Hub
Протокол доверия

Eric Ly, Rich Miller, Miko Matsumura
Переведено Finforge.vc

"*Валюта новой экономики – это доверие*", - Rachel Botsman [1].

Аннотация

Применения интернета предоставили миллиардам пользователей беспрецедентную возможность для общения, создания контента и торговли благодаря социальными сетям, мессенджерам, онлайн-вычислениям и p2p-маркетплейсам, дающим доступ к экономике совместного пользования. Пользователи получили возможность общаться в виртуальном пространстве, но на пути генерации большей экономической выгоды в этой среде стоит недостаток доверия между он-лайн контрагентами.

Появление технологии блокчейна и криптовалют породило «интернет ценности» [2]- протокол передачи не только информации, но и единиц экономической ценности. Эти технологии позволяют поддерживать в децентрализованной сети консенсус при передаче внутри сети токенов, стимулирующих пользователей совершать некоторые действия на благо всей сети. Защищая сеть от атак и поощряя активность, технология блокчейн обеспечивает появление сетевого эффекта, результатом чего становится быстрый рост подобных сетей.

Блокчейн впервые в истории человечества позволяет решить проблему недостатка доверия между контрагентами путем создания системы оценки благонадежности пользователей в различных приложениях. Проект Hub создает протокол доверия и дает доступ к проверяемой и транслируемой оценке степени благонадежности, которая может быть использована в самых различных интернет-проектах для обеспечения «доверия на расстоянии» при контакте с незнакомыми пользователями в сети.

1. Внутренний токен Hub стимулирует пользователей выполнять специализированные задания на платформе и, соответственно, генерировать данные, на основе которых будет оцениваться их репутация, а также повышает доверие пользователей друг к другу.

2. В задачах, предполагающих перераспределение внесенных средств в зависимости от результата исполнения задачи, в качестве залога можно использовать "доверительный залог" в форме токенов данного проекта.

3. Результаты исполнения заданий представляют собой историю изменения репутации пользователя и хранятся в неизменном виде в публичном блокчейне; на основе этих данных для сторонних приложений предоставляется защищенный способ оценки уровня доверия пользователей и репутации.

4. Аналог App Store (Хранилище Задач) и система вознаграждений для разработчиков новых Задач позволяет создать открытую экостистему, которая

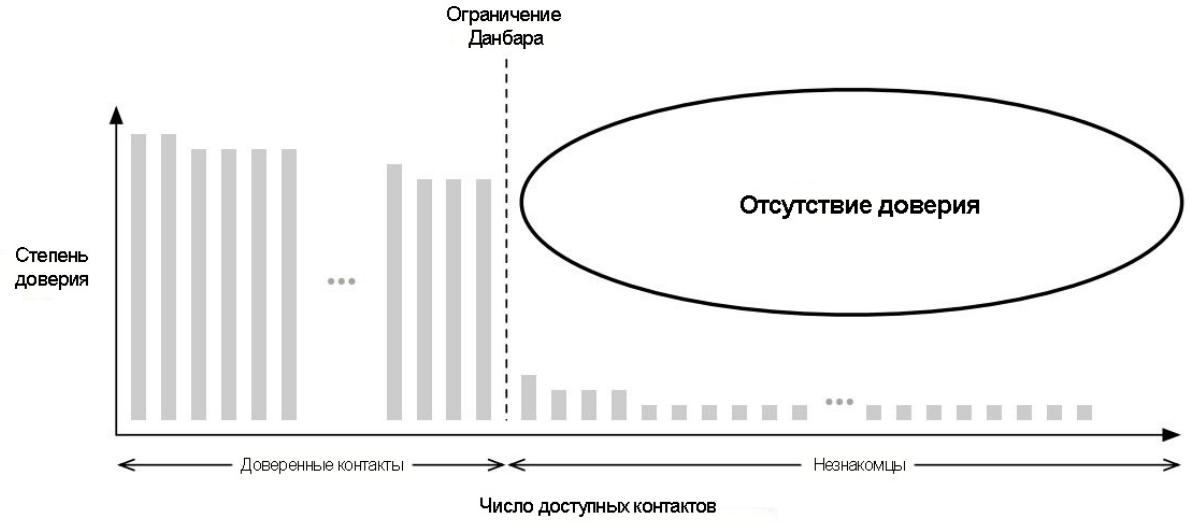
покроет большинство сценариев взаимодействия пользователей, невозможных без доверия

1. **Существующие проблемы**

Быстрая и масштабная консолидация централизованных маркетплейсов и платформ для социальных взаимодействий привела к нескольким проблемам, связанным с наличием доверия у незнакомцев, участвующих в сделках и транзакциях. Ежедневно производятся сотни миллиардов операций, в которых участвуют незнакомые пользователи. Если считать в транзакциях, к 2020 году их количество в интернете достигнет 450 миллиардов в день [3]. Из-за того, что пользователи не могут полноценно доверять друг другу, эти транзакции влекут за собой упущенные возможности, высокие транзакционные издержки и накладные расходы, повышенный риск плохих исходов, так называемое «трение» в обществе. Чем больше люди будут взаимодействовать друг с другом в интернете, тем хуже будет ситуация с недостатком доверия между ними.

1.1 Число Данбара и «доверие на расстоянии».

Удобным способом понять суть проблемы является показатель, известный как число Данбара, который описывает когнитивное ограничение на количество устойчивых отношений между людьми [4]. Антрополог Робин Данбар [5] предложил на основе проведенных наблюдений теорию, согласно которой средний человек может одновременно поддерживать до 150 социальных связей. Это значение известно как число Данбара. Такое количество социальных связей соответствует тем людям, которым данный человек может доверять и в кругу которых происходят наиболее ценные (и иногда имеющие экономическое значение) взаимодействия. Социальные приложения позволяют благодаря интернету увеличить количество потенциально доступных контактов до миллиардов человек, но так не обеспечивается доверие между людьми, при таком взаимодействии пользователь предполагает наличие риска.



Изобретатель смарт-контрактов Ник Сабо обозначил ценность биткоина как форму передачи ценности на расстоянии. Однако, в децентрализованном интернете ценности появляется проблема взаимного доверия между пользователями на

расстоянии, причем расстояние между ними измеряется не географически, а в единицах доверия. Для того, чтобы сделать полноценную децентрализованную систему, нужно обеспечить возможность транслировать доверие на расстоянии.

1.2 Плохая информация, слишком много информации.

Многие взаимодействия между людьми в интернете начинаются с того, что кто-то делится с другими информацией. На ее основе часто принимаются неточные решения, потому что умышленно искаженный контент из предвзятых источников повышает вероятность ошибки. Посмотрим, к примеру, на то, какое влияние социальные сети оказывают на политические движения, и какую роль они играют в электоральном процессе. Система для социальных взаимодействий полна участников, пользующихся благами системы, но не вносящих вклад в ее улучшение, включая спамеров, которые распространяют информацию низкого качества в форме скам-проектов, фейковых новостей и объявлений. Электронная почта – крупнейшая платформа для обмена сообщениями, ей пользуются 3,7 млрд человек [7], в день отправляют 226 млрд писем [8], но именно здесь можно наблюдать отсутствие хоть какого-либо достоверного механизма обеспечения доверия между пользователями. В результате половина отправляемых сегодня сообщений – это спам, поток нежелательных писем с какими-либо просьбами, которые отправляют только потому, что их дешево создать и отправить. Спам сам по себе создал экосистему из приложений по автоматизации отсева писем и борьбы со спамом. Доля рынка технологий, посвященных обнаружению угроз, защите пользователей от кражи персональных данных и вредоносных писем достаточно велика и представляет собой своеобразный налог, который нужно платить, чтобы иметь доступ к предлагаемым интернет-средой ценностям.

