Блокчейн для разработчиков

Предисловие

Эта книга написана для тех, кто хочет попробовать себя в разработке децентрализованных приложений. Она будет интересна как разработчикам, так и всем тем, кто принимает непосредственное участие в подобных проектах: руководителям, инвесторам, проджект менеджерам.

В книге не будет описания разработки самого блокчейна, так как в большинстве случаев не имеет смысл создавать собственный блокчейн под приложение. Но я разберу то, как устроены существующие блокчейны, их принципы работы, особенности, плюсы и минусы, для того чтобы вы могли принять решение о необходимости использования блокчейна для своего проекта.

Мы рассмотрим, как создать приложение на блокчейне Ethereum с использованием фреймворка Truffle. Узнаем, что надо предпринять до того, как Вы начали писать приложение, затронем юридические аспекты криптовалют, определим те технологии, которые надо учитывать при написании приложения для того, чтобы оно хорошо работало.

Также я расскажу об инструментах, используемых для создания приложений. На примере простейшего приложения для голосования объясню, как написать смарт-контракт, протестировать его, а также опубликовать в блокчейне.

Введение

Что такое блокчейн?

Технология блокчейн последнее время наделала много шума. Эксперты пророчат ей великое будущее, проекты на блокчейне собирают миллионы долларов, изобретение блокчейна сравнивают с изобретением Интернета. Даже если вы никогда не слышали об этой технологии, то о Биткойне вы слышали 100%. В основе криптовалюты Биткойн как раз и лежит технология блокчейн. Попробуем разобраться, что же это такое.

Говоря сухим научным языком, блокчейну можно дать следующее определение.

*Блокчейн – это темпоральная (хранящая с самого начала) информационная система нового типа с управлением бизнес-логикой на основе смарт-контрактов, содержащая факты, называемые блоками, реплицированная на несколько компьютеров, собранных в одноранговую сеть,* *обеспечивающая нефальсифицируемое хранение данных и защищенное от несанкционированного вмешательства преобразование этих данных. Участниками сети являются анонимные лица, называемые узлами. Каждый блок содержит криптографический хеш предыдущего блока, метку времени и данные транзакции. Добавление фактов в базу происходит только на основании консенсуса. Данные из блокчейна удалить нельзя. Главное в блокчейне — это не цепочка блоков, а программные контроллеры — смарт-контракты, которые автоматически и однозначно производят действия с данными. Именно такой контроллер обеспечивает генерацию и сохранение данных блокчейн-сети — создание новых записей и обмен ими посредством сетевых транзакций.*

Сложно для понимания, но к концу книги вы поймете, о чем речь.

Децентрализованные сети, которыми, по сути, и являются блокчейны, не новы в мире информационных технологий. Создание децентрализованных сетей началось намного раньше, чем вся эта шумиха с криптовалютами. Достаточно вспомнить о таких децентрализованных одноранговых сетях (их еще называют пиринговыми или P2P сетями), как BitTorrent или Napster, участники которых использовали сеть для обмена фильмами. Так в чем же отличие и ценность блокчейна? В чем была необходимость его создания? Какую проблему он должен решить? И нужен ли вообще блокчейн для вашего проекта? Попробуем разобраться.

Часть 1. Теория

В этой части я расскажу о том, как возник блокчейн, как он работает и где его можно использовать. Я помогу вам определиться, действительно ли для вашего проекта необходима блокчейн технология или же можно обойтись известными проверенными альтернативными решениями. Помните, что неграмотное или неуместное использование блокчейна скорее оттолкнет от вашего проекта, чем привлечет к нему внимание.

Глава 1. Значимость технологии блокчейн для современного мира. Основные понятия.

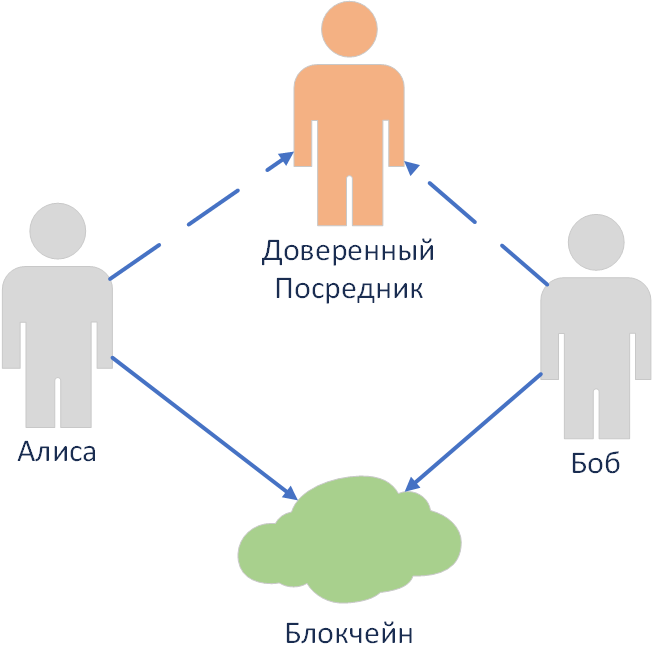
Современная экономика переживает трансформацию и постепенно уходит из реального мира в цифровой. Данный процесс принято назвать цифровой трансформацией. Этот процесс непрерывно связан с оптимизацией существующих решений, разработкой новых, исправлением ошибок, внедрением доработок. Роль человека в цифровой экономике минимизируется, многие процессы автоматизируются. Для совершения трансформации разрабатываются и применяются новые информационные технологии. Все они обладают определенной значимостью и новизной, но не все являются революционными.

Что же такое революционная или значимая технология? К значимым технологиям относят те технологии, которые привносят принципиально новый этап развития нескольких отраслей. Например, электричество позволило развивать промышленность в десятки раз быстрее, чем это было ранее, позволило появиться тысячам изобретений, которые без электричества просто были бы бессмысленны или нереальны. К значимой технологии относится Интернет. С его появлением началось бурное развитие дистанционных технологий, обмен информацией стал производиться практически мгновенно, у людей появилась возможность общения со всем миром.

С развитием Интернет возникли проблемы доверия между сторонами. Эту ситуацию замечательно характеризует известная картинка о двух собаках перед компьютером, где одна из них говорит другой: «В Интернете никто не знает, что ты – собака». Анонимность и отсутствие доверия сдерживает развитие многих процессов, необходимых во множестве отраслей

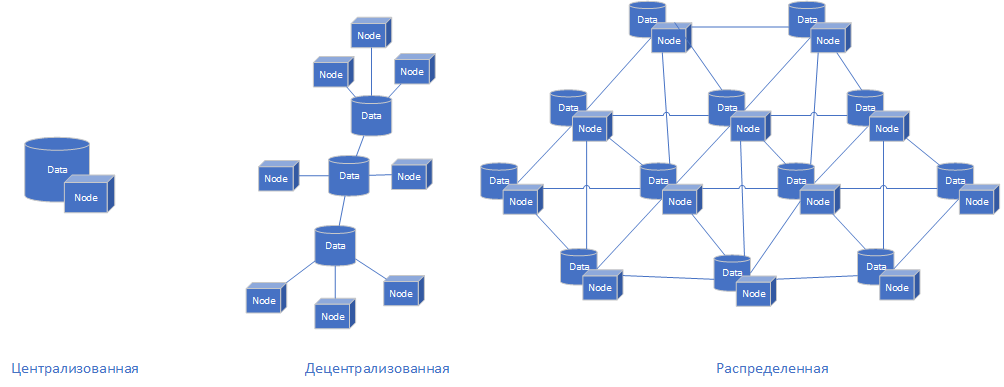
[](https://www.google.ru/url?sa=i&rct=j&q=&esrc=s&source=imgres&cd=&cad=rja&uact=8&ved=2ahUKEwis4ZX43KHfAhWLCCwKHXT4BhMQjRx6BAgBEAU&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FOn_the_Internet%2C_nobody_knows_you%2527re_a_dog&psig=AOvVaw1FLpItVKS_iBJ3nqRrMetL&ust=1544959541502708)

Перед человечеством стоит задача создания доверенной системы управления ценностями и блокчейн является одной из технологий, с помощью которой пытаются ее решить.



Итак, в чем же новшество технологии блокчейн и какие задачи она решает?

Обратимся к истории. Развитие информационных систем происходило от централизованных к децентрализованным и далее к распределенным.



Основным признаком централизованной системы, ярким примером которой в настоящее время является платежная система SWIFT, является единый сервер (единое место хранения, обработки и записи данных). Рассмотрим пример транзакций в централизованной системе. Если Алиса хочет отправить Бобу деньги, она инициирует банковский перевод. Банк Алисы ведет реестр всех исходящих и входящих транзакций, как и банк Боба. Как только Алиса инициирует транзакцию, ее банк проверит дебет средств в своем бухгалтерском реестре и отправит средства в банк Боба. Банк Боба в свою очередь проверит поступление денежных средств путем обновления своего собственного бухгалтерского реестра. Оба банка поддерживают свои реестры независимо друг от друга. Алиса обычно платит комиссию за то, чтобы банк осуществил этот перевод, и в зависимости от того, где живет Боб, время на перевод может занять от 3 до 10 (а иногда и больше) рабочих дней.

Изображение выглядит как электроника

Автоматически созданное описание

Помимо большого количества действий и времени осуществления транзакции при использовании централизованных систем возникают проблемы блокировки и подмены данных. Проблема блокировки может быть решена за счет распределенности.

Изображение выглядит как электроника

Автоматически созданное описание

Известным образцом распределенных систем являются сети Р2Р, в которых отсутствуют центральные сервера, а вся информация хранится на узлах. Сети P2P, как и другие распределенные системы, столкнулись с решением сложной проблемы, которая отсутствовала в реляционных базах данных: решение конфликтов и согласованность. Если в сети одновременно возникают два несовместимых факта, система должна иметь четкий алгоритм, какой факт признать верным.

Ярким примером несовместимых фактов является проблема двойной траты

Изображение выглядит как цепь, металлоизделия

Автоматически созданное описание

У Алисы есть всего $10. Алиса пытается отправить эти $10 двум получателям одновременно. Как система должна определить, кому достанутся деньги и должна ли она провести транзакцию по обоим получателям? Для того, чтобы ответить на этот вопрос, необходимо упорядочить факты. Если в сеть поступают два несовместимых факта, то выигрывать должен тот факт, который поступил первым. Но что же делать, если факты отправлены практически одновременно на отдаленных узлах сети? Как можно добиться согласованности всей сети с первым фактом? Именно этот вопрос был решен применением *алгоритма консенсуса* в распределенных сетях. Алгоритмы консенсуса необходимы для обеспечения достоверности транзакций через проверку точности приведенных действий.

*Ценность блокчейн-сетей в их уникальной функциональности. Они обеспечивают не только максимальную защиту уже записанных в блоки данных, но и гарантируют однозначность выполнения операций с этими данными независимо от намерений пользователей.* Если не использовать данную функциональность в полной мере, то намного проще и дешевле создать обычную базу данных (систему для записи и чтения информации) с необходимыми свойствами: темпоральность, криптозащита от фальсификации, гарантия сохранности. Для этого необходимо подписать данные электронной подписью и поместить их в децентрализованное хранилище с правами «read only». Для создания таких типов баз данных блокчейн не нужен.

Как же отнеслось общество к появлению новой технологии? Несмотря на то, что блокчейн криптовалюты Биткойн был запущен в 2009 году, многие не обращали на него внимание почти пять лет. Часто блокчейн связывают только с миром криптовалют, забывая о его характеристиках и преимуществах.

Главным преимуществом блокчейна является то, что с его помощью можно исключить бюрократическую волокиту и автоматизировать множество бумажных процессов, увеличить доверие между сторонами без привлечения третьих лиц.

В целом отношение к блокчейну со стороны общества можно назвать настороженно-восторженным. Происходит это по следующим причинам:

1. Однажды созданную сеть невозможно взять под контроль властей, ее невозможно изменить или заблокировать/закрыть. Люди доверяют независимым системам, хотя в большинстве случаев не представляют, как эти системы работают.
2. Большинство людей в мире живут с нестабильными политическими и финансовыми системами, с дорогими финансовыми услугами низкого качества и шаткой национальной валютой. Использование криптовалют сделало возможным международные торговые сделки в таких странах, дало уверенность и чувство финансовой защищенности.
3. Несмотря на то, что объем криптовалют составляет примерно 1% объема мировых денег, наблюдается устойчивая тенденция роста как объемов криптовалют, так и их курса.

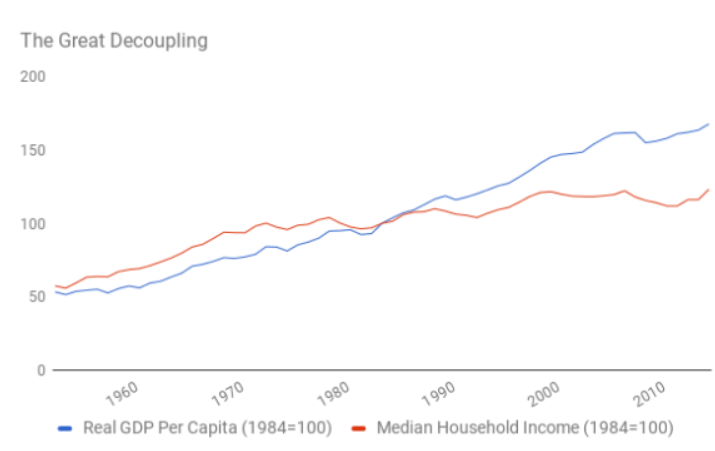
Итак, какие же изменения ждать от применения блокчейна?

Многие исследователи блокчейна пророчат его глобальное распространение. С ростом сетевого эффекта появятся многочисленные слои децентрализованных приложений, надстроенных на блокчейнах Биткойна, Ethereum и других криптовалют, и всему миру откроются невиданные ранее возможности для бизнеса и денежных потоков. Деньги будут передаваться, как чистая информация — мгновенно, безопасно, чрезвычайно дёшево, открыто, программируемо. Технология блокчейн будет использоваться вместо известных систем учета, на ее основе будут работать протоколы криптовалют, изменяющие существующие финансовые системы.

Реформирование рынков капитала станет первым шагом в мировой экономике для технологии блокчейн по словам многих аналитиков. Исследователи полагают, что после падения ажиотажа и снижения количества спекулятивных операций на рынке, постепенно произойдет демократизация криптовалют и переход к «не-банковской» экономике, экономика выйдет из «кредитной эпохи», которая продолжалась за счет увеличения государственных балансов и активов, необходимых для ежегодного рост ВВП. На рисунке ниже обозначены стадии развития криптовалют.

Чтобы понять природу криптовалют необходимо обратиться к истории и вспомнить как происходило развитие обычных валют. Веками у денег было две основные функции. С одной стороны, деньги — материальное образование, а с другой — они содержат совокупность свойств и связей, присущих экономическим отношениям. Рабочий класс, как правило, использует деньги для выживания, финансовый класс – для экономической ренты, как средство кредитования рабочего класса. Вспомните фразу, сказанную Дж.П.Морганом в своей знаменитой речи перед американским Конгрессом 19 декабря 1912г.: «Золото – это деньги и ничего более. Все остальное – кредит». При такой модели обращения денег рабочий класс хочет поддерживать низкие процентные ставки по кредитам, чтобы позволить себе обслуживать свои долги, в то время как финансовый класс хочет получить максимальную отдачу от своего капитала. В свою очередь любая валюта существует по закону Грэшама, заменяясь с течением времени худшей валютой. Яркий тому пример – переход от золотых монет к бумажным, а далее к электронным деньгам. Как известно, многие национальные валюты не обеспечены ничем и являются просто байтами в балансе, отражающие дебет и кредит. Кредит существует уже тысячелетия с тех пор, как первые фермеры расширили его в форме долговых расписок между циклами уборки урожая. Цикл кредит / долг создается каждый раз, когда человек или страна занимает деньги, и, как и любой цикл, он получает обороты с каждым завершением. Банки зарабатывают свои деньги в пределах разницы между ставкой, которую они берут в долг, и ставкой, по которой они их ссужают, известной как спред процентной ставки, и они усилили эффект взлетов и падений делового цикла, расширив кредит и расширив «деньги» в системе в хорошие времена и заключив контракт в плохие времена.

Джон Мейнард Кейнс считает вырождением существующую модель кредитования, поясняя, что система фиатных кредитов разработана с целью увеличения экономического пирога, который никогда не будет употреблен в пищу. Другими словами, мы не едим нашу долю «пирога», а постоянно вкладываем ее в остальную часть коллективного микса, а когда нам это действительно нужно, мы берем малюсенький кусок, вместо полагающегося, поскольку все позже и позже выходим на пенсию.



Источник: World After Capital

Разрушение традиционной корреляции между доходом США и ростом ВВП примерно в 1986 году было названо «Великой развязкой». Этот «рост» ВВП является эвфемизмом для долга. Рост происходит лишь потому, что люди берут на себя больше кредитов, чтобы обслуживать старые кредиты и иметь возможность покупать больше, чем они могут себе позволить. До недавнего времени Ирландия удерживала мантию первенства для домохозяйств с наибольшей задолженностью на уровне более 115% ВВП. Это было вызвано бумом и последующим спадом на рынке недвижимости. Однако, недавно Австралия опередила Ирландию благодаря долгу домохозяйств, превышающему 120% ВВП, что опять-таки зависело от спекуляцией недвижимостью и самых мягких условий кредитования в течение длительного периода. Эта ситуация запустила ускоренную денежную эмиссию. Например, эмиссия в США составляет порядка 1 трлн. долл. в год. Подобное расширение денежно-кредитной базы произошло во всем мире. Общемировой непогашенный долг составляет 250 триллионов долларов США и в четыре раза превышает сумму неоплаченных обязательств, не говоря уже об огромном количестве запутанных финансовых деривативов, примерно такого же размера, как эти долги и обязательства вместе взятые.

Благодаря криптовалютам у человечества появится возможность уйти от дуалистической природы денег, обеспечить деньги реальными активами, избавиться от раздутости и запутанности. Криптовалюты и блокчейн помогут осуществить переход к децентрализованной модели экономики. Возьмите, для примера Uber и AirBnB, чья децентрализованная модель хоть и не является идеальной, но стала первым шагом к развитию таких организаций. Мир медленно но верно переходит от модели владения собственностью к модели пользования коллективным, что в дальнейшем позволит практически отказаться от кредитной гонки. В будущем нас ждет экосистема множества криптовалют для приложений, существующих на цифровых суверенных децентрализованных платформах в Интернете и находящихся вне юрисдикции правительств и центральных банков. Эти криптовалюты уменьшат влияние депозитных ставок на нашу жизнь и еще более сократят маржу коммерческих банков вплоть до применения отрицательных депозитных ставок, т.к. банки тратят деньги на хранение денег, как это ни странно звучит. Век высоких депозитных ставок прошел и больше не повторится. Уже сейчас отмечается резкое снижение накоплений и пользования депозитными ставками среди населения. Молодое же поколение, согласно опросам, вообще не видит смысл замораживать деньги на несколько лет под столь низкий процент. Мало кто готов отказаться от части дохода за перспективу иметь порядка 1% прибыли с учетом инфляции.



Процентная ставка Десятилетнего казначейства (показанная TNX ETF), которая оказывает наибольшее влияние на краткосрочные процентные ставки в банках, в течение десятилетий находилась в состоянии нисходящего тренда.

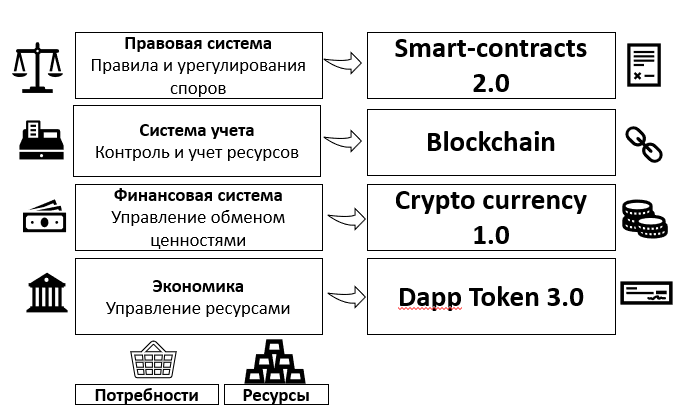
Интернет впервые в истории создал возможность конечного производства с нулевыми затратами - товаров, которые не требуют персонала или накладных расходов для производства. Это будет иметь огромное дефляционное влияние на дальнейшее развитие мира. Очевидно, что необходимо предпринимать какие-либо шаги, иначе мир столкнется с коллапсом банковской системы.

Идея альтернативных валют не нова. Они широко использовались в США и Европе во время Великой депрессии, когда банки страны иссякли и местные органы власти и юрисдикции выпустили собственную альтернативную валюту.

Самая старая альтернативная валюта, которая все еще существует с 1930-х годов и используется в комбинации со швейцарским франком для создания двойной валютной операции, - это швейцарская WIR. WIR-банк — швейцарский банк, эмитент независимой дополнительной валюты в Швейцарии, обслуживающий предприятия в сферах гостиничного бизнеса, строительства, производства, розничной торговли и профессиональных услуг. Валюта WIR существует только в безналичной форме и обращается внутри сообщества (кооператива) WIR. WIR-банк являлся некоммерческим юридическим лицом, однако этот статус изменился после расширения банка. Банк рассматривает себя как стабильный, утверждая, что является полностью работоспособным в периоды экономического кризиса и может ослабить спады в экономическом цикле, помогая стабилизировать Швейцарскую экономику в трудные времена.

В свете увеличения долговых обязательств, последующего спада в экономике и возможностями децентрализации, которую обеспечил Интернет, а также рабочими моделями локальных валют и выявленной жизнеспособности частично децентрализованных компаний, стало возможным осуществить переход к новой цифровой децентрализованной экономике.

Что она будет из себя представлять? Ниже представлены элементы системы, которые придут на смену классическим существующим.



Криптовалюты будут контролироваться децентрализованным сообществом, их эмиссия будет происходить в сетях, где нет центрального доверенного узла и которые исключают вмешательство в эмиссию какого-либо государства. В связи с тем, что передача криптовалют необратима — никто не сможет принудительно (без приватного ключа) совершить транзакцию, отменить, заблокировать или оспорить ее использование в экономике и финансовом секторе сможет предотвратить мошенничество и внешнее вмешательство в сделки.

Смарт-контракты или Блокчейн 2.0 – специальные компьютерные протоколы, позволяющие выполнять надежные транзакции без третьих сторон - помогут четко сформулировать правила взаимодействия, правила взаиморасчетов и практически решат проблему урегулирования споров т.е. станут основой правовой системы.

Управление ресурсами будет осуществляться с помощью децентрализованных приложений (DApp), использующих электронную подпись и дающих доступ к управлению *токенами*. **Токен** — это единица учета, которая используется для представления цифрового баланса в некотором активе, заменяющая традиционные ценные бумаги.

Многие крупные ИТ-компании уже разработали собственные блокчейны и решения. Успешные внедрения решений продемонстрировала компания IBM: ее решения на базе блокчейна внедрены в FedEx (система отслеживания грузов), совместно с Walmart IBM создала блокчейн-альянс продуктовой безопасности. Также разработаны самостоятельные решения не требующие внедрения в структуру компании.

Финансирование блокчейн проектов часто ведется через ICO (Initial Coin Offering) – выпуск проектом купонов (токенов) предназначенных для оплаты услуг площадки с использованием криптовалюты в будущем. Документы в рамках проведения ICO представляют собой публичную оферту, к которой присоединяются лица, приобретающие токены. И как публичная оферта определяет «правила игры» между владельцами различных интернет-сервисов и их пользователей, так и указанные документы определяют правила игры в каждом конкретном ICO. Исходя из вышеуказанных документов, например WhitePaper, определяется правовая природа токенов: токен как расчетная единица, токен как инвестиционный инструмент, токен как займ, токен как инструмент благотворительности (в случае если токен не приносит дивидендов и держатель не вправе требовать у эмитента обратного выкупа). В отличие от IPO, покупатели валюты не получают доли в компании и никак не могут воздействовать на внутренние управленческие решения, если это не было оговорено в изначальных условиях. Такими условиями может являться децентрализованное эскроу, при котором сами владельцы токенов большинством голосов управляют перераспределением токенов: принимают и исполняют решения о продолжении финансирования проекта, его прекращении, возврате инвестиций и прочие решения по управлению активами. На самом деле ICO —это ещё одна реализация модели краудфандинга, когда участники финансируют развитие компании сейчас для того, чтобы получить от неё какие-то блага в будущем. В разных странах регулирование ICO происходит по-разному.

Власти **США** с недавних пор приравнивают ICO к первому публичному размещению традиционных акций — IPO. Соответственно, к ICO применяется федеральное законодательство о ценных бумагах, а урегулированием всех вопросов связанных с ICO занимается Комиссия по ценным бумагам и биржам США (SEC). SEC официально заявил, что любая знаменитость или другое лицо, которое продвигает токены, имеющие статус ценной бумаги, должны раскрывать характер, объем и размер компенсации, полученные в обмен на продвижение. Отказ раскрыть эту информацию является нарушением положений федеральных законов о ценных бумагах.

В отношении ICO позиция **Канады** совпадает с позицией США. Вопросом регулирования ICO занимается Канадская администрация по ценным бумагам (CSA), которой подчиняется сразу несколько государственных органов, так как в каждой из шести провинций Канады действует свой финансовый регулятор.

**Сингапур** не только легализовал токены, но и активно поддерживает технологические проекты, ищущие финансирование через ICO. Денежно-кредитное управление Сингапура (MAS) разъяснило свою позицию касательно регулирования ICO, приравняв выпускаемые токены к ценным бумагам. MAS также пообещал внести поправки к правилам о торгах, согласно которым оно сможет выступать в качестве регулятора при некоторых первичных размещениях токенов.

