

HSE {LAB}

БЛОКЧЕЙН БИТКОИНА
МАСТЕР-КЛАСС



18 октября 2016 г.

(c) Nikita Zhavoronkov <nikzh@nikzh.com>

СОДЕРЖАНИЕ МАСТЕР-КЛАССА

- Теоретическая часть «Блокчейн Биткоина»
- Практическая часть №1 «Парсинг блокчейна Биткоина — создание примитивного блокчейн-обозревателя»
- Практическая часть №2 «Запись произвольной информации в блокчейн Биткоина — нотаризация данных»



ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

КРАТКО О КРИПТОВАЛЮТАХ

Самая известная и популярная на сегодняшний день — Биткоин :

- Создан в 2009 году;
- Около 250.000 транзакций в день;
- Общий дневной оборот около \$1 млрд.;
- Капитализация около \$10 млрд (~2% AAPL).

КРАТКО О КРИПТОВАЛЮТАХ

Две основные составляющие:

- Средство накопления (сравнение: золото, доллары, рубли);
- P2P ЭПС (сравнение: VISA, PayPal, Яндекс.Деньги).

Результат: смесь — цифровые наличные.

КРАТКО О КРИПТОВАЛЮТАХ

Основные особенности:

- Децентрализованность и плановость эмиссии (нет аналога ЦБ);
- Децентрализованность платежей (нет аналога обычных банков);
- (Псевдо)анонимность участников сети.

КРАТКО О БЛОКЧЕЙНЕ

- Блокчейн — основополагающая технология в криптовалютах;
- На сегодняшний день *публичные* блокчейны в реальной практике используются только в криптовалютах (что бы там не говорили маркетологи);

КРАТКО О БЛОКЧЕЙНЕ

- Имеющие ценность токены криптовалют являются «топливом» для поддержания жизни и безопасности блокчейна в их основе;
- Транзакции в блокчейне могут иметь как монетарное, так и немонетарное применение (в случае Биткоина — 99% транзакций имеют монетарный характер).

КРАТКО О БЛОКЧЕЙНЕ

- Блокчейн Биткоина не такой «интересный» как многие другие: скриптовой язык, используемый для созданий транзакций, довольно простой и не позволяет создавать сложные конструкции типа DAO.
- Зато безопасный (не является полным по Тьюрингу, по сути — простейший стековый язык).

КАКУЮ ПРОБЛЕМУ РЕШАЕТ БЛОКЧЕЙН?

- Что такое современные ЭПС? «У меня есть деньги» — это всего лишь запись на каком-то сервере. Необходимо доверять оператору ЭПС, что он не отнимет ноль справа;
- Что такое «у меня есть квартира в Москве»? Это всего лишь запись в ЕГРП, которая хранится на каком-то сервере. Пропала запись — ...;
- Это всё, безусловно, преувеличения — но их объединяет одно — проблема доверия к третьей стороне.

ПОЧЕМУ ИМЕННО БЛОКЧЕЙН?

- Блокчейн даёт возможность неограниченному кругу лиц проводить сделки без участия централизующего элемента;
- Почему не придумали это раньше? Долгое время никаких криптовалют не существовало, потому что не была решена проблема «двойных трат».

РЕШЕНИЕ ПРОБЛЕМЫ «ДВОЙНЫХ ТРАТ»

Алиса передала Бобу **реальное яблоко**: они совершили сделку, у Боба есть яблоко, а у Алисы — больше нет (так работает наличный расчёт).



РЕШЕНИЕ ПРОБЛЕМЫ «ДВОЙНЫХ ТРАТ»

- Алиса передала Бобу **цифровое яблоко** (послала по почте *Apple.jpg*): у Боба теперь есть яблоко, но оно осталось и у Алисы. Более того, она может послать его и Чарли: это и будет «двойной тратой».

РЕШЕНИЕ ПРОБЛЕМЫ «ДВОЙНЫХ ТРАТ»

- Решение: ведение **бухгалтерской книги**, где было бы записано, что Алиса передала цифровое яблоко Бобу, и только он теперь его может использовать (так работает безналичный расчёт).
- Проблема: необходимость **посредника**, которому все должны доверять.

РЕШЕНИЕ ПРОБЛЕМЫ «ДВОЙНЫХ ТРАТ»

- Криптовалюты решают эту проблему с помощью распределённой сети, каждый участник которой ведёт полную «бухгалтерскую книгу» (**блокчейн**).

СТРУКТУРА БЛОКЧЕЙНА БИТКОИНА

ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

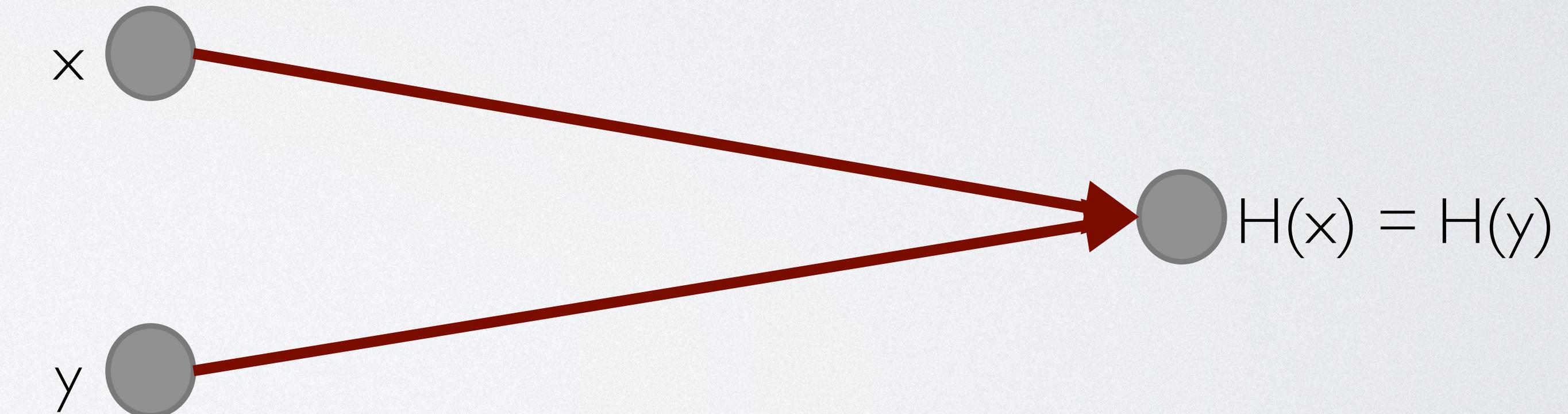
- Криптовалюты они на то и «крипто», чтобы там было что-то про криптографию:
 - Криптографические хеш-функции;
 - Структуры на основе хешей (указатели, блокчейн, дерево Меркла);
 - Ассиметричное шифрование — открытые ключи как идентификаторы;
 - Цифровые подписи.

КРИПТОГРАФИЧЕСКИЕ ХЕШ-ФУНКЦИИ

- Любая хеш-функция:
 - Берёт информацию (любую) в качестве параметра;
 - На выходе выдаёт результат фиксированного размера (обычно меньшего, чем входная информация). Пример: контрольная цифра в ИНН, банковском счёте и т.д.

КРИПТОГРАФИЧЕСКИЕ ХЕШ-ФУНКЦИИ

- Основное свойство криптографических хеш-функций — стойкость к коллизиям (когда два разных параметра дают один и тот же результат): никто не может найти такие x и y , что $x \neq y$, а $H(x) = H(y)$;



$SHA-256('Hello IFES!') = 0x201014633F4E4FB8F81679CD94B413237D6BF3CF79BBD93EE91BE827B6E86A2D$

$SHA-256('Hello IFES!!') = 0xECD0FD9AC5303B75308B5C991BF0FED0FFAB5B2312C4EA5FAB0668171705AFD4$

КРИПТОГРАФИЧЕСКИЕ ХЕШ-ФУНКЦИИ

- Что это даёт: мы можем посчитать хеш от информации (от большого объёма информации) и хранить только его (экономим в объёме). Чтобы проверить корректность информации нам будет достаточно только хеша, т.к. если информацию подменить, то хеш у неё будет другой;
- В Биткоине используется функция SHA-256.