1.3 Риск совершения транзакций

Для совершения транзакций еще важнее доверять партнеру по сделке и иметь доступ к правильной информации. С ростом спроса на совершение транзакций онлайн наибольший риск оказывается связан с существующими системами, которые были успешны в прошлом и которым продолжают доверять пользователи. Человек предполагает, что партнеру по сделке можно доверять, и проводит транзакцию вне платформы или оффчейн, то есть по телефону или при личной встрече, что связано с высоким риском и накладными расходами.

1.4 Централизованные владельцы, частичные данные, недостаток мобильности данных

Даже при наличии некоторой системы оценки репутации, например, на p2p маркетплейсе, исходными данными владеет компания, которая создала данный сервис, что не позволяет получить пользу от сетевых эффектов или внешних партнеров. Обзоры на Yelp, посты и комментарии на Quora, Reddit, StackOverflow не так просто получить, не являясь пользователем соответствующей платформы. Некоторые наиболее успешные приложения для шеринговой экономики работают именно на этих площадках, которые наименее вероятно станут шэрить (делиться) собранными данными [10]. С ростом популярности подобных площадок в течение последнего десятилетия владельцы приложений, по сути, стали распоряжаться репутационными данными пользователей. В результате владение и возможность распоряжаться данными по большей части перешла от пользователя к

централизованному хранилищу. Еще одно следствие сложившейся ситуации в том, что данные о репутации пользователя в сети оказались фрагментированы и разбросаны по разным платформам; зачастую для того, чтобы получить комплексные данные о репутации незнакомого человека, нужно собирать по кускам как паззл историю его активности в интернете. Кроме того, даже если на какой-то одной площадке пользователь приложил усилия к формированию своего подробного профиля, который служит инструментом для оценки доверия, зачастую нет механизма переноса этой информации на другую площадку. Присоединение к новому сообществу означает необходимость все начинать сначала.

1.5 Выводы из работы с LinkedIn

Мы со-основали LinkedIn [11], самую большую в мире профессиональную социальную сеть с более чем 500 миллионами пользователями, и наш опыт показывает, что несмотря на относительно тесные доверительные отношения пользователей в социальной сети, этого недостаточно для того, чтобы непосредственно на площадке стали возможны многие виды взаимодействия. Примерами причин тому служат фальсификация профессионального профиля, указание не соответствующих действительности должностей, времени трудоустройства, дипломов об образовании и прочего [12], [13], [14]. Некоторые пользователи могут злоупотреблять доступом к сообществу и рассылать спам-сообщения или размещать в группах нежелательную рекламу. Глядя на историю подобных случаев мы все еще не сумели спроектировать набор стимулов, который бы приводил к полному отсутствию злонамеренных действий пользователей. Эти недостатки снижают доверие профессионалов друг к другу и наносят ущерб самому бизнесу, снижая эффективность таргетированной рекламы и возможности подобрать подходящую вакансию. Каждой из социальный сетей присущи те же проблемы в том или ином виде.

2. Принципы доверия

При описании протокола мы будем говорить о доверии и тесно связанных с ним понятиях репутации и идентификации пользователей. Прежде чем говорить о доверии, нужно ввести основополагающие понятия репутации и идентификации, так как они уже тщательно изучены, в ранее созданных системах с ними уже пытались работать.

2.1 Репутация

Под репутацией пользователя мы будем понимать определенное сообществом знание о поведении пользователя в прошлом. Репутационные данные – представление этого знания в цифровой форме. На основе исторических данных один пользователь может прогнозировать будущее поведение другого. В [16] приведен полезный обзор работ на тему репутационных систем в p2p маркетплейсах; на сегодняшний момент некоторые из описанных в обзоре подходов могут уже быть реализованы. Во многих случаях площадки используют сжатие данных о пользователе до краткого набора стандартных показателей, которые легко интерпретировать как репутацию. Хорошо известные примеры – это кредитный рейтинг FICO [17] и Sesame Credit [18].

В данном протоколе для оценки репутации используется большой набор показателей и контекстные данные. В такой постановке уже невозможно ввести универсальный «репутационный скоринг» или привести единое определение репутации пользователя. Например, спортсмен может считаться очень успешным, потому что выиграл бесчисленное количество футбольных матчей, но он же будет плохим теннисистом даже несмотря на свою хорошую физическую форму и навыки командной игры. В данном протоколе репутационные данные пользователя определяются как неизменная история всех актуальных действий пользователя. ***Репутация*** формируется на основе выполненных пользователем ***задач***.

Результаты выполнения задач формируют существенную часть репутационных данных пользователя. Само по себе выполнение задач не составляет репутацию, нужно еще и взаимодействовать с другими пользователями, получать от них положительные отзывы. Решение о правильности выполнения задачи выносят участники данной работы, специальный арбитр или оракул, или алгоритм. Результаты зависят от конкретной задачи, могут иметь форму принятия или отвержения результатов, числовой оценки или отзыва. Оптимальная форма подбирается под каждую задачу отдельно.

Цель протокола – собрать подробные сырые репутационные данные, которые смогут быть использованы клиентскими приложениями, чтобы самостоятельно интерпретировать их для оценки благонадежности пользователей.

2.2 Идентификация

Уровень доверия рассчитывается для каждого пользователя персонально, это одна из основных составляющих протокола. Работа над децентрализованной, персональной идентификацией (Self-sovereign identity, SSI), при которой правом владения и распоряжения собственными данными владеет пользователь [19], идет в паре с развитием технологии распределенных реестров, дополняющих централизованные сервисы идентификации личности. Сервисы идентификации – конкурентная и развивающаяся сфера, где будут появляться новые усовершенствованные стандарты (для примера см. [20]). Сегодня системы идентификации позволяют работать с группами лиц и сценариями анонимного использования.

Однако, доверия на расстоянии невозможно достичь просто за счет более строгой идентификации. В любом сценарии найдутся злонамеренные участники, которые могут быть идентифицированы, но им нельзя будет доверять.

В данном протоколе идентификация не будет основным базовым элементом, доверие на расстоянии будет достигаться за счет соотнесения репутационных данных и идентификации личности. Для каждого аккаунта должна быть пройдена идентификация согласно открытому стандарту идентификации, как это сделано в [21], [22], [23].

Мы особенно подробно рассмотрим стандарты идентификации SSI, которые предлагают такие организации, как uPort [24] и Sovrin [25]. Подход к идентификации SSI, при котором владеет и распоряжается своими данными сам пользователь, а не сервис или компания, удобен для реализации концепции доверия на расстоянии. При этом доверительный центр, например, учебное заведение (при выдаче ученой степени) или орган государственной власти (при выдаче

водительских прав) обозначает набор ***проверяемых*** утверждений относительно идентифицируемого субъекта. Прохождение идентификации в централизованном сервисе может быть использовано в качестве такого проверяемого утверждения для SSI. (К примеру, если пользователь привязал для идентификации SSI свой аккаунт в LinkedIn'е, ему может быть присвоен некоторый уровень доверия). Описанные утверждения, касающиеся идентификации пользователя, совместно с репутацией формируют основу для численной оценки благонадежности пользователя; предполагается, что система будет содержать оба метода оценки данных утверждений и репутации, из чего будет складываться доверие.

2.3 Доверие.

Доверие – это прогноз будущего поведения идентифицированной личности. На основе данного протокола существует возможность оценить способности и намерения пользователя, что будет способствовать установлению доверия на расстоянии. Для оценки способностей пользователя может быть использована его репутация, справлялся ли он раньше успешно с тем, что от него ожидается сейчас. Намерения – мотивация выполнять будущую задачу. И даже если пользователь может справиться с поставленной задачей, ему нельзя ее доверить, если он не мотивирован ее выполнять.

