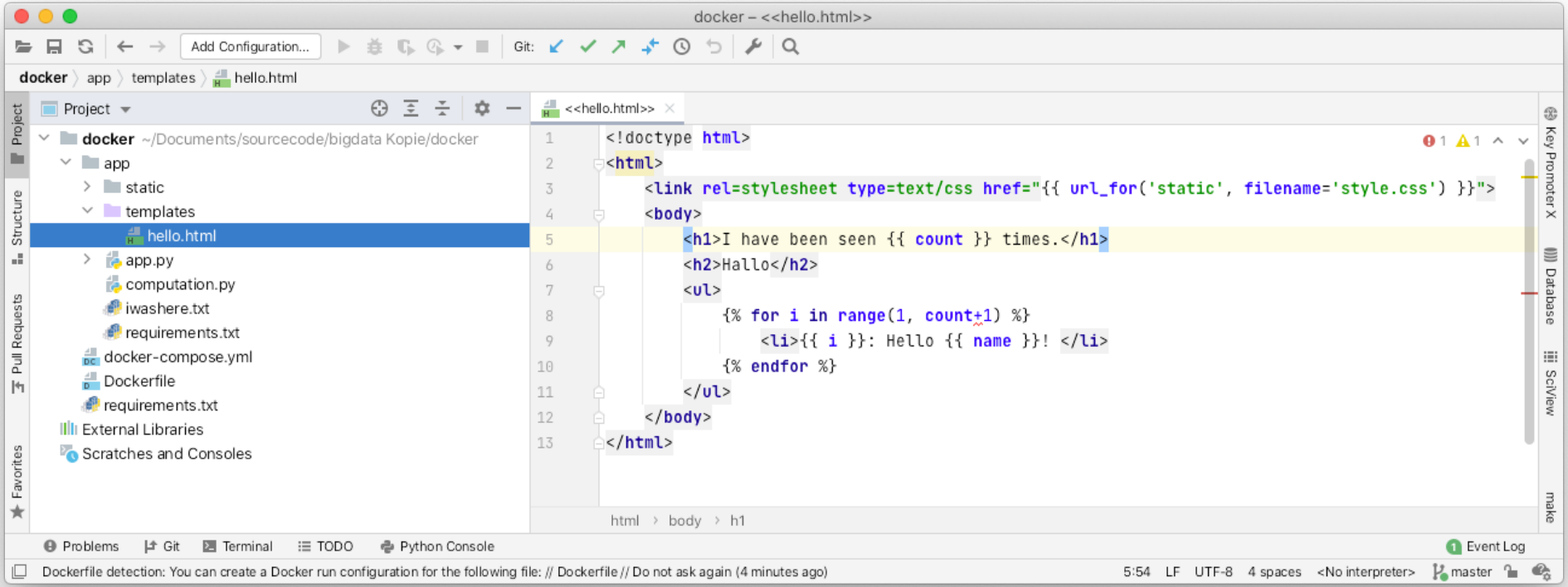
- VPS hosting provider
  - Digital Ocean Droplets
  - Hetzner Cloud Server
  - Amazon Lightsail
  - Amazon EC2
  - Azure Virtual Machines
  - Google Compute Engine

and many more ...