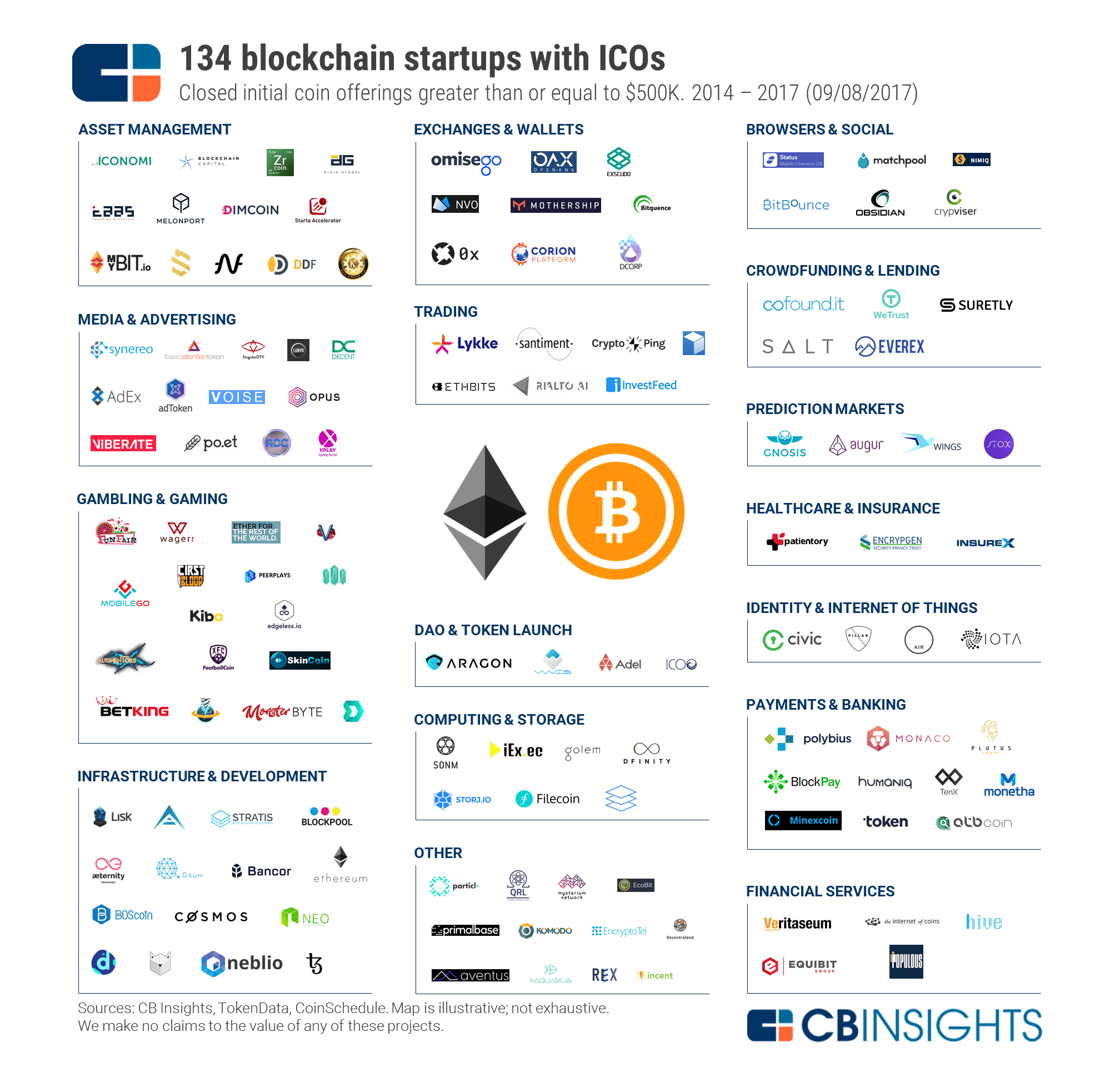
Вопросами регулирования ICO на территории **Швейцарии** занимается Швейцарская служба по надзору за финансовыми рынками (FINMA). Несмотря на то, что на данный момент законов о привлечении инвестиций через ICO в этой стране нет, из-за специфических характеристик определенных ICO могут применяться различные требования действующего законодательства.

**Китай** стал первой страной, объявившей ICO вне закона. В сентябре 2017 года власти КНР потребовали компании, которые уже провели или проводят ICO, остановить всю деятельность и вернуть средства вкладчикам. Запрет связан с повышенными рисками, так как ICO могут быть связаны с аферами и финансовыми пирамидами. Кроме этого, в Китае под запрет попала и деятельность криптовалютных бирж.

Вслед за Китаем о запрете привлечения инвестиций через ICO сообщили власти **Южной Кореи**. В конце сентября Комиссия по финансовым услугам Южной Кореи (FSC) заявила, что ICO повышает риск возникновения финансового мошенничества, в связи с чем размещение и покупка токенов на территории страны стала нелегальной.

Власти пообещали применять жесткие штрафные санкции в отношении любой компании или лица, выпускающего новые криптовалюты. При этом FSC заявляет, что торговля существующими криптовалютами должна жестко регулироваться властями.

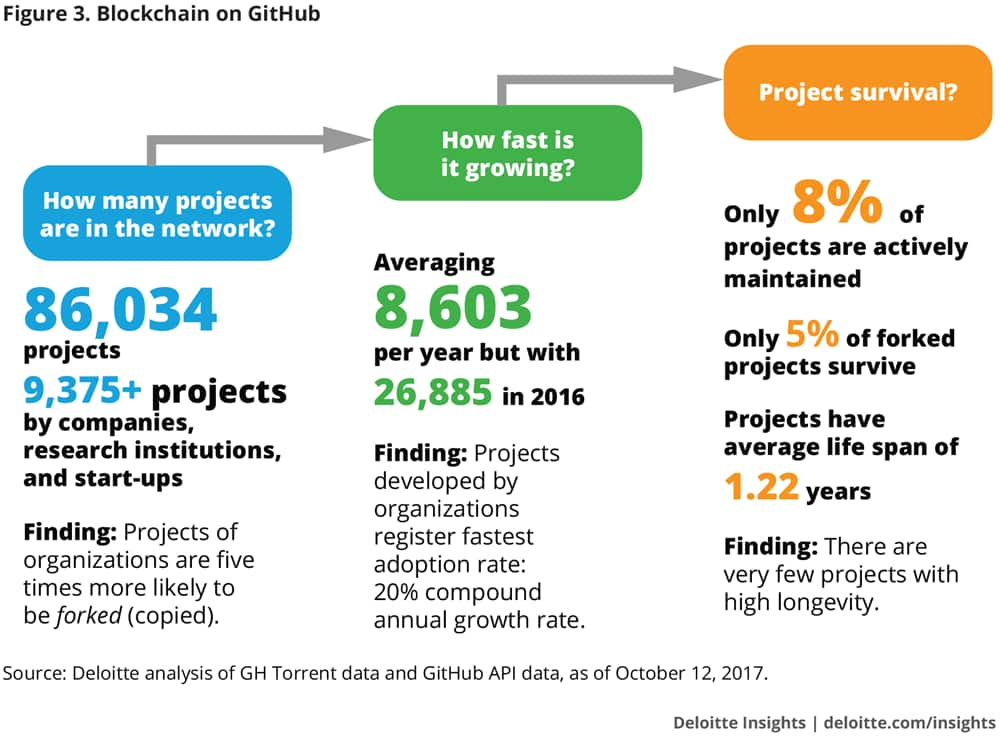
Несмотря на диаметрально противоположное отношение к ICO в разных странах, отсутствие законодательного регулирования, проекты на блокчейне вызывают повышенный интерес из-за широты применения технологии и ее очевидных преимуществ. На рисунке изображены наиболее известные проекты, прошедшие ICO.



https://www.cbinsights.com/research/report/blockchain-trends-opportunities/

*ICO не является обязательным этапом для проектов на блокчейне. Это всего лишь краудфандинговая компания.*

Единичные проекты на блокчейне стали появляться с 2014 года. Начиная 2016 года наблюдается бурный рост их количества. Однако большинство проектов оказывается нежизнеспособными по ряду причин: непродуманность проекта, невостребованность проекта рынком, изначальное желание собрать деньги, не развивая проект и т.д. Выживаемость проектов на блокчейне пока менее 10%. Как отмечает криптовалютный аналитик Кевин Рук 94 из 100 лучших проектов blockchain запускаются поверх сети Ethereum. Альтернативные blockchain-протоколы, намеренные составить серьезную конкуренцию Ethereum, развивают собственные децентрализованные сети. Как утверждают, например, представители проекта EOS, созданного компанией Block.one, они намерены изменить статус Ethereum в качестве платформы DApp для разработчиков. Однако EOS, также основан на сети Ethereum. В то же время NEO, Cardano и Qtum — это децентрализованные приложения, созданные с нуля, с реальными перспективами претендовать на роль ведущих blockchain-протоколов.



*Для того, чтобы проект стал успешным, перед началом разработки проекта необходимо иметь четкое представление о том, в какой отрасли проект найдет свое применение. Необходимо определить тот круг лиц, кто будет пользоваться продуктом или сервисом, понять их потребности.*

**Для бизнеса** использование блокчейна и криптовалют открывает множество возможностей. Во-первых, блокчейн упростит доступ к финансовым услугам жителям развивающихся стран, упростит ведение бухгалтерии и взаиморасчетов, улучшит взаимоотношение с финансовыми регуляторами и повысит качество финансовых операций. Во-вторых, он даст возможность быстрого роста стартапам и компаниям, благодаря умным контрактам улучшится и ускорится взаимоотношение как с партнерами, так и с контролирующими органами.

Как ни странно, но не все компании нуждаются во внедрении блокчейн-технологии. Поддавшись всеобщему помешательству много компаний отчаянно пытаются внедрить блокчейн, несмотря на то, что у них уже есть очень хорошая инфраструктура. Идея, что «все должно быть децентрализовано» неверна.

Возьмите для примера Amazon. Amazon является гигантом электронной коммерции стоимостью в несколько миллиардов долларов, и основным фактором их роста является их пользовательский интерфейс / UX. Амазон – это уникальная клиентоориентированная компания с практически безупречным сервисом. Стоит ли разрушать такой замечательный сервис, внедрив абсолютно новую технологию?

Чаще всего, если вы хотите повысить эффективность своей компании, внедрение блокчейна поможет при замене устаревших решений, которые уже явно нежизнеспособны. Помните, что систематические изменения принесут намного больше пользы для общего роста вашей компании, чем прыжок в неизвестное.

Нет сомнений в том, что технология блокчейн имеет множество блестящих вариантов использования. Тем не менее, большинство из них все еще находятся на теоретической стадии или стадии разработки и тестирования.

Если же решение о внедрении технологии было принято, то необходимо взвесить какой из имеющихся видов блокчейнов будет оптимален для компании:

 - Публичный блокчейн: открытый блокчейн, где каждый может присоединиться и стать частью сети.

 - Частный блокчейны: закрытая сеть, где существующие узлы должны верифицировать участников, прежде чем сделать их частью сети.

- Консорциум: объединение нескольких организаций в рамках одного блокчейна с целью координации их предпринимательской деятельности.

 Хотя публичный блокчейн обладает многими полезными качествами, факт заключается в том, что он просто не готов обрабатывать огромные объемы транзакций и данных, что часто является основным требованием компании.

С другой стороны, у нас есть разрешенные и даже частные блокчейны (частные блокчейны принадлежат одному объекту и контролируются им). Эти виды блокчейнов приобрели огромную популярность и интерес в различных отраслях, таких как финансы. Тем не менее, по мнению критиков, частные и разрешенные блокчейны являются излишне причудливыми реализациями обычной распределенной базы данных.

Перед тем, как принять решение о внедрении блокчейна, компания должна задать себе вопрос: « Что лучше в данной ситуации: применить альтернативное решение или сделать ставку на технологию блокчейн, которая еще относительно молода и может потребоваться несколько лет, чтобы добиться требуемого результата?»

Итак, когда вы должны рассмотреть возможность включения блокчейна, а когда от такого решения лучше отказаться?

Ваша компания зависит от третьей стороны во всех ее операциях? Высока ли стоимость услуг третьей компании? Нужен ли компании четкий контроль операций?

Если да, то блокчейн - хорошая идея для вас.

Однако если ваша компания имеет дело с огромными объемами данных, где время и скорость имеют первостепенное значение, тогда блокчейн может быть не лучшим решением, по крайней мере, на данный момент .

**В финансовом и банковском секторе** блокчейн окажет большое влияние на управление активами и пассивами, регулирование, риск-менеджмент, корпоративные финансы, управление счетами в обслуживании клиентов, торговые операции, процесс встречных сделок. Поможет значительно сэкономить финансы и время. Не за горами появление систем Open Banking. Компании Intel и Microsoft работают над созданием новой блокчейн-платформы CocoFramework для корпоративных клиентов. В платформу будут интегрированы расширения Intel® SoftwareGuardExtensions(Intel® SGX), которые обеспечивают дополнительные уровни защиты личных и конфиденциальных данных благодаря шифрованию данных об операциях, что также повышает масштабируемость. Помимо этого, новая платформа обещает повысить скорость проведения операций и позволит разработчикам создавать гибкие распределенные реестры, с которыми будет удобно работать компаниям.

В данном секторе существующие решения на блокчейне имеют обратную сторону. Многие сторонники техноогии говорят о том, что банки смогут сократить операционные расходы благодаря интеграции блокчейна. Тем не менее, не стоит забывать, что крупные финансовые институты, такие как банки, работают на мэйнфреймах. Вместо того, чтобы заменить мэйнфреймы, блокчейн фактически является причиной увеличить свои инвестиции в мэйнфреймы. Причина, по которой эти мэйнфреймы необходимы, заключается в проблеме масштабируемости блокчейна.

В **медицине и страховании** применение блокчейна позволит оценивать страховые риски и заключать договора с использованием смарт-контрактов, вести историю болезни, доступную в любой точке мира, отслеживать цепочку поставок препаратов.

Отслеживание цепочки поставок медицинских препаратов при помощи блокчейна поможет вести учет лекарств, , усложнит работу мошенников и распространение поддельных препаратов. Вся актуальная информация (от производства до поставки в аптеку) будет храниться в распределенном реестре.

Хранение данных пациентов с использованием блокчейна поможет увидеть всю историю болезни пациента из любого медицинского учреждения, даст возможность самим пациентам контролировать записи в их историях болезни, исключит подделку и пропажу медицинских карт.

Цифровые истории болезни также позволят контролировать расход лекарств, анализировать действия врачей и медсестёр, составлять статистику по любой области здравоохранения в реальном времени. В цифровую карточку каждого пациента станет возможным интеграция договора со страховой компанией. Договор будет представлять собой смарт-контракт, который срабатывает автоматически при оказании услуги, и деньги переводятся больнице, минуя кипу документов. Интеграция смарт-контрактов позволит сэкономить время врачей, персонала больниц и страховых компаний, станет проще процедура получения медицинской помощи.

Блокчейн поможет оснастить медицину унифицированной системой, которая будет одинаково работать на всех компьютерах страны и в будущем даже всего мира. Это уменьшит коррупцию, потому что база данных управляет собой сама, у неё нет администраторов и начальства. Кроме того, при использовании блокчейна станет возможным:

- скоростной доступ к нужной информации из любого уголка страны;

- быстрая обработка данных;

- беспроблемный обмен опытом лечения той или иной болезни;

- своевременная рассылка законов, приказов и актуальных медицинских новостей;

- конфиденциальность истории болезни;

Система окажется очень полезной и для аптек и разработчиков лекарств:

- количество препаратов станет определяться реальной потребностью в них;

- все препараты будут доходить до пациентов;

- исчезнут «подпольные» лекарства и «серые» схемы их продажи;

- будет чётко видна потребность в разработке той или иной фармацевтической продукции.

**В городской инфраструкутруре** блокчейн может быть применен при создании умных городов: в управлении ЖКХ, обеспечении безопасности сетей, взаиморасчетах, при подтверждении права владения и т.д

Перспективным направлением считается использование блокчейна совместно с **IoT (Internet of Things)**. Современные экосистемы IoT основаны на централизованных моделях коммуникаций, которые также называют парадигмой клиент / сервер. Все устройства идентифицируются, проходят проверку подлинности и подключаются через облачные серверы, которые обладают огромными возможностями обработки и хранения. Связь между устройствами должна осуществляться исключительно через Интернет, даже если они на расстоянии нескольких футов друг от друга. Несмотря на то, что эта модель используется на протяжении десятилетий и будет продолжать поддерживать небольшие сети IoT, какими мы их видим сегодня, она не сможет удовлетворить растущие потребности огромных экосистем IoT завтрашнего дня.

Существующие IoT-решения дороги из-за высокой стоимости инфраструктуры и обслуживания, они связанны с централизованными облаками, большими фермами серверов и сетевым оборудованием. Огромное количество сообщений, которые необходимо будет обработать, когда число устройств IoT увеличится до десятков миллиардов, существенно увеличит эти затраты. Даже если беспрецедентные экономические и инженерные проблемы будут преодолены, облачные серверы останутся узким местом и точкой отказа, которая может нарушить работу всей сети.

Кроме того, разнообразие владения между устройствами и поддерживающей их облачной инфраструктурой затрудняет обмен данными между компьютерами: machine to machine (M2M). В настоящее время не существует единой платформы, соединяющей все устройства. Применение блокчейн технологии позволит развиваться миру IoT.

Децентрализованный подход к созданию сетей IoT позволил бы решить многие из указанных выше вопросов. Принятие стандартизированной модели одноранговой связи для обработки сотен миллиардов транзакций между устройствами значительно сократит затраты, связанные с установкой и обслуживанием крупных централизованных центров обработки данных, и IoT. Это предотвратит остановку работы сети из-за сбоя в любом отдельном узле сети.

Технология Blockchain может использоваться для отслеживания миллиардов подключенных устройств, обеспечения обработки транзакций и координации между устройствами. Она сможет обеспечить значительную экономию для производителей IoT, а также позволит экономить энергетические ресурсы планеты. Децентрализованный подход устранит отдельные точки отказа, создав более устойчивую экосистему для устройств, на которых можно будет работать. Криптографические алгоритмы, используемые блокчейнами, сделают данные потребителей более конфиденциальными.

**На производстве**. Еще одна сфера, где блокчейн может помочь в ведении бизнеса, - это обеспечение качества продукта. Поскольку блокчейн прозрачен, каждая его часть может быть тщательно изучена и исследована.

Очень интересно применение блокчейна в пищевой промышленности. По словам Майка Алмейда, президента Empire ATM Group: «Такие события, как эпидемия сальмонеллы, можно будет отследить до источника заражения за считанные секунды, и каждый отдельный продукт из этой партии будет помечен для уничтожения, независимо от того, был ли он недавно упакован на заводе по переработке, в доставляемом грузовике или уже в розничном магазине. Эта система может уведомить всех, кто участвует, за секунды, а не за дни»

Шумиха вызванная блокчейном на данный момент довольно огромна. Фактически, недавние исследования показывают, что компании, которые добавляют слово «блокчейн» к своим именам, наблюдали значительный рост в цене своих акций. Например, акции небольшой американской компании по производству безалкогольных напитков выросли на 432% после того, как компания сменила название с Long Island Iced Tea Corp на Long Blockchain Corporation. Этот нереальный ажиотаж возлагает на блокчейн много несправедливых ожиданий.

Хотя считается, что блокчейн - отличная инновация, он ни в коем случае не является той волшебной кнопкой, которая решит все проблемы, в частности, плохая бизнес-идея не станет хорошей, если к ней прикрутить блокчейн.

Технология все еще очень молода, и есть множество новых возможностей, которые необходимо изучить и улучшить, таких как масштабируемость и функциональная совместимость. Однако, преодолев эти сложности, блокчейн станет более мощным и распространенным.

Глава 2. Как работает блокчейн.

Блокчейн представляет собой распределенную децентрализованную систему, в которой могут находиться десятки тысяч участников. В основе взаимодействия сети заложен алгоритм консенсуса. Консенсус помогает решить вопрос достижения одинаковой точки зрения на журнал транзакций блокчейна в распределенной сети при условии, что произвольные узлы могут «падать» или зависать, руководствуясь лишь общими правилами обработки сообщений в сети. Идея консенсуса не нова и имеет реализованные решения в других распределенных системах (например, база данных NoSQL). Задача византийского консенсуса, в котором узлы могут вести себя «плохо» была сформулирована в 80-х годах прошлого века и уже имеет ряд решений.

**Интересные факты.** *Что же представляет собой задача византийских генералов? Византия. Ночь перед великим сражением с противником. Византийская армия состоит из n легионов, каждым из которых командует свой генерал. Также у армии есть главнокомандующий, которому подчиняются генералы. В то же самое время, империя находится в упадке, и любой из генералов и даже главнокомандующий могут быть предателями Византии, заинтересованными в её поражении.*

*Ночью каждый из генералов получает от предводителя приказ о варианте действий в 10 часов утра (время одинаковое для всех и известно заранее), а именно: «атаковать противника» или «отступать».*

*Возможные исходы сражения:*

*- Если все верные генералы атакуют — Византия уничтожит противника (благоприятный исход).*

*- Если все верные генералы отступят — Византия сохранит свою армию (промежуточный исход).*

*- Если некоторые верные генералы атакуют, а некоторые отступят — противник уничтожит всю армию Византии (неблагоприятный исход).*

*Также следует учитывать, что если главнокомандующий – предатель, то он может дать разным генералам противоположные приказы, чтобы обеспечить уничтожение армии. Следовательно, генералам лучше не доверять его приказам.*

*Если же каждый генерал будет действовать полностью независимо от других (например, сделает случайный выбор), то вероятность благоприятного исхода весьма низка.*

*Поэтому генералы нуждаются в обмене информацией между собой, чтобы прийти к единому решению.*

Консенсус в блокчейне отличается тем, что участники сети заранее неизвестны (в отличие от византийского консенсуса, у узлов которого есть «личности»), могут подключаться и отключаться от сети произвольно, блокчейн обладает объективностью (для определения текущей версии журнала транзакций не нужно доверие неким авторитетным источникам — корень доверия находится в самом блокчейне) и отсутствием регуляторов (никто не может вмешаться в работу сети).

Для технологии блокчейн уже разработано около десяти видов алгоритмов **консенсуса**. Рассмотрим их отличительные особенности

*Proof-of-Work(PoW)*. PoW консенсус (его еще называют консенсус Накомото) – был самым инновационным решением, предложенным в статье Сатоши «Электронная денежная наличность». PoW решает многие проблемы электронных денег, такие как, например, атаки Сибиллы. Кроме того, PoW обеспечивает честность в сети: успех майнера пропорционален мощности вычисленых способностей его оборудования (hashrate). Если хешрейт майнера составляет 1% всей сети, то наиболее вероятно, что он будет создавать 1% блоков и получать 1% награды. Именно поэтому биткоин появился на основе PoW. В данном протоколе узлы блокчейн сети, подтверждающие транзакции проделывают достаточно сложную вычислительную работу, результат которой легко и быстро проверяется другими узлами сети. Первый узел, который полностью провел все необходимые вычисления — получает вознаграждение от блокчейн сети. Все узлы борются между собой (наращивая емкость вычислительных ресурсов), чтобы оказаться тем самым, первым узлом, получившим вознаграждение. В PoW– вероятность успешного решения криптографического пазла, от которого зависит создание нового блока, как уже было сказано выше, больше у обладателей более мощного вычислительного оборудования. Основной недостаток – огромные энергетические затраты, т.к. работает множество узлов, а принимается результат работы только одного узла. Второй недостаток – отсутствие полезности: вычисления, осуществляемые для доказательства работы – не служат никакой полезной цели (разве что обеспечение безопасности). Решение криптографических задач происходит ради их решения. При этом для того, чтобы произвести расчеты, нужно потратить ресурсы, в первую очередь электричество. Третий недостаток – проблема сохранения децентрализации. Основная задача этого протокола — предотвратить DDoS атаки со стороны злоумышленников, желающих навредить работе финансовой системы. Для обработки данного протокола требуется так называемое «компьютерное время» - время, которое расходуется на майниг блоков. Безопасность сети обеспечивается суммой вычислительных мощностей всех участников сети. Именно для работы с этим протоколом люди постоянно покупают все новое и новое оборудование, стремясь увеличить свои мощности в сети и получить за это вознаграждение. Таким образом, чтобы атаковать сеть, хакерам нужно иметь 51% мощности всей сети, что затруднительно, но в принципе реально. Хакерам не обязательно запускать собственные узлы – достаточно взять под контроль нужное количество уже запущенных.

Блокчейн Биткойна работает именно с этим протоколом. Протокол показал свою состоятельность, хотя не обошлось без атак и сбоев. Например, учитывая что основная вычислительная мощность сосредоточена в руках нескольких крупных пулов, теоретически можно предположить успешный сговор и атаку 51%. Подобная атака была уже однажды предпринята с целью установить Bitcoin Cash в качестве основной валюты.

Пример алгоритма PoW, реализованного на Python ниже.

import itertools

from hashlib import sha256

# Интерпретирует последовательность символов как little-endian число

to\_long = lambda x: sum(ord(b) << (8\*i) for i, b in enumerate(x))

# Комбинирует nonce и сообщение для вычисления хэша

combine = lambda nonce, msg: str(nonce) + ":" + msg

# Проверяет доказательство работы

def verify\_pow(msg, nonce, difficulty):

    h = sha256(combine(nonce, msg)).digest()

    # Равны ли первые difficulty битов хэша нулю?

    return to\_long(h) % (1 << difficulty) == 0

# Создает доказательство работы для сообщения

def create\_pow(msg, difficulty):

    for nonce in itertools.count(0):

        if verify\_pow(msg, nonce, difficulty):

            return nonce

msg = "blickchain"

nonce = create\_pow(msg, 16)

combine(nonce, msg), sha256(combine(nonce, msg)).hexdigest()

# 43952:blockchain 000027b5022f88d2da21bd2802268966050f5a0b031058ce4562939c13727303

*Proof-of-Stake (PoS)*. Протокол Proof of Stake (PoS) (доказательство доли владения) впервые был представлен в работе Санни Кинга (Sunny King) и Скотта Надаля (Scott Nadal) в 2012 году как решение проблем PoW, связанных с большой тратой энергии и возможностью атаки 51%. В этом алгоритме создателем следующего блока в цепочке блоков выбирается узел, который обладает б*о*льшим балансом — количеством ресурсов, например монет в криптовалюте. За само создание блока, узел вознаграждение не получает. Вознаграждение за проведение транзакции выплачивается в виде комиссии.

При разработке структур на базе Proof of Stake разработчики сталкиваются с четырьмя основными видами сложностей:

Распределение. Проблема в распределении первоначальных монет связана с тем, что в самом алгоритме заложено их перераспределение внутри блокчейна в результате транзакций. Кто должен изначально владеть монетами блокчейна? Этот вопрос необходимо решить в самом начале.

Монополизация. Пользователям, владеющим значительными суммами монет, достается основная доля всех будущих монет, что ведет к централизации.

