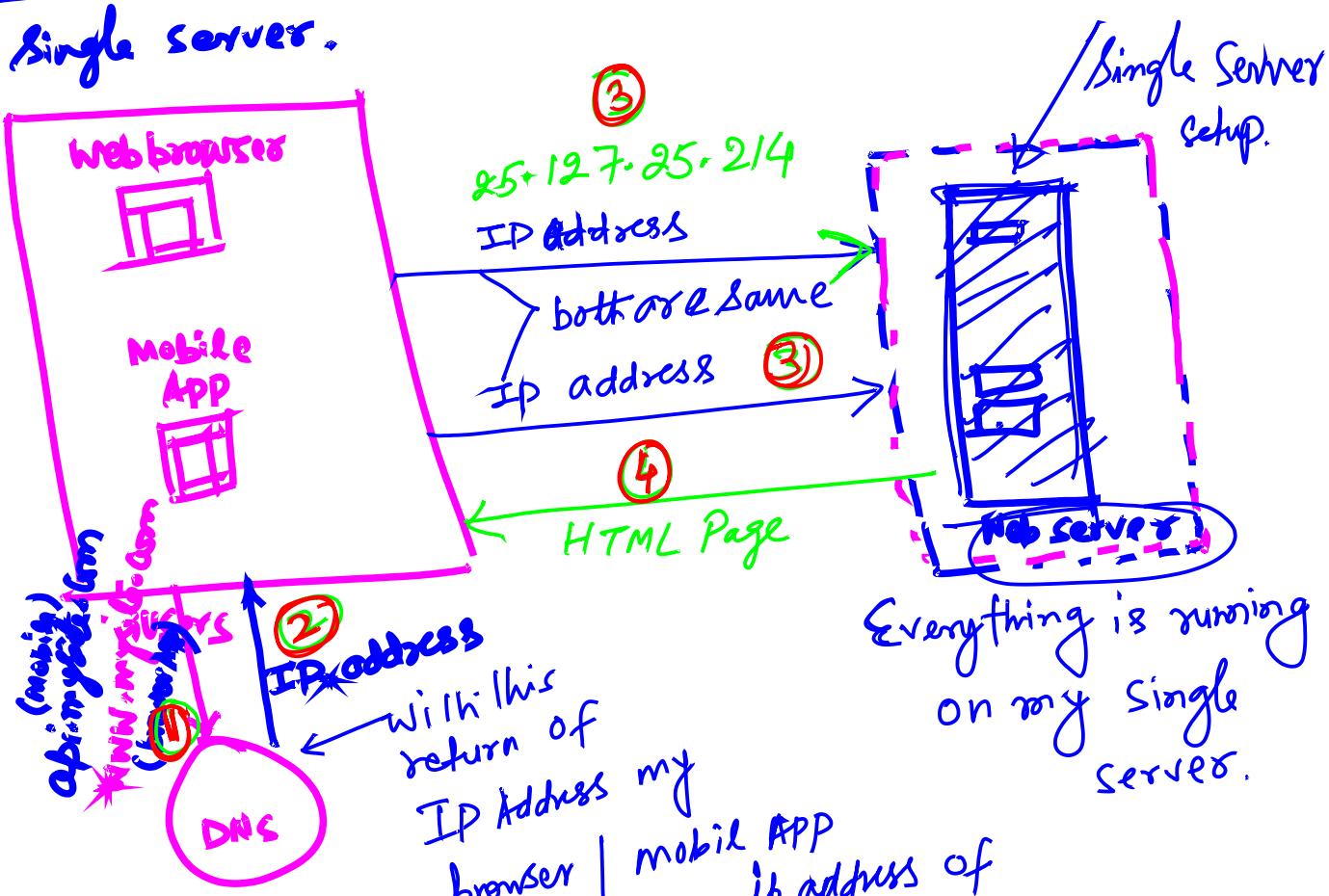
18/Feb 7:30PM

Scale from Zero to million of users

Scale is always a challenging game.

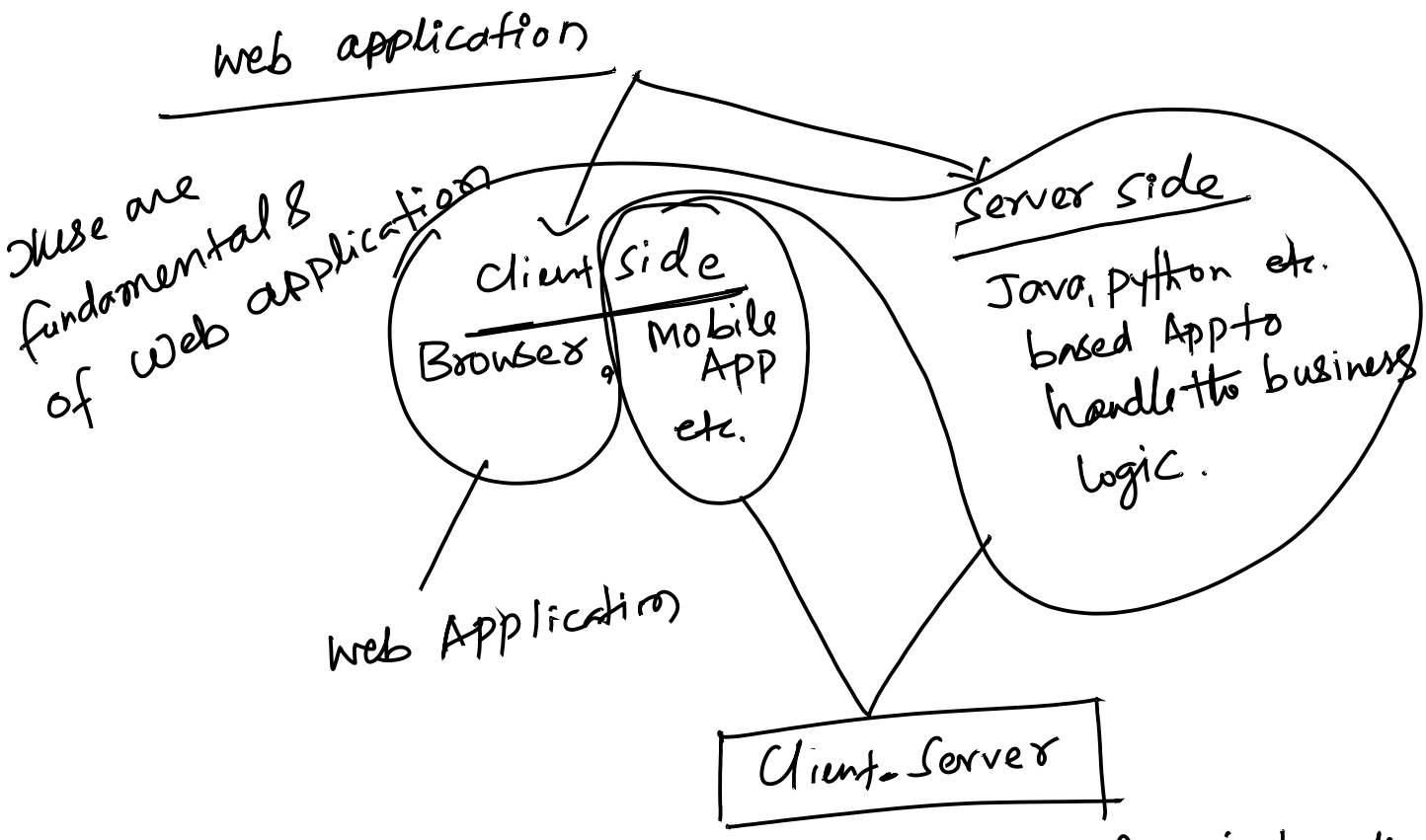
Single server setup, we should always start from single server.



Steps

1. Get the IP address
2. received Ip address from DNS
3. Call the service by IP address
4. receive HTML Page from webServer

Let us examine the traffic source.
In the above diagram, we have two traffic source mainly 1. Browser (Client 1), 2. mobile App (Client 2.



HTTP Protocol generally used for interaction between client & server. JSON object is used commonly for request and response in HTTP protocol.

Example of REST API & Response

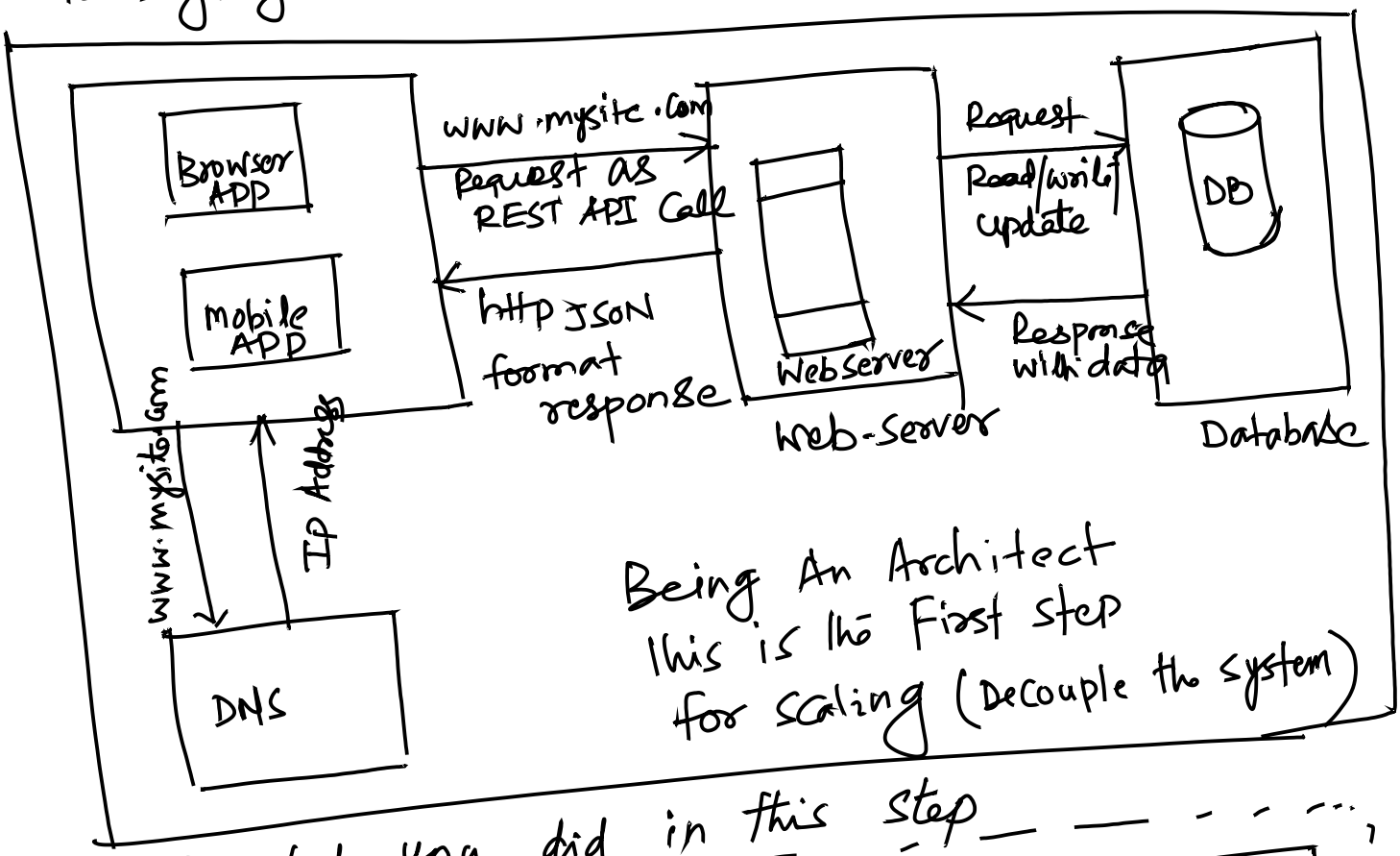
An Example of JSON response format

→ GET/User/12 → Retrieve user object for id = 12

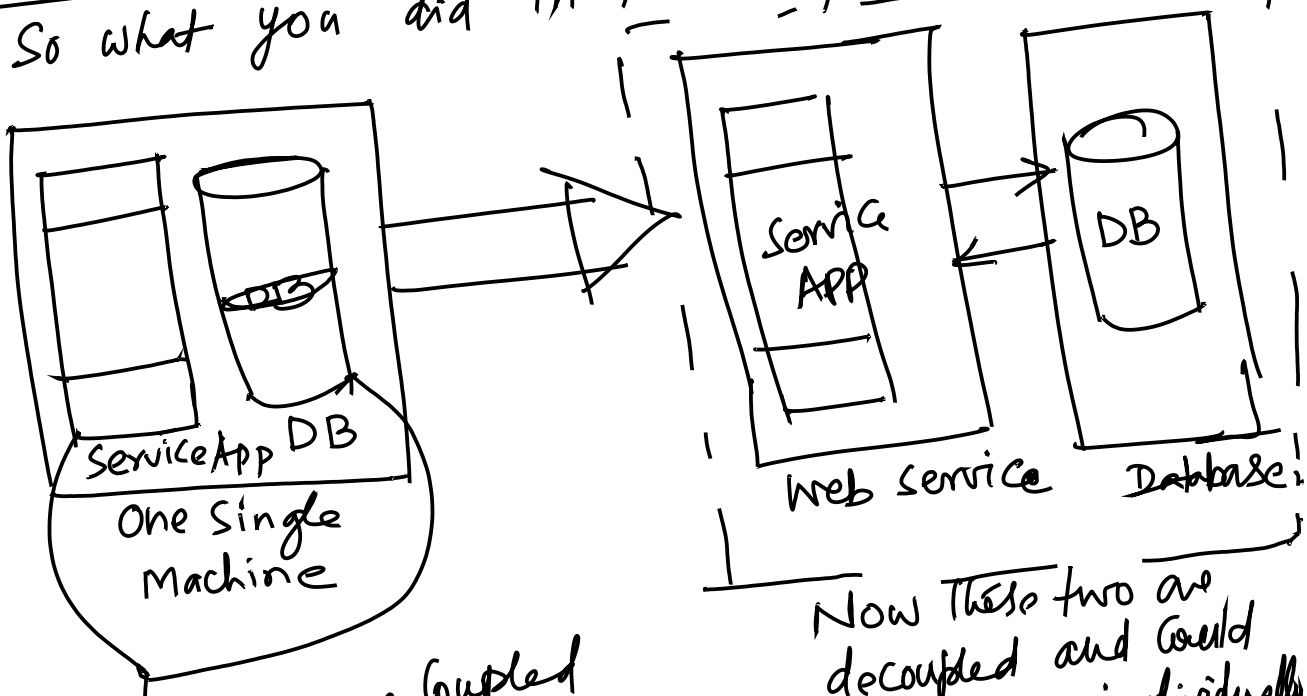
```
{
  "id" : 12,
  "firstName" : "Abinash",
  "lastName" : "Mishra",
  "address" : {
    "streetAddress" : "21 2nd Street",
    "city" : "New York",
    "state" : "NY",
    "postalCode" : 10021
  },
  "phoneNumbers" : [
    "9971003***",
    "8075***215"
  ]
} // end of response
```

"Key-value Pair" Contd...

Now, being an archect, the first highest priority task is to segregate web-server and database.



So what you did in this step



These two are Coupled in one machine so scaling is definately problematic also system is tightly coupled, for ex: if DB Servic crashed it might crash your complete system.

Now These two are decoupled and could be scaled individually without any problem.

Being an Architect you can observe that you decoupled the database, but next challenge for you is to how do you scale the database? "OR", what exactly is the challenge you can see in "database".

And even before, you might ask yourself, which type of database do you think you need here. And as you start thinking about database, you might get question like, what type of data your system require to save.