Приведенные выше понятия используются для описания децентрализованной сети доверия, работающей на данном протоколе. Саму сеть можно рассматривать как децентрализованный рынок предсказаний поведения пользователей в будущем. Как предлагали авторы проекта Augur, ценность децентрализации – в данном случае применимо к поведению пользователей – в том, чтобы по-новому взглянуть на то, как люди формируют доверие и проверяют, можно ли доверять другим [26].

3. Цели протокола доверия

Представьте себе мир, в котором взаимодействия и транзакции между незнакомыми пользователями осуществляются через «доверительный слой» интернета, который работает на Протоколе доверия. Репутация формируется на основе данных, которыми владеет и распоряжается сам пользователь. Любому можно выдать соответствующие права доступа к данным, чтобы он смог оценить благонадежность того, с кем он собирается взаимодействовать, а пользователи могут передавать информацию о доверии к себе из одного сообщества в другое. Если коротко, то в сети есть механизм для установления социальных связей и работы шэринговой экономики, благодаря которому возможны более доверительные взаимодействия Пользователей.

Опишем следующие основные принципы Протокола.

3.1. Ценность Доверия-на-расстоянии

Возможность численно оценить доверие-на-расстоянии привносит в интернет более достоверный контент, надежные отношения и благонадежные транзакции. В сообществах реального мира доверие способствует большему количеству возможностей, то же самое верно и в цифровом мире. С учетом того, что в интернете осуществляются миллиарды взаимодействий ежедневно, люди и компании с высоким показателем доверия-на-расстоянии получат преимущества над конкурентами [27], которые выльются в:

- Большее количество возможностей
- Возможность устанавливать повышенные цены на свои продукты и услуги
- Больше запросов на сотрудничество от других игроков.

Доступность доверия-на-расстоянии демократизирует возможности для взаимодействия, что даст преимущество талантливым Пользователям. Для бизнеса доверие-на-расстоянии представляет даже большую ценность, потому что он взаимодействует, совершает транзакции и устанавливает доверительные отношения с очень большим количеством клиентов и партнеров.
Рэйчел Ботсман, ведущий эксперт по доверию, репутационным системам и экономике сотрудничества, описала следующую картину на выступлении на TED в 2012 году: «Возможность использовать поиск , как в Facebook и Google, и видеть полную картину поведения человека в различных ситуациях за нужный срок – лишь вопрос времени. Я вижу, как в режиме реального времени будет формироваться список из тех, кто вам доверился, когда и в каких обстоятельствах и почему это происходило, сюда добавится оценка вашей ответственности на TaskRabbit, вашей аккуратности на основе профиля в Airbnb, знаний, которые вы продемонстрировали на Quora. Все это окажется в одном месте в виде некоторого стенда, который будет показывать ваш репутационный капитал». [28]

3.2. Доверие должно быть проверяемым

Проверяемость означает, что возможен существенно более высокий уровень доверия между Пользователями, так как у них есть возможность перепроверить факты на основе информации из надежного источника. Доверие должно быть проверяемо двумя способами: (1) посредством корректно проведенных транзакций, которые повышают доверие и могут быть проинспектированы другими пользователями при наличии у них соответствующих прав доступа; (2) посредством неизменяемых результатов предыдущих взаимодействий, удостоверенных цифровой подписью участвовавших в них Пользователей. Информация о прошлых результатах приходит на проверку не от самого пользователя и не подвергается изменению.

3.3 Доверие должно быть портативным

Должна быть возможность транспортировать информацию о доверии к данному Пользователю из одного приложения в другое. Должен быть доступ из разных приложений к истории взаимодействий данного пользователя с другими, на основе которой строится доверие к нему. Если Пользователь добился доверия к себе в одном приложении, должна быть возможность это использовать в других приложениях. Портативность позволит Пользователям повысить доверие к себе и распространить его на новые приложения. Она будет стимулировать владельцев приложений внедрять Протокол доверия, так у них будет доступ к более точным данным о Пользователях. Таким образом, новые приложения и сообщества смогут быстро нарастить базу благонадежных пользователей, у которых есть профили в Протоколе и которые заинтересованы в возможностях новых приложений.

3.4. Пользователи контролируют репутационные данные

Принцип SSI для расчета репутации предполагает, что у Пользователя есть полный контроль сохранности и доступа к репутационным данным, доступ к которым будут

запрашивать приложения и сообщества. Пользователь может выборочно предоставлять права доступа к релевантным данным. Пользователь мотивирован делиться своими репутационными данными, чтобы заработать доверие к себе в новом сообществе. Раскрывая свои репутационные данные, Пользователь быстро становится благонадежными членом сообщества.

К примеру, на профессиональном маркетплейсе пользователь раскроет репутационные данные о своих проектах разработки веб-сайтов, а вот информация о том, насколько хорошим квартиросъемщиком он является, будет уже нерелевантной. Пользователь также может контролировать, сколько данных о себе раскрывать. В некоторых случаях Пользователь может принять решение не раскрывать о себе имеющуюся релевантную информацию и быть на правах нового пользователя с низким уровнем доверия. И хотя у пользователя полная власть над своими данными, так же важно, что приложения заинтересованы в том, чтобы создавать Пользователям условия для раскрытия полной информации о себе. В обратном случае отказ пользователя от предоставления данных может трактоваться как важный сигнал о том, стоит ли ему доверять.

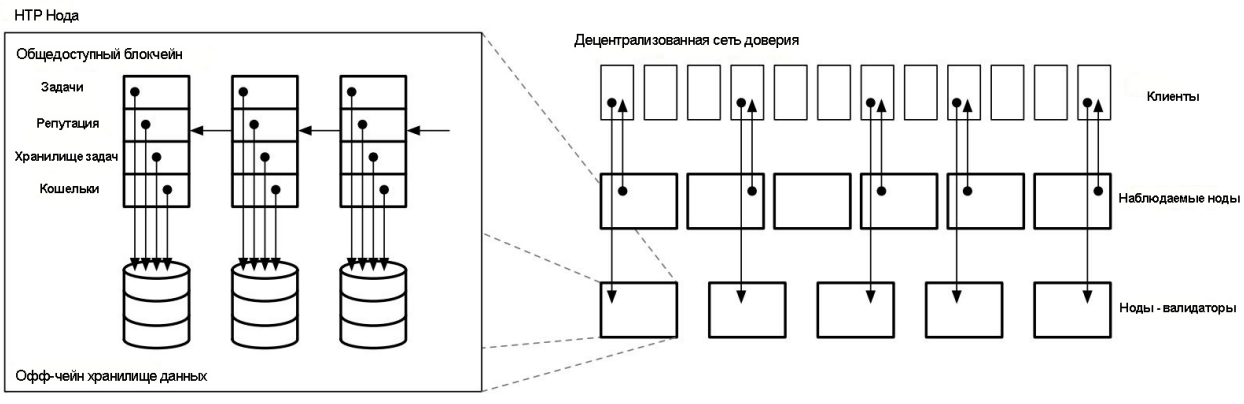
4. Определение Протокола доверия

Протокол доверия создает в интернете слой доверительных отношений. Пользователи определяются, как субъекты, для которых релевантно понятие доверия-на-расстоянии, сюда могут быть отнесены как люди, так и бизнес.
Протокол:

1. Позволяет Пользователям совершать важные взаимодействия с другими участниками
2. Стимулирует Пользователей получать хороший результат
3. Стимулирует Пользователей в результате взаимодействий генерировать репутационные данные
4. Позволяет Пользователям оценивать благонадежность других лиц перед тем, как иметь с ними дело

4.1 Архитектура

Протокол работает как децентрализованная сеть нод на основе публичного блокчейна. Блокейн публичный, так как любой субъект должен иметь возможность получить доступ к этим данным при условии, что Пользователь выдал соответствующие права доступа. Следующая диаграмма описывает основные сущности системы и то, как они друг с другом взаимодействуют.



