1. Rate limiting
2. Logging & monitoring
3. Security & HTTPS
4. API design
5. Proxy

① Rate limiting

→ mechanism that restricts the rate or frequency at which requests can be made to the system or API.

Helps control

1. resource utilization
2. prevents abuse
3. maintain system stability

Example :

→ Social media use rate limiting to prevent users from making too many requests in a short time.

→ Platform might limit a user to 100 API requests per hour.

Twitter → "Rate limit exceeded"

↳ If they exceed the max number of API calls.

→ 500 dms
→ 2400 tweets
→ 400 accounts followed

in one day

② Logging & monitoring

ⓐ Logging

→ A log file records details of events occuring in an application

→ It is helpful to debug the flow of an event in the system.

ⓐ Logging

→ A log file records details of events occuring in an application

→ It is helpful to debug the flow of an event in the system.

Need for logging:

→ on facing a failure, logging helps to pinpoint when and how the system failed.

→ helps to find root cause of the failure.

Benefit:

→ saves time to find, diagnose, resolve issues.

```
if (          )
{
    logger.error("Document upload failed, maximum
                  limit exceeded");


}
else
{
    //

}
```

② Monitoring

↳ observation of the system's performance, health & behaviour

Why monitoring is Important?

1) Early issue detection
2) performance optimization
   - insights about resource utilization, response time
3) capacity planning
   - tracking resource usage
   - anticipate when additional resources needed

components of monitoring

metrics : measurement about system

1. cpu uhilizahion
2. memory usage
3. network traffic

components of monitoring

metrics : measurement about system

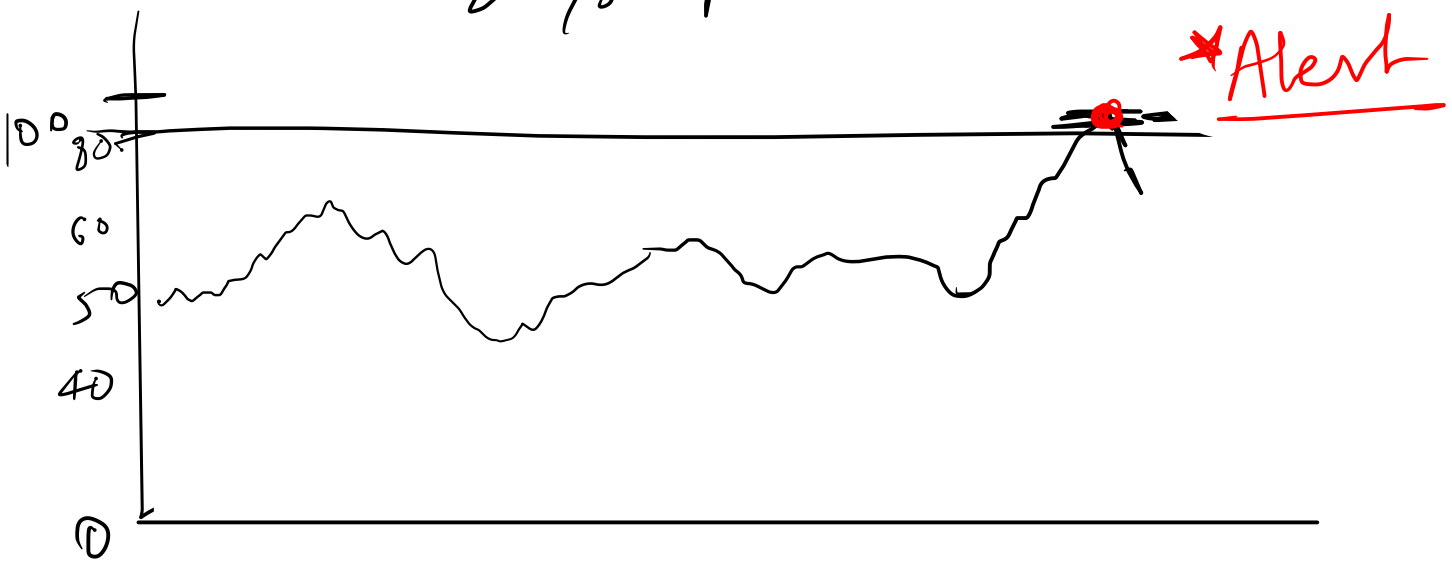
1. cpu uhilizahion
2. memory usage
3. network traffic

Alerting:

process of setting up thresholds to trigger notifications when predefined conditions are met.

Example: Alert when cpu usage is 80%.