$\text{SHA-256}(\text{'Hello IFES!'}) = 0x201014633F4E4FB8F81679CD94B413237D6BF3CF79BBD93EE91BE827B6E86A2D$

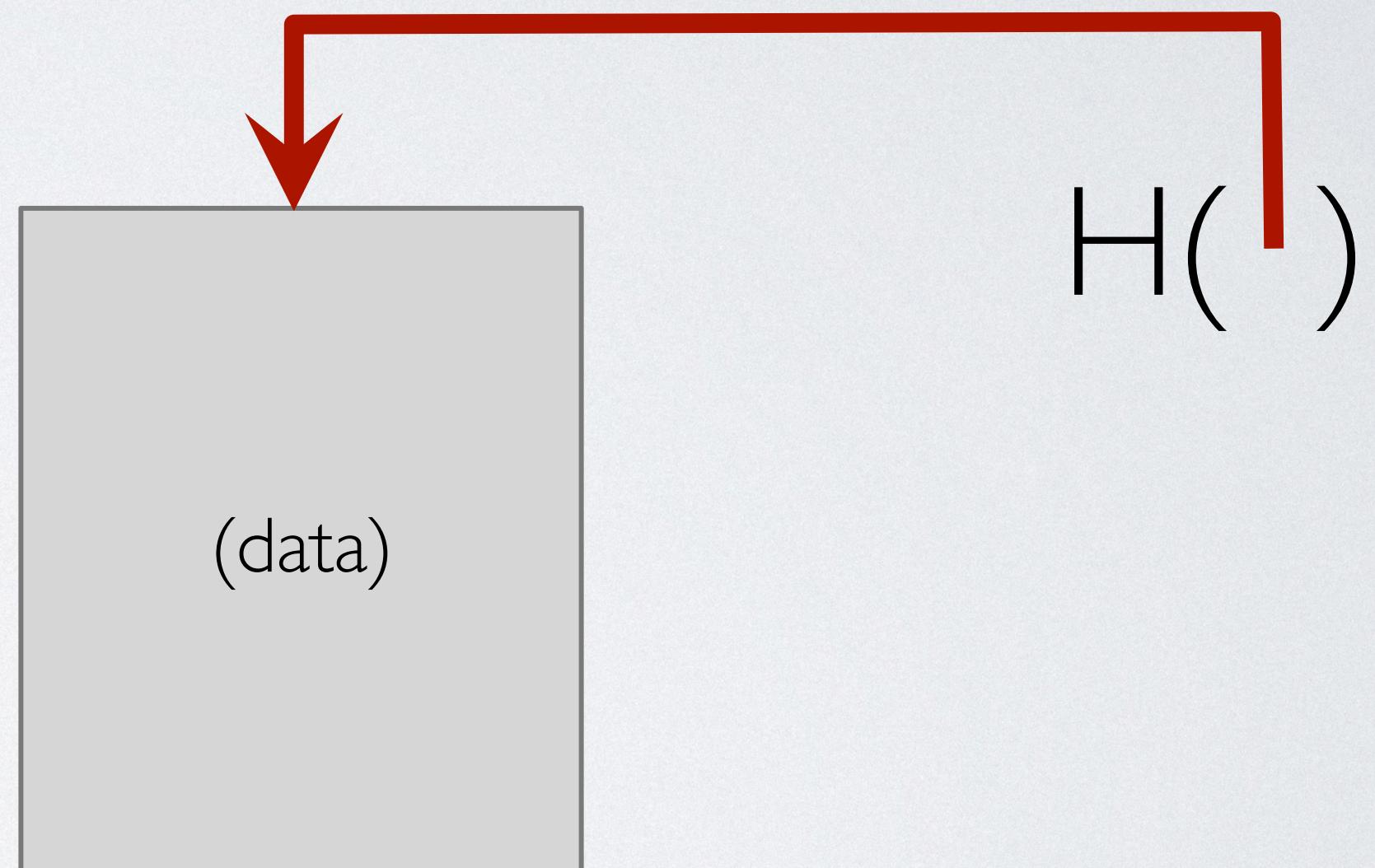
$\text{SHA-256}(\text{'Hello IFES!!'}) = 0xECD0FD9AC5303B75308B5C991BF0FED0FFAB5B2312C4EA5FAB0668171705AFD4$

СТРУКТУРЫ НА ОСНОВЕ ХЕШЕЙ

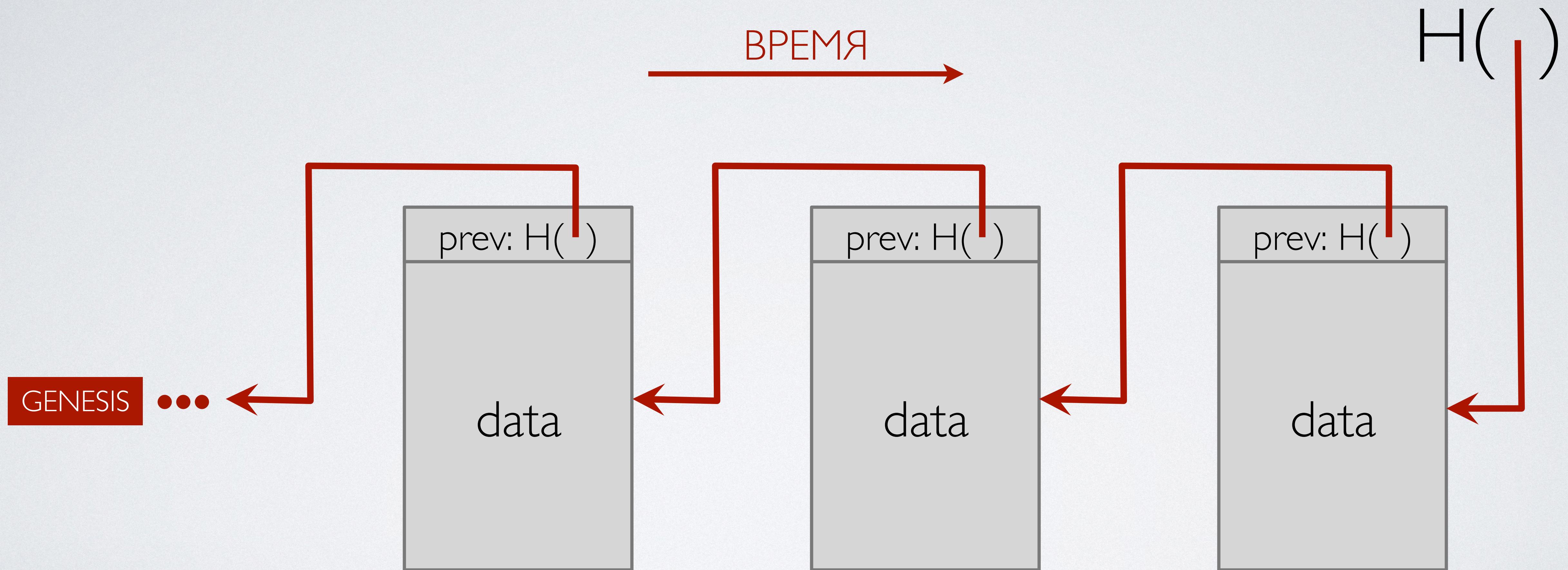
- Хеш-указатель;
- Связанный список (блокчейн);
- Дерево Меркла (содержимое блоков — транзакции).

ХЕШ-УКАЗАТЕЛЬ

- Свойства:
 - Указывает на место в памяти, где хранится блок информации;
 - Является хешем этой информации;
- Если у нас есть хеш-указатель, мы можем:
 - Получить блок информации;
 - Подтвердить, что информация не изменилась.

