NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

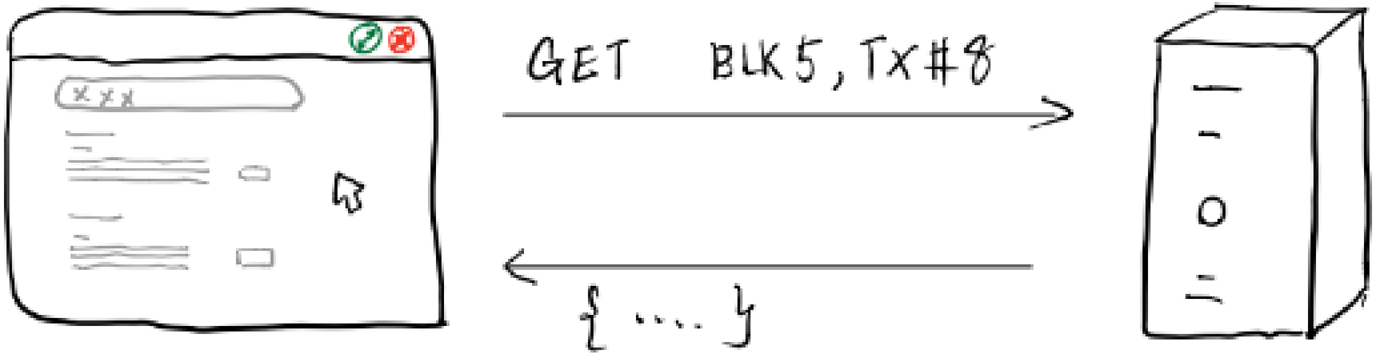
Hands-on Workshop

How to Build a dApp: Tools, Libs & Guides

