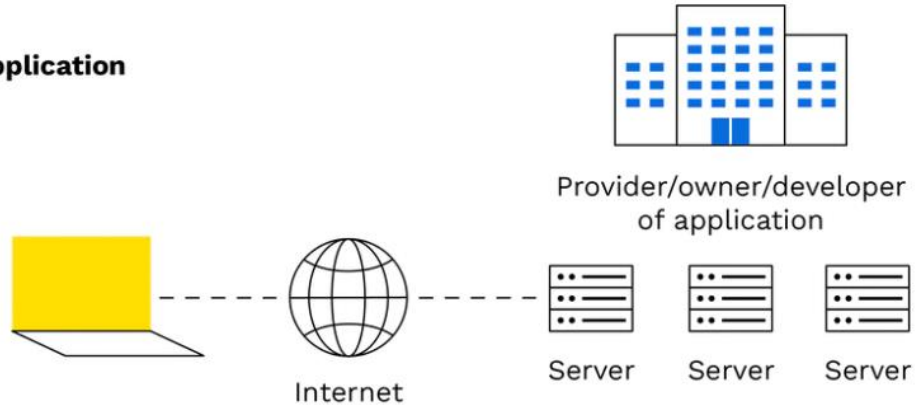
# Decentralized Applications Dapps

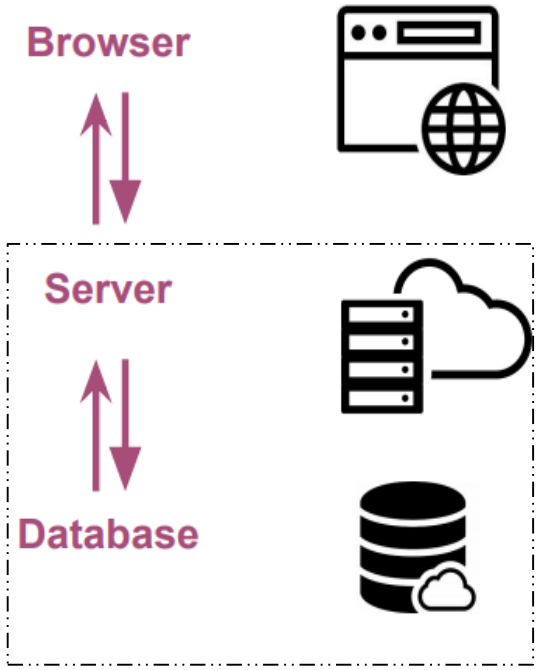
# Contents

1) Decentralized applications (DApp)
2) Building a DApp
3) Demo

# 1) Decentralized applications

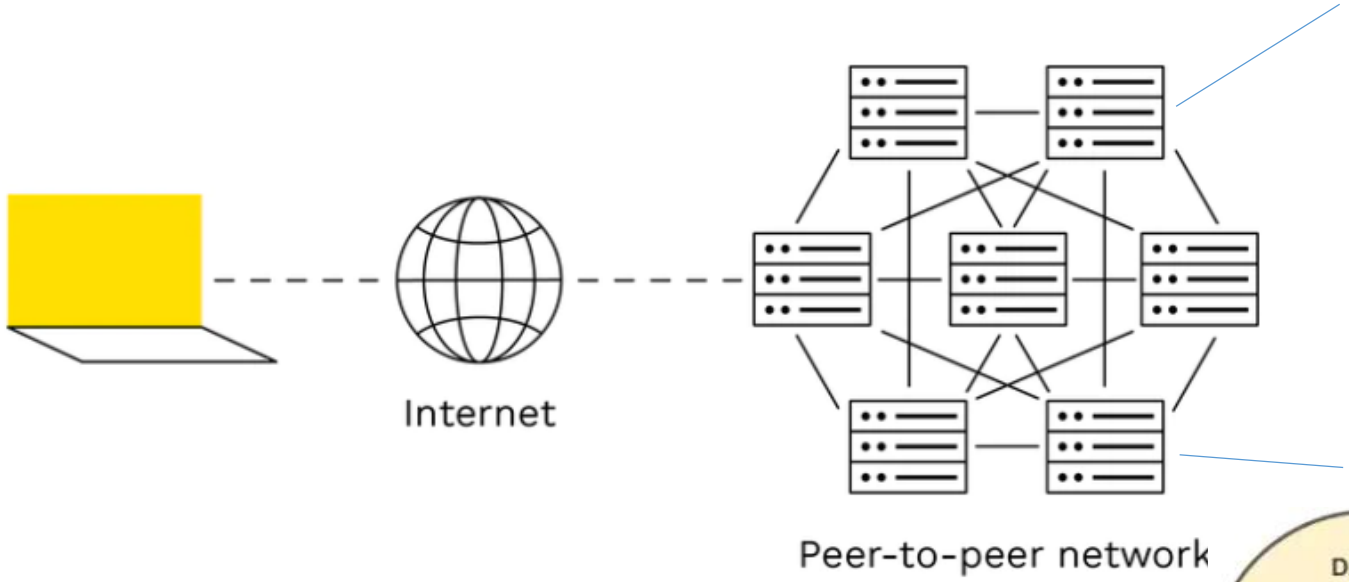
## Centralized applications



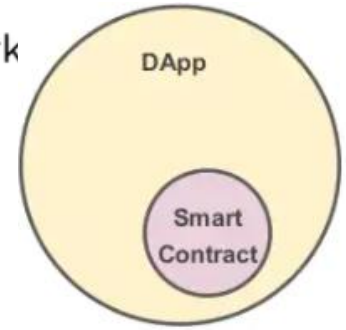



# 1) Decentralized applications

*Decentralized applications are digital applications