Взаимодействие пользователей формализовано в виде смарт-контрактов, исполнение которых координируется HTP-нодами; они же служат хранилищем проверяемых репутационных данных и способом генерации этих данных на основе шаблонов взаимодействия. В каждой ноде записаны неизменяемые связи между фактами взаимодействия, записанными в блокчейне и хранимыми в локальной базе данных.
Стабильность в сети достигается за счет многослойной структуры нод. Ноды-валидаторы на базовом уровне имеют возможность записывать информацию о транзакциях и синхронизировать ее с использованием консенсуса BFT. Также они проверяют пользователей на предмет совершения атаки Сивиллы (Sybil attack). (процедура проверки будет описана в одном следующих разделов). Вторая их функция – помочь Пользователям совершать взаимодействия, которые повышают доверие в сети.

В протоколе используются следующие основные понятия: задачи, репутационные профили, хранилище задач, кошельки; все они описаны ниже.

4.2 Задачи

Мы реализуем взаимодействия пользователей в виде сущностей Задач, исполнение которых влияет на репутацию участников. К примеру, Пользователи, которые инициируют нежелательные контакты, реализующие отправку сообщений для целей маркетинга и продаж или запроса экспертного мнения, должны стимулироваться к установлению релевантных контактов. Благонадежность Пользователя в сторонних приложениях будет связана с тем, как он себя ведет в онлайн сообществах, где работают с контентом, т.е. пишут посты, ставят лайки, комментируют или задают вопросы, и где другие Пользователи оценивают его действия. Результаты взаимодействия Пользователя с другими также сигнализируют о его благонадежности в зависимости от того, как он себя ведет на маркетплейсах, где совершают сделки продавцы и покупатели, где продают товары или ищут работу. Стандартизованные Задачи (и их расширения) обеспечивают процесс взаимодействия и позволяют сохранить его результат.

Шаблон Задачи – это абстрактный прототип Задачи, которую можно реализовать и предложить к исполнению Пользователями. Она реализуется в виде смарт-контракта с параметрами, но никогда не исполняется самостоятельно. В протоколе прописаны основные Задачи, которые могут понадобиться в наиболее общих ситуациях. Сообщество разработчиков может генерировать новые шаблоны Задач, которые позволят координировать более сложные взаимодействия и сохранять их результаты. Шаблоны можно корректировать, можно создавать новые на основе уже существующих. Они будут валидными, если соответствуют базовым требованиям написания шаблона. Благодаря возможности создавать новые Задачи появляется классификация Задач. В более поздней главе показаны примеры шаблонов из этой классификации.

4.3 Репутационные профили

Ноды сети хранят регистр репутационных профилей. Каждый репутационный профиль определяется для конкретного Пользователя. Он представляет собой историю всех Задач, в которых Пользователь принимал участие. Репутационный профиль позволяет получить доступ к проверяемой репутации, так как в нем

хранится история реальных действий Пользователя. Так как она хранится в сети, ее можно переносить с одной платформы на другую, а доступ к ней может получить любой сервис, с которым работает Пользователь. Репутационный профиль неизменяем, хотя так как управлять доступом к нему может только Пользователь, протокол позволит частично раскрывать данные сторонним участникам.

4.4 Хранилище задач

В готовящейся версии протокола будет создано хранилище задач. По аналогии с App Store, в хранилище Задач содержится библиотека шаблонов Задач, из которой можно выбирать шаблоны для использования. Сообщество стимулируется пополнять библиотеку Задач новыми полезными шаблонами, повышая тем самым ценность Протокола.

Ссылки на библиотеку Задач хранятся в блокчейне, соответственно, учитывается версионность шаблонов, а также сохраняется история их обновления. Сами же шаблоны хранятся вне блокчейна в хранилищах нод. Каждому шаблону также приписывается репутация, что позволяет Пользователям давать обратную связь относительно его эффективности.

Шаблоны в хранилище Задач публичны и доступны любому Пользователю или клиенту Платформы. Протокол позволяет создавать приватные задачи, если они совместимы с базовым шаблоном.

4.5 Кошельки

Чтобы Пользователи могли работать с Протоколом через множество клиентских приложений, в сети будут храниться защищенные блокчейном Протокола кошельки, в которых есть доступ к балансу токенов Hub. Каждый кошелек связан с аккаунтом Пользователя, приватный ключ находится в распоряжении Пользователя.

5. Токеномика

В этой секции описывается токен Hub и экономика токена, связанная с его использованием.

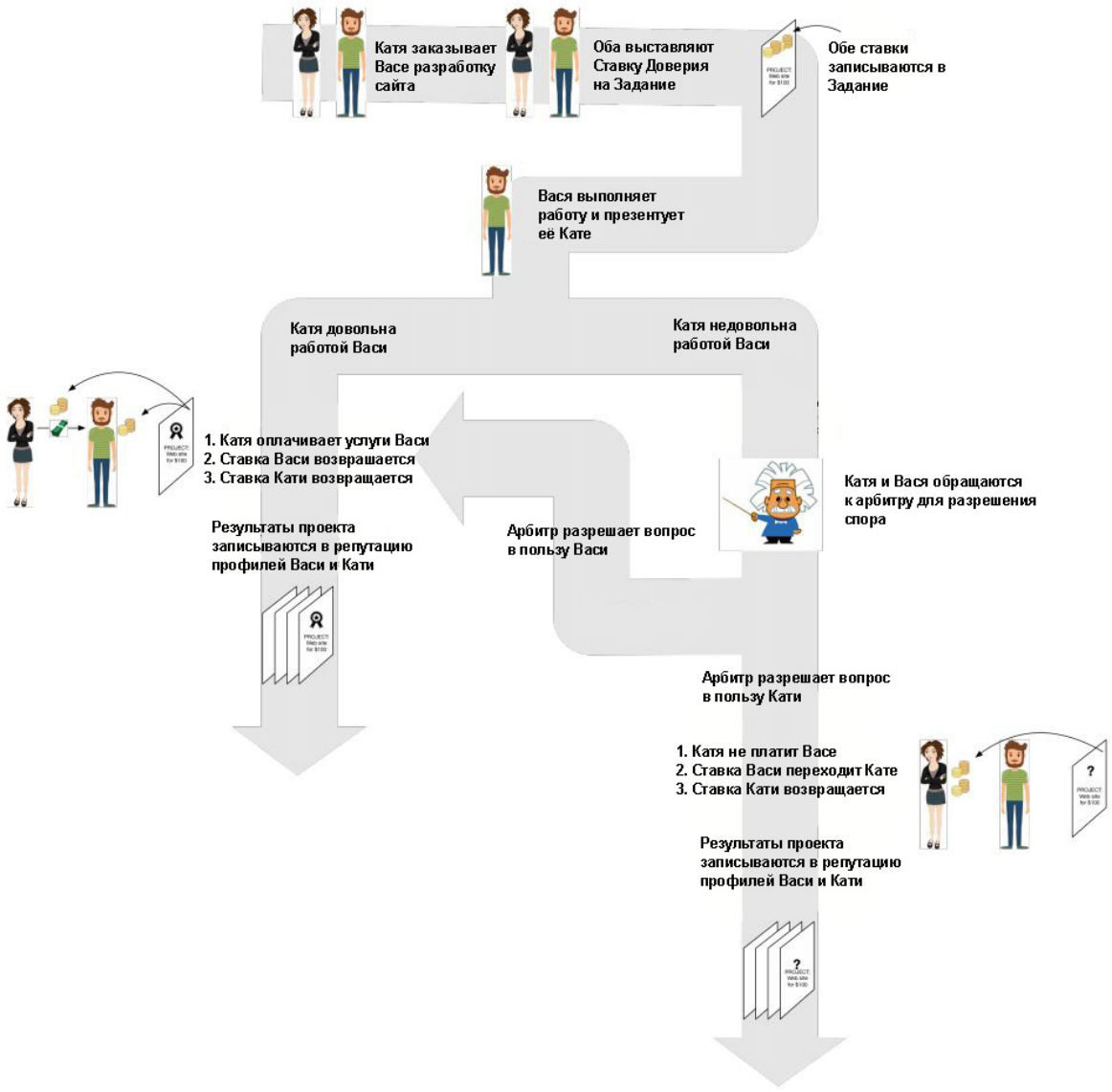
Одна из основных целей Протокола – стимулировать доверительные взаимодействия в сети интернет. Важно не только предложить правильные стимулы, но и избежать или снизить возможность покупки доверия. Протокол останется целостным только если доверие можно будет заработать, подтверждая свою репутацию действиями. Чтобы этого достичь, мы вводим понятие Доверительного залога.