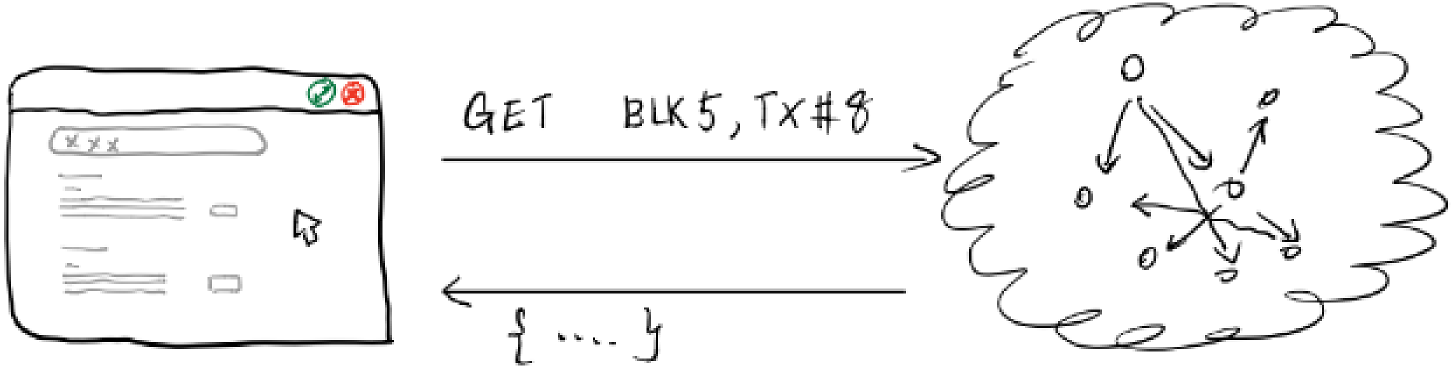
Alex Xiong

Findora

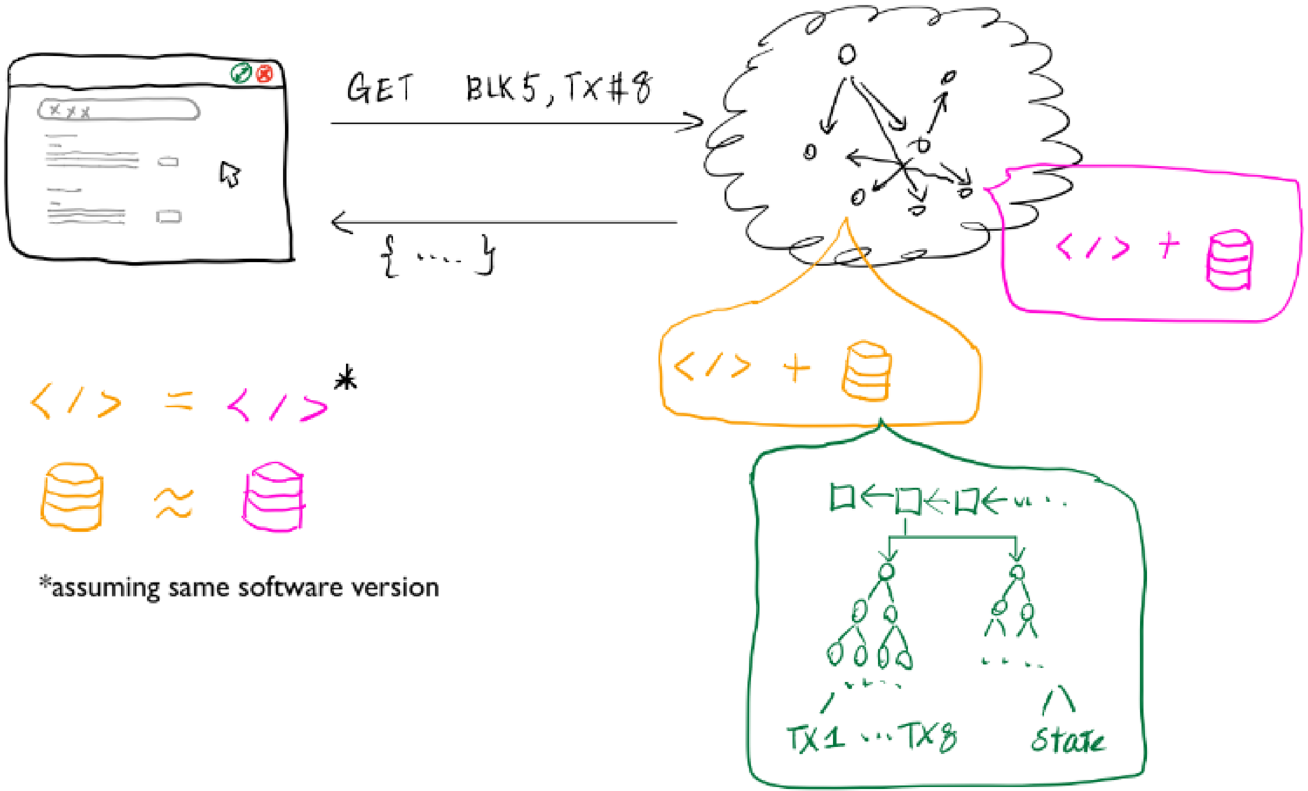
Traditional WebApp



Decentralized Applications (dApp)



Where the hell is "Blockchain"

