PANTERCON©

Реализуй свою идею вместе с Pantercon

Aubergine Paper

Social Media



Содержание Технической Документации Блокчейна Hydra

1. Плоскости гибридного блокчейна Hydra

1 Общая информация

Для гарантии высокой производительности и низкой емкости, блокчейн Hydra состоит из нескольких слоев.В блокчейне HYDRA используются функция MapReduce, когда в обход дерева Меркла процессы проверки принудительно распараллеливаются в некоторое количество childchains (дочерние цепи).

Использование sidechains (childchains), в mainchain, значительно увеличивает масштабируемость.

1.2 Mainchain(основная цепь)

Это классический основной вариант и сердце блокчейн системы. С помощью Hydra, mainchain(основная цепь) будет состоять из множества различных уровней, которые описаны ниже.

1.3 Sidechains(боковые цепи)

Боковая цепь - это самостоятельный блокчейн, который присоединен к его родительскому блокчейну с помощью" двухсторонней привязки". Эта «двухсторонняя привязка» (Zerberus) регулирует взаимозаменяемость объектов между вышестоящими блокчейнами и сэйдчейнами. Для этого определена фиксированная ставка. Также возможно прикрепить несколько сэйдчейнов к сейдчейну.Таким методом можно сопоставлять целые иерархии.

Этот иерархический пример, можно назвать «Родительская» цепь и подчиненные ей "Дочерние" цепи.

Техническая процедура заключается в следующем: пользователям необходимо всегда отправлять объекты на определенный выходной адрес высшей цепи. Там объекты блокируются, таким образом, что их больше не возможно использовать либо дублировать в данной цепи. После проверки транзакции, объекты освобождаются в сэдчейн(боковой цепь), где и смогут быть использованы пользователями. Обратный процесс, это когда объект возвращается из боковой цепи в главную.

1.4 Распараллеливание процессов верификации.
Если текущее количество транзакций для проверки превышает указанное значение, новый процесс проверки автоматически запускается параллельно. Этот процесс может повторяться несколько раз и будет ограничен определенным количеством параллельных процессов. В зависимости от используемого алгоритма консенсуса и количества точек проверки, при прямом сравнении может быть большое отклонение в производительности.

Во время преобразования, посредством тестов определяется количество транзакций, запускающих новый параллельный процесс, и ограничивающей его. В сэйдчейн (боковых цепей) техническое преобразование деревьев Меркле происходит принудительно.

Поэтому, скорость транзакции 50 000 операций в секунду должна бить достигнута изначально.
На последнем этапе расширения Hydra, эта цель вполне осуществима.

1.5 ECHIDNA (базовые данные объектов Hydra)
Содержит базовые данные всех дочерних и боковых цепей, а также главной цепи. В ECHIDNA проверяются только основные данные цепей и их объектов. Этот уровень используется исключительно для функциональности всех других цепей и никаких классических процессов здесь не происходит. Управление всей комиссией также осуществляется в ECHIDNA.

В принципе в sidechains(боковых цепях), есть возможность взимать дополнительные либо другие комиссии, которые затем передаются назначенной организации. Это зависит от алгоритма консенсуса и от того, являются ли боковые цепи «общедоступными» или «частными».

Чтобы получить упрощение для конечного пользователя, комиссии за транзакции в основной цепи делятся на 3 уровня приоритета (высокий, средний и низкий). Высокий приоритет связан с высокими тарифами, но гарантирует немедленный перевод и т. д.

На заключительном этапе пользователи также смогут сами создавать новые комиссии.

1.6 NYX (анонимные транзакции)

Под сегмент NYX предназначен для облегчения анонимных транзакций.

Тем не менее все еще существует необходимость в юридическом разъяснении.

1.7 PLUTOS (транзакции B2B)

Здесь хранятся B2B транзакции. Подобно другим публичным цепям, теоретически эти адреса можно просматривать. Благодаря авторизации пользователя владельцы кошелька смогут передавать контролируемые по времени права в другое место. Пример: компания предоставляет права на чтение учетной записи для бухгалтера налогового консультанта с ограничением по времени (пример: только последние 3 месяца).

1.8 HERMES (торговая площадка HYDRA)

Здесь задокументированы сделки с предметами товаров В этой области основное внимание, с одной стороны, уделяется отслеживаемости движения товаров, а с другой - предоставлению данных. Особое внимание здесь уделяется следующим моментам:

- **Серийные номера**
- **Партии**
- **Токены логистики**

В области предоставления данных целью является предоставление данных о процессах B2B для других пользователей. Возможные варианты использования:

поставщики предоставляют товары со всеми атрибутами (спецификация размеров, текстовые описания, наличие, а также цены).Посредством присвоение прав можно сделать их доступными только для заранее определенных партнеров. После они, могут работать с этими данными и заполнять свои системы ERP (планирование ресурсов предприятия).