5.1 Доверительный залог

Доверительный залог («залог»): токены, которые вносятся участником Задачи и которыми он рискует, если результат исполнения будет отличаться от запланированного.

5.1.1 Пример доверительного залога

Чтобы лучше понять, как в Протоколе работает залог, давайте изучим пример. Представим сценарий, в котором Пользователь по имени Катя подыскивает дизайнера для проекта создания веб-сайта. Другой пользователь, Вася, заинтересован в том, чтобы работать в проекте Кати. Выполнив проект, Вася получит не только оплату, но и возможность повысить свою репутацию как хорошего дизайнера сайтов.
Чтобы приступить к проекту, Вася и Катя оба вносят залог в обеспечение успешного выполнения проекта. Вася таким образом предоставляет гарантию, что он успешно создаст дизайн сайта. Менее очевидно, но настолько же важно то, что Катя тоже вносит залог, чтобы обозначить свое намерение заплатить Васе в случае успеха. Оба при этом договорились о не зависящей от внесенных залогов сумме оплаты за работу на проекте.



Если проект будет выполнен хорошо, и Кате понравится дизайн Васи, то Васе заплатят, и оба Пользователя заберут назад залоги и вознаграждения за результат трудов. (Детали системы залоговых вознаграждений описаны в последующих главах). Если возникает неразрешимый спор, Катя и Вася соглашаются оплатить услуги арбитра, который примет решение в пользу одного или другого. Если арбитр примет решение в пользу Васи, то исполняется сценарий успешного выполнения проекта, и обоим участникам возвращают залоги (вместе с вознаграждениями). Если

же арбитр выносит решение в пользу Кати, то залог Васи отойдет Кэти. А Васе могут заплатить или не заплатить за работу, в зависимости от того, что решит арбитр.

В любом случае, сам проект, тип проекта, результаты исполнения в терминах залогов записываются в историю проектов Кати и Васи, чтобы любой, кому нужно оценить их благонадежность, мог получить доступ к этой истории. В случае Васи проект отразится на его репутации как поставщике услуг дизайнера. Для Кати будет составлена ее репутация как клиента, и сложится мнение о том, как она работает с исполнителем.

Этот пример можно обобщить на более сложные сценарии, иллюстрируя несколько вещей: (1) для сотрудничества участники вносят существенный залог; (2) существуют четкие правила того, как перераспределяются залоги в зависимости от исхода Задачи; (3) аннотация Задачи и результата сохраняется в историю для оценки доверия-на-расстоянии; (4) залоги – не то же самое, что платежи. В задаче всегда участвуют залоги, но задача может не предполагать оплаты за ее исполнение.

5.1.2 Экономическое обоснование

Понятие залога естественно присутствует в экономической теории и на практике. Оно похоже на гарантию исполнения обязательств [29], обещание одной стороны заплатить другой, если она не сможет исполнить заявленные обязательства. Более строгое обоснование залогов вводится в работе Алекса Табаррока «Контракты с доминирующим убеждением» [30]. В его теоретико-игровом «предприниматель» стимулирует «агентов» (или игроков) вносить свой вклад в производство «общественного блага» при условии, что будет участвовать достаточно большое количество игроков. Если же участников недостаточно, и общественное благо создать невозможно, то игроки все же получают некоторый доход в виде обещанной выплаты от предпринимателя. Если же удается собрать нужное количество участников, то выигрывают все, включая предпринимателя, который в таком случае получает дополнительный доход. Соответственно, игрокам выгодно участвовать независимо от того, будет данный договор успешен или нет. Табаррок показал, что «доминирующей» или лучшей стратегией в данном случае будет участие и внесение своего вклада для всех игроков. В случае Протокола, залоги – это аналоги вкладов, которые делают игроки, чтобы мотивировать других создавать «общественное благо», благоприятный исход.

Примитивная форма контрактов с доминирующим убеждением (которая не решает проблемы безбилетников[31]), «Убеждающий контракт», позволила основать такие компании, как Groupon и Kickstarter [32]. Табаррок установил, что сегодня, в эру децентрализации и смарт-контрактов [33], [34] формы деятельности в виде «Убеждающих контрактов» более актуальны, чем когда-либо.

5.1.3 Требования к залогу

Для большинства задач требуется, чтобы в качестве подтверждения благонадёжности залог вносили все участники. Это мы называем полностью обеспеченными залогом задачами. Требование залога от всех участников позволяет решить проблему nothing at stake (ничего на кону) [35], в которой нарушители ничего не теряют, если ведут себя нечестно по отношению к остальным. В других Задачах может не требоваться залога сразу от всех, такие случаи требуют

тщательной оценки. Они называются частично обеспеченными залогом Задачами. Хотя бы один из участников должен вносить залог. (Примером проблемы «ничего на кону» может служить холодная рассылка почты).

5.2 Арбитры, оракулы и споры

Если Задача идёт не так, как планировалось, и возникает спор, участники могут принять решение обратиться к арбитру, который служит для задачи оракулом (источником достоверной информации), и которому можно доверить суждение о результате выполнения Задачи.

Исполнение обязанностей арбитра - тоже Задача, из категории записываемых в цепочку задач, в отличие от остальных (этот тип задач будет описан в последующих главах). Процедура арбитража может варьироваться. В некоторых арбитражных процессах задействован только один арбитр, в других - целое жюри с заранее определенной схемой голосования для вынесения общего решения. В зависимости от конкретного случая можно разработать подходящий шаблон задачи арбитра.

Арбитры получают за свои услуги плату. Условия компенсации прописываются в шаблоне задания арбитра, источником средств служат либо внесенные под данную задачу залоги, либо сторонние платежи. Когда арбитр выносит решение в пользу одной или нескольких сторон, исполнение исходной Задачи может быть закончено. Благонадёжность арбитра и репутацию, связанная с шаблоном задачи арбитража можно оценить на основе их репутационных данных в системе.

Арбитраж может потребоваться не для всех Задач из-за сопряженных с ним сложности исполнения и издержек. В простых задачах участникам имеет смысл самостоятельно брать на себя роли арбитров, чтобы прийти к окончательному решению.

5.3 Формализация Задач и залога

Формально, залог для Задачи определяется следующим образом:

1. Необходимый залог утверждается отдельно для каждой Задачи, и требует согласия всех участников.
2. У каждого из участников должно быть достаточное количество токенов, чтобы вести залог.
3. Участники проявляют активность при выполнении Задачи.
4. После завершения задачи залоги перераспределяются среди участников.
5. Если Задача успешно выполнена, все получат свои залоги назад за вычетом комиссии ноды за обслуживание выполнения Задачи.
6. В ином случае выбирается метод разрешения спора (участниками самостоятельно или с привлечением арбитра).
7. Если в итоге задача не считается выполненной успешно, залог виновника распределяется среди остальных за вычетом комиссии ноды.