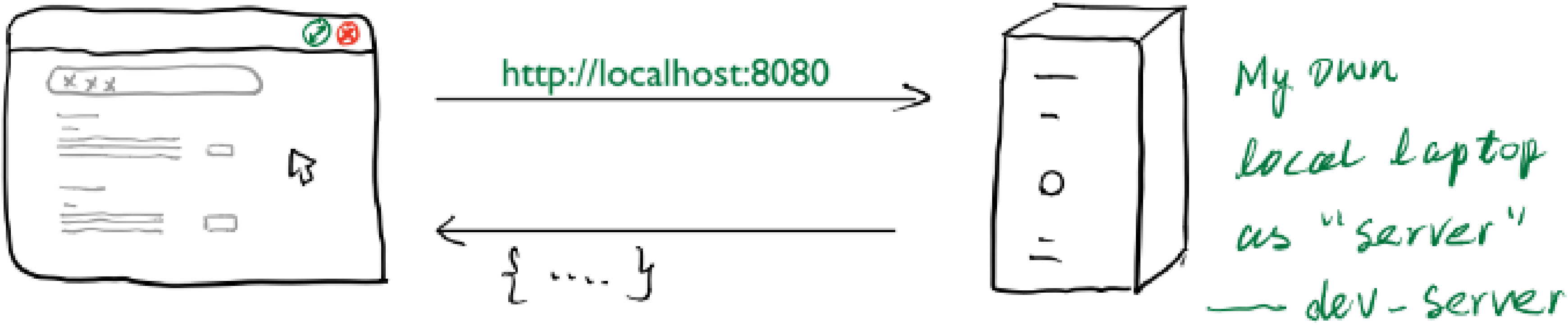


*assuming same software version

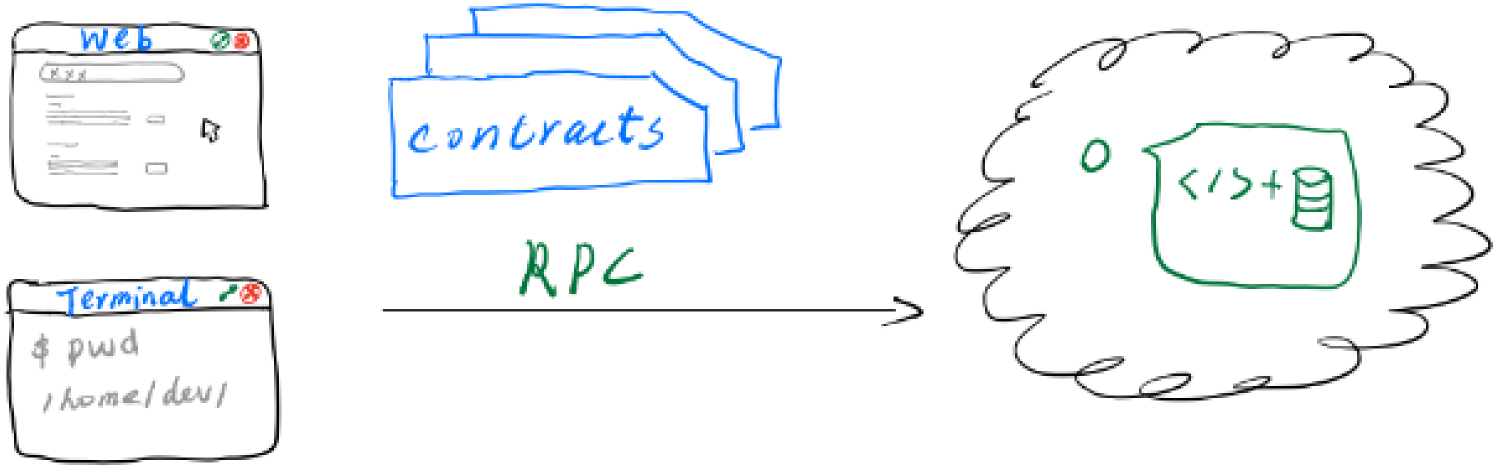
Where the hell is "Blockchain"

- The visual representation of "Blockchain", a list chained blocks, is a data structure only – it is how we store data (incl. Transactions, states for each account)
- Everyone has a shared "blockchain"/ledger
  - Shared, but not exactly the same
  - The code are mostly the same
  - The snapshot at any moment is almost certainly slightly different – some receive more blocks, transactions than others, because network is unreliable, and delivery of any update takes time!!
  - There is no central server where an "authoritative blockchain" lives, every node has its own view of the states