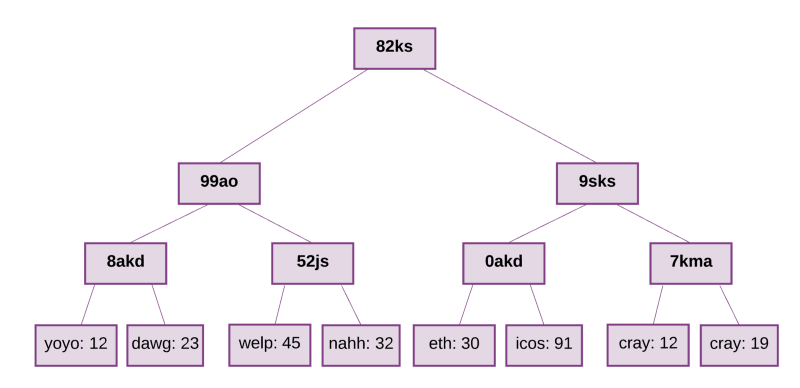
Атака 51%. Аналогично тому, как алгоритм Proof of Work (PoW) уязвим для атаки со стороны 51% майнеров, так и PoS уязвим для атаки со стороны пользователей, чья доля составляет 51% и более, хотя логически менее вероятна.

Нулевая доля, «ничего на кону» (Nothing at Stake). Данная проблема заключается в том, что в случае ошибки консенсуса, майнеры блоков ничего не теряют, голосуя за несколько ветвей (форков) одновременно или создавая блоки с нулевой стоимостью. Создание подобной ситуации также может вести к возникновению двойных трат. Решение проблемы с помощью внесения обеспечительного взноса для валидации лишь усложнило задачу недобросовестным участникам сети, но не искоренило проблему.

Помимо вышеперечисленных алгоритмов консенсуса, которые нашли широкое применение, также существуют алгоритмы, не менее перспективные, но используемые реже: *Delegated PoS, Leased PoS, Proof-of-Capacity, Proof-of-Importance, Proof-of-Activity, Proof-of-Authority*.

Что же обрабатывается и согласовывается в блокчейнах?

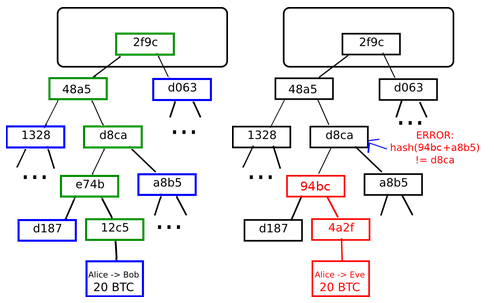
Основу блокчейна составляет дерево Меркла или хэш-дерево - полное двоичное дерево, в вершины которого помещены хеши от блоков данных, а внутренние вершины содержат хеши от сложения значений в дочерних вершинах. Корневой узел дерева содержит хеш от всего набора данных, то есть хеш-дерево является однонаправленной хеш-функцией. В нем хранится сопоставление адресам учетной записи состояний счета. В самом общем смысле дерево Меркла — это структура данных, которая хеширует большое количество элементов данных (chunk). Для этого несколько элементов объединяются в блоки (bucket), эти блоки хешируются, получившиеся в результате хеши объединяются в новые блоки и т. д., пока в итоге мы не получаем один-единственный корневой хеш.



Дерево Меркла делает возможными так называемые доказательства Меркла (Merkle proof). Доказательство Меркла состоит из элемента, корневого хеша дерева и «ветви», которая включает все хеши от элемента до корня. Благодаря этому можно легко убедиться, что правила хеширования соблюдены на протяжении всей ветви, а это доказывает, что элемент занимает свое законное место.

В блокчейнах корень дерева всегда известен и заслуживает доверия, т.к. он либо подписан большим количеством доверенных сторон, либо подтвержден большим объемом майнинг-мощностей. Следовательно, любой пользователь блокчейна, желающий проверить то или иное значение, может запросить доказательство Меркла и убедиться, что ему возвращено верное значение, проверив правильность доказательства.

При попытке изменения оригинальной транзакции происходит изменение хэша каждого верхнего узла, что приводит в итоге к изменению корня



В блокчейне Биткойна доказательство Меркла используется для хранения данных о транзакциях в блоках. Для упрощенной верификации платежей (SPV) легкий клиент может скачать только цепочку заголовков блоков, каждый из которых содержит всего 80 байтов:

- хэш предыдущего заголовка;

- временную метку;

- одноразовый код подтверждения работы (число nonce);

- значение сложности майнинга;

- корневой хэш дерева Меркла.

Проблема легких клиентов Блокчейна заключается в том, что они позволяют проверить только наличие транзакции в блоке, но не текущее состояние ситстемы: невозможно проверить, кому принадлежит тот или иной цифровой актив, наличие доступных биткойнов на конкретном адресе и т.д

В Ethereum каждый заголовок блока содержит не одно, а три дерева Меркла для объектов трех различных типов: транзакции, квитанции, состояния.

Благодаря этому легкие клиенты могут получать ответы на нетривиальные запросы:

- наличие данной транзакции в конкретном блоке;

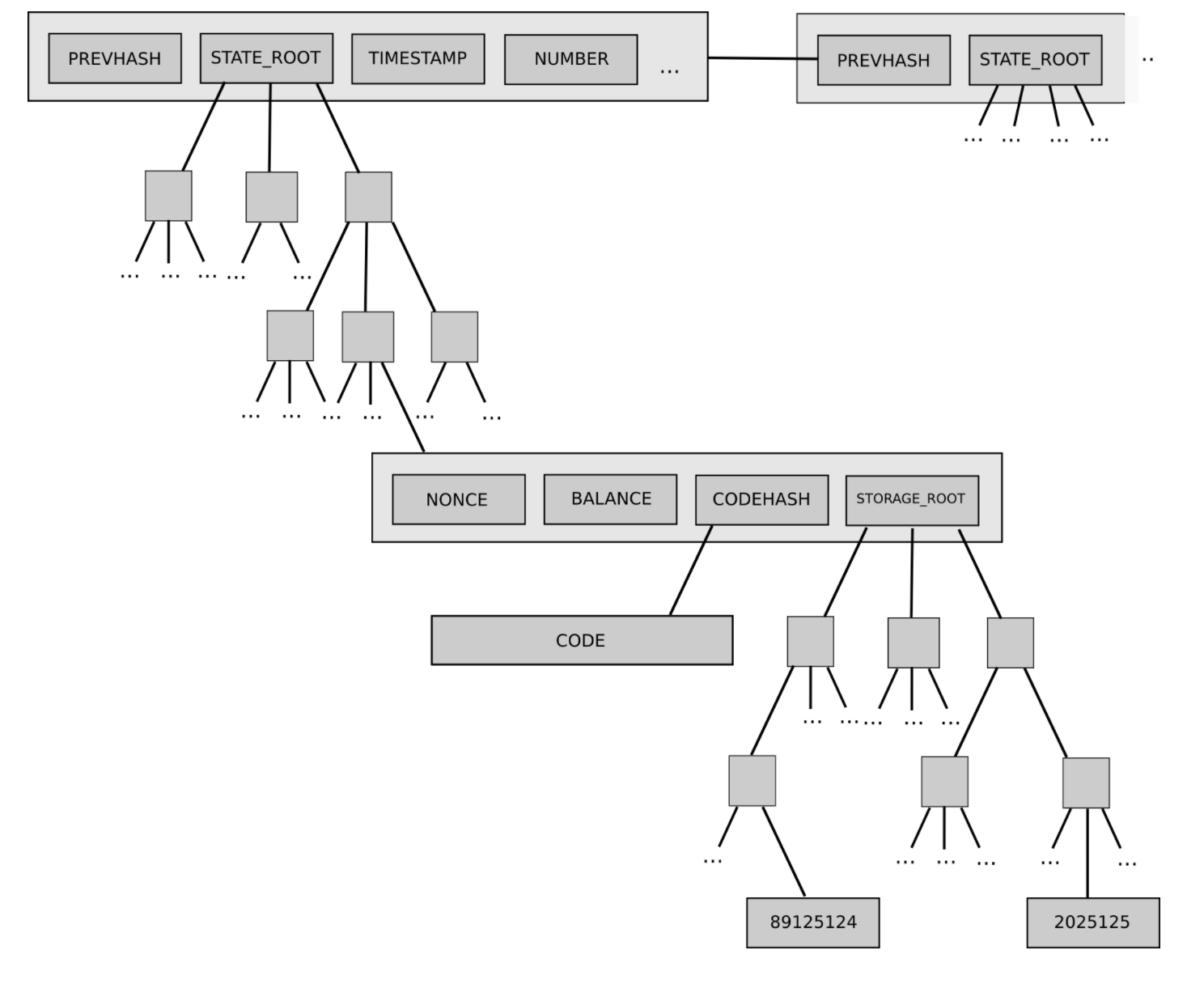
- количество событий определенного типа, сгенерированных определенным адресом за определенное количество дней.

- текущий баланс собственного счета

- проверка существования счета;

- проверка результата выполнения транзакции для определенного смарт-контракта.

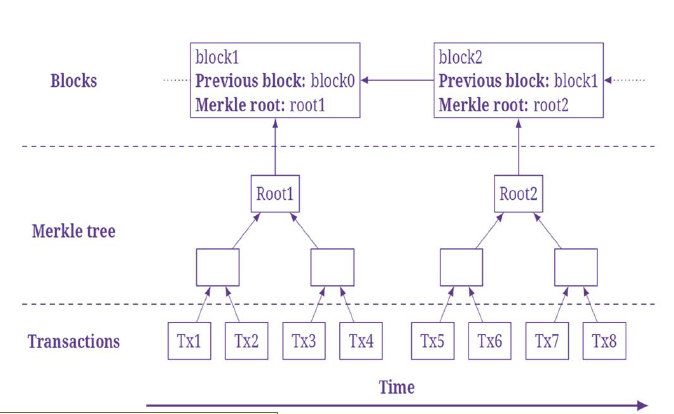
Для выполнения последнего запроса используется Меркл-доказательство изменения состояния, которое является по сути доказательством того. Что «если выполнить транзакцию T в состоянии с корнем S, результатом будет состояние с корнем S’, протоколом (log) L и выводом O». Для этого необходимо, чтобы сервер создал поддельный блок с состоянием S и выполняет транзакцию под видом облегченного клиента, в том числе запрашивает баланс, если транзакция содежит получение баланса счета, или запрашивает элемент из хранилища контракта, если это необходимо в транзакции. Сервер выполняет свои собственные запросы, записывая отправляемые данные, а затем пересылает объединенные данные клиенту как доказательство. После этого клиент проделывает то же самое, используя в качестве базы данных полученное доказательство и принимает доказательство при совпадении результатов.



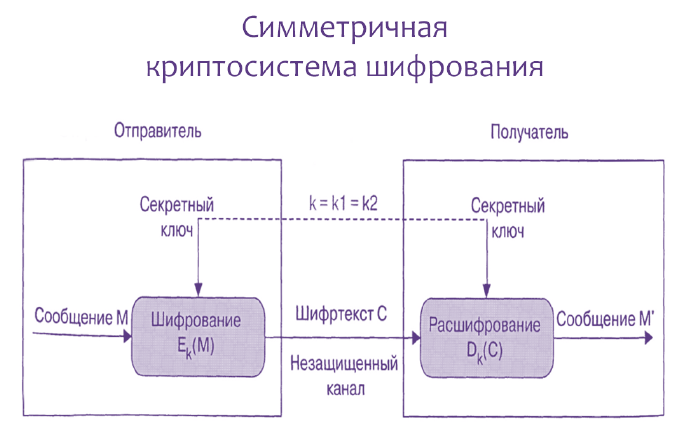
Двоичные деревья Меркла подходят для представления деревьев транзакций. Для деревьев состояний используют префиксные деревья Меркла (Mercle patricia tree). Это связано с динамически изменяющимися данными при работе с которыми желательно использовать структуру, у который новый корень вычисляется легко после вставки, обновления, изменений\удалений данных, не вычисляя заново все дерево. Также при работе с такой структурой желательно, чтобы глубина дерева была ограничена из-за угрозы того, что злоумышленники могут попытаться создать как можно более глубокие деревья для проведения DDos-атаки, а корень дерева должен быть зависим только от самих данных, но не от порядка выполнения обновлений.

В Ethereum состояние – это таблица ключей и значений, где ключи являются адресами, а значения – счетами с балансами, одноразовым кодом и хранилищем. В префиксном дереве ключ, по которому хранится значение, кодируется в виде ветви дерева. У каждого узла существует 16 дочерних узлов, позволяющих кодировать ключи в шестнадцатеричной системе счисления.

В любом блокчейне обрабатываются лишь два типа сообщений – **транзакции и блоки**, представляющие собой списки транзакций.

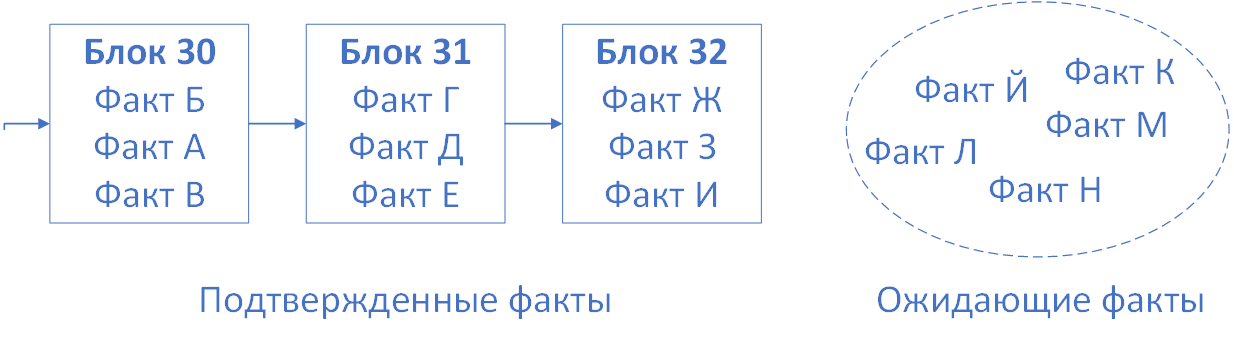


Транзакции формируются участниками системы. Для формирования и отправки транзакций, применяется алгоритм симметричного шифрования с использованием открытых и закрытых ключей участников. О них мы поговорим далее.



Открытые и закрытые ключи хранятся в **криптовалютных кошельках**, представляющих собой специальное программное обеспечение, которое взаимодействует с различными цепочками блоков и позволяет пользователям отправлять и получать криптовалюту. Когда кто-либо отправляет криптовалюту, он, по сути, подписывает право владения монетами получателем. Чтобы иметь возможность потратить эти монеты и разблокировать средства, закрытый ключ, хранящийся в кошельке, должен соответствовать общедоступному адресу, которому отправляется данная криптовалюта. Если открытый и закрытый ключи соответствуют друг другу, баланс в цифровом кошельке получателя увеличится, а у отправителя соответственно уменьшится. Реального обмена криптомонетами не существует: просто происходит запись транзакции в блокчейне и изменение баланса в криптовалютных кошельках .

Блоки являются основным продуктом алгоритма консенсуса и определяют порядок включения транзакций в журнал транзакций. Идея проста: факты (транзакции) группируются в блоки, и существует только одна цепочка блоков, реплицированных во всей сети. Каждый блок ссылается на предыдущий. Таким образом, если факт F находится в блоке 21, а факт E находится в блоке 22, то факт E рассматривается всей сетью как дочерний по отношению к факту F. Прежде, чем добавиться к блоку, факты находятся на рассмотрении, то есть до добавления они остаются неподтвержденными.



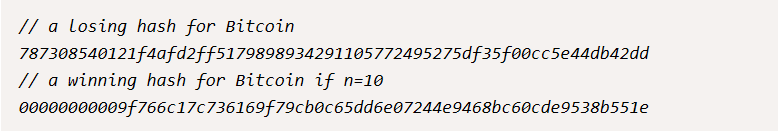
Теоретически, можно обойтись и без использования блоков, включая транзакции в журнал по отдельности. На практике блоки экономят объем трафика и вычислительные ресурсы сети.

Блоки создаются особой категорией узлов сети блокчейна – узлами (нодами) консенсуса. Во многих блокчейнах (Bitcoin, Ethereum) эти узлы называют майнерами и они получают вознаграждение за свою работу, или комиссию. Майнеры соревнуются за то, чтобы их блок включили в сеть, т.к. только после проверки всеми узлами и включения блока в блокчейн все факты, находящиеся в блоке становятся подтвержденными и майнер получает свое вознаграждение. В некоторых блокчейнах майнеры также могут получать комиссию, которая приплюсовывается к вознаграждению. Майнеру всегда необходимо запускать полный узел (полную копию истории транзакций блокчейна), чтобы можно было выбрать действительные транзакции для формирования нового блока. Без полного узла майнер не может определить, какие предложенные транзакции действительны в соответствии с историей транзакций текущего блокчейна, потому что он не имеет доступа к полной истории блокчейна.

Сложность майнинга - это степень, определяющая, насколько сложно майнерам с точки зрения времени и сложности хеширования найти приемлемый хеш (подпись) для своего блока (блок транзакций требует проверки приемлемого хеша и добавления его в цепочку блоков). В биткойн-блокчейне майнеры пытаются найти подходящий хеш, хешируя случайные числа. Давайте рассмотрим этот процесс.

*Основная идея майнинга в Биткойне заключается в том, что майнеры группируют биткойн-транзакции в один блок, который уже подвергают хэшированию для нахождения очень редкого значения хэша, попадающего под специальные условия. Когда такое значение находится, блок считается смайненным и попадает в цепочку блоков. Для майнинга в Биткойне используется хеширование строки SHA-256. Криптографическая функция хэширования на вход получает блок с данными, а выдаёт небольшой, но непредсказуемый, выход. Она спроектирована так, что не существует быстрого способа получить нужный выход, и необходимо продолжать перебор пока не будет получено подходящее значение. Причём для усиления стойкости SHA-256 применяется к блоку дважды и называется уже двойным SHA-256. Алгоритм работает с данными, разбитыми на куски по 512 бит (64 байт), криптографически их смешивает и выдаёт 256-битный (32 байта) хэш. SHA-256 состоит из относительно простого раунда, повторяющегося 64 раза. В биткойне критерием валидности хэша является достаточное число нулей в его начале.*

*Узел выигрывает, если его хэш содержит хотя бы n ведущих нулей.*



*Число n корректируется время от времени, чтобы поддерживать фиксированную длительность блока, несмотря на различия в количестве узлов. На текущий момент, правильный хэш должен содержать 17 стартовых нулей, чему удовлетворяет только 1 из 1.4x1020.*

Как уже говорилось выше блок транзакций будет принят остальной частью сети, только если у него есть подпись (хэш), которая соответствует определенным требованиям. Чтобы найти эту подпись, майнеры тратят вычислительную мощность (мощность хеширования) для выполнения набора заранее определенных операций со случайными числами, пока не найдут число, которое приводит к выходному числу, которое соответствует требованиям. Найти выходное число, которое начинается только с одного нуля, гораздо проще, чем найти выходное число, которое начинается с пяти последовательных нулей и на поиск такого числа потребуется гораздо больше времени. Таким образом заданное количество нулей в начале подписи определяет сложность майнинга. Сложность майнинга в сети автоматически корректируется через определенный период времени в зависимости от текущего значения времени, затраченного на майнинг одного блока. Когда все больше майнеров присоединяются к сети для майнинга, общая мощность хеширования увеличивается и майнеры находят подписи быстрее.

*Сложность майнинга на блокчейне Биткойн корректируется каждые две недели, так что общая мощность хеширования в сети в среднем составляет 1 блок за 10 минут. Если многие майнеры присоединятся к блокчейну Биткойн, и совокупная мощность хеширования возрастет, то скорость производства блоков может увеличиться с 1 блока за 10 минут до 1 блока за 9 минут. Через некоторое время это также вызовет увеличение сложности майнинга, так что даже при такой дополнительной мощности хеширования производительность блока будет оставаться стабильной на уровне 1 блока в 10 минут. То же самое относится и к обратному: когда майнеры прекращают майнинг на блокчейне и совокупная скорость хэширования падает, сложность майнинга также снижается.*

Скорость майнинга блока напрямую зависит от того, с какой скоростью обновляются узлы блокчейна. Если скорость майнинга увеличится, то может наступить рассинхронизация сети. Скорость майнинга также можно снизить увеличив размер блока. В этой ситуации в один блок будет помещаться больше транзакций, но в то же время будет создаваться больше дублирующих блоков (uncle blocks), которые будут засорять блокчейн.  
Второй тип узлов — это легкие клиенты (кошельки), не имеющие полной версии блокчейна. Они содержат лишь важные для узла данные. Легкие клиенты требуют меньше вычислительных ресурсов и объемов памяти, поэтому могут работать на мобильных платформах. Эти приложения позволяют им транслировать транзакции из своего кошелька без необходимости загружать всю историю блокчейна на собственное устройство

Изображение выглядит как небо, провод

Автоматически созданное описание

Третий тип узлов – это мастер-узлы (мастерноды). Их можно сравнить с большими серверами в сети. Мастерноды, как правило, работают 24/7, имеют большой объем памяти. Их задачей, помимо стандартных действий узла, также является управление событиями голосования, обеспечение выполнений операций протокола и контроль соответствий в блокчейне, распределение средств для развития сети, повышение уровня конфиденциальности и скорости транзакций. В связи с тем, что мастерноды могут оказывать сильное влияние на сеть, не каждый пользователь может запустить их, а для его запуска часто требуется гарантированный депозит. Это своеобразные аудиторы сети. «Аудиторы» также регулярно проверяют работу майнеров и занимаются распределением нагрузки по сети, практически выполняя функцию сети доставки контента (CDN) для данных блокчейна. Чтобы запустить мастерноду, необходимо купить и зарезервировать определённое количество монет и поднять полный узел сети, настроенный специальным образом. C практической точки зрения это означает, что вы покупаете монеты, арендуете или используете свой сервер, на котором запускаете круглосуточно работающий кошелёк. За это сама сеть из пула эмиссии новых монет будет вознаграждать работу вашей мастерноды. Для многих содержание мастерноды является хорошим способом заработка, альтернативой майнингу.

Четвертый тип узлов в блокчейнах – это загрузочные узлы (bootnode). Они не синхронизируют данные блкчейна. Их единственное назначение – помочь другим узлам в обнаружении друг друга. Загрузочные узлы являются добавочными узлами для пиров в сетях. Поэтому рекомендуется знать IP-адрес компьютера, на котором размещен загрузочный узел, и ссылаться на него при подключении к блокчейну.

Узлы могут быть онлайн или оффлайн. Онлайн-узлы получают, сохраняют и транслируют все последние блоки транзакций с других узлов. Когда оффлайн (автономный) узел возвращается в оперативный режим, ему необходимо сначала догнать блокчейн, загрузив все блоки, которые были добавлены в блокчейн с тех пор, как узел перешел в автономный режим (не касается легких клиентов). Этот процесс часто называют синхронизацией с блокчейном. Теоретически блокчейн остается работоспособным, даже если работает всего один полный узел. Если внезапно вследствие какой-либо катастрофы отключатся все узлы, кроме единственного, содержащего в себе полную историю блокчейна, то работоспособность блокчейна можно будет восстановить.

Каждую секунду каждый майнинговый узел в блокчейне проверяет тысячи случайных строк, пытаясь сформировать новый блок. Таким образом, запуск майнера в блокчейне потребляет огромное количество ресурсов, таких как электричство, использование хранилища и процессора. Вот почему мы должны платить за хранение фактов в блокчейне. В то же время, чтение фактов бесплатно: вам просто нужно запустить свой собственный узел, и вы восстановите всю историю фактов, выпущенную всеми другими узлами. В связи с тем, что во многих блокчейнах содержится огромный объем данных, запуском узлов занимаются в основном майнеры и криптоэнтузиасты, желающие внести свой вклад в развитие блокчейна. Обычным пользователям достаточно иметь у себя приложение криптокошелька, транслирующего транзакции в блокчейн без необходимости загрузки узла на устройство. Узел может запустить любой пользователь блокчейна, но многие, учитывая то, что блокчейны содержат в себе большой объем данных, предпочитают запускать легкие клиенты (кошельки). Существует два вида кошельков: холодные и горячие. Холодные или оффлайн кошельки, например, аппаратные, хранят цифровые монеты в оффлайне, без доступа к интернету на специальном устройстве или обычном USB-накопителе. Горячие кошельки или онлайн используются для хранения небольших объемов монет для повседневного использования. Среди онлайн кошельков бывают обычные кошельки, часто сразу встроенные в биржу, и мультивалютные кошельки, используемые для одновременного хранения нескольких типов монет.

Про какие монеты в блокчейнах идет речь? Фактически, каждый блокчейн имеет свою собственную **криптовалюту**: Биткойн (BTC) в сети Биткойн, Ether (ETH) в сети Ethereum и т. д. Чтобы совершить платеж в сети Биткойн, вы должны заплатить небольшую комиссию в биткойнах - точно так же, как вы платите комиссию банку. Майнеры получают вознаграждение за поддержание работоспособности и безопасности сети. Каждый раз, когда они успешно добывают блок, они получают фиксированную сумму в криптовалюте. Таким образом, блокчейн генерирует собственные деньги. Практически все криптовалюты можно конвертировать в реальные деньги. Их номинальная стоимость определяется только спросом и предложением, поэтому это является предметом спекуляции и торговли на биржах.

*Не все криптовалюты построены на блокчейне. Первая криптовалюта, не основанная на блокчейне, появилась в 2015 году. Ее название DagCoin связано с направленными ациклическими графами - directed acyclic graph (DAG). Криптовалюта IOTA, основанная на технологии Tangle (клубок), появилаь в 2016 году. Tangle (клубок, сплетение) – это модификация DAG, основное отличие которой в минимальном количестве транзакций, которые должны быть подтверждены в новой транзакции. Если в DagCoin каждая транзакция должна иметь минимум одну родительскую транзакцию, то в IOTA таких транзакций должно быть минимум две. Такие криптовалюты выигрывают перед классическими криптовалютами, основанными на блокчейнах по нескольким параметрам: скорости, надежности и упрощении майнинга.*

На одном блокчейне может существовать несколько криптовалют: материнская (исходная) криптовалюта и криптовалюты, возникшие в результате хард-форка. Хард-форк блокчейна — это внесение изменений в исходный код блокчейна, вследствие чего создаются новые правила функционирования системы, несовместимые со старыми алгоритмами. Поскольку старые и новые правила не могут работать в пределах одной цепочки, происходит разветвление сети (отсюда и название процесса, ведь hard fork переводится как «жесткая развилка»). «Материнская» система продолжает работать по установленным алгоритмам, в то время как в новом блокчейне образуется своя экосистема.