1.9 DEMETER (права собственности на коммерческие товары)

Эта область предлагает очень высокий потенциал вариантов использования.Права собственности автоматически присваивается тому пользователю которому они передаются. Они могут быть окончательными или бить подчинены ограничениям по времени.

1.10 HERA (права на chains(цепи), смарт контракты)

1.10.1 Права в целом

В этой области задокументированы права отдельных пользователей, их роли и, следовательно, разрешения для соответствующих цепей. Система авторизации будет обсуждена во всех деталях. Наиболее важные области здесь - это то, что два типа контроля времени связаны с правами.

- Само право действует только в течение определенного периода времени (например, замена отпуска, тогда для сотрудника создается право на 14 дней).
- Право ограниченное по времени (например, работник может просматривать только транзакции текущего квартала).

1.10.2 Передача прав

Права распределяются по двухэтапному принципу. Права могут быть созданы и автоматически сохранены. Тем не менее они становятся активными только после проверки и предусматривают плату.

1.11 HADES (архив)

Область HADES используется для архивирования данных. Метод Phoenix используется для перемещения данных из главной цепи в HADESChain. На заключительном этапе разработки такая функция также может быть использована для боковой цепи. Подробное разъяснение о методе Phoenix смотрите ниже.

1.12 ARCHIMEDES (объекты HYDRA)

Здесь хранятся детали объектов Hydra. В общем, термин объект используется для всех видов токенов,приложений и т.д.

Кроме этого, здесь есть возможность генерировать совершенно новые типы объектов, которые не сопоставимы с «классическими» токенами. Все в пределах воображения.

- Пример: To-Do Lists (список задач) для управления проектом
- Управление границами собственности
- Счетчик калорий для худеющих и т.д

1.13 Тестовая сеть

Тестовая сеть обладает всеми возможностями продуктивной системы. Они отличаются тем что здесь каждый квартал удаляются не смарт контракты, а данные транзакций (транзакции). Кроме того, KRONOS позволяет имитировать время. Это означает, что контролируемые временем контракты могут быть протестированы в 10 раз быстрее. К примеру Это будет означать, что контракт, в настоящем действующий 30 дней, будет действовать в течение трех. KRONOS обладает не только способностью ускорять время, но и останавливаться в определенных точках или сознательно переходить в определенные моменты времени для начала тестирования с этой точки.

1.14 Токены

Предусматривается разные виды токенов которые имею определенные свойства. Такие классические виды как токен безопасности, служебный токен, членский токен или токен доступа. Для каждого токена любого вида будет возможно свободно программировать лю любые необходимые функции.

2. Регулируемая по времени концепция авторизации пользователя основанная на ролях

2.1 Общая информация

Эта концепция сочетает в себе классическую авторизацию пользователей на основе ролей с контролируемым по времени назначением прав и контролируемым по времени доступом к данным транзакции

2.2 Концепция компонентов

Пользователю можно назначить одну или несколько ролей (администратор, клерк и т. д.). Эти роли имеют особые права. Пользователь может выполнить действие, если ему или ей было предоставлено право на основании одного (или более) из его или ее ролей. Пользователь может выполнить действие, если ему или ей было назначено право на основании одной (или более) из его или ее ролей. Здесь будет возможным то, что эти роли могут иметь ограничение по времени. Пример: передача роли другому сотруднику для замены отпуска.

На заключительном этапе разработки станет возможной дальнейшая индивидуализация прав на уровне пользователя независимо от ролей.

2.2.1 Организации

Объявлять Организации лучше всего юридическими лицами, компаниями, НПО и т. д.

Они могут запускать боковые цепи или работать с основной цепью. Для созданных вами объектов (токенов, боковых цепей и т. д.) У вас есть возможность назначать полномочия.

2.2.2 Организационные подразделения

Это подразделения, которые как в классической организационной структуре составляют организацию и представляют отделы. Эти организационные подразделения могут быть ролями с подписью, которые передаются пользователям.

2.2.3 Авторизация пользователя

Для перехода на различные уровни пользователь должен войти в основную сеть(Mainnet). После входа будут загружены разно уровневые разрешения, которые позволят осуществить навигацию в отдельных областях.

2.2.4 Права

Права определяются освобождение или предотвращение действия. Пример: чтение данных транзакции, создание объектов.

2.2.5 Роли

Роли определяются как функции лиц, которые должны иметь несколько прав для их осуществления. Эти роли могут быть, например: администратор, клерк и т. д.