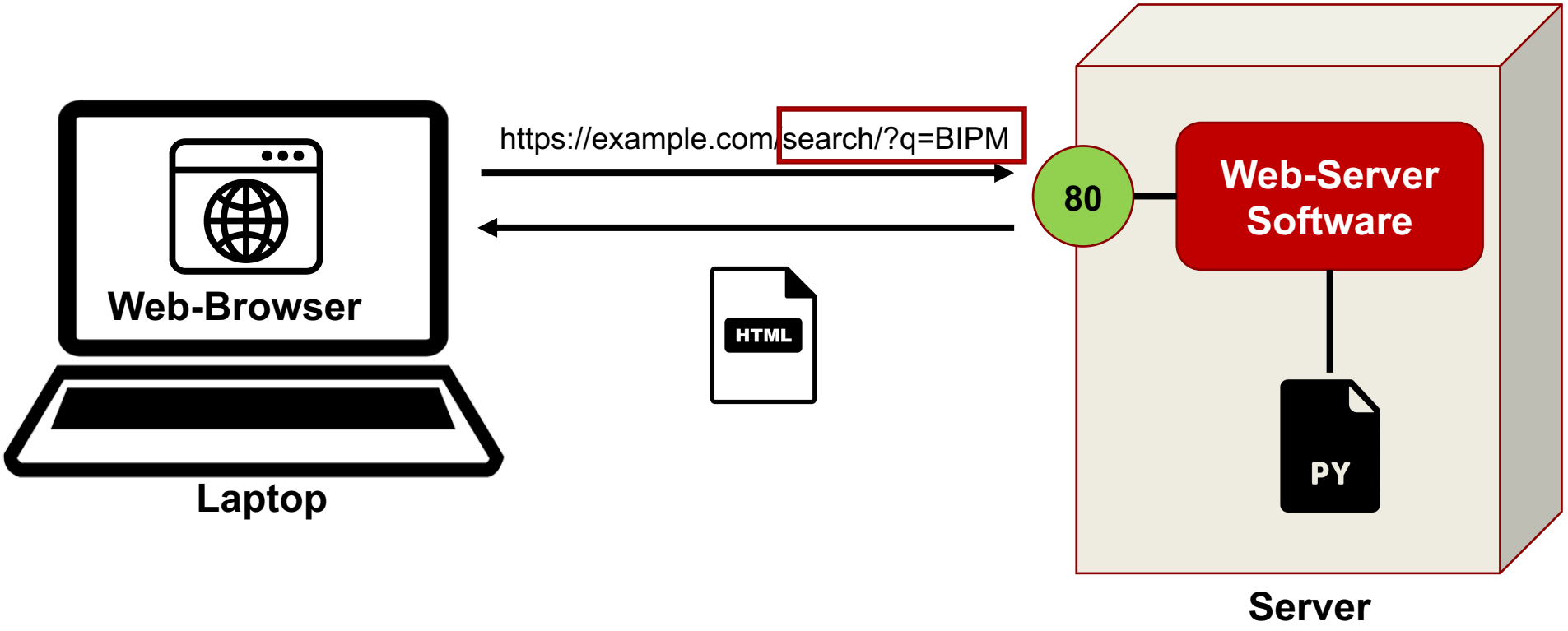
Server-side Dynamic Website



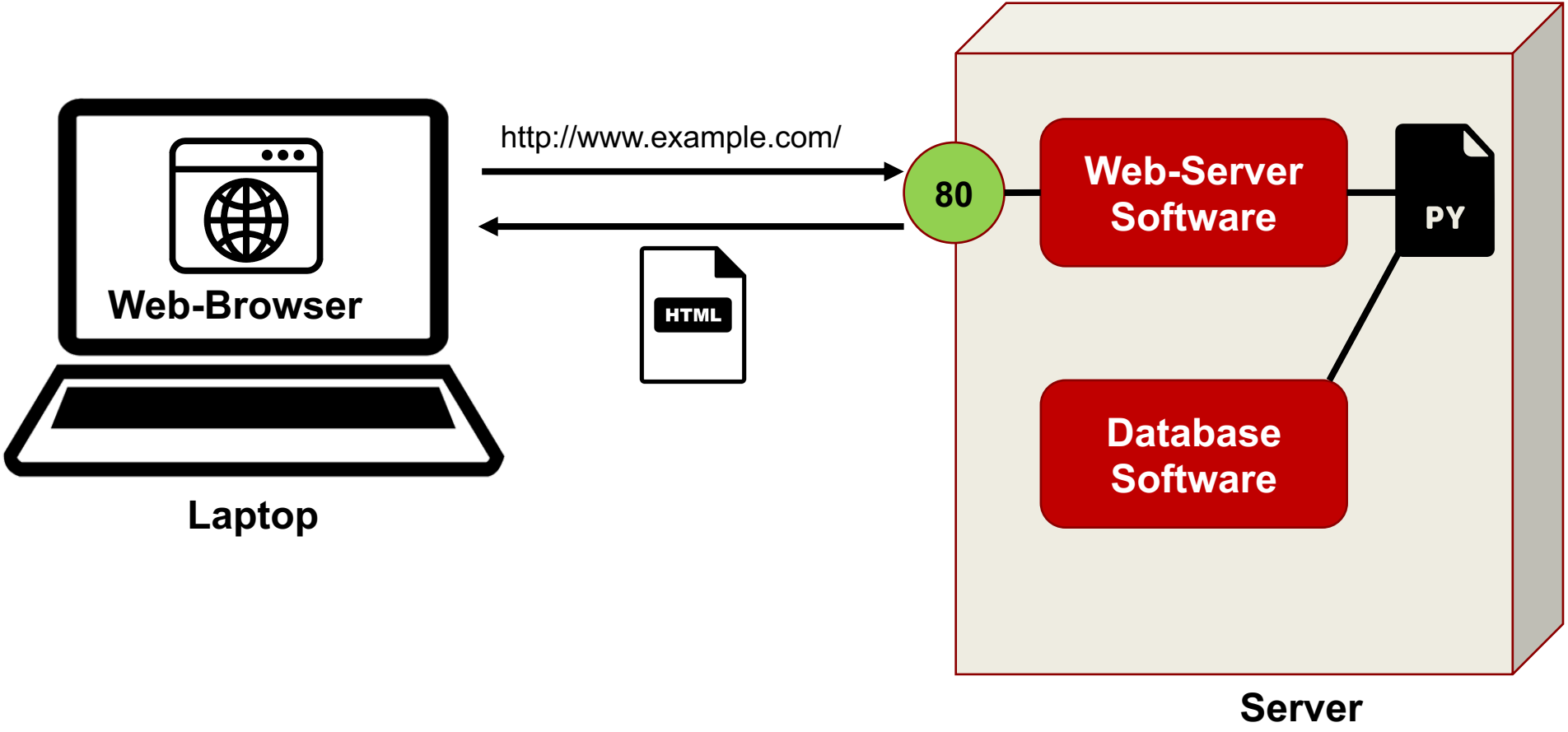
Jinja: Python Template for dynamic HTML generation



