

Содержание

Описание 3

Введение 4

Децентрализованный подход 6

Архитектурный дизайн 10

Ключевые термины 12

Архитектура OPEN API 14

Применение Scaffolds и OPEN_States 19

Кошелек разработчика и OPENToken 20

Зависимости от третьих сторон 23

Вопросы безопасности 26

Будущая функциональность OPEN 31

Следующие шаги 33

OPEN: Децентрализованная платформа платежей и приложений

В настоящем документе представлена система технологий OPEN, призванная облегчить принятие блоков путём создания уникальной платформы, которая позволит разработчикам интегрировать децентрализованные технологии. Платформа OPEN состоит из нескольких ключевых компонентов: это Шаблоны (Scaffolds), Создатели шаблонов (Scaffold Creators), и OPEN_States. Эти ключевые технологии работают вместе на платформе OPEN и обеспечивают комплексный подход к схемам децентрализованных платежей для любых приложений и услуг. Инфраструктура OPEN включает в себя пользовательские состояния, связанные с схемами ценообразования и атрибутами приложения, которые можно портировать в любой смарт-контракт, связанный с блокчейном. Таким образом возникает всеобъемлющая децентрализованная платформа для любых форматов платежей.

Кроме того, в настоящем документе описываются различные риски и ограничения безопасности для платформы OPEN касательно сетевых и внесетевых взаимодействий, с которыми разработчик может столкнуться в разных случаях применения. Платформа OPEN и API работают в связке, обеспечивая решение под ключ для разработчиков, стремящихся использовать мощь технологии blockchain в своём нынешнем арсенале технологий. Наряду с возможностью добавления любой тарифной схемы, основанной на пользовательских приложениях, платформа минимизирует воздействие волатильности цен и позволяет проверить состояние пользователя в сети относительно статуса платежа. OPEN позволяет разработчикам внедрять любую платежную схему, привязанную к любой валюте по своему выбору. Одним из самых больших препятствий для принятия криптовалют разработчиками была невероятная волатильность цен криптовалют. Архитектура OPEN спроектирована таким образом, что воздействие волатильности на разработчика сводится к минимуму.

Архитектура, представленная в этом документе, представляет новый децентрализованный платежный шлюз для приложений, который легко интегрируется с существующими бэкендами приложений. Кроме того, архитектура OPEN публикует доказательства действительных транзакций, содержащих информацию, связанную с транзакцией, и это открывает безграничные варианты использования приложений. Основным препятствием для широкомасштабного внедрения подобных платформ была сложность интеграции. Многие из них требовали понимания технологии блокчейна, исходя

из предположения, что люди будут ее изучать. Это предположение было ошибочным. Даже для большинства технически продвинутых пользователей не хватает стимулов для изучения навыков, необходимых для внедрения блокчейн-технологий. Команда OPEN признает эту проблему и использует свой многолетний опыт разработки, чтобы устранить болевые точки, препятствующие развитию блокчейн-проектов, способных преобразить наше общество. Команда OPEN считает, что обеспечив взаимодействие между платформами, инструментальные средства для содействия разработчикам, а также увеличив пользовательский охват, в выигрыше будут все.

Введение

Технологии блокчейн могут дать невероятные преимущества разработчикам приложений. Существуют технические ограничения для децентрализованных решений функционирования приложений в обычных условиях. В настоящее время отсутствуют простые решения, позволяющие разработчикам интегрировать технологии блокчейн в свою основную инфраструктуру. Отсутствует инфраструктура для применения конкретной схемы ценообразования приложения и определения пользователей, удовлетворяющих условиям этой конкретной схемы на децентрализованной платформе, то есть имеющих достаточно средств в валюте приложения. В частности, разработчикам требуется возможность интегрировать децентрализованный платежный шлюз в собственную схему платежей своих приложений, который сможет принимать любые криптовалюты и синхронизировать такие транзакции с их существующим бэкэндом.

Более того, централизованные системы демонстрируют невероятную неэффективность, которую можно быть легко устранить, включив аспекты децентрализованного стека. Рассмотрим пример использования централизованной системы для размещения или работы приложения. Часто разработчики сталкиваются с долгим процессом аудита в рамках централизованной структуры, в которой каждое обновление приложения тщательно проверяется, что приводит к длительным задержкам публикации последних обновлений, содержащих более эффективную структуру данных или модель ценообразования. Существуют и другие проблемы с шлюзами проверки платежей, поскольку текущий поток покупок в приложениях проходит через множество процессов и посредников, увеличивая время и затраты на достижение учетной записи разработчика. OPEN обеспечивает решение этой проблемы, используя системы сложных смарт-контрактов как инструмент транзакций и верификаций для программных приложений. Платежи не просто ускоряются, снижаются также транзакционные издержки благодаря дополнительным преимуществам безопасной среды. Это дает уникальные преимущества разработчику приложения и его пользователям.

Текущая платежная инфраструктура



Серая область представляет собой транзакции, происходящие через Интернет, в то время как сплошная область представляет собой частную финансовую сеть. Сплошные стрелки представляют транзакции в режиме реального времени, а пунктирные стрелки - нет..

Как видно на схеме выше, текущий процесс платежей в приложении требует многочисленных шагов и значительных инфраструктурных затрат. Этот процесс часто стоит разработчику более 30% издержек, если рассматривать заявку в мобильном приложении, использующем традиционный магазин приложений. Такие транзакции могут длиться 30-60 дней до тех пор, пока оставшиеся средства не будут наконец доставлены и доступны разработчику. Кроме того, геолокационные границы и правила, установленные платформами для увеличения собственной прибыли, ограничивают эффективность приложения и пользовательский опыт. Технология блокчейн позволяет значительно упростить и демократизировать этот процесс, что приведет к значительной экономии времени и денег.

Включенная инфраструктура OPEN



На приведенной выше схеме показана платформа OPEN во взаимодействии с блокчейном, формирующая децентрализованный шлюз для проверки платежей, который может придать гибкость многочисленным схемам ценообразования в любых традиционных или нетрадиционных приложениях. Кроме того, так возникает децентрализованное сложное решение, которое надежно связывает платежи пользователей с централизованной базой данных, что часто требуется архитектурой приложений при принятии платежей на основе криптовалюты. Эта архитектура использует блокчейн для того, с чем он лучше справляется: он способен быть неизменяемым якорем данных, применяющим текущую централизованную инфраструктуру для обработки других потребностей приложения, таких как хранение данных и масштабируемая вычислительная мощность. В данный момент не нужно, неэффективно и нереалистично децентрализовать всё в этом стеке, и в результате OPEN создала модульное решение, которое можно легко интегрировать в существующие приложения, позволяя множеству традиционных приложений использовать архитектуру блокчейн. OPEN создает необходимое решение для платежных шлюзов разработчика, полностью самостоятельное и простое в реализации на уже

