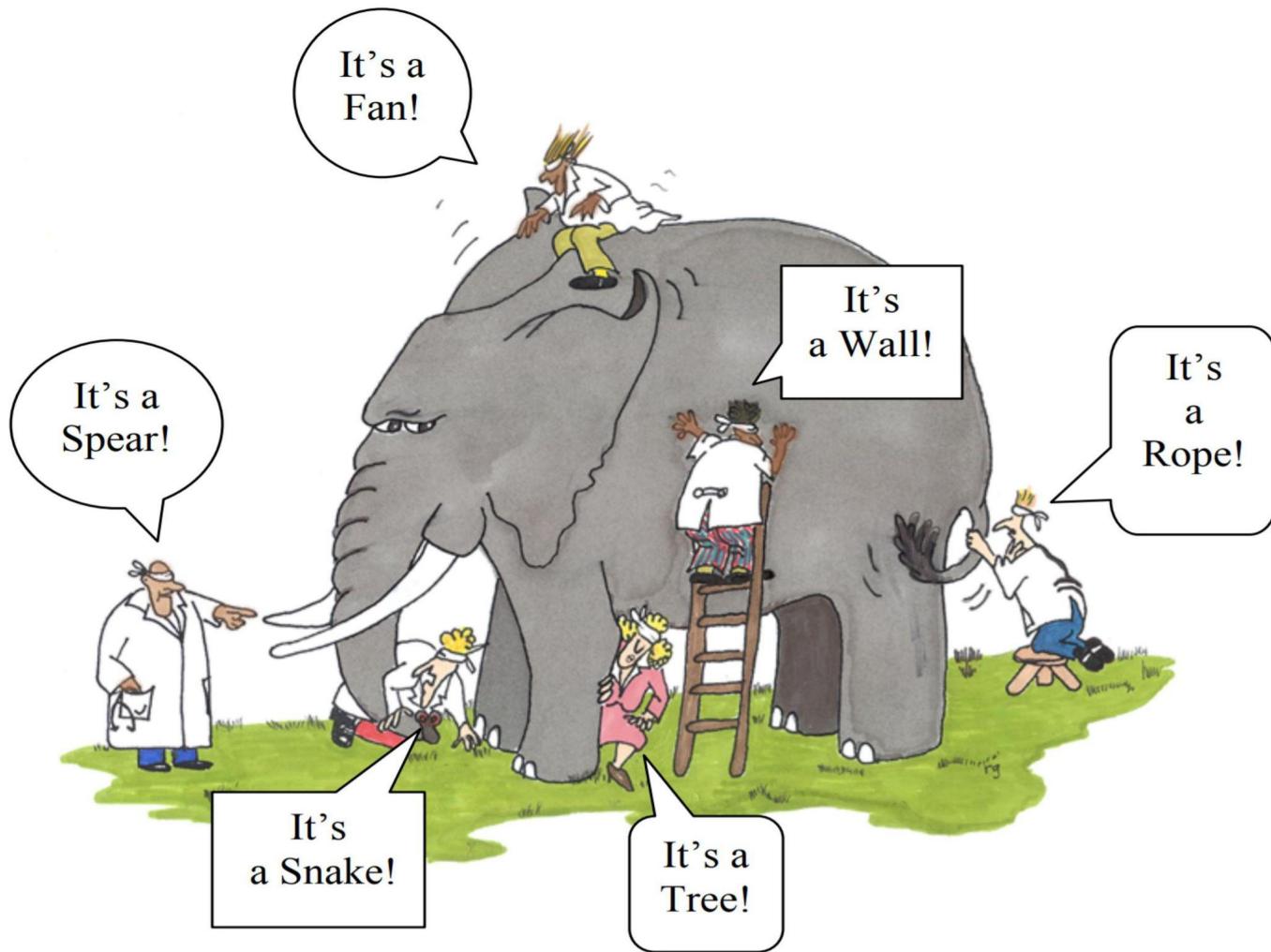


Software Engr & Web Engr

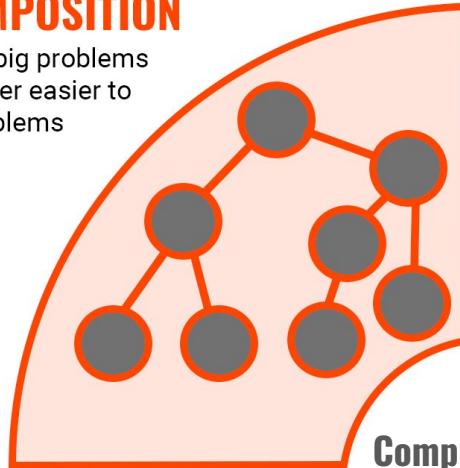
Abel Sanchez, John R Williams



What is computational thinking?

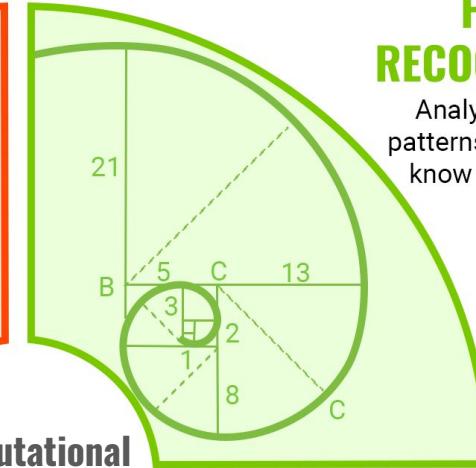
DECOMPOSITION

Breaking big problems
into smaller easier to
solve problems



PATTERN RECOGNITION

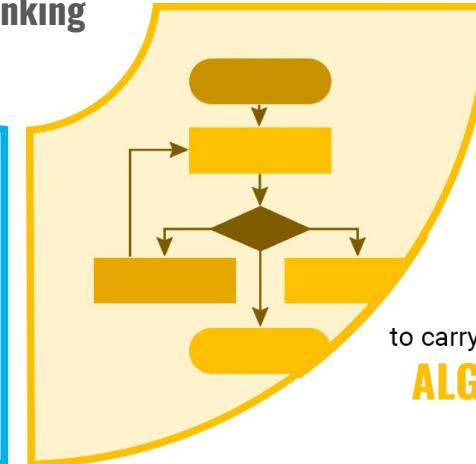
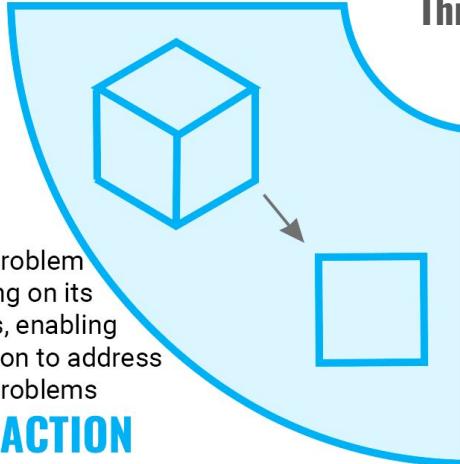
Analyze & look for
patterns you already
know how to solve



Computational Thinking

Simplify problem
by focusing on its
essentials, enabling
one solution to address
multiple problems

ABSTRACTION



Step-by-step
instructions
to carry out solution

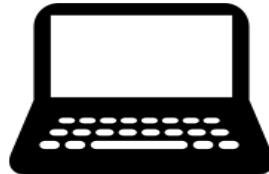
ALGORITHM DESIGN

Computational thinking

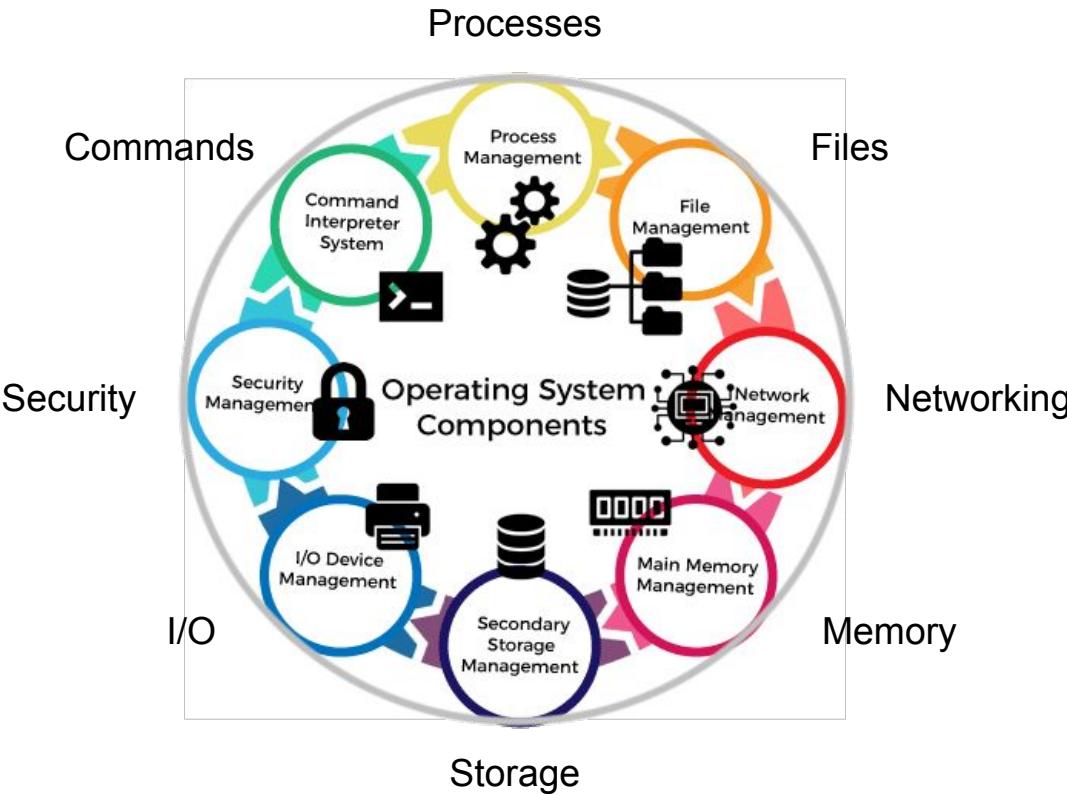
*Posing a problem/solution in terms
of computation.*

