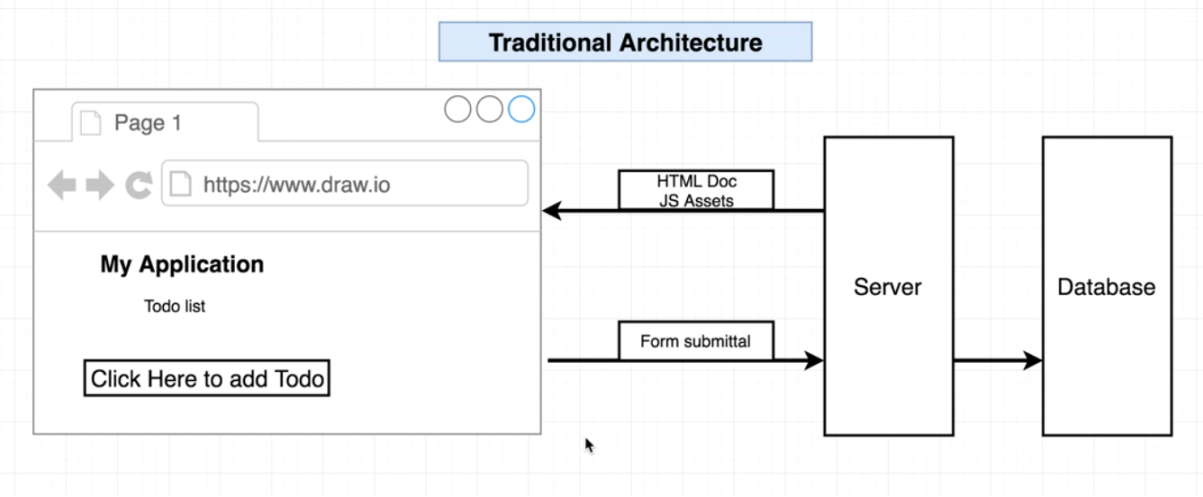
Building Interactive Frontends

- We will build our frontend in React.