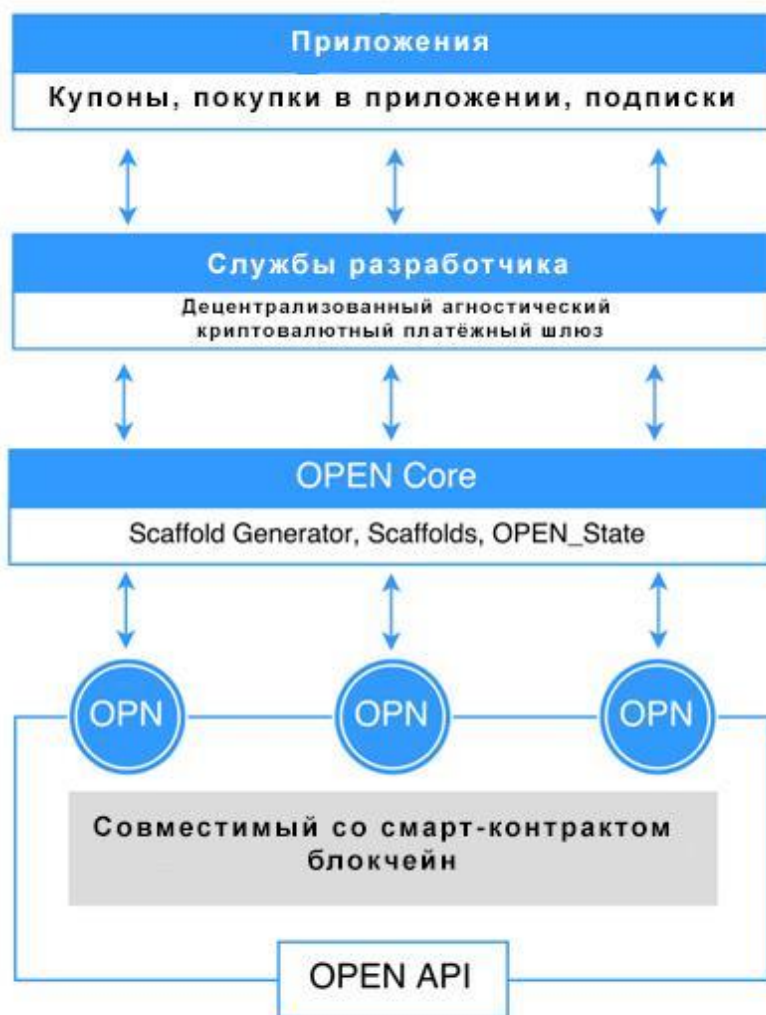
существующей инфраструктуре. Переключение с централизованного на децентрализованный платежный шлюз с помощью этого метода может быть таким же простым, как вызов функции, и это даст уникальные преимущества с минимальными затратами ресурсов на интеграцию.

Децентрализованный подход

Используя мощь децентрализации, можно создать надежную экосистему, которая позволит любому разработчику приложений развернуть настроенные платежные схемы через децентрализованную сеть и принимать криптовалюты в качестве способа оплаты. С дальнейшим повышением надежности информации, устранением отдельных сбоев и обеспечением такого уровня целостности данных, который невозможно воспроизвести в централизованных архитектурах приложений. Предоставляя разработчикам привязку своей существующей централизованной базы данных к блокчейну, OPEN предоставляет возможность любому разработчику программного обеспечения подключаться к технологиям блокчейн без существенного изменения текущих бэкэнд-процессов. Также разработчики могут немедленно внедрить существующие приложения в блокчейн, инфраструктура OPEN предоставляет платформу «под ключ» там, где разработчик и приложение нуждаются в этом. Это выводит совместимость всей экосистемы на беспрецедентный уровень, позволяя разработчикам создавать множество цифровых продуктов без посредников, устанавливающих значительные тарифы и ограничения.

OPEN создала агностическую платформу блокчейн с полным API, разработанным специально для упрощения доступа и интеграции в уже существующую инфраструктуру приложений. OPEN API устраняет сложности децентрализованной архитектуры платформы OPEN, можно выйти на прикладной уровень через децентрализованную платформу, с помощью которой разработчики могут подключаться к своим существующим базам кодов. Это позволяет разработчику отключать централизованные функции, такие как платежный шлюз или хостинг, для функций, которые используют децентрализованную платформу приложений OPEN. На данный момент функциональность, которую OPEN стремится предоставить своей платформой и API, в основном касается платежных шлюзов. OPEN API будет использовать функциональные возможности, уже присутствующие во внешних библиотеках, таких как web3.js и ETHJS, для разработки безопасного способа взаимодействия с блокчейном. Реализовав это, OPEN откроет потенциал технологии блокчейн, способный коренным образом изменить способ создания, размещения, проверки и монетизации приложений.

Обзор модели OPEN Platform. Разные слои от совместимого со смарт-контрактами блокчейна до ядра OPEN Core, в котором работает наш уникальный Scaffold. Службы разработчика включают принятие различных криптовалют на любой платежной схеме приложения. Приложения,



использующие шаблон Scaffold разработчика, расположены поверх служб разработчика в рамках экосистемы OPEN. Аспекты каждой службы и их взаимодействия с приложениями, пользователями и разработчиками, более подробно описаны ниже.

Архитектурный дизайн

Модульность является ключевой точкой реализации OPEN. Децентрализованный бэкэнд должен быть максимально гибким, чтобы разработчики взаимодействовали с API, который упрощает технические моменты использования блокчейна. Платформа OPEN использует несколько различных компонентов архитектуры блокчейна, наряду с протоколом Scaffold, который выступает в качестве платежной инфраструктуры и схемы проверки для конкретного приложения, и OPEN_State, который действует как проверенный вывод, содержащий желаемый результат транзакции или состояние любой схемы оплаты для пользователя приложения. На высоком уровне OPEN API служит связующим звеном между этими компонентами на блокчейне и текущей инфраструктурой разработчика.

OPEN задействует Scaffolds в качестве шаблонных смарт-контрактов, которые можно создавать под требуемые разработчиком спецификации, с помощью простого вызова API. OPEN использует настраиваемые шаблоны Scaffolds как компонент, который дает разработчикам уникальную гибкость работы со структурами и типами цен, а также транзакциями, необходимыми для широкого спектра приложений. Кроме того, поскольку OPEN устанавливает в своей архитектуре всего одну точку, с которой пользователь взаимодействует через OPEN API, становится очень просто отключать и менять шаблоны Scaffolds, когда в приложении появляются новые функции. Шаблон Scaffold может запросто представлять схему оплаты подписки, схему ценообразования на корпоративное программное обеспечение или схему ценообразования в игровой валюте. Красота этого подхода возникает из децентрализованности, при которой пользователь может заплатить в рамках Scaffold и затем этот платеж может быть проверен на блокчейне с помощью OPEN API. Это не только увеличивает скорость транзакций, но и предоставляет разработчикам, приложениям и пользователям децентрализованную среду оплаты, использующую OPEN API для проверки состояния пользовательского платежа на блокчейне.

Шаблоны Scaffold позволяют нашей технологии работать с любым блокчейном, который поддерживает смарт-контракты. Элегантность, свойственная Scaffolds, - это не только их способность обеспечить беспрепятственный рабочий процесс, автоматически реагируя на любое взаимодействие с пользователем, но и возможность упростить процесс настройки приложения. Особая роль Scaffold, его техническая реализация и взаимодействие в экосистеме будут описаны на следующих страницах.

OPENToken - это токен утилиты, который позволяет разработчику активировать Scaffold после его создания. Благодаря тому, что функция токена ориентирована на разработчика, введение нового токена не становится препятствием для конечного пользователя.. Это дает OPEN возможность стать катализатором внедрения и интеграции блокчейна среди основных

разработчиков программного обеспечения. Разработчик несет ответственность за покупку OPENTokens и использование их для создания шаблонов Scaffold в рамках своих приложений.

Данный документ использует слово «состояние», имея в виду конкретную информацию, которая дает контекст пользователю в приложении. Например, «состояние» пользователя в мобильном игровом приложении может включать в себя количество валюты в приложении, которое может быть у пользователя, независимо от того, имеет ли он премиальный план или другие приобретения в приложении. Просматривая информацию через этот объектив, становится проще перевести действия пользователя в данные о блокчейне. Вместо представления атрибутов как множества разных токенов, которые перегружают экосистему и добавляют слой избыточной сложности, можно привязать данные к транзакции, которая использует встроенный смарт-контракт для упрощения изменения значений одного из этих состояний. Таким образом, любая операция, которую пользователь делает в приложении, может быть представлена как транзакция в архитектуре блокчейна OPEN. Это обеспечивает эффективный, линейный и, самое главное, безопасный рабочий процесс. Чтобы немного расширить пример использования выше, допустим, что пользователь приложения желает получить дополнительную сумму в валюте приложения. Вместо того, чтобы представить внутриигровую валюту или драгоценные камни как своего рода индивидуальный токен, создав чрезмерное количество уникальных токенов, этот метод представляет количество драгоценных камней, которое имеет пользователь, в качестве переменной в состоянии пользователя. Поступая таким образом, пользователь может просто заплатить за Scaffold, который опубликует новое состояние для этого пользователя на блокчейне, что означает, что пользователь действительно заплатил и имеет право на эту новую валюту. Этот опыт не только прост и интуитивен, но и защищает от потенциально вредных действий нечестных пользователей.