**What are the components
of a software application?**

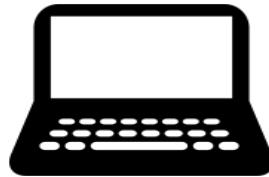
Computation Context



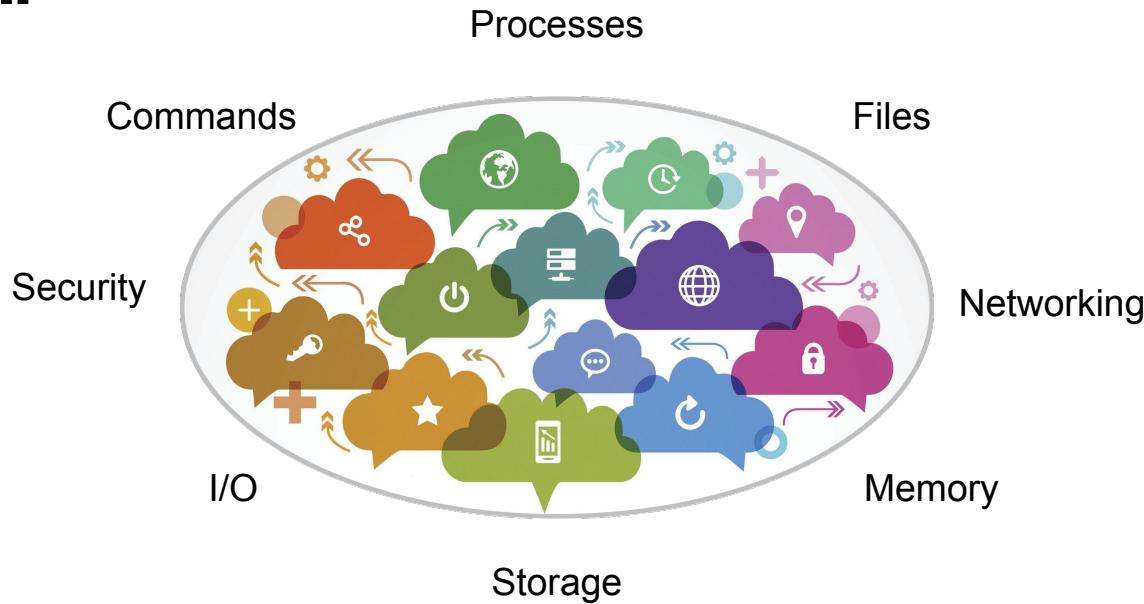
OS:
Windows
Mac OS
Linux



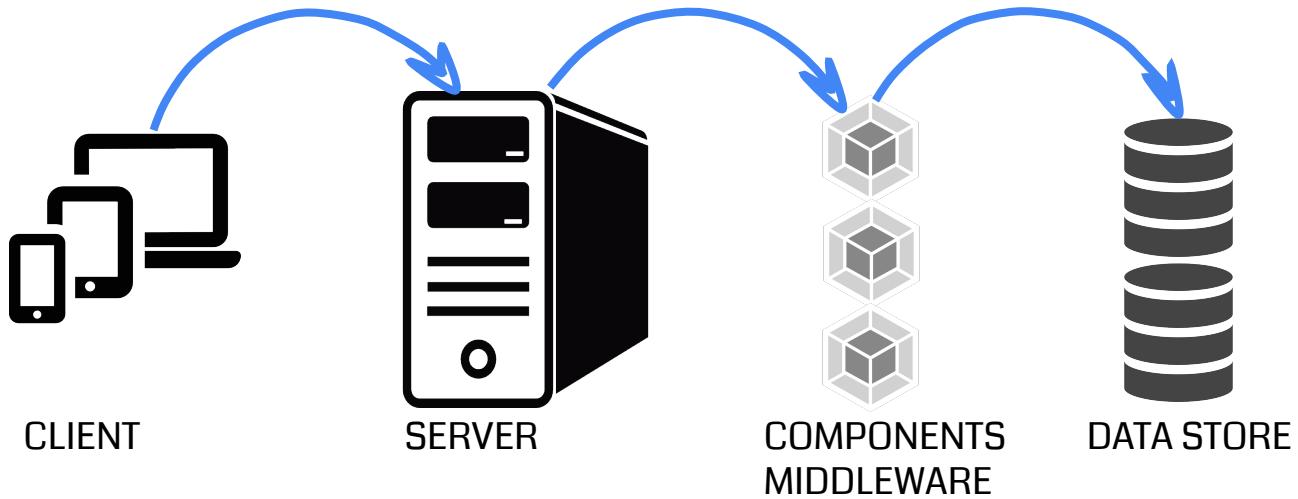
Computation Context



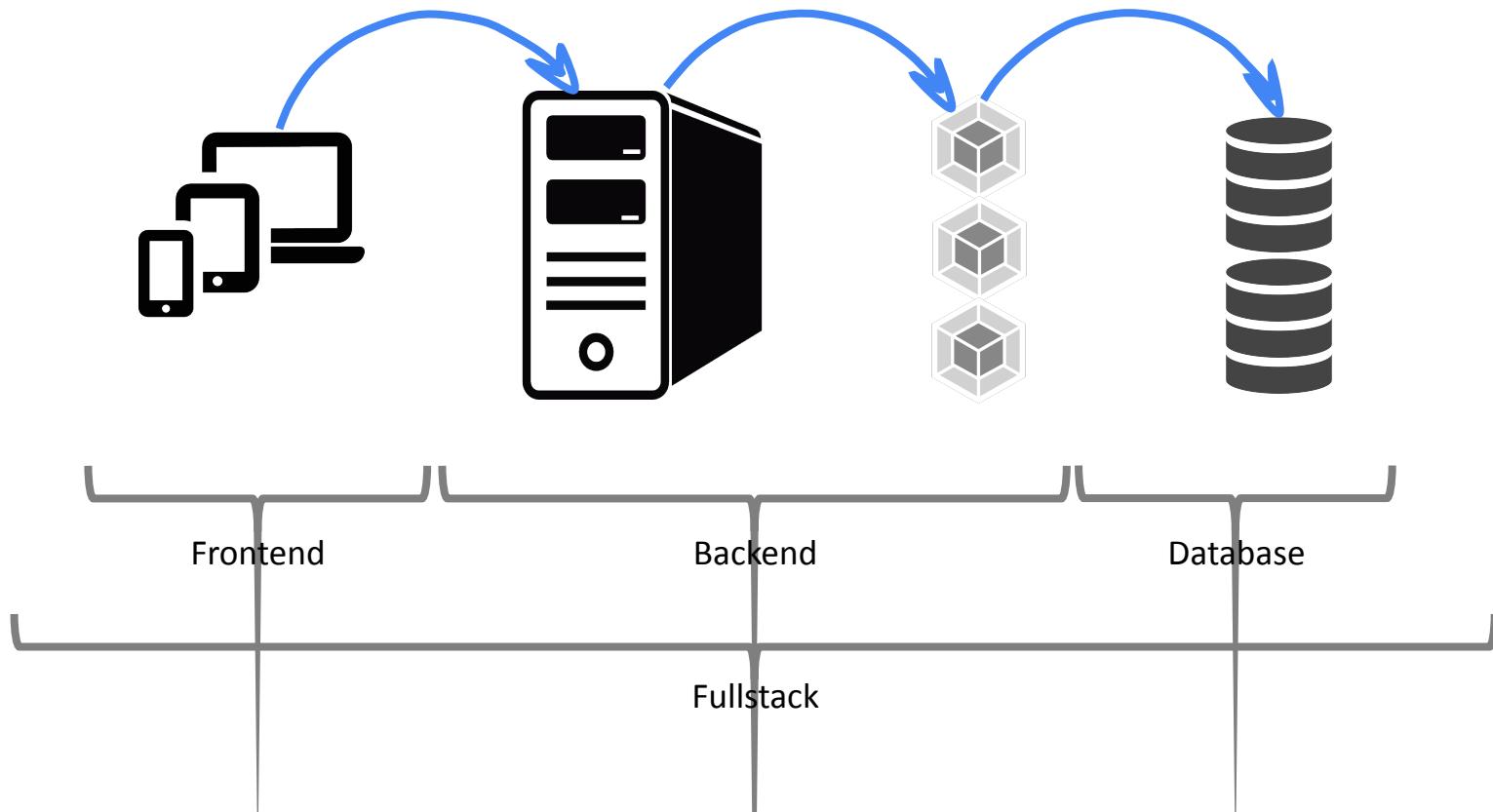
OS:
Windows
Mac OS
Linux

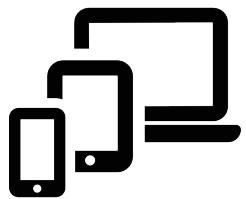


Components of a software application

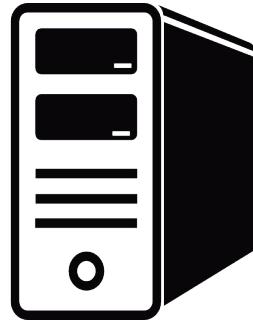


Front, back, storage - full stack





CLIENT



SERVER

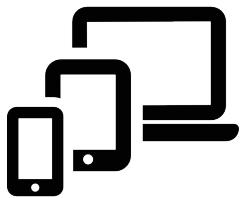
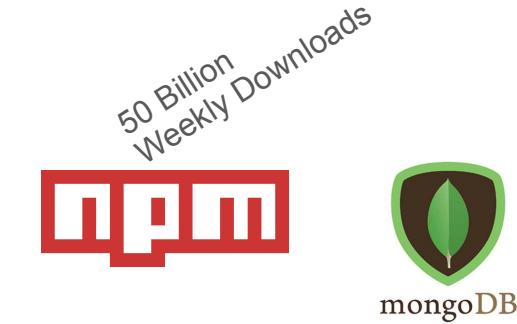


