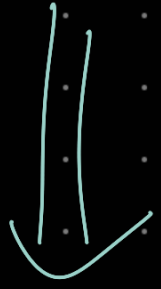
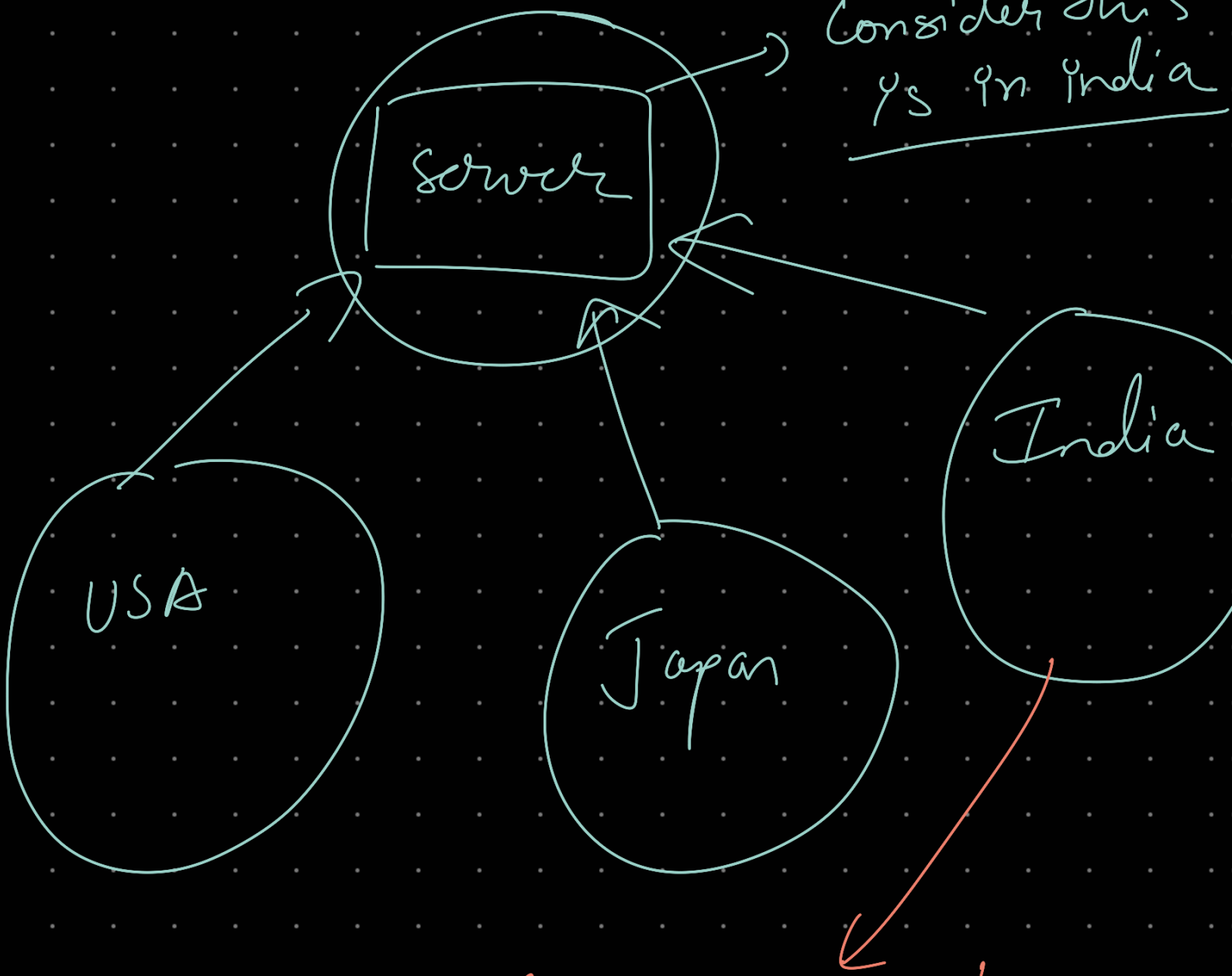
[System Design / HLD Building Blocks]

[CDN]

- Content Delivery Network
  - Physical connected / distributed geographically over the world
    - Used for Caching / storing static files
      - Html, js, css
      - Images, gif etc.