OPEN API предоставляет разработчикам возможность создавать индивидуальные Scaffolds из шаблонных Scaffolds, которые могут соответствовать различным схемам платежей. Вызов OPEN API - это начальная точка взаимодействия разработчика для создания Scaffold,, соответствующего требованиям разработчика или приложения. Наличие схемы взаимодействия на высоком уровне позволяет разработчикам изменять или обновлять некоторые аспекты приложения. Они могут просто закрыть оригинальный Scaffold и создать новый Scaffold с обновленными спецификациями через простой и знакомый вызов API. Смысл этой системы заключается в том, что она позволяет разработчикам изменять свою схему приложений через OPEN API без необходимости перераспределения всей структуры платежей.

Ключевые термины

OPENWallet:

OPENWallet - это блок хранения для криптовалют пользователя (например, ETH, BTC и т. д.), А также OPEN_States, которые пользователь приобрел через OPEN API. Эти OPEN_States обеспечивают подтверждение оплаты пользователя в разных приложениях. OPENWallet обеспечивает интуитивно понятный пользовательский интерфейс, который освобождает от необходимости узнавать, как работает блокчейн, и поддерживает тот же пользовательский интерфейс, что и текущие централизованные реализации.

Кошелёк разработчика (Developer Wallet):

Кошелек разработчика - это кошелек (адрес), на который Scaffold передает средства, которые пользователь отправил ему для OPEN_State (см. схему ниже). В то время как OPEN позволяет разработчикам использовать любой кошелек, разработчик должен использовать OPENWallet, поскольку он использует OmiseGo SDK, который имеет встроенный децентрализованный обмен криптовалютами и огромную поддержку сообщества. Благодаря SDK OmiseGo Wallet разработчик может автоматически конвертировать любой перенос Scaffold в любую желаемую криптовалюту или фиатную валюту.

OPENToken:

Токен ERC20, который будет распространен в рамках OPEN ICO и позволит разработчикам создавать Scaffolds в экосистеме OPEN. По мере развития платформы, OPENToken может получить дополнительную ценность, например, предоставить пользователям, которые платят в OPENToken, скидку на приобретенные услуги.

OPEN_State:

Смарт-контракт, в котором хранится криптографически хэшированное состояние пользователя в приложении (реализации, для которых может пригодиться OPEN_State, рассмотрены в разделе «Дальнейшие улучшения»). Текущее состояние базы данных приложения на момент покупки может быть закрыто с использованием криптографического хеша, а затем привязано к блокчейну. OPEN_State будет содержать информацию, позволяющую проверить, принадлежит ли Open_State определенному пользователю и Scaffold. Эта информация включает адрес пользователя OpenWallet и адрес Scaffold.

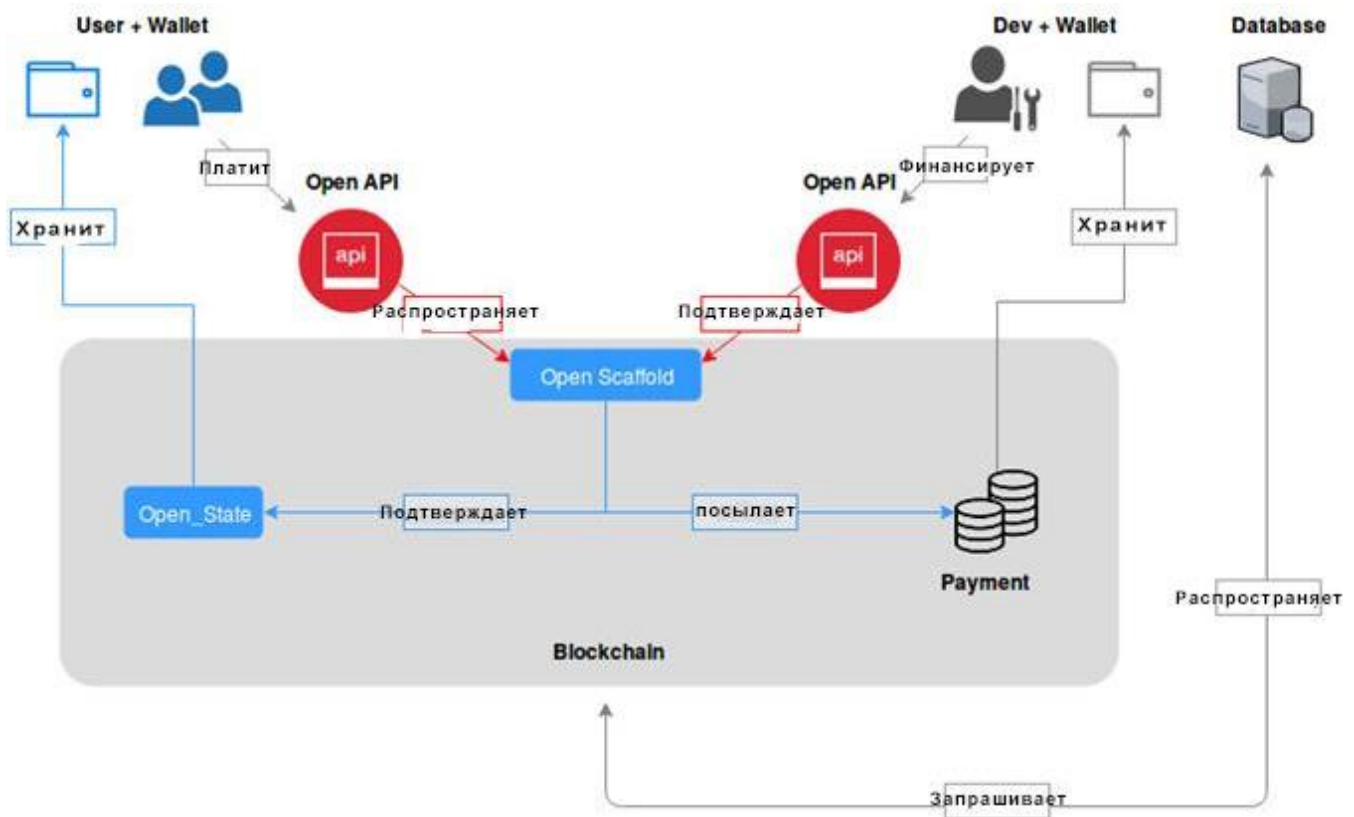
Scaffold:

Шаблонный смарт-контракт, который создается разработчиком и включает в себя информацию о конкретном приложении, такую как схема ценообразования или количество драгоценных камней, принадлежащих сети. Scaffold создается разработчиком, когда разработчик платит за создателя Scaffold (Scaffold Creator). Scaffold становится активными, когда на него начисляется достаточное количество OPENTokens. Процесс финансирования Scaffold подразумевает привязку определенного количества OPEN Tokens в Scaffold для его запуска. Эти OPENTokens должны поддерживать активность, но будут возвращены разработчику, когда он деактивирует Scaffold. Scaffold несет ответственность за создание OPEN_States для приложения-разработчика. Таким образом, Scaffold нуждается в конкретной схеме ценообразования для OPEN_State (например, 10 долларов США за 10 драгоценных камней), и он использует эту схему ценообразования в качестве способа проверки, достаточно ли внесено средств. Scaffold также несет ответственность за передачу платежа, предоставленного ему для OPEN_State, в кошелек разработчика. Для Scaffold требуется адрес разработчика для передачи средств, схем ценообразования и доступа к ценовому оракулу, но он также может содержать любое количество дополнительных переменных и данных.