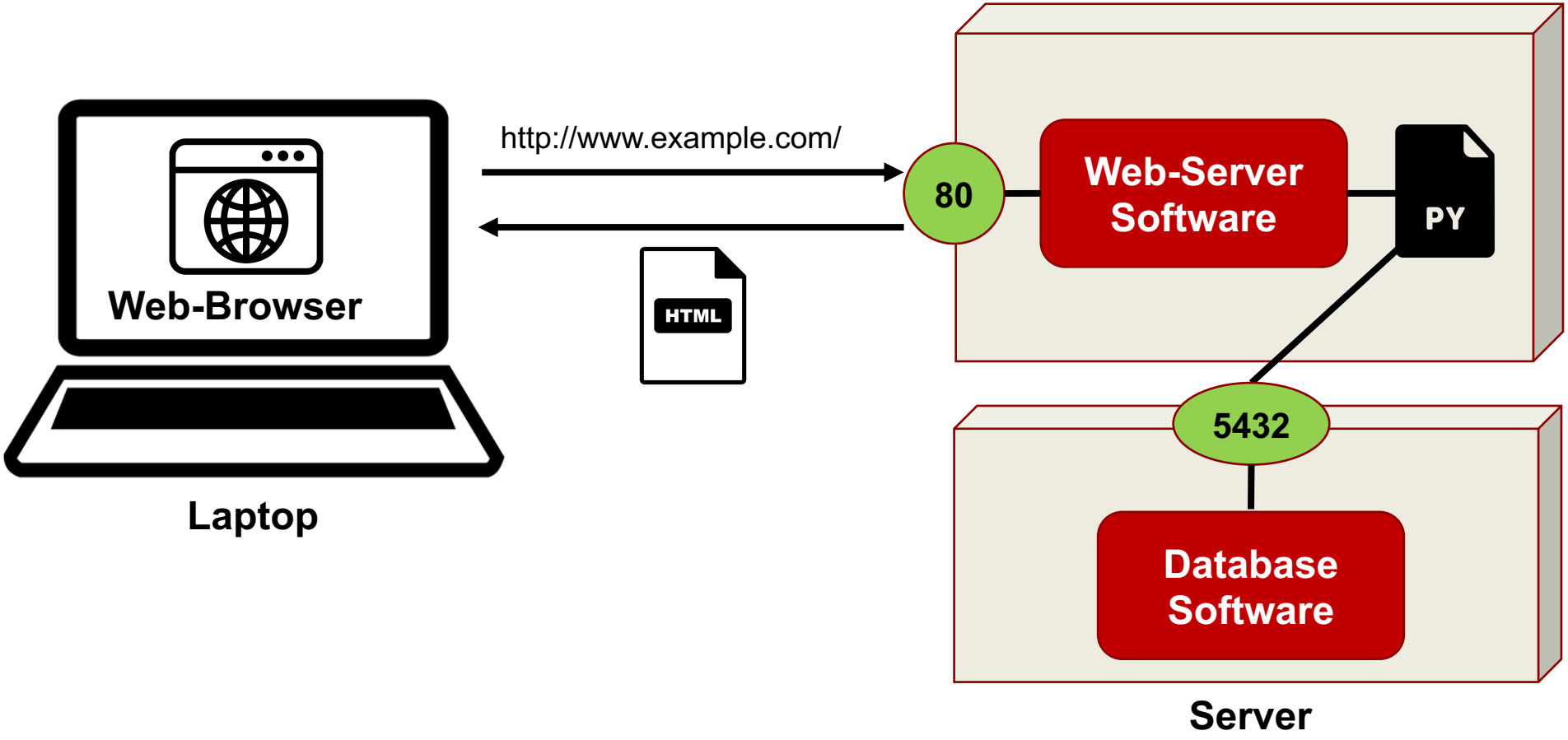
Server-side Dynamic Website



Server-side Dynamic Website



Server-side Dynamic Website



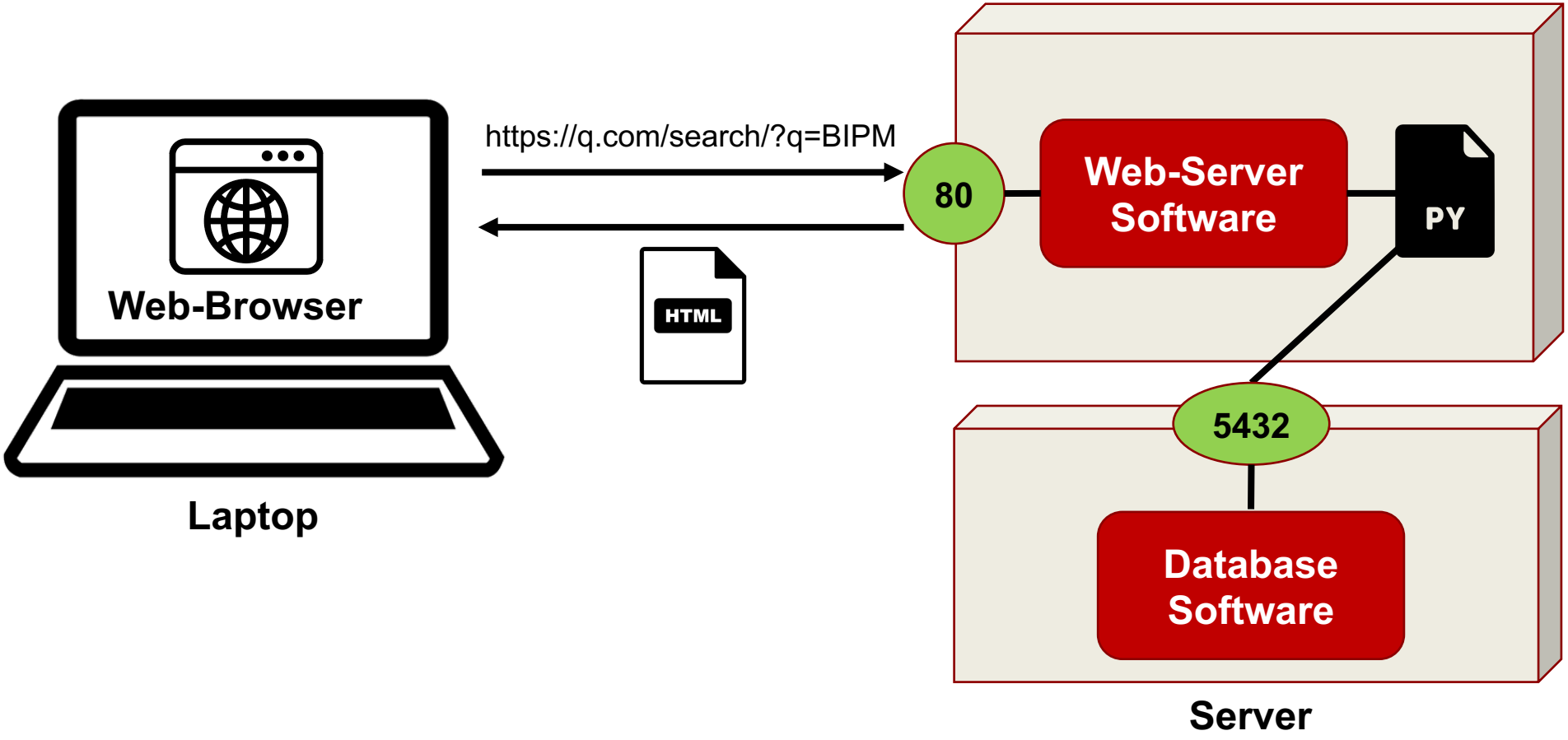
Popular Web Application Frameworks for different Programming Languages