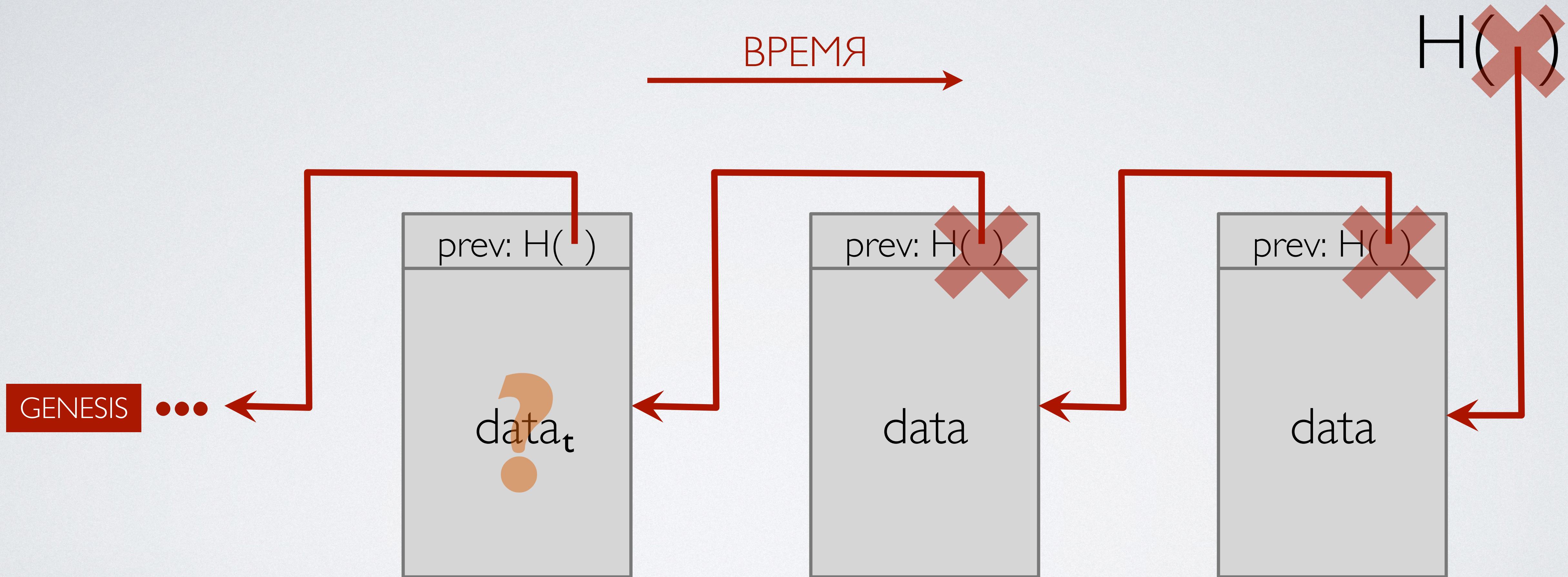


СВЯЗАННЫЙ СПИСОК (БЛОКЧЕЙН)



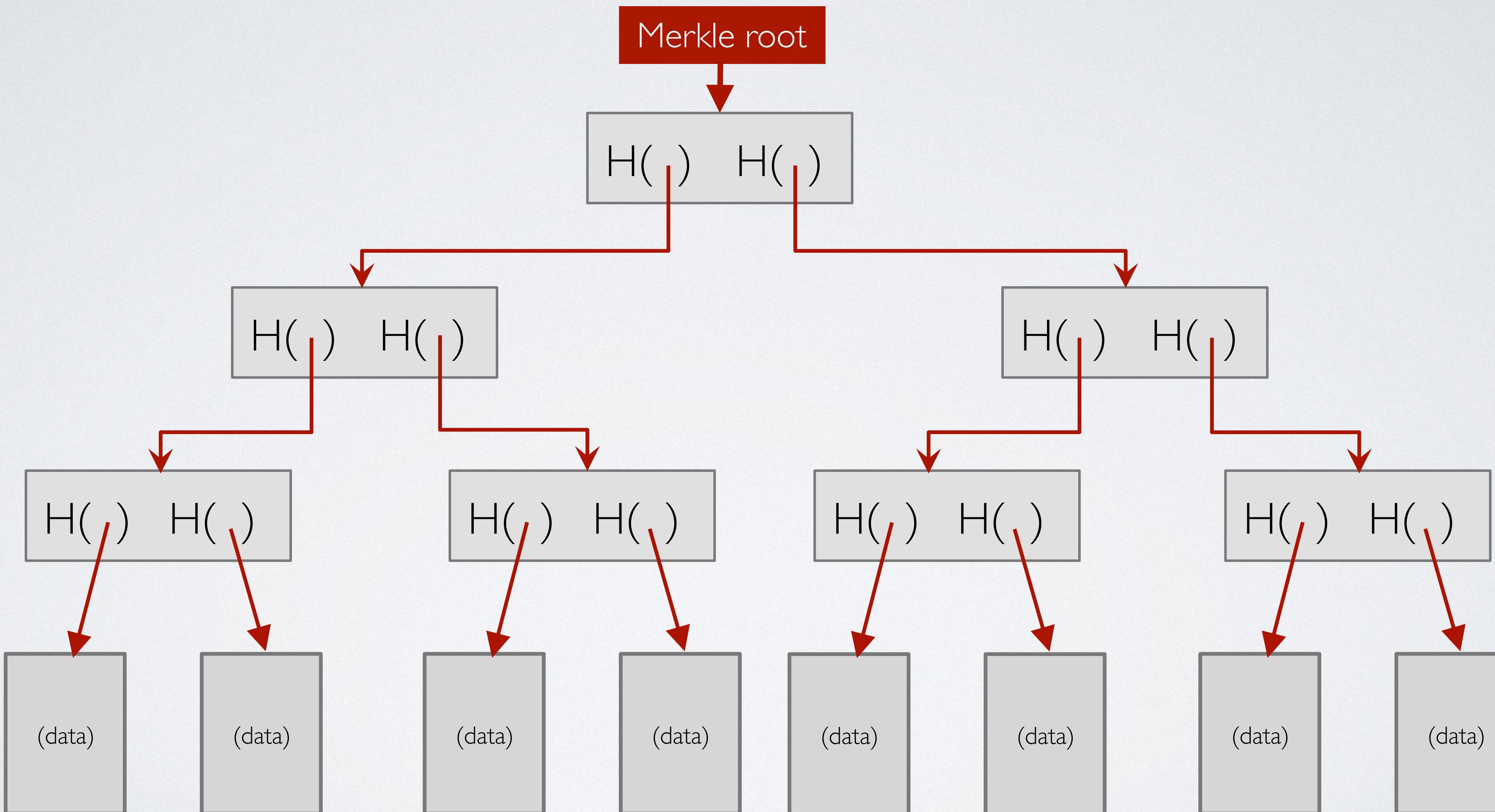
Это и есть тот самый блокчейн — цепь блоков

СВЯЗАННЫЙ СПИСОК (БЛОКЧЕЙН)