2.2.6 Права ролей

Пользователям с определенными ролями присваиваются определенные права по ролям. Например, администратор может прочитать «Продажи и сделки» за последний квартал, но не может делать какие-либо банковские переводы.

2.2.7 Роли пользователя

Отдельным пользователям назначается одна или несколько ролей.В случае назначения двух ролей одному пользователю есть возможность что права от наданых ролей будут противоречить друг другу.Пример.

Пользователю назначена роль «продавец-консультант», и поэтому ему запрещается осуществлять переводы. Тем не менее он также назначен на роль «Управление». Эта роль действительно имеет право осуществлять переводы, поэтому ему предоставляется это право.

2.3 Ограничение по времени

2.3.1 Временные права

Как уже упоминалось в начале, вы можете временно назначать права. Они могут быть назначены на определенный период времени или с датой истечения срока действия. В окончательной версии будет возможно управление этими событиями через смарт контракт. Пример: после получения платежа право продлевается на один месяц.

2.3.2 Доступ к данным с ограниченным периодом времени

Другим важным моментом является возможность предоставления данных транзакций с ограничениями. Это может быть статический период или динамический период.

Таким образом, вполне возможно, что некоторые пользователи могут видеть только транзакции с начала года, а другие всегда только те транзакции, которые не старше 3 месяцев.

3. Алгоритм консенсуса

В основной цепи будет использоваться алгоритм аналогичный алгоритму Proof-of-importance (PoI) (алгоритм доказательства важности). Значение важности присваивается каждой отдельной точке, которая состоит из различных факторов, таких как количество монет PANX, время удержания, высота и количество транзакций и т. д. Этот метод помогает убедиться в совпадении всех компьютеров в сети. Пользователи с высокой «значительностью» могут «собирать урожай» и получать вознаграждение.

В блокчейне Hydra с помощью этого процесса можно собирать HXP, которые могут быть проданы либо использованы для различных немало важных действий. Для самостоятельного создания объектов Hydra могут использоваться различные алгоритмы консенсуса особенно в сэдчейнах(боковых цепях). На заключительном этапе будут предоставлены наиболее распространенные алгоритмы, а также возможность программировать собственные методы консенсуса.

4. HERAKLES

Herakles - это графический интерфейс для создания боковых / дочерних цепей для управления правами и объектами, который формирует основную часть для всей системы управления.
Особое внимание уделяется для интуитивного и удобного использования HERAKLES. Дальше больше, будут созданы интерфейсы для взаимодействия с отдельными областями блокчейна / смарт контракта.

5. PHOENIX

С помощью Phoenix можно сжигать и одновременное возрождать токены не только внутри цепи. Эту функцию можно использовать для действий с токенами и внутри блокчейна HYDRA. Простое сравнение это закрытия и открытия баланса. Балансы для всех зарезервированных учетных записей создаются и переносятся в новый период проводки. Обычно это происходит в конце года, но в блокчейне HYDRA может происходить в свободно выбранном периоде.

Здесь тип балансирования происходит в протоколе Phoenix. Главная цепь становится архивной цепью Hades, и создается новая основная цепь, в которую из протокола Phoenix передаются сбалансированные значения. Переданные транзакции не теряются и могут быть просмотрены через ArchiveChain (архивную цепь) Hades.

Для пользователя это будет выглядеть так, как будто бы эти данные все еще находятся в главной цепи.

С помощью этой методологии можно бесконечно создавать новые архивные цепи не нагружая блокчейн. Теоретически тем же методом можно и уничтожать архивы.

Поскольку эта методология применяется и в боковых цепях, здесь в зависимости от выполняемых функций с хранимыми данными могут присутствовать связи с пунктами **DSGVO**. Для удовлетворения правовых аспектов применяется метод Phoenix.

DSGVO(*Основополагающий регламент Европейского Союза о защите персональных данных*)

6. Действительный умные контракты

6.1 Общие сведения

На языке программирования для создания действительного умного контракта используется C#(объектно-ориентированный язык программирования). Суть заключается в том, чтобы создать контракты настолько „разумными“, что бы они работая в автоматическом режиме могли выполнять все важные компоненты обычного контракта: условия, сроки уведомления и т. д.

Однако, в зависимости от типа контракта при его создании предусматриваются функции «отмена» или «откат». Функция «отмена» отменяет договор и переопределяет его, например: отмена постоянного распоряжение на зарплату работника, в случае его увольнения. Функция «откат» отменяет контракт и выполняет установленные функции отката, например: в случае не достижения проектом софткапа инвесторам возвращаются их средства.