5.4 Стимулирование доверия

Рассмотрим следующую ситуацию: Пользователь, который успешно участвует и вносит залог в большое количество Задач, получит более высокое значение доверия-на-расстоянии на основе истории полученных успешных результатов. Тот же

Пользователь, который оказывается участником Задач, исполненных неудачно, получит сниженный рейтинг не только из-за записи истории неудачных Задач, но и из-за того, что его регулярно лишают внесенного залога.

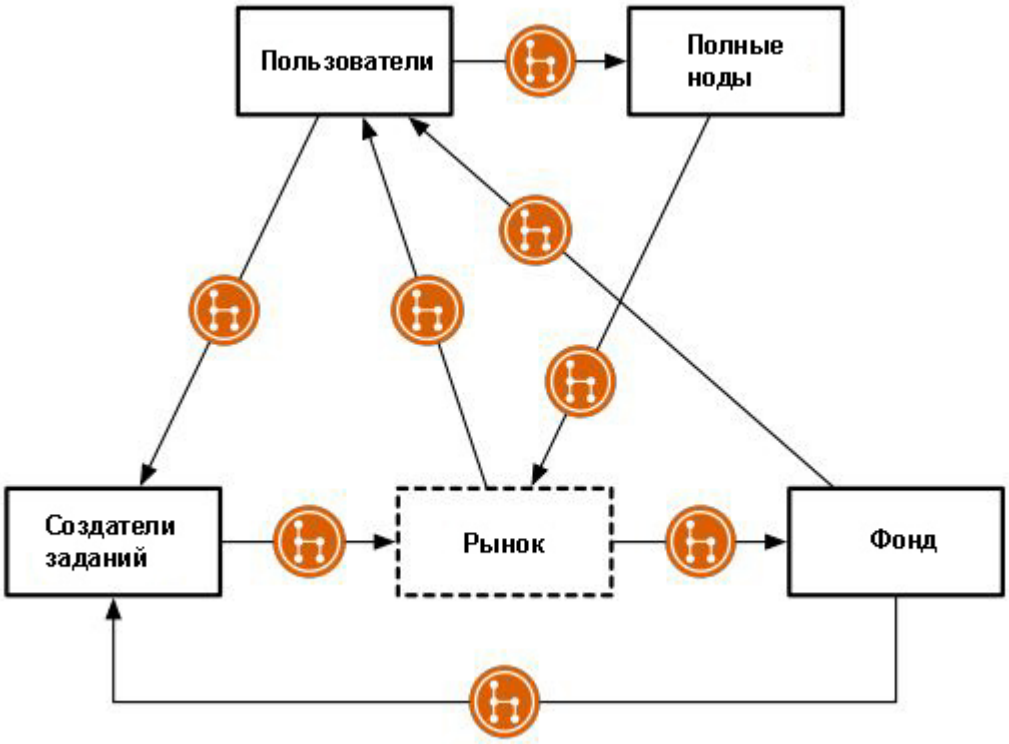
Размер залога определяется типом задачи. Внесённый залог указывает на мотивацию Пользователя работать над Задачей. Покупатель может указать минимальный размер залога, который исключит продавцов, у которых не хватит ресурсов для внесения залога, чтобы участвовать в торгах за возможность выполнять задачу. Продавец может сделать максимально высокую ставку, чтобы покупатель, сравнивающий несколько предложений, выбрал именно его. В основном, для более важных задач потребуется вносить более высокий залог, чем для менее важных. Набор этих условий вынесен из Протокола на уровень Задач. Чтобы упростить процедуру внесения залога, конкретный сервис может вносить залог от имени Пользователя в соответствии с его предпочтениями. Это тема для дальнейших разработок.

Спрос на доверие в интернете подстегнет спрос на токены проекта. Такой же эффект окажет конкуренция за доверие Пользователей. Чем больше токенов у Пользователя, тем больше у него шансов вносить залог и выполнять Задачи, особенно с высокими требованиями к размеру залога. А это влечет за собой рост его благонадёжности и даёт преимущества в реальном мире.

6. Токен Hub

Токен Протокола (Токен Hub) обеспечивает стимулы для доверительного поведения Пользователей и участия в создании Репутационных данных в Протоколе. Пользователи могут становиться майнерами и зарабатывать Токены Hub за деятельность в сети: поскольку они поспособствуют доказательству доверия, участвуя в создании Репутационных данных.

На схеме представлены способы, с помощью которых Токены Hub перемещаются между основными акторами экосистемы:



На приведенной выше схеме рынок представляет собой совокупность способов, с помощью которых пользователи могут торговать Токенами Hub: например, на централизованных биржах или посредством атомных трансакций [36].

Foundation (Фонд) будет создан с несколькими целями. Во-первых, он будет начислять Залоговое Вознаграждение Пользователям после завершения Заданий. (Подробная информация о Вознаграждениях за выполнение Заданий описана в следующем разделе.) Во-вторых, Foundation стимулирует развитие экосистемы, в том числе создание новых и полезных Задач Разработчиками. Фонд может также время от времени выкупать Токены Hub с рынка для дальнейшего стимулирования экосистемы.

6.1. Роль Пользователей

Пользователи используют Токены Hub как Залог в Задачах. Они также могут вознаграждать Токенами Hub Разработчиков задач и Держателей нод за написание и хостинг Задач соответственно. Они могут получать Токены по завершении Задач в виде Залогового Вознаграждения и, возможно, получать Залоги других участников (по спорным Задачам, в которых Арбитр признал их правоту). Вознаграждения поощряют пользователей к доверительному взаимодействию с другими людьми, к дальнейшему распространению Протокола и созданию дополнительных Репутационных данных.

Кроме того, также можно будет производить платежи с использованием Токенов Hub другим участникам для Задач, которые подразумевают платежи.

При появлении потребности в дополнительных Токенах Hub пользователи могут приобретать их с рынка.

6.2. Роль Держателей нод

Держатели нод вознаграждаются Токенами Hub за вычислительную мощность нод. Заработанные Токены Hub они могут свободно продавать на рынке.

6.3. Роль Разработчиков задач

Разработчики, которые создают Шаблоны задач, будут вознаграждены долей из Пользовательского Залога в Задачах, которые они выложили в публичный доступ. Первоначально Foundation также будет стимулировать разработку новых Шаблонов задач. Поскольку разработчики, как и Держатели нод, накапливают Токены Hub, они также могут продавать их на рынке.

7. Залоговое Вознаграждение

Протокол включает в себя систему залоговых вознаграждений, которая стимулирует ее применение Пользователями. Система вознаграждения действий стимулирует пользователей взаимодействовать надежными способами и генерировать Репутационные данные посредством взаимодействия. Ключевыми принципами системы вознаграждения является: (1) обеспечение первоначального распределения токенов Hub при создании учетной записи в сети, чтобы новый Пользователь мог начать участвовать в Задачах; (2) Залоговое Вознаграждение должно возвращаться с бонусом по сравнению с изначальным Залогом.

Поскольку разница между Залоговым Вознаграждением и Залогом будет поступать из пула, который создан для развития экосистемы, стимулы будут постепенно уменьшаться со временем и в конечном итоге нивелироваться по мере созревания сети; в конечном счете система вознаграждения упразднится по мере роста внутренней ценности самого Протокола.

При выполнении обоих принципов, для защиты от возможных Sybil-атак, Пользователь должен выполнить следующие предварительные условия, чтобы получить Регистрационное вознаграждение:

a. Учетная запись связана с одним или несколькими «сильными» идентификаторами, например. Civic [37], SSI, LinkedIn и т. д.
b. За Пользователем не замечена подозрительная активность
c. По Идентификатору новой учетной записи не было получено регистрационное вознаграждение на другой аккаунт

Для принципа (1) система смягчает влияние Sybil-атак, поскольку предпосылки диктуют, что учетная запись должна быть связана хотя бы с одним новым «сильным» идентификатором.