Scaffold Creator:

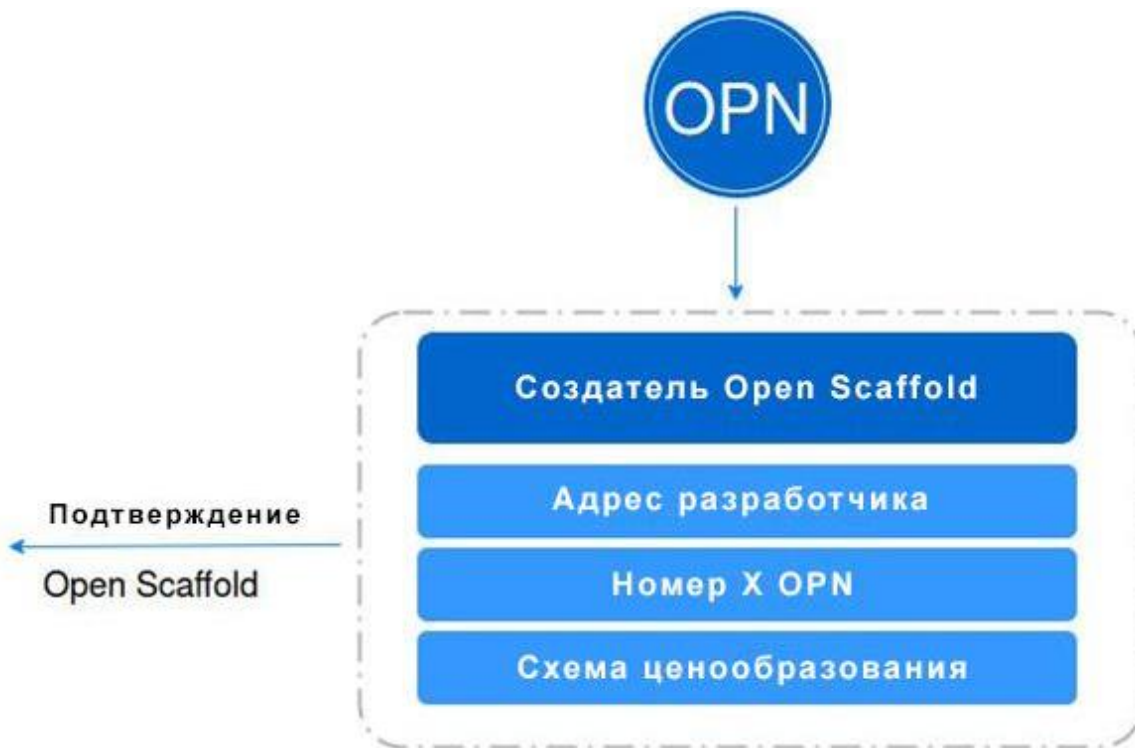
API, который OPEN предоставляет разработчикам для создания Scaffolds с помощью простого вызова API. Scaffold Creator может создавать Scaffolds с ETH, но для того, чтобы Scaffolds стали активными, необходимо профинансировать их с помощью OPENTokens.

Архитектура OPEN API



Высокоуровневый обзор рабочего процесса для разработчиков и для пользователей приложений через архитектуру OPEN, а также через API. Подробные сведения о наиболее важных взаимодействиях рассмотрены ниже..

Создание Scaffold разработчиком

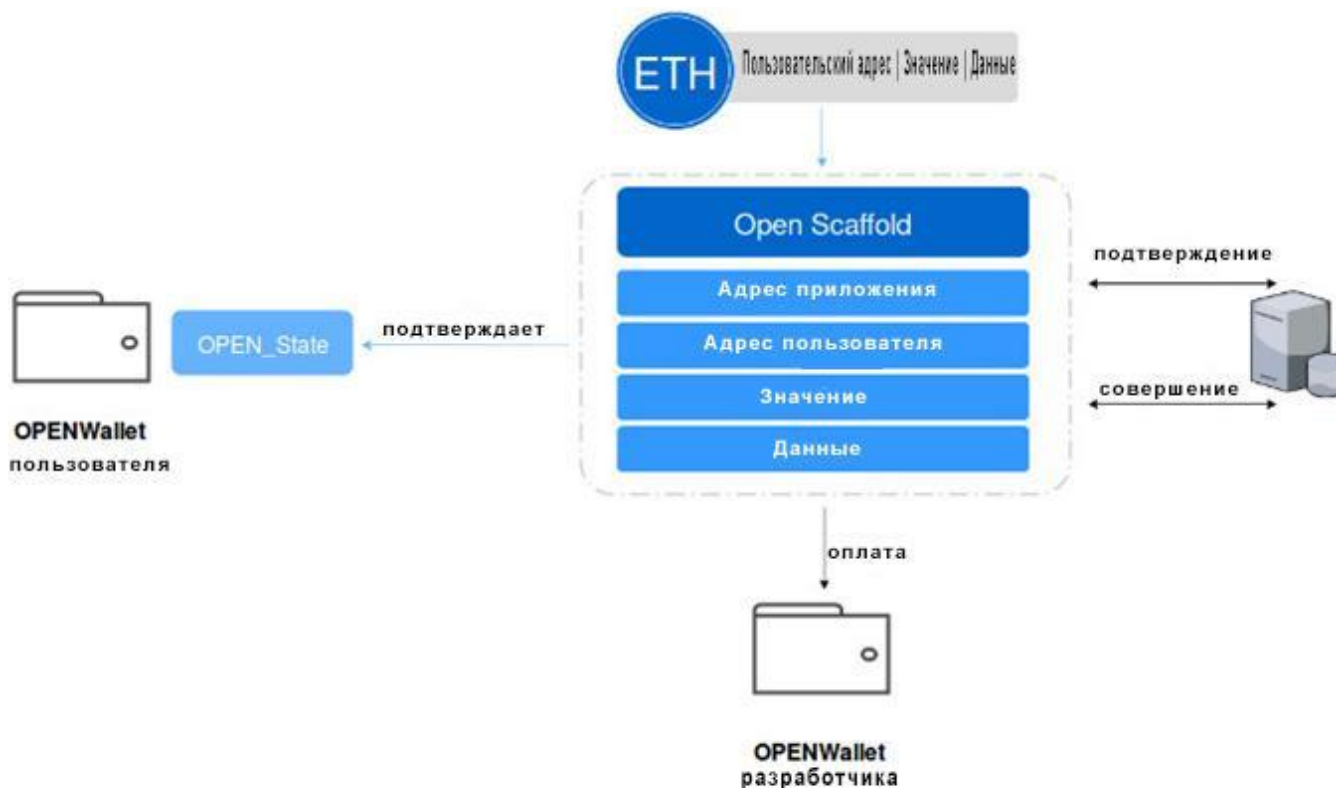


OPEN API будет иметь встроенную функциональность, которая даст разработчикам возможность создавать новые Scaffolds, установленные на адрес разработчика в качестве владельца. Используя OPENTokens для инициализации Scaffold, разработчик может создать уникальный Scaffold в экосистеме блокчейн.

Второй шаблонный контракт - это Scaffold. Как указано выше, Scaffold создается через API с помощью OPENTokens. Сам Scaffold принимает транзакцию, которая представляет собой платеж, предоставляемый конечным пользователем, а также некоторые данные, представляющие действие, которое пользователь хочет принять. Это распространяется через API. Затем Scaffold создает OPEN_State, который представляет результат выполнения указанного действия, и сохраняет его децентрализованным образом, чтобы уменьшить накладные расходы, с которыми сталкивается разработчик, и использовать целостность и строгость данных, предлагаемые блокчейном. OPEN_State хранится в OPENWallet пользователя, чтобы интуитивно организовать данные и повысить эффективность выполнения за счет сокращения времени запроса. С точки зрения пользователя, пользовательский кошелек может быть пользовательским интерфейсом, похожим на тот, который функционирует в Google Play или Apple App Store. Последней особенностью Scaffold является то, что как только он проверяет правильность суммы платежа, он не хранит криптовалюты, а автоматически передает средства на кошелек разработчика, что обеспечивает мгновенную ликвидность и защиту от волатильности для приложения и