В договорах функции «отмена» или «откат»определяют отдельные процессы, например, при отмене поставщиком договора он оплачивает клиенту неустойку размер которой заранее определен тем же договором. Идеальным решением для этих примеров являются количественный и call-off (рамочный) контракты. Так же эти функции могут быть использованы в других областях,например уплата комиссионных внешними компаниями, договоры с конечными потребителями, а также консигнационные контракты.

Контракты должны иметь период действия.
Ответственность за активацию контракта несет создатель заказа, но при согласии эти полномочия можно назначить и другим партнерам по контракту.

6.2 Автоматизм

При создании смарт контрактов особое внимание уделяется автоматизму. Для этого рассматриваются два основных элемента: контролируемые событием и временем функции. Вышеупомянутое событие определяется как получение транзакции, которая затем запускает всю цепь процессов. Контролируемое временем это когда каждый день в полночь автоматически из филиала в главный офис будет переводиться определенная сума.

6.3 Стандартизация через модуляризацию

смарт контрактов
Мы предоставляем стандартизированные модули, которые могут быть объединены пользователем. Каждый модуль предоставляет входные и выходные параметры и имеет способность цикличного выполнения. В качестве примера модуль можно назвать «ценообразование контролируемое временем». Таким образом, можно разместить таблицу, которая дает новую цену на каждый день.
Комбинируя модули, можно также отобразить цепь процессов, управляемых событиями.

6.4 Настройки модулей

Здесь можно будет писать и хранить код для специальных приложений, вместо самого кода модуля. Также будет присутствовать возможность ввода определенного пользователем кода вместо предлагаемого.Таким образом, модульное ценообразование управляемое временем может быть перепрограммировано таким образом, что после первых 100 продаж в день цена автоматически будет увеличиваться на 10%. Для этого должен быть сохранен новый входной параметр для количества ежедневных продаж.

Как правило, самописный код временно сохраняется в базе данных. Код записывается только в блокчейн и становится действительным, после активации смарт контракта

6.5 Взаимодействие с контрактами во время их выполнения

Перед активацией контракта должны быть назначены полномочия и определены параметры, разрешающие взаимодействие.Примером этого является продление контракта. В зависимости от разрешения владелец контракта может принудительно продлить его самостоятельно, либо с помощью полученных разрешений от партнеров по контракту. Для взаимодействия со смарт контрактами здесь будет использоваться графический интерфейс. В зависимости от разрешения пользователи смогут получать доступ к функциям, предоставляемым владельцем смарт контракта.

Также будет присутствовать интерфейс для пакетной обработки данных.С помощью которого владелец контракта сможет загружать в него данные. Пример: несколько адресов, которые должны быть помещены в белый список, или платежи по нескольким адресам с разными суммами (например, программы Air-drop или Bounty).

6.6 Прекращение в случае несоблюдения

При несоблюдении условий контракта в силу вступает событие расторжения. Например: не была произведена оплата. Таким образом для потерпевшей стороны освобождается сценарий «выход». В большинстве таких случаев применяется функция «Прервать», но может и быть использована функция «Откат», если она определена типом контракта. Потерпевшая сторона должна выполнить эти функции и активировать их вручную.

7. «Проблема времени»

Для обеспечения единого времени при создании блока как правило используется время UNIX. Параллельно, время проверенных точек сохраняется для всех блоков (редактор времени). Проверка центральной единицей времени здесь проходит заранее . Также проверяется на отклонение время редактора и, если оно не превышает 13 секунд, действие выполняется.

8. Merkle Tree - Дерево Меркла (Хеш-дерево)/ Merkle-Proof(алгоритм доказательства Меркла)

Проверка выполняется с помощью Дерева Меркла.
Дерево Меркла являет собой блоки с данными, основаной на древовидной структуры. Каждая ветвь содержит только несколько блоков, которые объединяются, и ведут к другой ветке. Теперь каждая из этих ветвей проходит один и тот же процесс, который повторяется до тех пор, пока общее количество оставшихся хешей не станет единственным значением: «корневой хеш».
Алгоритм Доказательство Меркла состоит из «корневого хеша» дерева и «хеша», который состоит из всех хешей, идущих от ветви к корню.
Таким образом, для этой ветви можно проверить, было ли хеширование последовательным до конца дерева и действительно ли положение ветви в дереве находится в этой точке дерева.
Простое применение: существует большая база данных, и все ее содержание хранится в дереве Merkle, "корень" которого общедоступен и заслуживает доверия (например, он был подписан цифровой подписью достаточно надежными сторонами). Пользователь может запросить доказательство Merkle и после его получения проверить, является ли оно правильным.
В ходе реализации используются не только двоичные деревья Меркла, но и более сложный вариант, аналогичный принципу дерева Патрисии.