COMPONENTS

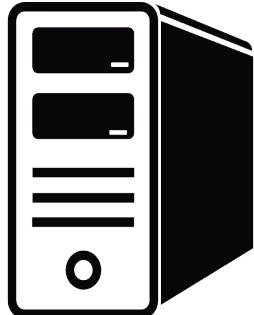


DATA STORE





CLIENT



SERVER



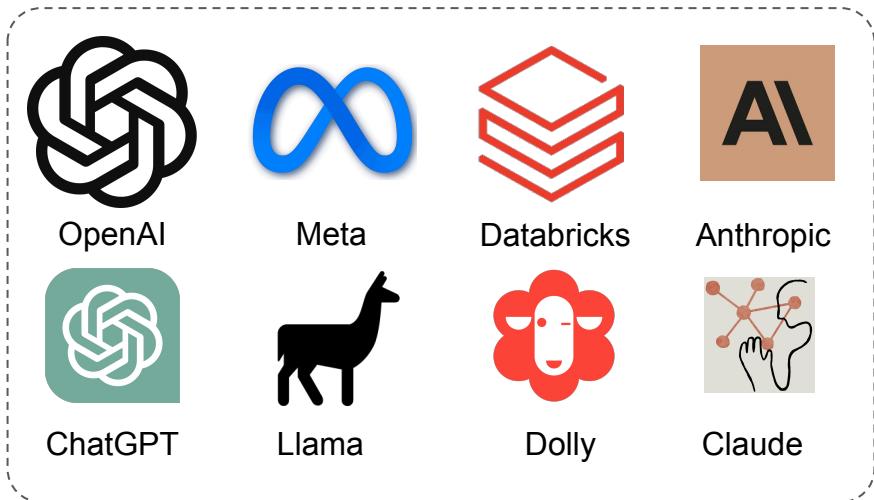
COMPONENTS



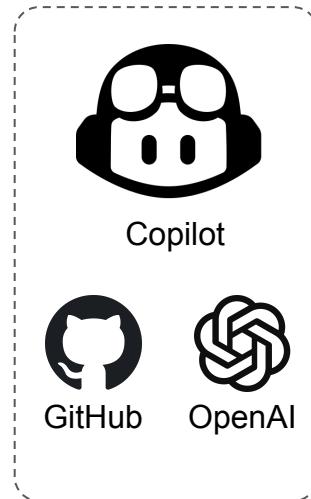
DATA STORE



Low code, no code



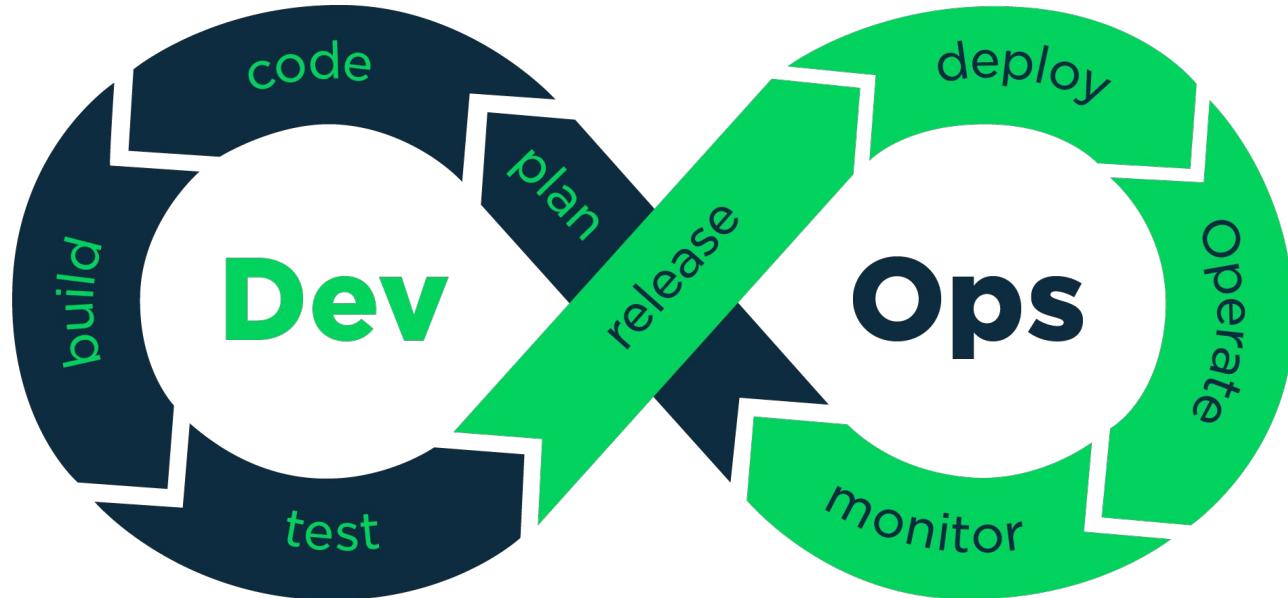
LLMs



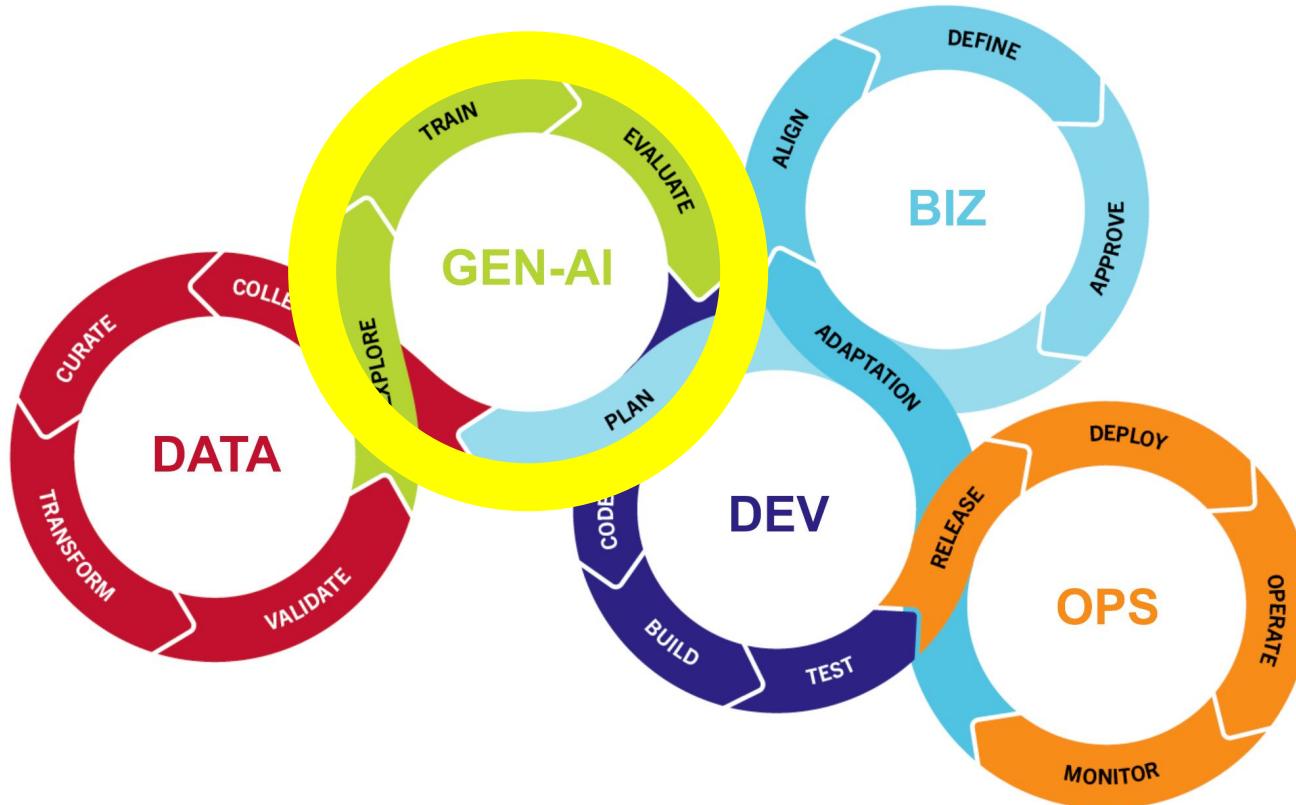
LLM App

Where does Generative-AI fit in?