Изображение выглядит как музыка

Автоматически созданное описание

Поскольку в результате хард-форка появляются 2 самостоятельные платежные системы, дальнейшие события развиваются по одному из таких сценариев:

1. Новая сеть привлекает на свою сторону больше пользователей, в результате чего «материнская» система отмирает.
2. Хардфорк оказывается неудачным и новый блокчейн перестает функционировать, в то время как старая сеть продолжает работать.
3. Обе системы существуют параллельно, но независимо друг от друга. Это происходит в том случае, если старый и новый блокчейн имеют достаточное количество пользователей.

Bitcoin Cash (BCH) является самым известным примером хард-форка оригинального блокчейна Биткойна, ответвление которого произошло в августе 2017 года. Хард-форк произошел в связи с увеличением размера блока с 1 МБ до 8 МБ, чтобы помочь масштабируемости.

Как правило, при проведении хард-форка затрагиваются следующие аспекты:

- Устанавливается максимальная эмиссия новой валюты.

- Прописывается время генерации блока.

- Устанавливается сумма награды за раскрытие блока и периодичность её корректировки.

- Создается алгоритм, регулирующий сложность майнинга.

- Устанавливается размер блока (дополнительно прописываются возможности последующего изменения этого параметра).

Хард-форк также может быть реализован для исправления важных угроз безопасности или для отката блокчейна и отмены транзакций. Хард-форк по случаю отмены транзакции произошел в Ethereum в связи со взломом DAO. Сообщество Ethereum проголосовало за откат хардфорк, чтобы вернуть десятки миллионов средств, украденных в результате ее взлома.

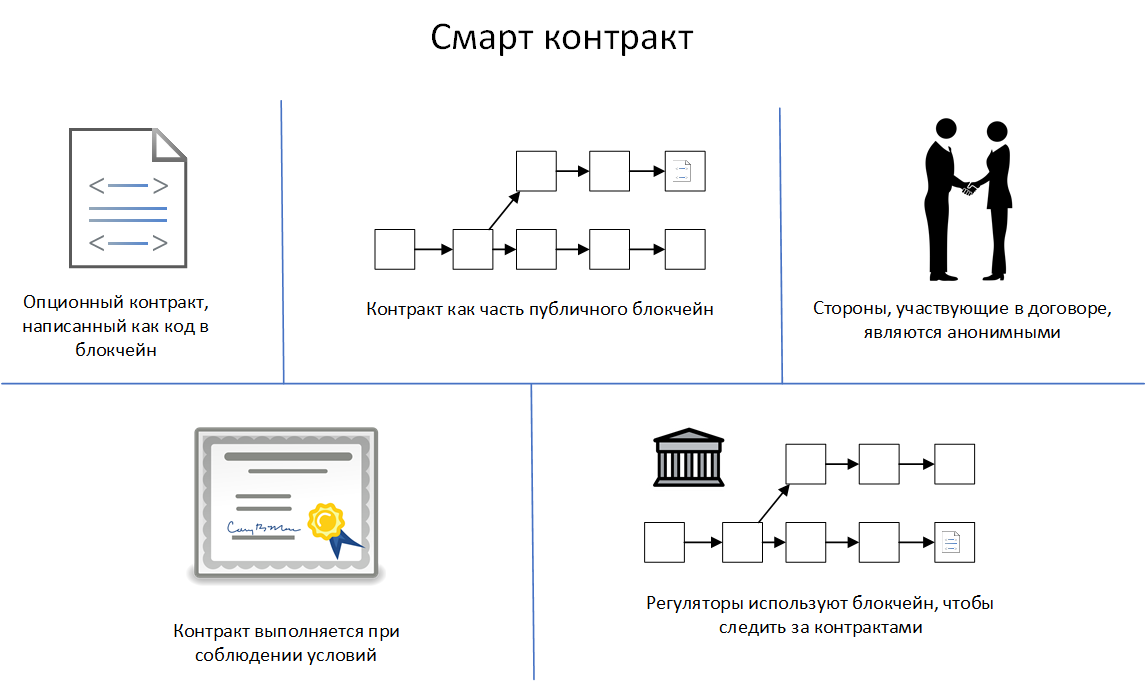
С помощью хард-форка участники блокчейна также имеют право на средства в виде токена, связанного с новым блокчейном. Например, если кто-либо из участников держал 10 биткойнов во время хард-форка Bitcoin Cash и использовал кошелек, поддерживающий вилку, он также получил 10 биткойнов Cash.

Теоретически, провести хард-форк может любой пользователь сети: достаточно взять оригинал кода, внести в него соответствующие изменения и запустить новый блокчейн. Практически для успешного хард-форка нужна большая команда, состоящая из единомышленников, средства на рекламу и продвижение новой монеты, а также уникальная идея, способная привлечь новых пользователей.

В большинстве случаев хард-форки проводятся членами команды разработчиков какого-либо актива, майнинг-пулами, держателями большого количества монет, известными аналитиками, трейдерами. Большинство известных хард-форков было проведено по причине разногласия в сообществе относительно стратегии развития существующих монет или с целью создания новой более совершенной монеты, которая будет иметь все преимущества старой монеты, но лишена ее недостатков.

При улучшении некоторых компонентов сети (повышении пропускной способности транзакций, внедрении дополнительной защиты от хакерских атак и прочее) разработчики осуществляют софт-форк. При этом старая и новая системы совместимы и взаимодействуют между собой. Никакого разветвления не происходит; новая криптовалюта не образуется. Это означает, что существующим узлам в сети не нужно обновляться или принимать новый протокол для того, чтобы новые транзакции считались действительными. Одним из самых известных примеров софт-форка является внедрение технологии SegWit в систему Биткоина.

Помимо хранения фактов блокчейны могут выполнять программы. Такие программы реплицируются вместе с фактами и каждый узел выполняет их при получении фактов. Так, в Биткойне можно привязать отправку криптовалюты к определенной дате. Другие блокчейны допускают более сложные контракты. Например, в Ethereum каждый контракт содержит мини-базу данных и предоставляет методы для изменения этих данных. Заранее запрограммированные условия, связанные с реальным миром и транслируемые всем, называются умными контрактами или **смарт-контрактами**. В реальном мире контракт - это обещание, которое подписавшие стороны соглашаются выполнить с учетом юридических условий. Смарт-контракт в блокчейне практически то же самое, нужно только заменить слово «юридических» на «технических». Применение смарт-контрактов устраняет необходимость в суде или любом третьем органе, признанным обеими сторонами для разрешения споров и контроля выполнения условий.



Простейшим примером наиболее часто используемого смарт-контракта является мульти-подпись (multisig, escrow). При помощи такого контракта не доверяющие друг другу контрагенты могут заморозить некоторую сумму монет на блокчейне таким образом, что в случае необходимости потратить эту сумму потребуются подписи более половины участников. Дальнейшее усложнение таких контрактов позволяет выстраивать модели для голосований о распределении средств в рамках децентрализованных фондов или запускать подписи по наступлению определенных цифровых событий.

На практике это означает, что инвестор, принимая участие в ICO и отправляя криптовалюту на кошелек проекта, может быть уверен в том, что в случае провала краудсейл-кампании, его средства будут автоматически возвращены. А в случае успешного сбора заявленной суммы, средства будут перечислены разработчикам только тогда, когда достаточное количество участников мультиподписи активируют свои ключи, тем самым лично подтверждая добросовестность проекта.

Применение смарт контрактов в будущем станет неотъемлемой частью цифровой экономики. В некоторых странах применение смарт-контрактов для голосования уже рассматривается на государственном уровне (ЕС, Австралия, Россия). Наиболее перспективным и масштабным направлением развития смарт-контрактов является Интернет вещей (IoT), так как именно они сделают возможным «общение» между собой миллиардов устройств, подключенных к Интернету.

Смарт контракты часто лежат в основе **децентрализованных приложений** ( DApp). DAPP состоит из backend-кода, работающего в децентрализованной пиринговой сети. DAPP может также иметь пользовательский интерфейс, создаваемый frontend-кодом, который обращается к бэкенду. Для функционирования DAPP не требуется центральный сервер: взаимодействие между пользователями и приложениями осуществляется напрямую.

Изображение выглядит как стол, компьютер, небо, внутренний

Автоматически созданное описание

Поскольку концепция DApp все еще находится в зачаточном состоянии, ей сложно дать четкое определение. Тем не менее существуют общие признаки DApp:

1. Открытый исходный код. В идеале такие приложения должны будут регулироваться автономно, а все изменения должны будут приниматься консенсусом. Кодовая база должна быть доступна для проверки.

2. Децентрализованность данных и back-end кода. Все записи о работе приложения должны храниться в общедоступной и децентрализованной цепочке блоков, чтобы избежать ошибок централизации

3. Вознаграждение валидаторов блокчейна.

4. Протокол. Криптографический алгоритм, который согласуется сообществом.

5. DApp использует технологию блокчейна и работает в одноранговой сети.

С точки зрения архитектуры возможно создание трех типов Dapp:

Тип I

DApps со своим собственным блокчейном, являющиеся наиболее сложной формой приложений с точки зрения разработки. Примером могут служить всем известные криптовалюты Bitcoin, Ethereum и другие альткойны. Криптовалюта как раз и является DApps: Ether является DApp для блокчейна Ethereum и т.д.

Тип II

DApp, которые работают на существующих блокчейнах - это в основном приложения, работающие в сети Ethereum: Cryptokitties и др.

Тип III

DApp, работающие на DApp типа II, - это приложения, работающие по протоколу, который работает в сети блокчейн, например, Ethereum. Примером типа III является сеть SAFE (использующая SafeCoins), которая использует Omni Protocol на блокчейне биткойнов.

Большинство DApp не добавляют функциональности к существующим блокчейнам, поскольку они получают свои собственные функциональные возможности благодаря блокчейну: безопасность, доверие и т.д. По сути это просто приложения, которые взаимодействуют с блокчейном и хранят кодовую базу децентрализованно в блокчейн.

В настоящее время выделены три основных направления в разработке DAPP:

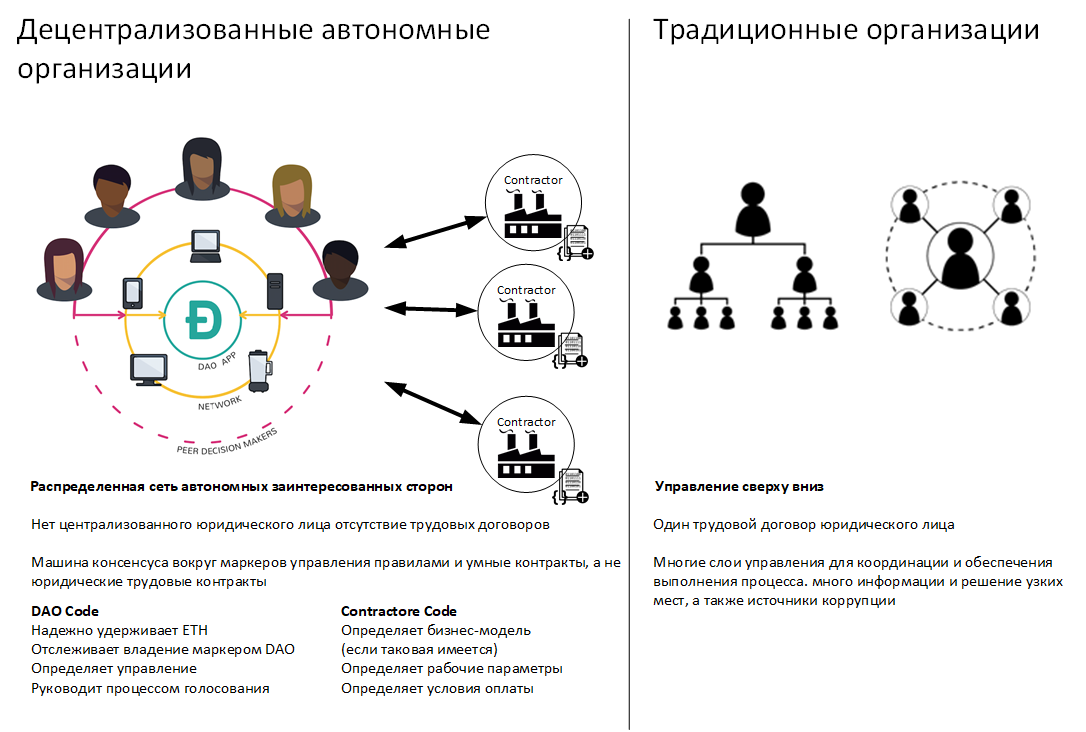
1. Приложения для управления деньгами: пользователи могут проводить транзакции друг с другом в сети блокчейна, используя её внутреннюю валюту. Такие DAPP обычно имеют собственные блокчейны, и мы часто называем их криптовалютами (например, Биткойн).
2. Приложения, интегрирующие деньги с внешними событиями в реальном мире: например, логистическая компания может с помощью местонахождения чипа радиочастотной идентификации определять, достиг ли груз порта, и только тогда оплачивать перевозку. Это может даже осуществляться с помощью средств в блокчейне, без человеческого вмешательства, если покупатель и продавец заключат смарт-контракт.
3. Децентрализованные автономные организации (DAO): децентрализованные, не имеющие руководителя организации на блокчейне. Они с самого своего создания работают согласно программно заданным правилам, определяющим, кто может быть их участниками, как участники могут голосовать, каким бизнесом или какой деятельностью они могут заниматься, и как происходит обмен токенами, средствами или стоимостью. После запуска DAO работают автономно согласно своим правилам. Их участники могут находиться в любой точке земного шара.

Фактически самым первым DApp является Bitcoin. Наиболее часто используемой платформой для создания DApp является Ethereum. Пример архитектуры DApp на Ethereum схематично можно выразить в виде

**DApp=Front-end(HTML+CSS+JS) Web page + Back-end(Solidity Smart contract) Programming code + Server(TestRPC) Private Blockchain/Dummy Network in Ethereum platform**

На базе всех трех уровней (блокчейн+смарт контракты+ децентрализованные приложения) возможно создание четвертого уровня – **децентрализованной автономной организации (DAO)**.

DAO можно рассматривать как наиболее сложную форму смарт-контракта, где устав децентрализованной организации встроен в код смарт-контракта с использованием сложных правил управления токенами.



Это сущность, которая расположена в Интернете и существует автономно, но в то же время сильно зависимая от людей в плане выполнения определенных задач, которые сама не может выполнить автоматически.

Существует несколько ключевых компонентов DAO:

1. Токены: для того, чтобы существовать, DAO требуется внутренняя ценность, которая может быть использована в качестве механизма вознаграждения за определенные действия. Финансирование происходит непосредственно после создания организации. DAO не имеют ни иерархической структуры, ни ключевых фигур, ни руководства.
2. Автономность: однажды развернутая сущность не зависит от своих создателей и не может быть подвержена влиянию внешних сил. DAO имеют открытый исходный код. Записи о финансовых операциях и правила программы DAO хранятся в блокчейне. Такой подход устраняет необходимость вовлечения доверенной третьей стороны в финансовую транзакцию.
3. Консенсус: для того, чтобы вывести или перевести средства из DAO, большинство заинтересованных сторон (этот процент может быть указан в коде DAO) должны согласовать решение. Даже если в коде обнаружены ошибки, их нельзя исправить, пока не будет проведена процедура голосования, и большинство избирателей согласится с этим.
4. Подрядчики: DAO не может создавать продукт, писать код или разрабатывать оборудование. Ему нужен подрядчик для достижения своих целей. Подрядчики назначаются путем голосования владельцев токенов.
5. Предложения: Предложения являются основным способом принятия решений в DAO. Чтобы люди не перегружали сеть предложениями или спамом, DAO часто запрашивает депозит, необходимый для внесения предложений.
6. Голосование: Все предложения после подачи проходят процедуру голосования. DAO позволяют людям обмениваться экономической ценностью с любым человеком в мире без участия посредника.

DAO явно отличаются от других типов организаций. Традиционно любая организация является централизованной, независимо от того, автономна она или нет. Основная причина, по которой люди создали DAO, заключается в использовании функционала организации без бюрократических проволочек и недостатков, связанных с управлением. С DAO нет необходимости тратить время и силы на создание посредника для управления организацией; организация делает эту работу сама.

Глава 3. Блокчейн Биткойна

Блокчейн Биткойна навсегда вошел в историю информационных технологий как революционная идея системы, способной функционировать при полном отсутствии доверия между ее участниками. Все взаимодействия основаны на строгих математических действиях, никакого человеческого фактора. Биткойн положил основу развития технологии блокчейна и децентрализованных приложений. Первым прототипом смарт-контракта на блокчейне нужно считать Bitcoin Script – stack-based не Тьюринг полный язык в сети Bitcoin. Вот пример программы на этом языке



Основным отличием Биткойна является то, что в нем не существует аккаунтов в их традиционном понимании. Вместо них используются входы (inputs) и выходы (outputs). При совершении транзакции (создании выхода) необходима ссылка на транзакцию получения (входа). Bitcoin Script определяет, в каком случае тот или иной выход может быть потрачен.

Inputs — это транзакции, на которые вы "ссылаетесь". Представим, что на ваш адрес X когда-то было отправлено три транзакции:

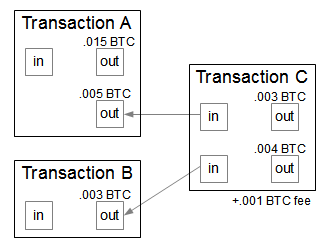
TXN\_ID — 777777, VALUE — 10 BTC

TXN\_ID — 888888, VALUE — 50 BTC

TXN\_ID — 999999, VALUE — 100 BTC

Если вам нужно потратить, например, 60 BTC, то вы можете сослаться на транзакцию 999999, или сразу на две транзакции: 777777 и 888888. Если Вам нужно потратить, например, 8 ВТС, то создается транзакция с одним входом 777777 и двумя выходами (outputs) на 8 ВТС в адрес получателя и на 2 ВТС обратно на свой адрес. Выходов также может быть несколько, и каждому из них указывается своя сумма.

На картинке ниже создается новая транзакция C, которая ссылается на два выхода — A и B. В результате на входе у транзакции получается 0.008 BTC, которые потом разделяются на два выхода — на первый адрес отправляется 0.003 BTC, а на второй 0.004 BTC. При создании транзакции, в которой сумма выходов меньше суммы входов, разница отправляется майнеру, записавшему эту транзакцию в блок. Эта разница называется transaction fee или комиссией за транзакцию.



Майнеры уделяют большое внимание сумме комиссии и обрабатывают в первую очередь те транзакции, комиссия которых выше.

Каким образом можно осуществить майнинг в блокчейне Bitcoin?

Для успешного майнинга необходимо осуществить несколько действий.

1. Пользователь при отправке токена или криптовалюты подписывает свою транзакцию из принадлежащего ему приложения-криптокошелька.

2. Транзакция отправляется в пул неподтвержденных транзакций (транзакций ожидающих обработки)и ожидает там до тех пор, пока майнер не выберет ее для майнинга.

3. Майнер включает транзакцию в блок, максимальный размер которого составляет 1 МБ. Хочется отметить, что одну и ту же транзакцию могут выбрать несколько майнеров.

4. Перед тем, как добавить блок в цепочку блоков майнеры также подписывают его. Эта подпись создается путем решения сложной математической задачи, уникальной для каждого блока транзакций. Каждый блок имеет свою математическую задачу, поэтому каждый майнер будет работать над своей проблемой, уникальной для каждого блока.

5: Майнер, который первым находит подходящую подпись для своего блока, передает этот блок и свою подпись всем остальным майнерам.

6: Остальны майнеры проверяют легитимность подписи, беря строку данных транслируемого блока и хэшируя ее, чтобы увидеть, действительно ли выходной хэш соответствует включенной подписи. Если он действителен, другие майнеры подтвердят его действительность и согласятся с тем, что блок может быть добавлен в цепочку блоков, т.е. достигают консенсуса. Подпись является «доказательством» выполненной работы (вычислительной мощности, которая была потрачена). Теперь блок можно добавить в цепочку блоков и распределить по всем остальным узлам сети. Другие узлы примут блок и сохранят его в своих данных транзакций, если транзакции внутри блока правильно соответствуют текущим балансам кошелька (истории транзакций) на тот момент времени.

Действие 7: После того, как блок был добавлен в цепочку, каждый второй блок, добавленный поверх него, считается «подтверждением» для этого блока.

Когда один из узлов добавляет новый блок информации к общему блокчейну, остальные узлы должны удостовериться в том, что проведенная транзакция является валидной (например, не происходит двойная трата) путем проверки условий, установленных в [скрипте](http://cryptowiki.net/index.php?title=%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82(Bitcoin)). Скрипт в данном контексте – программируемые булевые функции. В их самой распространенной форме они проверяют цифровую подпись с помощью открытого ключа. Так как блокчейн защищен от изменений уже подтвержденных блоков, подделка одной транзакции, находящейся уже в блокчейне, повлечет за собой изменение всех дальнейших частей блокчейна.

При проверке корректности формата транзакции ведутся следующие проверки:

- Проверка синтаксиса

- Размер транзакции в байтах не должен превышать установленной границы

- Сумма входов >= сумма выходов

- Проверка, что скрипты транзакций, на которые ссылается транзакция, дают значение "истина"

Если транзакция прошла все проверки, то она добавляется в пул транзакций, которые еще не подтверждены и остается дождаться, когда транзакция будет занесена в блок, добавленный к блокчейну.

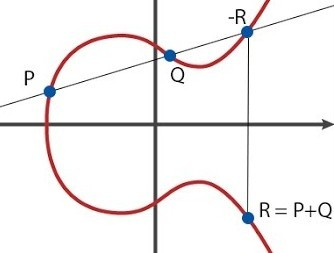
Добавление нового блока транзакций в блокчейн требует решение криптографический задачи большой вычислительной сложности. В том случае, когда в блокчейн разветвляется (после i-ого блока добавляются сразу два противоречащих блока), то участок цепи выбирается тот, на который было потрачено больше вычислительной мощности, в приведенном примере валидной цепочкой выбирается самая длинная.

Как только новая транзакция занесена в блокчейн, ее выходы могут быть использованы в качестве входов. Для таких, пока еще не использованных выходов, существует специальное название — UTXO (unspent transaction output). Каждый выход может быть использован в качестве входа только один раз, поэтому на практике интерес представляют именно не использованные, а уже использованные хранятся в целях безопасности системы. Массив неиспользованных выходов записывается в виде UTXO pool или UTXO set.

Чтобы осуществлять транзакции в Биткойне, пользователю необходимы публичный, приватный ключи и адрес (хэш от публичного ключа). Для того, чтобы понять, как они создаются, нужно вспомнить, что в основе биткойна лежат криптографические алгоритмы . В частности, алгоритм ECDSA — Elliptic Curve Digital Signature Algorithm, который использует эллиптические кривые (elliptic curve) и конечные поля (finite field) для подписи данных, чтобы третья сторона могла подтвердить аутентичность подписи, исключив возможность её подделки. В ECDSA для подписи и верификации используются разные процедуры, состоящие из нескольких арифметических операций.

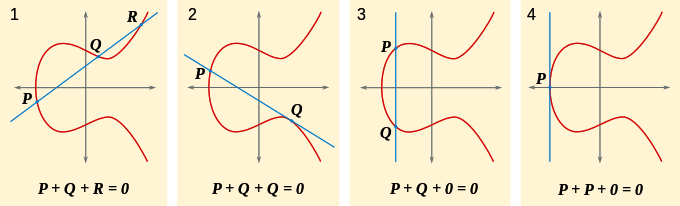
Если взять упрощенное определение эллиптической кривой, то можно сказать, *что эллиптическая кривая - это внешне довольно простая функция, как правило, записываемая в виде так называемой формы Вейерштрасса:*

Пусть есть две точки . Их суммой называется точка , которая в простейшем случае определяется следующим образом: проведем прямую через P и Q — она пересечет кривую  в единственной точке, назовем ее -R. Поменяв y координату точки -R на противоположную по знаку, мы получим точку R, которую и будем называть суммой P и Q, то есть .

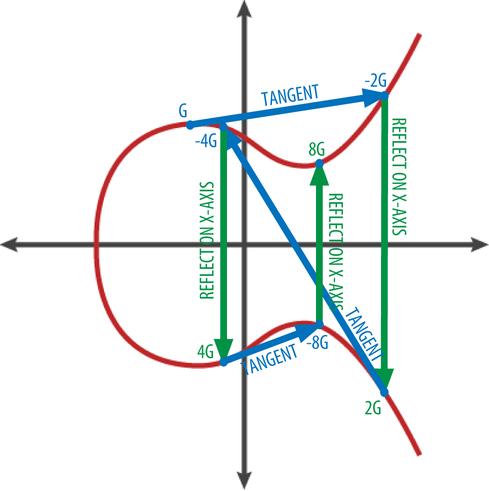


Необходимо *ввести* такую операцию сложения — если складывать точки в привычном понимании, то есть складывая соответствующие координаты, то получится совсем другая точка , которая, скорее всего, не имеет ничего общего с R или -R и вообще не лежит на кривой  .

Если, например, провести прямую через две точки, имеющие координаты вида и , то прямая, проходящая через них, будет параллельна оси ординат (третий график). В этом случае отсутствует третье пересечение с кривой , которое мы называли -R . Для того, чтобы избежать этого, введем так называемую точку в бесконечности (point of infinity), обозначаемую обычно O. Таким образом получаем, что в случае отсутствия пересечения .

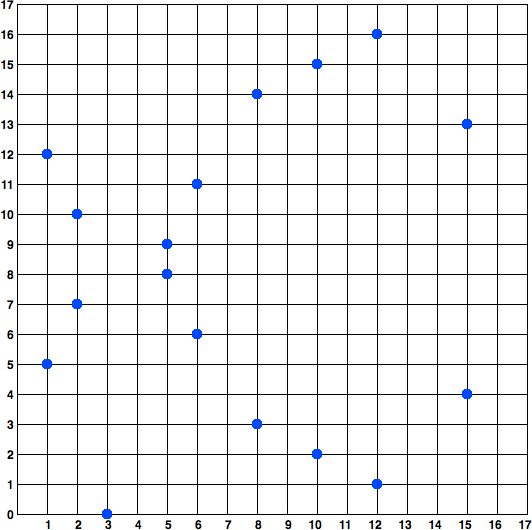


Можно ввести операцию умножения точки на какое-то число N. В результате получим новую точку K=G\*k, то есть K=G+G+…+G k раз.



В ECC используется точно такая же кривая, только рассматриваемая над некоторым конечным полем p — простое число. То есть

Все названные свойства (сложение, умножение, точка в бесконечности) также применимы и для такой функции, хотя, если попробовать ее нарисовать, то напоминать привычную эллиптическую кривую она будет лишь отдаленно. А понятие «касательной к функции в точке» вообще потеряет всякий смысл. Вот пример функции для :



Чем большее р мы возьмем, тем больше точек получим на графике.