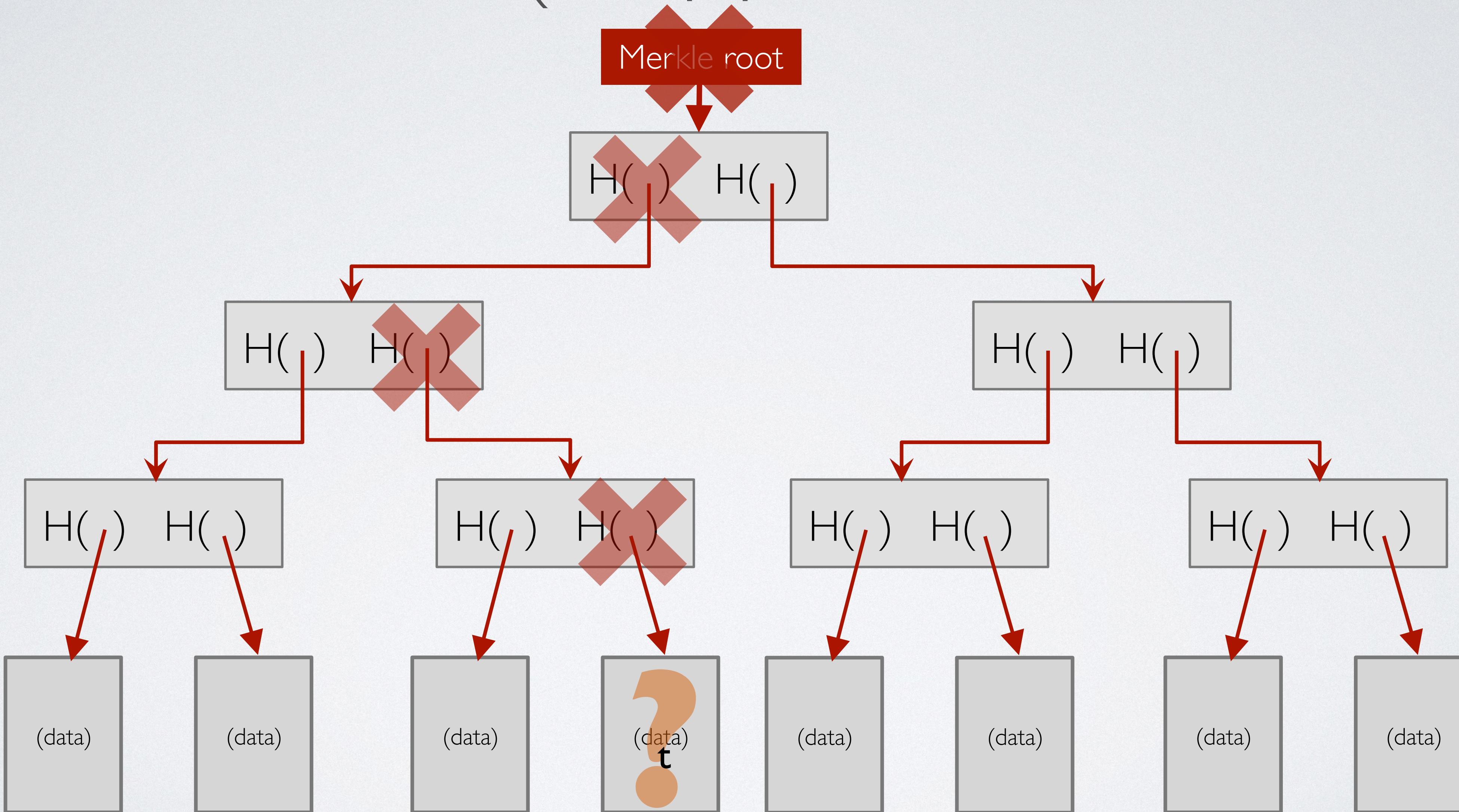
Подменяем информацию

ДЕРЕВО МЕРКЛА (СОДЕРЖИМОЕ БЛОКОВ)



А вот это уже кусочки блока — транзакции

ДЕРЕВО МЕРКЛА (СОДЕРЖИМОЕ БЛОКОВ)



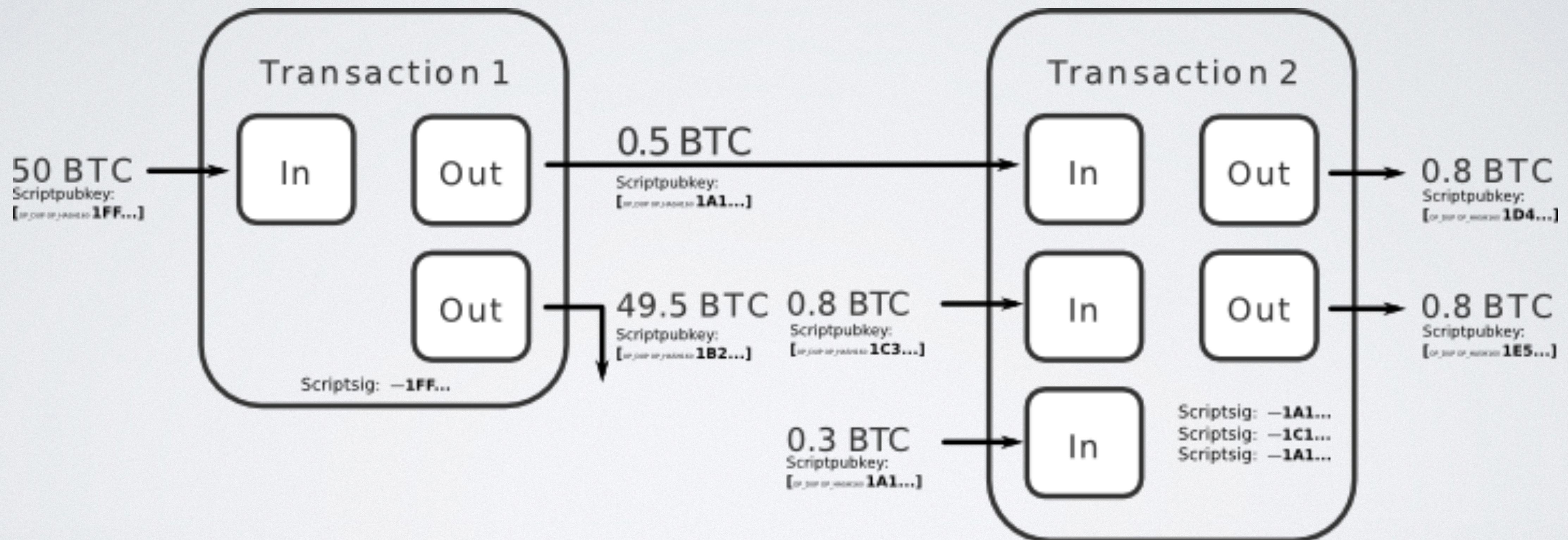
БЛОКЧЕЙН БИТКОИНА

- Основной элемент — транзакции (Алиса переводит Бобу биткоины);
- Для удобства транзакции объединяются в блоки;
- Последовательность блоков, содержащих транзакции и представляет собой блокчейн Биткоина.

БЛОКЧЕЙН БИТКОИНА

- Транзакции состоят из входов и выходов;
- Результатом транзакции являются выходы. Выходы можно рассматривать как «купюры», которые можно потратить.
- Когда мы тратим выходы, они являются входами в других транзакциях.

БЛОКЧЕЙН БИТКОИНА



АССИМЕТРИЧНОЕ ШИФРОВАНИЕ И ЭЦП

- Генерируем закрытый и соответствующий ему открытый ключ;
- Считаем специальным образом хеш открытого ключа — это биткоин-адрес. Пример адреса: `1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa`.
- Свой адрес или открытый ключ мы можем сообщить любому, чтобы нам могли перевести биткоины;
- Закрытый ключ мы храним в секрете, так как он используется для подписи сообщений при трате полученных ранее биткоинов.

ВХОДЫ И ВЫХОДЫ В ТРАНЗАКЦИЯХ

- Выход содержит в себе: сумму (**n** биткоинов) и специальный скрипт, который в стандартном случае говорит «*разрешаю потратить эти биткоины тому, кто предоставит валидную подпись для адреса **a***»;
- Вход — траты выхода — содержит в себе: информацию о том, какой выход **o** тратится и скрипт, который для стандартного случая говорит «*для этого выхода я предоставляю валидную подпись **s***».

СКРИПТЫ

- Скрипты позволяют задавать условия траты выхода;
- Описанный в предыдущем слайде и он же — наиболее часто используемый тип выхода в Биткоине — P2PKH (Pay-to-PubkeyHash) — «оплата в пользу хеша открытого ключа»;
- Но вообще можно создать и другие условия.

СКРИПТЫ

- Скриптовой язык в Биткоине — стековый (позволяет загружать переменные и проводить над ними простые операции) — работает слева направо;
- Операции называются «оп-кодами», например, **OP_EQUAL** сравнивает два последних элемента стека и возвращает **TRUE**, если они равны (**3 3 OP_EQUAL**);
- Для проверки валидности скрипт входа объединяется со скриптом выхода, и всё это вместе должно вернуть **TRUE**.

«РЕШИ УРАВНЕНИЕ, ЧТОБЫ ПОЛУЧИТЬ БИТКОИН»

- Создаём такой скрипт для выхода: **OP_ADD 12 OP_EQUAL**

«РЕШИ УРАВНЕНИЕ, ЧТОБЫ ПОЛУЧИТЬ БИТКОИН»

- Создаём такой скрипт для выхода: **OP_ADD 12 OP_EQUAL**
- Чтобы потратить такой выход, нам нужно создать такой скрипт, чтобы **<SCRIPT> OP_ADD 12 OP_EQUAL** вернул **TRUE**

«РЕШИ УРАВНЕНИЕ, ЧТОБЫ ПОЛУЧИТЬ БИТКОИН»

- Создаём такой скрипт для выхода: **OP_ADD 12 OP_EQUAL**
- Чтобы потратить такой выход, нам нужно создать такой скрипт, чтобы **<SCRIPT> OP_ADD 12 OP_EQUAL** вернул **TRUE**
- Решением будет **6 6**. Или **7 5**. В самом деле, **6 6 OP_ADD 12 OP_EQUAL** — две шестёрки складываются, в остатке получается **12 12 OP_EQUAL**, который даёт **TRUE**

СТАНДАРТНЫЙ СКРИПТ P2PKH

- Выход P2PKH: `OP_DUP OP_HASH160 <pubKeyHash>`
`OP_EQUALVERIFY OP_CHECKSIG`
- Для его траты требуется такой скрипт: `<sig> <pubKey>`

СТАНДАРТНЫЙ СКРИПТ P2PKH

| Описание (что делаем) | Скрипт (оставшийся) | Стек |
|---|---|---|
| Объединение скриптов выхода и входа | <sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG | (Пустой) |
| Константы добавляются в стек | OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG | <sig> <pubKey> |
| Последний элемент стека дублируется | OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG | <sig> <pubKey> <pubKey> |
| Последний элемент стека хешируется | <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG | <sig> <pubKey> <pubHashA> |
| В стек добавляется константа | OP_EQUALVERIFY OP_CHECKSIG | <sig> <pubKey> <pubHashA> <pubKeyHash> |
| Проверка равенства последних двух элементов | OP_CHECKSIG | <sig> <pubKey> |
| Проверка соответствия подписи открытому ключу | (Пустой) | TRUE |

ПОДЫТОЖИМ СТРУКТУРУ БЛОКЧЕЙНА

- Блоки состоят из множества транзакций;
- В транзакциях есть входы и выходы;
- Выход включает в себя сумму в биткоинах и скрипт, который указывает как его можно потратить;
- Вход включает в себя указание на выход, который мы хотим потратить и скрипт, который разрешает трату.