Development cycle - what are we missing



A new building block in the dev cycle





**Each platform has
a stack of technology**



CLI



Browser

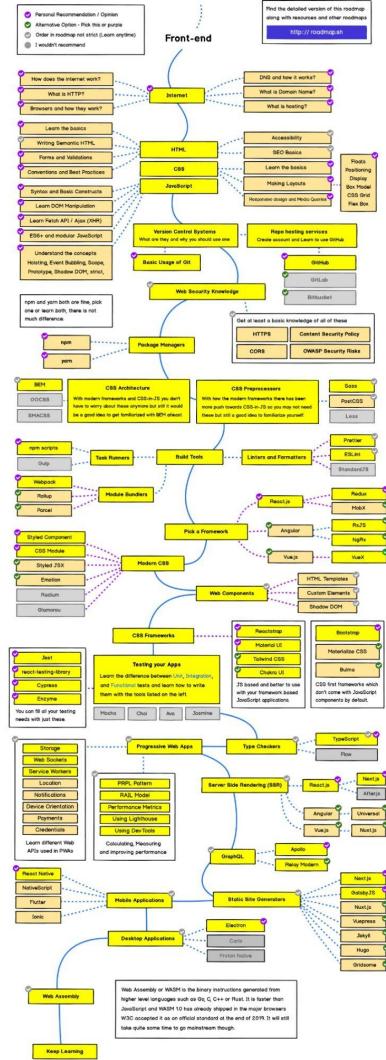


Native

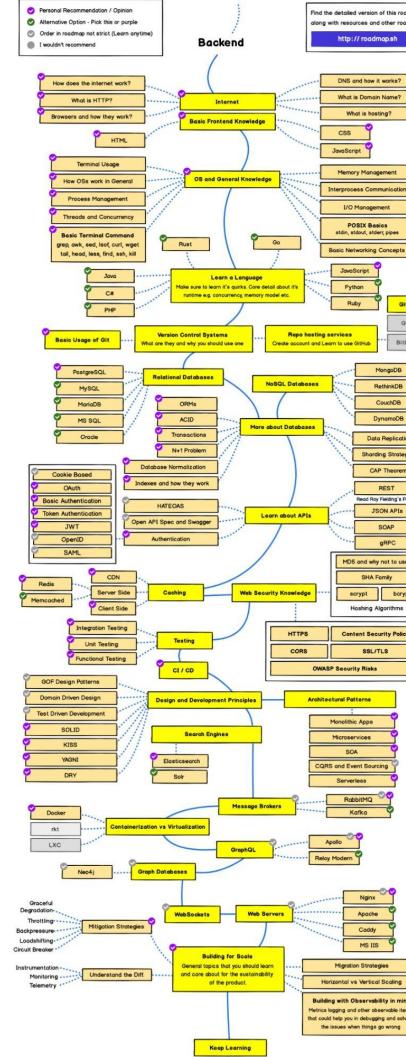


Mobile

Frontend Architecture

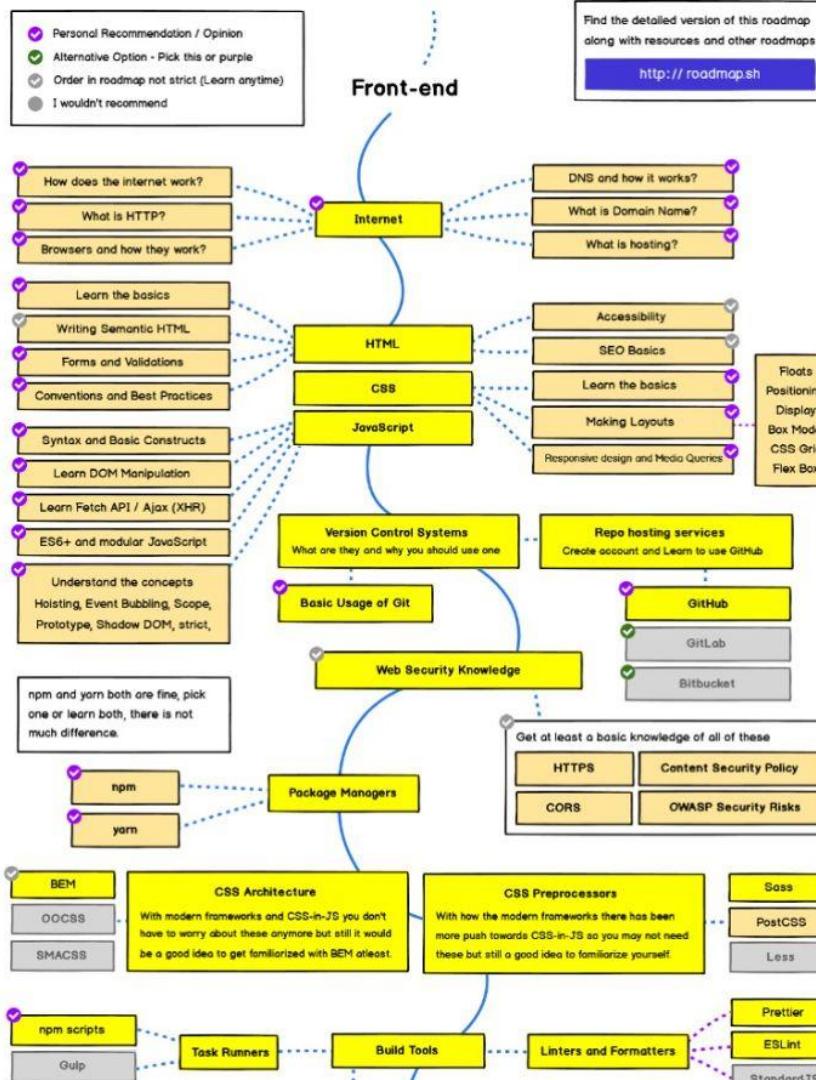


Backend Architecture



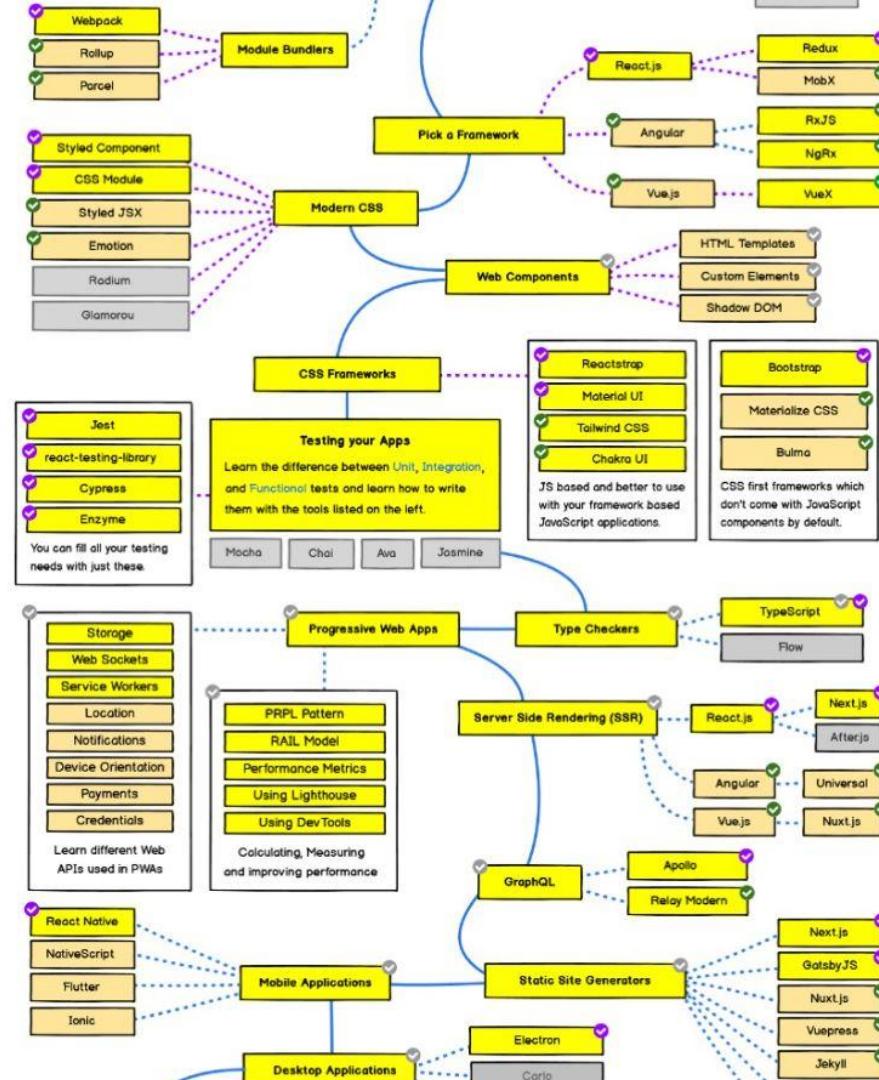
Frontend

Part I



Frontend

Part II

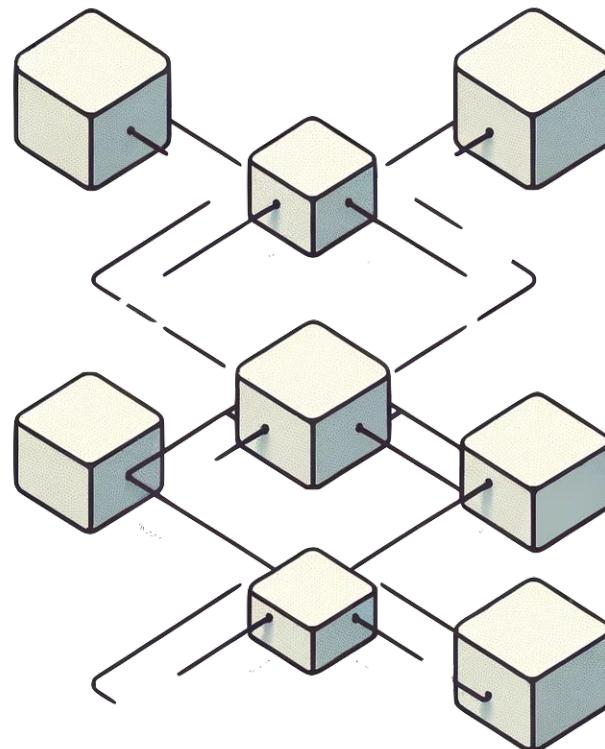


Even understanding the roadmap is hard

40+ Developer Roadmaps

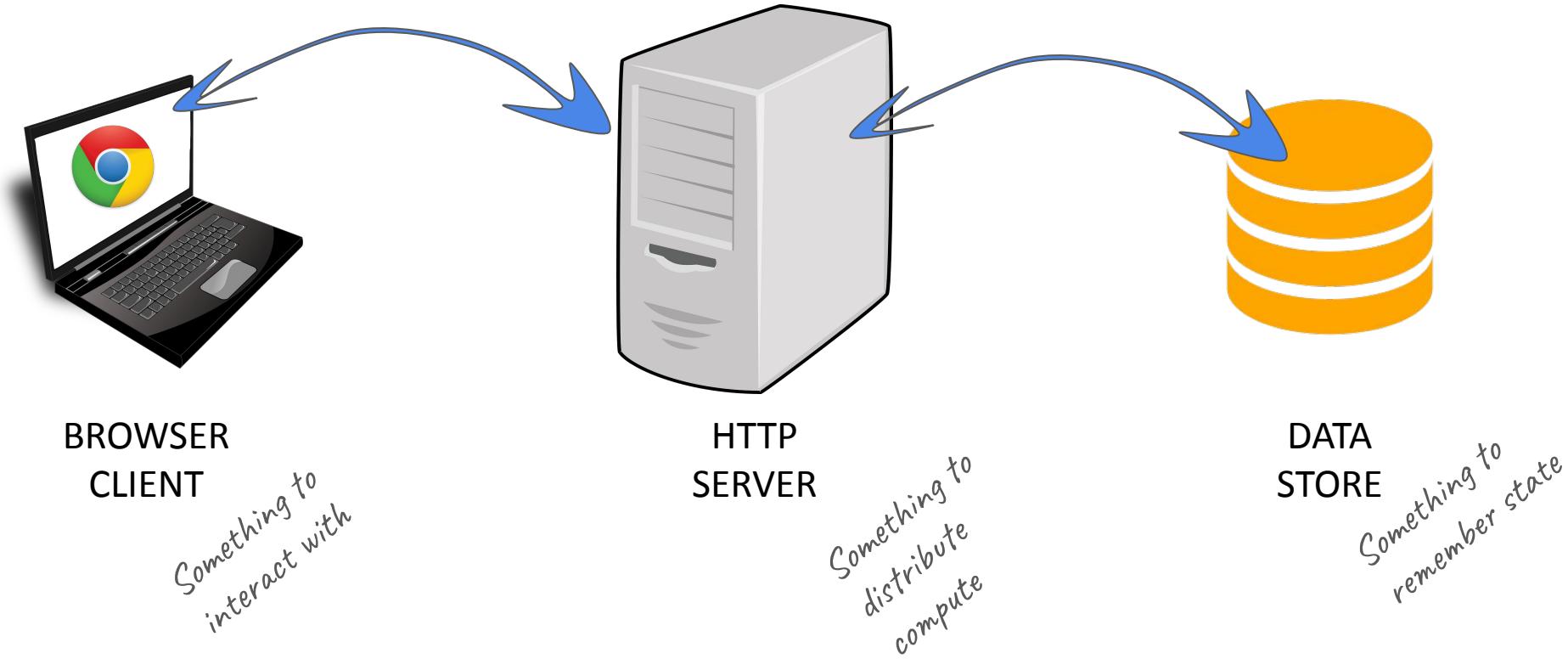
<https://roadmap.sh/>

Sample: Blockchain Roadmap

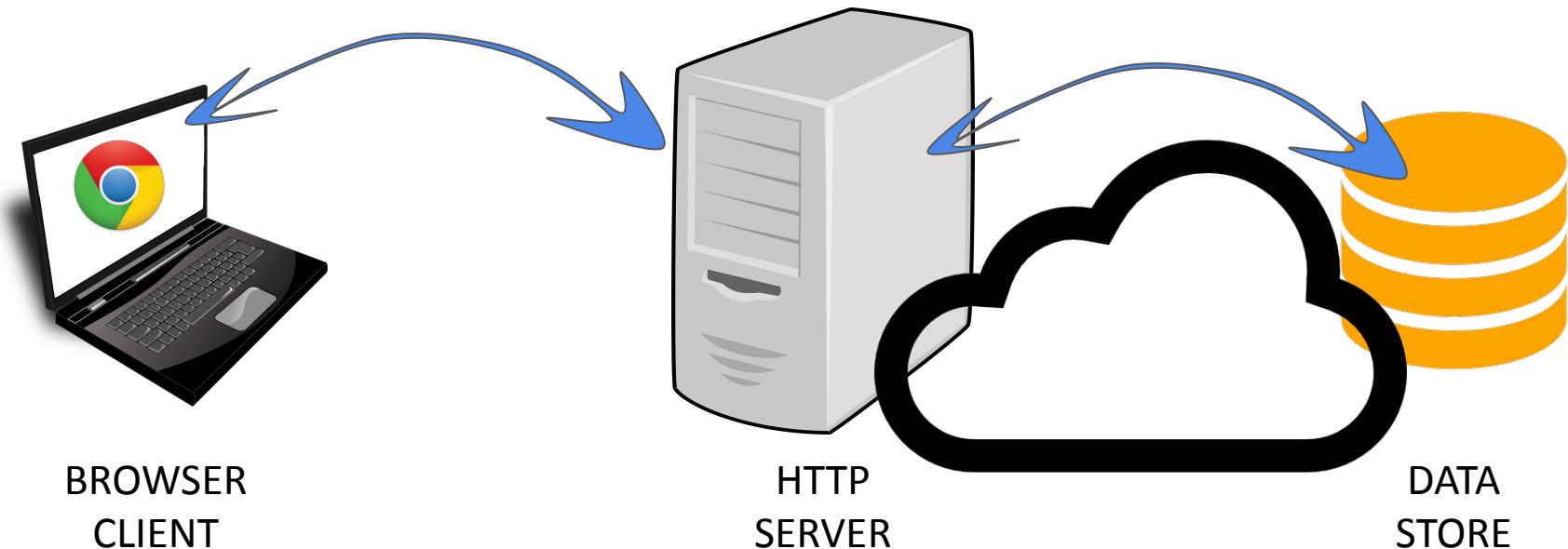


Architecture Evolution

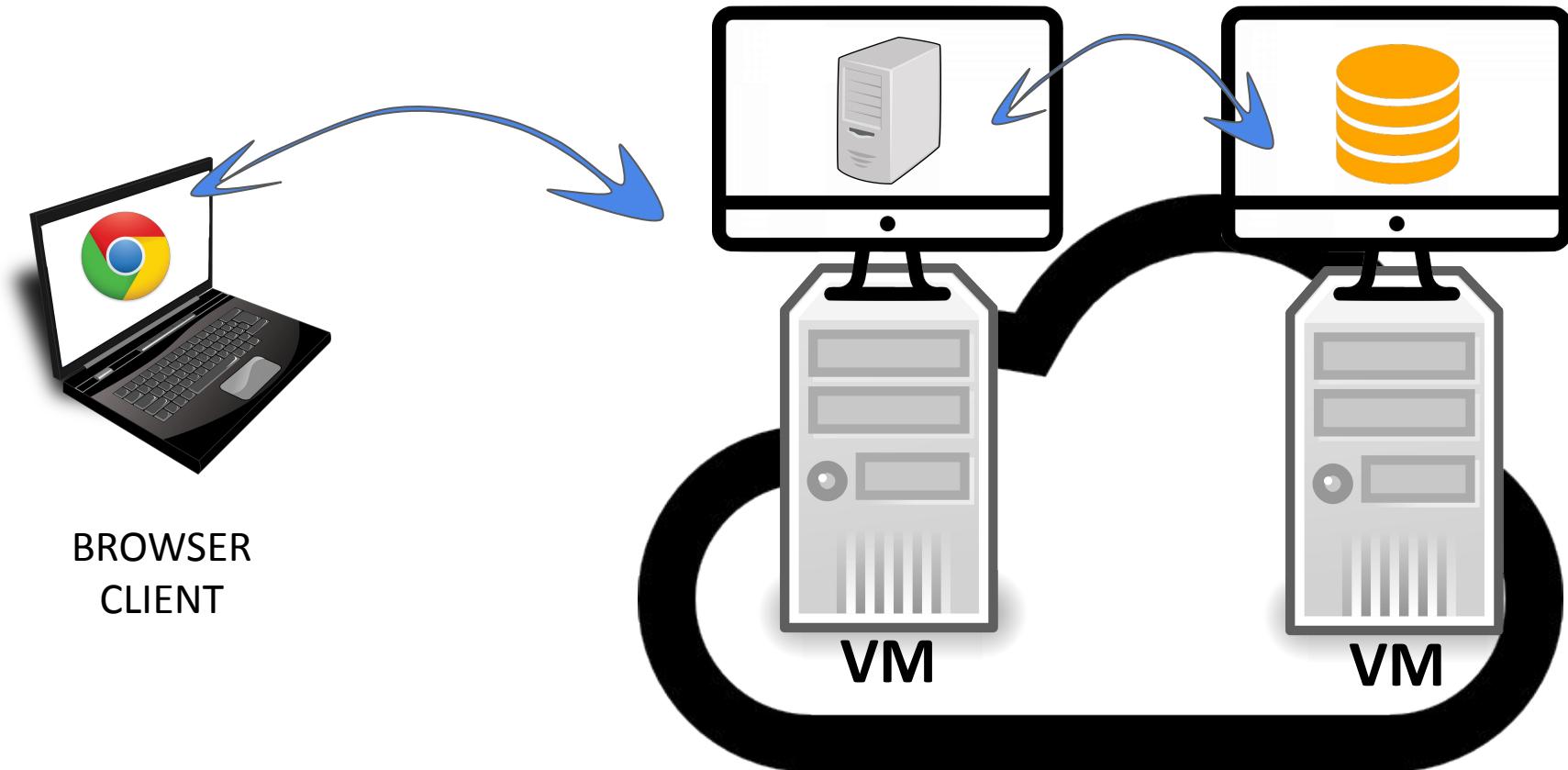
Three Tiers: Client, Server, & Storage



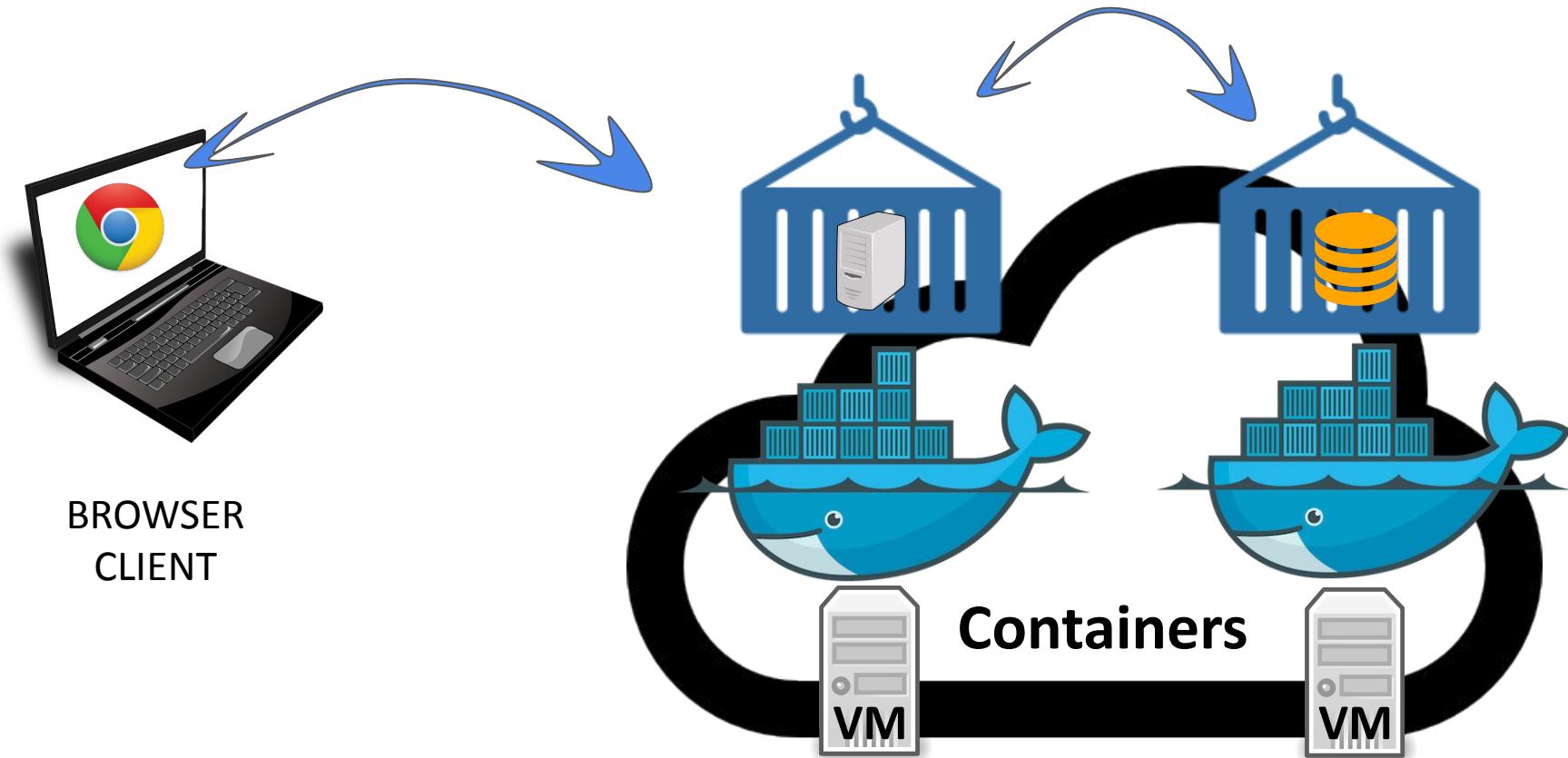