For people connecting from India → Time to connect to server is less as geographically server is on India

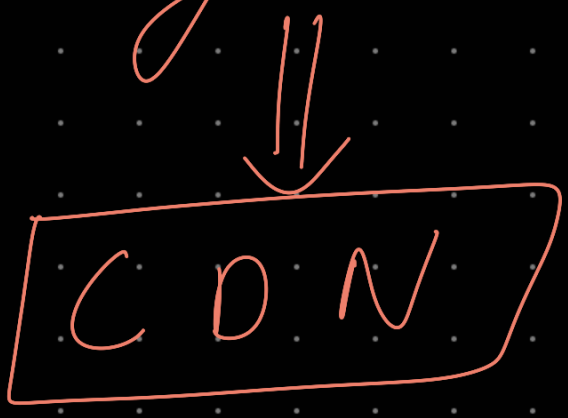
↓

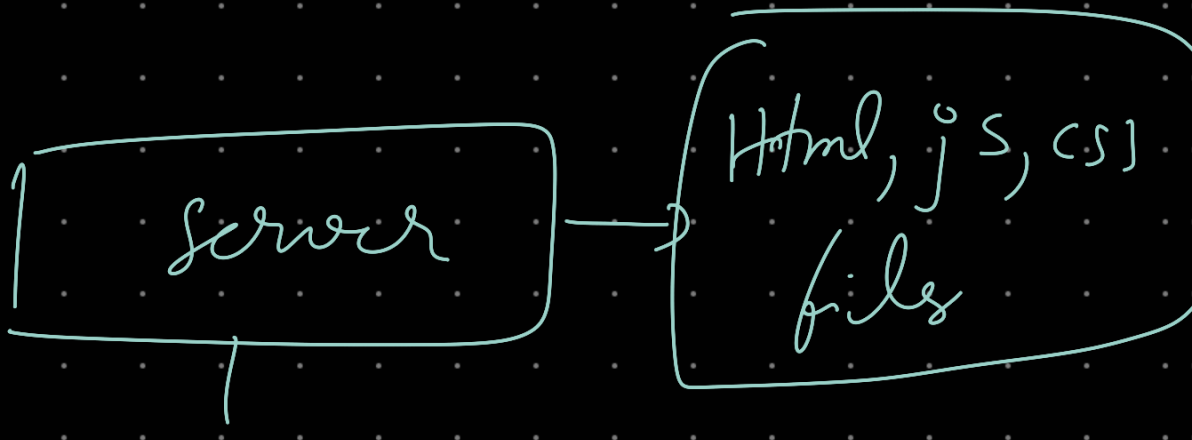
Lets say server has Html, js, css and some image files in its HDD → For Indian customers all these files takes less time to load.

