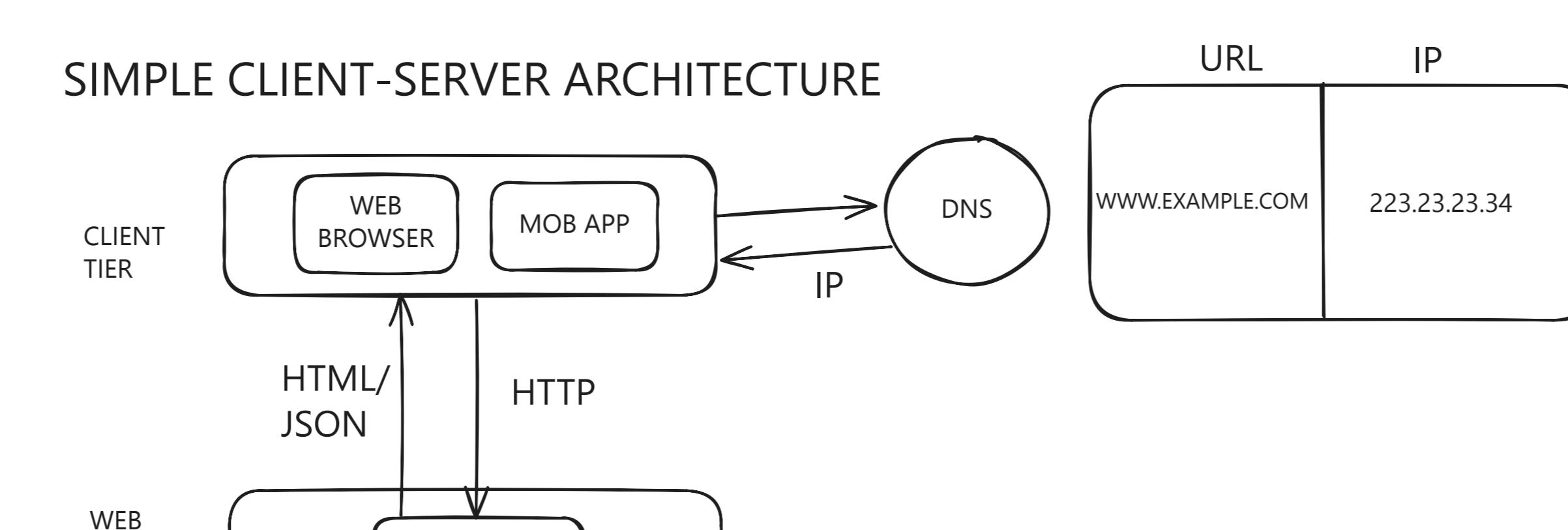