Cloud



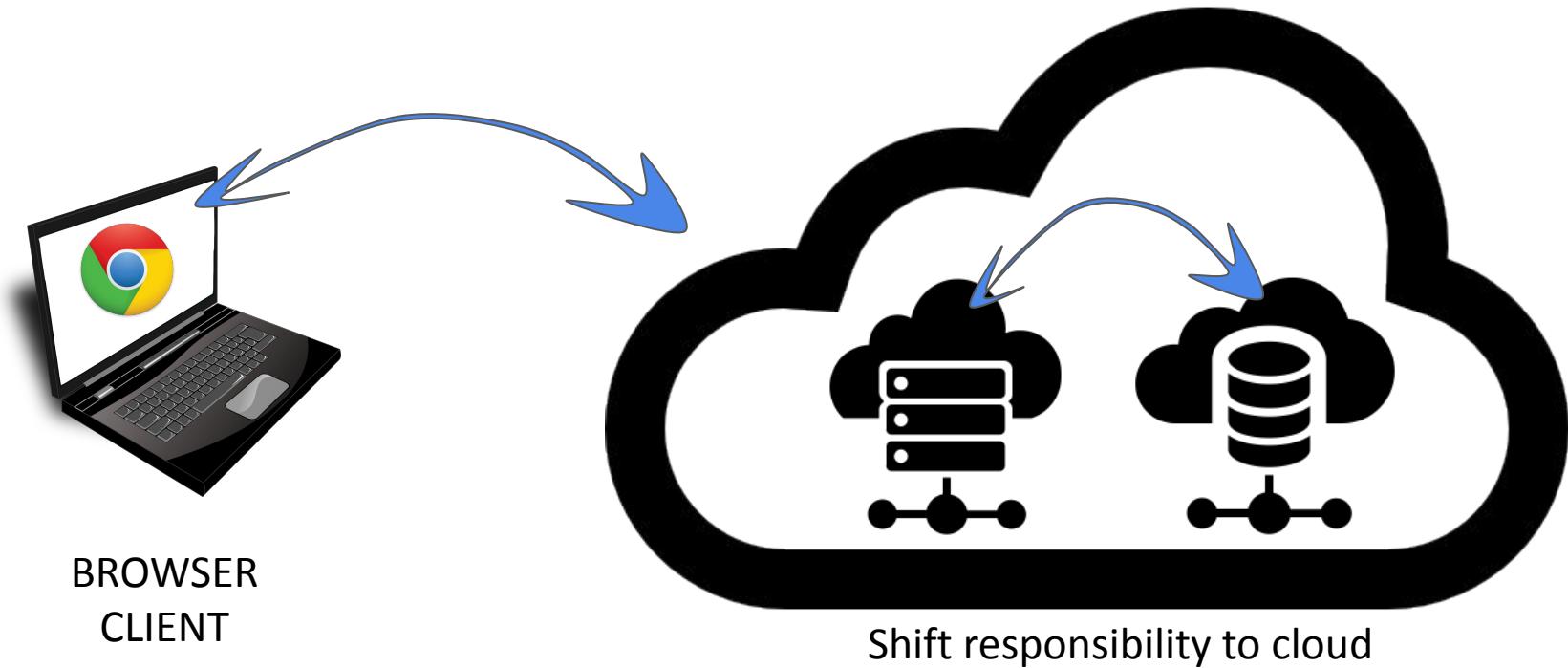
Cloud – VM, Lift & Shift



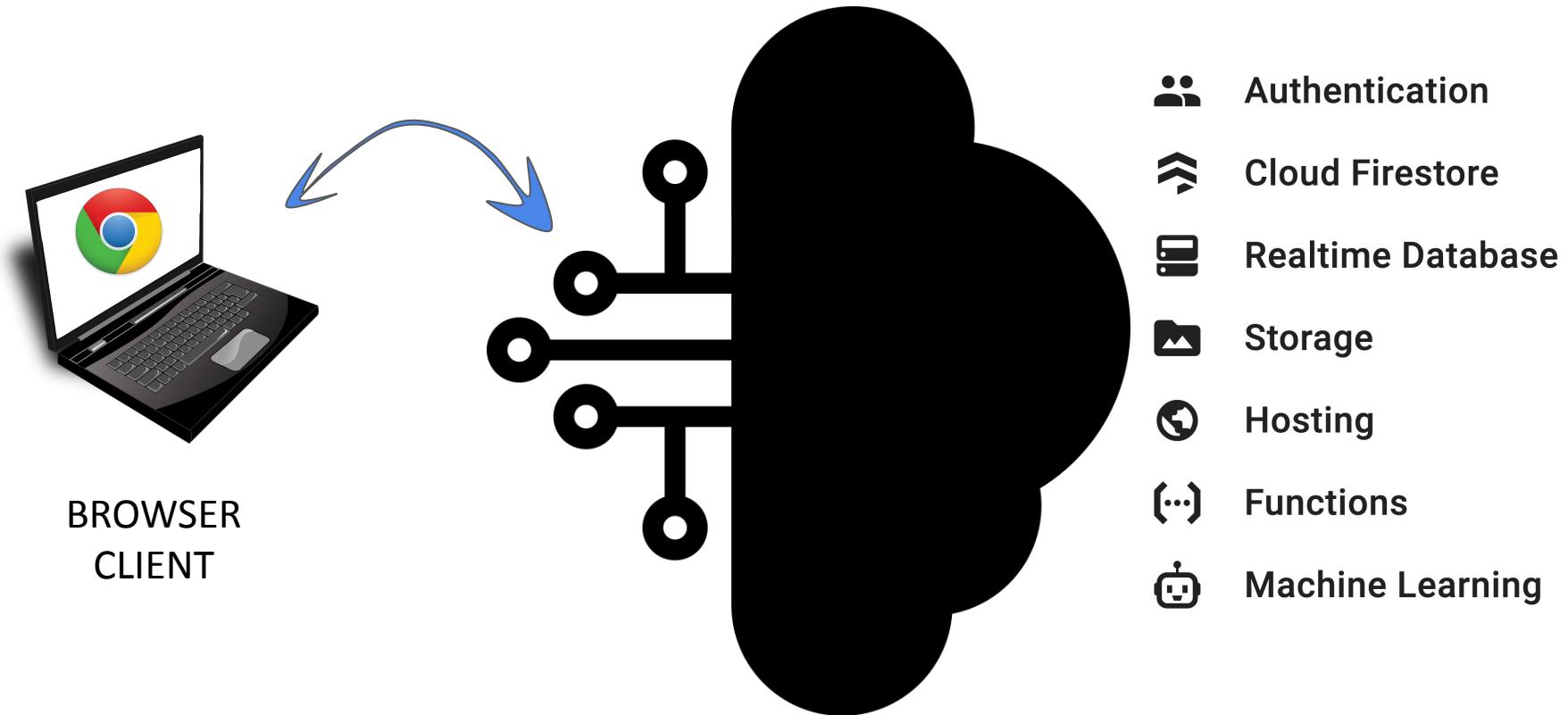
Cloud – Dockerise, Lift & Shift



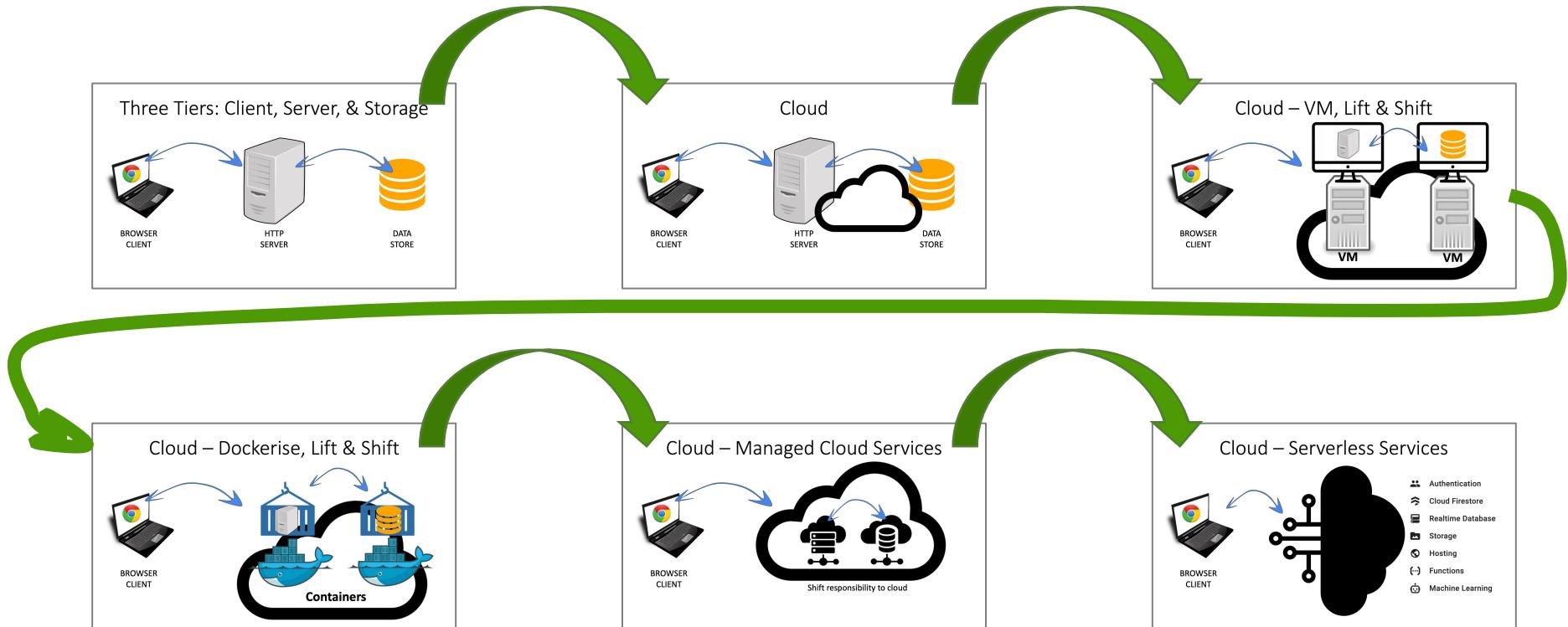
Cloud – Managed Cloud Services

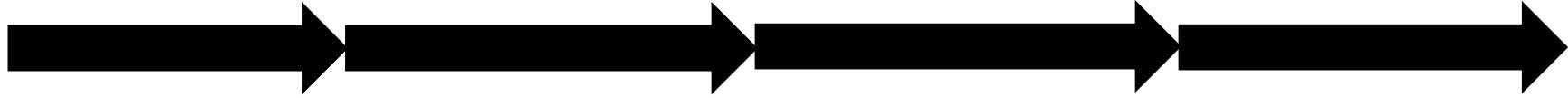


Cloud – Serverless Services



Architecture Evolution





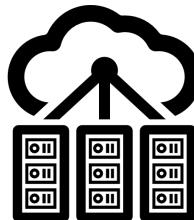
Datacenter



- Deploy in months
- Live for years



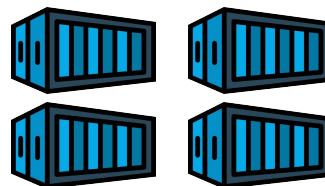
Virtual Machine



- Deploy in minutes
- Live for weeks



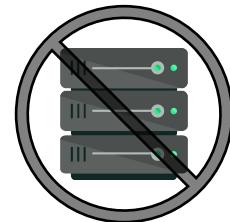
Containers



- Deploy in seconds
- Live for minutes/hours



Serverless



- Respond in milliseconds
- Live for seconds



AppSheet

The intelligent no-code platform

Create apps to transform your workplace.