ПОДДЕРЖАНИЕ РАБОТЫ И
БЕЗОПАСНОСТИ БЛОКЧЕЙНА
БИТКОИНА

ДЕЦЕНТРАЛИЗАЦИЯ В КРИПТОВАЛЮТАХ

- Основные вопросы децентрализации протокола:
 - Кто обслуживает историю транзакций (блокчейн)?
 - Кто определяет какие транзакции валидны?
 - Кто создаёт новые монеты?
 - Кто определяет правила системы?
 - Как монеты получают свою стоимость?

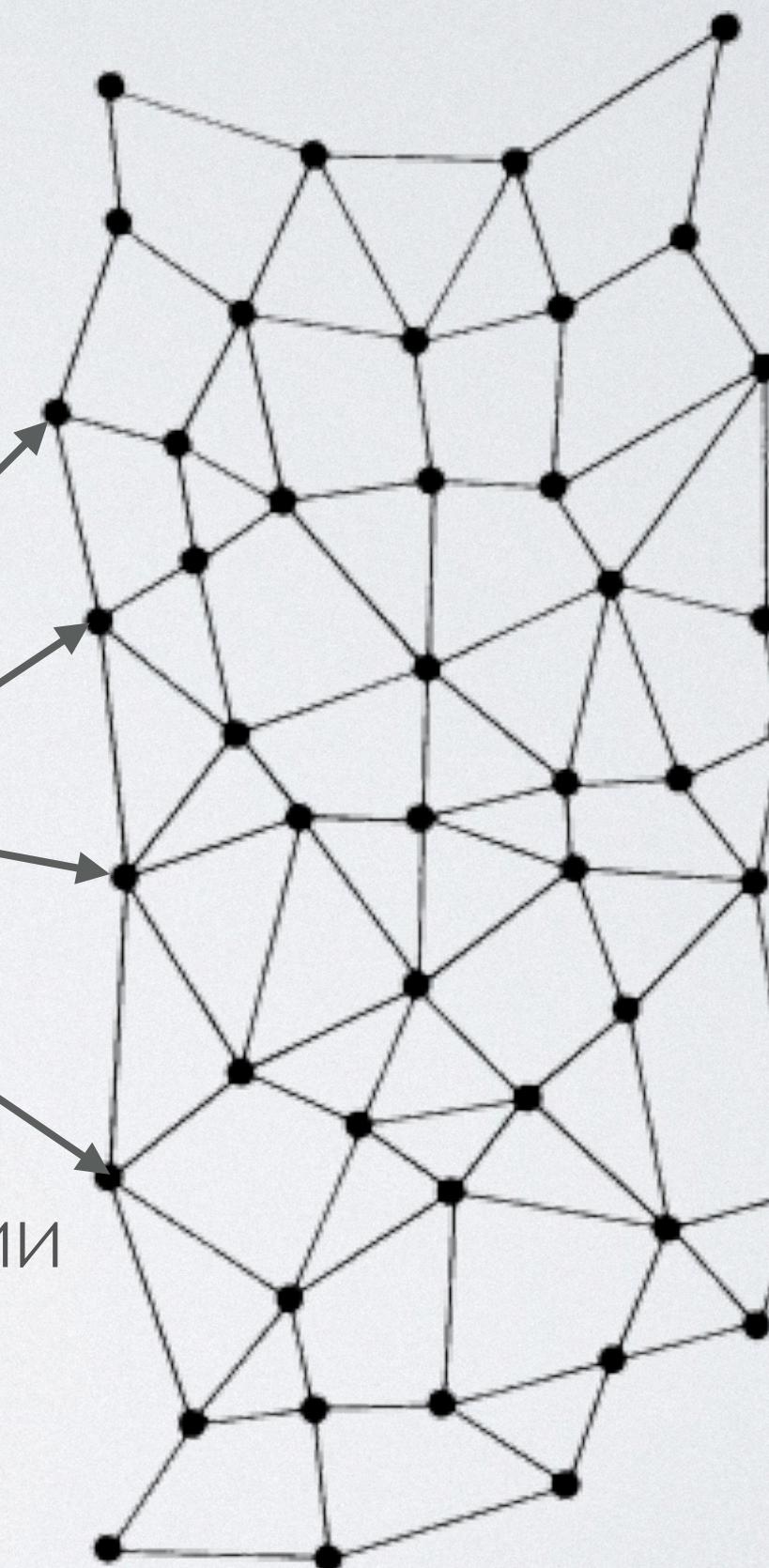
РАСПРЕДЕЛЁННЫЙ КОНСЕНСУС

- Когда Алиса хочет заплатить Бобу, она распространяет своё сообщение всем нодам:

P2P-СЕТЬ БИТКОИНА
Ноды — узлы сети

Подписано Алисой
Заплатить OK_{Боб} : H()

«Полные ноды» хранят все транзакции



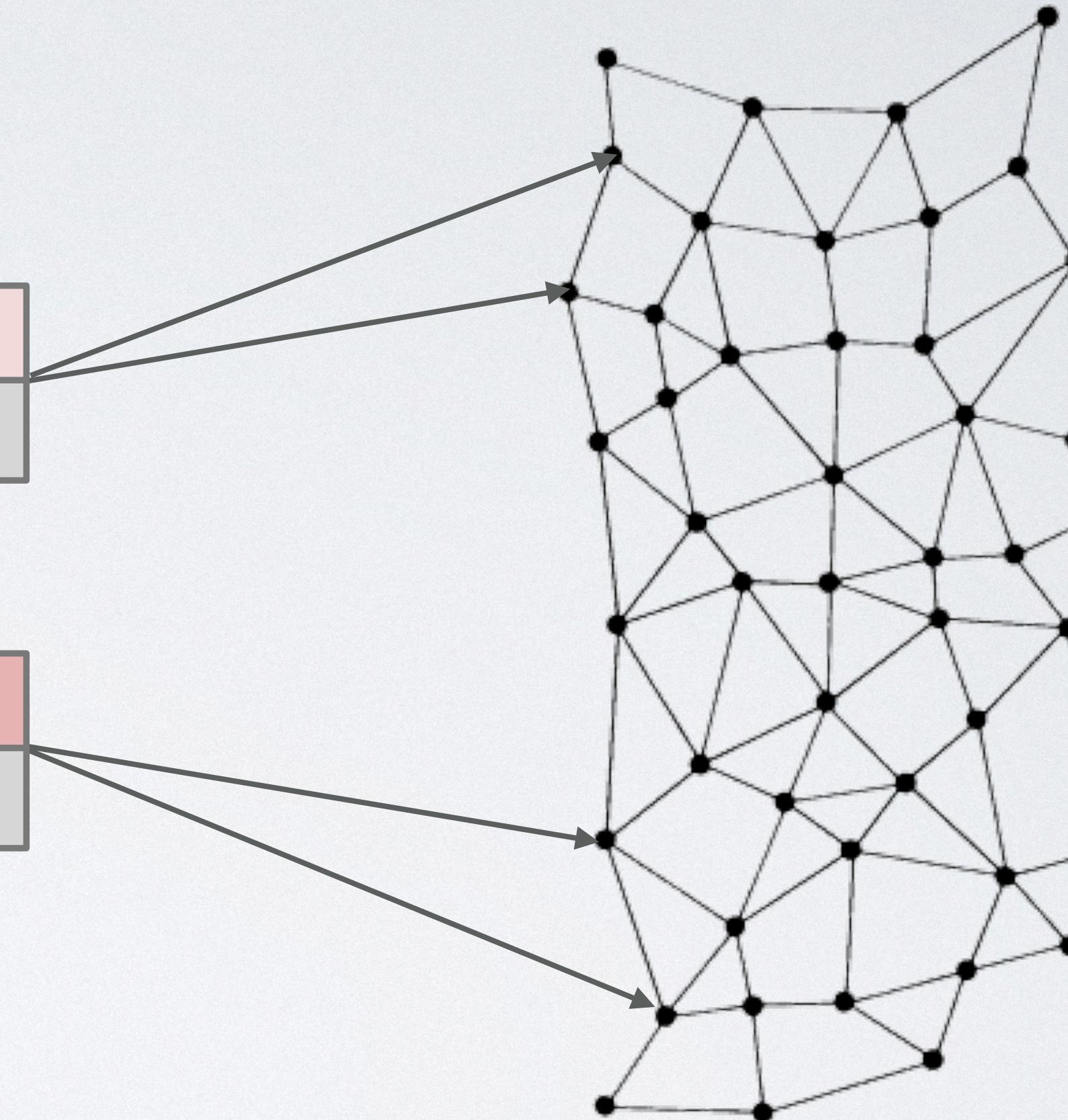
Примечание: ноды Боба может не быть в сети