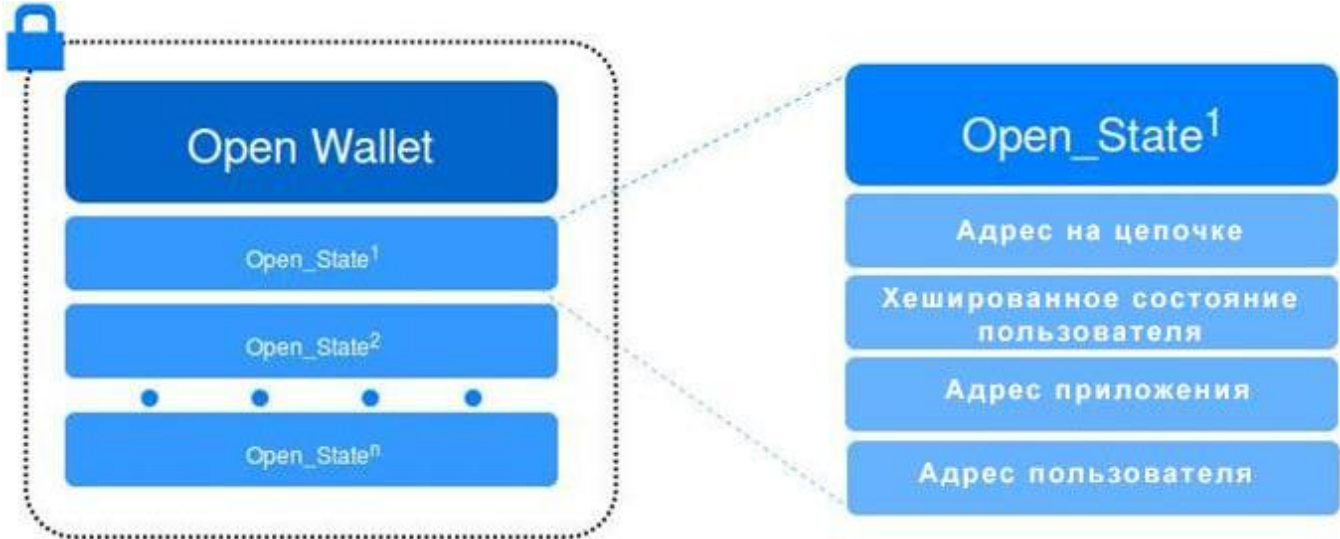
разработчика. Этот почти мгновенный переход от кошелька клиента к разработчику - значительное улучшение по сравнению с текущей платежной системой, которая может занимать 30-60 на перемещение средств на счет разработчика, при одновременном смягчении проблем волатильности, которые могут возникнуть в такой период при работе с криптовалютами.

Взаимодействие пользователя со Scaffold



На этой схеме показан пользователь, отправляющий платеж ETH в Scaffold разработчика через OPEN API. Поскольку платеж пользователя может быть обработан через децентрализованный обменный центр OmiseGo (OMG), пользователь может заплатить различными криптовалютами и конвертировать этот платеж в ETH, чтобы оплатить Scaffold. Затем система управления базами данных разработчика может передать изменения, проверив OPEN_State адреса пользователя и обновив соответствующее состояние, расшифровав информацию ключом, предоставленным ему разработчиком. Одновременно ETH-платеж будет отправлен из Scaffold в кошелек разработчика (который может быть выбран разработчиком), где можно будет выбрать сохранение суммы, немедленную конвертацию в доллары США или в другую криптовалюту, которую предпочитает разработчик.

Схема OPENWallet + OPENState



OPENWallet представляет собой коллекцию различных состояний пользователя для каждого приложения, в котором он участвует, эти состояния представлены как уникальные смарт-контракты. Каждое состояние, называемое OPEN_State, является смарт-контрактом в сети Ethereum, который привязывает данные к блокчейну и позволяет проводить подключения и проверки на текущем бэкэнде разработчика. Это достигается путем передачи данных в Scaffolds. Модульность обеспечивает гибкость и итеративные возможности.

Применение Scaffolds и OPEN_States

OPEN_States и Scaffolds, которые составляют платформу OPEN, открывают возможность применения множества цифровых сервисов. Ниже приведен пример реализации OPEN_States в приложении потоковой передачи видео, но другие примеры включают в себя: добавление валют в приложении, создание лицензий на программное обеспечение, распространение купонов и предоставление частичного администраторского доступа к базам данных.

Разрешения для потоковой передачи видео (разрешения для использования онлайн):

Через платформу OPEN любой издатель видео, например Джон, который хочет монетизировать свой контент, может сделать это, не беспокоясь о сложном платежном бэкэнде. В этой структуре Джон и его приложение иницируют Scaffold, чтобы дать пользователю доступ к первому сезону шоу, которое он выпускает. Джон иницирует этот Scaffold, введя стоимость доступа к этой службе потоковой передачи и переменное значение, указывающее, имеет ли пользователь доступ.

Когда пользователь оплачивает этот Scaffold, сумма денег, которую он заплатил, проверяется через оракул (oracle), чтобы узнать, соответствует ли она сумме, указанной Джоном. Если это так, создается OPEN_State, которое отправляется в кошелек покупателя, обеспечивая покупателю доступ к контенту. На данный момент база данных Джона должна лишь сделать запрос в OPEN_State, чтобы узнать, существует ли данное состояние, и если это так, отредактировать свою собственную базу данных, чтобы отразить сигнал OPEN_State (например, покупатель может смотреть Сезон 1). После этого изменения покупатель имеет доступ к видео и может просматривать его.

Эффективность этой системы - это лучшая упорядоченность по сравнению с существующими платежными шлюзами. Кроме того, высокая эффективность OPEN приводит к уменьшению числа посредников, требующих непомерные взносы, которые мешают Джону получить истинную стоимость его контента.

Кошелек разработчика и OPENToken

Кошелек разработчика может быть любым онлайн-кошельком, способным получать криптовалюты и держать OPENToken. Хотя OPEN позволяет разработчикам использовать любой кошелек, рекомендуется использовать OPENWallet, поскольку он использует OmiseGo SDK, который имеет отличную поддержку сообщества и встроенный децентрализованный обмен. OmiseGo SDK является общедоступным для использования и легко реализуется. Важно отметить, что архитектура OPENWallet, однако, универсальна для любого кошелька для широкого использования.

Кошелек разработчика используется в трех разных сценариях. Первый сценарий: разработчик хочет создать новый Scaffold, профинансировав Scaffold Creator. Эта сделка будет включать фиксированный платеж в OPENTokens вместе с некоторыми данными и будет передоваться с помощью OPEN API в Scaffold Creator, эффективно абстрагируясь от лежащей в основе технологии blockchain.

Второй сценарий, в котором используется кошелек разработчика, - это когда разработчик получает платежи от пользователя. Как упоминалось выше, такая транзакция будет отправляться с помощью Scaffold, на который платит пользователь. Как только Scaffold подтвердит, что выплаченная сумма достаточна для запрашиваемого действия, он создает экземпляр OPEN_State и сразу же пришлет платеж на адрес кошелька разработчика. OPEN использует автоматическую ликвидацию для снижения воздействия волатильности. Риск волатильности имеет важное значение для решения в децентрализованной экосистеме разработчика, поскольку криптовалюты могут быть чрезвычайно изменчивыми, а ценность, которую они имеют, подвержена непредсказуемым колебаниям. Это является препятствием для внедрения блокчейна, и для его смягчения OPEN сводит к минимуму время, когда криптовалюта блокируется в своей децентрализованной архитектуре. Немедленно перенаправляя все платежи в кошелек разработчика во время транзакции, разработчик может затем выбрать, хочет он хранить криптовалюты как есть, или конвертировать их в другую криптовалюту или фиат. Это позволяет повысить функциональность и гибкость для разработчика.