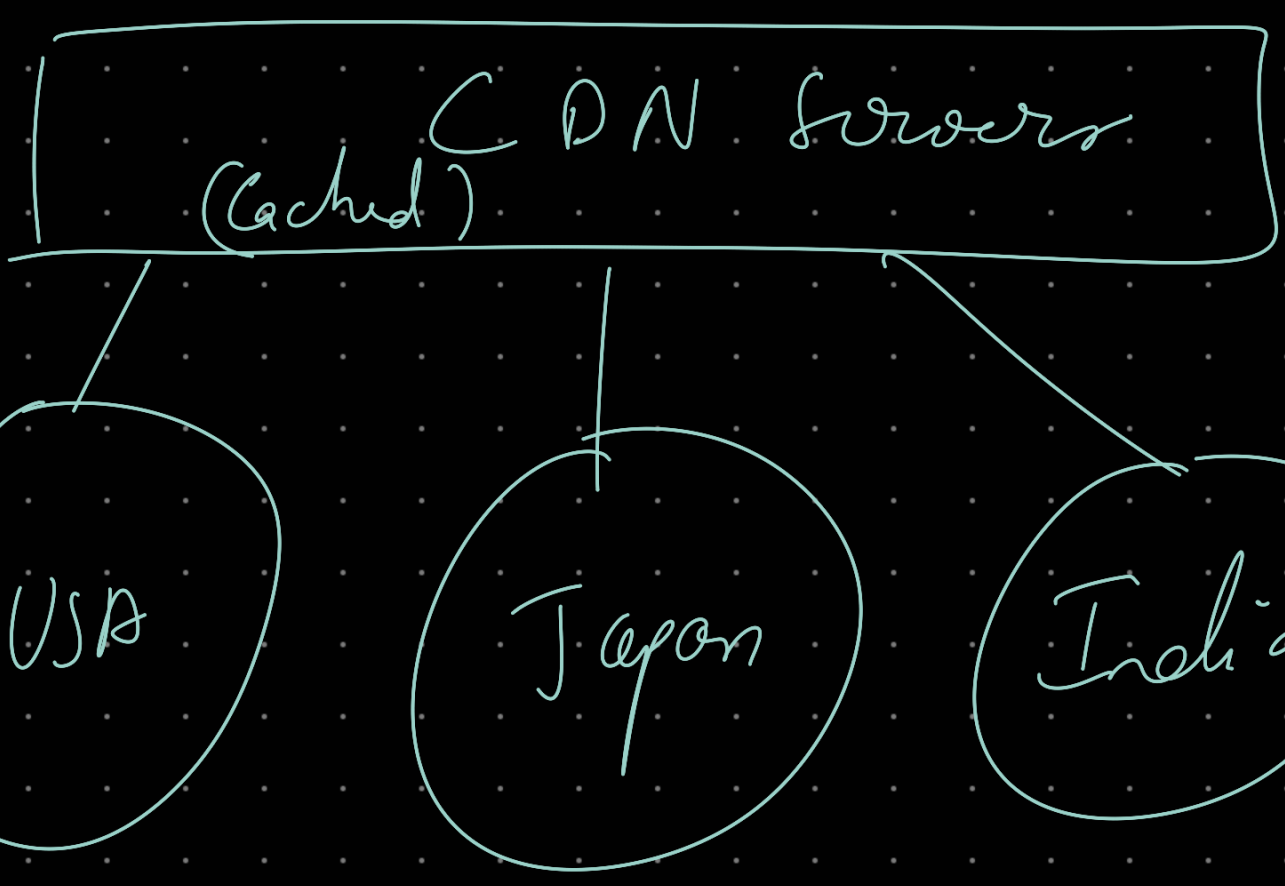
↳ What if you have users across globle

How to optimise and decrease loading time for static files

Need Multiple Cache Servers physically across globe



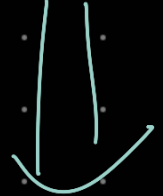




* CDN is generally provided by large companies like Amazon, cloudfare, as it is not easy to distribute server.

* Note → CDN server don't do much processing, just serve static content.

1. Increase Availability
2. Prevent DDOS to origin server
3. Decrese Repose time



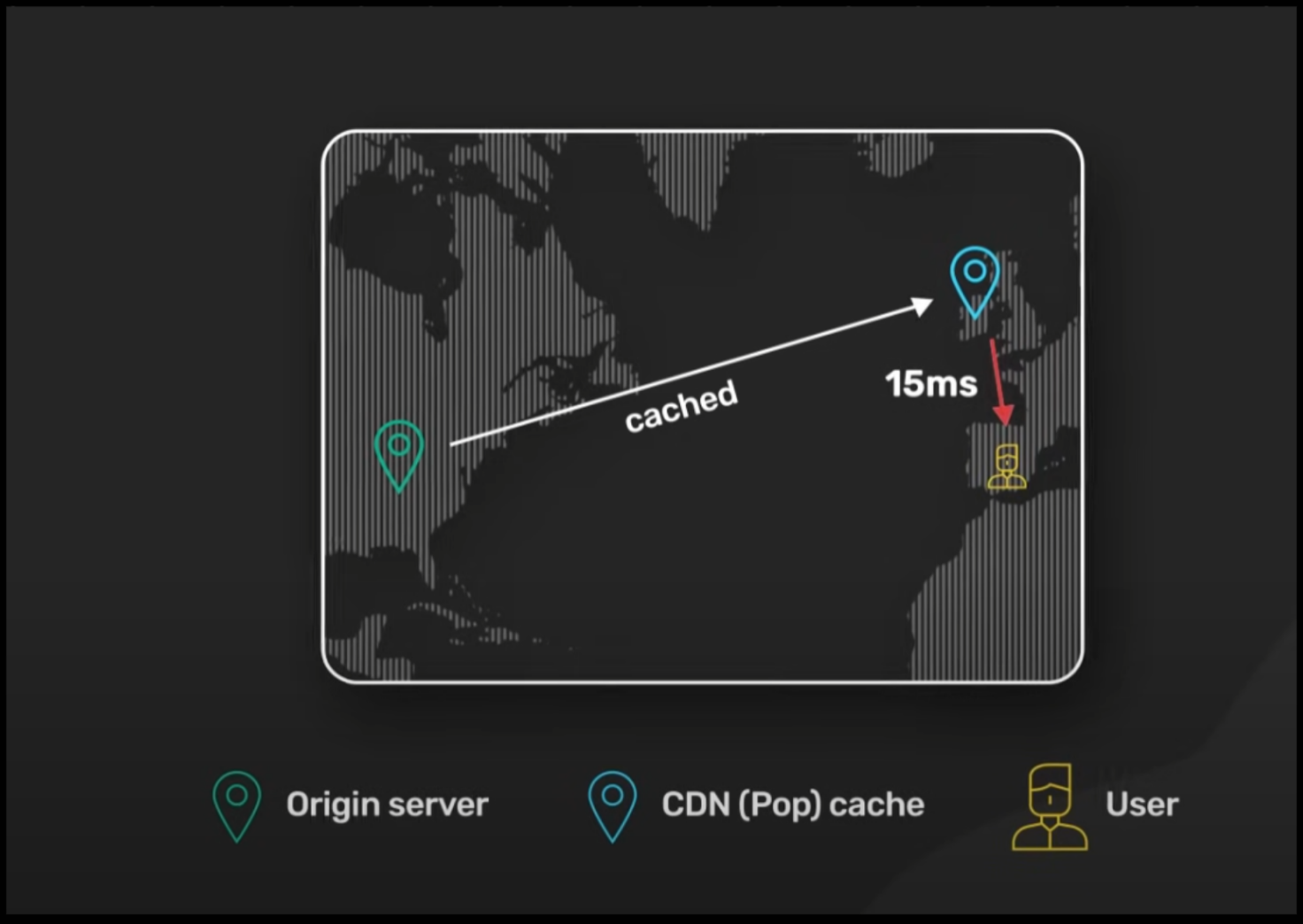
How does it work?

