Build on Ethereum Quickly and Easily with 🏗️ Scaffold-ETH 🛠️

Kevin Jones

Technical Evangelist @ NGINX 🌎

Developer Advocate for Scaffold-ETH 👷 at the Buidl Guidl 🏰

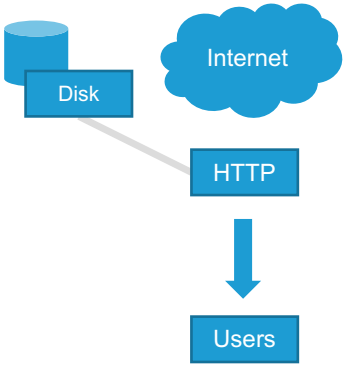
Founder, Blockchain Education Fund 📖

Lead Solidity Mentor @ Growic 👩‍🏫

Photographer & Filmmaker 📸

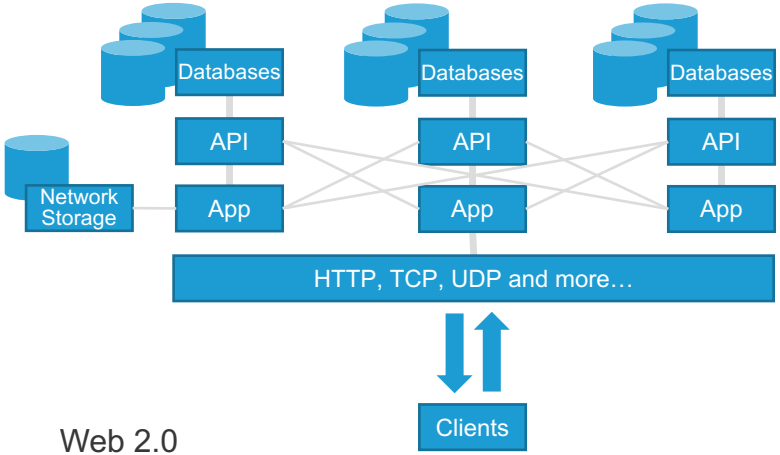
Whaaaaaa….??!

How it started…



Web 1.0
(early 1990s)

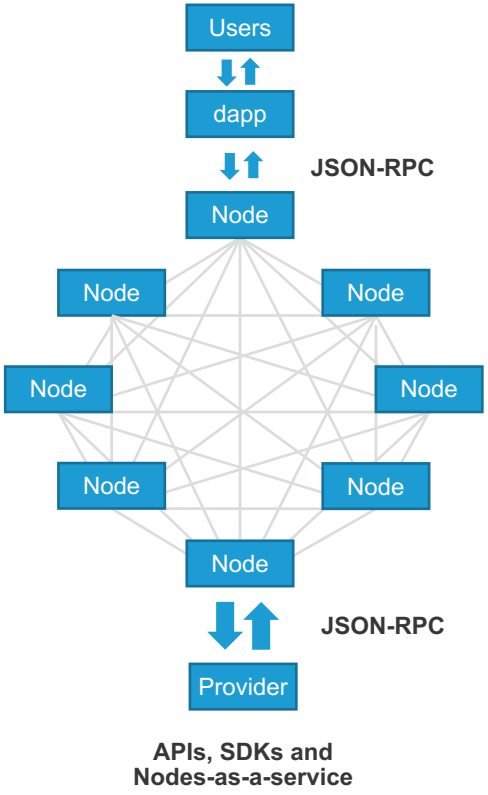
- Means for broadcasting information
- Simple Static HTML Websites
- Some Javascript
- The “Static” Web
- Mainly bare metal servers



Web 2.0
(late 1990s in to 2000s)

