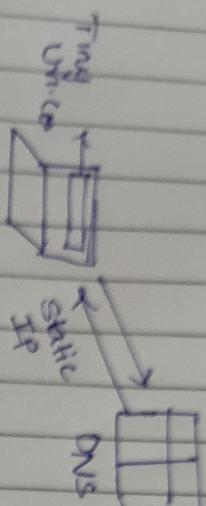


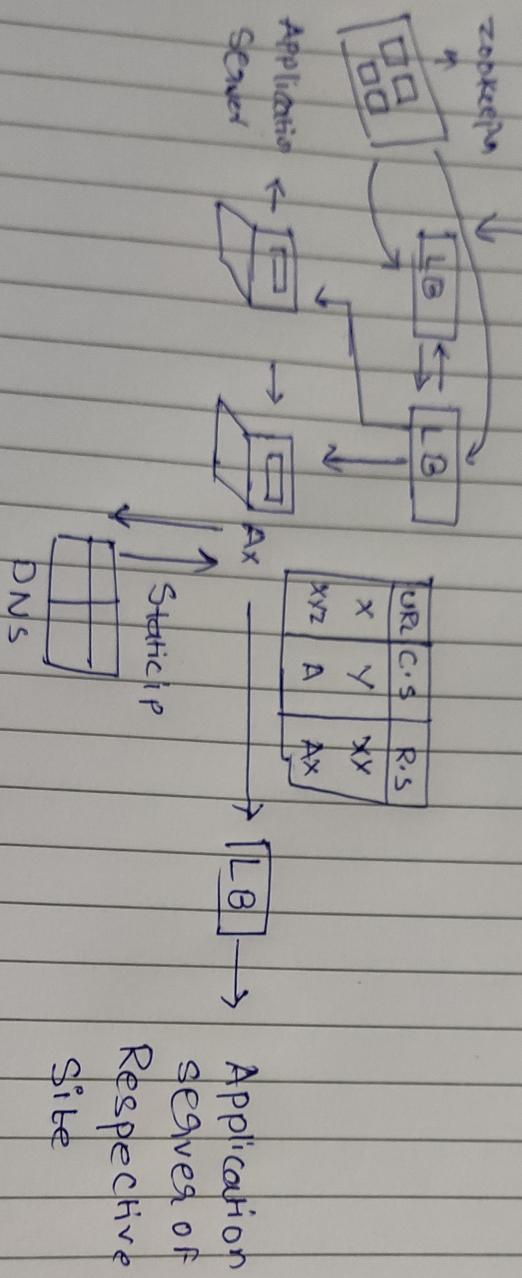
## TINY URL ARCHITECTURE

ICANN



In order to reduce  
duplicate entry make IP URL as unique Id

UR C.S R.S  
X Y XX  
XYZ A AX



→ How does your system ensure that 2 URL's never map to the same shortened URL?

Ans:- We can use hash function to Actual URL as Input and convert it into new shortened URL

→ How will you ensure the system has very low latency?

Ans:- by Scaling Application horizontally depending upon incoming traffic and using consistent hashing technique in load balancers in order to distribute load to application servers and as well as for data sharding.

→ What will happen if the machine storing the URL mapping dies?

Ans:- We need to have a hot copy sync with actual DB as backup and to avoid Data Center failure spread DB across different regions

→ How do you make sure your system is consistent?  
This is to say, if I have shortened a URL given  
the shortened URL, my system should always  
return the original URL no matter when I call  
your system?

Ans:- I will be using hash function to  
generate shortened URL of actual URL  
so the consistency should be maintained  
by reducing the factor of hash collision

→ How do you make sure that your service never  
goes down? (No outage). You have to assume that  
your machine will die. How do you make sure  
your service is unaffected by those incidents  
(unless all of your machines die at once -  
lets assume that never happens)

Ans:- I will make sure have backup to  
reduce single point of failure by using  
horizontal scaling.

I will be checking health check  
the instance which are working. In case  
if instance goes will add more system to  
share load and keep the things go on.