В протоколе биткойна зафиксирован набор параметров для эллиптической кривой и её конечного поля, чтобы каждый пользователь использовал строго определенный набор уравнений. Среди зафиксированных параметров выделяют уравнение кривой (equation), значение модуля поля (prime modulo), базовую точку на кривой (base point) и порядок базовой точки (order).

В случае Биткойна используется кривая SECP256k1. Она имеет вид и рассматривается над полем , где р — очень большое простое число, а именно .

Так же для SECP256k1 определена так называемая base point, она же generator point — это просто точка, как правило, обозначаемая G , лежащая на данной кривой. Она используется для создания публичного ключа.

Базовая точка:

**04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798** 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

Жирным шрифтом выделена координата X в шестнадцатеричной записи. За ней сразу следует координата Y.

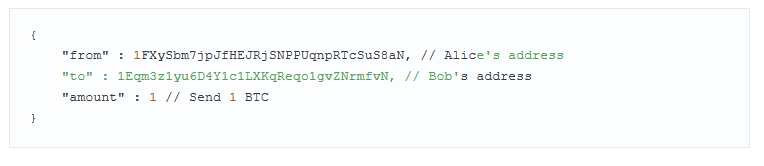
Этот набор параметров для эллиптической кривой известен как secp256k1 и является частью семейства стандартов SEC (Standards for Efficient Cryptography), предлагаемых для использования в криптографии. В биткойне кривая secp256k1 используется совместно с алгоритмом цифровой подписи ECDSA (elliptic curve digital signature algorithm). В ECDSA секретный ключ — это случайное число между единицей и значением порядка.

В блокчейне биткойна используется два типа ключей: приватный и публичный.

*Публичный ключ* — это уникальный биткоин-адрес, который используется в технологии блокчейн. Его может видеть каждый участник сети. Публичный ключ выглядит как строка из цифр и букв, созданная на основе приватного ключа. Она идентифицирует получателя и отправителя средств.

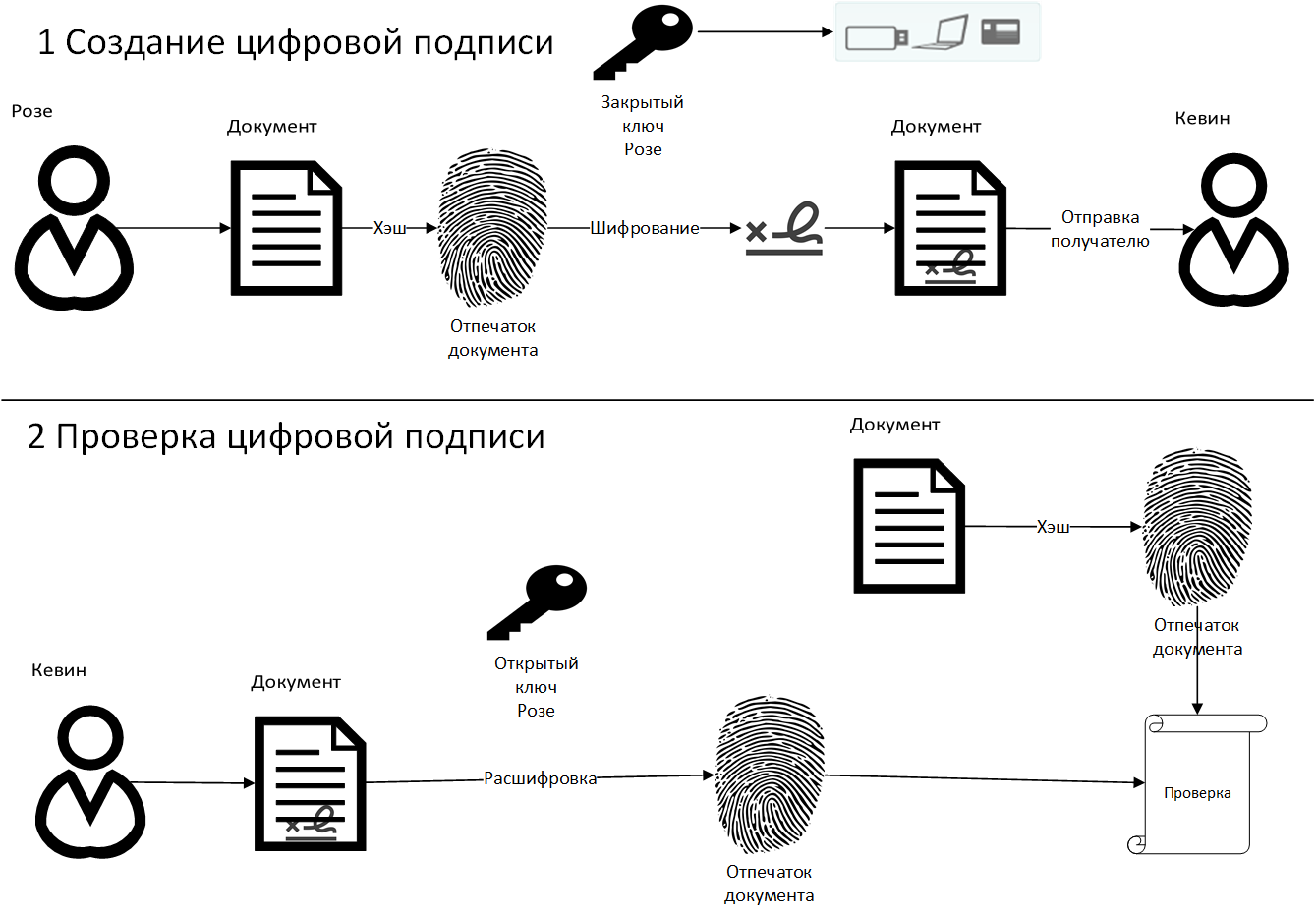
*Приватный ключ* — это один из важнейших инструментов контроля кошелька с криптовалютой. Только с закрытым ключом, который соответствует публичному ключу (биткоин-адресу), пользователь может перевести средства другому пользователю или разблокировать их.

Рассмотрим пример применения ключей. Алиса хочет перевести 1 BTC Бобу. Для этого она создает сообщение типа:



Потом Алиса берет свой приватный ключ, хэш сообщения и функцию вида . На выходе она получает подпись своего сообщения — в случае ECDSA это будет пара целых чисел, для других алгоритмов подпись может выглядеть по другому. После этого она рассылает всем участникам сети исходное сообщение, подпись и свой публичный ключ.

В результате, каждый пользователь при желании сможет взять эти сообщение, подпись и ключ, а также функцию вида и проверить, действительно ли владелец приватного ключа подписывал это сообщение или нет.



Пусть злоумышленник перехватил сообщение Алисы и что-то в нем изменил, тогда в этом случае проверка подписи на валидность покажет, что сообщение было изменено. Так как биткойн – распределенная сеть, никто заранее не знает к кому попадет транзакция, но может быть уверен, что транзакция, подписанная приватным ключом, не может быть изменена.

Как уже было сказано выше, приватный ключ открывает доступ к пользовательским цифровым активам. Поэтому важно подобрать надежный код. Неправильный подбор ключа приведет к краже средства пользователя

В протоколе Биткойна *приватный ключ* - некоторое натуральное 256 – битное число, т.е. число от 1 до 2256. В связи с тем, что 2256 - огромно, подобрать ключ методом перебора невозможно.

*Приватный и публичный ключ тесно взаимосвязаны. Операция получения публичного ключа в Биткойне однозначна и конкретному приватному ключу всегда соответствует один единственный публичный ключ. Обратная операция является вычислительно трудной и, в общем случае, получить приватный ключ из публичного можно только полным перебором первого*.

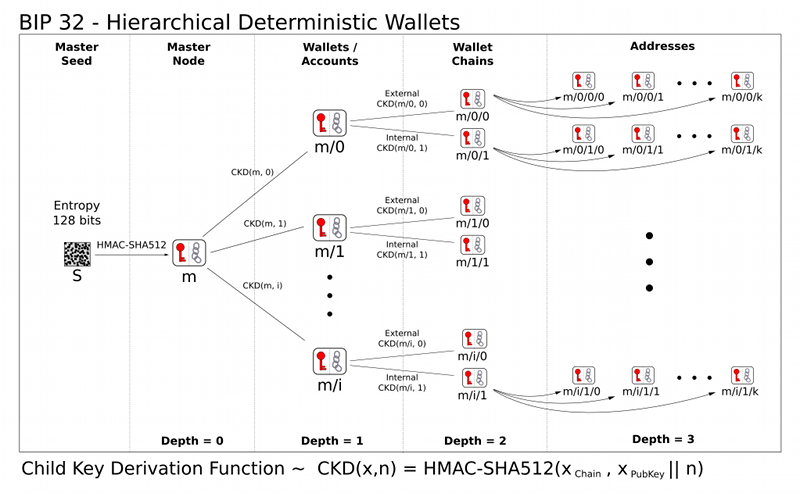
Для максимальной анонимности транзакций в блокчейнах была разработана криптографическая процедура, называемая “иерархически детерминированный кошелёк”, также известная как функция BIP32. Она позволяет создавать Мастер-ключ, из которого в “детерминистическом” порядке развёртываются все остальные приватные ключи, связаные друг с другом так, что другие пользователи не могут проследить эту связь. Такие адреса именуются «дочерними адресами».

Существует стандарт для разворачивания Мастер-ключа из SEED (мнемонической фразы, хранящей информацию для восстановления кошелька) в 12 или 24 слова (BIP39), а также стандарт для создания ключей с поддержкой нескольких аккаунтов (BIP44). Функция поддержки нескольких аккаунтов означает, что Мастер-ключ А разворачивает аккаунты *а, в, с* и так далее в неограниченном количестве, а внутри, скажем аккаунта *а*, он генерирует ключи. При их создании пользователь получает один адрес, который имеет один приватный ключ, именуемый мастер-ключом. Объединив его с хеш-функциями, пользователь может получить неограниченное количество других адресов, которые связаны друг с другом так, что другие пользователи не могут проследить эту связь. Такие адреса именуются «дочерними адресами».

В то время, как ключевая фраза (SEED) и Мастер-ключ одинаково стандартизированы, существует два способа восстанавливать ключи и адреса кошелька, по стандарту BIP32 и BIP44.

Некоторые кошельки вроде Ledger используют метод с фразой из 24 слов, тогда как другие, вроде Exodus, будут использовать SEED из 12 слов. SEED из 12 слов несовместим с кошельком, поддерживающим SEED из 24 слов. Так же кошельки используют разный словарный набор, и разную реализация путей развёртывания; например, Exodus может восстановить лишь первые 4 адреса, созданные с помощью Мастерключа, а на Coin.Space не было возможности восстановить адреса, созданные при помощи другого кошелька, и так далее.

Есть сайт с генератором BIP39, вроде https://iancoleman.io/bip39/ предлагаемого Иэном Колеманом, или https://bip32jp.github.io/english/ генератором BIP32. Здесь, можно просто ввести SEED-ключ, и получить всю необходимую для восстановления кошелька информацию. Первое, приложение отобразит приватные ключи и адреса. Их можно использовать для восстановления адресов в Bitcoin Core, Electrum и других кошельках, позволяющих импортировать приватные ключи. Второе, показывает “Расширенный Ключ BIP32” (Extended Key) и Мастер-Ключ, который можно использовать для восстановления кошелька при помощи, например, Electrum.

[](https://github.com/bitcoin/bips/blob/master/bip-0032/derivation.png)

https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

Пусть — k - наш приватный ключ, тогда G — base point - публичный ключ . То есть, фактически, публичный ключ — это некоторая точка , лежащая на кривой SECP256k1.

В Биткойне используется формат кодирования Base58. Он осуществляет баланс между компактной производительностью, удобочитаемостью, четкостью и предотвращением ошибок. Base58 - это подмножество Base64, которое использует строчные и прописные буквы и цифры, но без некоторых символов, которые часто ошибочно принимаются друг за друга и могут быть идентичными в некоторых шрифтах. В частности, Base58 - это Base64 без цифр 0 (ноль), O (заглавная буква O), l (маленькая L), I (большая i) и символов «\ +» и «/». Или, проще говоря, это набор заглавных и прописных букв и цифр без четырех (0, O, L, I), упомянутых выше. В отличие от Base64, цифры кодирования плохо совпадают с байтовыми границами исходные данные. По этой причине метод хорошо подходит для кодирования больших целых чисел, но не предназначен для кодирования более длинных частей двоичных данных. Фактический порядок букв в алфавите зависит от приложения, что является причиной того, что одного термина «Base58» недостаточно для полного описания формата. Вариант Base56 исключает 1 (один) и o (строчные o) по сравнению с Base 58.

Base58Check - это формат кодирования Base58, который однозначно кодирует тип данных в первых нескольких символах и включает код обнаружения ошибок в последних нескольких символах.

Изображение выглядит как снимок экрана

Автоматически созданное описание

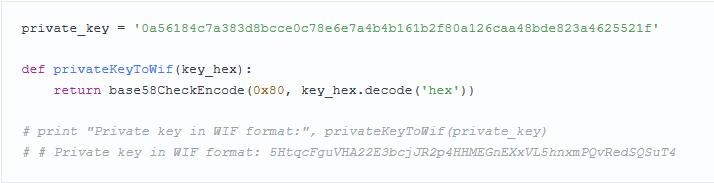
Wallet Import Format (WIF, также известный как Wallet Export Format) - это способ кодирования приватного ключа ECDSA для упрощения копирования. Кодирование приватного ключа осуществляется следующим образом. Берется приватный ключ



Конвертируем его в Base58Check и добавляем впереди 0х80 байт. Получаем



Программно кодировка выглядит так



*Публичный ключ.* Если не вдаваться в сложные вопросы эллиптической криптографии, то можно просто сказать, что открытый (публичный) ключ на самом деле является лишь координатами на кривой системы Bitcoin, рассчитанными путем умножения точки генератора на номер закрытого (частного) ключа.

Формат хранения публичного ключа бывает *compressed* или *uncompressed*. Суть его в следующем: публичный ключ — это точка на кривой, то есть пара чисел удовлетворяющая уравнению

А значит можно записать только Х координату и если нам понадобится Y координата — просто решаем уравнение. Тем самым мы уменьшаем размер публичного ключа почти на 50%. Если точка лежит на кривой, то для ее Х координаты очевидно существует два решения такого уравнения. Воспользуемся следующим свойством: если для Х координаты существуют решения уравнения, то одна из точек будет иметь четную Y координату, а вторая — нечетную.

В первом случае используется префикс 0x02, во втором — 0x03. Вот иллюстрация процесса

Изображение выглядит как текст

Автоматически созданное описание

*Адрес* получается из публичного ключа однозначным образом, обратную операцию провести невозможно из-за применения стойких хэш-функций RIPEMD160 и SHA256. Вот алгоритм перевода публичного ключа в адрес:

Возьмем приватный ключ, например

45b0c38fa54766354cf3409d38b873255dfa9ed3407a542ba48eb9cab9dfca67

Получим из него публичный ключ в uncompressed формате, в данном случае это

04162ebcd38c90b56fbdb4b0390695afb471c944a6003cb334bbf030a89c42b584f089012beb4842483692bdff9fcab8676fed42c47bffb081001209079bbcb8db.

Считаем RIPEMD160(SHA256(public\_key)), получается

5879DB1D96FC29B2A6BDC593E67EDD2C5876F64C

Переводим результат в Base58Check с префиксом 0x00 — 17JdJpDyu3tB5GD3jwZP784W5KbRdfb84X. Это и есть адрес.

Когда пара секретный/публичный ключ получена, её можно использовать для подписи данных. Эти данные могут быть любой длины. Обычно первым шагом выполняется хеширование данных с целью получения уникального значения с числом битов, равным битности порядка кривой (256). После хеширования, алгоритм подписи данных z выглядит следующим образом. Здесь, G — базовая точка, n — порядок, а d — секретный ключ.

Выбирается некоторое целое k в пределах от 1 до n-1

Рассчитывается точка (х, у) = k \* G с использованием скалярного умножения

Находится r = х mod n. Если r = 0, то возврат к шагу 1

Находится s = (z + r \* d) / k mod n. Если s = 0, то возврат к шагу 1

Полученная пара (r, s) является нашей подписью

После получения данных и подписи к ним, третья сторона, зная публичный ключ, может их верифицировать. Шаги для проверки подписи такие (Q — открытый ключ):

Проверка, что и r, и s находятся в диапазоне от 1 до n-1

Рассчитывается w = s-1 mod n

Рассчитывается u = z \* w mod n

Рассчитывается v = r \* w mod n

Рассчитывается точка (x, y) = uG + vQ

Если r = x mod n, то подпись верна, иначе — недействительна

В самом деле,

uG + vQ = u + vdG = (u + vd)G = (zs-1 + rds-1)G = (z + rd) s-1G = kG

Последнее равенство использует определение s на этапе создания подписи.

Безопасность ECDSA связана со сложностью задачи поиска секретного ключа, описанной выше. Помимо этого, безопасность исходной схемы зависит от «случайности» выбора k при создании подписи. Если одно и то же значение k использовать более одного раза, то из подписей можно извлечь секретный ключ. Поэтому современные реализации ECDSA, в том числе используемые в большинстве биткойн-кошельков, генерируют k детерминировано на основе секретного ключа и подписываемого сообщения.

Глава 4. Платформы для разработки децентрализованных приложений

Как уже говорилось выше, в большинстве случаев для того, чтобы написать DApp, нет необходимости разрабатывать и разворачивать собственный блокчейн. Это очень долго, дорого и сложно. Достаточно лишь выбрать существующую платформу и научиться ее использовать. В настоящее время разрабатывается более 20 платформ для написания DApp и практически у каждой есть свой собственный язык программирования. Ivy-lang, Plutus, Solidity, Scrypto, Michelson, Hoon, Rust - это лишь часть языков, используемых на платформах для разработки DApp. Для того, чтобы разобраться в каждом из них, вам потребуется уйма времени. Ниже в таблице вы можете найти небольшой сравнительный анализ платформ, который поможет вам составить более четкое представление об их количестве, особенностях и стадиях разработки. Некоторые разработчики многообещающих платформ, к сожалению, так и не смогли подготовить достойный продукт, некоторые платформы имеют схожие черты, по некоторым недостаточно информации в сети. В любом случае, это направление развивается и значительно упрощает разработку Dapp.

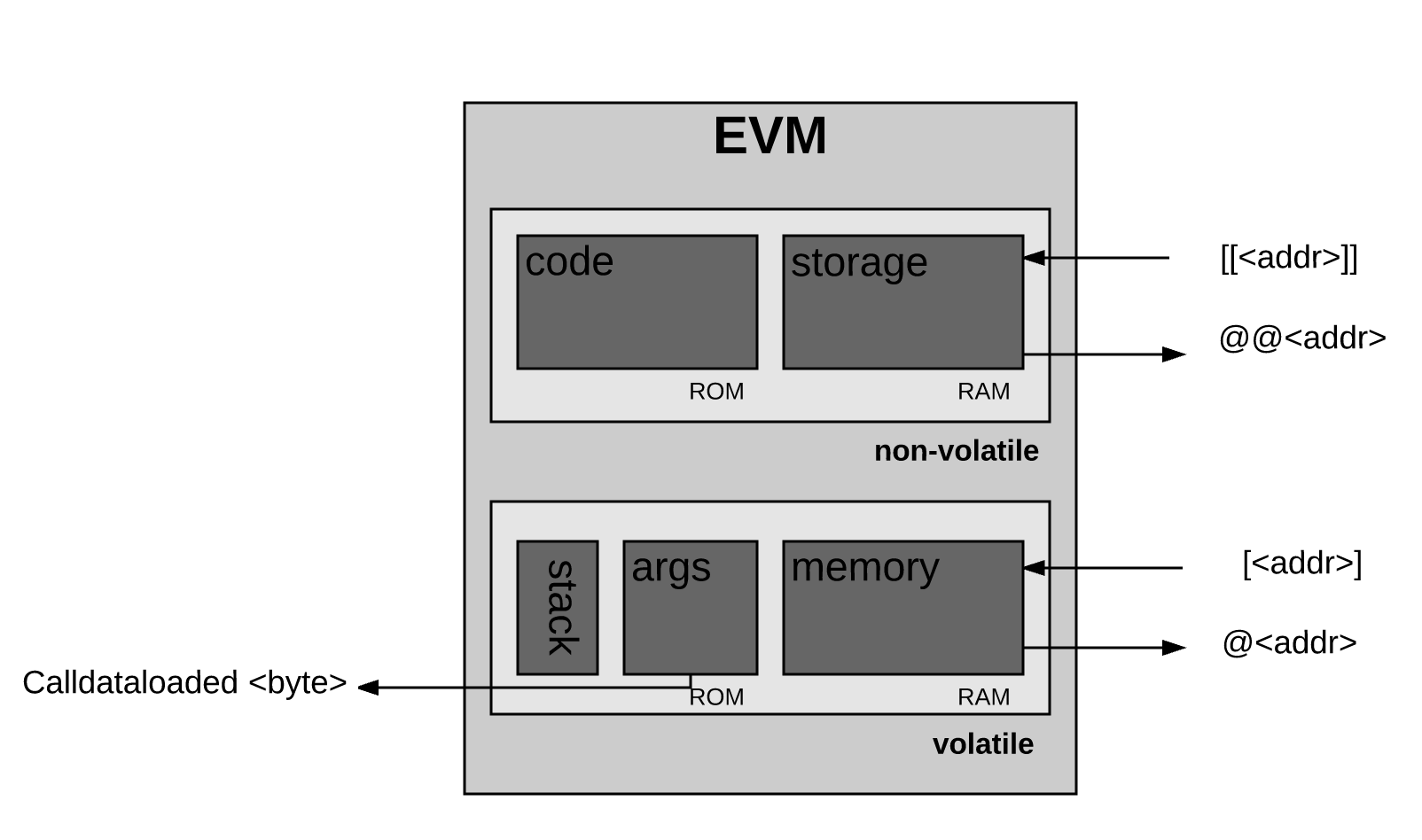


<https://github.com/Overtorment/awesome-smart-contracts>

Мы не будем разбирать каждую платформу, а выберем две основных наиболее часто используемых: Ethereum и Hyperledger Fabric. Давайте разберемся в их возможностях и выясним, как они работают.

**Ethereum.** Большинство разработчиков ассоциируют смарт-контракты именно с этой платформой и языком Solidity. Ethereum – это open-source блокчейн платформа для создания децентрализованных приложений со своей собственной валютой Ether. До создания Ethereum приложения на блокчейне были предназначены для выполнения очень ограниченного набора операций. Например, биткойны и другие криптовалюты были разработаны исключительно для работы в качестве одноранговых цифровых валют.

С течением времени разработчики столкнулись со следующей проблемой: возникла необходимость либо расширить функционал Биткойна или других существующих платформ, что на практике очень сложно осуществимо и на это потребовалось бы много времени, либо разработать совершенно новую платформу на блокчейне. Осознав эти трудности, Виталик Бутерин разработал платформу Ethereum. Вместо того, чтобы создавать новый блокчейн для каждого приложения, Ethereum позволяет разрабатывать тысячи различных приложений на одной платформе. В WhitePaper Ethereum написано, что целью платформы является создание альтернативного протокола для разработки децентрализованных приложений с акцентом на время разработки, безопасность и масштабирование. Обладая собственным языком Solidity, Ethereum позволяет разработчикам создавать умные контакты с помощью полной по Тьюрингу виртуальной машины Ethereum (EVM), имеющей все особенности стэковой архитектуры с максимальным размером стэка 1024 бит и размером элемента стэка в 256 бит. Виртуальная машина Ethereum имеет три области, в которых она может хранить данные: хранилище, память и стек. Память не является величиной постоянной, в отличие от хранилища, которое не изменяется и является частью системы.

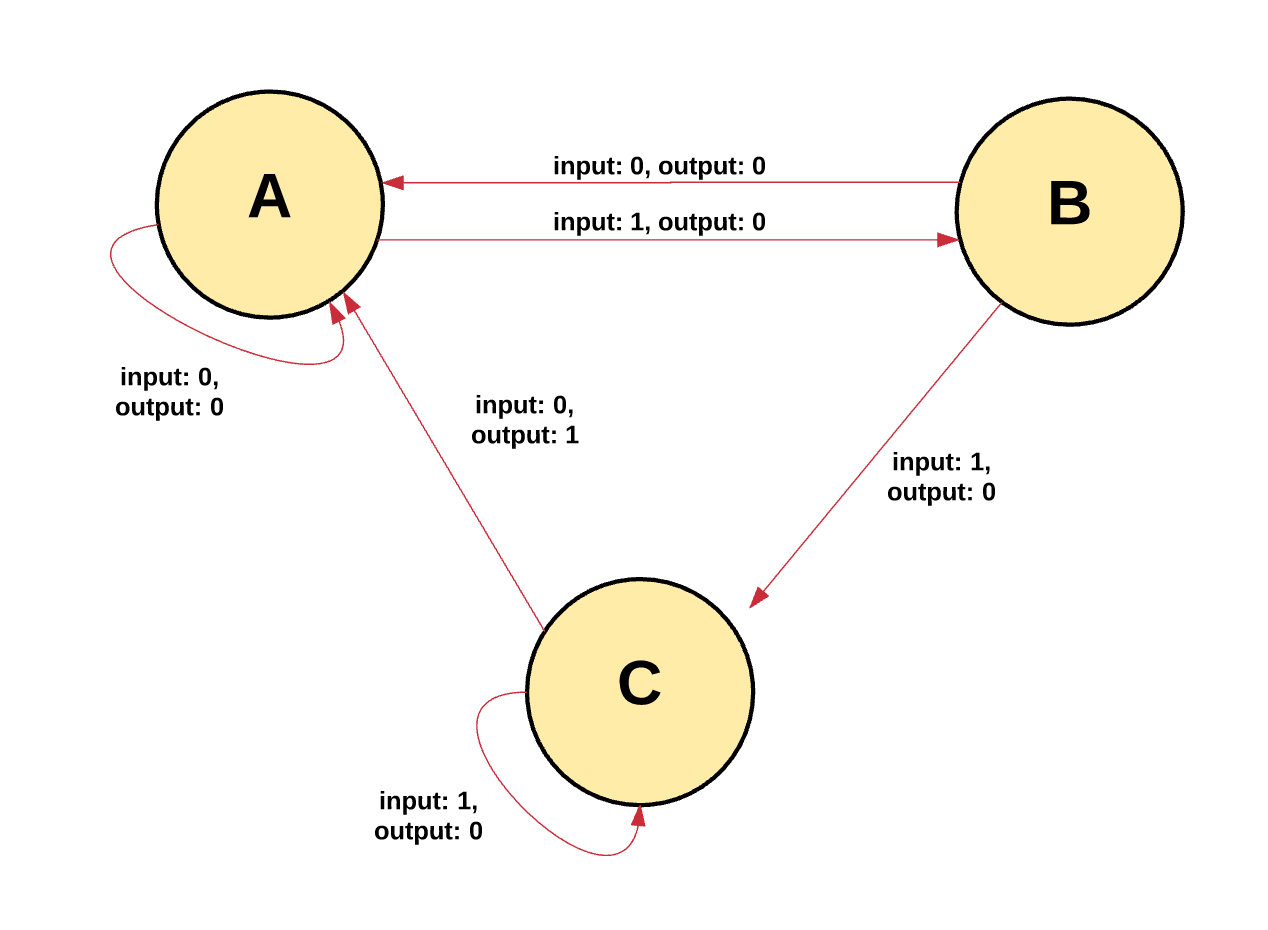


Solidity исполняется в виртуальной машине внутри кажой ноды. Для работы EVM используется виртуальное «горючее» - газ (gas). Перед началом работы EVM проверяет валидность и доступность информации о состоянии системы, достаточности необходимого для транзакции газа, адрес отправителя и инициатора транзакции (могут различаться), входные данные, информацию о выполняемом машинном коде, заголовке блока. После проверки EVM начинает рекурсивное выполнение транзакции: происходит вычисление состояния системы и машины после каждого цикла. Существует три вероятных варианта окончания цикла:

1. Исключение, в результате которого процесс останавливается. Может происходить из-за полного расходования газа, ограничения по стэку, неверных инструкций и т.д.
2. Переход к выполнению следующего цикла
3. Логическое завершение процесса

Имея эти инструменты, разработчики могут создавать Dapp, которые имеют реальные сценарии использования, от управления активами до планирования ресурсов. Также хочется отметить, что язык для написания front end кода может быть любым.