Третий сценарий заключается в том, что кошелек разработчика будет использоваться для ведения учета транзакций, которые проходят через Scaffold приложения. Поскольку транзакции являются общедоступными для всех в блокчейне, можно построить кошелек с функциями анализа всех взаимодействий пользователей со Scaffold, и отсюда предоставить разработчику ценную статистику использования. Например, разработчик может узнать, сколько пользователей взаимодействует со Scaffold, как часто они взаимодействуют, средний размер транзакции, данные о доходах и т. д. Эта ключевая функциональность делает экосистему OPEN конкурентоспособной с существующими централизованными магазинами приложений.

OPENToken

OPENToken используется в первую очередь разработчиками при инициализации Scaffolds. На высоком уровне разработчик может создавать Scaffold с помощью OPEN API, но чтобы его использовать, разработчик должен профинансировать его некоторым количеством OPENToken. При размещении этого количества OPENToken, оракул (oracle) регистрирует рыночную цену OPENToken, и когда разработчик захочет деактивировать Scaffold, он сможет закрыть его и получить обратно вложенные OPENTokens.

Чтобы ограничить быструю активацию и деактивацию Scaffold, OPENTokens, которые размещены в Scaffold, будут заблокированы в течение определенного периода времени. После периода блокировки разработчик сможет свободно закрыть свой Scaffold или сохранить OPENTokens в Scaffold.

Поскольку ценность OPENToken может колебаться в течение периода времени хранения токенов, ценность OPENToken, удерживаемых Scaffold, может отличаться от ценности в тот момент времени, когда они были вложены. Платформа OPEN решает проблему волатильности цены OPENToken с помощью механизм отделения, когда разработчик закрывает Scaffold. Если стоимость OPENToken уменьшилась или сохранилась с момента активации в Scaffold, разработчик получает обратно то же количество OPENTokens, которое он вложил в Scaffold. Если стоимость OPENToken увеличилась с момента активации Scaffold, прибыль приобретает разработчиком. Определенный процент прибыли разработчика отделяется и распределяется равномерно среди пользователей как членов сообщества, у которых есть положительный баланс OPENToken. Разработчик получает обратно первоначальную стоимость своих OPENTokens плюс прибыль, которая не была отделена. Эта функция OPEN, распространение и распределение по воздуху пользователям с положительным балансом OPENToken служит для сохранения ликвидности в экосистеме и поощрения участия в сообществе. Эта функция также дает стимул разработчикам для использования Scaffold и обеспечивает всесторонний подход ко всей OPEN-экосистеме. Более того, вложение токенов в Scaffold спасает разработчиков от создания ненужных Scaffolds в системе OPEN.

Кроме того, OPENToken может предоставлять утилиту, если используются соответствующие Scaffolds, для получения скидок или денежных бонусов лицам, которые хранят OPENTokens в своем OPENWallet, или оплачивают Scaffolds в OPENTokens. Эта добавленная утилиты даст стимул пользователям покупать и использовать криптовалюты, такие как ETH и BTC, и предпочитать OPENTokens фиатным валютам для поддержки обычного приложения в децентрализованной среде. Кроме того, для пользователей имеет смысл удерживать OPENTokens, потому что каждый раз при закрытии Scaffold, новые OPENTokens будут распределяться по всем кошелькам с достаточным балансом OPENTokens.

Зависимости от третьих сторон

Платформа OPEN построена с возможностью взаимодействия и модульности, так что текущие компоненты можно будет легко заменить при появлении более эффективных технологий в будущем. Эта модульность дает нам необходимую гибкость, чтобы адаптироваться к достижениям в индустрии, сохраняя при этом безопасность независимо от базового протокола. Таким образом, OPEN действительно сможет создать сеть приложений или слой, привлекательный для всей отрасли. При проектировании нашей архитектуры большое внимание было уделено тому, какие сторонние платформы OPEN может в настоящее время эффективнее использовать. Ниже приведен список сторонних зависимостей, которые помогают платформе OPEN, расширяя ее возможности и цели.

OmiseGo White-Label Wallet SDK

OPEN использует SDK OmiseGo кошелька для подключения к децентрализованному обмену криптовалютами OmiseGo (DEX). Это дает пользователям возможность оплачивать Scaffolds (смарт-контракты, которые действуют как платежные шлюзы) множеством различных криптовалют. Более того, используя OmiseGo DEX, разработчики могут конвертировать ETH, который они получают от пользователей, в другие криптовалюты или фиат.

<https://omg.omise.co/>

Tendermint

Хотя Tendermint не требуется для реализации OPEN, его использование связано с высокой пропускной способностью, низкими комиссиями за транзакцию и возможностью запуска виртуальной машины Ethereum (EVM) через Ethermint. Tendermint предоставляет возможность репликации машин состояния (например, EVM) на своем блокчейне, что позволяет использовать смарт-контракты, написанные в Solidity, и это делает переход от Ethereum к Ethermint простой операцией, которая обеспечивает значительное увеличение пропускной способности и снижает затраты. Эти сокращенные транзакционные издержки приведут к снижению затрат для разработчиков, принимающих платежи через OPEN, а увеличенная пропускная способность ускорит обработку транзакций по цепочке. Это поможет решить проблемы внедрения и масштабируемости, которые возникают при приеме большого количества платежей сразу. Имеет смысл использовать Ethereum для PoS, Tendermint для увеличения пропускной способности в ходе тестирования платформы OPEN по мере ее расширения, и в итоге вернуться к сети Ethereum в дальнейшем по мере реализации дорожной карты Ethereum. Возврат к сети Ethereum после реализации алгоритма Proof-of-Stake, будет зависеть от того, какая платформа будет предлагать лучшее время сделки и комиссию.

<https://tendermint.com/>

GNOSIS, Oraclize, ChainLink

Так как цена криптовалют изменчива, имеет смысл привязывать цены в рамках Scaffolds к фиатной валюте. Это спасает пользователей от разочарования, если цены вырастут, и они вынуждены будут должны заплатить больше за покупки, и это же спасает разработчиков от разочарования, если цены снизятся, и они получают меньше денег с одних и тех же покупок. Для решения проблемы волатильности необходимо иметь надежный источник данных по цене различных криптовалют. Этот надежный источник может использовать правильные цены в режиме реального времени, чтобы подтвердить операцию пользователя. Чтобы сделать это, требуется оракул (oracle) для запроса API. Есть несколько потенциальных oracle сетей, которые можно использовать в Open API. Они включают в себя централизованный oracle от Gnosis, oracle, предоставленный Oraclize, и сеть oracle, представленная Chainlink. Выбор oracle будет зависеть от скорости, при которой oracle может получить данные о ценах, а также безопасности самого oracle. Oracle, который не сможет достичь высокой пропускной способности, создаст узкое место для сделок и может дать неточные данные о ценах, а Oracle с недостаточной безопасностью данных представляет очевидную угрозу.

<https://gnosis.pm/>

<http://www.oraclize.it/>

<https://link.smartcontract.com/#chainlink>

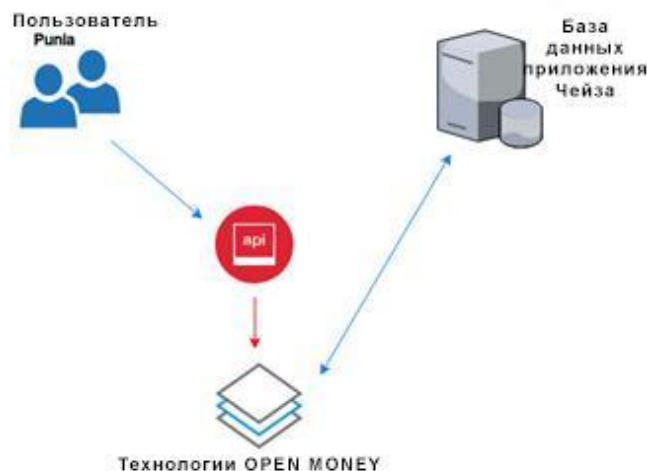
Web3, ETHJS

OPEN намеревается использовать такие существующие библиотеки, как Web3 и ETHJS для взаимодействия с Scaffold Creator, Scaffold, и OPEN_State Tokens. Это делается, чтобы исключить ошибки, которые возникают при добавлении собственных связей, и мотивировать разработчиков на помощь в развитии платформы OPEN. Web3.js является Javascript API для Ethereum и обеспечивает взаимодействие через JSON, предоставляя Scaffold и смарт-контракты Scaffold, умные с помощью JSON RPC. ETHJS является Javascript-библиотекой, основанной на web3.js, но имеет более облегченную реализацию для взаимодействия с Ethereum.