[Start for free](#)

No-Code

Anyone on your team can make apps.



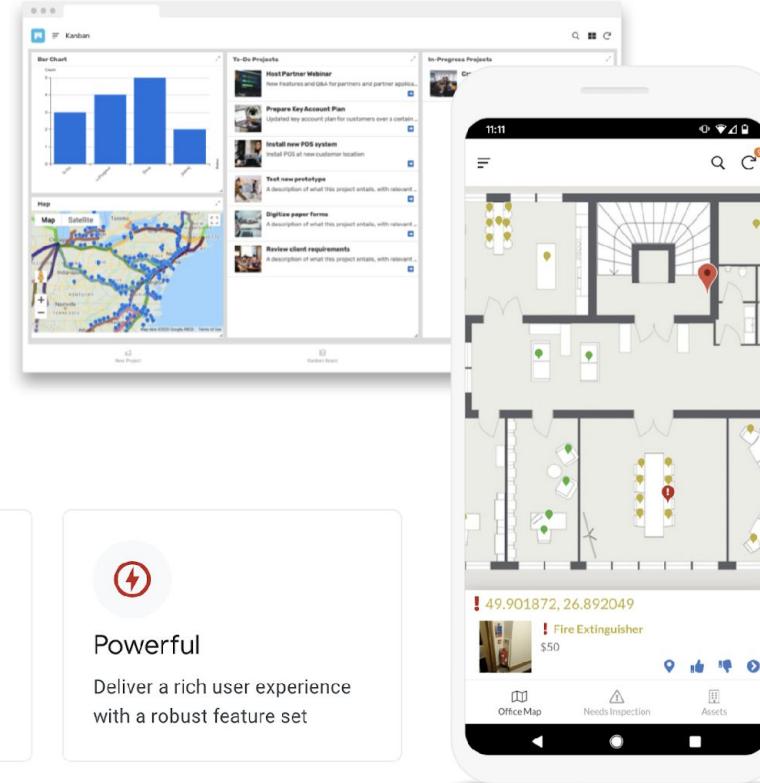
Agile

Create and deploy multi-platform apps in real-time



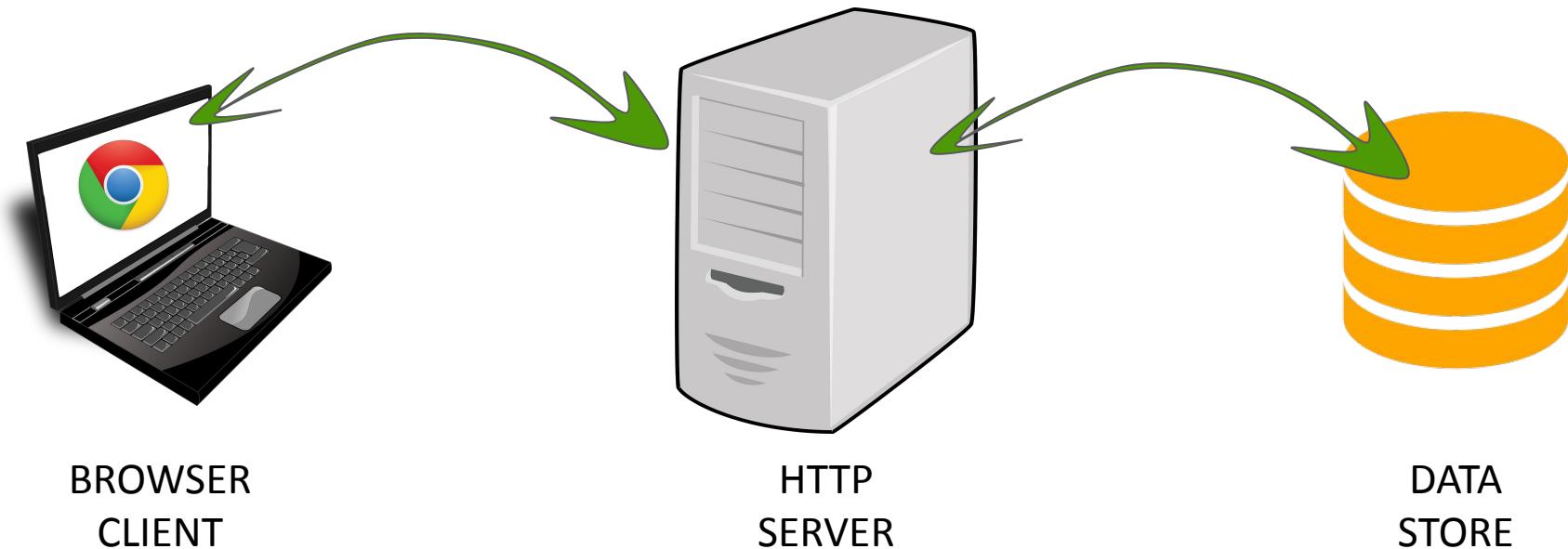
Powerful

Deliver a rich user experience with a robust feature set

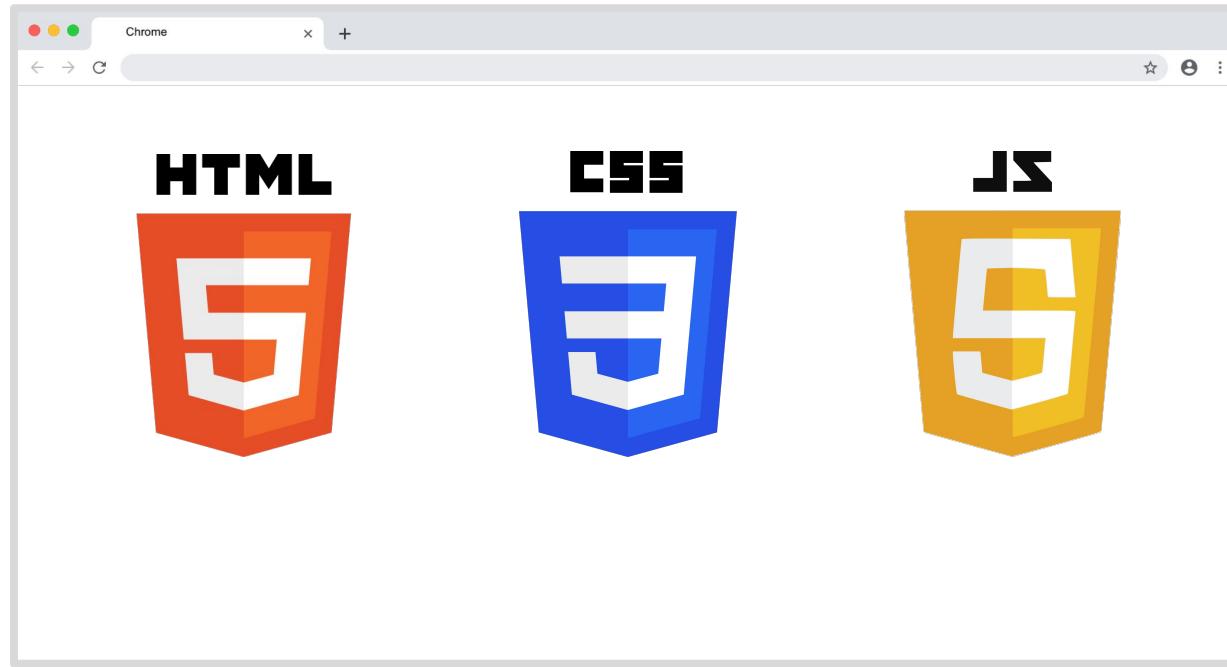


Frontend

Conceptually: let's start with a three tier architecture



Frontend (browser)



{JSON}



mapbox



Google
Charts



Active Learning: navigate to URL

<https://bit.ly/3xtNxr0>

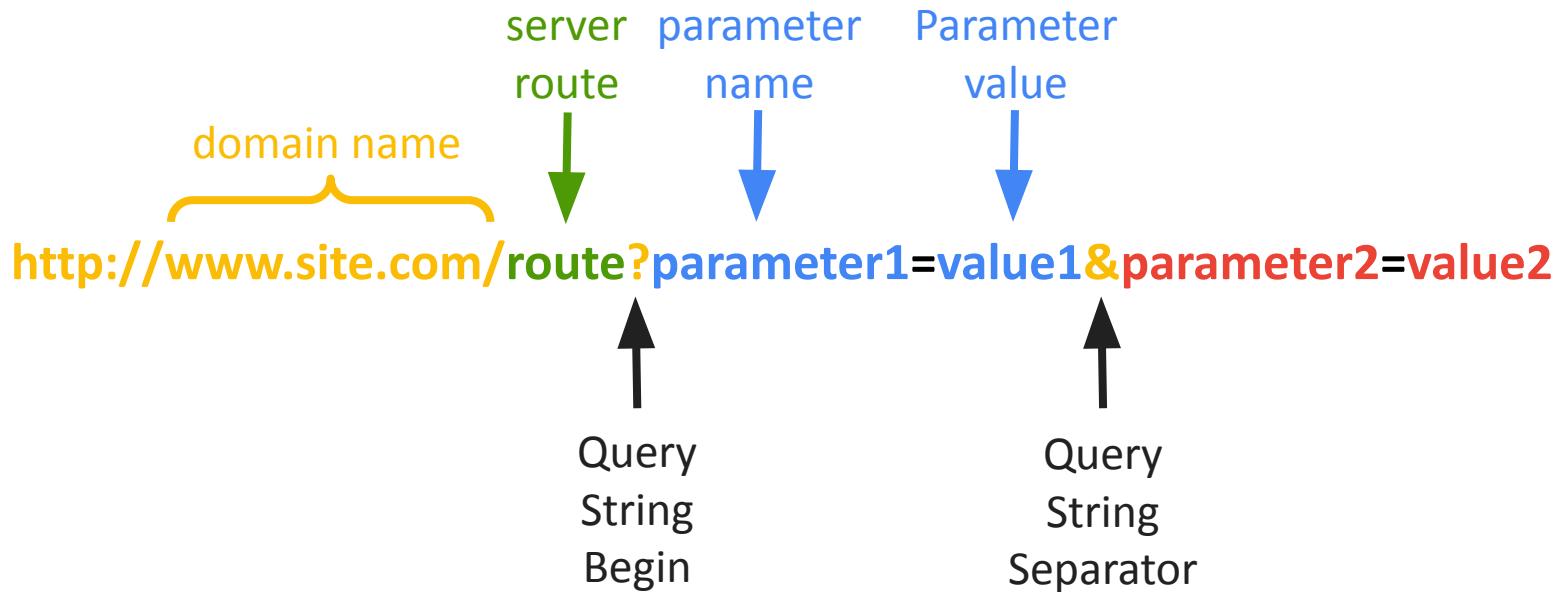
Answer the following:

- What did you find?
- What type of content is it?
- How would you use it?
- What application could you build?