Блокчейн Ethereum является, по сути, системой состояния транзакций. В информатике такое понятие, как «система состояний» или «машина состояний» – это система, которая обрабатывает вводимую информацию и на основании последней преобразуется в новое состояние.



Все процессы в системе состояний Ethereum начинаются с «первоначального состояния» - аналога нулевого состояния, в котором находится машина до момента начала каких-либо действий, связанных с транзакциями. С началом таких действий изначальное состояние заменяется на конечное, отображающее текущее состояние Ethereum.

Технически Ethereum представляет собой стэк из трех слоев. Первый базовый уровень, который обеспечивает всю работу Ethereum, - это сеть компьютеров, которые обрабатывают транзакции и постоянно обновляют общую базу данных (блокчейн Ethereum). Как и любой блокчейн, Ethereum работает на протоколе одноранговой сети. База данных блокчейна Ethereum поддерживается и обновляется многими узлами, подключенными к сети. Каждый узел сети запускает EVM и выполняет одни и те же инструкции. Это сделано для поддержания консенсуса блокчейна. Децентрализованный консенсус дает Ethereum экстремальные уровни отказоустойчивости, обеспечивает нулевое время простоя и делает данные, хранящиеся в блокчейне, навсегда неизменными и устойчивыми к цензуре. По этой причине Ethereum иногда называют «мировым компьютером».

Второй уровень - это программный уровень, который позволяет разработчикам запускать смарт-контракты в блокчейне Ethereum с использованием языка программирования Solidity.

Третий уровень состоит из приложений, которые предлагают различные услуги (от общего управления до управления идентификацией) пользователям Ethereum. Примечательной особенностью этой платформы является то, что благодаря использованию аппаратного и программного уровней Ethereum эти приложения децентрализованы, лишены центральной точки отказа и не останавливаемы.

Большинство новых платформ для разработки децентрализованных приложений взяли за основу архитектуру Ethereum: виртуальная машина + свой язык контрактов.

Изображение выглядит как снимок экрана

Автоматически созданное описание

В отличие от Биткойна, который является просто списком транзакций, основной единицей Ethereum является учетная запись. Блокчейн Ethereum отслеживает состояние каждой учетной записи, и все переходы между состояниями в блокчейне Ethereum являются переносом значений и информации между учетными записями. Существует два типа учетных записей:

* Externally Owned Accounts (EOA) или внешние, которые контролируются закрытыми ключами, находящимися у пользователей. Для внешней учетной записи предусмотрена возможность отправлять сообщения другим внешним учетным записям, а также другим контрактным учетным записям. Для данной цели необходимо создать и зарегистрировать новую транзакцию, используя закрытый ключ. Сообщение между двумя внешними учетными записями является всего лишь значением для передачи. С другой стороны, сообщение, отправленное от внешней учетной записи к контрактной, подразумевает активацию кода контрактной учетной записи, при этом появляется возможность совершения определенных действий (например, с помощью такого сообщения можно переводить токены, записывать значения во встроенную память, создавать токены, выполнять некоторые вычисления, создавать новые контракты и т. д.).
* Contract Accounts или контактные, которые контролируются их контрактным кодом и могут быть «активированы» только EOA (в них выполняются смарт-конртакты). С помощью контрактных учетных записей, в отличие от внешних, самостоятельно инициировать новые транзакции невозможно. Вместо этого с помощью контрактных учетных записей можно только запускать транзакции в ответ на другие полученные транзакции (например, полученные из внешней учетной записи или из другой контрактной учетной записи).

Процедура генерации адреса происходит схожим с генерацией адреса в блокчейне Биткойна образом:

- генерируется хэш открытого ключа по алгоритму keccak-256

- отбрасываются первые 96 битов

- адрес кодируется в шестнадцатеричную строку, состоящую из 40 символов.

Состояние каждой из учетных записей, вне зависимости от их типа, может принимать одно из четырех значений:

balance: общее количество wei, находящихся на балансе данной учетной записи. Каждый эфир, содержит 1018 wei.

nonce: во внешних учетных записях данное число представляет собой количество транзакций, отправленных с адреса учетной записи. В контрактных учетных записях элемент nonce – это количество контрактов, созданных в данной учетной записи.

codeHash: хэш EVM-кода учетной записи. Для контрактных учетных записей данное поле является кодом, который хэшируется и хранится в виде codeHash.

storageRoot: хэш корневого узла префиксного дерева Меркла. Дерево Меркла используется при кодировке хэша содержимого учетных записей, по умолчанию является пустым.

У каждой учетной записи есть постоянное хранилище значений ключей. Кроме того, у каждого аккаунта есть баланс, который можно изменить, посылая транзакции, содержащие эфир. Транзакция является криптографически подписанной частью инструкции, которая сначала задается внешней учетной записью, а затем упорядочивается и передается в блокчейн. Подпись транзакции, как и в блокчейне биткойна, использует алгоритм ECDSA. В Ethereum существует два типа транзакций: отправка сообщения и создание смарт-контракта. Транзакция может включать в себя бинарные данные (которые называются «полезной нагрузкой»), эфир и содержит следующие элементы:

gasPrice – цена газа, измеряемая в Wei.

nonce – количество отправленных отправителем транзакций.

gasLimit – максимальное количество расходуемого газа на транзакцию.

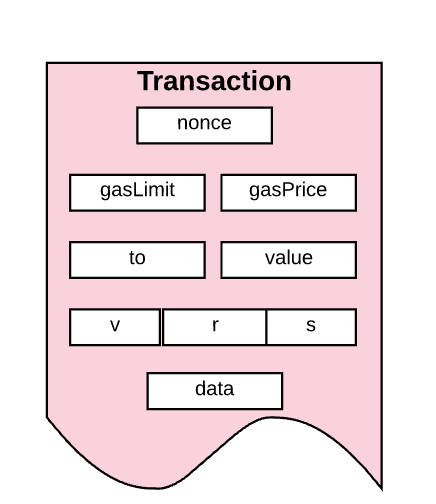
value – количество Wei, передаваемые отправителем получателю. Является стартовым балансом для вновь созданной учетной записи при создании контракта.

to – адрес получателя, может быть пустым, если транзакция создает новый контракт.

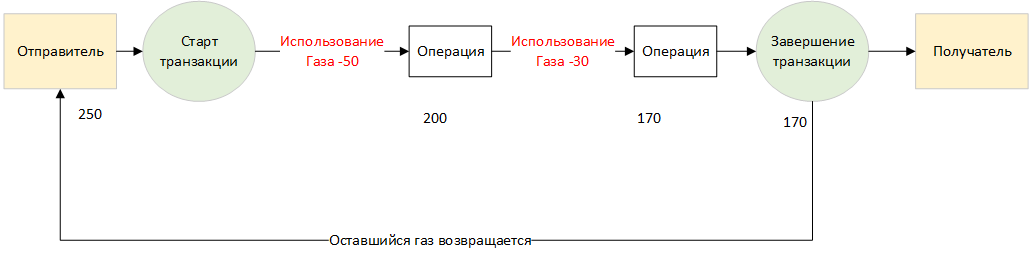
s, v, r – используются для подписи, идентифицирующей отправителя транзакции.

data – это входные данные (параметры) для вызова сообщения.

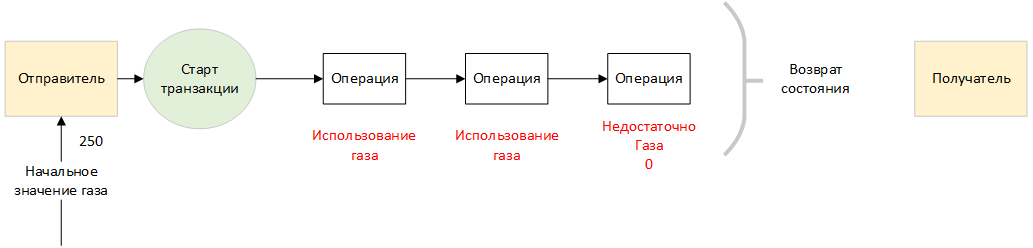
init – фрагмент EVM-кода для инициализации созданной учетной записи, используется только в транзакциях при создании контрактов. Запускается однократно и возвращает тело кода учетной записи.



Для выполнения транзакций в блокчейне Ethereum используется газ (gas), задача которого состоит в ограничении работы. Цена на газ - это значение, установленное создателем транзакции, который должен заплатить gas\_price \* gas заранее с отправляющего транзакцию аккаунта. Каждая операция, которая может быть выполнена посредством транзакции или контракта на платформе Ethereum, стоит определенного количества газа, причем операции, которые требуют больше вычислительных ресурсов, потребляют больше газа, чем операции, которые требуют меньше вычислительных ресурсов. Таким образом газ измеряет, сколько «работы» требуется выполнить действию или набору действий: например, для вычисления одного криптографического хэша Keccak256 потребуется 30 газов каждый раз, когда вычисляется хеш, плюс стоимость еще 6 газов для каждых 256 битов. Данные хешируются. Пока EVM выполняет транзакцию, газ постепенно расходуется согласно определенным правилам. Если после обработки транзакции останется некоторое количество газа, то он будет возвращен отправителю.

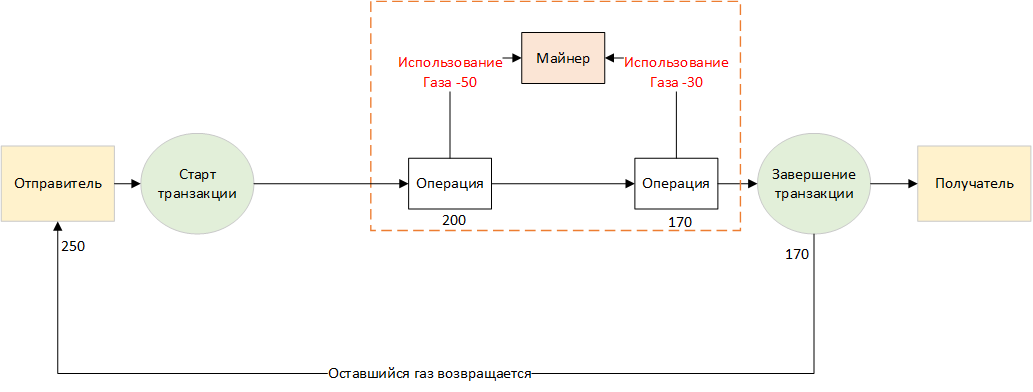


Если газ израсходован в любой точке до окончания транзакции (т.е. значение станет отрицательным), срабатывает исключение из-за газа, которое отменяет все изменения, сделанные в текущем вызове. Таким образом, проведение транзакции прерывается, а любые изменения состояния аннулируются, вследствие чего система возвращает участников сделки в первоначальное состояние. Стоит отметить, что информация о такой неудавшейся транзакции записывается в системе, так что вы можете отследить, проведение каких транзакций было отменено и на каком этапе произошел сбой. Стоит отметить, что в связи с тем, что машиной уже были затрачены определенные усилия для выполнения расчетов, убытки, связанные с расходованием газа, не будут возмещены отправителю.



Причина, по которой газ важен, заключается в том, что он помогает гарантировать отправку соответствующих транзакций в сеть. Требуя оплату за каждую операцию, пользователи гарантируют, что сеть не будет перегружена выполнением большого количества интенсивной работы, которая никому не нужна. Эта стратегия отличается от платы за транзакцию в биткойнах, которая основана только на размере транзакции в килобайтах. Так как Ethereum позволяет выполнять произвольно сложный компьютерный код, короткая длина кода может на самом деле привести к большой вычислительной работе. Поэтому важно оценивать проделанную работу напрямую, а не просто выбирать комиссию в зависимости от продолжительности транзакции или контракта.

Кто же является получателем газа? Поскольку майнеры выполняют расчеты и проверку транзакций, то именно они получают газ, а вернее его эквивалент в Ether, в качестве вознаграждения.



Откуда же берется газ? Несмотря на то, что газ - это единица, в которой можно измерять работу, фактически его не существует. То есть вы не можете владеть 1000 газами. Газ существует только внутри виртуальной машины Ethereum, поскольку подсчитывает, сколько работы выполняется. Когда дело доходит до фактической оплаты за газ, плата за транзакцию взимается в виде определенного количества эфира.

Поначалу это может показаться странным. Почему в операциях стоимость не измеряется напрямую в эфире? Дело в том, что Ether, как и Биткойны, имеет рыночную цену, которая постоянно изменяется. Поэтому пользователям необходимо отделять цену вычислений от цены криптовалюты, чтобы стоимость операции не менялась при каждом колебании рынка.

Как же проводятся транзакции в Ethereum?

Для того, чтобы транзакция была успешно выполнена, она должна соответствовать ряду требований:

1. Она должна соответствовать требованиям RLP - рекурсивной длины префикса (от англ. Recursive Length Prefix), представляющий собой формат данных, применяемый при кодировании вложенных массивов двоичных данных и используемый в Ethereum для упорядочивания объектов.
2. Необходимо наличие валидной подписи транзакции и валидного значения nonce.
3. Лимит газа для транзакции должен быть равным или больше заданного количества газа.
4. Баланс отправителя должен содержать достаточное количество эфира для проведения транзакции

Если все требования успешно выполнены, то начинается выполнение транзакции, в течение которой происходит отслеживание промежуточных значений:

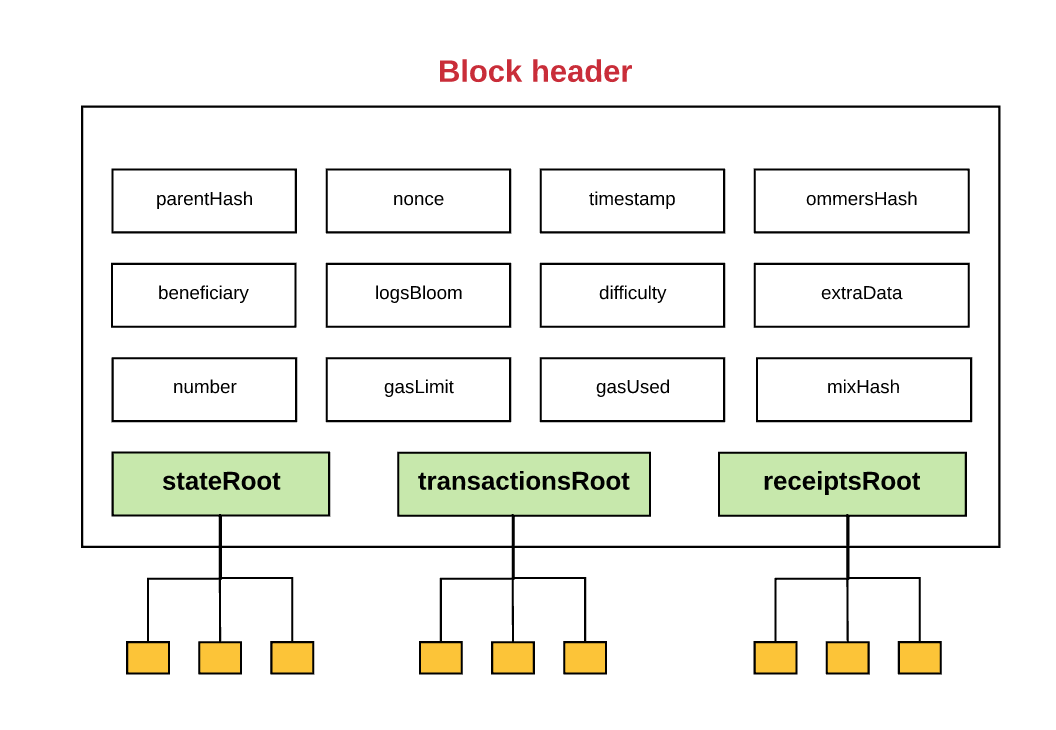
Self-destruct set: набор учетных записей, которые будут удалены по завершении проведения транзакции

Log series: заархивированные и проиндексированные контрольные точки, необходимые для выполнения кода виртуальной машины.

Refund balance: сумму, которая должна быть возвращена отправителю по завершении транзакции.

По окончании транзакции, при условии, что все вышеуказанные требования выполнены, неизрасходованные средства возвращаются отправителю, а информация из Self-destruct set удаляется.

Все транзакции в блокчейнах, как уже было сказано выше, формируются в блоки. Каждый блок в Ethereum имеет «header» - заголовочный файл, в котором хранится хэш корневого узла трех разных структур дерева Меркла: состояние префиксного дерева, транзакции префиксного дерева, страницы приема оплаты (квитанции) для префиксного дерева. Квитанции аналогичны чекам, которые выдают в магазине за покупки. Каждый перевод Ether получает чек, куда входят основные сведения: хэш транзакции, хэш блока, порядковый номер блока, сумма использованного газа, информация о том, какие логи были сформированы в связи с данным переводом.



Заголовок блока – это часть блока, состоящая из:

number – номер текущего блока

gasLimit – лимит газа для данного блока

gasUsed – количество газа, используемого для транзакции

timestamp – временная метка блока, содержащая время добавления блока

beneficiary – адрес учетной записи для оплаты за блок

transactionsRoot – хэш корневого узла префиксного дерева, содержащий все транзакции, которые перечислены в данном блоке

parentHash – хэша заголовка родительского блока (благодаря чему, собственно, блок попадает в цепочку блоков)

ommersHash – хэш текущего списка блоков оммеров (блок, родителем которого является родительский элемент текущего блока)

stateRoot – хэш корневого узла состояния префиксного дерева

receiptsRoot – хэш корневого узла префиксного дерева, содержащий информацию об оплате для всех транзакций, перечисленных в данном блоке

logsBloom – Фильтр Блума, используемый для хранения информации, содержащейся в журналах, цель которых записывать информацию о различных транзакциях и сообщениях. Благодаря ему существует возможность эффективно хранить большое количество данных.

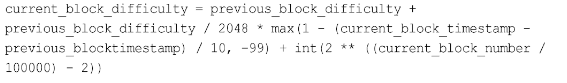
difficulty – уровень сложности текущего блока, понятие, используемое для обеспечения согласованности времени, которое необходимо для создания блоков. Трудность создания блока определяется по одному критерию — времени, затраченному на формирование. Самый первый (genesis block) имел сложность 131 072, все остальные рассчитывались по конкретной формуле. Заметив ускорение создания, сеть автоматически увеличивает сложность.

extraData – дополнительные данные блока

mixHash – хэш, который в сочетании с элементом nonce утверждает, что для текущего блока выполняется достаточно вычислений

nonce – хэш, который в сочетании с элементом mixHash утверждает, что для текущего блока выполняется достаточно вычислений. Является 64-битным беззнаковым целым числом – решением задачи.

Задача майнера при формировании блока – найти такое число nonce, при котором хэш готового блока меньше целевого числа или равен ему. Целевое число – это 256-битное число, которое вычисляется исходя из различных факторов. Уровень сложности в заголовке является другим представлением целевого числа: чем меньше целевое число, тем больше времени потребуется для нахождения числа nonce. Уровень сложности вычисляется по формуле:



Данная формула содержит 10-секундную задержку, которая гарантирует, что разница времени между родительским и дочерним блоком не превысит 10-20 секунд, что обеспечивает пользователям понимание времени формирования нового блока. Формирование блоков может происходить по двум возможным сценариям в зависимости от того создан блок на данный момент времени или происходит процесс его создания. Если блок уже создан, то он должен пройти процедуру валидации, если он еще находится в состоянии создания, то его окончательное оформление происходит в процессе майнинга. Для окончательного формирования блока необходимо, чтобы следующие условия были выполнены:

1. Валидация или определение оммеров, представляющих собой блоки, созданные из текущих блоков с целью выплаты вознаграждения майнерам даже в случае потери блока. Действительные оммеры имеют валидные заголовки и соседствуют с текущими блоками на расстоянии не более шести единиц.
2. Валидация транзакций – сопоставление значения GasUsed с суммарным значением газа по всем транзакциям в блоке
3. Верификация состояния и значения nonce – проверка выполнения всех транзакций в блоке, а также измерение результирующих состояний
4. Назначение оплаты в случае майнинга.

Отметить, что число nonce ((от англ. nonce — «number that can only be used once» — число, которое может быть использовано один раз) в Ethereum применяется в двух случаях:

- в аккаунтах – в качестве счетчика транзакций в каждом аккаунте. Это предотвращает атаки повторной отправки транзакции с целью двойной траты средств. Два различных аккаунта могут иметь одинаковое число nonce.

- в блоках – случайное значение в блоке, которое подтверждает количество выполненной работы

Иногда при формировании блоков два майнера одновременно формируют последующий блок. Оба таких блока могут быть корректными, но в блокчейне не может быть два блока с одинаковым номером и двойное вознаграждение за один блок. В этом случае сетью принимается блок, сложность создания которого была выше. Непринятый блок будет являться устаревшим блоком uncle block, который будет образовывать ответвление. При образовании форков (ответвлений) в блокчейне для определения корректного пути и предотвращения образования множества цепей в Ethereum используется протокол GHOST, суть которого сводится к тому, чтобы выбрать тот путь, на котором было произведено наибольшее количество вычислений, и принять его за единственно верный. Чтобы не засорять блокчейн и побудить майнеров включать uncle block в новые блоки, майнерам выплачивают вознаграждение за подобную утилизацию. Новый блок, включивший в себя uncle block называется nephew block. Протокол GHOST ограничивает количество устаревших блоков в новом блоке.

Платформа Ethereum сама по себе не имеет ценности. Подобно языкам программирования, предприниматели и разработчики сами решают, для чего ее использовать. Очевидно, что некоторые типы приложений получат больше преимуществ от возможностей, предоставляемых платформой. В частности, Ethereum подходит для приложений, которые автоматизируют прямое взаимодействие между пирами или облегчают согласованные групповые действия в сети. Например, приложения для координации одноранговых торговых площадок или автоматизации сложных финансовых контрактов. Теоретически, финансовые взаимодействия или обмены любой сложности могут выполняться автоматически и надежно с использованием кода, запущенного в Ethereum. Помимо финансовых приложений, платформа Ethereum может оказать сильное влияние на любую среду, в которой важны доверие, безопасность и постоянство, например, реестры активов, голосование, управление и Интернет вещей.

**Hyperledger Fabric**. В соответствии с потребностями современных рынков IBM и ряд других компаний инициировали совместный проект по созданию открытой отраслевой реализации технологии блокчейна для коммерческого использования, который получил название Hyperledger Fabric, (один из проектов Hyperledger, работающих на ресурсах Linux Foundation). Hyperledger Fabric представляет собой фрэймворк для распределенных регистров в контролируемых сетях, в которых все участники имеют известные идентификационные данные. Это позволяет использовать фрэймворк в тех решениях, которые регулируются законами о защите данных и где необходима информация о лицах, имеющих доступ к таким данным. Решение имеет модульную архитектуру, позволяющую использовать по принципу plug-and-play различные компоненты, и применяет технологию контейнеров для выполнения «умных» контрактов, реализующих логику приложений системы. Благодаря такой архитектуре Hyperledger Fabric обеспечивает максимальный уровень конфиденциальности, устойчивости, гибкости и масштабируемости для решений блокчейна. Одна из наиболее востребованных областей модульности - «принеси свою личность – bring your own identity», т.к. некоторые компании уже имеют систему управления идентификацией и хотят использовать ее повторно, а не перестраивать. Другие компоненты архитектуры, которые можно легко подключить, включают консенсус или шифрование (в некоторых странах существуют свои собственные стандарты шифрования). Умные контракты в Fabric называются «chaincode».

Обработка транзакций в Fabric разделена на три этапа: распределенная логическая обработка и согласование, упорядочение транзакций, проверка транзакций и принятие. Такое разделение дает несколько преимуществ: требуется меньшее количество уровней доверия и проверки для разных типов узлов, что позволяет оптимизировать масштабируемость и производительность сети, обеспечивает ее динамический рост. Каналы, поддерживаемые в Hyperledger Fabric, позволяют передавать данные только тем сторонам, которые нуждаются в этих данных.

Давайте посмотрим на поток транзакций в v1.0 Hyperledger Fabric, показанный на рисунке 1.

Начиная слева:

1) Предложение по сделке представляется приложением подтверждающей стороне.

2) Политики *индоссамента* определяют, сколько и / или какая комбинация индоссантов требуется для подписания предложения. Индоссант выполняет цепной код для имитации предложения в одноранговой сети, создавая набор для чтения / записи. (Индоссаме́нт — передаточная надпись на ценной бумаге, векселе, чеке, коносаменте и т. п., удостоверяющая переход всех или части прав по этому документу к другому лицу)

3) Подтверждающие партнеры отправляют обратно подписанные ответы на предложения (одобрения) в приложение.