- 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

Source: https://www.botreetechnologies.com/blog/top-15-websites-built-with-python/

Server-side Dynamic Website



Cache

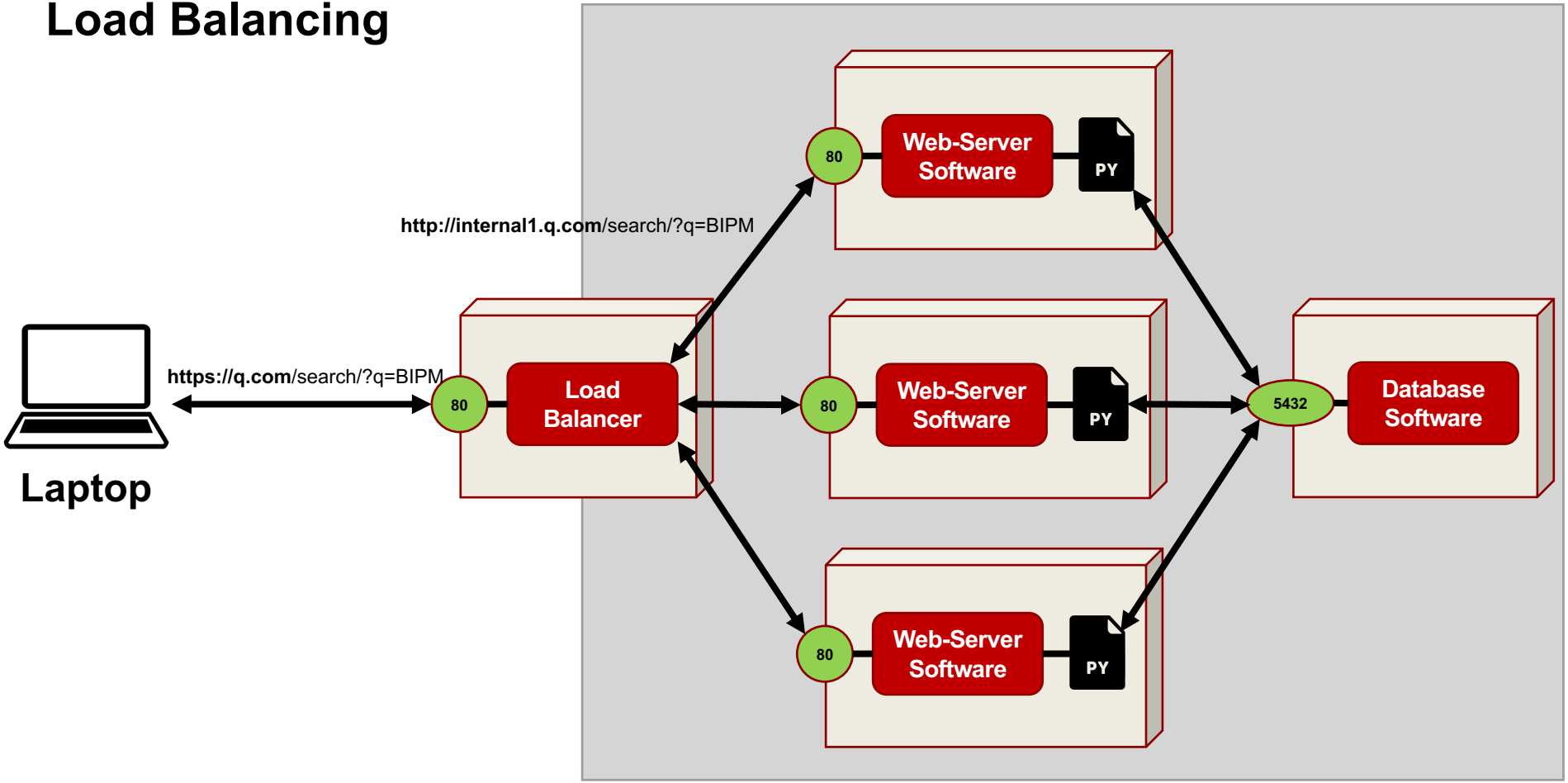
- Save computational expensive results
- Example: Front Page of New York Times
- “There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors.” — Leon Bambrick
- “There’s two hard problems in computer science: we only have one joke and it’s not funny.” — Phillip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