or programs that operate on a decentralized network (such as blockchain)*




**DApp's components**



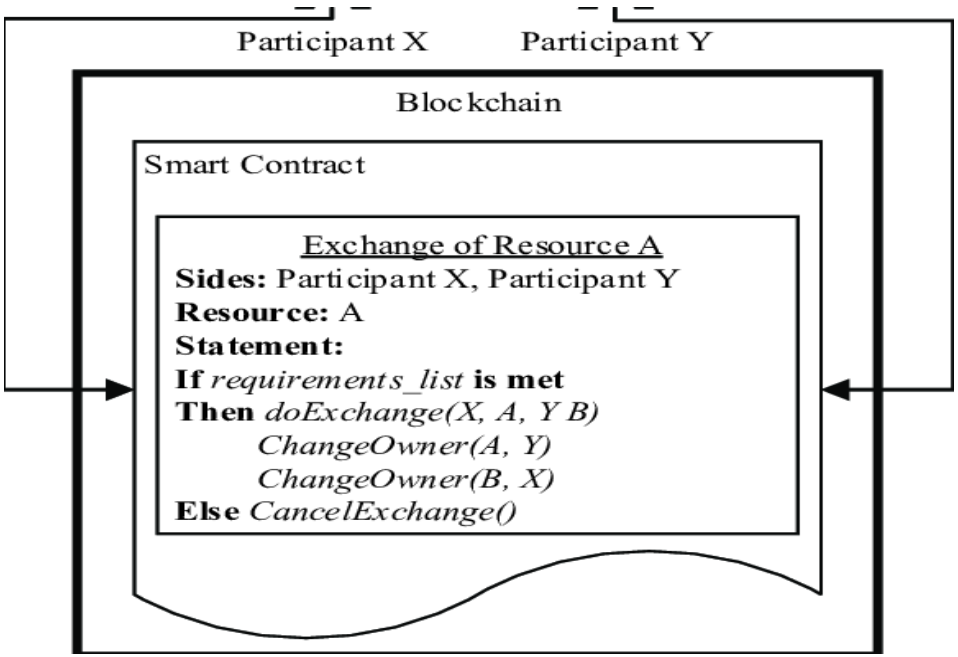

***(1) Backend:***

- smart contract (the business logic is programmed).
- The backend should be kept as light as possible because deploying a smart contract or executing a function costs gas

**Smart Contract**

- Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met.


