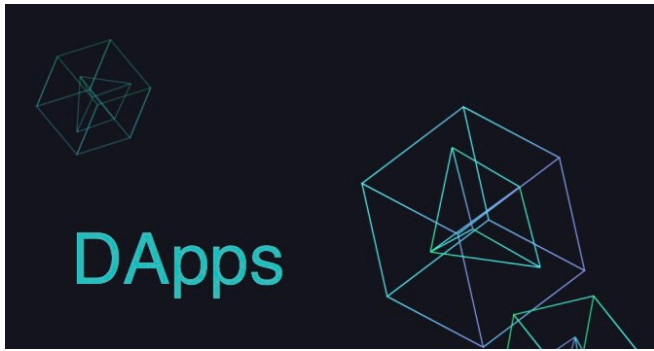




DApps

Module 3 (part 1)

Part 1

- Think of Ethereum like the internet and all the **DAPPS** as websites that run in it.
- DAPPS, they are all decentralized and not owned by an individual, they are owned by people.

What is dApp?

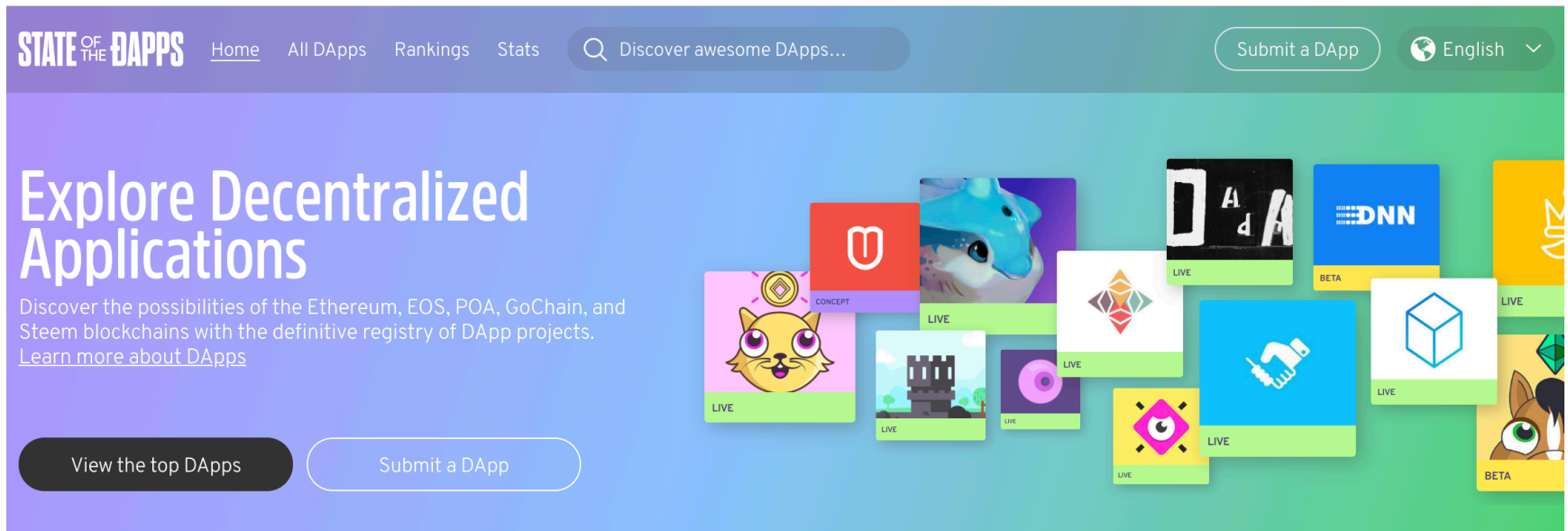
- dApps are a 'blockchain enabled' **website**, where the Smart Contract is what allows it to connect to the blockchain.

dApps

[Source from ConsenSys]



State of the DApp



<https://www.stateofthedapps.com/>

Traditional Web vs. Dapp

- The traditional web application uses HTML, CSS and Javascript to render a page. It will also need to grab details from a database utilizing an API. When you go onto Facebook, the page will call an API to grab your personal data and display them on the page.
- Traditional websites:

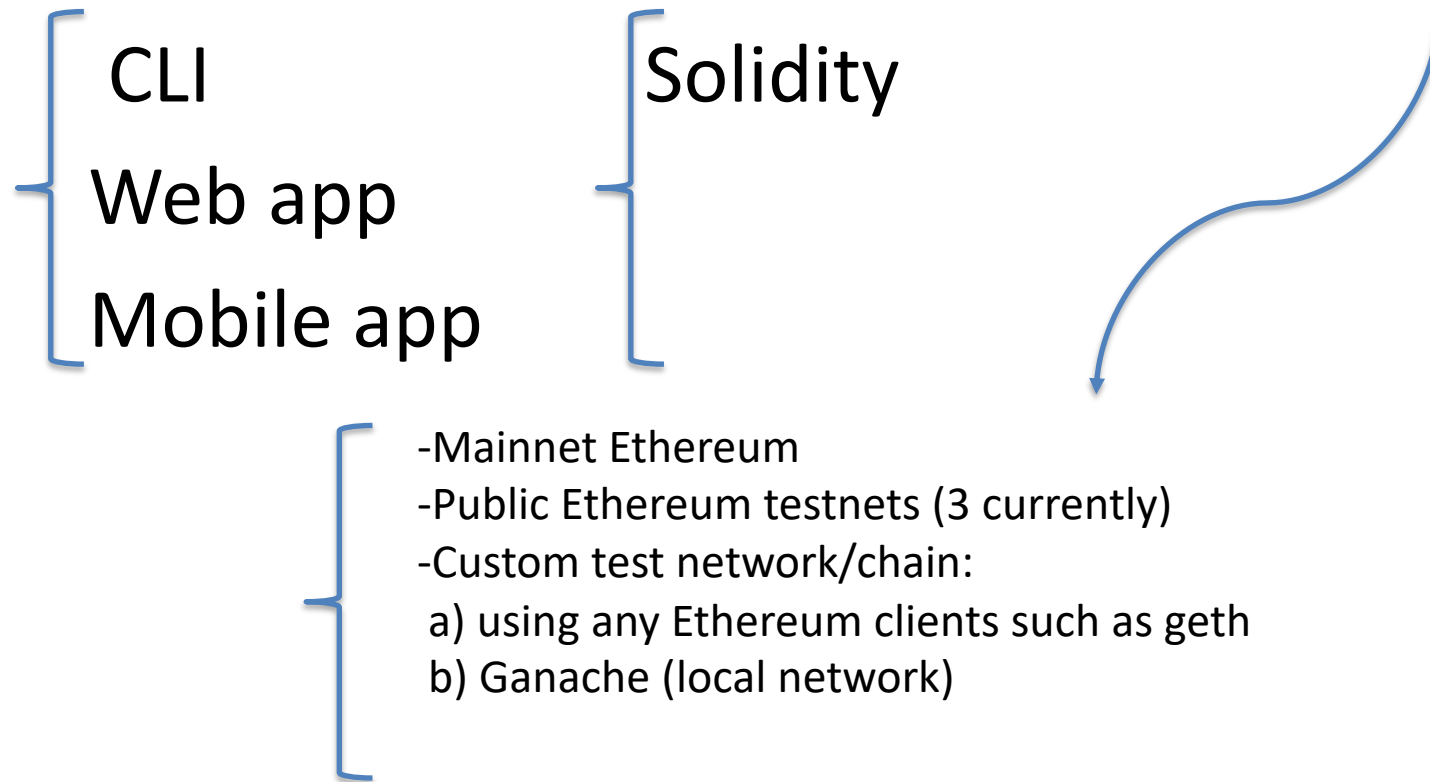
Front End → API → Database

- dApps are similar to a conventional web application. The front end uses the *exact same* technology to render the page. The one critical difference is that instead of an API connecting to a Database, you have a Smart Contract connecting to a blockchain.
- dApp enabled website:

Front End → Smart Contract → Blockchain

Conceptual view of Dapp stack

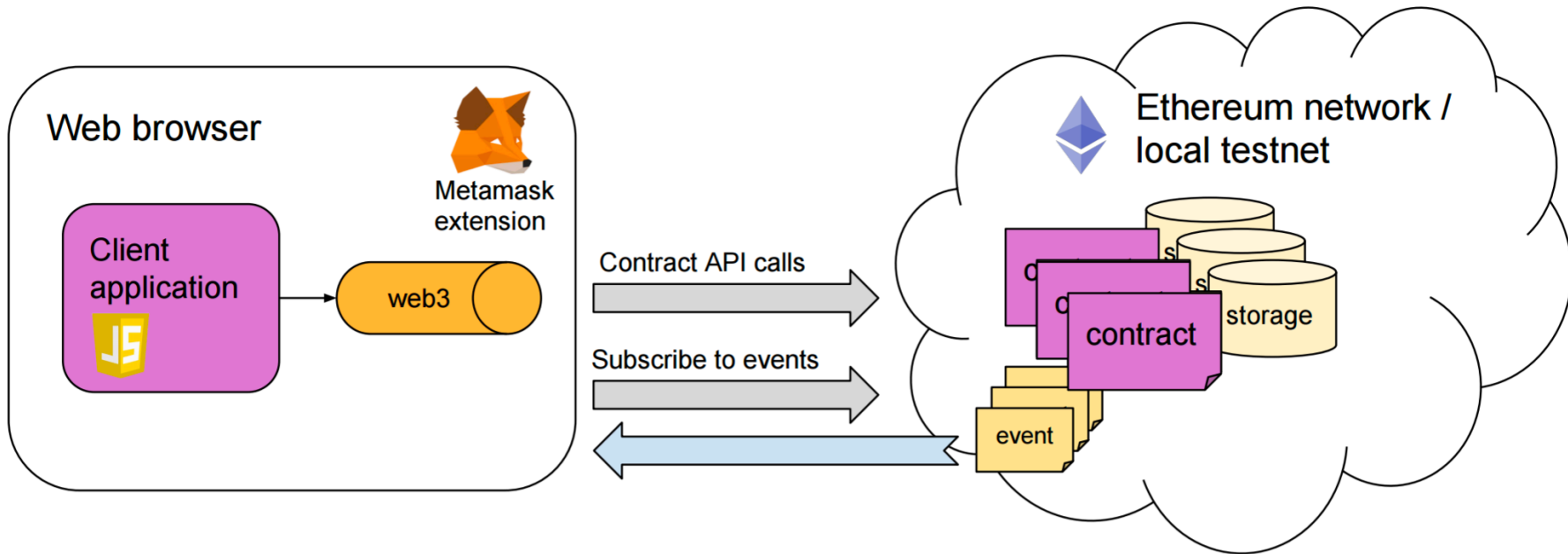
- **Front End** → **Smart Contract** → **Blockchain**