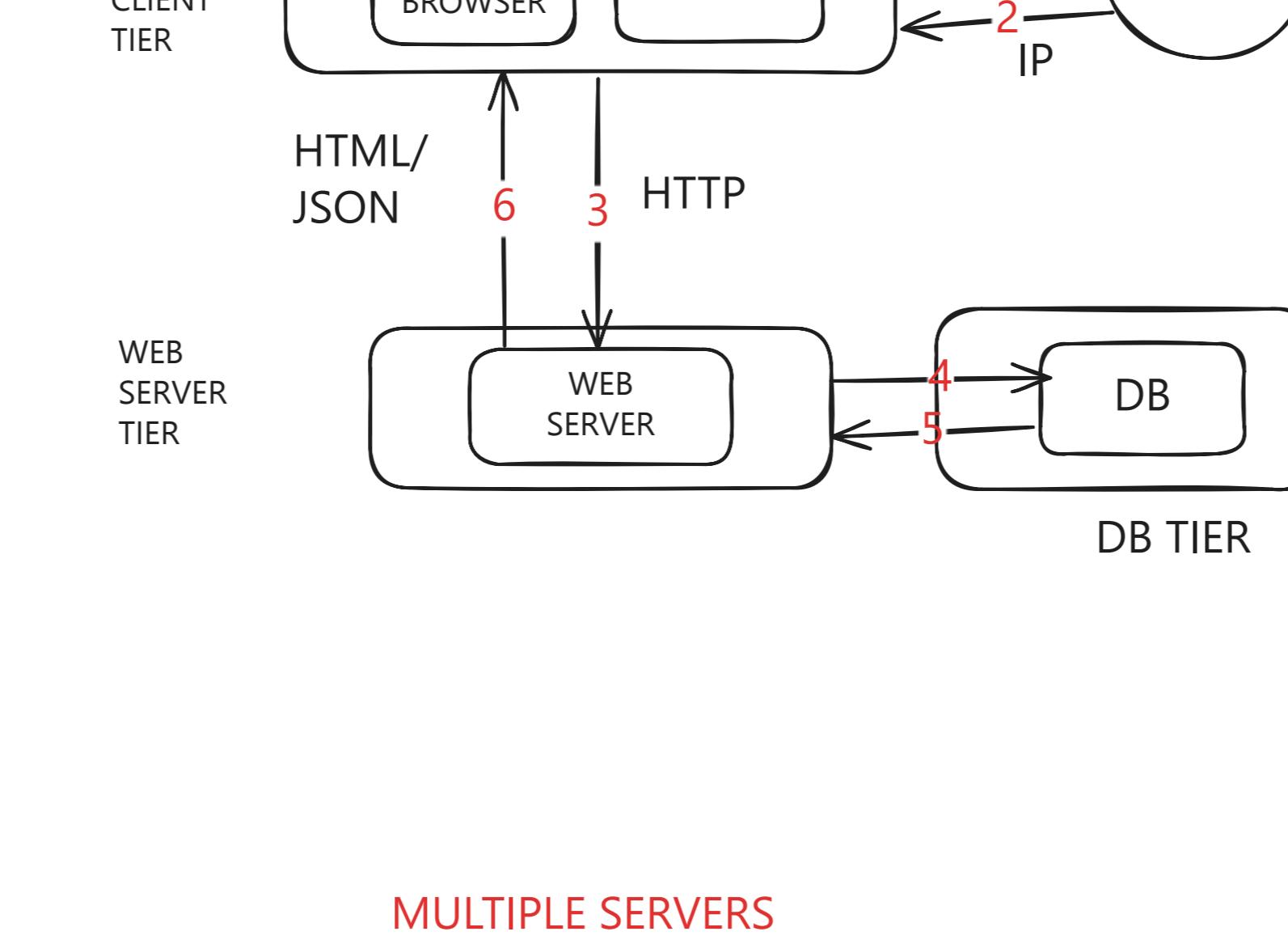