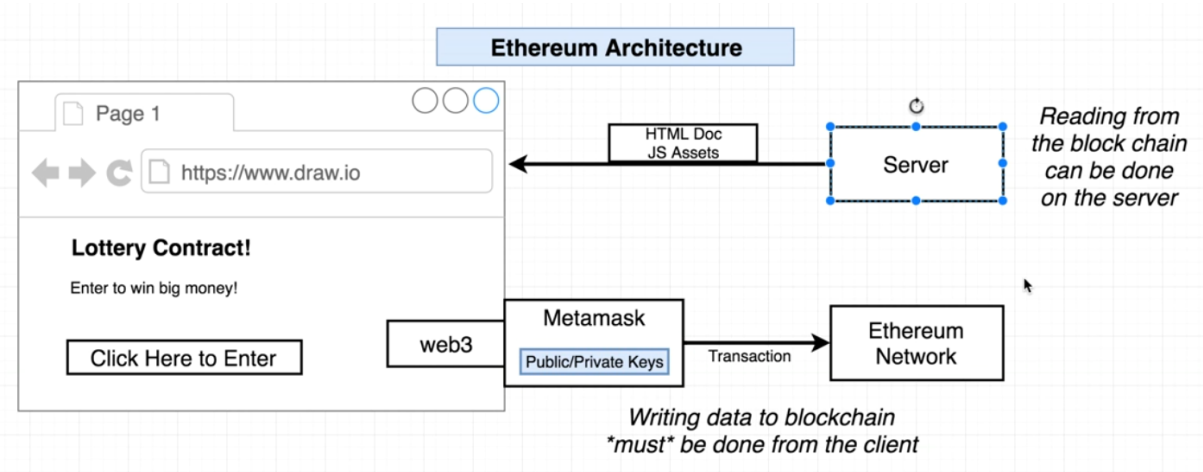
Traditional Architecture -



Achritecture of a full-stack web3 app -

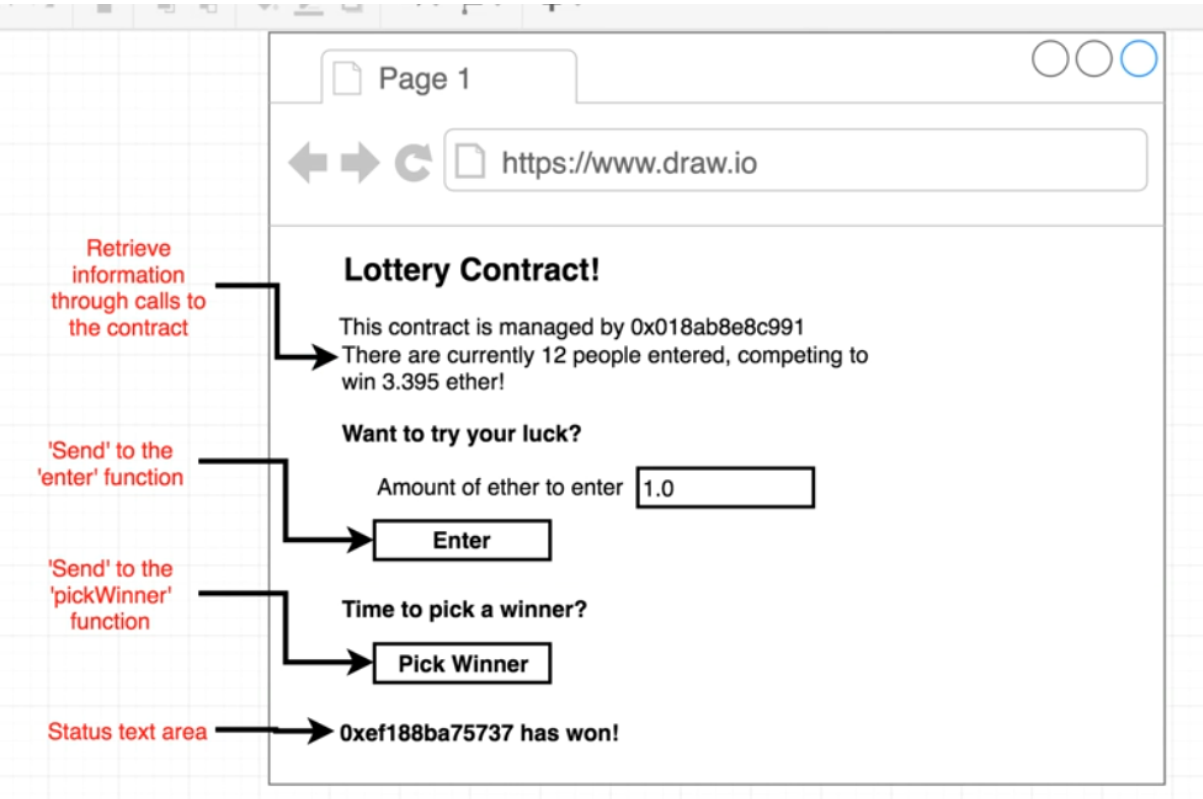
- **Very Important -**
  - Whenever we want to change some data on the blockchain, we do that ONLY on the frontend.
  - Because that transaction has to be signed by the user's public/private keys which we CANNOT in any case send to backend/store on DB.
- Server pe bhi transaction issue karenge but that will be signed using our public/private keys not the user's.

- We can observe that now our frontend has to do a lot of work, so, we need our frontend to be intelligent, thats where **React** comes into the picture.