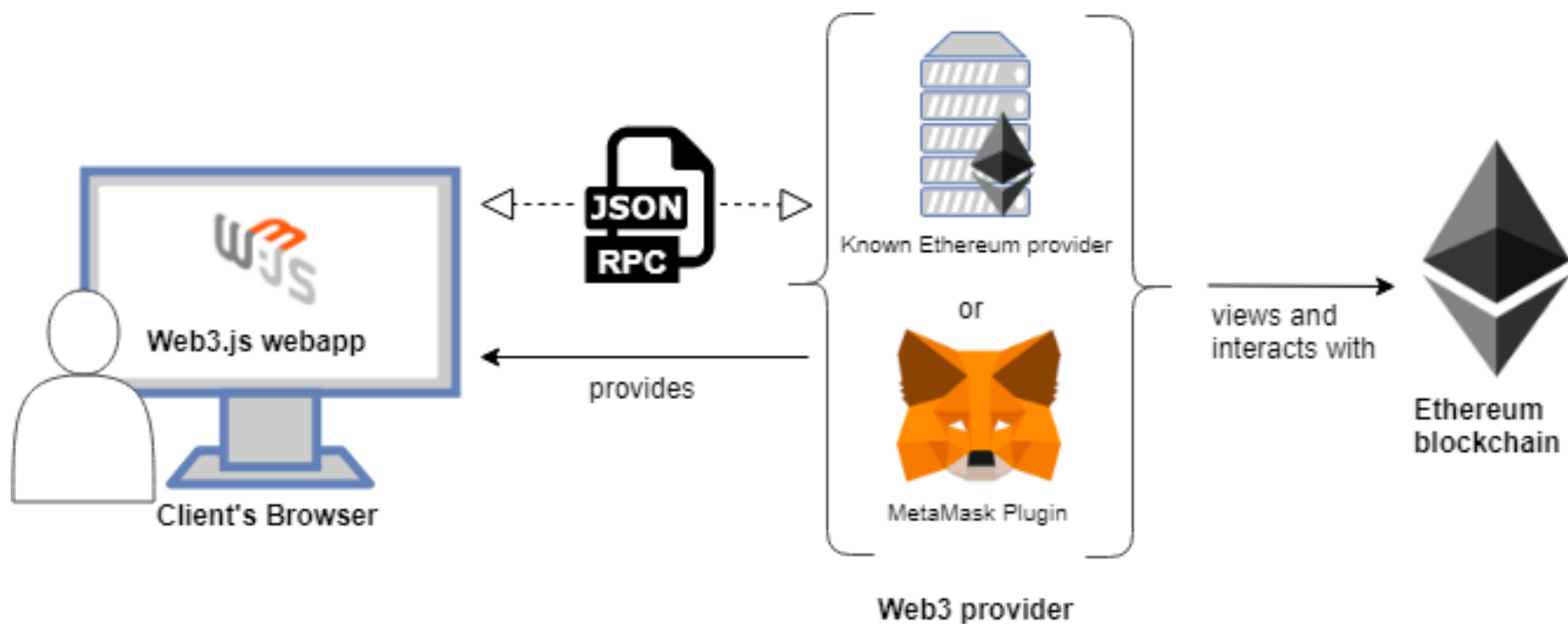


A testnet can either be local (in which case it's only running on a single machine for development and is very lightweight and fast) or public with "fake gas" so it doesn't cost anything to use.

Side note: Create an Ethereum Blockchain from Ground-up

- You can create a private Chain/Testnet (for your organization)
- With Multi-nodes
- You need to config details (gas, difficulty, mining etc.).
- To learn: [Link 1](#), [link 2](#).





Truffle Framework



Learning source for Truffle

- Truffle docs:

<https://truffleframework.com/docs>

Note of Truffle configuration

- Main reference:
<https://truffleframework.com/docs/truffle/reference/configuration>