ANYTHING THAT WE WILL DO IN HLD IS TO MAKE OUR APPLICATION SCALE BETTER



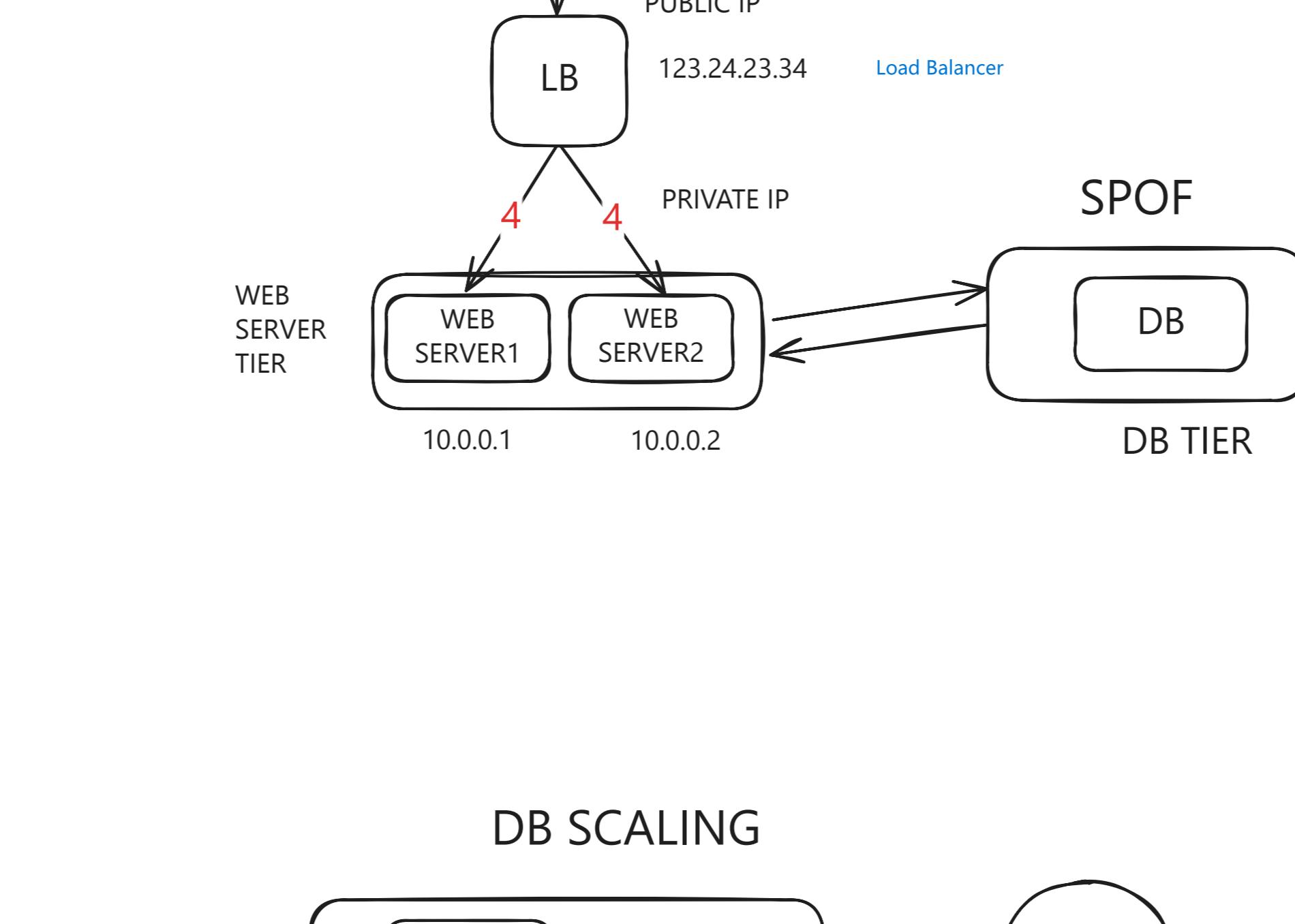
HTTP :
POST, GET, PUT ETC
URL : WWW.EXAMPLE.COM
HEADERS:
BODY : {
"USERID" : 1,
"NAME" : "ADITYA";
}



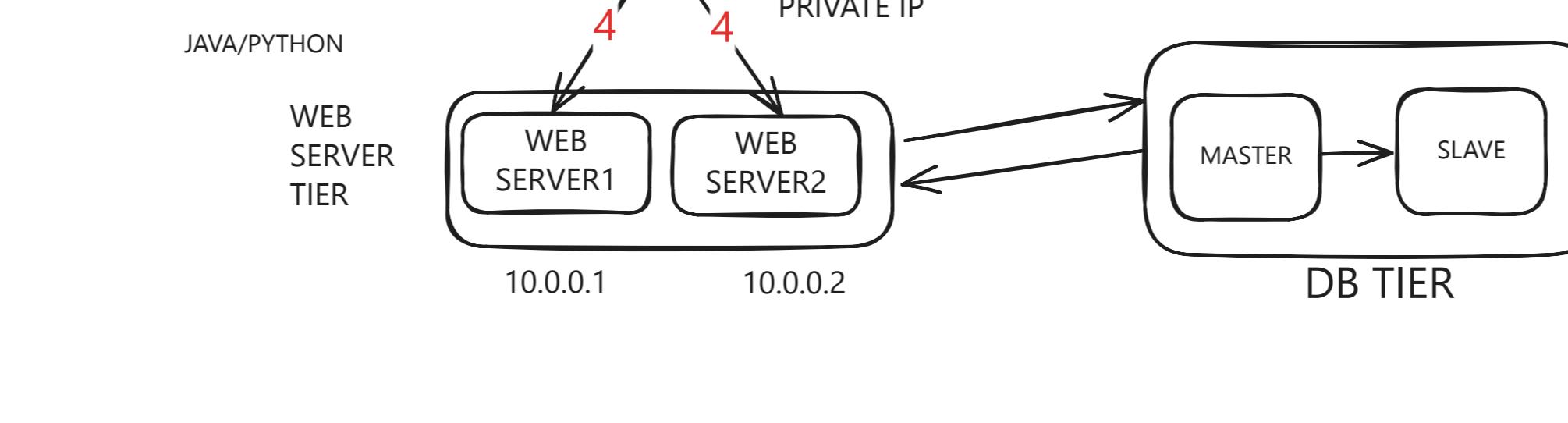
WHICH DB TO USE?
RELATIONAL : SQL BASED : MYSQL, POSTGRESQL, ORACLE SQL
NON - RELATIONAL