Proxy Server

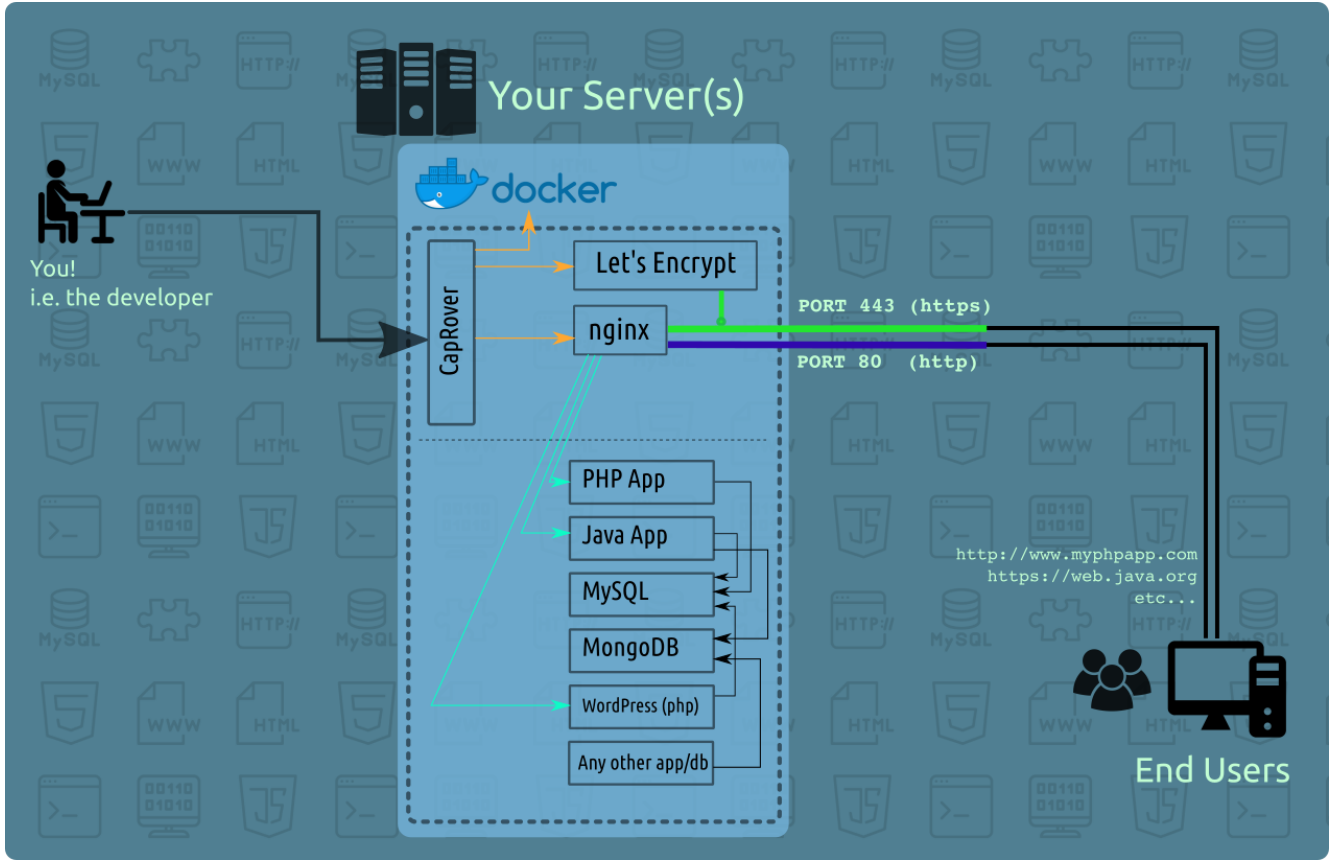
- Intermediary for requests
- **Reverse Proxy**
  - Forwards to a server and returns the server's response to the client
- **Load Balancer**
  - Distributes requests among a group of servers in each case returning the response from the selected server to the appropriate client
  - Types of Load Balancing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