**How would you structure a URL to perform
the same action as a function call?**

`http://www.domain.com/route`

Think of the URL Query String as a function call



Sample transportation data - Boston MBTA

[https://api-v3.mbta.com/vehicles?filter\[route\]=1&include=trip](https://api-v3.mbta.com/vehicles?filter[route]=1&include=trip)

URL components:

https://api-v3.mbta.com/



Domain

vehicles?



Transport type

filter[route]=1&



Route

include=trip



transit

police

energy

health

Frontend: HTML Basics

What I cannot create, I do not understand

Richard Feynman

Install Copilot - free for students



GitHub Copilot v1.229.0

GitHub github.com

20,124,618

★★★★★ (1071)

Your AI pair programmer

[Disable](#) | [▼](#)

[Uninstall](#) | [▼](#)

[Switch to Pre-Release Version](#)



Auto Update



<https://github.com/features/copilot>

Install Liveshare



Live Share v1.0.5940

Microsoft microsoft.com | ⚡ 18,214,302 | ★★★★☆ (154)

Real-time collaborative development from the comfort of your favorite tools.

[Disable](#) | [Uninstall](#)

Auto Update

Tags

- Plain text
- Heading
- Paragraph
- Image
- Link
- Form

Forms

Add User

localhost/

Add User

First Name

Last Name

SUBMIT

Form components

```
<form method="POST" action="/addUser">  
    firstname <input type="text" name="first" /><br>  
    lastname   <input type="text" name="last"  /><br>  
    email      <input type="text" name="email" /><br>  
    <input type="submit">  
</form>
```

Backend: Server basics

Flask server “hello world”

(flask documentation, <https://flask.palletsprojects.com/en/3.0.x/quickstart>)

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

# start server
if __name__ == '__main__':
    app.run(debug=True, port=3000)
```

Routing

```
@app.route('/')
def index():
    return 'Index Page'

@app.route('/hello')
def hello():
    return 'Hello, World'
```

Access Posted Data

```
# inspect posted data
@app.route('/viewer', methods=['GET', 'POST'])
def viewer():
    inspect = {}
    inspect['args'] = request.args
    inspect['form'] = request.form
    inspect['values'] = request.values
    inspect['first'] = request.form.get('first')
    inspect['last'] = request.form.get('last')
    inspect['path'] = request.path
    inspect['url'] = request.url
    inspect['base_url'] = request.base_url
    inspect['content_encoding'] = request.content_encoding
    inspect['content_length'] = request.content_length
    inspect['content_type'] = request.content_type
    inspect['host'] = request.host
    inspect['mimetype'] = request.mimetype
    if (request.method == 'POST' and
        'application/json' in request.headers.get('Content-Type')):
        inspect['receivedJson'] = request.get_json()
    return jsonify(inspect)
```

Invoke the “viewer” method

```
{ # query string sample
# curl 'localhost:3000/viewer?username=alex'

# post using curl
# curl --data "first=peter&last=parker" localhost:3000/viewer

# posting json
# curl -X POST -H "Content-Type: application/json" -d '{"first":"peter"}' localhost:3000/viewer
```

Serving static content

(Jinja2, <https://palletsprojects.com/p/jinja/>)

application.py

```
from flask import render_template

@app.route('/form')
def form():
    return render_template('form.html')
```

hello.html

```
<form method="POST" action="/addUser">
    firstname <input type="text" name="first" /><br>
    lastname  <input type="text" name="last" /><br>
    email     <input type="text" name="email" /><br>
    <input type="submit">
</form>
```

File system

```
/app.py
/templates
    /form.html
```

Enable Codespace's Simple Browser

Create configuration directory

```
% mkdir .devcontainer
```

Create configuration directory

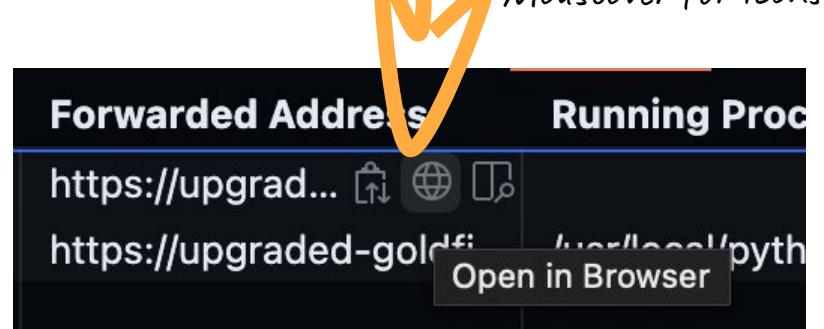
```
% touch devcontainer.json
```

Add to “devcontainer.json”