* Size of Images, Videos are too large, 5MB, 500MB, so CDN plays a crucial role here
* CDN can also Cache API responses.
* Best example is Giphy

Server billion of Gifs

* CDN server also known as Edge server.
* Every CDN server has its own Cache

What happens when we enter CDN url in browser tab?



* Url ⟶ DNS ⟶ IP address

* CDN use CNAME records

cdn.example.com ⟶ cdn1234.globalcdn.net

* CDN DNS service checks user IP address (geolocation)
↳ select nearest edge server and return IP address of that server.

* Other steps are similar to how https req. is processed

Two ways to route to nearst IP address

DNS based Load balancing

Unicast routing

1. DNS-Based Load Balancing

When a user enters a URL like `https://cdn.example.com`, the resolution to the nearest IP address happens as follows:

1.1 CDN's DNS CNAME Record

- The CDN URL (e.g., `cdn.example.com`) is typically a **CNAME record** that points to a CDN-managed domain (e.g., `cdn.provider.net`).
- The CDN's authoritative DNS server is configured to resolve the request to the **best edge server** based on the user's:
  - **Geographic location** (proximity to the nearest edge server).
  - **Network latency** or congestion.
  - **Availability of resources** on specific edge servers.

1.2 Geolocation of User

- When the DNS resolver sends a query to the CDN's authoritative DNS, it includes the **source IP address** of the resolver. This IP is used to infer the user's geographic location.
- Based on the location, the DNS server responds with the IP address of the nearest edge server.

1.3 Example

- If a user in New York queries `cdn.example.com`, the CDN DNS might resolve it to `192.0.2.1` (an edge server in New York).
- A user in London querying the same URL might resolve it to `203.0.113.1` (an edge server in London).

2. Anycast Routing

CDNs often use **Anycast IP addresses**, where the same IP address is advertised from multiple servers globally. The routing decision is made by the network infrastructure (routers) using **BGP (Border Gateway Protocol)**.

2.1 How Anycast Works

- Multiple edge servers in different locations announce the same IP address to the global internet.
- When a user sends a request to the Anycast IP, the request is routed to the **nearest server** based on:
  - **Network hops**: Fewer hops mean lower latency.
  - **Routing policies**: BGP routing policies prioritize certain paths.
  - **ISP interconnects**: Traffic might prefer a server in the same ISP or with a high-speed connection to the user's ISP.

2.2 Example

- Anycast IP: `192.0.2.1` is announced from:
  - New York edge server.
  - London edge server.
  - Tokyo edge server.
- A user in Tokyo hitting `192.0.2.1` will connect to the Tokyo edge server, as BGP routing ensures it's the shortest and most efficient path.