80% cpu utilization



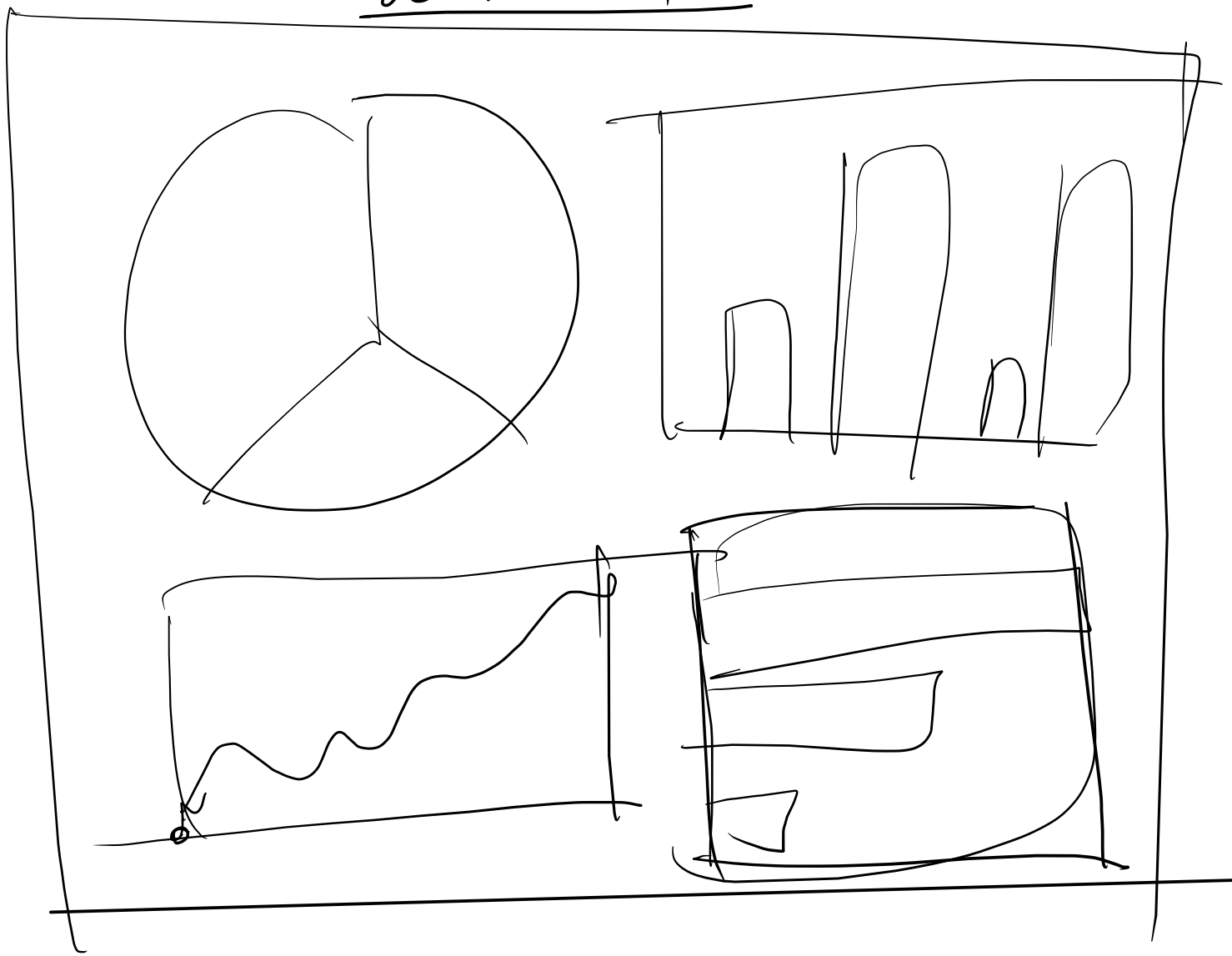
monitoring

Tools & Platforms

open-source : Kibana

Clouds-based : AWS cloudwatch

dashboard



③ Security

→ protecting data, resources, systems from unauthorized access, breaches, various threats.

① Confidentiality :

Sensitive info / data is available only to the right set of users.

② Integrity :

data is complete, trustworthy, and not modified by any unauthorized user.

(3) Authentication 

verifying the identity of users.

→ Password

→ multi-factor Authentication (MFA)

2-Factor Authentication

(4) Authorization

→ user permission & roles

→ giving rights to user

→ what actions users are allowed to perform

(5) Availability

→ ensure systems are available when needed.

→ protecting against DOS (Denial of service) attack

↓

(sends rapid & continuous request to a target server to overload the server)