KEY - VALUE STORE : AMAZON DYNAMO DB
COLUMN BASED STORE : CASANDRA DB
DOCUMENT BASED STORE : MONGO DB
GRAPH STORE : INSTAGRAM, FB, TWITTER ETC

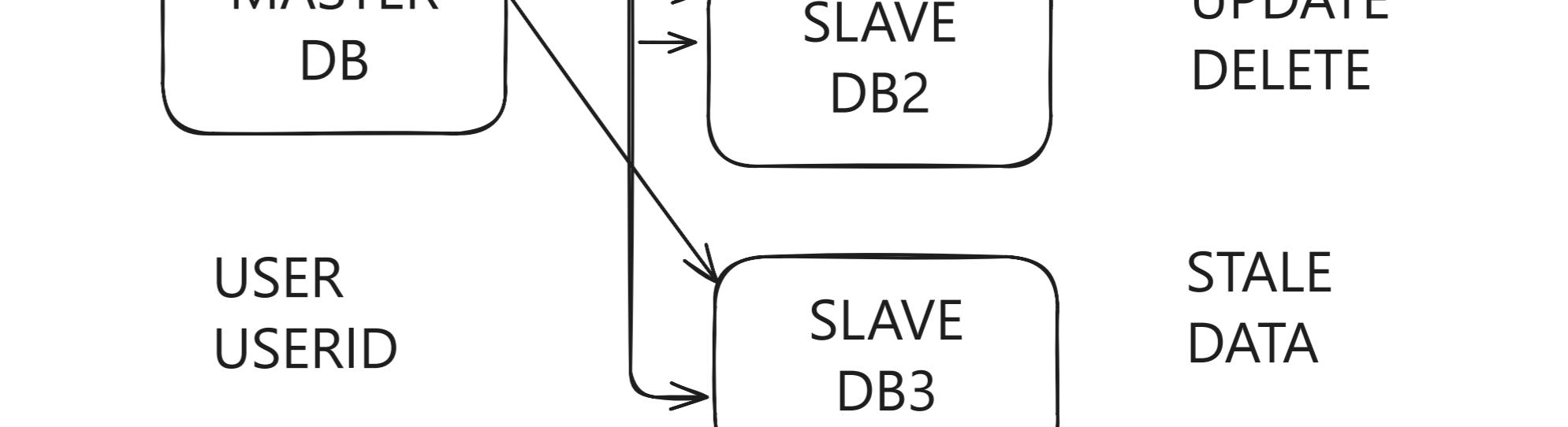
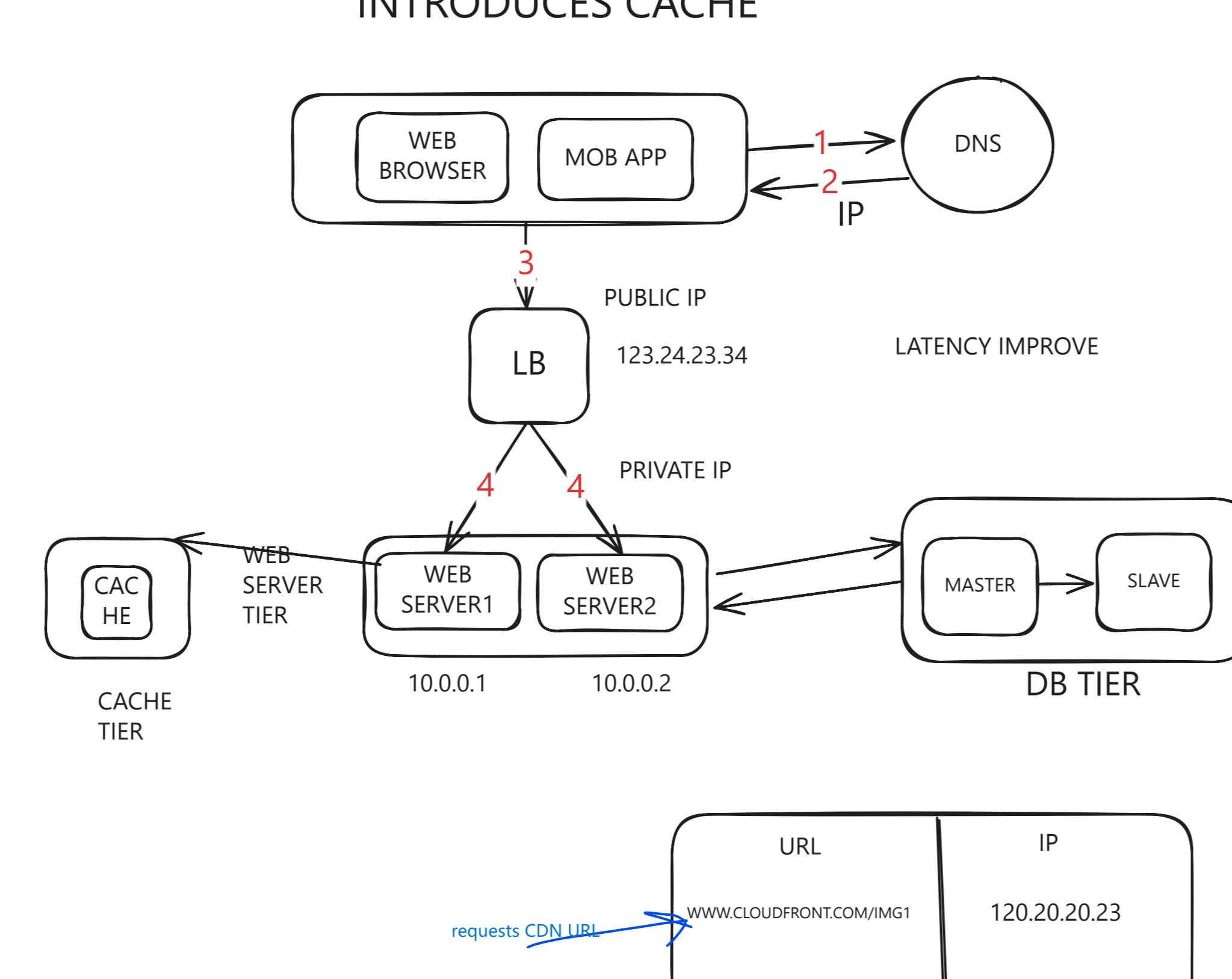
EVERY DATA CANT BE STRUCTURED