Для принципа (2) функция стимулирует создание и успешное выполнение Заданий; система также решает проблему «нечего терять», так как если участник оставил нулевой Залог за Задание, он не получит Залогового Вознаграждения, как бы он себя ни повел.

Пусть следующая формула представляет коэффициент затухания ϵ размера Регистрационного вознаграждения по мере увеличения числа пользователей:

$$\varepsilon = \max\{1 - \frac{Log(N)}{Log(T)}; 0\},$$

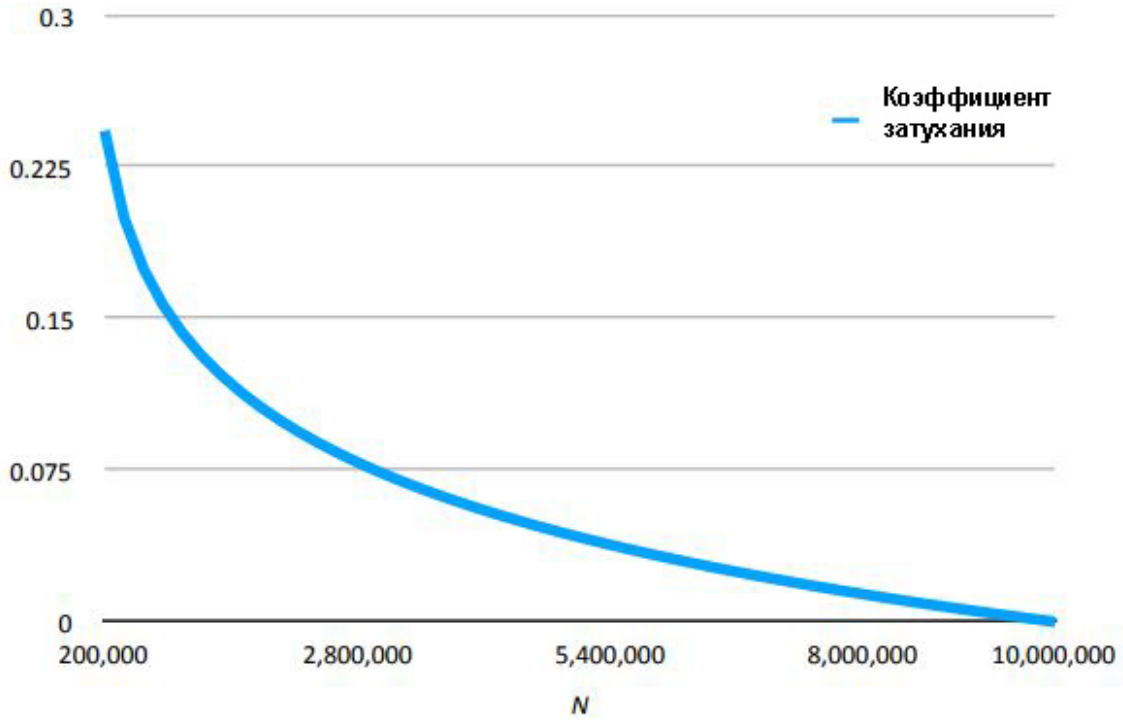
где:
ε: коэффициент затухания
N: количество пользователей в сети
T: постоянная, N-й Пользователь не получит вознаграждение при N>T

Функции вознаграждения при выполнении обоих принципов определяются следующим образом.

Создание учетной записи	Урегулирование Задач
$b\lceil\varepsilon\rceil$ где: b: сумма вознаграждения за создание учетной записи ε: коэффициент затухания	$s \cdot \varepsilon\,(an + bn')$ где: s: Залоговое вознаграждение ε: коэффициент затухания a: пересчитанный коэффициент пользователя, $a \in [0, 1]$ n: количество других участников задачи, которые являются новыми для Пользователя и отвечают предварительным условиям b: существующий коэффициент пользователя, $b \in [0, 1]$, где $b << a$

В модели присутствуют положительные сетевые эффекты, поскольку система вознаграждения способствует взаимодействию с новыми пользователями больше, чем взаимодействию с уже знакомыми пользователями.

Следующий график показывает ε, определяющий коэффициент затухания, для обоих случаев вознаграждения, когда для T установлено значение 10 миллионов Пользователей.



8. Пример систематики Задач

Ценность протокола (и, следовательно, ценность Токена Hub) увеличивается с ростом применимости и разнообразия Задач, доступных в системе, поэтому важно стимулировать сообщество Разработчиков, которые могут создавать Задачи и делать их доступными Пользователям.

На следующей диаграмме показаны некоторые из Заданий, предусмотренных для протокола, которые в конечном итоге будут доступны в системе. Из-за многообразия возможных сценариев и приложений, где важно доверие на расстоянии, множество Заданий не исчерпывается представленным ниже списком. Мы планируем предоставить большинство основных задач и далее будем поощрять сообщество Разработчиков создавать новые Задачи, которые повысят полезность Протокола.

Виртуальные взаимодействия	Отправить сообщение	Отслеживает поток сообщений; один-к-одному или один-ко-многим; получает бонус или штраф в зависимости от реакции получателя
	Сделать рекомендацию	Любая форма рекомендации пользователем одного или нескольких других Пользователей; получает бонус или штраф в зависимости от результатов
	Присоединиться к сообществу	Отмечает присоединение к сообществу
	Покинуть сообщество	Отмечает исключение из сообщества
	Задать вопрос	Отмечает и оценивает ответ на вопрос
	Сделать сообщение или комментарий	Собирает отзывы о публикации или комментариях

Профессиональные взаимодействия	Разместить вакансию	Отслеживает публикацию вакансии и обеспечивает выплату бонуса
	Нанять консультанта	Подписывает контракт, отмечает качество выполненной работы и предоставляет обзор
	Нанять сотрудника	Нанимает сотрудника и отслеживает производительность
	"Пошэрить" актив	Могут быть жилье, автомобиль и т. д. Отслеживает исход для покупателя и продавца
	Оспорить Задачу	Вовлекает арбитра для разрешения спора
Квалификация	Образование	Отслеживание завершения обучения
	Профессиональные сертификаты	Отслеживает получения проф. сертификата
Продукты и услуги	Продать товар или услугу	Отслеживает успешность сделки и отзывы покупателя и продавца
	Подписывать контракт	Отслеживает успешное выполнение контракта и отзывы покупателей и продавцов
	Собрать счет	Трекинг статуса по счету
Передача права собственности	Присвоить название недвижимости	Отслеживает успешную передачу права собственности между сторонами; существует множество форм сделок с титулом; эти Задачи интересны, поскольку они не могут включать финансовый платеж, но тем не менее для успешной транзакции требуется доверие сторон
	Назначить название автомобиля	
Финансы	Получить ссуду	Отслеживает успешное погашение ссуды
	Получить кредит	Отслеживает успешное погашение кредита

9. Возможные атаки и способы защиты

Как и многие децентрализованные системы, система подвержена риску Sybil-атак, поскольку структура взаимодействия между пользователями имеет граф как у социальной сети [38] [39]. В этом разделе мы изложим соответствующие векторы атак и обсудим предлагаемые защитные меры:

9.1. Сговор

Атака: Сговорившиеся злоумышленники создают несколько идентифицированных Пользовательских профилей и выполняют Задачи, чтобы повысить свое доверие или сговориться с другими (мы включаем общий класс Sybil-атак в эту категорию). Текущие решения для атак Sybil обычно делятся на три категории [40]: (1) сертификация доверенным центральным органом; (2) тестирование ресурсов; (3) использование Интернет-сети доверия. Первый способ эффективен в централизованных системах, но не подходит для полностью децентрализованных систем. Тестирование ресурсов - это децентрализованный подход, и доказательство выполненной работы в Bitcoin (PoW), безусловно, популярно, но имеет проблемы с масштабируемостью по мере увеличения числа пользователей сети.