Dev Mode – Classical Webapp



Dev Mode – Blockchain App

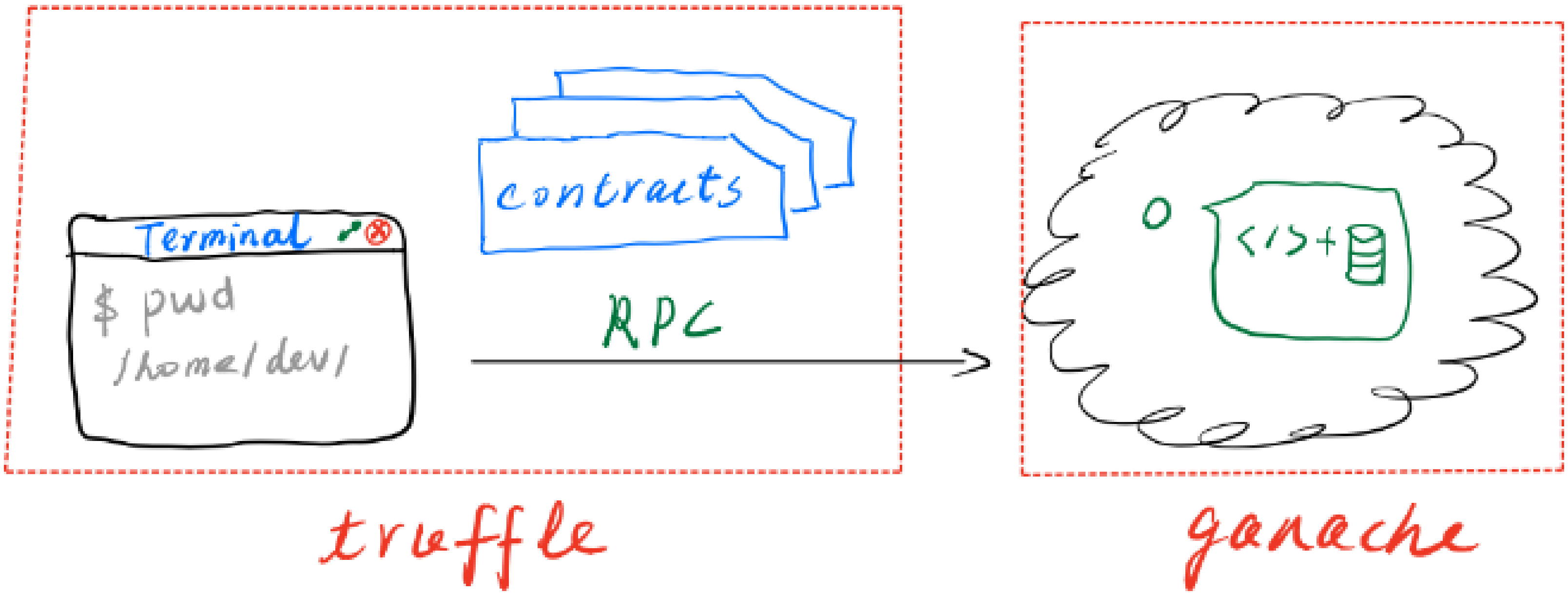


```
$ pwd
/home/dev/
```

Steps to Develop a Simple dApp

1. Design your architecture, components and interaction among them
2. Backend
    - Write business logic in smart contract
    - Write unit tests
3. Deploy Backend
    - Deploy your smart contract to one of Ethereum testnet (e.g. Ropsten, Kovan, Rinkeby, Goerli)
4. Frontend
    - Write a simple WebApp, import web3 libraries (e.g. web3.js, ethers.js, ethjs) to talk to your smart contracts on chain
5. Launch 🚀