This time the server responds the IP address of the load balancer and not the server IP



INTRODUCES CACHE



READ : SELECT

WRITE

INSERT

UPDATE

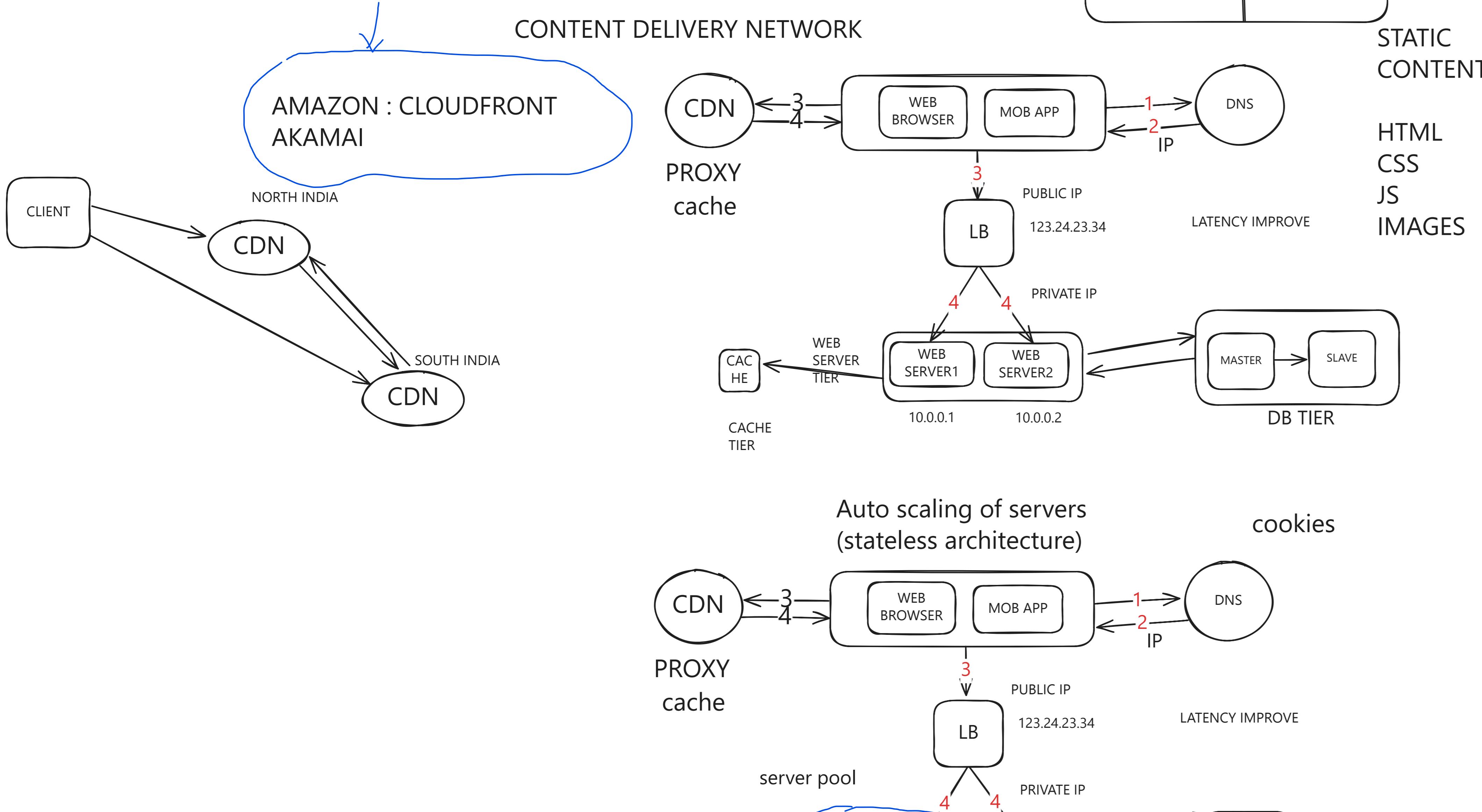
DELETE

STALE

DATA

CACHE MISS

CACHE EVICTION ALGORITHM



STATIC CONTENT
HTML
CSS
JS
IMAGES

Some of the CDN

AMAZON : CLOUDFRONT

AKAMAI

NORTH INDIA

SOUTH INDIA

Auto scaling of servers (stateless architecture)

cookies

if traffic increases the 3rd server gets active (scaling up) and vice v
but even if all the servers in the pool are active and cannot handle incoming requests then the requests will be dropped
if server is not active then there is no costing for that server

server pool

shared DB

DB TIER

for international users the latency is very high because data center is in India

sticky sessions : stateful architecture

HTTP : stateless

stateless architecture

Multiple Data centres

GEO Enabled DNS

DC1

India

DC2

USA

shared DB

Kafka, RabbitMQ (Asynchronous)

MESSAGING QUEUE

PROXY cache

CDN

HTTP : stateless

workers

consumers

web server

producer

DeQueue

consumer

PUB-SUB (publisher - subscriber model)

SHARDING (DB partitioning)

Horizontal scaling

Data De-Normalization

MOD % 4

erwe

aditya das

abhay sdas

DB0

DB1

DB2

DB3

erwe

aditya das

abhay sdas

DB0

DB1

DB2

DB3

erwe

aditya das

abhay sdas

DB0

DB1

DB2

DB3