Source: https://www.nginx.com/resources/glossary/reverse-proxy-vs-load-balancer/

Load Balancing



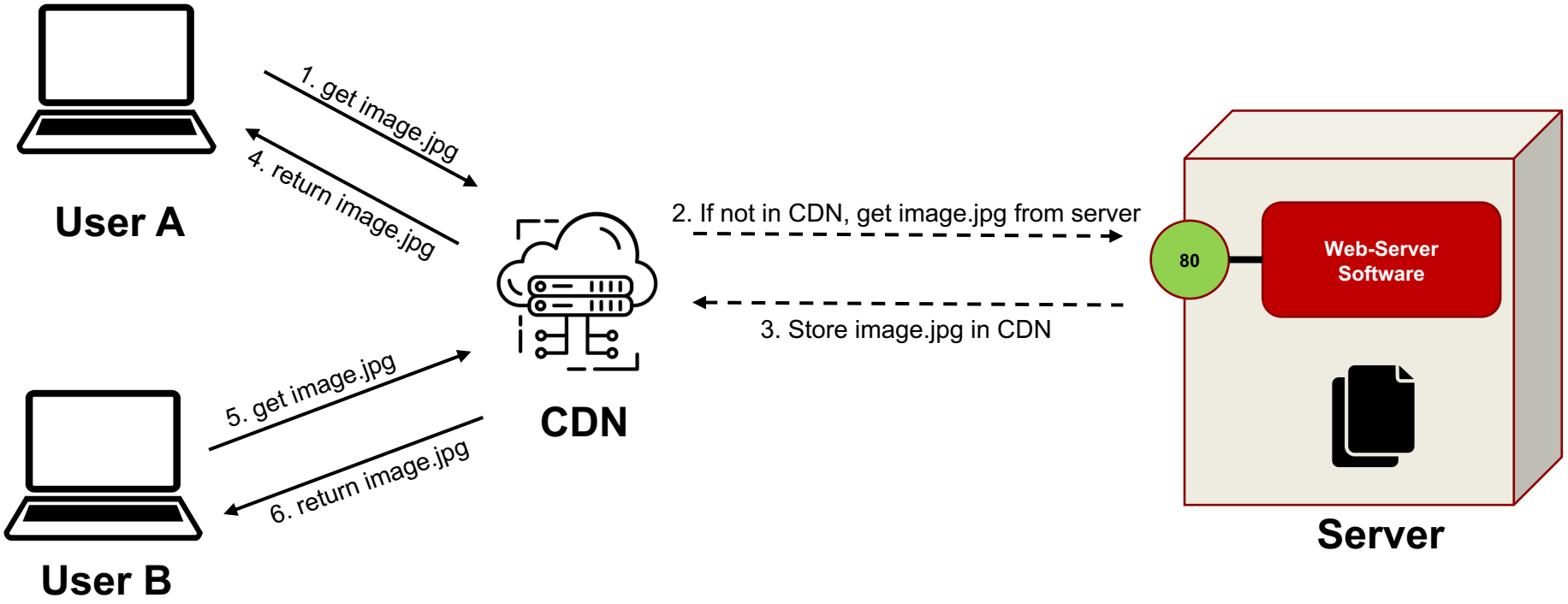
CapRover: Reverse Proxy for Docker Containers with easy HTTPS



Content Delivery Network (CDN)

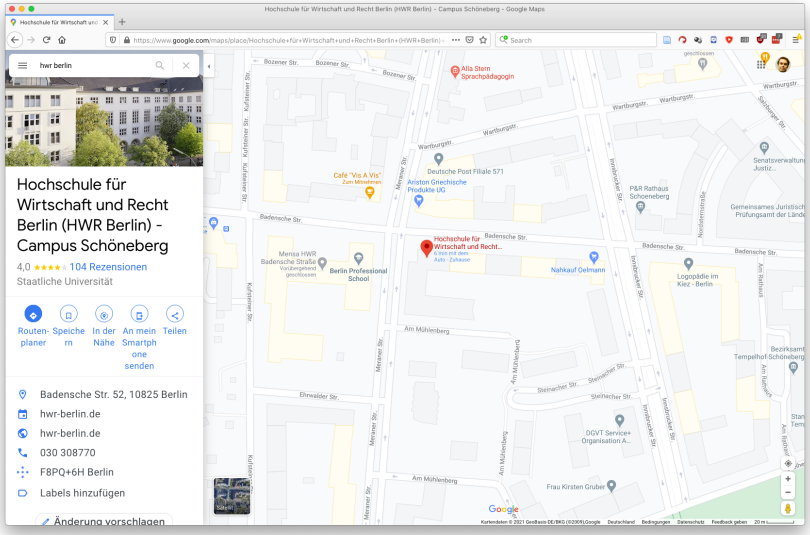
- Geographically distributed network of proxy servers and data centers for delivering static assets
- Goals
  - Reduction of load to original servers
  - Increasing availability
  - Geographically nearer located to end user
- Static Assets: any non-dynamic file like Images, Videos, CSS, JavaScript, Static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

Content Delivery Network (CDN)