- Smart contracts work by following simple "if/when…then…" statements that are written into code on a blockchain.



**(2) Frontend:**

- Programming languages like HTML, CSS, or JavaScript can be used to program the graphical user interface part.
- This front-end is usually programmed with JavaScript and uses libraries like web3.js or ethers.js.

**(3) Data storage**

Smart contracts are not well suited to storing and processing a large amount of data => the information needs to be stored off- chain (IPFS, SWARM)

# 2. Building a DApp

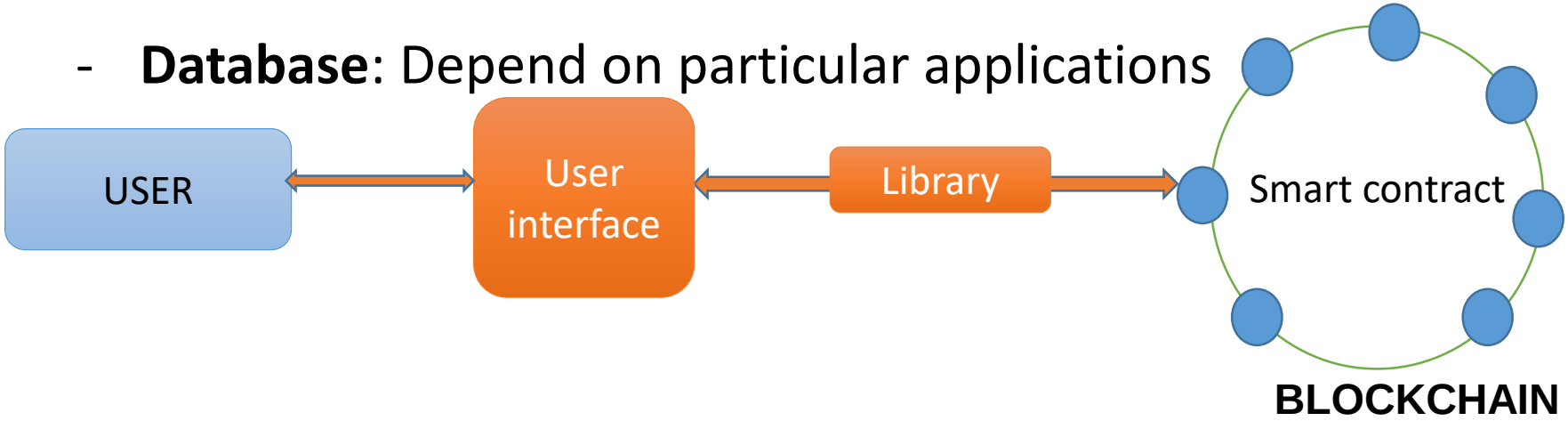
Main principles to develop a DApp

- **Front-end:** create app's user interface;