4) Приложение отправляет транзакции и подписи службе заказа

5) Служба заказа создает пакет или блок транзакций и доставляет их совершающим одноранговым узлам.

6) Когда принимающий узел получает пакет транзакций, для каждой транзакции он проверяет, была ли соблюдена политика одобрения, и проверяет наборы для чтения / записи для обнаружения конфликтующих транзакций. Если обе проверки пройдены, блок фиксируется в блокчейне, и обновления состояния для каждой транзакции отражаются в базе данных состояний.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Поскольку транзакции видны только индоссаторам и коммиттерам, расположенных в разных частях системы, для осуществления транзакций требуется меньше уровней доверия, что обеспечивает большую безопасность.

Каждая транзакция в Fabric приводит к набору пар «ключ-значение актива», которые фиксируются в регистре при создании, обновлении или удалении. Опции базы данных состояния включают LevelDB и CouchDB. LevelDB - это база данных состояния ключ-значение по умолчанию, встроенная в одноранговый процесс. CouchDB - это необязательная альтернативная база данных внешнего состояния. Как и хранилище значений ключей LevelDB, CouchDB может хранить любые двоичные данные, которые смоделированы в коде смарт-контракта (функциональность вложений CouchDB используется внутри для двоичных данных не в формате JSON). Но как хранилище документов JSON, CouchDB дополнительно разрешает расширенный запрос к данным смарт-контракта, когда значения смарт-контракта (например, активы) моделируются как данные JSON. Формат JSON помогает минимизировать работу, необходимую для создания простых отчетов и выполнения функций аудита. Например, в сценариях цепочки поставок вы можете использовать JSON, чтобы помочь выделить конкретные данные для товаров и транспортных объектов. Вы можете легко создать отчет об активе для различных местоположений и объектов транспортировки, которые использовались при доставке в конечный пункт назначения актива. При моделировании активов как JSON и использовании CouchDB, можно выполнять сложные расширенные запросы к значениям данных смарт контракта, используя язык запросов CouchDB JSON.

Как LevelDB, так и CouchDB поддерживают основные операции с chaincode, такие как получение и установка ключа (актива) и выполнение запросов на основе ключей. Ключи могут быть запрошены по диапазону, а составные ключи могут быть смоделированы так, чтобы включить запросы эквивалентности для нескольких параметров. Например, составной ключ владельца, *asset\_id* может использоваться для запроса всех активов, принадлежащих определенной сущности. Эти основанные на ключах запросы могут использоваться для запросов только на чтение регистра, а также в транзакциях, которые обновляют регистр.

По умолчанию LevelDB имеет базу данных значений ключей и поддерживает запросы с ключами, запросы составных ключей и запросы диапазона ключей. Если вам также нужны сложные расширенные запросы, CouchDB поддерживает базовые возможности LevelDB и добавляет полные запросы с большим количеством данных. При необязательной поддержке базы данных документов, такой как CouchDB, содержимое является JSON и полностью запрашиваемым, где модель данных совместима с существующей моделью программирования ключ / значение. В результате изменения приложения не требуются при моделировании данных цепного кода как JSON при использовании CouchDB.

По сути, Fabric предназначен для проектов, в которых требуется технология распределенных регистров (DLT). Фрэймворк поддерживает смарт-контракты в языках программирования Go (Golang), Java и JavaScript (через Hyperledger Composer, или с версии 1.1).

Почему же мы не рассматриваем самый первый Dapp **Биткойн** в качестве платформы для разработки приложений? Несмотря на то, что некоторые принципы смарт-контрактов были заложены в протоколе Биткойна, сеть не может реализовать их в клиентском ПО из соображений безопасности. Справедливости ради надо заметить, что некоторые форки (ответвления) Биткойна позволяют это делать. Проблема в том, что сценарный язык Биткойна не является полным по Тьюрингу. язык Bitcoin подходит для программирования операций, содержащих простые условия, типа If…Then…Else…Его полнота считается недостаточной, а сеть не имеет маркеров состояния, чтобы на такой платформе можно было создавать программируемые алгоритмы. Да и пропускная способность Биткоина сильно ограничена, что тоже мешает созданию многофункциональных контрактов.

Глава 5. Существующие проблемы в развитии платформ

Современные платформы смарт-контрактов сталкиваются с рядом системных проблем, которые не всегда очевидны широкой публике на данном этапе развития индустрии, но могут помешать глобальному внедрению блокчейна в ближайшем будущем.

***Масштабируемость***

В настоящее время все согласованные протоколы блокчейна (например, Биткойн, Ethereum, Ripple, Tendermint) имеют сложное ограничение, связанное с децентрализацией: каждый узел в сети обрабатывает каждую транзакцию и поддерживает копию всего состояния.

Хотя децентрализованный механизм консенсуса обеспечивает некоторые важные преимущества, такие как отказоустойчивость, гарантия безопасности, политический нейтралитет и подлинность, он же препятствует масштабированию сети. Децентрализация по определению ограничивает число транзакций, которые блокчейн способен обработать, до масштабов работы одного полнофункционального узла в сети.

В традиционной системе баз данных решение масштабируемости заключается в добавлении большего количества серверов (то есть вычислительной мощности) для обработки добавленных транзакций. В мире децентрализованной цепочки блоков, где каждый узел должен обрабатывать и проверять каждую транзакцию, нам потребуется добавить больше вычислений на каждый узел, чтобы сеть работала быстрее. Отсутствие контроля над каждым общедоступным узлом в сети приводит нас в замешательство.

В результате все общедоступные протоколы согласования блокчейнов, которые работают таким децентрализованным образом, обеспечивают компромисс между низкой пропускной способностью транзакций и высокой степенью централизации. Другими словами, с ростом размера блокчейна возрастают требования к хранилищу, пропускной способности и вычислительной мощности, необходимых для полноценного участия в сети. В какой-то момент он станет настолько громоздким, что только несколько узлов смогут обрабатывать один блок. Это приведет к риску централизации.

Для масштабирования протокол блокчейна должен определить механизм ограничения количества участвующих узлов, необходимых для проверки каждой транзакции, без потери доверия сети к действительности каждой транзакции. Это может показаться простым на словах, но технологически очень сложно осуществимо.

Поскольку каждый узел не участвует в проверке каждой транзакции, узлам необходимы статистические и экономические средства для гарантии безопасности других блоков (в проверке которых они не участвуют). Другими словами, даже если блок выглядит действительным с точки зрения узла, который не выполняет прямую проверку этого блока, сокрытие данных для этого блока приведет к ситуации, когда никакой другой валидатор в сети не сможет проверить транзакции или создавать новые блоки. Блокчейн застрянет в текущем состоянии.

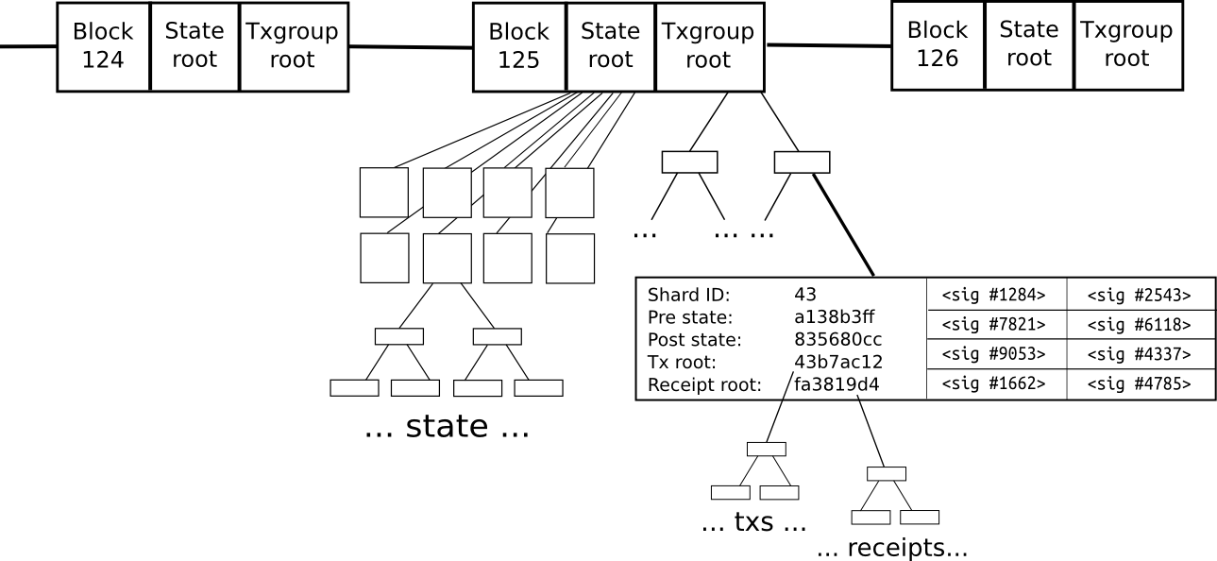
Как проблема масштабируемости выглядит снаружи? Многие энтузиасты надеются, что криптовалюты в конечном итоге смогут конкурировать с PayPal и Visa. Однако в нынешнем виде блокчейны далеки от этого. Существующие криптовалюты изначально не были разработаны с учетом их массового использования и адаптации к реальному миру. С ростом количества транзакций в криптовалюте выявляется все больше проблем.

В настоящее время Visa является самой быстрой платежной сетью. Она способна обрабатывать около 24 000 платежей в секунду. Для сравнения, PayPal может обрабатывать приблизительно 193 транзакции в секунду. Скорость транзакции для криптовалют определяется временем, затрачиваемым на добавление транзакции в блок, плюс временем, необходимым для достижения консенсуса. Большинство криптовалют работают намного медленнее Visa. Например, блокчейн одной из самых быстрых криптовалют Ripple может обрабатывать только до 1500 транзакций в секунду. Между тем, Bitcoin и Ethereum, со скоростью обработки транзакций 7 и 20 транзакций в секунду соответственно, отстают еще больше. Эта проблема серьезно ограничивает вероятность того, что в будущем существующие криптовалюты станут популярными платежными решениями.

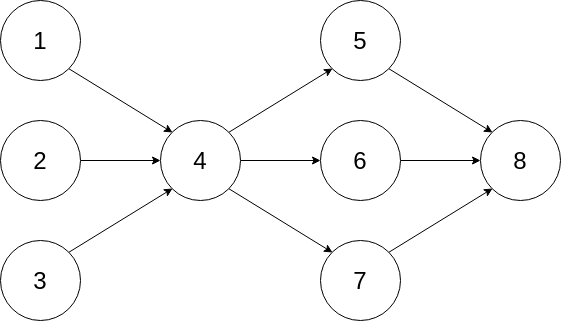
Каковы же решения?

Во-первых, это **создание платежных каналов и осуществление вычислительных операций за пределами блокчейнов**. Блокчейн обрабатывает финальную транзакцию в серии операций на пути к окончательному платежу, что позволяет избавить основную сеть от лишней нагрузки. Тем самым решается проблема пропускной способности, поскольку в таком случае блокчейн может масштабироваться под больший объём транзакций. Более того, поскольку транзакция происходит, как только её обработал платёжный канал, а не тогда, когда блок получает подтверждение, каналы микроплатежей решают и проблему скорости, позволяя избавиться от типичных задержек. Перевод вычислительных операций и процесса верификации из блокчейна в отдельный протокол позволит эффективнее обрабатывать транзакции.

Во-вторых, это применение **шардинга**, в основе которого лежит схема разделения общего состояния блокчейна на сегменты и хранение и обработка разными узлами в сети каждой части состояния. Каждый узел обрабатывает лишь малую часть состояния, делая это параллельно с остальными.

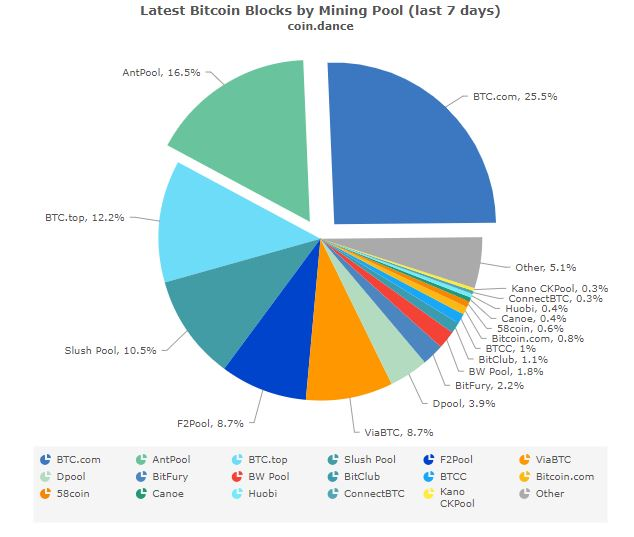


В третьих, это **разработка протоколов на основе направленных ациклических графов**, или DAG (Directed Acyclic Graph), представляющих собой графическую структуру данных, имеющую рёбра и вершины. Вершина — это точка на графе, а ребро — путь от одной вершины к другой. DAG гарантирует невозможность начать движение в любой вершине и следовать череде рёбер, которая в итоге возвращается к этой вершине (то есть невозможны замкнутые циклы). Это позволяет иметь последовательность узлов (или вершин) в топологическом порядке.



***Централизация***

Главной идеей блокчейна является его децентрализация. Однако с развитием блокчейнов именно децентрализация оказалась под угрозой. Старейший протокол консенсуса PoW, применяемый в Биткойне и Ethereum, обнаружил существенный недостаток. Он заключается в том, что со временем появилась опасность пренебрежения принципами децентрализации, и этому вовремя не уделили внимания. Это привело к тому, что в настоящий момент сеть Биткойн работает с все более централизованной системой майнеров. В текущей реализации PoW Биткойна различают пользователей сети и провайдеров сети. Не смотря на то, что майнерами по алгоритму POW могут быть любые пользователями сети, на практике большинство пользователей не имеют средств для участия в майнинге блоков. Вероятность майнера для следующего блока в цепочке зависит от их относительной мощности хеширования. Чтобы гарантировать стабильное вознаграждение за участие в сети, большинство майнеров объединяются в так называемый майнинг-пул, передавая управление майнинговым оборудованием управляющему пулом. Например, в Биткойне пять крупнейших майнинговых пулов контролируют 75% мощности хеширования. Также в 2013 году появились устройства, называемые интегральными схемами прикладной ориентации (микросхемы ASIC). Они были разработаны исключительно для майнинга биткоина, увеличив его эффективность в 10-50 раз. С тех пор майнинг с обычным процессором стал совершенно невыгодным: майнить можно только посредством устройства ASIC, сделанного самостоятельно или купленного у производителя. Это противоречит децентрализованной сущности блокчейна, где у каждого есть возможность вносить вклад в безопасность сети.



Считалось, что уйти от централизации поможет внедрение нового протокола консенсуса PoS, в котором выбор майнера для майнинга следующего блока осуществляется среди владельцев монет. В алгоритме PoS токены выдаются подтверждающим узлам в самом начале существования сети. То есть, в отличие от PoW, с добавлением в реестр новых блоков токены не создаются (хотя есть блокчейны, использующие гибридный алгоритм PoW/PoS, допускающий майнинг PoW с последующим переходом к PoS). Каждые несколько секунд или минут выбирается узел, записывающий новый блок. Но чем больше у узла монет, тем больше его вес в определении того, что в реестре считается истиной. Следовательно, на выбор сильно влияют те, у кого больше всего монет. Ещё один влияющий фактор – время владения монетами и активность участников. Поэтому в PoS пользователи, имеющие больший вес (большее количество монет, ранний вход в блокчейн, большую активность), несут угрозу децентрализации и являются нежелательными, даже несмотря на то, что именно они обеспечивают устойчивость блокчейна к атакам.

***Конфиденциальность***

Проблема конфиденциальности в блокчейнах основана на условной анонимности пользователей: транзакции записываются и хранятся в публичном реестре и связанны с адресом аккаунта, состоящим исключительно из чисел и букв. Поскольку адрес не связан с личностью в реальном мире считается, что инициатора транзакции невозможно отследить. Все это работает до тех пор, пока не установлена связь реального человека и его аккаунта в блокчейне. Веб-трекеры и cookie-файлы на коммерческих сайтах позволяют информации о транзакции «просачиваться» в сеть, где все, включая правительства, спецслужбы и преступников, легко могут ей воспользоваться. Также анонимность и конфиденциальность оказывается под угрозой из-за использования смарт-контрактов, в которые часто содержат данные отправителей и получателей и данные самих транзакций, выполняемый код и состояние. В настоящее время крайне нежелательна загрузка в публичные блокчейны следующей информации:

- Электронная медицинская документация, представляющая собой в высшей степени конфиденциальную информацию.

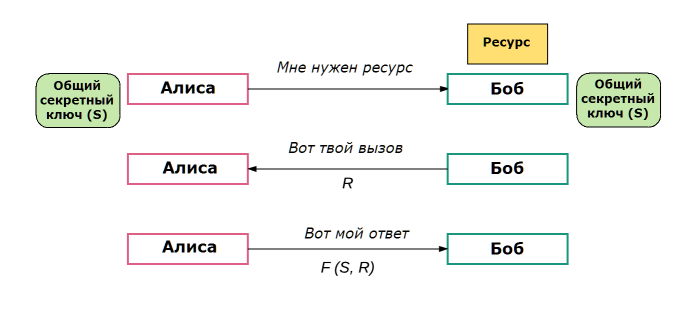
- Данные удостоверений личности: номера социального страхования, паспортные данные и т.п.

- Информацию по управлению учётными данными.

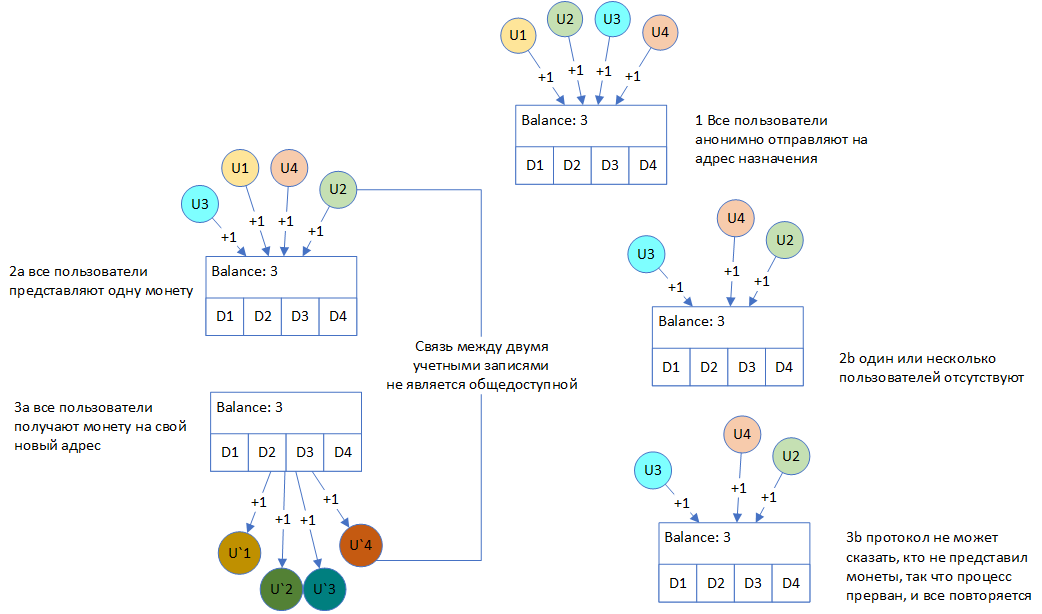
- Финансовые документы, представляющие собой коммерческую тайну.

Существует несколько решений проблемы конфиденциальности, предложенных разработчиками.

*Адреса на протоколе Диффи — Хеллмана — Меркля на эллиптических кривых (ECDHM)*. Идея работы протокола заключается в том, что с помощью совместно используемого секретного ключа происходит конфиденциальный обмен сообщениями в публичной сети. Отправитель и получатель могут публично поделиться адресами на протоколе Диффи — Хеллмана — Меркля на эллиптических кривых, а затем использовать совместный секретный ключ для того, чтобы создать анонимный биткоин-адрес. Такой адрес может открыть только пользователь, обладающий совместным секретным ключом. Адрес на протоколе является публичным и пригодным для многоразового использования. Таким образом, пользователь может не опасаться того, что транзакции будут отслежены.



*Миксеры*. Идея миксера состоит в том, что группа людей может объединить свои платежи в один пул, используя для отслеживания данных приватный реестр. После расходования средств пула источники оригинальных платежей скрываются. Публичными остаются объёмы платежей и получатели, но при этом (в теории) невозможно установить, кто авторизовал платёж.



К сожалению, в настоящий момент миксеры являются ненадёжным решением из-за низкой защиты от атак Сибиллы и DDoS-атак, необходимостью наличия доверенной третьей стороны, «смешивающей» транзакции.

*Приватные криптовалюты*. Другой способ решения проблемы конфиденциальности состоит в создании криптовалюты, которая приватна по умолчанию.

*Доказательства с нулевым разглашением*. Доказательство с нулевым разглашением конфиденциальных сведений подразумевает ситуацию, в которой доказывающий убеждает верифицирующего, что у него есть некое секретное знание, не открывая самое знание. Иначе говоря, программа может хранить секретные параметры, и доказывающий ничего не открывает проверяющему. Доказательства с нулевым разглашением обеспечивают фундаментальную основу, которую можно использовать для создания сохраняющих конфиденциальность механизмов.

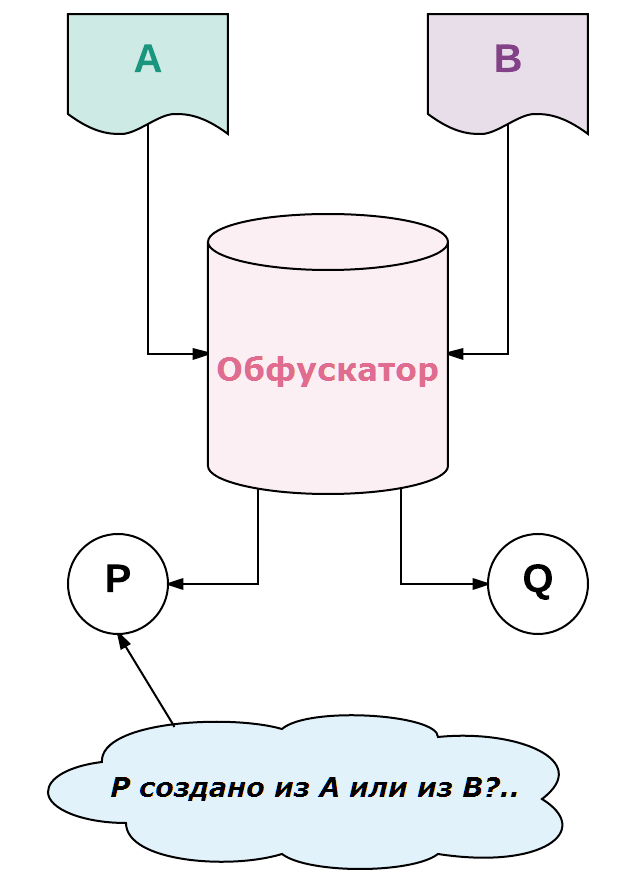
Примером технологии является zk-SNARК: zk = zero-knowledge (дословно — нулевое знание, здесь — нулевое разглашение). Это значит, что для доказательства существования информации необязательно делиться этой информацией. SNARK: Succinct Non-interactive Adaptive ARgument of Knowledge (сжатое не интерактивное адаптивное доказательство знания). «Сжатое» означает, что доказательство может быть предоставлено в короткий срок. «Неинтерактивное» означает, что проверяющей стороне не нужно напрямую взаимодействовать с доказывающей стороной. Вместо этого, доказывающая сторона может заранее опубликовать доказательство, которое потребуется проверяющей стороне. «Адаптивное доказательство знания» означает доказательство знания о некоем вычислении.

Изображение выглядит как снимок экрана

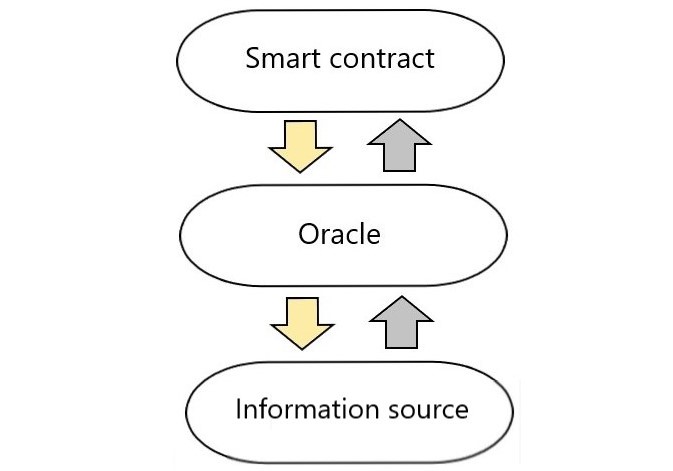
Автоматически созданное описание

Однако, данная технология не лишена недостатков. SNARK требует предварительной фазы настройки, во время которой фиксируется подлинность вычисления (circuit). В этой фазе участники закрытой группы людей, доверяющих друг другу, должны выполнить определенные действия. То есть это не только требует от участников доверия к людям, проводящим необходимую настройку, но также делает SNARK не лучшим выбором для выполнения произвольных вычислений по причине необходимости предварительной настройки. SNARK требует больших вычислительных мощностей и ответственности пользователей за сохранение секрета и предоставление этих данных по необходимости.

*Обфускация* (запутывание кода). Одним из механизмов обеспечения конфиденциальности является обфускация, или запутывание кода. Цель в том, чтобы запутать программу P так, чтобы запутывающий мог создать вторую программу O(P) = Q таким образом, чтобы P и Q возвращали аналогичные выходные данные, получая те же самые вводные данные, но при этом Q не раскрывала бы информацию о внутренних составляющих P. Такая схема позволяет скрывать приватные данные внутри Q (пароли, номера документов и т.п.) и всё же использовать их внутри программ.



*Оракулы*. В измерении блокчейна оракул — это сторона, которая передаёт информацию между смарт-контрактами и внешними источниками данных. По сути, оракул действует как переносчик информации между смарт-контрактами в блокчейне и внешними источниками данных вне блокчейна. Так что один из способов обеспечения приватности — использование оракулов для доставки информации из внешних источников данных.



*Доверенная среда выполнения программы (TEE).* Это защищённая область главного процессора. Она гарантирует, что код и данные, загруженные в неё, защищены в смысле конфиденциальности и целостности. Доверенная среда функционирует параллельно с операционной системой, призванной удовлетворять потребности пользователей, но предполагается, что она превосходит ОС степенью конфиденциальности и защиты.