<https://github.com/ethereum/web3.js/>

<https://github.com/ethjs/ethjs>

Простой пример использования



1. Чейз является разработчиком, который хочет создать способ оплаты для своего нового приложения потокового видео, чтобы пользователи платили за подписку.
2. Чейз решает использовать открытую платформу для управления своими платежными протоколами.
3. Он использует Open API для отправки ETH на Creator Scaffold и получения Scaffold для своего приложения.
4. С помощью Open API и его дружественного интерфейса, Чейз добавляет схему ценообразования, которая дает каждому пользователю возможность получить состояние подписчика в базе данных за \$10.
5. Затем Чейз использует Open API, чтобы профинансировать этот Scaffold, купив OPENTokens и вложив их определенное количество в Scaffold.
6. Когда пользователь Punia, решает стать абонентом через OPEN, он создает OPENWallet с помощью OPEN API. Punia затем отправляет платеж через OPENWallet, а затем OPENWallet направляет платеж, а также желание Punia стать абонентом, на Scaffold Чейза.
7. Scaffold получит оплату и убедится, что пользователь Punia заплатил достаточно денег, чтобы стать абонентом, через oracle.
8. Если оплата пользователя Punia подтвердится, то Scaffold создаст экземпляр OPEN_State с необходимыми переменными, и отправит его в OPENWallet пользователя Punia.
9. Теперь OPEN API передаст базе данных Чейза информацию о том, что произошло обновление и сможет обеспечить возможность проверки достоверности обновлений, гарантируя, что они записаны на блокчейн.
10. Чейз принимает средства в ETH от пользователя Punia и может конвертировать их в фиатную или криптовалюту, сразу же, если он использует OPENWallet.

11. Теперь пользователь Punia может просматривать видео или войти в приложение Чейза, потому что он является абонентом.

Примечание: пользователь Punia может войти в приложение Чейза с помощью традиционных способов, потому что его OPENWallet представляет запросы на изменение состояния, а не идентичность. Один OPENWallet может иметь возможность для обновления нескольких различных пользователей, отправляя различные хеширования состояния.

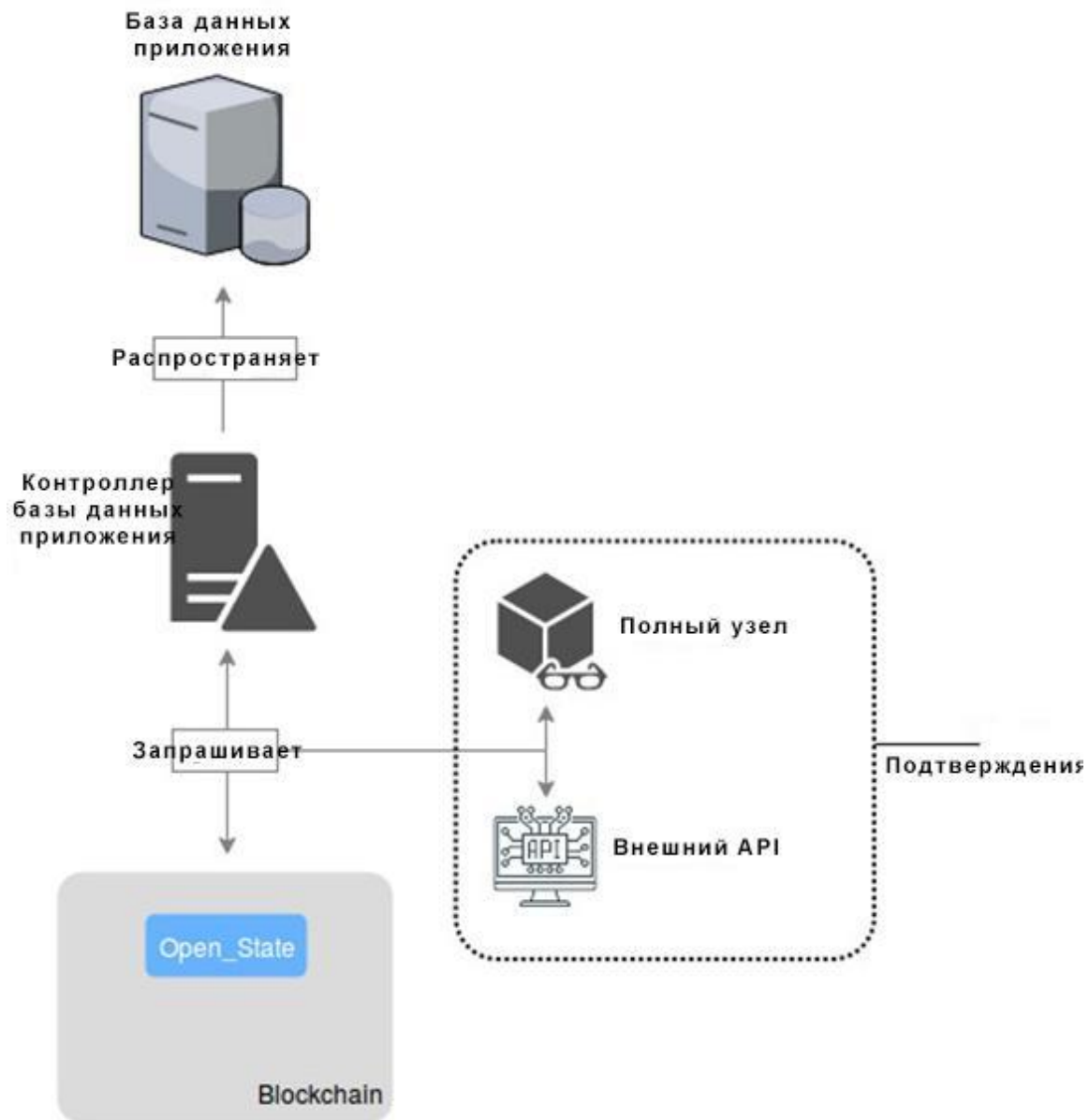
Вопросы безопасности

Платформа OPEN включает в себя различные компоненты вне цепочек, которые могут создавать проблемы безопасности и реализации. Этот раздел посвящен наиболее актуальным проблемам и включает в себя варианты их решения..

Синхронность базы данных

Потенциальный вызовом для OPEN является трудность определения того, находится ли база данных в актуальном состоянии с децентрализованным балансом OPEN_States (которые можно просмотреть на блокчейне Ethereum). Чтобы решить эту проблему, OPEN может заставить блокчейн записывать каждый вызов API в базу данных и сохранять его в OPEN_State, но это бы поставило ограничение на число «безвредных» API вызовов (например, получение имени пользователя или изменение адреса электронной почты), которые приложения должны выполнять на ежедневно. Таким образом, невозможно сохранить базу данных и блокчейн в полностью синхронизированном состоянии, и поддерживать реалистичные решения экономической эффективности.

Вместо этого, OPEN выбрал частично синхронную базу данных. Эта частичная синхронность использует информацию о том, что каждая база данных уже разрешила вызовы API и внесение изменений, которые проходят через платежный шлюз. Таким образом, необходимо только следить за количеством платежей и состояний или действий, которые разрешены пользователям.



Эта схема показывает, как база данных приложения, узнает конкретного пользователя на цепочке OPEN_State. Как правило, OPEN API можно реализовать на контроллере приложения, чтобы проверить Scaffold для любых новых OPEN_States, а затем распространить эти изменения. База данных приложения затем отслеживает блокчейн для OPEN_States, возникших из соответствующего Scaffold. При обнаружении, база данных распространяет изменение состояния, представленное OPEN_State. Проверяющий элемент может одним из множества реализаций, но он должен проверить, что OPEN_State действительно создан конкретным Scaffold разработчика и что этот OPEN_State является самым последним OPEN_State, который был создан. Если OPEN_State подтверждается, то контроллер приложения вносит изменение в базу данных.