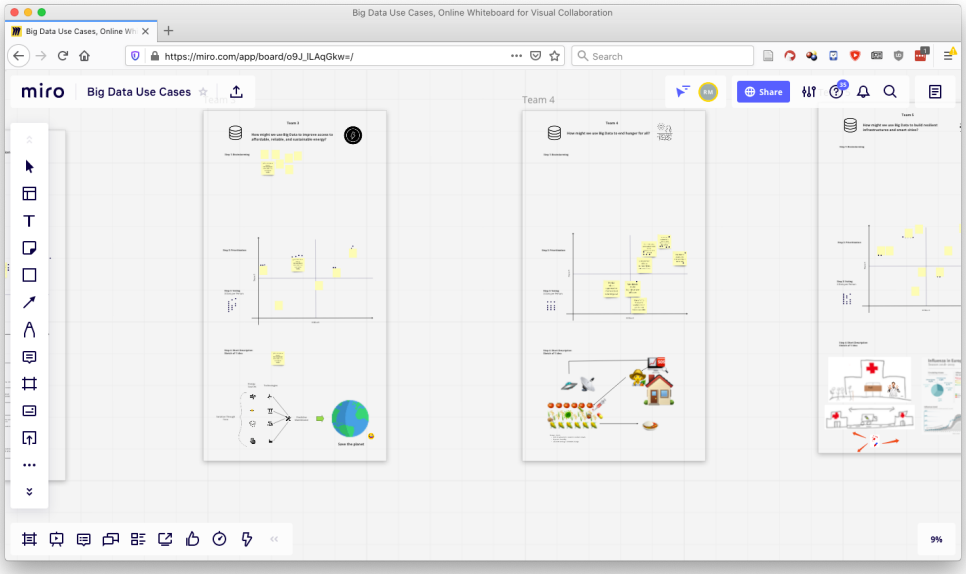


Examples of highly interactive websites

Google Maps



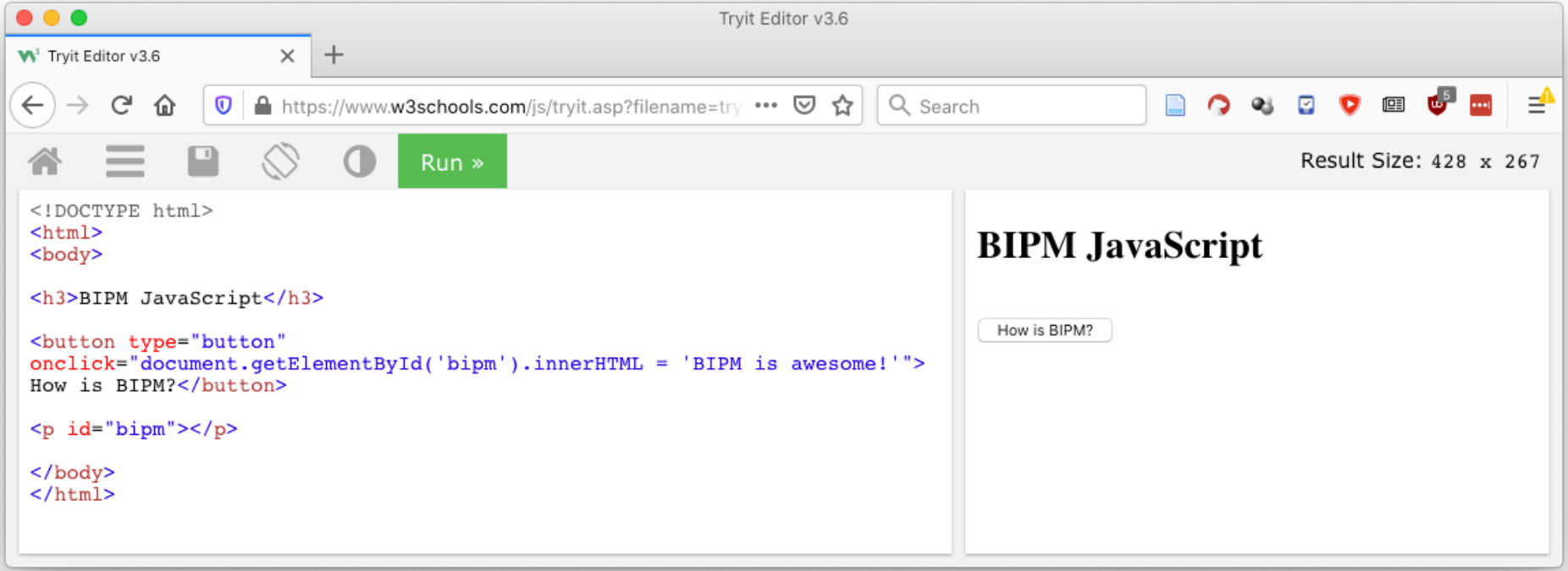
Miro



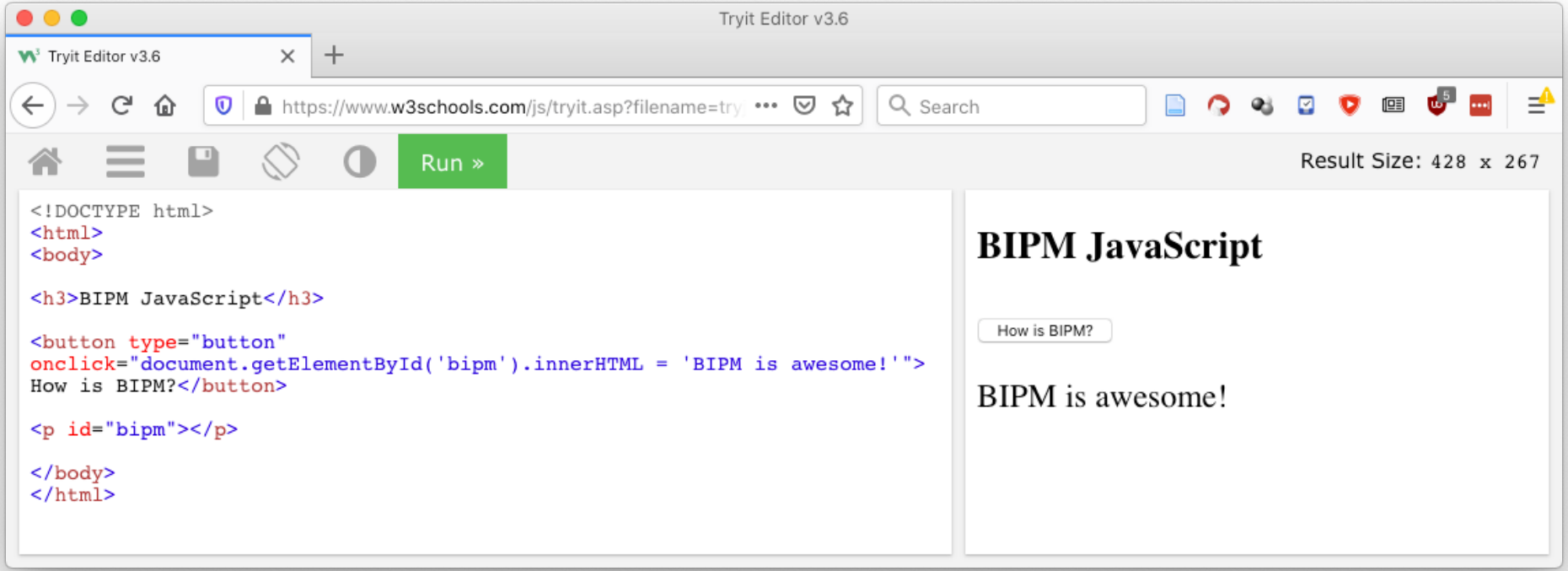
JavaScript

- 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

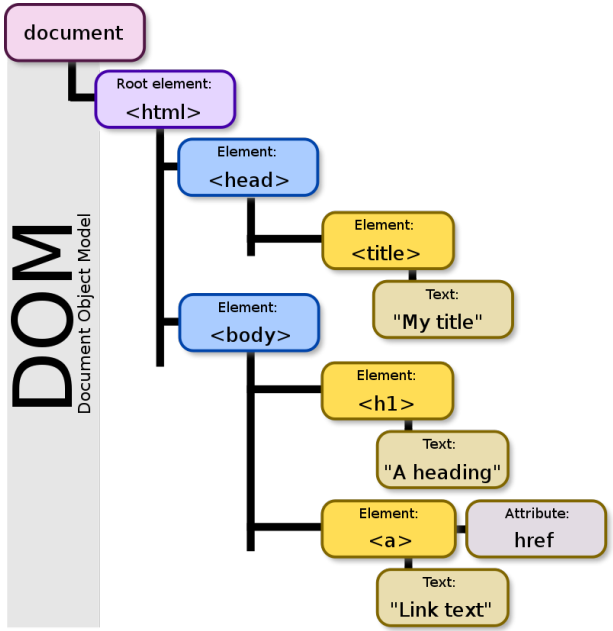