***Отсутствие формальной верификации контрактов***

В настоящее время формальная проверка смарт-контрактов выполняется для каждого контракта и каждого его использования. У смарт-контракта нет возможности выявить, был ли проверен другой контракт, с которым он связан, или нет. Если кто-то официально проверил некоторые контракты, у вас нет возможности проверить это, не выполнив программу формальной проверки самостоятельно. В противном случае все, что вы можете сделать, - это доверять человеку, который осуществил формальную проверку.

***Хранение больших данных***

Поскольку сеть блокчейна неизменяема и данные туда можно только добавлять, но не удалять, возникла серьезная проблема, заключающаяся в том, что каждый узел будет вынужден хранить бесконечно увеличивающийся поток данных: идентификационную информацию пользователей, финансовую информацию и т.д. Проблему может решить стратегия разделения данных на сегменты и их хранение распределенным образом на участвующих в обработке данных узлах (то есть распределённого хранения) . Базовый принцип состоит в том, что каждый узел не хранит всё: наборы узлов расщепляют или «распределяют» данные в своей среде.

***Отсутствие управления и стандартов***

Полностью открытые и децентрализованные блокчейны показали свою несостоятельность и неэффективность, т.к. не существует безопасного пути обновления протокола, и никто не ответственен за установление и поддержание стандартов. Пока совершенно неясно каким образом можно координировать деятельность разработчиков и других представителей экосистемы, чтобы приходить к согласию по вопросам новых стандартов, функций и обновлений. Управление развитием блокчейна - одна из серьезных проблем в виду сложности достижения баланса между централизованным и распределённым контролем.

***Недостаточный инструментарий***

В данный момент в экосистеме инструментария разработчика смарт-контрактов слабыми местами являются:

- интегрированная среда разработки (IDE)

- система сборки кода и компилирующая программа,

- средства развертывания

-тестовая среда

- инструменты отладки и логирования

- аудит безопасности

- инструменты аналитики

Совершенствование инструментария разработки скажется позитивным образом на развитии технологии.

***Угроза со стороны квантовых вычислений***

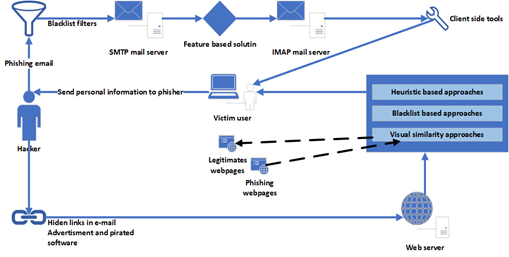
Самые популярные алгоритмы шифрования с открытым ключом могут быть эффективно разрушены с помощью достаточно мощного квантового компьютера. Решением является разработка постквантовой криптографии: криптографии на основе решёток, многовариантной криптографии, криптографии на основе хеша, криптографии на основе кода, криптографии на основе изогении суперсингулярных эллиптических кривых и системы на основе симметричных ключей — AES и SNOW 3G.

Глава 6. Существующие проблемы с безопасностью блокчейнов.

В последние несколько лет, благодаря популяризации технологии блокчейн, которая представляет собой выстроенную по определенным правилам непрерывную последовательную цепочку блоков, содержащих информацию, на ее основе было создано множество сервисов и приложений, использующих различные криптовалюты. Многие криптовалюты неразрывно связаны с этой технологией по ряду причин: решения на блокчейне не требуют доверия между участниками, они открыты и валидируемы. Бурный рост и инвестиционная привлекательность технологии постоянно привлекают внимание злоумышленников. Блокчейны часто являются объектами для атак и попыток взлома. Условно все атаки на блокчейны можно разделить на *атаки внешнего мира* и *внутренние атаки*. Атаки внешнего мира не затрагивают саму систему блокчейна , а направлены на окружающую инфраструктуру и пользователей, они не требуют глубоких знаний и большой подготовки для их осуществления, также в большинстве случаев они достаточно дешевы. Внутренние атаки – атаки непосредственно в самой сети. Такие атаки технологически сложны и часто требуют немалых затрат на их проведение. **Разработчикам необходимо уделять проблеме безопасности особое внимание, поэтому, пожалуйста, внимательно знакомьтесь с этой главой.**

Большинство успешно реализованных атак внешнего мира (по разным оценкам на них приходится 80-96%) было осуществлено с использованием фишинга. Фишинговые атаки направлены на получение учетных данных пользователей, важной информации, введение пользователей в заблуждение с целью завладеть средствами. Целью злоумышленников может быть что угодно: от попытки заставить людей отправить деньги мошенникам, передать конфиденциальную информацию, до принуждения простого скачивания вредоносных программы. Авторы этих атак используют ложь, обман, подделку, манипуляции, чтобы добиться успеха. Опасность фишинговых атак заключается не в наличии уязвимостей в системе, а в доверчивости человека и его невнимательности. Часто Фишинг Использует Социальную инженерию: вид нападения, которое опирается на человеческие ошибки, а не аппаратное или программное несовершенство. Многие блокчейн проекты, которые имели сильные политики безопасности, оказались беспомощными перед фишингом.

Условно, все фишинговые атаки можно разделить на два типа: схемы социальной инженерии и технические атаки. Схемы социальной инженерии основаны на обмане и последующих независимых неправильных действиях жертвы, в то время как технические атаки используют уязвимости и недостатки программного обеспечения и инфраструктуры.



**Схемы социальной инженерии.** Особенностью таких схем является прямое участие пользователей сети в них. Нападение осуществляется путем передачи жертве ложной информация и осуществлении атаки после того, как пользователь выполняет определенные действия (открытие письма, перемещение по ссылке или скачивание вредоносного вложения). Согласно исследованию компании Verizon, 4% пользователей любой компании нажимают на фишинговые ссылки. Для того, чтобы злоумышленник атаковал системы, достаточно лишь одного неосторожного клика.

|  |  |  |
| --- | --- | --- |
| **Схема** | **Проект** | **Описание** |
| Клоны | Blockchain.info, MyEtherWallet, BinanceIota | Мошенники используют атаку гомографа для создания клонов сайтов, проводят рекламную кампанию и распространяют ссылки. $50 000 000 были украдены на поддельном сайте Blockchain.info. $4 000 000 были украдены благодаря фишинговому веб-сайту генерации секретных кодовых фраз для криптокошельков MIOTA. |
| Социальные сети | Telegramm, Blockchain.info | Мошенники создают фальшивые аккаунты в социальных сетях, выдавая себя за известные проекты или ключевые фигуры. |
| Раздувание пузыря | ChainCoin, HighCoin | Искусственно созданный спрос, агрессивная реклама в СМИ и социальных сетях приводят к беспрецедентному росту и последующему резкому падению курса крипто-валюты |
| Поддельные ICO | PlexCoin, E-coin | Проекты не владеют ни технологиями, которые были заявлены в WhitePaper. ни командами, которые могли бы создать продукт. $15 000 000 собранных PlexCoin на ICO были заморожены SEC. |
| Пирамиды, Ponzi | Bitcoin, Bitconnect | Организатор предполагаемой финансовой пирамиды рекламировал "инвестиционные возможности" в онлайн-форуме сообщества Bitcoin . Инвесторам обещали небывалый рост до 7% процентов в неделю Вместо этого, свежеинвестированные биткойны использовались для оплаты существующих инвесторов и обменивались на доллары США для оплаты личных расходов организатора. |
| Направленный фишинг | Enigma, | Хакеры получили доступ к электронной почте Генерального директора Enigma Гая Зискинд. В результате всем инвесторам были отправлены сообщения с предложением участвовать в раннем старте продаж токенов и были отправлены реквизиты счета для перевода средств. |

*Клонирование*. Для тoго, чтобы осуществить этот тип атаки, злоумышленники создают копию сайта с аналогичным названием, или поддельные страницы на официальном сайте, и отправляют ссылку на поддельные ресурсы потенциальным жертвам.

Часто, для создания фальшивых страниц и сайтов, мошенники используют кодирование Punycode. Эта кодировка позволяет регистрировать домены с нелатинскими символами. Она работает путем преобразования одной буквы домена в альтернативный формат, используя символы ASCII. Чаще всего при написании названия сайта мошенники используют кириллицу. Например, символ *а* может быть представлен как латинской буква "a" (U + 0061) так и кириллической "a" (U + 0430). Это означает, что человеческий глаз не может распознать разницу. В результате становится невозможным идентифицировать сайт как мошеннический, без тщательной проверки URL сайта. Данные атаки приято называть *атаками гомографа*.

Для достижения наибольшего сходства и усыпления бдительности жертв, поддельные сайты и страницы оснащаются SSL-сертификатами. Обычно для этих целей используются бесплатные 90-Day сертификаты Let's Encrypt и центры аттестации Comodo.

*Фишинг в социальных сетях*. Фишинговые атаки в социальных сетях стали очень распространенными. Участились случаи взлома счетов известных личностей и публикации сообщений от их имени, содержащих фишинговые ссылки, создавая страницы клон-известных личностей, сообществ и т.д. Мошенники используют тот факт, что Facebook позволяет создавать страницы с любым именем.

*Направленный фишинг*. Объектом фишинга являются крупные инвесторы, владельцы кошельков, первые лица компаний, владельцы криптовалют. Злоумышленники четко представляют, что именно, сколько, от кого и как они хотят получить. Распространение получил новый вид такого взлома. Злоумышленники выявляют активность жертвы в других сферах деятельности, не связанных с блокчейнами и криптовалютами, и крадут необходимые данные, имитируя интерес в этих сферах, вступая в контакт с жертвами.

*Раздувание*. Мошенники искусственно поднимают цены для простых в управлении крипто-валют с низкой ликвидностью и небольшой рыночной капитализацией. Беспрецедентное повышение цен широко освещается в многочисленные медиа-каналах (YouTube, Tвиттер Tелеграм), обещая высокую доходность для внешних инвесторов. После значительных инвестиций со стороны пользователей, мошенники прекращают поддерживать курс крипто-валюты, и его цена возвращается в исходное положение.

*Пирамиды, Ponzi*. Не имея никакой блочейн инфраструктуры и коммуникаций с биржами, эти проекты обещают высокую рентабельность инвестиций в крипто-валюты и блокчейн проекты, а также высокий процент от инвестиций новых привлеченных пользователей. Мошенники неожиданно закрывают проекты для всех после сбора необходимой суммы. Чаще всего эти схемы используются в псевдо облачном майнинге и поддельных крипто-обменных биржах.

*Фальшивый ICO*. ICO (первоначальное предложение монеты) является мечтой любого хакера. Молниеносная, часто довольно простая атака на крипто-валютные сервисы и блокчейн сообщества приносит миллионы долларов прибыли с минимальным риском для преступников. Ранние инвесторы ICO наиболее уязвимы для злоумышленников.

В связи с тем, что в законодательстве большинства стран существуют серьезные пробелы в обращении криптовалют и нет судебной практики, мошенники используют ажиотаж вокруг технологии для вывода на ICO проектов, не имеющих реального продукта и команды. Для этого мошенники создают страницу проекта, которая представляет собой только видео или презентацию проекта, техническую документацию (в лучшем случае) и криптовалютный кошелек для сбора инвестиций. Кампания широко освещается в средствах массовой информации, на конференциях. Мошенники обращаются непосредственно к инвесторам, организуя личные встречи. Также мошенники используют схему взлома базы адресов ранних инвесторов реальных проектов с последующим приглашением на ранний ICO на выгодных условиях, кошелек для перевода средств принадлежит мошенникам.

**Технические схемы**

Этот тип атаки является более трудоемким, но менее заметным, процент достижения цели намного выше, чем в социально-инженерных схемах.

|  |  |  |
| --- | --- | --- |
| **Схема** | **Проекта** | **Описание** |
| На основе DNS | Blockchain.info, MyEtherWallet | Замена адресов DNS, пользователи перенаправлялись на вредоносные сайты |
| Перехват сессий | Аппаратный кошелек Ledger | Злоумышленники заменяют код, ответственный за создание адреса получателя с собственным адресом, в результате чего все будущие депозиты будут высланы злоумышленнику. |
| Вредоносные программы | Lazar Group | Хакеры из группы Lazar вели переписку в чате, выдавая себя за "ключевых людей" в крипто-валютной индустрии. Во время разговора они публиковали небольшой фрагмент кода, который был на самом деле вредоносным. Если пользователь скачивал этот код, хакер имел возможность войти в систему и украсть крипто-валюту. |
| Кейлоггеры | Blockchain.info | Тысячи биткоинов были украдены с помощью кейлоггеров |

*Фишинг на основе DNS*. В этой атаке злоумышленник изначально создает удаленные точки доступа и заманивает клиента для подключения к такой точке, где атакующий запускает поддельные DNS-сервера. Этот сервер перенаправляет отдельные запосы на фишинг-сервер злоумышленника.

*Перехват сеансов (перехват куки-файлов, сессий)*. Атака основана на перехвате ключа сеанса, для получения несанкционированного доступа к информации или службам в компьютерной системе.

*Вредоносные программы*. При использовании фишинга на основе вредоносных программ вредоносные программы используются для кражи учетных данных на компьютере жертвы и отправки их злоумышленнику. Например, угроза может быть доставлена через спам-сообщения с прикрепленным файлом doc, который содержит сценарий PowerShell, который загружает вредоносное ПО. В свою очередь данное ПО находит сохраненные кошельки и учетные данные и дает к ним доступ злоумышленникам. Количество вредоносных программ, используемых злоумышленниками, постоянно увеличивается, а сами инструменты постоянно модифицируются.

*Кейлоггеры*. Они используются для кражи данных, когда пользователь вводит информацию с устройства. С появлением виртуальных клавиатур и сенсорных экранов, злоумышленники используют снимки экрана.

Из вышесказанного следует, что для отражения фишинговых атак необходимо постараться исключить необоснованные действия пользователей, отслеживать клоны в Интернете и сообщать о них регистраторам доменов, информировать пользователей о безопасном входе и хранении личной информации, быть на связи с пользователями для незамедлительной реакции, если атака все-таки была реализована. Никогда не забывайте, что неграмотные пользователи могут похоронить ваш проект.

**Внутренние атаки (атаки в сети блокчейн) и уязвимости.** В связи с тем, что в основе любого децентрализованного приложения лежит выполнение смарт-контрактов, основные атаки направлены именно на них. Существующие уязвимости смарт-контрактов условно можно разделить на три класса, в зависимости от уровня, на котором обнаружена уязвимость. Для сети Ethereum это уязвимости соответствующих уровней: Solidity, EVM bytecode, blockchain. Каждый класс уязвимости может породить одну или несколько разновидностей атак.

На уровне блокчейна это *атака 51%*, *атака Сибиллы,* *DDOS-атаки*, *взлом криптографии*. В большинстве случаев разработчики приложений не могут каким-либо образом повлиять на эти атаки или предотвратить их. О данных атаках необходимо знать при выборе платформы под приложение, внимательно изучить, насколько платформа защищена от подобных атак, как сообщество видит пути развития платформы в плане безопасности.

*Атака 51%.* Если злоумышленник получает большую часть «голосов», то он может контролировать консенсус и включать в блокчейн только нужные ему данные. Подобные условия позволяют злоумышленнику провести еще одну атаку, известную как «Double-spending»: потратить больше денег, чем у него есть на самом деле. Для этого он создает несколько транзакций с использованием одних и тех же монет. По идее, сеть посчитает лишние транзакции некорректными и отклонит их: майнеры просто не включат их в блок. Тем не менее, если злоумышленник получает возможность разместить блок, то он может включить в него информацию с double-spending самостоятельно. В этом случае честные майнеры разветвят блокчейн и начнут строить параллельную валидную ветку, игнорируя блок атакующего. Однако, если у последнего в распоряжении оказывается 51% вычислительных мощностей, он сможет повлиять на консенсус и выстроить собственную цепь с «некорректными» транзакциями, которая будет считаться «истинной».

*Атака Сибиллы*. Она основывается на том, что сеть не может достоверно различать физические машины. Принимались попытки разработки механизмов для установления идентичности компьютеров: с помощью сертификационного ПО, IP-адресов, логинов и паролей, однако они не дали эффекта. Друзья могут передавать данные аккаунтов друг другу, а некоторые сервисы предоставляют один IP-адрес для всех своих пользователей. Вред от атаки Сибиллы проявляется по-разному. Для реализации атаки атакующий может попытаться заполнить сеть подконтрольными ему клиентами. Это позволяет использовать следующие схемы:

- Злоумышленник может отказаться передавать и принимать блоки, «отключив» пользователей от сети.

- Возникает опасность «атаки 51%» и double-spending.

- Появляется возможность видеть все транзакции, используя специальные программы.

Защита от атаки реализуется путем предъявления особых требований и ограничений к генерации блоков в протоколе блокчейна.

*DDOS-атаки –* это всем известные атаки, идея которых заключается в пересылке большого количества запросов с целью обрушения сети. В большинстве блокчейнов атаки блокируются путем введения ограничений на размер блока, количество проверок подписей, которое может затребовать блок, размерами скриптов, количеством мультиподписей.

*Взлом криптографии*. Производительность квантовых систем в определенном спектре задач выше, чем у классических компьютеров. Например, система D-Wave, использующая метод квантового отжига, в 100 миллионов раз быстрее обычных компьютеров. Одной из задач, которые необходимо учитывать в данном отношении, является проблема факторизации. Квантовые алгоритмы, например, алгоритм Шора, в теории смогут сломать RSA-шифрование и, как следствие, цифровые подписи, используемые в биткойн-сетях. Поэтому уже сегодня ведется разработка решений, цель которых — помочь криптопроектам устоять под натиском квантовых машин.

*На уровне языка написания контрактов* наиболее распространенным типом уязвимостей является *уязвимость рекурсивного вызова* , также известная как *Reentrancy*. Проблема кроется в том, что уязвимый контракт совершает вызов к другому контракту, при этом внешний контракт может делать ответный вызов функций уязвимого контракта внутри начального вызова. Часто можно столкнуться с заблуждением, что если функция не рекурсивная, то она не допустит многократных повторных выполнений. Однако, это заблуждение может привести к тому, что нерекурсивная функция запускает цикл вызовов, которые в конечном итоге потребляют все средства.

Также в Solidity, например, существуют уязвимости, приводящие к отправке криптовалюты на неустановленные адреса (call to the unknown), существующие ограничения могут приводить к незапланированной отмене уже осуществленных действий (exception disorder).

При отправлении транзакции в сеть пользователь не всегда может быть уверен в состоянии контракта, которое определяется стоимостью его полей и балансом. Это может произойти потому, что в момент отправки состояние контракта было изменено другой транзакцией, либо контракт содержит в себе динамические переменные, связанные с другими контрактами. Такая уязвимость может быть использована злоумышленниками для привязки вызываемого контракта к злонамеренным компонентам, которые позволяют красть криптовалюту.

Особое внимание необходимо уделять конфиденциальности информации, публикуемой в контрактах. Поля в контрактах могут быть как закрытыми, так и общедоступными для всех пользователей. Однако объявление поля закрытым не гарантирует его недоступность для других. Это связано с тем, что для установки приватности поля, пользователь должен направить соответствующую транзакцию майнерам, которые затем опубликуют ее в блокчейне. Поскольку сам блокчейн является общедоступным, то любой пользователь может проверить содержимое транзакции и внести изменения в приватность поля. Для того, чтобы наилучшим образом защитить поле контракта нужно использовать подходящие криптографические методы.

Известны методы как заставить пользователя авторизоваться в нужном злоумышленнику контракте или стать владельцем чужого контракта. Это относится к уязвимостям контроля доступа.

Как уже известно, после публикации контракта в блокчейне, его уже невозможно исправить, поэтому контракты с ошибками могут проявить себя совершенно непредсказуемо. Иногда, когда последствия выполнения таких контрактов крайне негативно сказывается на всем блокчейне, сообщество приходит к решению применения софтфорка или хардфорка.

**Будьте очень внимательны при тестировании смарт-контрактов перед публикацией. Помните, что опубликованный смарт-контракт невозможно отменить или изменить.**

Часть 2. Практика

Глава 1. Знакомство с языком Solidity.

Solidity — один из четырех языков для EVM (три других: Serpent, LLL и Mutan), являющийся кроссплатформенным языком, но на практике используемый преимущественно на Ethereum. Он является наиболее проработанным, хотя на данный момент не имеющим полноценной версии, поэтому многие функциональности в нем пока находятся в урезанном виде, а количество багов достаточно велико. Его синтаксис схож с JavaScript, что позволяет разработчикам быстро адаптироваться и приступить к работе.

*Solidity - это объектно-ориентированный высокоуровневый язык программирования для написания смарт контрактов, исполняющихся на виртуальной машине Ethereum (EVM), который был впервые предложен для этих целей в 2014 году. Solidity статически типизированный JavaScript-подобный язык программирования, на создание которого повлияли языки C ++, Python и JavaScript. Программы на языке Solidity транслируются в байткод EVM. Solidity позволяет разработчикам создавать самодостаточные приложения, содержащие бизнес-логику, результирующую в неотменяемые транзакционные записи блокчейна. Solidity статически типизирован, поддерживает наследование, включая множественное и C3-линеаризацию, библиотеки, бинарный интерфейс программирования (ABI), имеющий множество типобезопасных функций в каждом контракте, комплексные переменные контрактов, включая произвольные иерархические отображения (mappings) и структуры. С Solidity вы можете создавать контракты для таких целей, как голосование, краудфандинг, слепые аукционы и кошельки с несколькими подписями.*

При развертывании контрактов вы должны использовать последнюю версию Solidity. Это связано с тем, что разработчиками языка постоянно вносятся критические изменения, добавляются новые функции и исправляются ошибки.

С Solidity можно работать в браузере без потребности загружать или компилировать что-либо. Если же необходимо выполнить код или ввести его в блокчейн, следует использовать клиент, такой как Geth или Parity.

Контракты в Solidity - то же, что и классы в объектно-ориентированных языках. Они могут содержать так называемые переменные состояния, которые постоянно хранятся в хранилище вместе с контрактом и функциями, указателями входа для работы с этими переменными состояния. Кроме переменных состояния, существуют также локальные переменные, объявленные в функциях, содержание которых очищается как только поток управления возвращается из функции.. Solidity является статически типизированным языком, следовательно тип каждой переменной(состояния и локальных) должен быть указан (или по крайней мере известен ) во время компиляции. Solidity предоставляет несколько элементарных типов, которые могут быть объединены в составные типы.

**Элементарные типы (типы значений)**

Переменные таких типов всегда передаются по значению, т.е. они всегда копируются, в случае использования в качестве аргументов функции или в присвоениях.

*Логический тип*

bool: Возможные значения константы true, false

*Операторы:*

! - логическое отрицание

&& - логическая конъюнкция, "and"

|| - логическая дизъюнкция, "or"

== - равенство

!= - неравенство

Операторы || и && применяют обычные правила short-circuiting. Это означает, что в выражении ***f(x) || g(y)***, если f(x) оценивается как true, g(y) не будет рассмотрен даже в случае последствий.

*Числа*

int• / uint•: Целые числа со знаком и без знака различных размеров.

Ключевые слова uint8 до uint256 с шагом 8 (Целые числа без знака с 8 до максимум 256 битов) и int8 до int256. uint и int являются псевдонимами для uint256 и int256, соответственно.

*Операторы:*

Сравнения: <=, <, ==, !=, >=, > (возвращают bool)

Битовые операции: &, |, ^ (Побитовое исключающее ИЛИ), ~ (Побитовое отрицание)

Арифметические операторы: +, -, унарный -, унарный +, \*, /, % (Деление с остатком (деление по модулю) ), \*\* (Возведение в степень)

*Адресный тип*

address: Содержит 20 байтное значение (размер адреса Ethereum).

*Операторы:*

<=, <, ==, !=, >= и >.

*Массивы фиксированного размера*

bytes1, bytes2, bytes3, ..., bytes32: массивы фиксированного размера, byte является псевдонимом для bytes1.

*Операторы:*

Сравнения: <=, <, ==, !=, >=, > (возвращают bool)

Битовые операции: &, |, ^ (Побитовое исключающее ИЛИ), ~ (Побитовое отрицание)

*Динамично измеряемые массивы*

bytes: динамично измеряемый массив, см. Массивы. Не является типом значения!

string: динамично измеряемая UTF8-кодированная строка, см. Массивы. Не является типом значения!

*Целочисленные Литералы*

Целочисленные литералы являются целыми числами произвольной точности, пока они не используются вместе с не-литералом. В var x = 1 - 2;, например, 1 - 2 имеет значение -1, которое присваивается x и таким образом x получает тип int8 -- наименьший тип который может хранить -1, несмотря на то, что естественные типы 1 и 2 фактически uint8.

Возможно даже временно превысить максимум 256 битов до тех пор пока для вычисления используются целочисленные литералы: var x = (0xffffffffffffffffffff \* 0xffffffffffffffffffff) \* 0; Здесь, x будет иметь значение 0 и таким образом тип uint8.

*Строковые Литералы*

Строковые литералы записываются с двойными кавычками("abc"). Как с целочисленными литералами, тип может варьироваться, но они неявно конвертируемы в bytes• если помещаются, в bytes и в string.

**Глобальные переменные**

block.coinbase (address): адрес майнера блока

block.difficulty (uint): уровень сложности блока

block.gaslimit (uint): лимит газа

block.number (uint): номер блока

block.blockhash (function(uint) returns (bytes32)): хэш данного блока

block.timestamp (uint): временная метка блока

msg.data (bytes): выполнение calldata

msg.gas (uint): оставшийся газ

msg.sender (address): отправитель сообщения (текущий вызов)

msg.value (uint): количество wei отправляемых в сообщении

now (uint): настоящая временная метка (псевдоним для block.timestamp)

tx.gasprice (uint): цена газа для транзакции

tx.origin (address): отправитель транзакции (full call chain)

sha3(...) returns (bytes32): вычислить хэш Ethereum-SHA3 аргументов

sha256(...) returns (bytes32): вычислить хэш SHA256 аргументов

ripemd160(...) returns (bytes20): вычислить RIPEMD 256 аргументов

ecrecover(bytes32, byte, bytes32, bytes32) returns (address): восстановить открытый ключ из сигнатуры эллиптической кривой

this (current contract's type): текущий контракт, конвертируемый по адресу

super: контракт на один уровень выше в иерархии наследования

suicide(address): уничтожение текущего контракта и вывод средств по указанному адресу

<address>.balance: баланс адреса в Wei

<address>.send(uint256) returns (bool): отправить указанное количество Wei на адрес, вернуть false в случае неудачи.