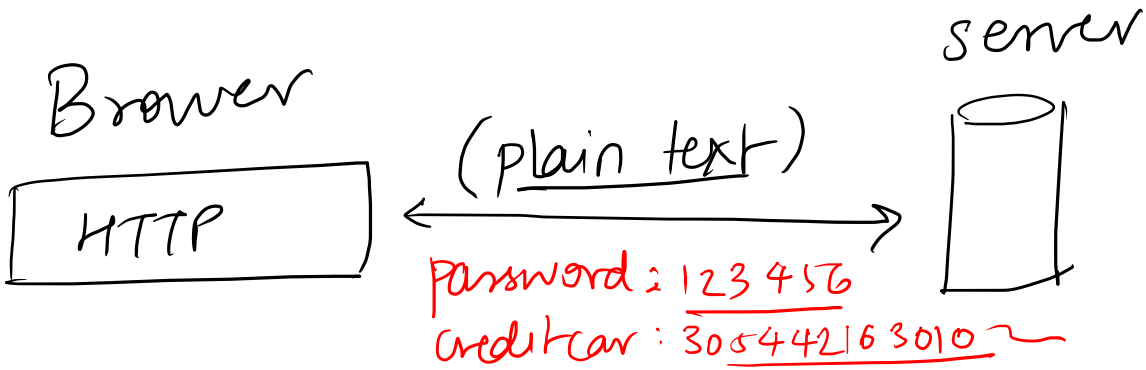
HTTP → Hyper Text Transfer Protocol

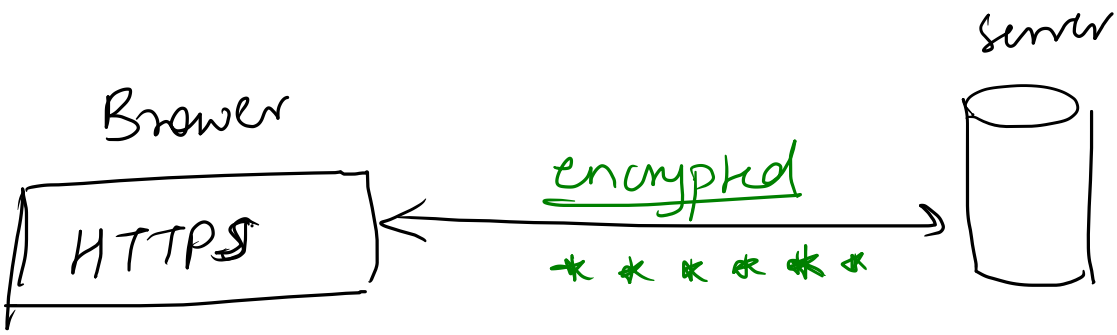
HTTPS

↓

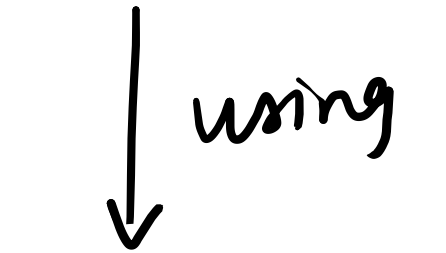
secure

→ SSL certificate


Brower
server
HTTP
(plain text)
password : 123456
creditcar : 3054421630 10


server
Brower
HTTPS
encrypted

With HTTPS data is sent in an encrypted form



(transport layer security)

API design

⇒ Application Programming Interface

Weather API

1. Request

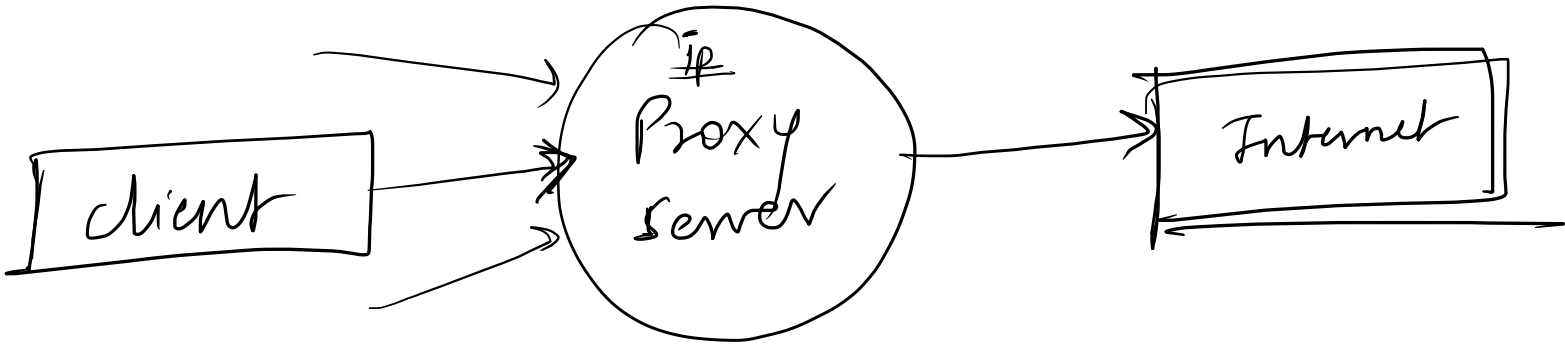
→ Request HTTP GET request
→ Parameters for location

Request → "New York City"

Response: JSON format

```
{ "location": "New York City"
  "temperature": "72°F"
  "humidity": 45%
  "windspeed": "10mph"
}
```

⑤ Proxy



→ server that sits between a client and the internet

① Forward Proxy

→ Protects the clients online identity.
(IP address of user is hidden)

→ block access to certain content

→ filtering rules / social network blocked

② Reverse Proxy

① protects the web server
(IP address of web server from client)

→ prevent DOS attack

→ caching the content of frequently visited website (fast access)