Tools: Truffle + Ganache



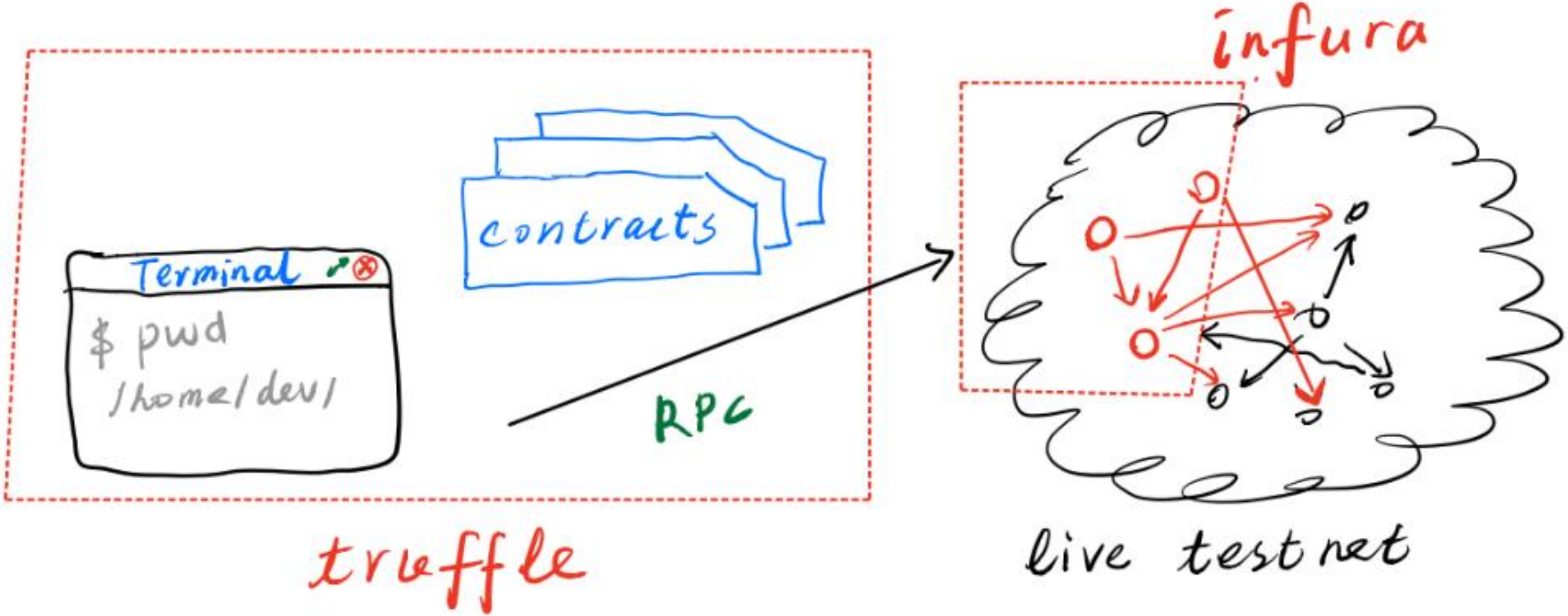
Why Truffle Framework

- Idiosyncratic project structure
- Easier time directly writing tests (less boilerplate of linking artifacts)
- Easier time deploying your contracts to local blockchain or testnets or main net

Why Ganache

- Try locally, just like why you need a dev server when you develop your webapp and try to see how it looks before actually putting them on a production server
- Fast, easy to debug, test and iterate

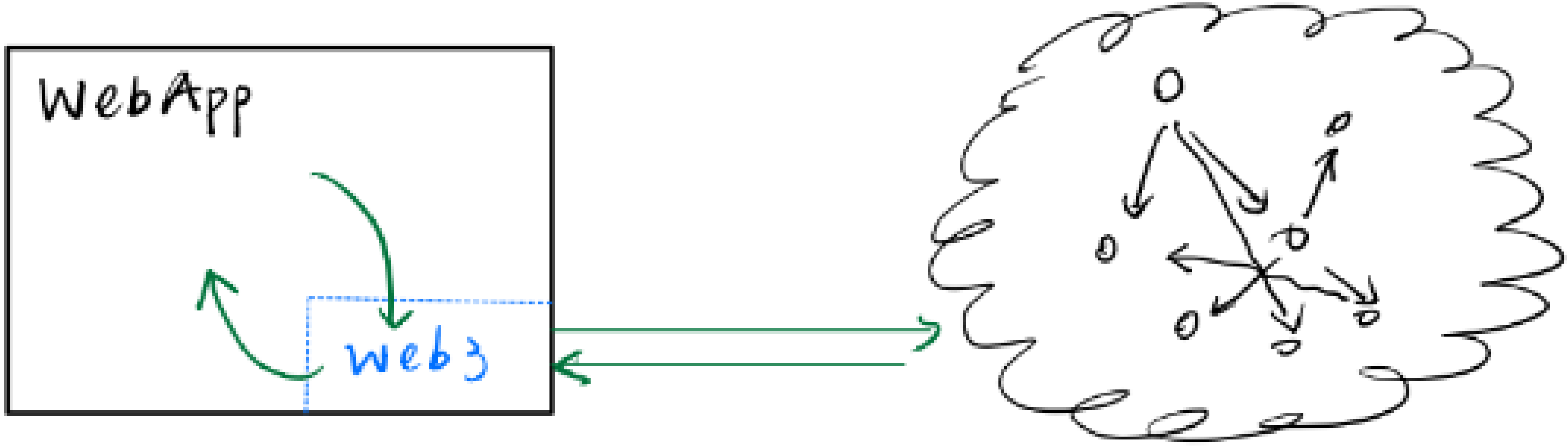
Tools: Infura



Why Infura

- When you try to deploy contracts to live testnets, you either have to actually run an Ethereum testnet node, which is a non-trivial task, or you ask some existing testnet nodes to deploy it for you.
- Infura.io is such a service where you send over your requests to read/write to Ethereum testnet, and it does it for you without forcing you to set up your own node

Libs: web3, ethers.js, or ethjs



Enough talk, let's code

Ethereum Internals

https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture