Ethereum & DeFi for JS/TS developer

ARTEM VOROBEV

1inch Network

План доклада:

- Посмотрим что такое Ethereum, EVM и децентрализованные финансы (DeFi) и как с ними работать

- Сделаем пару транзакций через BscScan и Metamask

- Разберемся как устроент фронтенд децентрализованных приложений (dApp), посмотрим чем занимаются JS/TS разработчики

- Напишем код

Это интересно, потому что в DeFi:

- Работа с финансами идет в `permissionless` экосистеме
- Разработчики это ключевые лица
- Высокая зарплата, а также высокая личная ответсвенность разработчиков
- Транзакции не откатываются, а патчи не накладваются

Обо мне

- С 2006'того шел по наклонной:
- `C++ → C# → JS/TS → Ethereum & DeFi`
- Разрабатываю под Ethereum 3 года года
- Попал в Ethereum через ETH Global хакатоны
- Core Contributor 1inch Network

1inch Network - начало

- Проект основан инженерами на ETH хакатоне в 2019

- Изначально `Single Page Application` в качестве бэкенда Ethereum

- Изначально: агрегация 3 - 5 децентразованных бирж (DEX) на сети Ethereum,

1inch Network - сейчас

~ 70% трафика в своей нише,

- 900K пользователей, обмены на $360M в день
- Google flight для обменов в сети Ethereum, и не только
- Экосистема протоколов / смарт контрактов

7d Volume 1inch Stats

@k06a

$2,717,689,823

7d Volume

24h Volume 1inch Stats

@k06a

$359,297,386

24h Volume

Total Swaps 1inch Stats

@k06a

3,951,749

Total Swaps

Total Users 1inch Stats

@k06a

856,856

Total Users

Quick intro в Ethereum и DeFi

Концепт Ethereum — виртуальная машина поверх блокчейна

1. Транзакции запускают код в машине

2. Код меняет `state` машины

3. `state` хранит что угодно

Если провести аналогию блокчейн, EVM и Git:

- Транзакции — коммиты
- Блоки — merge pull request'а в мастер
- Snapshot мастера на определенном комите это state EVM
- У вас нет прав на `force push` мастер 🙂
- Контракты и транзакции останутся в истории навсегда

Ethereum — за все надо платить эфиром

1. `MOV ADD XOR AND ...` у всего есть цена в `Gas Units`

2. Лимит элементарных операций на тразакцию и блок

3. Концепт газа: `Gas Units * Gas Price` (п)

Appendix G. Fee Schedule

The fee schedule G is a tuple of scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{jumpdest}	1	Amount of gas to pay for a JUMPDEST operation.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
G_{verylow}	3	Amount of gas to pay for operations of the set W_{verylow}.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{\text{warmaccess}}$	100	Cost of a warm account or storage access.
$G_{\text{coldaccountaccess}}$	2600	Cost of a cold account access.
$G_{\text{coldsload}}$	2100	Cost of a cold storage access.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.

И сейчас это очень дорого

- Пример транзакции - вызова метода контракта, комиссия сети $179 за computation

Поэтому для демо будем использовать `Binance Smart Chain`

- Пример транзакции - комиссия сети ($0.48 за computation

Смарт контракт

- В вирутальную машину можно деплоить программы
- Программа = `исполняемый код + состояние`
- Это и есть смарт контракт

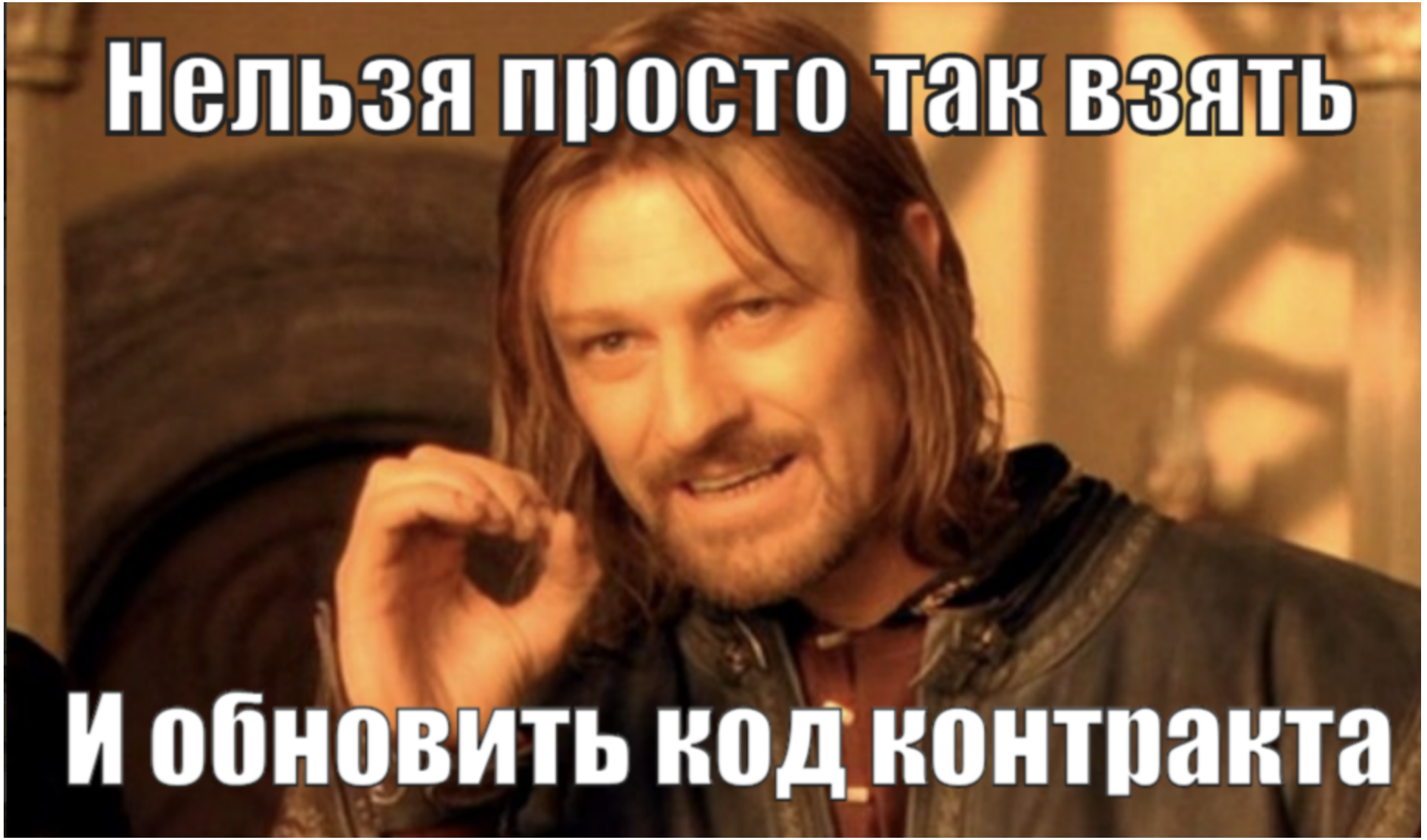
Код это закон

- Программа делает только-то, что в ней запрограммировано

- Прогамму можно верефицировать - скомпилировать исходный код и сравнить

- Пример, контракт токена WBTC

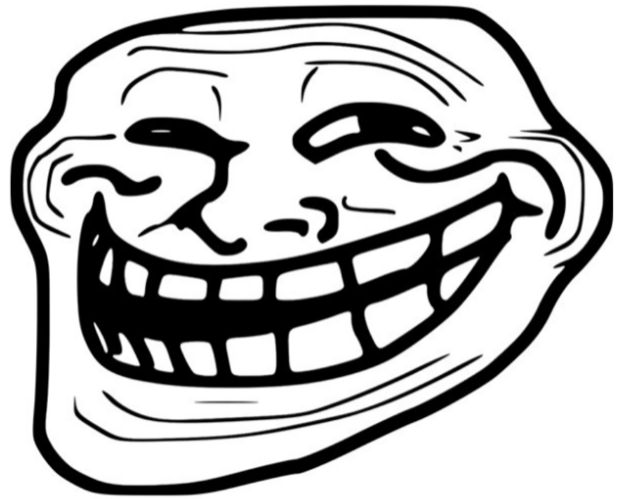
А как изменить код контракта?


Нельзя просто так взять
И обновить код контракта

Можно:

- Использовать `Proxy` паттерн
- Написать `Upgradable` контракт
- Или даже вызвать деструктор (`selfdestuct`)
- Примеры OpenZeppelin

Но ты не написал



этот код в контракте

И это даже хорошо

- Код это закон
- Вопрос доверия

А как понять что контракт обновили ?

А как понять что контракт обновили ?

- Очень просто, у нас же блокчейн
- Можно найти на etherscan

Погружаемся в DeFi

Смарт контракты могут работать сообща

1. Пользователь отправяет транзакцию

2. Транзакция запускает метод на контракте

3. Метод контракта может вызвать метод другого контракта

Что бы контракты были совместимы нужны интерфейсы

ERC-20 токены USDT, DAI

Пример, разрешим контракту 1inch потратить потратить 10,000 BUSD моего кошелька

Контракты - обменники (AMM или Liqudity Pool)

Automated Market Maker

Пример uniswap

Обменивает токены по формуле `x * y = const`

Пример транзакции

Контракты ломбарды

Кредиты в одном токене под залог другого

Контракт может распродать обеспечение

Пример Compound

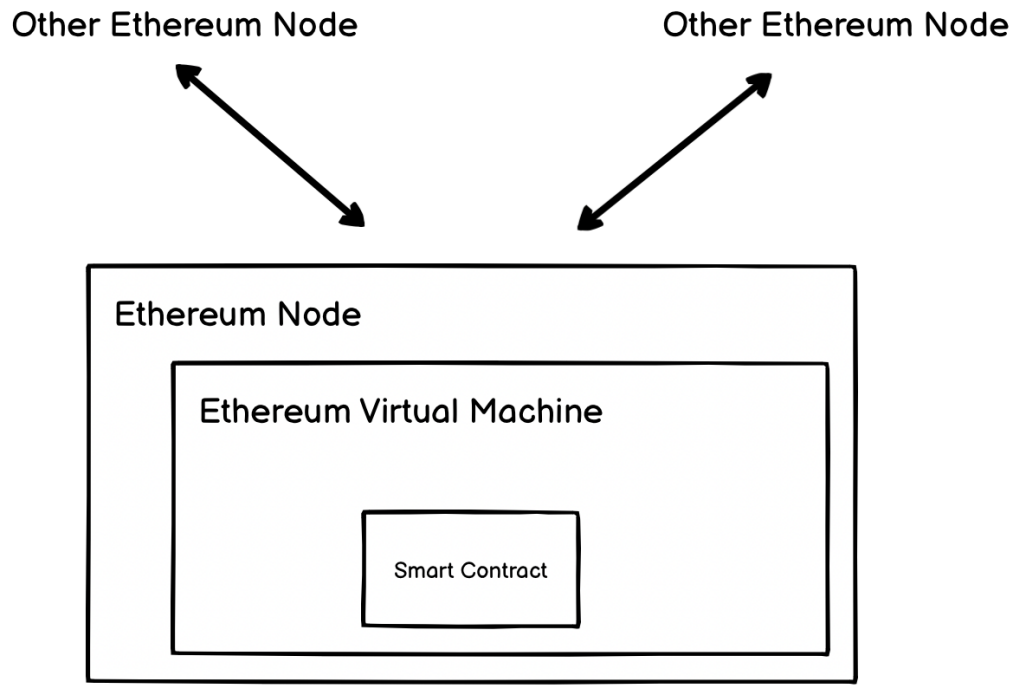
4. Контракты Оракулы - поставщики цены/погоды/ результатов выборов, Chainlink

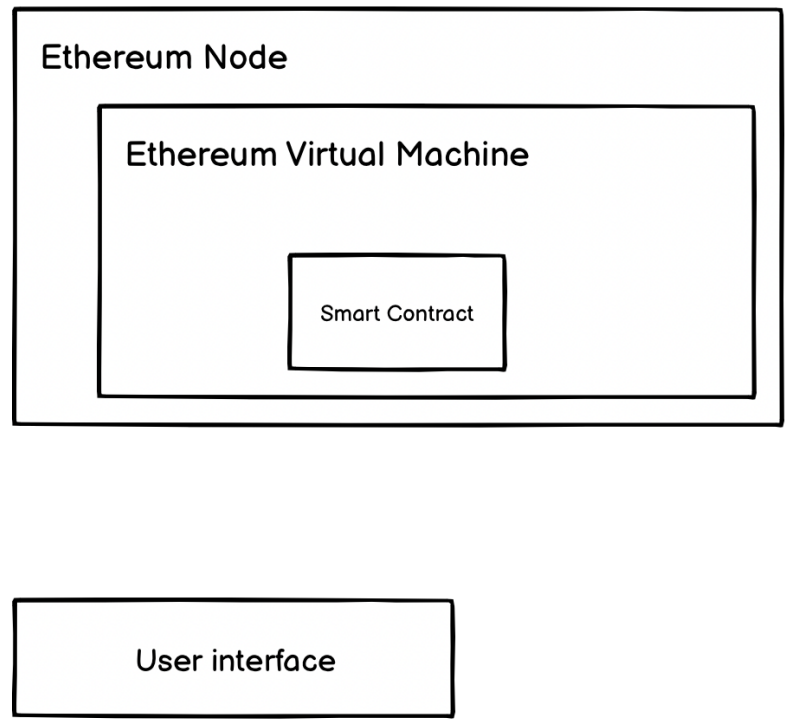
dApp - Decentralized Application

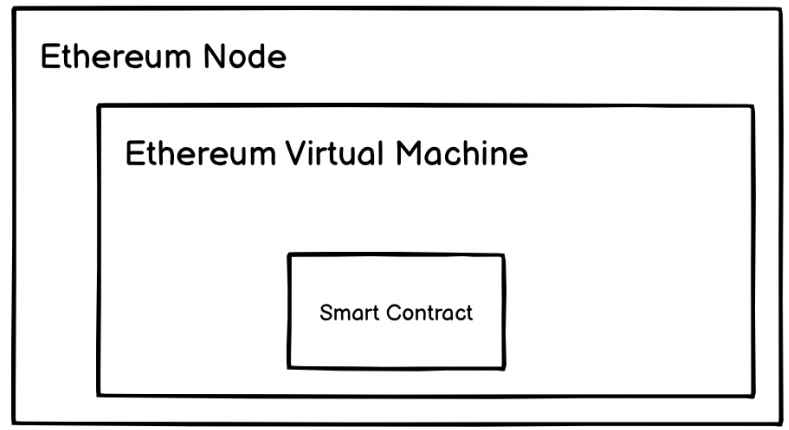
- Uniswap
- Compound
- 1inch все это dApp


Эх, надоело писать фронтенд
А давай как я разрабатывать dAPP
Да, ты че пес я dAPP архитектор
дак ты просто фронтендер как и я
А что нужно делать то?
npm install web3

Посмотрим как устроен dApp UI

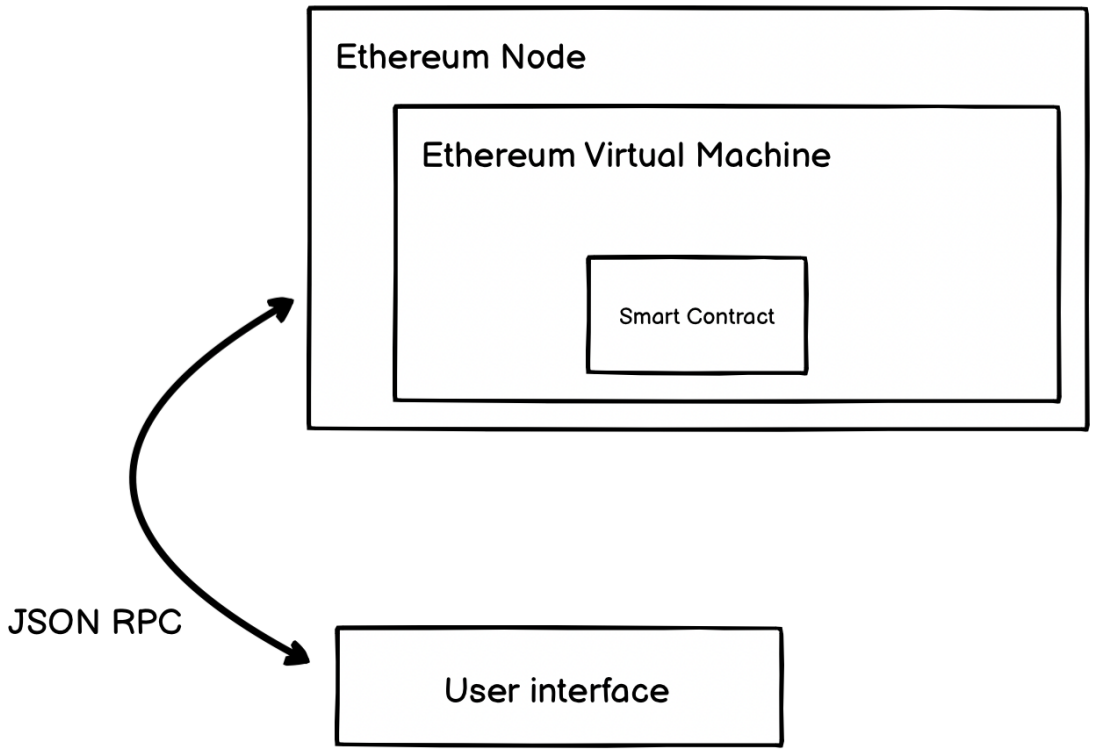

Other Ethereum Node
Other Ethereum Node
Ethereum Node
Ethereum Virtual Machine
Smart Contract


Ethereum Node
Ethereum Virtual Machine
Smart Contract
User interface



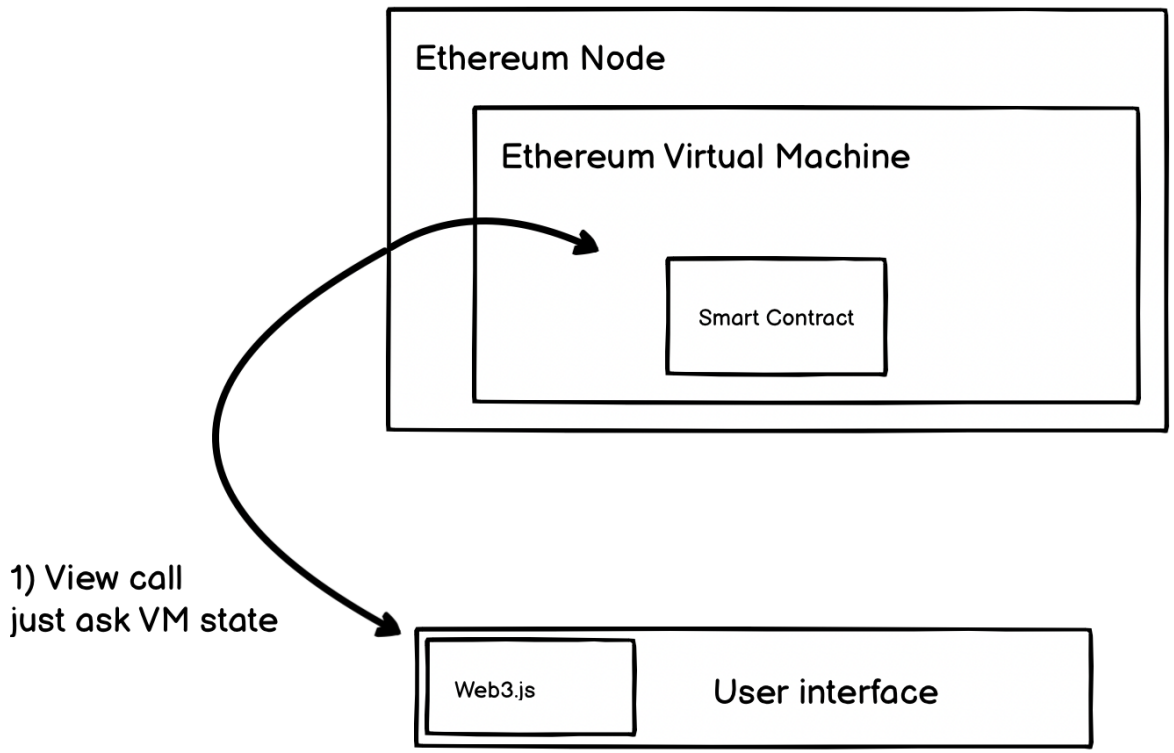
User interface

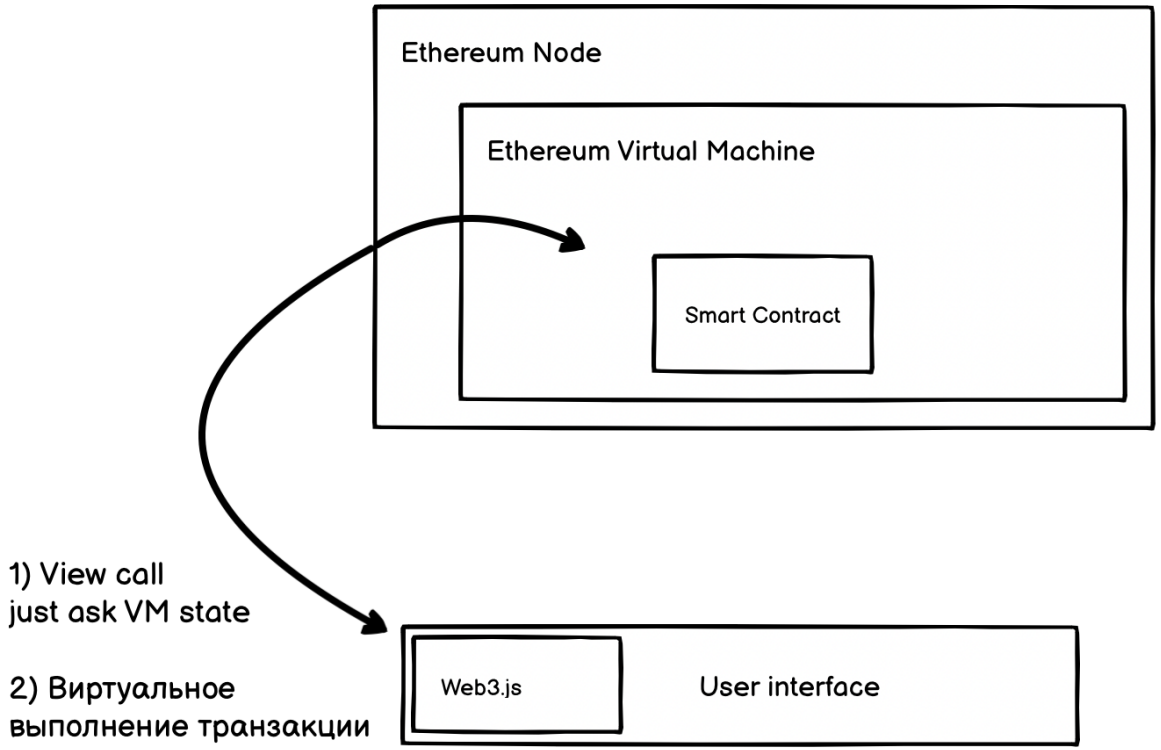
- Usually SPA hosted on CDN with fallback to IPFS
- Usually allow open source to allow everyone to verify
- The matter of trust


Ethereum Node
Ethereum Virtual Machine
Smart Contract
JSON RPC
User interface

1) View режим

До того как кошелек подключен


Ethereum Node
Ethereum Virtual Machine
Smart Contract
1) View call
just ask VM state
Web3.js
User interface



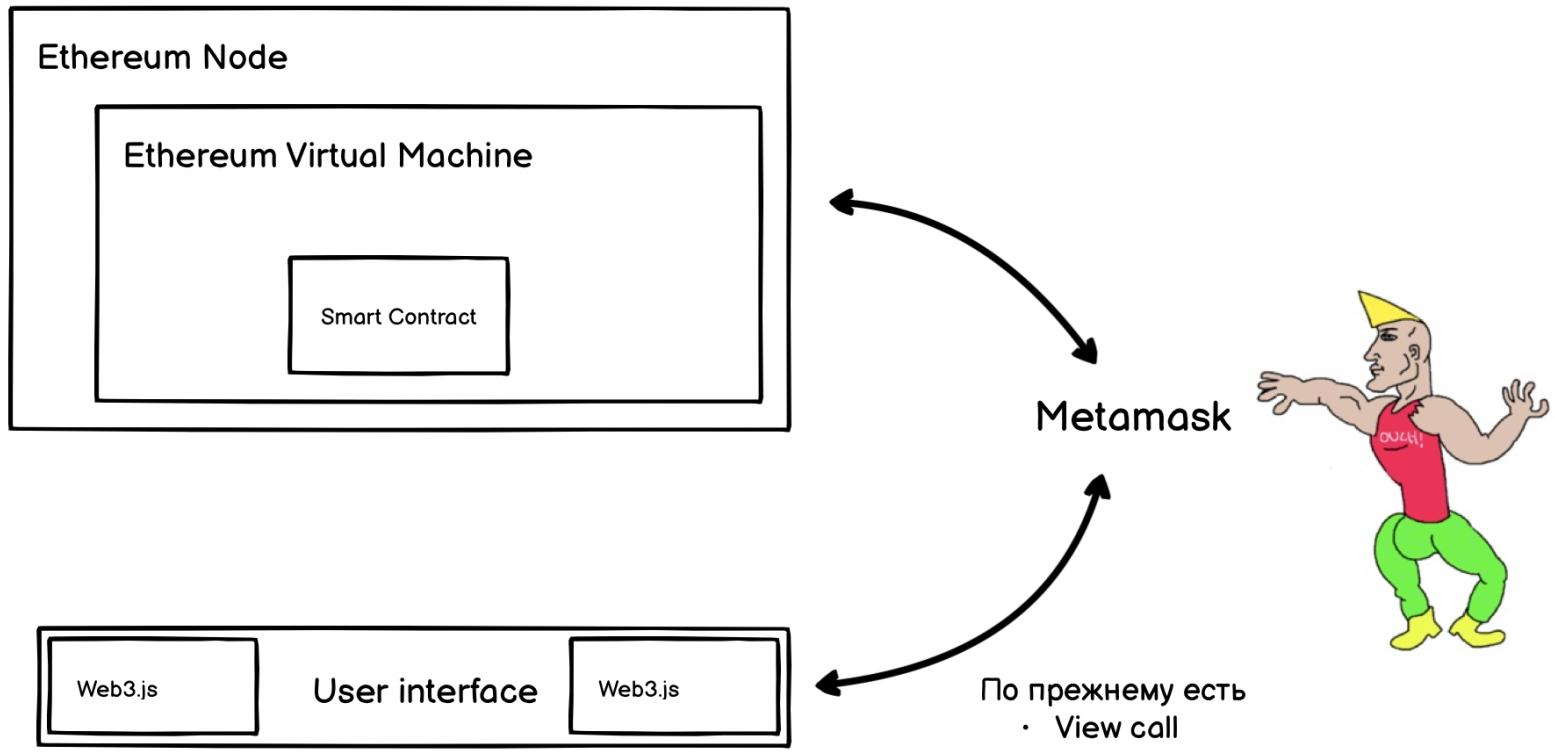
1) View call
just ask VM state

2) Виртуальное
выполнение транзакции
Estimate transaction

execute transaction without
signature to get result and
gas consumed

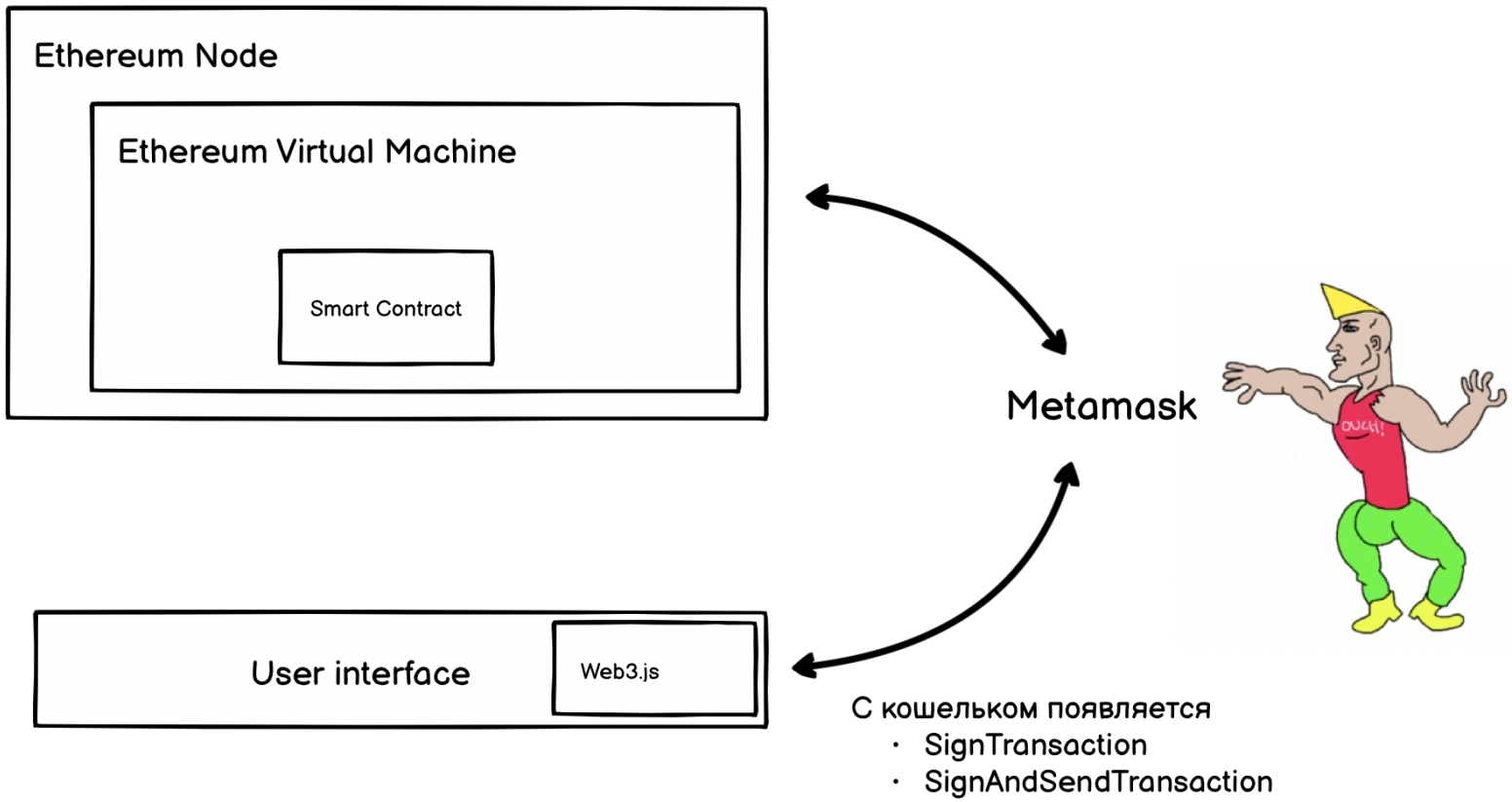
2) Write Режим

Когда кошелек подключен



По прежнему есть

- View call
- Эстимейт


Ethereum Node
Ethereum Virtual Machine
Smart Contract
Metamask
User interface
Web3.js
С кошельком появляется
• SignTransaction
• SignAndSendTransaction

3) Режим Reallife

- Много контрактов

- Нужно составлять сложный байт код вызовов контрактов

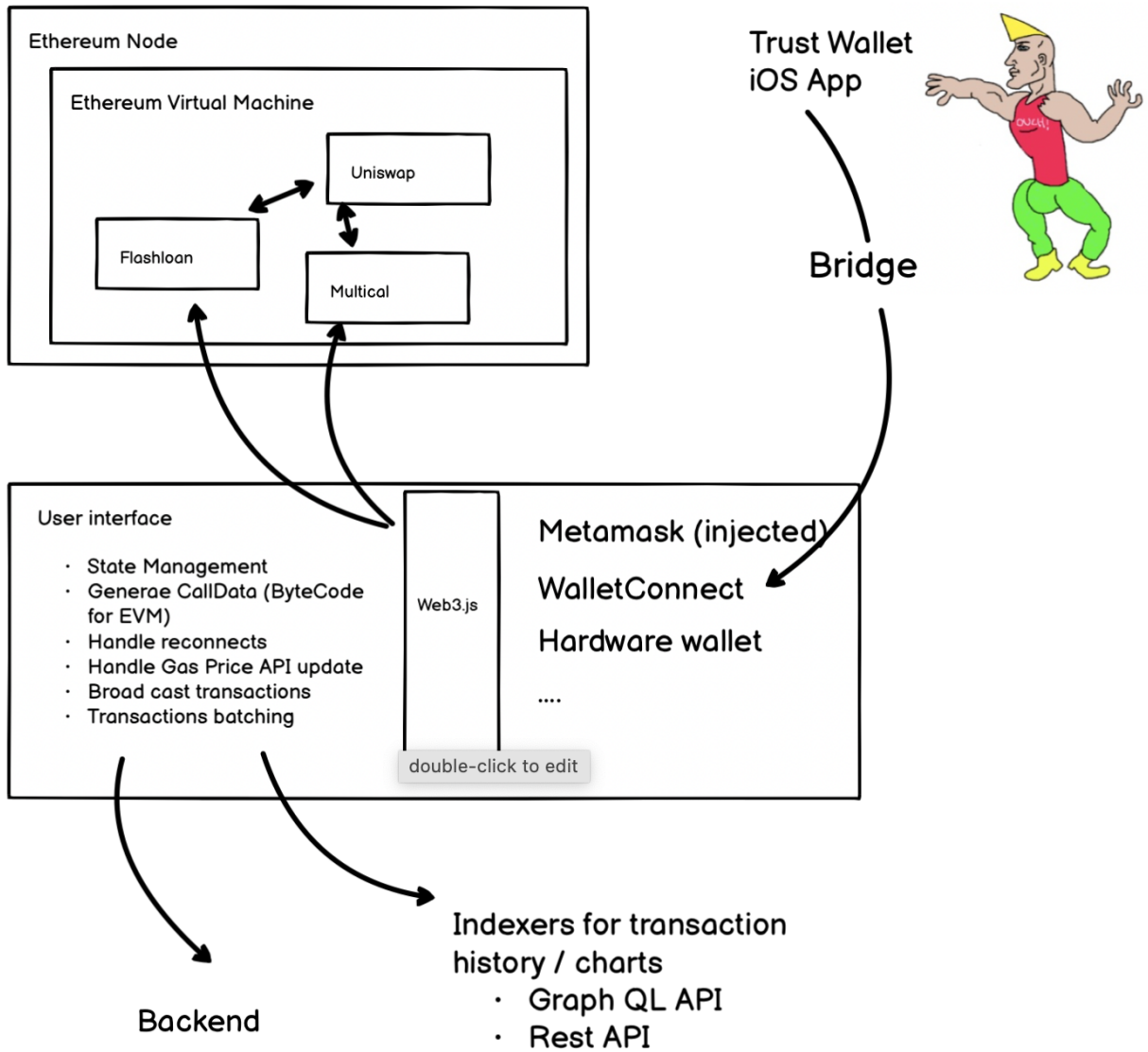
- Иногда нужно создавать новые контракты через фабрики

- Много провайдеров кошельков

- Появляется backend

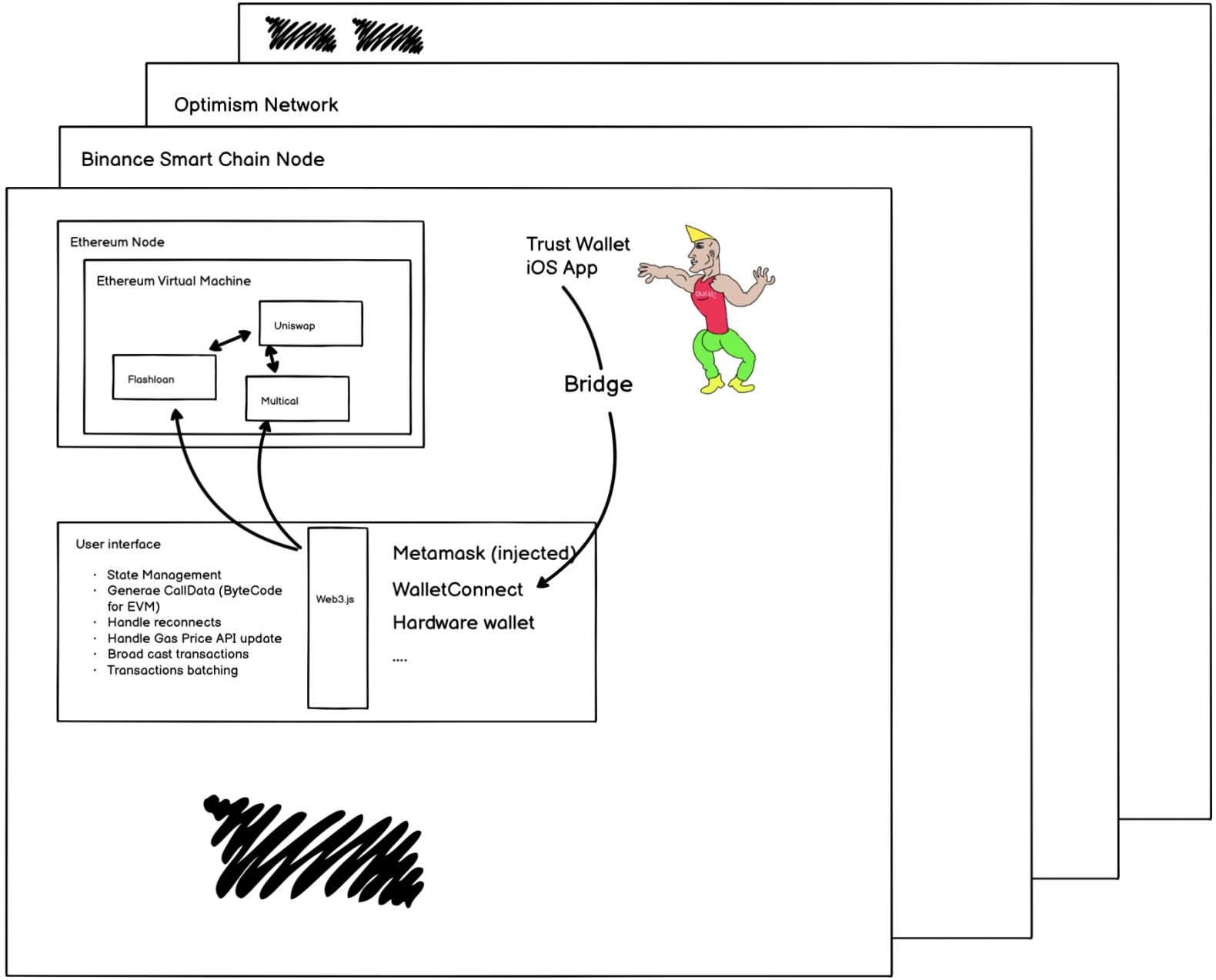
- Некоторые данные можно достать только через индексаторы

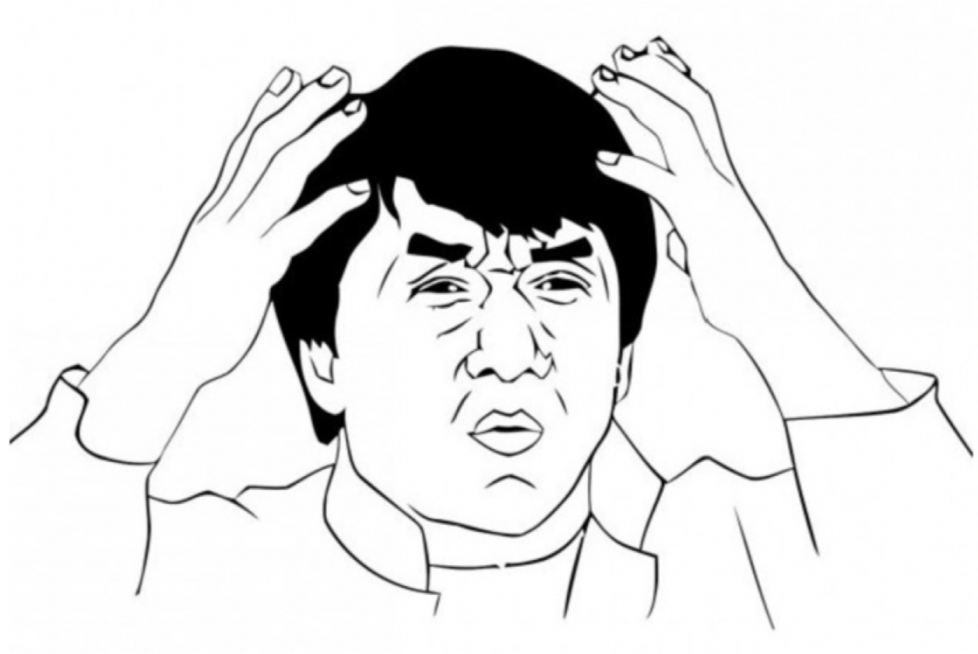
Режим реальной жизни



4) Режим Hardcore

Режим Хардкор - Support Several networks





Давайте напишем код

Полезные ссылки

- MetaMask Guide
- Web3, Ethers
- EIP-20
- ERC-20

Полезные ссылки

- 1inch.io - 1inch Network

- Programming Bitcoin - книжка для тех кто решил разобраться как следует

- Awesome DeFi

- Ethereum yellowpaper

Q&A

Увидимся в DeFi