```
{
  "portsAttributes": {
    "3000": {
      "label": "Application",
      "onAutoForward": "openPreview"
    }
  }
}
```

Open Ports

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS 2
Port	Forwarded Address	Running Process	Visibility	Origin
○ 7860	https://upgraded-goldfish-wrjv6q...		<input type="radio"/> Public	Auto Forwarded
● 8000	https://upgraded-goldfish-wrjv6q...	/usr/local/python/3.10.8/bin/pyth...	<input type="radio"/> Public	Auto Forwarded
Add Port				





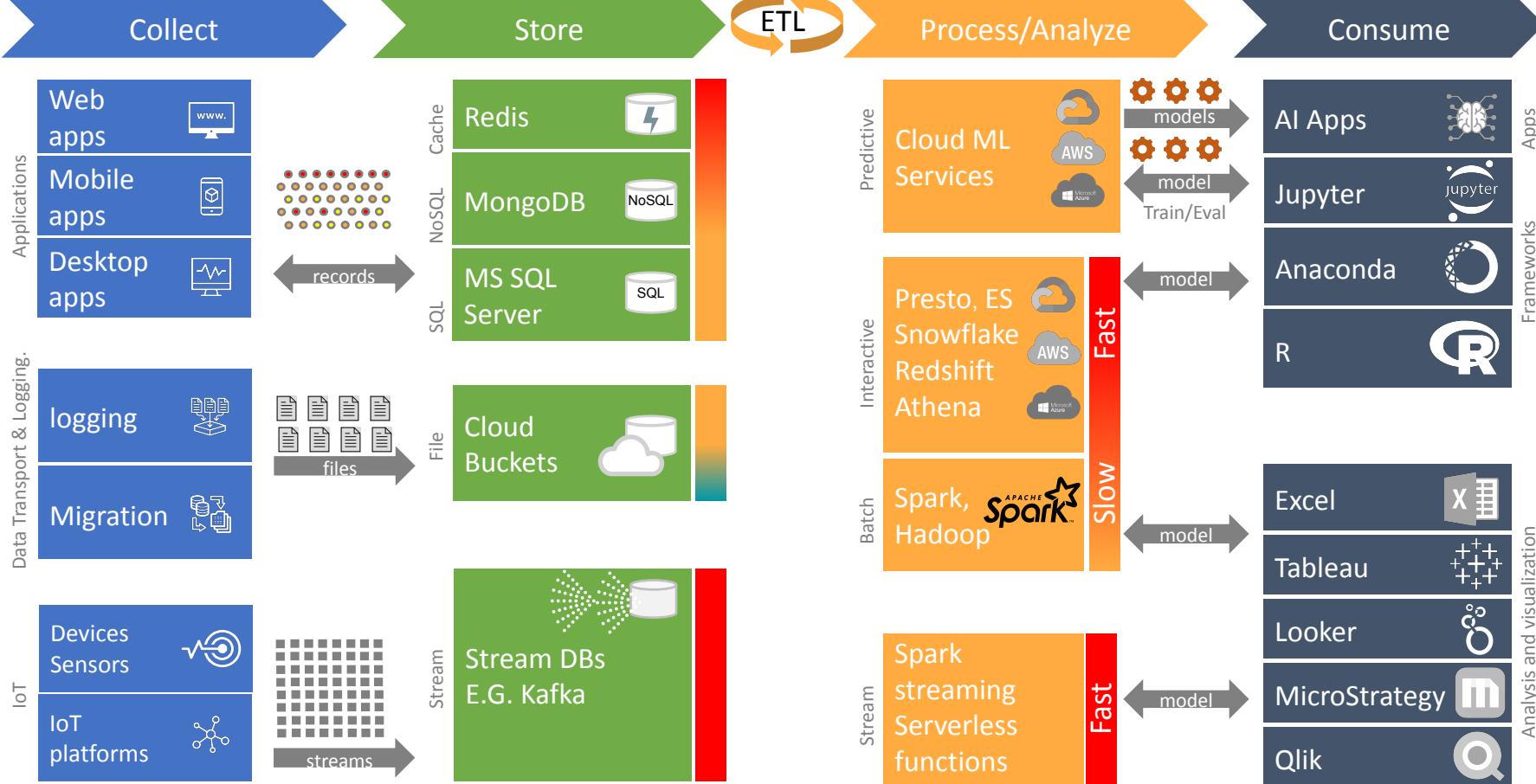
What about data?

What is data? Tell me about a data cycle you went through.

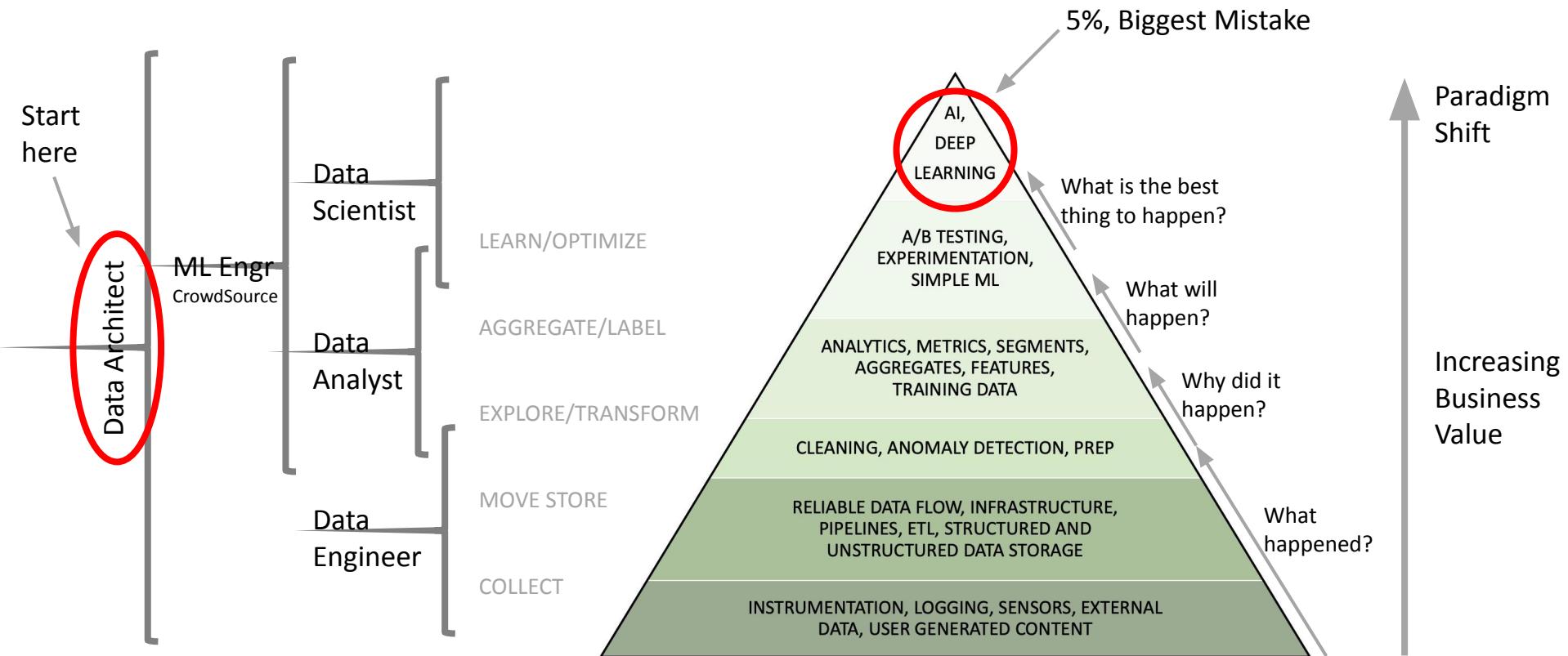
Data Cycle



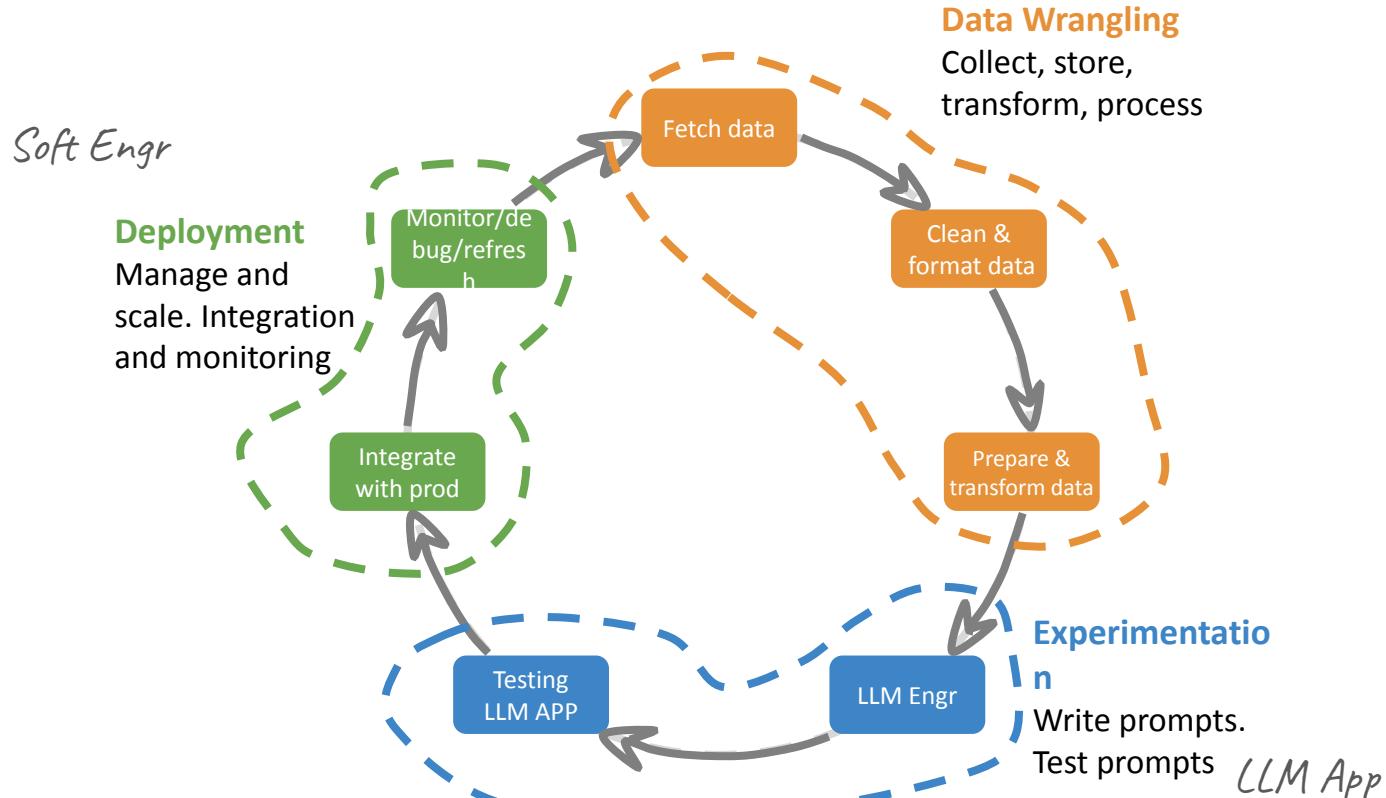
DATA ENGINEERING - DATA SCIENCE



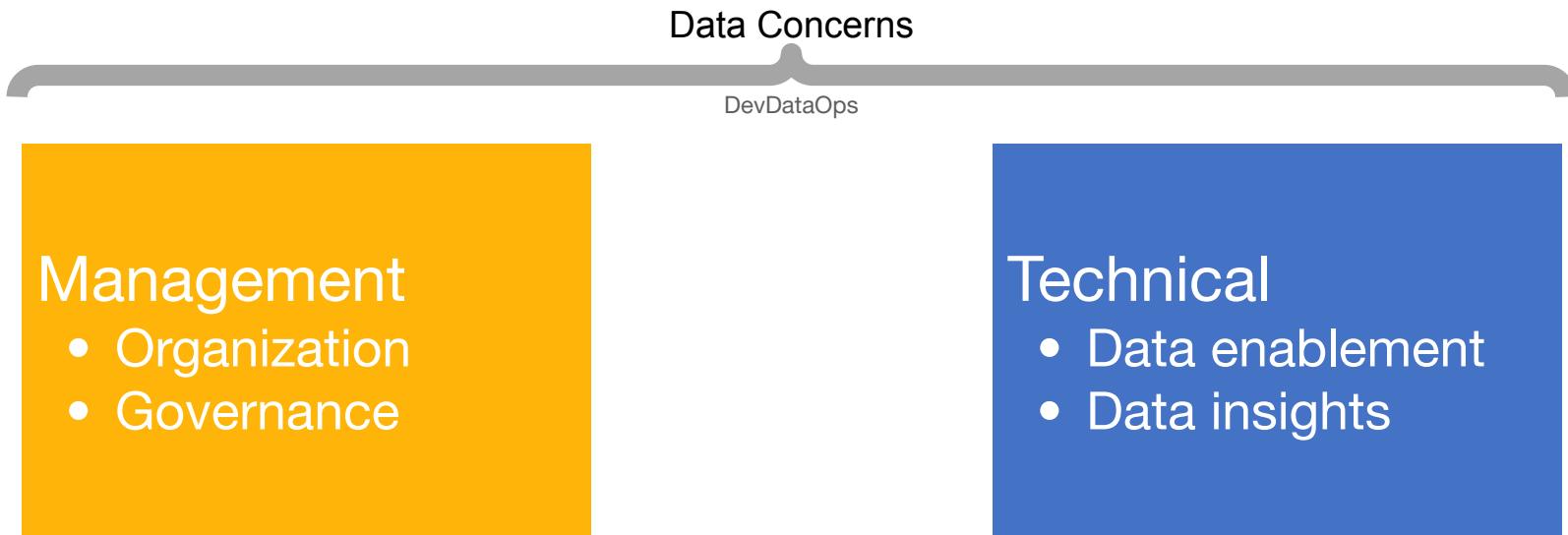
Data Hierarchy of needs



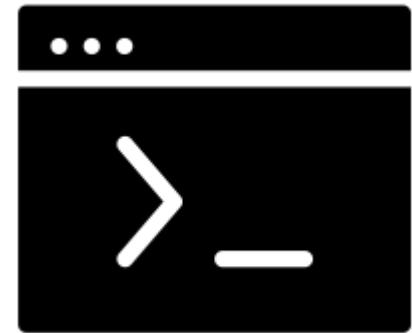
LLM APP CYCLE



Two concerns



As you scale (in the context of platforms)



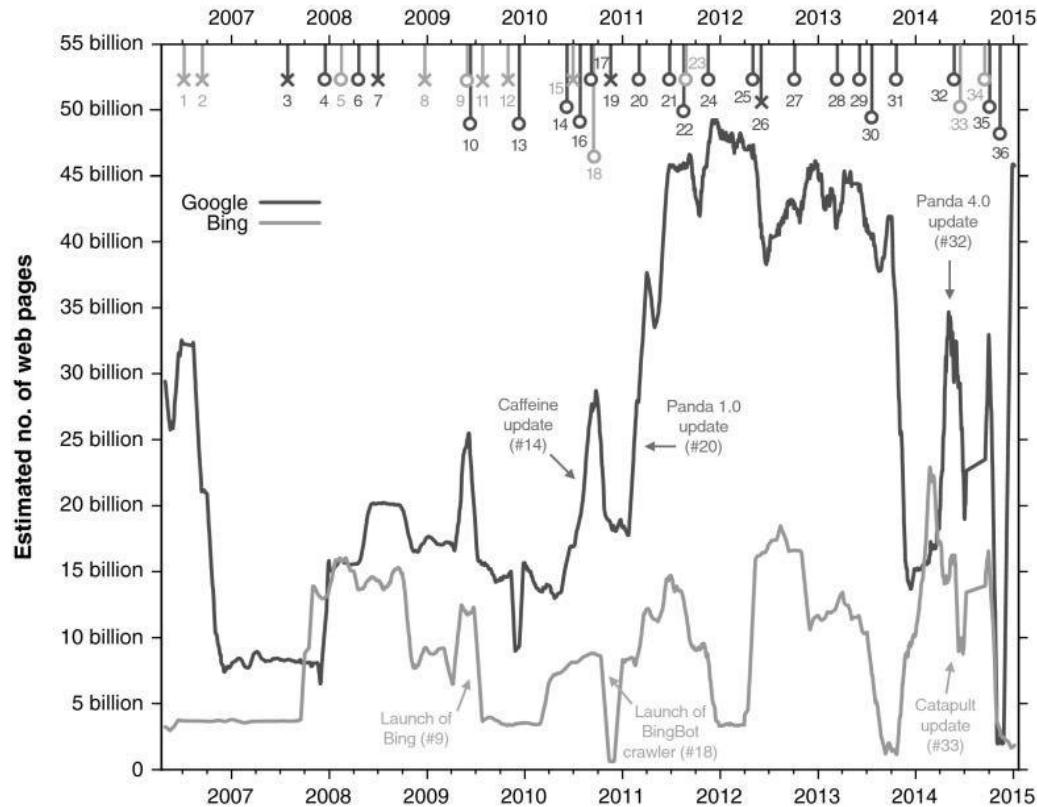


Sensemaking

Course Catalog

Google scraping/index

130,000,000,000
(130 trillion) web
pages in 2016



Search engine optimization (SEO)

Search engine optimization (SEO) is the practice of optimizing a website to help search engines understand and rank it higher in search results. The goal of SEO is to increase the amount of relevant, quality content on a website and the amount of traffic it receives from search engines.

SEO is important because search is one of the main ways people find content online. When a website ranks higher in search results, it can lead to more traffic and potentially more sales.

OpenAI, the maker of ChatGPT, has discussed training its next model, GPT-5, on transcriptions of public YouTube videos ... transcribing high-quality examples of video and audio on the internet using Whisper, its automatic speech-recognition tool



WSJ

Data is among several essential AI resources in short supply. EMIL LENDOF/THE WALL STREET JOURNAL, ISTOCK

For Data-Guzzling AI Companies, the Internet Is Too Small

Firms such as OpenAI and Anthropic are working to find enough information to train next-generation artificial-intelligence models

By [Deepa Seetharaman](#) [Follow](#)
April 1, 5:30 am ET



ARTIFICIAL INTELLIGENCE / TECH / GOOGLE

OpenAI transcribed over a million hours of YouTube videos to train GPT-4 / A New York Times report details the ways big players in AI have tried to expand their data access.

By [Wes Davis](#), a weekend editor who covers the latest in tech and entertainment. He has written news, reviews, and more as a tech journalist since 2020.

Apr 6, 4:29 PM EDT

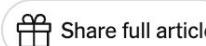


[45 Comments \(45 New\)](#)

Researchers at OpenAI's office in San Francisco developed a tool to transcribe YouTube videos to amass conversational text for A.I. development. Jason Henry for The New York Times

How Tech Giants Cut Corners to Harvest Data for A.I.

OpenAI, Google and Meta ignored corporate policies, altered their own rules and discussed skirting copyright law as they sought online information to train their newest artificial intelligence systems.



Share full article



By [Cade Metz](#), [Cecilia Kang](#), [Sheera Frenkel](#), [Stuart A. Thompson](#) and [Nico Grant](#)

Reporting from San Francisco, Washington and New York

Published April 6 Updated April 8

In its second ruling on Monday, the Ninth Circuit reaffirmed its original decision and found that scraping data that is publicly accessible on the internet is not a violation of the Computer Fraud and Abuse Act, or CFAA, which governs what constitutes computer hacking under U.S. law.

The Ninth Circuit's decision is a major win for archivists, academics, researchers and journalists who use tools to mass collect, or scrape, information that is publicly accessible on the internet. Without a ruling in place, long-running projects to archive websites no longer online and using publicly accessible data for academic and research studies have been left in legal limbo.



Security

Web scraping is legal, US appeals court reaffirms

Zack Whittaker / 12:16 PM PDT • April 18, 2022

Like “free speech” the limits are continuously challenged



Government & Policy

Meta drops lawsuit against web-scraping firm Bright Data that sold millions of Instagram records

Sarah Perez

/ 8:01 AM PST • February 26, 2024



Featured Article

News outlets are accusing Perplexity of plagiarism and unethical web scraping

Ambiguity around copyright laws and AI web crawlers complicate matters

Rebecca Bellan / 8:00 AM PDT • July 2, 2024



Government & Policy

TechCrunch Minute: Meta acknowledges it's scraping all public posts for AI training

Anthony Ha

/ 9:00 AM PDT • September 13, 2024

Q&A