РАСПРЕДЕЛЁННЫЙ КОНСЕНСУС

Двойная траты:

Подписано Алисой
Заплатить ОК_{Кэрол} : $H()$

Подписано Алисой
Заплатить ОК_{Боб} : $H()$



Если бы была 100% синхронизация времени и не было бы сетевых лагов, то можно было бы определить, какая транзакция была отправлена раньше.

РАСПРЕДЕЛЁННЫЙ КОНСЕНСУС

- Консенсус о чём должен быть достигнут:
 - Какие именно транзакции были распространены по сети;
 - В каком порядке это случилось.

РАСПРЕДЕЛЁННЫЙ КОНСЕНСУС

- Как консенсус работал бы в идеальном мире:
 - У каждого участника сети запущена нода;
 - Каждый участник имеет один голос в решении всех вопросов (в случае двойной траты — какую из трат всё же включать в блокчейн);
 - Все участники — честнейшие люди.

РАСПРЕДЕЛЁННЫЙ КОНСЕНСУС

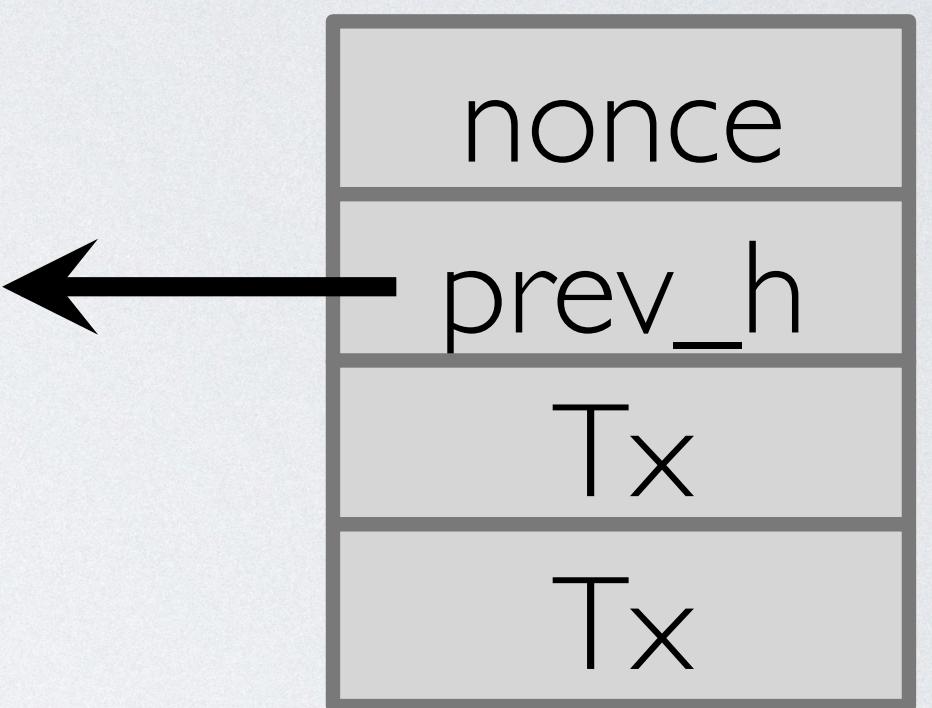
- Но мир не идеален:
 - Ноды могут прекращать работать;
 - Ноды могут быть злоумышленниками;
 - P2P-сеть не идеальна: не все ноды друг с другом соединены, сетевые ошибки, задержки, нет единого установленного времени;
 - И главное — один участник может создать миллион нод-злоумышленников.

PROOF OF WORK

- Чтобы (более-менее) справедливо посчитать «вес голоса» одного участника, можно делить веса пропорционально какому-то ограниченному ресурсу, имеющемуся у участников:
 - В Биткоине используется Proof of Work (PoW) — ограниченным ресурсом является мощность (из воздуха её не достать);
 - В других криптовалютах могут использоваться другие подходы, например Proof of Stake (ограниченным ресурсом являются сами токены криптовалюты).

PROOF OF WORK

- Proof of Work — математическая задача нахождения наименьшего хеша от нового блока:
 - Чтобы создать блок, надо найти такое значение *nonce*, чтобы $H(\text{nonce} // \text{prev_hash} // \text{tx} // \dots // \text{tx})$ было очень маленьким;
 - Если хеш-функция надёжна, то единственная возможность найти такое *nonce* — перепробовать их максимальное количество пока не повезёт.



PROOF OF WORK

Hello IFES! | 0: a94b9320a708ec88aa5cde0f5e42cb5250b8870192bcd9f6ba5f06244b6ebc4

Hello IFES! | 1: f62a7fa593f11cbc海棠da6072d0a18c2e3e53909cabb5bd4bf7249c1fb442c6

Hello IFES! | 2: 37d4f8edba4f1a2accab0e8a5206637f8ca3b38bdb50784940f97e34238ebab5

Hello IFES! | 3: f8d1b265a9094a7dac6db158333d59aea6e91337cc1ae2d83d5250509e490169

Hello IFES! | 4: df1b40733219654456284f5e7a6344c17d94843a4d4304837807e2b052a703cf

...

Hello IFES! | 356: **00**566090f02841ab6b89c6ea52542313eaf04b5560ff00810b1267674e183bd2

Hello IFES! | 420: **00**43b7c14df7424c8d824839b504d411ba80168b5d2fe9ab3472ad20c5ec0288

...

Hello IFES! | 1077: **000**2abe145d2192a067986f47429a60bdb5affdc6bd944ed21e5f8567f7cae91

Hello IFES! | 9705: **000**4f315ff687f820411cbd267fe9fd99a98979ddaffdbf134878377030de82

...

Hello IFES! | 66468: **0000**595be78fc31cce8fc1b2071a5556ce0d5f93eb68a135cf8d8e06d7825bf2

Hello IFES! | 68130: **0000**8a0a2e1c1b50e0cee23c2b4699e6ddfd03cc916e84115d667a7e56c1959d

...

Hello IFES! | 568863: **00000**33a19a6eb1f6452deac5f3954a8d10b9f618526e4134ae04b7915019119

Hello IFES! | 1306700: **00000**d47d1b42eab05d27686761d91821f0055180c9d6a61ee8064370b891eae

PROOF OF WORK

- Свойство №1: тяжело найти необходимое значение попсе
 - На октябрь 2016 каждую секунду просчитывается более 1.600.000.000.000.000.000 хешей (1.6 млрд Gh/s);
00000000000000004a987905ce4c9c3c78d952bace011bb2272c977f350c5b5
 - Обычный компьютер (CPU) имеет очень маленькую вероятность везения, поэтому многие ноды не участвуют в соревновании создания новых блоков, участвуют т.н. майнеры;

PROOF OF WORK

- Свойство №2: изменяемая сложность (*difficulty*)
 - В соответствии с правилами протокола целевое минимальное значение (*target*, «количество нулей») пересчитывается каждые 2 недели;
Hello IFES! | 356: **00**566090f0284|ab6b89c6ea525423|3eaf04b5560ff008|0b|267674e|83bd2 — цель «2»
Hello IFES! | 9705: **000**4f3|5ff687f8204||cbd267fe9fb99a98979ddaffdbf|34878377030de82 — цель «3»
Hello IFES! | 66468: **0000**595be78fc3|cce8fc|b207|a5556ce0d5f93eb68a|35cf8d8e06d7825bf2 — цель «4»
 - Цель: среднее время между блоками — 10 минут;
(нормальное распределение)
 - Вероятность найти блок равна доле хеш-мощности.

