



NodeJS

MERN Stack



MySQL



Express

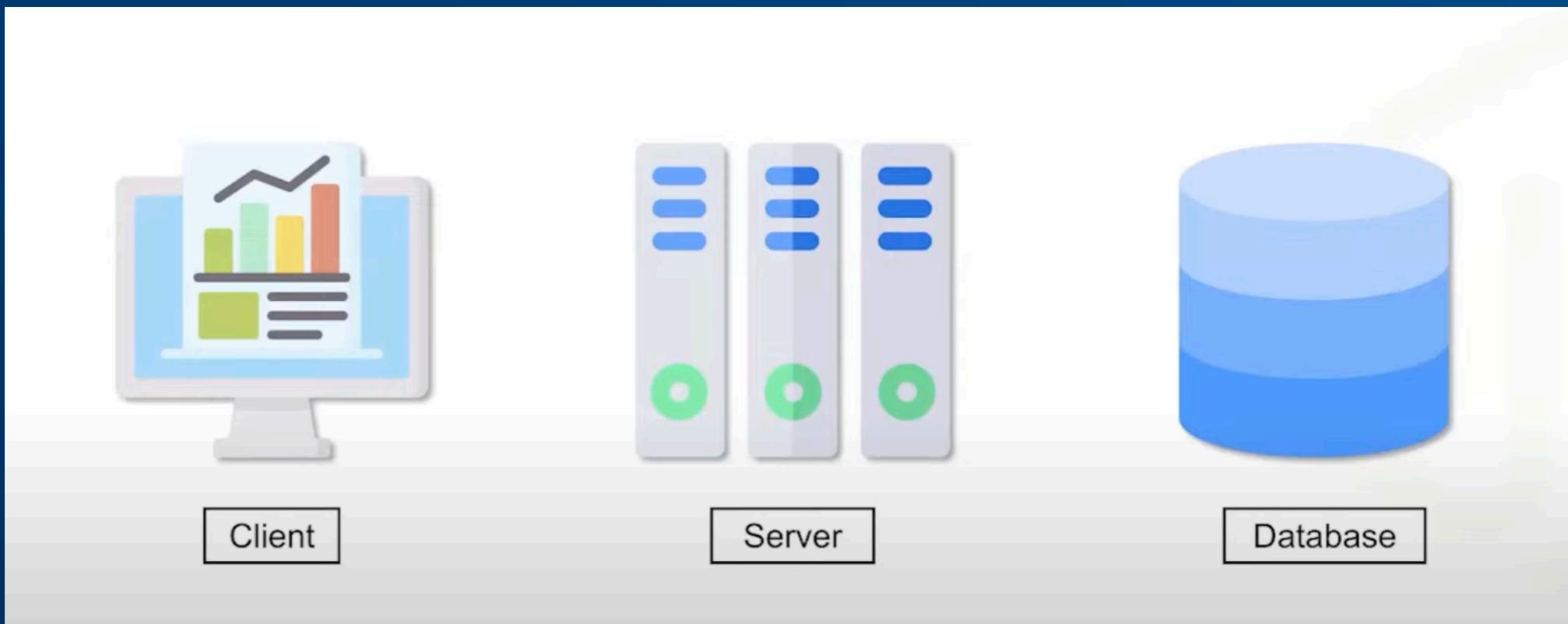


React



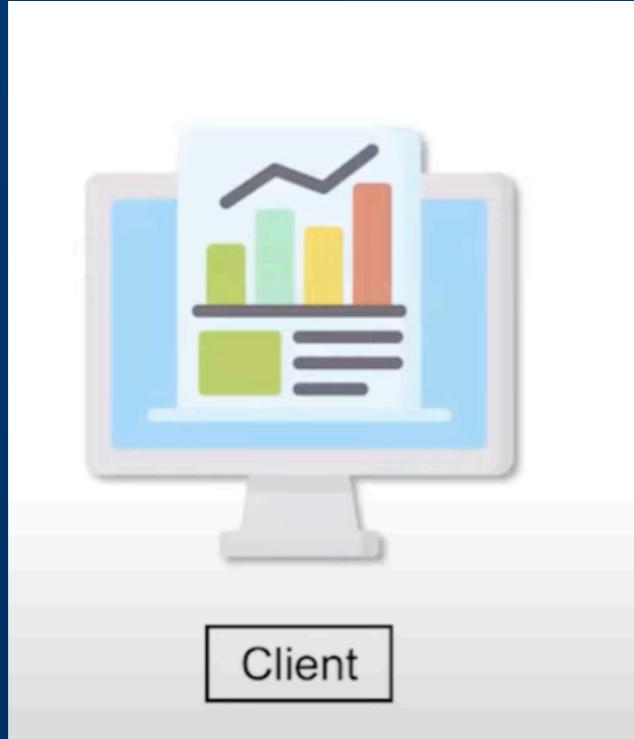
NodeJS

Client Server Architecture



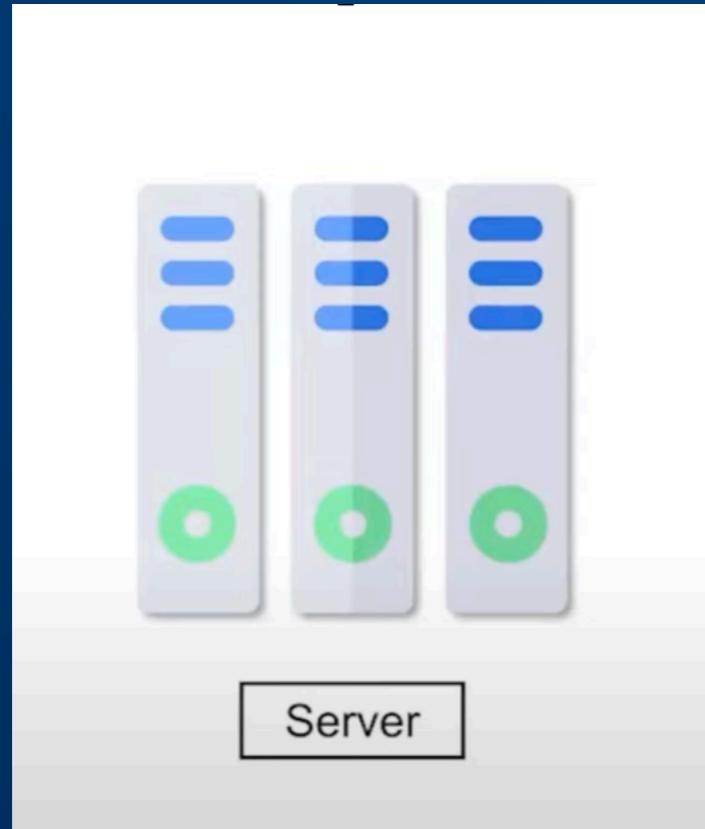
Client

Front end Frameworks



- ReactJS
- AngularJS
- Next.js
- Vue.js
- Bootstrap
- Jquery and more

Server

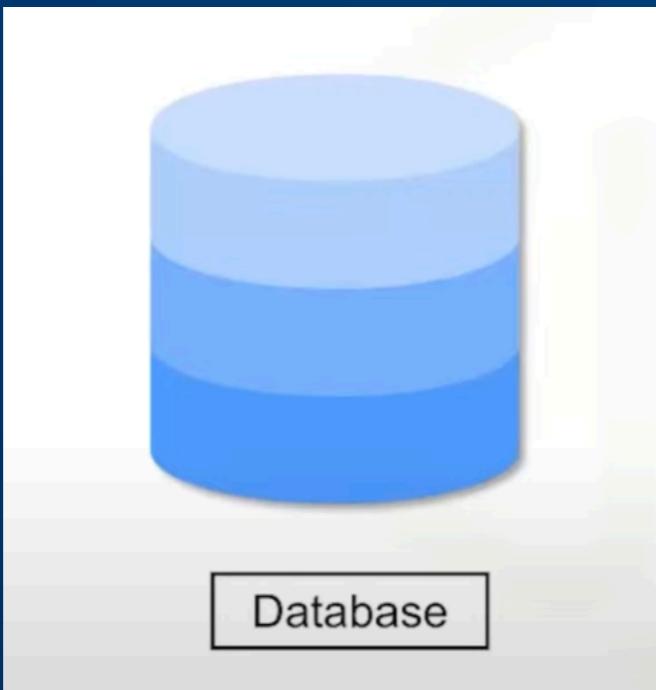


Backend Servers

- NodeJS
- Python
- Java
- Ruby
- PHP and more

Database

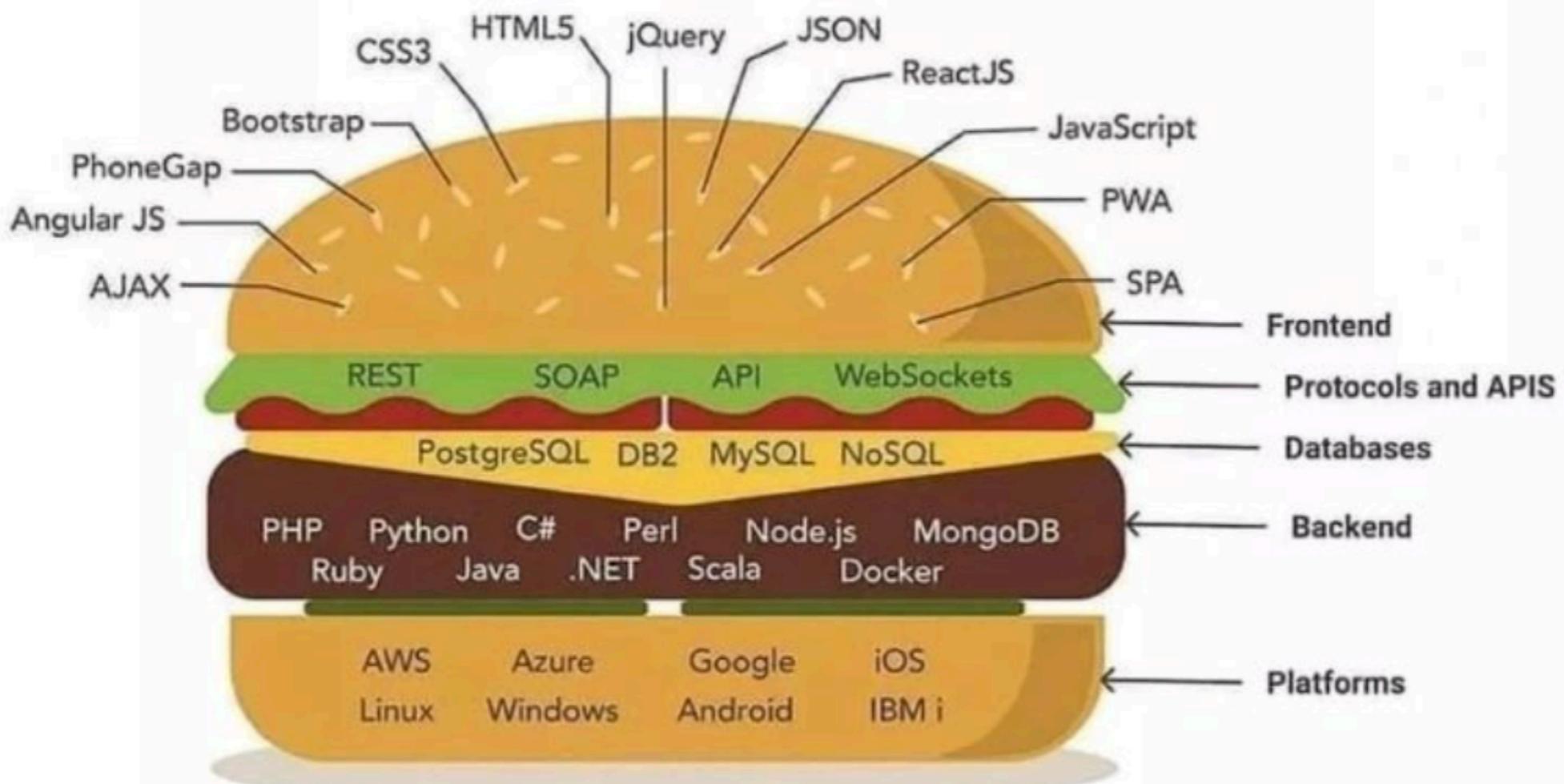
Backend Databases



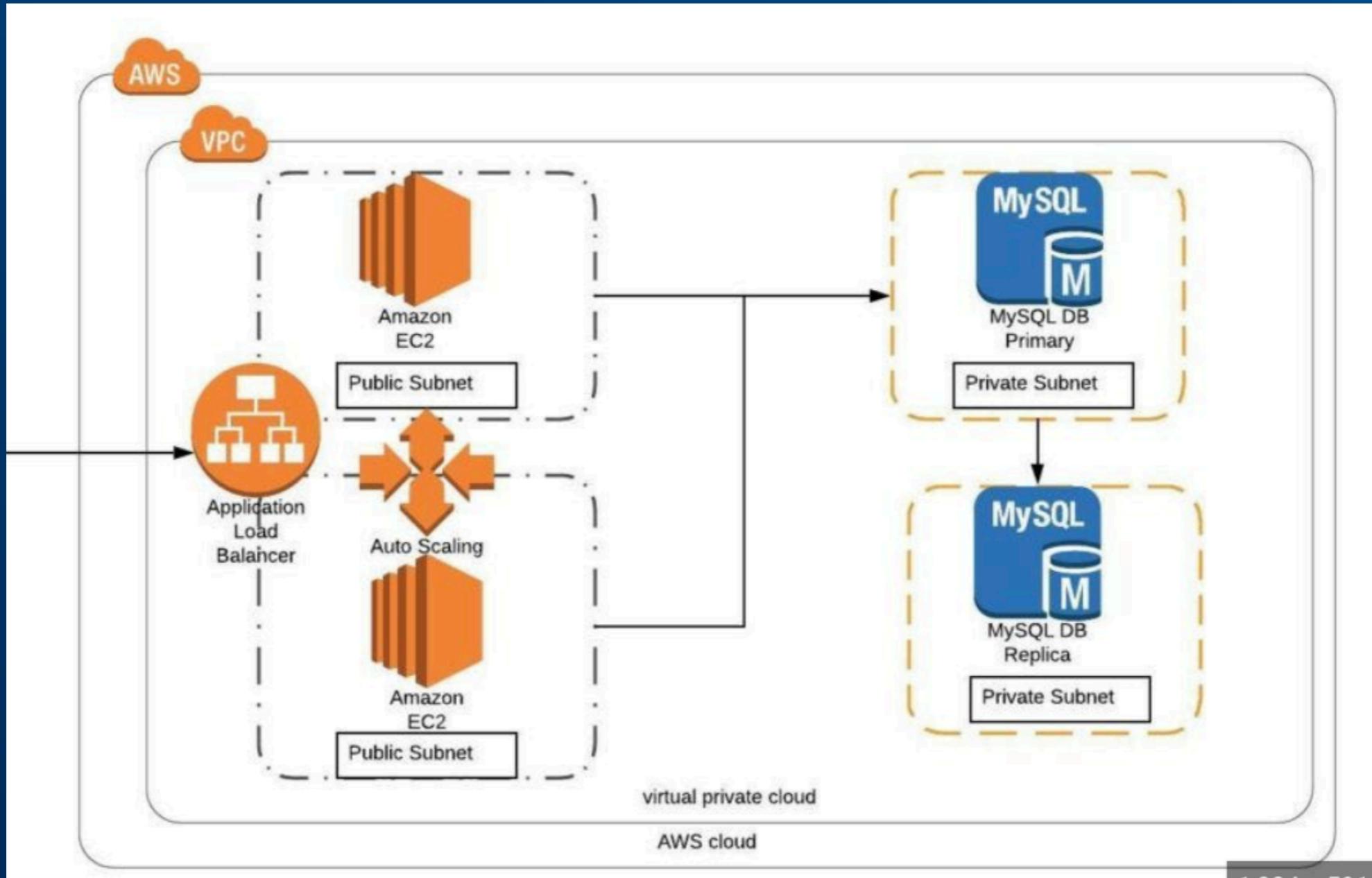
- MySQL
- MongoDB
- PostgreSQL
- DynamoDB
- more

Client Server Architecture

“Full Stack Developer”

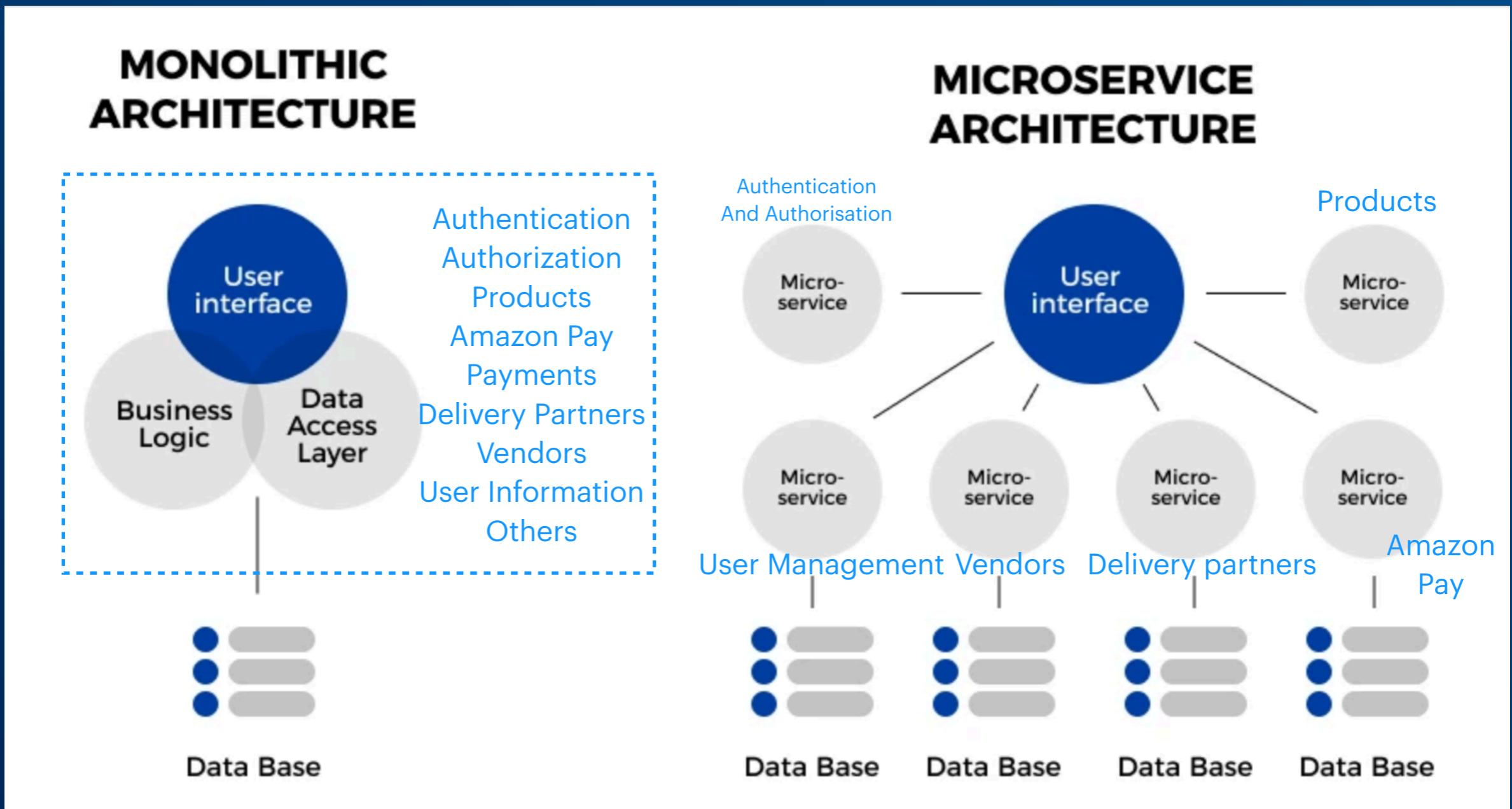


AWS Hosting

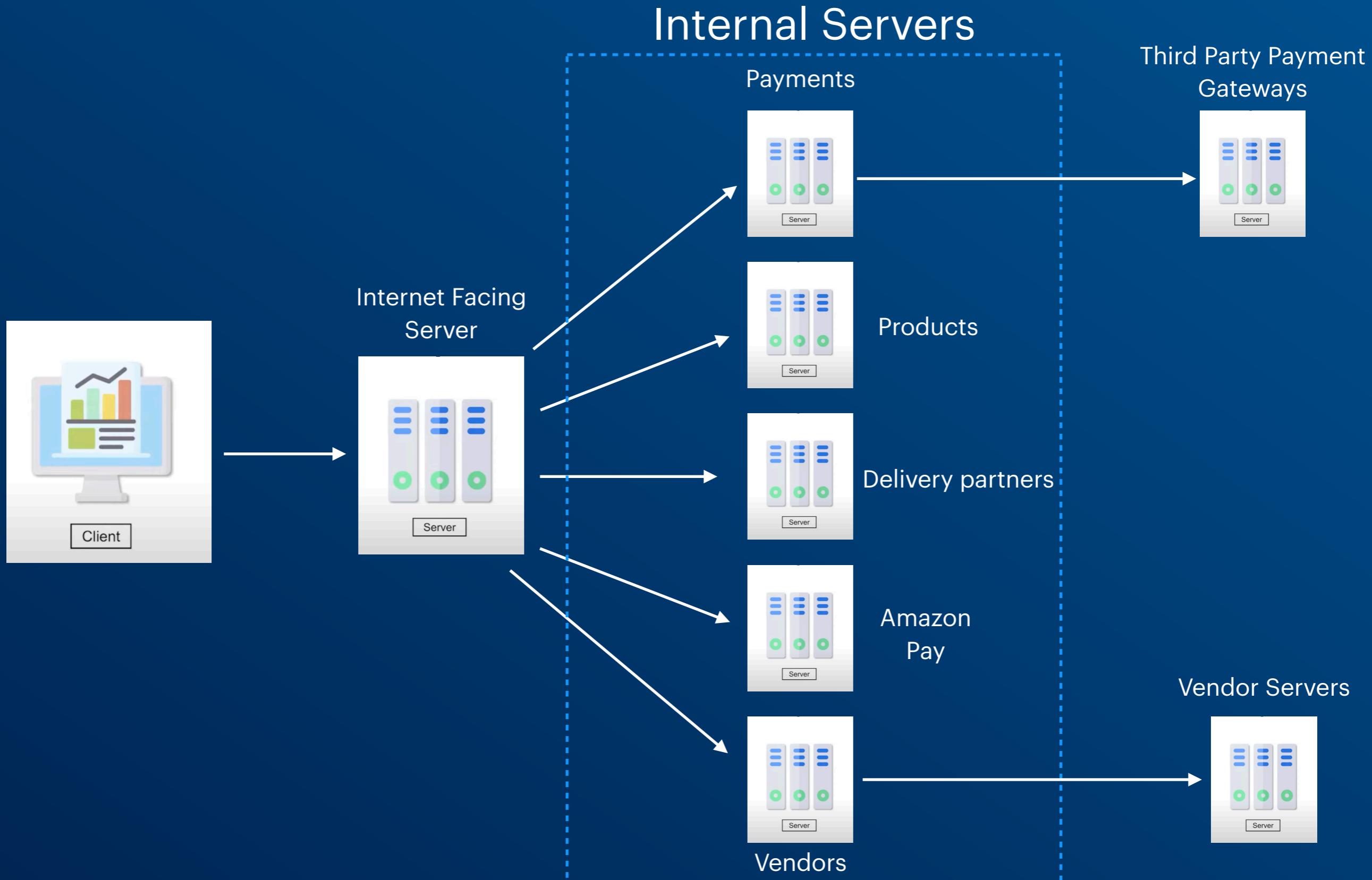


Client Server Architecture

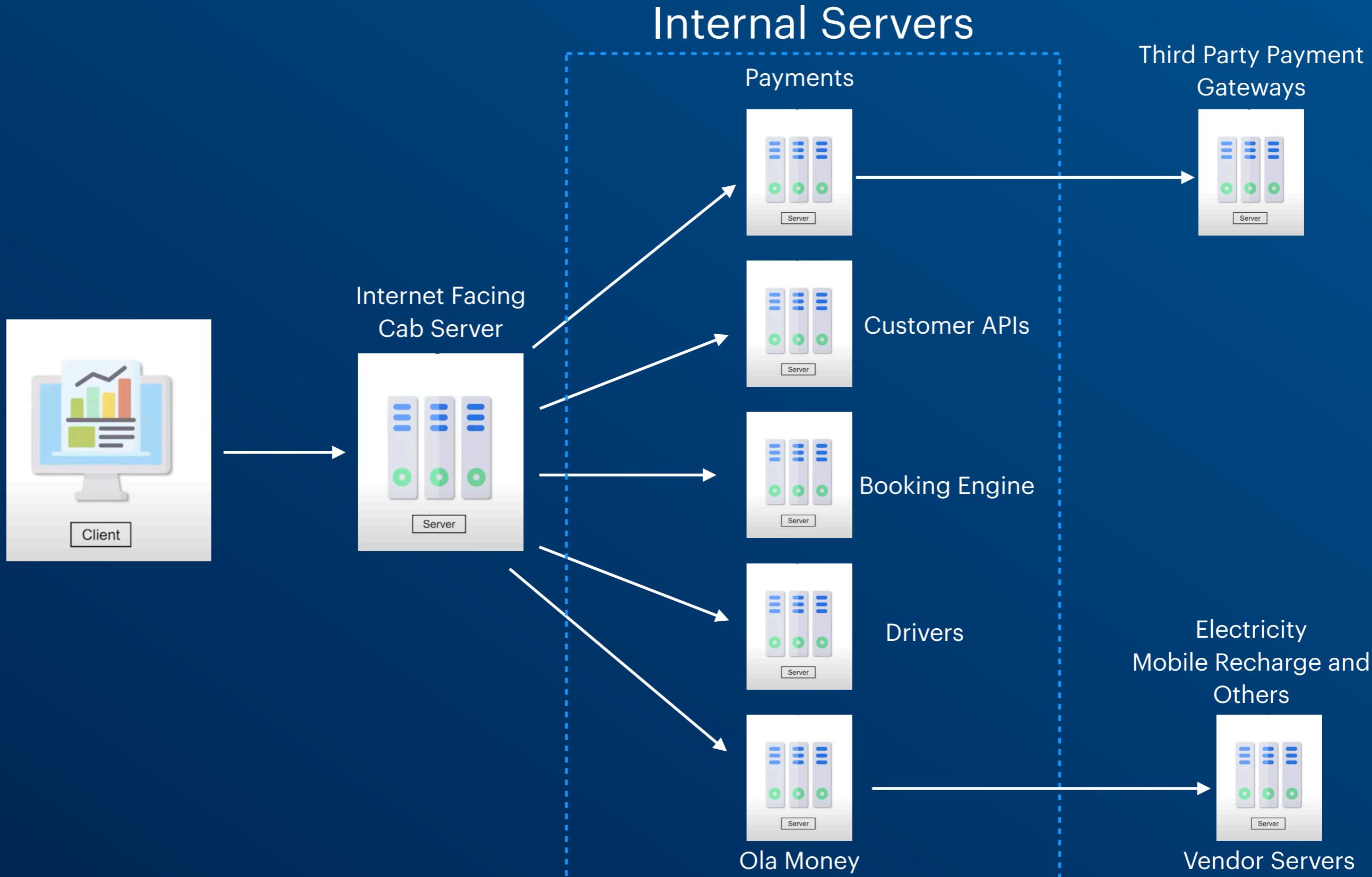
Ex: Amazon



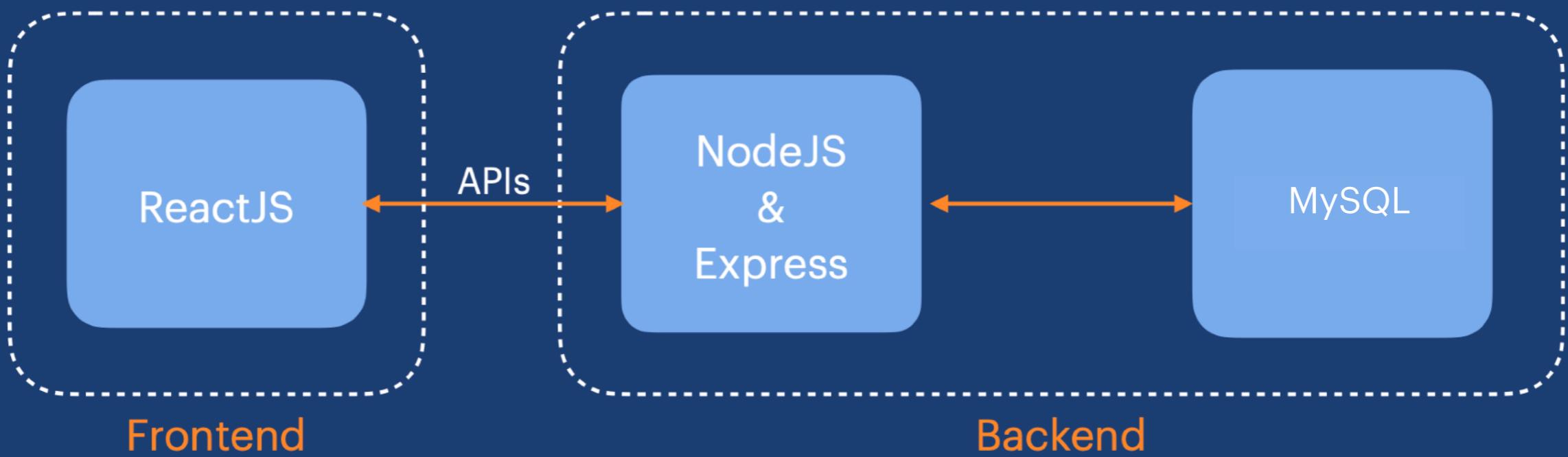
Amazon - Client Server Architecture



OLA - Client Server Architecture



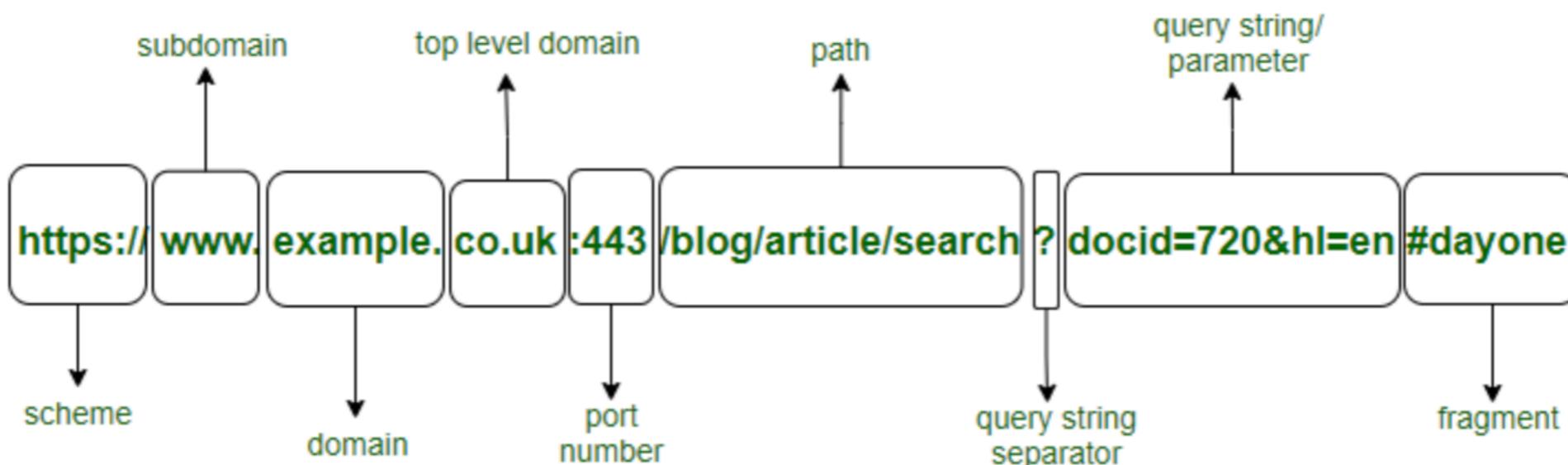
Client Server Architecture



URL

Parts of a URL

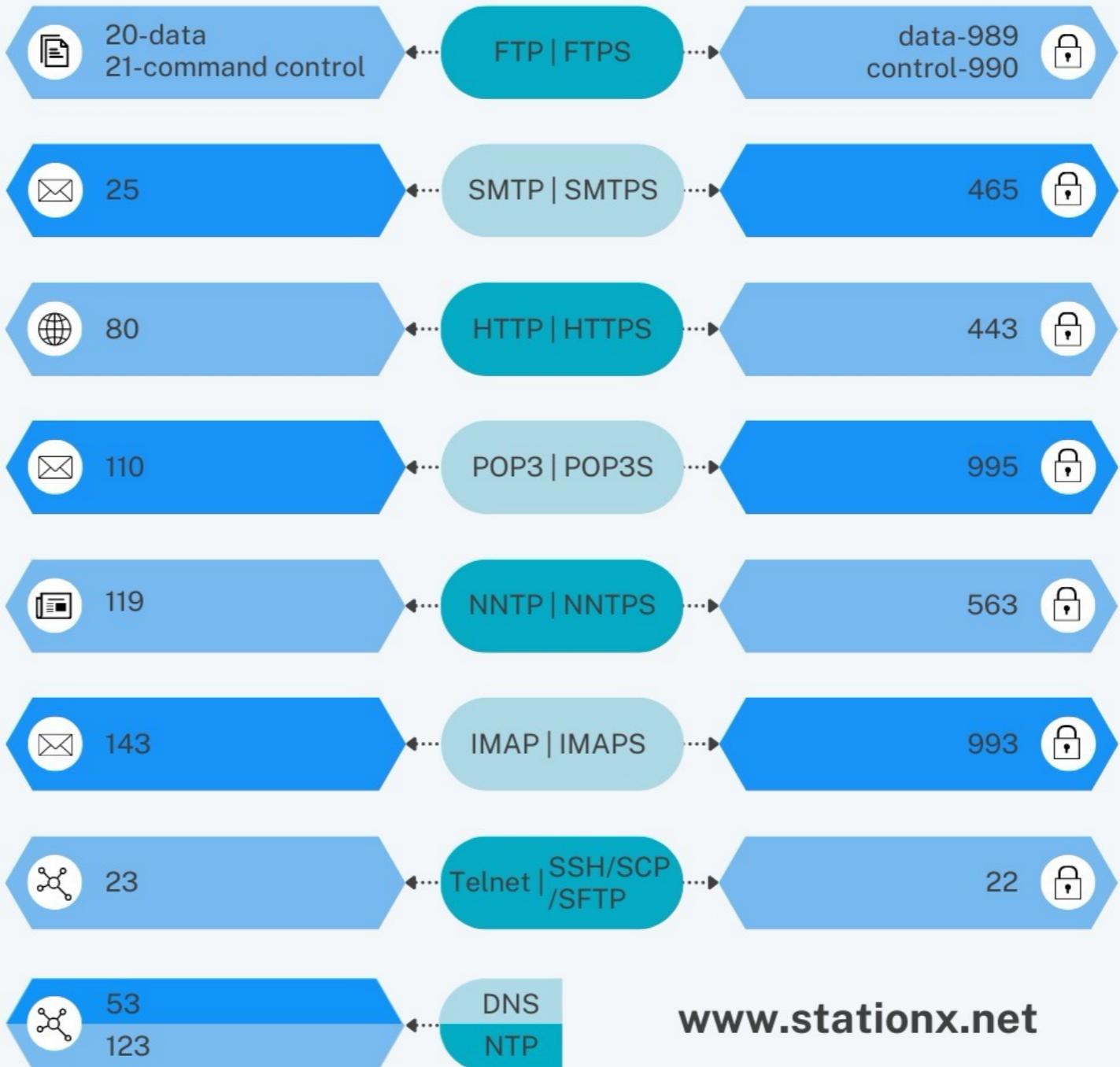
URL : `https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayone`



Ports

Well-Known Ports: Unencrypted vs Encrypted

Must-know commonly used ports to memorize



API

Two software components to communicate with each other using a set of definitions and protocols.



API

- REST APIs
- WebSocket APIs
- ~~SOAP APIs~~

API

- Private APIs
- Public APIs
- Partner APIs

API Format

Request

Pretty Raw Hex ⌂ ⌂ ⌂ ⌂

```
1 POST /doLogin HTTP/1.1
2 Host: demo.testfire.net
3 Cookie: JSESSIONID=97D77081A17B397D489D5B7BE99D46CE
4 Content-Length: 37
5 Cache-Control: max-age=0
10 Origin: https://demo.testfire.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/124.0.6367.118 Safari/537.36
18 Referer: https://demo.testfire.net/login.jsp
22 Connection: close
23
24 uid=admin&passw=admin&btnSubmit=Login
```

API/HTTP Methods

Get

To **read** the data from the server

POST

To **create** the data in the server

Put

To **update** the data in the server

Delete

To **Delete** the data in the server

CRUD Operations

POST

To CREATE the data in the server

Get

To READ the data from the server

Put

To UPDATE the data in the server

Delete

To Delete the data in the server

Example for API methods

Method	Sample
GET	GET /api/address/1
POST	POST /api/address/
PUT	PUT /api/address/1
DELETE	DELETE /api/address/1

Example for API methods

Method	Sample
GET	GET /api/address?address_id=1
POST	POST /api/address
PUT	PUT /api/address
DELETE	DELETE /api/address

HTTP Status Codes

1xx: Information	3xx: Redirection	4xx: Client Errors
100 - Continue Server accepts initial request; client should continue.	301 - Moved Permanently Resource has been moved permanently.	400 - Bad Request Request cannot be processed.
101 - Switching protocols Server acknowledges a switch in protocol.	302 - Found Resource has been moved temporarily.	401 - Unauthorized Authentication needed.
102 - Processing Request is processing (WebDAV).	303 - See Other Redirect to a different resource, typically after a POST operation.	403 - Forbidden Client is authenticated but lacks permissions.
103 - Early Hints Hints for preloading resources.	304 - Not Modified Resource hasn't changed since the last request used for caching purposes.	404 - Not Found Resource cannot be found.
2xx: Success	5xx: Server Errors	405 - Method Not Allowed The HTTP method used isn't allowed for the resource.
200 - OK Request succeeded.	500 - Internal Server Error Generic error message indicating a server-side issue.	406 - Not Acceptable Server can't produce a response that matches the client's Accept headers.
201 - Created Request fulfilled new resource created.	502 - Bad Gateway Upstream server sent an invalid response.	409 - Conflict Request conflicts with the current server state.
202 - Accepted Request accepted, but not yet completed.	503 - Service Unavailable Server temporarily unable to handle the request.	413 - Payload Too Large The request body is too large for the server to process.
204 - No Content Response is successful, but there's no content to send in the body.	504 - Gateway Timeout Upstream server did not respond in time.	429 - Too Many Requests Client has made too many requests in a given amount of time.
206 - Partial Content Partial resource sent (for byte-serving).		

Request Headers

Request

Pretty Raw Hex

1 POST /api/login HTTP/1.1
2 Host: 127.0.0.1:8000
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
12 Content-Type: application/json
16 Connection: close
17 Content-Length: 60
18
19 {
20 "email": "sunil@sunil.com",
21 "password": "sunil"
22 }

Response Headers

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Access-Control-Allow-Credentials: true
5 Set-Cookie: user_token=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InN1bmLsQHN1bmLsLmNvbSIiZpcnN0X25hb
WUi0iJzdW5pbCIsImxhc3RfbmFtZSI6Imt1bWFyIiwicGhvbmVfbnVtYmVyIjo5MDM2MDc2MTYzLCJ1c2VyX2l
kIjoxMCviaWF0IjoxNzE2MDMwNzEzLCJleHAiOjE3MTYwMzQzMjN9.yP-6pC88iNWuxF6H0ZwE4-bk4UnuMpxM
8rt5AXITQSg; Max-Age=3000; Domain=localhost; Path=/; Expires=Sat, 18 May 2024 12:01:53
    GMT; HttpOnly; SameSite=Lax
6 Content-Type: application/json
7 Date: Sat, 18 May 2024 11:11:53 GMT
8 Connection: close
9 Content-Length: 316
10
11 {
    "message": "Login successful!",
    "user_token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InN1bmLsQHN1bmLsLmNvbSIiZpcnN0X2
    5hbWUi0iJzdW5pbCIsImxhc3RfbmFtZSI6Imt1bWFyIiwicGhvbmVfbnVtYmVyIjo5MDM2MDc2MTYzLCJ1c2
    VyX2lkIjoxMCviaWF0IjoxNzE2MDMwNzEzLCJleHAiOjE3MTYwMzQzMjN9.yP-6pC88iNWuxF6H0ZwE4-bk4
    UnuMpxM8rt5AXITQSg"
}
```

JSON Data

```
{  
  "email": "sunil@sunil.com",  
  "password": "sunil"  
}
```

```
{  
  "address": "JP nagar",  
  "pincode": "2122112",  
  "city": "Bangalore",  
  "state": "Karnataka",  
  "country": "India"  
}
```

```
{  
  "address": {  
    "line1": "JP nagar",  
    "line2": "1st cross"  
  },  
  "pincode": "2122112",  
  "city": "Bangalore",  
  "state": "Karnataka",  
  "country": ["India", "Australia"]  
}
```

application/www-x-form-urlencoded

Request

Pretty Raw Hex

1 POST /doLogin HTTP/1.1
2 Host: demo.testfire.net
3 Cookie: JSESSIONID=F4A367FBFEA427CAF30DEC217BD98506; AltoroAccounts=
"ODAwMDAwfkNvcnBvcmF0ZX4xLjBFMjZ80DAwMDAxfkNoZWNrav5nf105Ljk50Tk40DE1NTMyNTU5NEUyNXw=";
4 Content-Length: 37
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not-A.Brand";v="99", "Chromium";v="124"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
0 Origin: https://demo.testfire.net
1 Content-Type: application/x-www-form-urlencoded
2 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/124.0.6367.118 Safari/537.36
3 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
4 Sec-Fetch-Site: same-origin
5 Sec-Fetch-Mode: navigate
6 Sec-Fetch-User: ?1
7 Sec-Fetch-Dest: document
8 Referer: https://demo.testfire.net/login.jsp
9 Accept-Encoding: gzip, deflate, br
0 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
1 Priority: u=0, i
2 Connection: close
3
4 uid=admin&passw=admin&btnSubmit=Login