Примечание: в случае операции, когда пользователь пытается приобрести внутреннюю валюту в приложении, лучше использовать операцию действия вместо операции состояния, чтобы злоумышленники не смогли использовать внутреннюю валюту в приложении до проверки сделки. Эти действия объясняются более подробно ниже.

Эта частичная синхронность базы данных во многом зависит от последовательности и скорости метода проверки. Методику проверки можно осуществить одним из нескольких способов: с помощью внешнего API, который проверяет блокчейн Ethereum для OPEN_State, что может представлять серьезную угрозу безопасности и затратам времени, или с помощью полного узла, что потребует от разработчика развернуть полный узел на сети Ethereum.

Поскольку оба этих варианта создают проблемы с безопасностью или с вычислительными затратами, для OPEN имеет смысл завести свой собственный API для опроса блокчейна и проверки наличия конкретного OPEN_State.

Несколько OPEN_States

Другой потенциальной проблемой для OPEN является то, что OPENWallet может и должен иметь несколько OPEN_States для одной и той же базы данных, поскольку злоумышленник может запросить внесение предыдущего OPEN_State в базу данных, и база данных может принять его без приема платежа, поскольку OPEN_State существует на блокчейне. Поэтому необходима возможность проверить последний OPEN_State пользователя в базе данных. Хотя несложно определить, какой OPEN_State является новейшим, проверив блокчейн, это занимает много времени и позволяет злоумышленникам атаковать базу данных, запрашивая несколько предыдущих OPEN_States и заставляя базу данных выполнять по ним проверки, чтобы подтвердить, что они являются последними.

Вместо этого к пользователю применяется такой тип реализации, который позволяет следить за всеми его состояниями через кошелек. Эта история может быть частью открытого API, в базе данных, или частью кошелька разработчика, и будет структурирована как хэш-цепь, где каждое новое состояние зависит от хэша предыдущего состояния. Эта история позволяет быстро проверять OPEN_States пользователей, и обеспечивает безопасность от атак DDoS.

Реализация истории множественных OPEN_States



Состояние vs действие в OPEN_State

Хотя это не является формальным вопросом безопасности для платформы OPEN, следует отметить, что OPEN_State может содержать действия и состояния в своей схеме. Действие отличается от состояния тем, что действие обеспечивает манипуляции текущего состояния, в то время как состояние будет обеспечивать само состояние. Схема действия и состояния в OPEN_State должна быть реализована по разным причинам. Ниже представлен набор общих принципов для применения действия или состояния, а также потенциальные риски безопасности для обоих методов.

Состояния

Состояние должно быть зарезервировано для приложений на основе разрешения, в которых пользователь не сможет использовать систему путем изменения состояния, прежде чем ему предоставят новое состояние.

Примером грамотного применения может быть изменение состояния пользователя с подписчика на не-подписчика, или набор конечных состояний, которые пользователь может приобрести и не может потратить в промежутке времени до завершения проверки.

Тем не менее, в некоторых случаях состояние может быть уязвимым. Примером такой уязвимости будет пользователь, который имеет 500 единиц валюты приложения, допустим, драгоценных камней, и запрашивает состояние 550 драгоценных камней, и до того, как база данных вносит изменения, пользователь тратит все свои 500 драгоценных камней. Поскольку сделка действительна, и база данных должна подтвердить, что Open_State является корректным, а не историческим, пользователю удастся украсть 500 драгоценных камней из приложения через систему. Чтобы предотвратить это, необходимо использовать действие.

Действия

Действие должно быть зарезервировано для приложений, где пользователь хочет добавить определенное количество внутриигровых денег, или существует множество игровых состояний, которые может иметь пользователь, и он может потратить или обменять их до того момента, когда база данных распространит изменения.

Потенциальный риск использования состояний дает большие возможности использования действий. При действии, пользователь должен указать, что он хочет добавить 50 драгоценных камней на свой счет. После того, как он оплатит сделку, он получит возможность запустить эту функцию определенное число раз (в данном случае один). Независимо от того, какую сумму он тратит он не сможет украсть драгоценные камни. Действие будет размещено в истории базы данных после того, как оно запустится, что означает, что не получится использовать запуск действия больше, чем отведенное количество раз.

Действие может представлять опасность в виде запуска плохих действий для разработчиков, поэтому разработчик должен вносить конкретные ограничения по аргументам. Состояние лишено этой проблемы, поскольку оно кодирует некоторый набор действий, которые разработчик уже запрограммировал.

Будущая функциональность OPEN

Инфраструктура в экосистеме блокчейн постоянно меняется и адаптируется. Для того, чтобы платформа OPEN не только была частью будущего блокчейн, но и возглавила это будущее, OPEN разрабатывали с учетом роста по мере эволюции технологии вокруг блокчейн. В качестве платформы, на которую обычные разработчики приложений могут положиться при использовании блокчейна и криптовалют в своих приложениях, OPEN необходимо постоянно оставаться актуальной.

Первое решение, которое в OPEN приняли с оглядкой на будущее - построение инфраструктуры на Ethereum и решение вопросов масштабируемости и операционных издержек путем портирования системы в Tendermint на более позднем этапе. Поскольку OPEN написан в Solidity (и в конечном итоге Viper) и работает на виртуальной машине Ethereum (EVM), можно перейти с сети Ethereum на EVM Tendermint, известной как Ethermint (через ABCI). Это не только позволит OPEN воспользоваться более высокой пропускной способностью и более низкими операционными издержками, предоставляемыми алгоритмом консенсуса Tendermint, при переходе не будет трений, поскольку Ethermint работает на том же EVM. OPEN даже можно будет портировать обратно на Ethereum, когда он реализует более быстрый алгоритм консенсуса Proof-of-Stake. В соответствии с проектом, платформа OPEN является переносной между многочисленными системами и не ограничивается одним протоколом блокчейн. В будущем рост OPEN может привести нас к желанию развивать собственную цепочку для повышения эффективности, особенно по вопросам Scaffold и сделок.

Модульность OPEN дает возможность разработчикам не только с легкостью включать новые системы вне цепи, но и позволяет сообществу OPEN обеспечивать обновления безопасности и дополнения к платформе, тем самым способствуя инновации на блокчейне и за его пределами.

Согласованные усилия были приложены для того, чтобы разработчики и владельцы Scaffolds не были ограничены экосистемой OPEN при интеграции с существующей инфраструктурой приложений. Обеспечена значительная гибкость для разработчиков по выбору типа структуры базы данных, кошельков, методов проверки. В то время как OPEN обеспечивает кошельки разработчиков и механизмы подтверждения, платформа специализируется на создании взаимосвязанного прикладного уровня, действующего в качестве клея между различными блокчейн компонентами. Например, если разработчик хочет использовать децентрализованные базы данных, нет ничего проще, чем включить эти виды взаимодействия баз данных в OPEN.

Платформа OPEN рассчитана на то, чтобы ее можно было легко изменять и обновлять, с целью включения новых возможностей для новых систем, а ее модульная конструкция дает большую степень свободы, предоставляя код открытому сообществу. Можно с легкостью внести такие дополнения в фреймворк OPEN, как упрощение учета Scaffold или включение проверки кошелька для повышения безопасности в системе.

Следующие шаги

OPEN поделится открытым исходным кодом всех инструментов этой платформы и реализует токен утилиты для поддержки проекта платформы OPEN. Платформа OPEN развивает магистральную и будущую инфраструктуру, необходимую разработчикам приложений для увеличения областей применения криптовалют. Продвигая платформу вперед, OPEN продвигает применение блокчейна в массы, предоставляя необходимые инструменты для приложений и создания инструментов с открытым кодом для сообщества.

В рамках этого проекта было затрачено огромное количество труда на улучшение уровня протокола при поддержке уровня приложений, что позволит использовать эту инфраструктуру приложениям и их разработчикам