PROOF OF WORK

- Свойство №3: хоть блок и тяжело найти, необходимо, чтобы кто угодно (децентрализация!) мог легко проверить проделанную работу:
 - Значение *nonce* публикуется в заголовке блока;
 - Ноды проверяют, что
$$H(\text{nonce} // \text{prev_hash} // \text{tx} // \dots // \text{tx}) < \text{target}$$

ПРАКТИЧЕСКАЯ ЧАСТЬ №1 «ПАРСИНГ
БЛОКЧЕЙНА БИКТОИНА — СОЗДАНИЕ
ПРИМИТИВНОГО БЛОКЧЕЙН-ОБОЗРЕВАТЕЛЯ»

ПОСТАНОВКА ЗАДАЧИ

- Так как блокчейн Биткоина публичный — любому участнику сети доступна информация обо всех транзакциях начиная с 2009 года;
- Основная цель — научиться работать с историей транзакций (чтение блокчейна): на основе этого можно создавать различные сервисы или собирать статистику;
- В качестве первого простейшего лабораторного примера поставим себе задачу посчитать количество транзакций за последние сутки.

ЧТО НАМ НЕОБХОДИМО

- В первую очередь нам необходимо получить историю транзакций. Чтобы не доверять сторонним сервисам, самое универсальное решение — установить ПО полной ноды (Bitcoin Core) себе на компьютер (или на VPS). Объём блокчейна превышает 80 Гб, поэтому синхронизация может занять более дня.
- Рекомендация: можно использовать облачный сервис типа Digital Ocean или Linode — для поддержания ноды достаточно VPS-конфигурации с 2+ Гб оперативной памяти.

ЧТО НАМ НЕОБХОДИМО

- Перед первым запуском Bitcoin Core необходимо создать конфигурационный файл `bitcoin.conf` со специальными настройками (<https://github.com/Har0ld/workshop-hse-20161018/blob/master/bitcoin.conf>)
- Если Bitcoin Core уже был установлен, но настройки `txindex=1` не было, то надо перезапустить клиент с ключом `-reindex`

ЧТО НАМ НЕОБХОДИМО

- Есть два варианта дальнейшей работы:
 - Через GUI (если клиент установлен на компьютере) — удобно в качестве «песочницы», чтобы понять принципы работы;
 - Через RPC API (если клиент установлен на сервере без графической оболочки) — это будем использовать в нашем коде — с API будем общаться с помощью PHP.

ЧТО НАМ НЕОБХОДИМО

- Работа с API выглядит примерно так:

ДЕМОНСТРАЦИЯ

Код, используемый в демонстрации, доступен в репозитории по ссылке
<https://github.com/Har0ld/workshop-hse-20161018/tree/master/practice-1>

ДЕМОНСТРАЦИЯ

```
const DAEMON_PROTOCOL      = 'http';
const DAEMON_LOGIN         = 'user';
const DAEMON_PASSWORD      = 'password';
const DAEMON_HOST          = 'hse.blockchair.com';
const DAEMON_PORT          = '80';
```

ДЕМОНСТРАЦИЯ

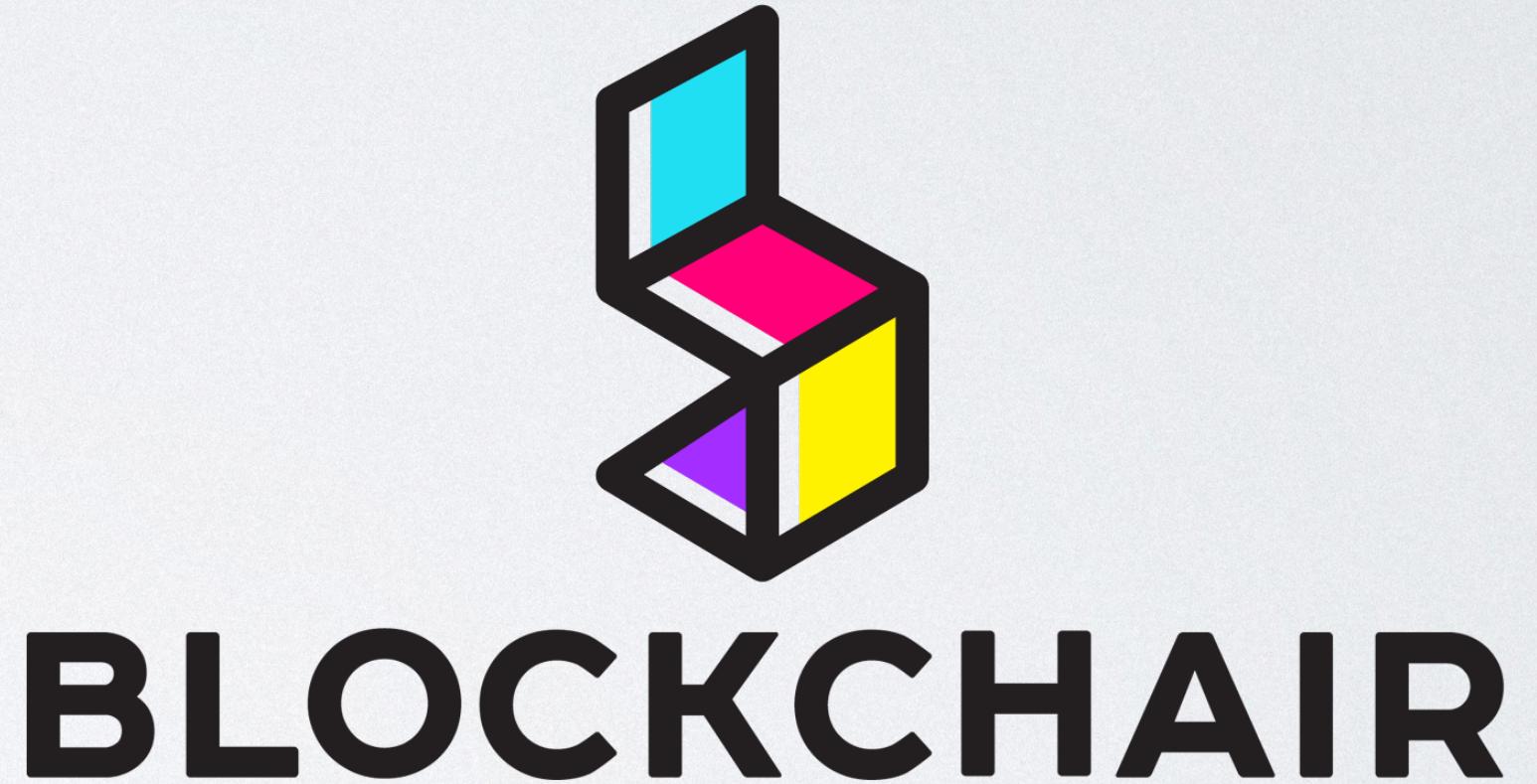
Код, используемый в демонстрации, доступен в репозитории по ссылке
<https://github.com/Har0ld/workshop-hse-20161018/tree/master/practice-1>

РЕЗУЛЬТАТ

- Научились настраивать полную ноду Биткоина;
- Научились к ней подключаться;
- Научились делать RPC-запросы к ноде;
- Создали прототип простейшего сервиса, который делает выборку информации из блокчейна по определённым критериям.

ПОИСК ПО БЛОКЧЕЙНУ

- На основе этого простого «кирпичика» можно строить и более сложные сервисы. В частности, такое же общение с нодой используется и в блокчейн-поисковике Blockchair (<https://blockchair.com/>)



ПОИСК ПО БЛОКЧЕЙНУ

The screenshot shows a web browser window titled "Blockchair" displaying the homepage of whiskey.blockchair.com. The page features a search bar with the placeholder "Search the Bitcoin blockchain for anything...". Below the search bar is a section titled "Examples:" with various search queries. At the bottom of the page are three large buttons labeled "Bitcoin / Blocks", "Bitcoin / Transactions", and "Bitcoin / Outputs", each with a corresponding chart below it.

Blockchair

https://whiskey.blockchair.com

Blockchair

Blockchair

Search the Bitcoin blockchain for anything...