Add library to let front-end connect to the decentraliz

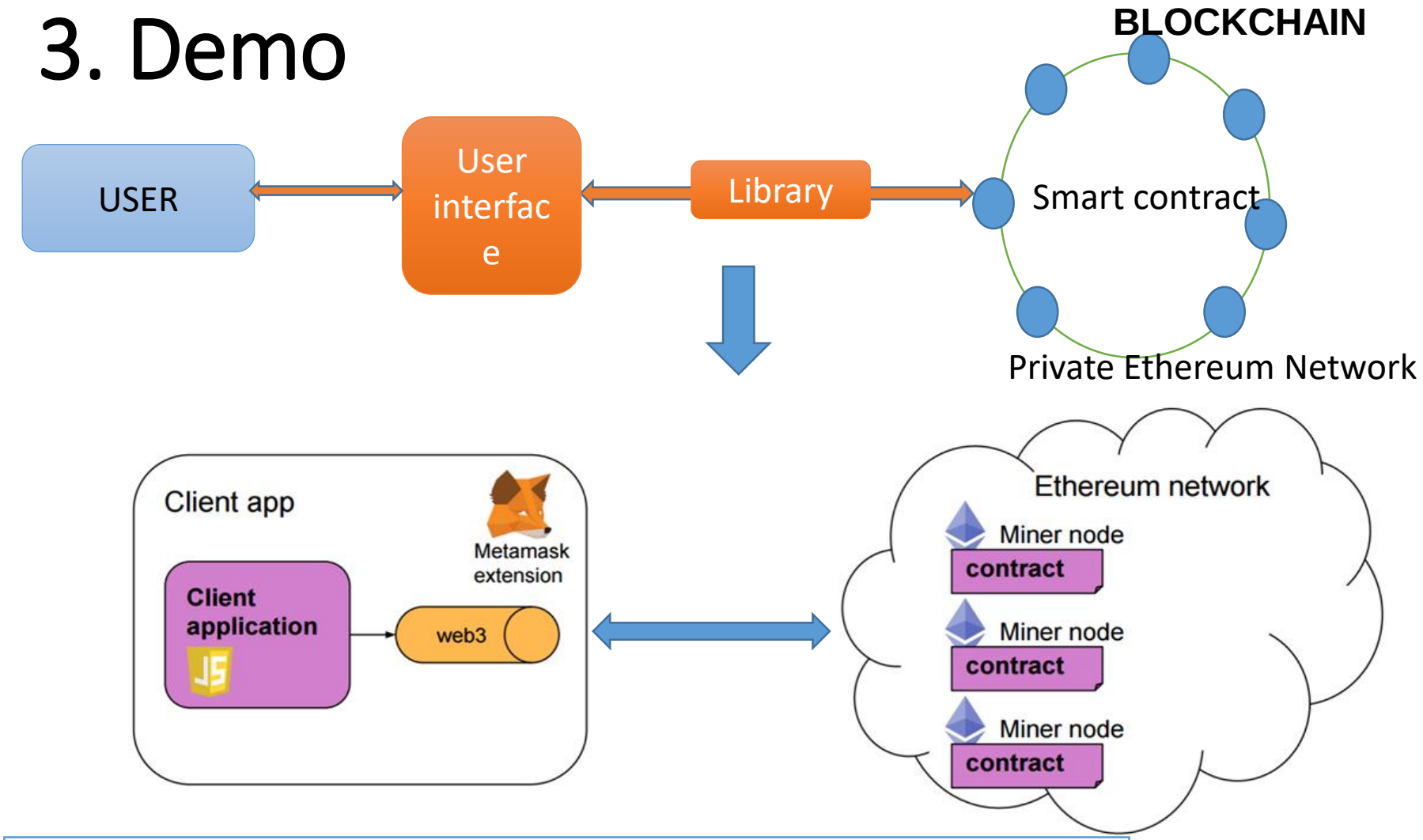
network


- **Back-end:** Smart contract
- **Database**: Depend on particular applications



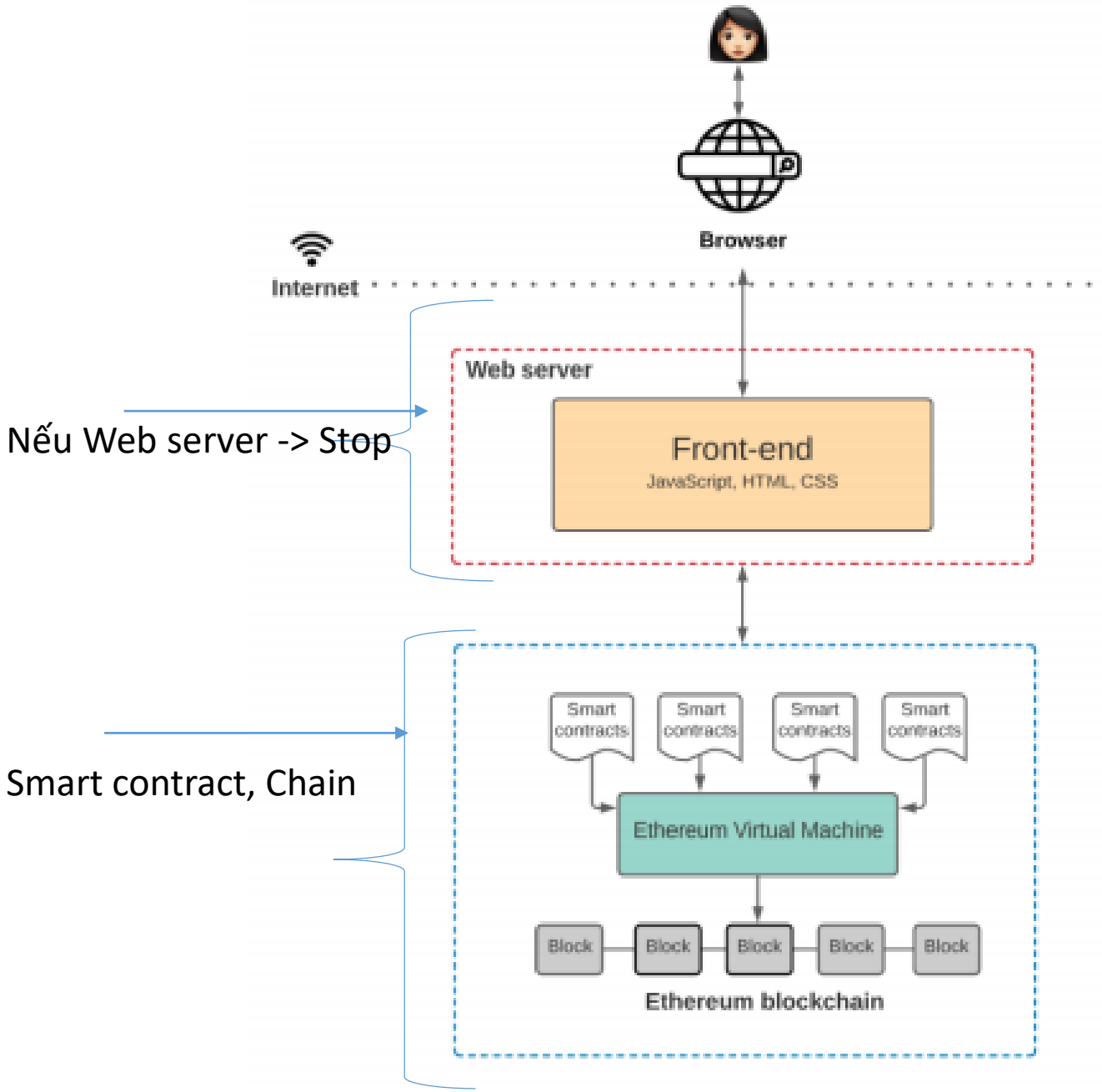

DApp Architecture

# 3. Demo




**Client Application**: User interface; containing API of **smart contract**

**Metamask** is a wallet ethereum.

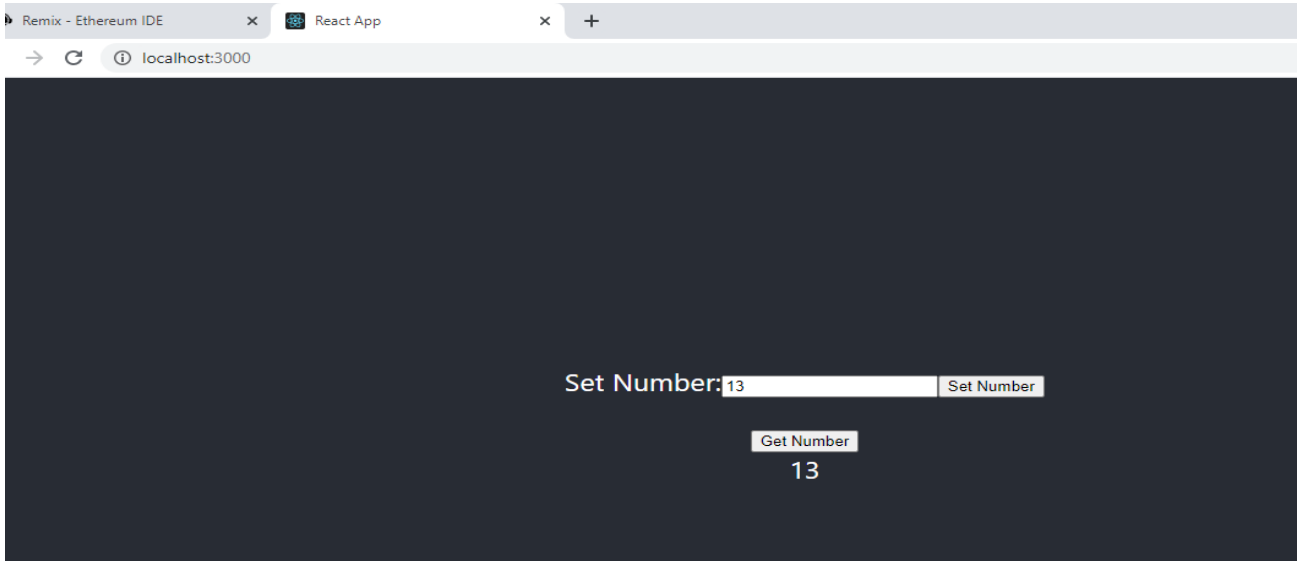


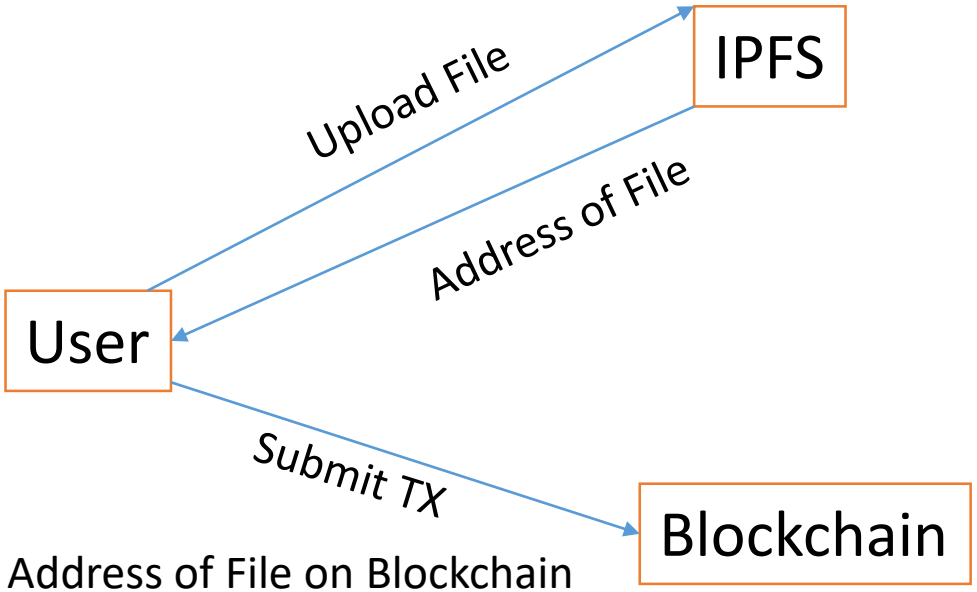

Nếu Web server -> Stop

Smart contract, Chain