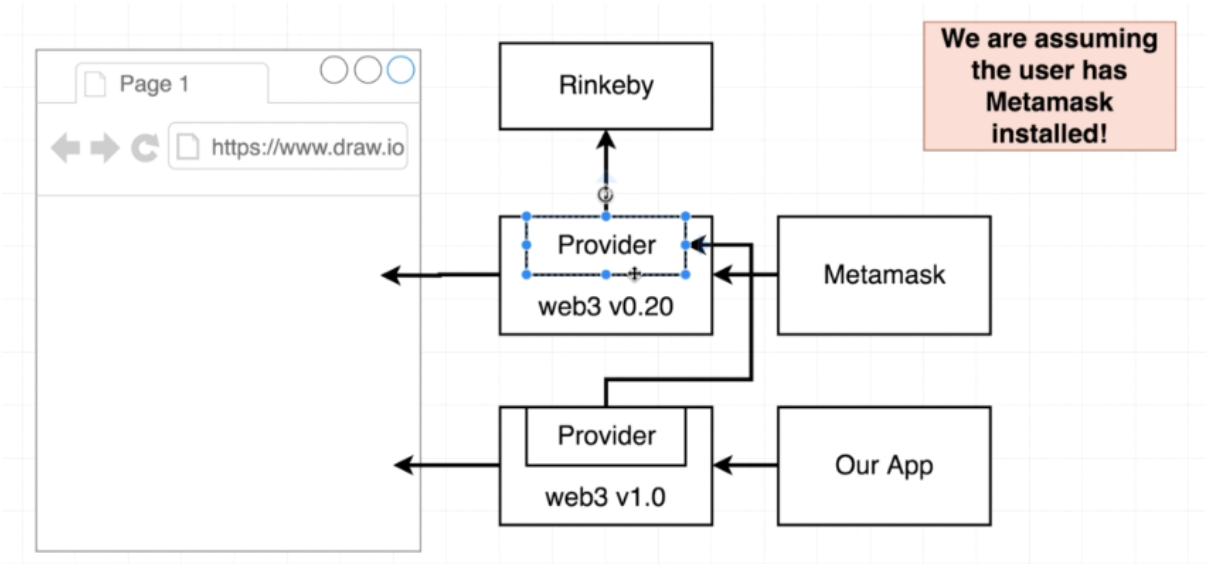
App we are going to build -

- We will use Goerli test network to deploy our contract.

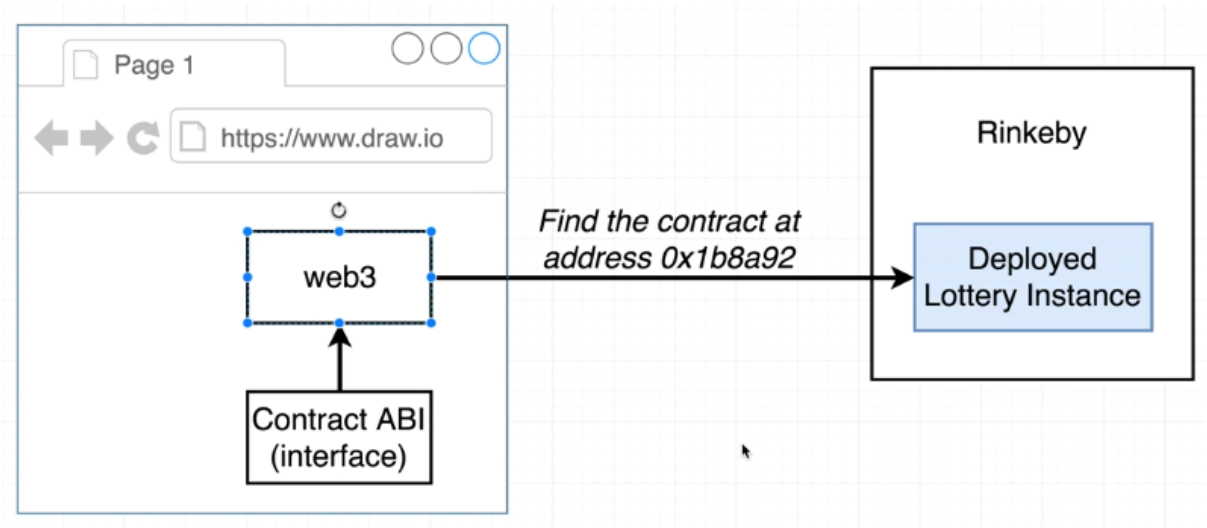


Setting up web3 package -

- Metamask injects a version of web3 into the webpage.
- We want to use the provider of that metamask injected ver of web3 into our ver of web3.



- We are going to feed ABI to web3 instance so that we can access the contract's methods. Also, we will tell it where our contract has been deployed on test net so that it can access the deployed instance.



- Now we create a local copy of the contract instance that is meant to **represent** what's happening on the blockchain.

- This local copy is not the deployed instance of the contract its just a representation of it.
- Basically its a JS object.
- To create this local copy, we need the contract ABI and address.

- **Process of working with the local contract instance using React -**