Examples:

- Hello World
- The biggest blocks
- ~\$1M transactions
- 1dice8EMZmqKvrGE4Qc9bUFf9PX3xaYDp
- satoshi.id
- Blocks mined by Slush
- Tx paid more than \$0.50 in fees
- Addresses start with 1dice
- Marry me
- BIP 109 blocks
- Tx destroyed the most coindays
- Sequence < max(uint4)
- Happy Birthday
- Blocks collected more than 1 BTC in fees
- Bailout
- Coinbase transactions
- 123456

Bitcoin / Blocks

Average size (kB) by days and hours over last 3 months

Bitcoin / Transactions

Daily circulation (billions USD) over last 3 months

Bitcoin / Outputs

Percent of non-monetary outputs over last 3 months

ПОИСК ПО БЛОКЧЕЙНУ

Blockchair

https://whiskey.blockchair.com/bitcoin/transactions?q=time(2016-08-21..2016-09-20)&s=fee_usd(desc)

Blockchair

Blockchair

BLOCKCHAIR

Search the Bitcoin blockchain for anything... 🔍

Bitcoin / Transactions Time

| Block # | Id | Hash | Time | Inputs # | Outputs # | Output (BTC) | Output (USD) | Fee (BTC) | Fee (USD) | Fee/kB (USD) | Size (kB) |
|---------|---------------------------|---|------------------|----------|-----------|--------------|--------------|------------|-----------|--------------|-----------|
| 429736 | 155967997 | 22 ██████████ 1d | 2016-09-14 07:15 | 1 | 2 | 0.60000000 | 365.32 | 1.00000000 | 608.87 | 2,706.09 | 0.225 |
| 429453 | 155547467 | da ██████████ 36 | 2016-09-12 11:29 | 4 | 2 | 1.69170160 | 1,062.83 | 0.30030000 | 188.67 | 282.43 | 0.668 |
| 428577 | 154333090 | ea ██████████ ac | 2016-09-06 19:11 | 25 | 2 | 3.31299893 | 2,005.79 | 0.24267000 | 146.92 | 39.01 | 3.766 |
| 429124 | 155174076 | b2 ██████████ ef | 2016-09-10 11:45 | 37 | 2 | 6.82262147 | 4,270.62 | 0.20000000 | 125.19 | 18.64 | 6.718 |
| 428574 | 154326065 | a9 ██████████ 16 | 2016-09-06 18:04 | 11 | 2 | 10.17291893 | 6,158.99 | 0.19383000 | 117.35 | 69.15 | 1.697 |
| 430148 | 156590526 | 1d ██████████ ad | 2016-09-17 00:13 | 11 | 1 | 2.16893179 | 1,334.74 | 0.15000000 | 92.31 | 55.37 | 1.667 |
| 429484 | 155591691 | fd ██████████ 24 | 2016-09-12 16:28 | 275 | 2 | 10.01000003 | 6,288.88 | 0.10000000 | 62.83 | 1.55 | 40.645 |
| 429484 | 155591656 | 4e ██████████ b4 | 2016-09-12 16:28 | 232 | 2 | 10.01000011 | 6,288.88 | 0.10000000 | 62.83 | 1.83 | 34.302 |
| 429330 | 155415024 | 62 ██████████ 13 | 2016-09-11 18:25 | 547 | 1 | 48.66140000 | 30,484.91 | 0.10000000 | 62.65 | 0.78 | 80.707 |
| 429330 | 155414965 | 31 ██████████ fa | 2016-09-11 18:25 | 390 | 1 | 31.92250000 | 19,998.49 | 0.10000000 | 62.65 | 1.09 | 57.578 |

Load more (6,631,867 results left)... 👉

ПОИСК ПО БЛОКЧЕЙНУ

The screenshot shows a web browser window titled "Blockchair" displaying search results for the query "Hello World" on the Bitcoin blockchain. The URL in the address bar is <https://whiskey.blockchair.com/bitcoin/search?q=Hello+World>. The main content area shows two sections: "Coinbase data" and "Scripts".

Coinbase data:

- Block #359,179 (Mined by AntPool bj5B) - [Hello world](#) (Uno) - [blockchair.com/bitcoin/block/359179](#)

Scripts:

| Output # | Mined At | Script | Link |
|---------------|------------------|-----------------------------|--|
| 153,807,045:1 | 2016-09-04 09:14 | jEW hello world | blockchair.com/bitcoin/transaction/153807045#1 |
| 153,319,460:1 | 2016-09-01 22:06 | jEW hello world | blockchair.com/bitcoin/transaction/153319460#1 |
| 153,207,824:1 | 2016-09-01 12:06 | jMETAFACTORY: Hello world ! | blockchair.com/bitcoin/transaction/153207824#1 |

ПРАКТИЧЕСКАЯ ЧАСТЬ №2 «ЗАПИСЬ
ПРОИЗВОЛЬНОЙ ИНФОРМАЦИИ В БЛОКЧЕЙН
БИТКОИНА — НОТАРИЗАЦИЯ ДАННЫХ В
БЛОКЧЕЙНЕ»

ПОСТАНОВКА ЗАДАЧИ

- Если в предыдущей задаче мы хотели изучить существующие транзакции в блокчейне, то теперь нас интересует запись в блокчейн;
- GUI биткоин-кошельков позволяют создавать монетарные транзакции, но нет возможности записывать произвольную информацию в блокчейн.

ПОСТАНОВКА ЗАДАЧИ

- Самый простой пример — автор написал свой новый роман, но боится, что его кто-то украдёт!
- Что делают люди сейчас? Распечатывают роман и посылают его сами себе по почте заказным письмом с описью вложения. В случае спора о том «кто первый это написал» у автора есть это письмо со штемпелем с датой отправки.

ПОСТАНОВКА ЗАДАЧИ

- Что можно сделать с помощью блокчейна? Можно подсчитать криптографический хеш от файла с романом и записать его в блокчейн. Тогда в случае спора автору будет достаточно предоставить цифровой оригинал (который лучше до этого никому не показывать), а также запись в блокчейне (хеш оригинала сойдётся с записью, а у записи будет дата);
- Такой подход называется Proof of Existence.

ЧТО НАМ НЕОБХОДИМО?

- Таким образом, надо записать в блокчейн хеш (в общем случае — 64 hex-символа);
- Для записи немонетарной информации в Биткоине есть специальный тип выходов — *nulldata*;
- Проблема: чтобы провести транзакцию нужно заплатить комиссию!

ЧТО ТАКОЕ NULLDATA-ВЫХОД?

- Nulldata-выход — выход, содержащий скрипт следующего стандартного формата: **OP_RETURN <До 80 байт произвольной информации>**
- Его нельзя потратить, потому что использование кода **OP_RETURN** не позволяет создать суммарный скрипт, который вернёт **TRUE**.

ЧТО ТАКОЕ NULLDATA-ВЫХОД?

ЧТО ТАКОЕ NULLDATA-ВЫХОД?

> Scripts

Output #157,975,367:1

2016-09-23 05:55

j2EW Dipika, I love you. Will you marry me? -Sankalp

 blockchair.com/bitcoin/transaction/157975367#1

Output #150,310,584:1

2016-08-19 08:36

j!EW Angel will you marry me? -Alen

 blockchair.com/bitcoin/transaction/150310584#1

Output #108,368,120:1

2016-02-06 19:50

jWould you marry me Amélie?

 blockchair.com/bitcoin/transaction/108368120#1

Output #106,908,147:1

2016-01-30 22:18

jEW Would you marry me Amélie?

 blockchair.com/bitcoin/transaction/106908147#1

Output #85,889,662:1

2015-09-30 00:19

j EW Catherine, will you marry me?

 blockchair.com/bitcoin/transaction/85889662#1

Output #76,643,257:1

2015-07-18 06:05

j(Galitskiy:Will you marry me? Komolova:Da

 blockchair.com/bitcoin/transaction/76643257#1

Output #46,171,122:0

2014-09-07 12:23

jYuki will you marry me? Tetsu.

 blockchair.com/bitcoin/transaction/46171122#0



BLOCKCHAIR

ДЕМОНСТРАЦИЯ

Код, используемый в демонстрации, доступен в репозитории по ссылке
<https://github.com/Har0ld/workshop-hse-20161018/tree/master/practice-2>

РЕЗУЛЬТАТ

- Научились делать RPC-запросы, которые меняют состояние блокчейна и кошелька;
- Открыли для себя мир готовых решений (зачем писать код самому, когда его уже кто-то написал);
- Создали прототип простейшего сервиса, который реализует Proof of Existence.

Спасибо за внимание!