Client-side Dynamics with JavaScript



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

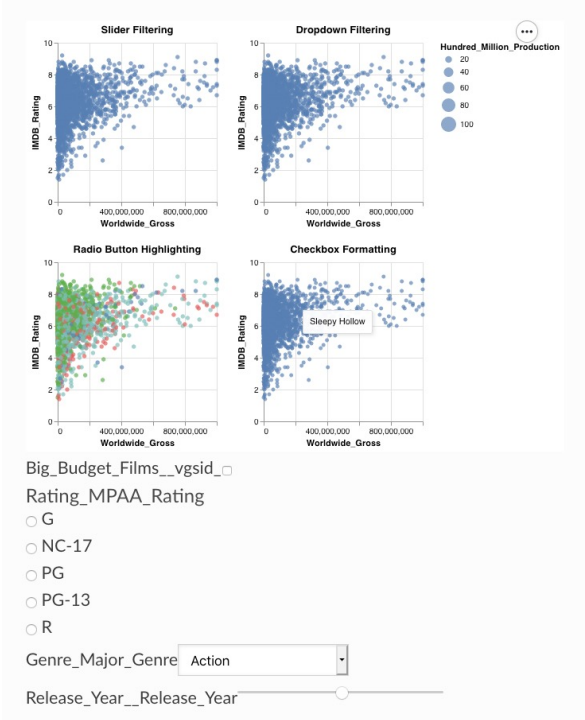


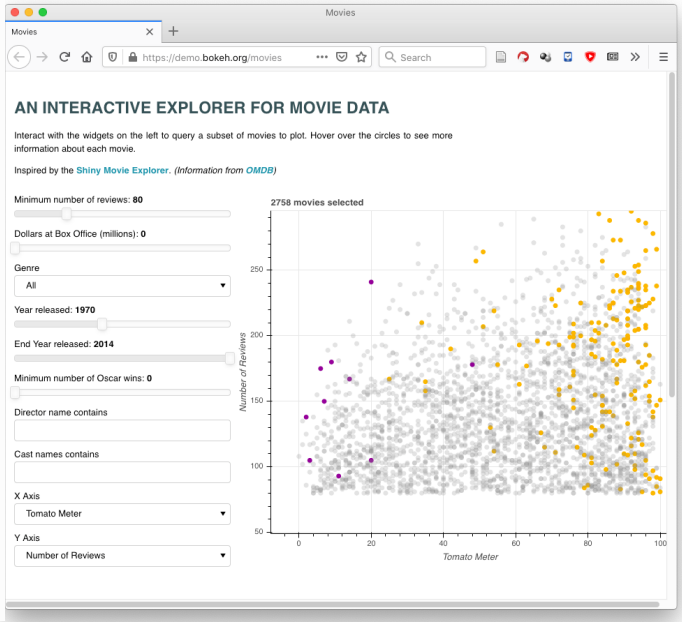
JavaScript Front-end Libraries

- 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)