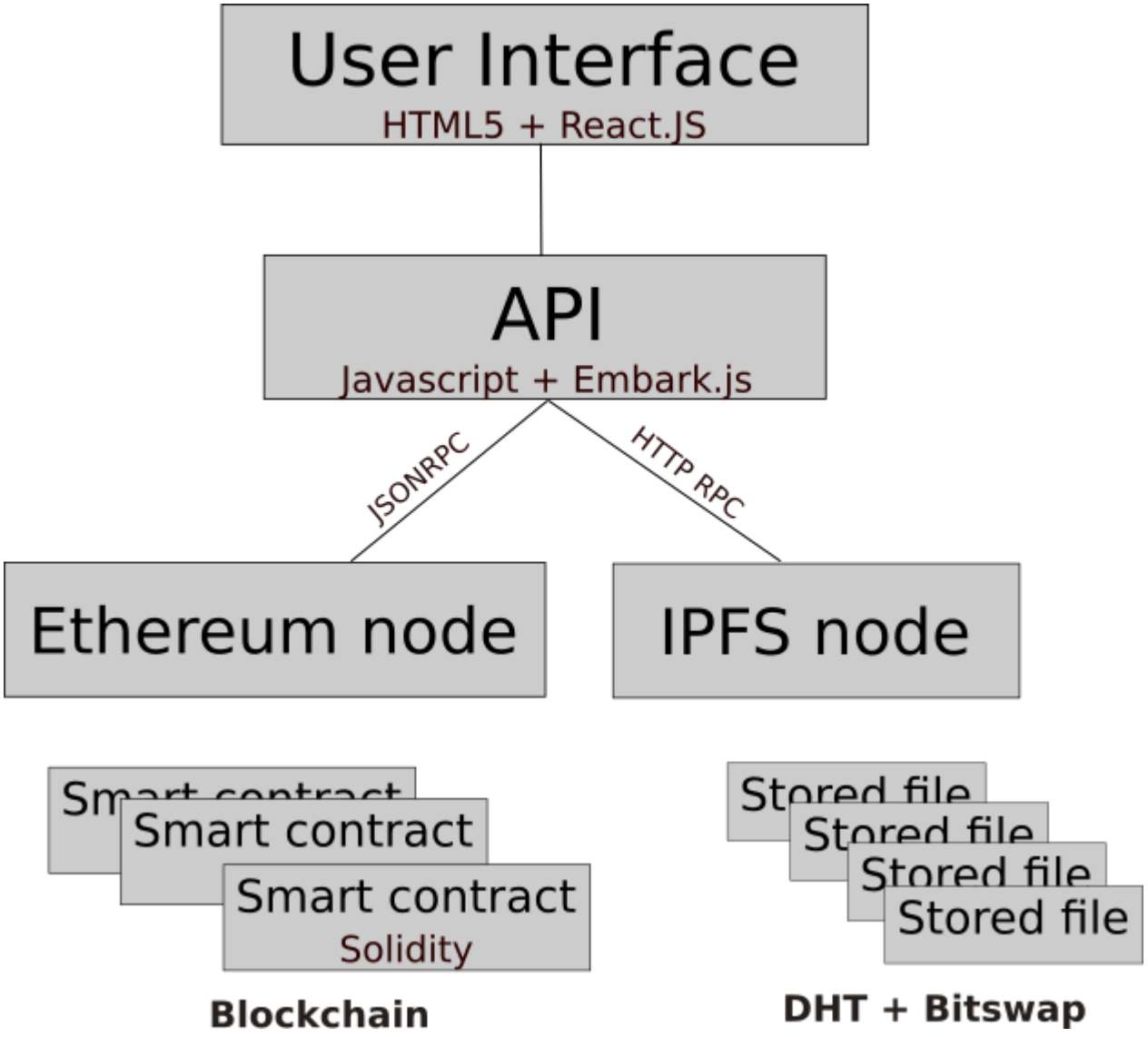
# 3. Demo

- Building a locally hosted instance of Ethereum (*Ganache*)
- Deploy a smart contract on ethereum (*Remix*)
- Deploy a DApp (*React*, *Web3*)
  - +*Set number*
  - +*Get number*






Store the Address of File on Blockchain


User Interface
HTML5 + React.JS
API
Javascript + Embark.js
JSONRPC
HTTP RPC
Ethereum node
IPFS node
Smart contract
Smart contract
Solidity
Stored file
Stored file
Stored file
Stored file
Blockchain
DHT + Bitswap


Thank You!