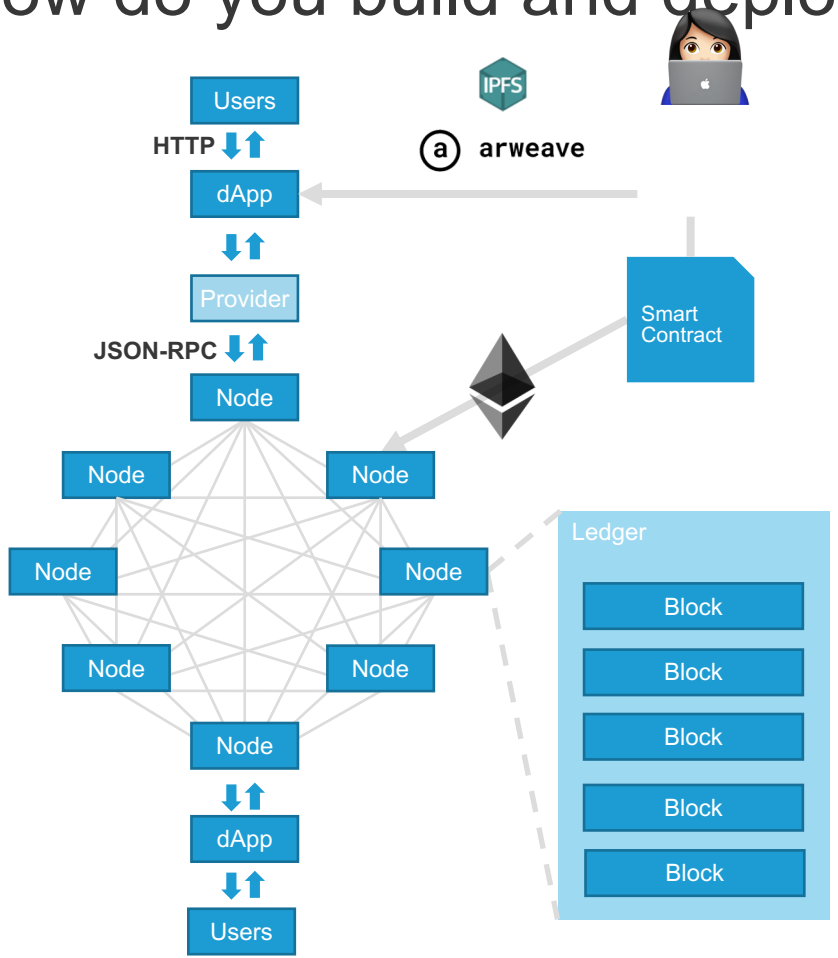
- Infrastructure expanded and concept of distributed systems adopted
- Hosted on cloud providers, virtual machines, containers
- Sites built with User registration and “Dynamic” content
- Information flows between site owner and the user
- More protocols, more layers and much much more complexity
- Birth of APIs (B2B) and wide range of utility (Podcasting, Blogging, Social Media etc)

How its going…



- ➔ ***Decentralized***
  - ◆ Blockchain based (Ledger)
  - ◆ Secured by Consensus (Ether)
  - ◆ Transaction based (Mempool)
  - ◆ “dapps” or “Providers” connect to Nodes using public key cryptography
- ➔ ***Permissionless / Open***
  - ◆ Any user can send transactions
  - ◆ Take your identity with you
- ➔ ***Immutable***
  - ◆ Application logic is stored in “Smart Contracts”
  - ◆ Censorship resistant
- ➔ ***Transparent***

How do you build and deploy on web3?



dApps

- Mostly written in JavaScript (React, Next)
- Usually Deployed on IPFS or Distributed File System like Arweave

Smart Contracts

- Code is stored on the public ledger on verified “blocks”
- Self executing, all logic is handled within the confines of the contract
- Remove the need for an intermediary
- Ability to store value, or act as a bank
- Mostly written in a specialized languages, such as Solidity
- dApps interact via JavaScript API libraries, such as web3.js and Ethers.js
- Providers, provide APIs, SDKs and Nodes-as-a-service

Solidity

- Current version is 0.8.14
- Object-oriented, high-level language
- Static typed, Curly-braces
- Optimization is key
- Source compiled into lower-level Bytecode
- ABI is needed to encode contract calls and read data from transactions

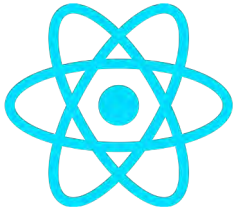
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.14;


contract a {
    uint256 b;
    function c() public {
        b = 2;
    }
}
```


Scaffold-ETH

by BuidlGuidl

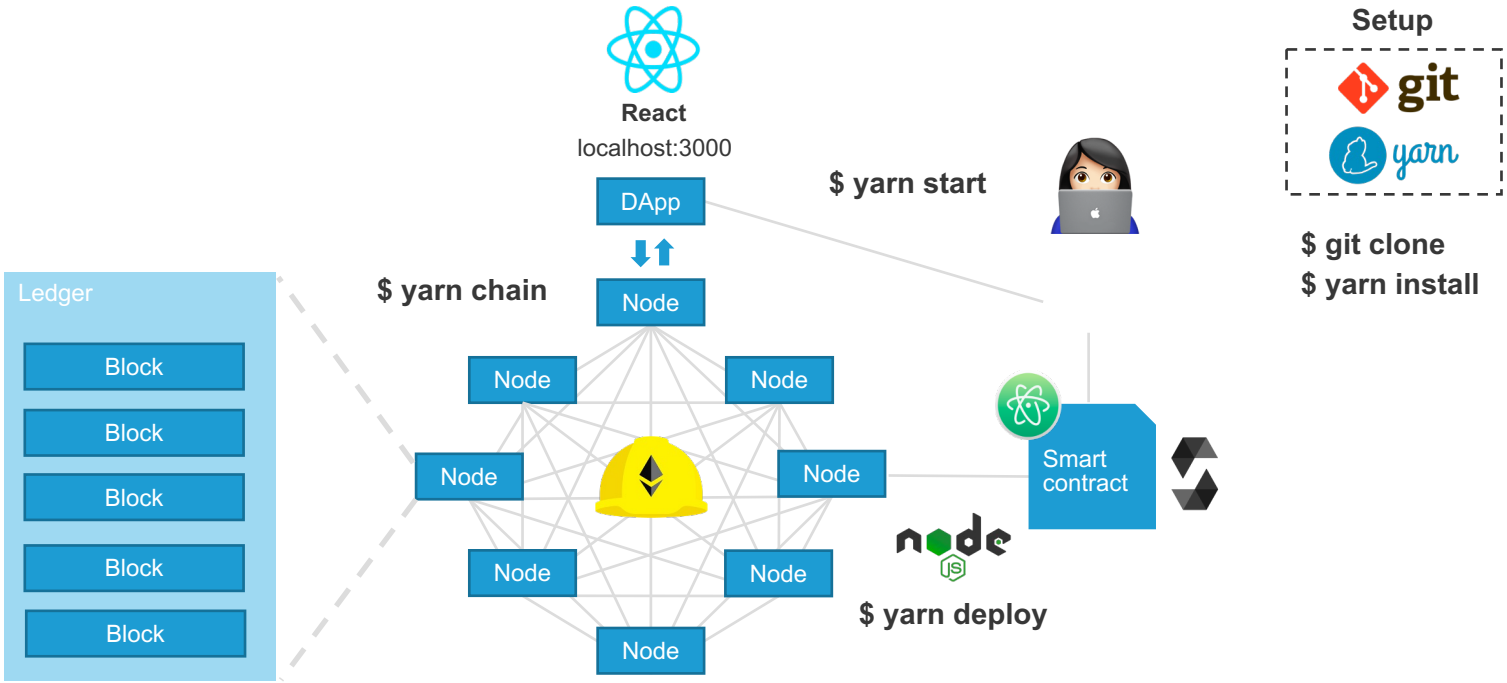
Hardhat



React

Solidity

🏗 Scaffold-ETH



🏃‍♀️ SpeedRunEthereum.com

Need help?