Решение: наш Протокол будет применять Интернет-сеть доверия при валидации Пользователей для защиты от Sybil-атак.

Недавно разработанный и относительно успешный подход – SybilShield – изложен в [41]. Как и другие алгоритмы на основе графов, SybilShield предполагает, что граф социальной сети состоит из нескольких честных и нечестных (Sybil) сообществ пользователей, связанных друг с другом посредством «режущих граней». Он идентифицирует сообщества Sybil, используя стратегию случайного блуждания в сочетании с агентским подходом для минимизации ошибки первого рода. Когда сообщества Sybil найдены в сети, для того, чтобы их надежность была увеличена, они дестимулируются от причинения дальнейшего ущерба. В экспериментах с реальными данными SybilShield дает превосходные результаты по сравнению с аналогичными алгоритмами в своем классе, особенно в снижении ошибки первого рода (FPR), сохраняя при этом эффективность идентификации нод Sybil.

Протокол будет использовать децентрализованную Интернет-сеть доверия для валидации своих пользователей следующим образом:

1. Полные ноды будут периодически проверять базу Интернет-сети доверия с помощью алгоритмов SybilShield. Когда выполняется проверка, пользователи и связанные с ними Репутационные данные будут помечены как честные, нечестные или подозреваемые (переходное состояние в процессе проверки).
2. Чтобы быть уверенными в честности валидаторов, пользователи будут отмечены как Sybil, когда по крайней мере 2/3 валидаторов достигнут консенсуса.
3. Пользователи, отмеченные как Sybil или подозреваемый, получат временный статус. Они не будут участвовать в системе вознаграждения по Задачам до тех пор, пока их статус не будет улучшен или аннулирован. По мере дальнейшего совершенствования Протокола, статус может быть аннулирован частично, т. е. убрана часть из списка привилегий.

По мере своего совершенствования, Протокол внедрит новые способы для анализа

Sybil-атак. Однако новые анализаторы должны учитывать тот факт, что взаимодействия в социальной сети быстро перемешивается [42].

9.2. Нечестные рейтингующие и динамические Пользователи

Атака: в литературе по социальным сетям существуют разные виды недоброжелательных Пользователей. Нечестный рейтингующий ошибочно оценивает хорошую Задачу как неудачу и наоборот. Динамичный Пользователь ведет себя честно, а затем становится злонамеренным.

Защита: в Протоколе влияние обоих типов плохих участников смягчается использованием арбитров и оракулов. Когда результаты оспариваются, участник может предложить использовать арбитра для разрешения спора. Разумеется, арбитры могут оцениваться по собственной «шкале» достоверности с помощью Протокола.

9.3. Нечестный майнинг Залоговых Вознаграждений

Атака: в результате ненадлежащего влияния на сеть злоумышленники могут присваивать вознаграждения, которые должны были зарезервированы для честных Держателей нод.

Защита: чтобы свести к минимуму нечестный майнинг, система вознаграждения построена таким образом, чтобы дестимулировать майнеров к появлению двух версий блока, тем фактом, что для валидации Задач требуется залог Токенов Hub. По мере того, как в спорном статусе оказывается все больше Задач, тем больше затраты на их выполнение, так что в конечном итоге неблагонадежные майнеры исчерпывают свой запас Токенов Hub. Система вознаграждения по Залогам аналогичным образом была разработана для минимизации возможности противоречий.

10. Оценщики доверительности

Оценщики доверительности являются последним компонентом системы. Они помогают пользователям (Доверяющим лицам) оценивать доверие других пользователей (Доверенных лиц). Оценщики доверительности встроены в клиентские приложения и являются алгоритмами, обеспечивающими доступ к Протоколу для создания доверительной аналитики, обычно путем доступа к соответствующим разделам профилей репутации доверенных лиц. Оценщики смогут предоставить краткую сводку Доверяющему лицу, подсчитав метрики доверительности. Они также могут предоставлять оценку конкретных историй о потенциальном доверенном лице.

Проект предоставит библиотеки с открытым исходным кодом Оценщиков доверительности, которые могут быть интегрированы в клиентские приложения. Эти Оценщики будут внедрять передовую практику для оценки доверия, чтобы все клиентские приложения могли извлечь выгоду, анализируя такие аспекты, как:

1. Новизна Доверенного лица
2. Повторяемость Задач («угасание» старых задач для определения динамических личностей [43])
3. Размер Залога в задачах

4. Разнообразие участников, вовлеченных в задачи Доверенного лица
5. Было ли Доверенное лицо отмечено как злоумышленник Sybil

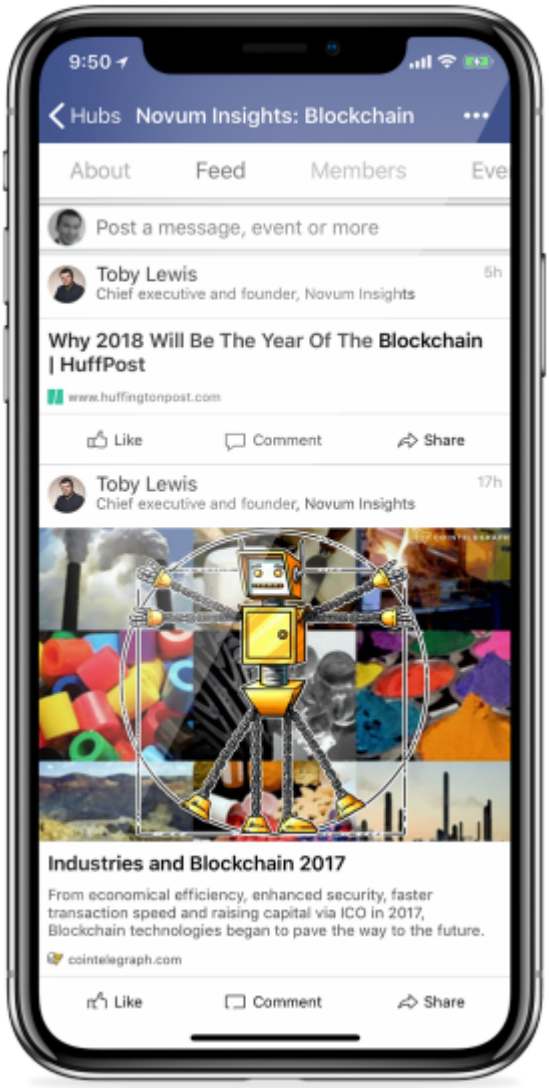
Оценщики доверительности также могут помочь в анализе намерения Доверенного лица по новой Задаче, оценив его Залог по сравнению с аналогичными Задачами.

Мы подчеркиваем, что доверительные оценщики не работают одинаково хорошо для всех типов Задач, потому что репутация в значительной степени контекстуальна. Поэтому Оценщики доверительности должны быть разработаны специально для их предполагаемого применения. Предоставленные библиотеки должны служить отправными точками в Разработке, а сами Оценщики более точно настраиваться под контекст Задачи.

Оценщики доверительности остаются важной областью исследования, поскольку новые шаблоны заданий разрабатываются сообществом, и их следует непрерывно улучшать, чтобы дополнять экосистему новыми формами взаимодействия.

11. Приложение Hub

Мы уверены, что Протокол доверия (описан в части 3) покажет свою эффективность и займет свою нишу в Интернете, однако, чтобы завоевать доверие пользователей, его ценность должна быть широко доказана. В целях ускорения повсеместной интеграции Протокола мы планируем использовать приложение Hub. Приложение Hub – это профессиональная социальная сеть нового поколения, выросшая из мессенджера. Это приложение даст большому количеству профессионалов доступ к новым возможностям в своей сфере и профессиональных сообществах, а также позволит получить нужные связи и наладить бизнес-процессы.



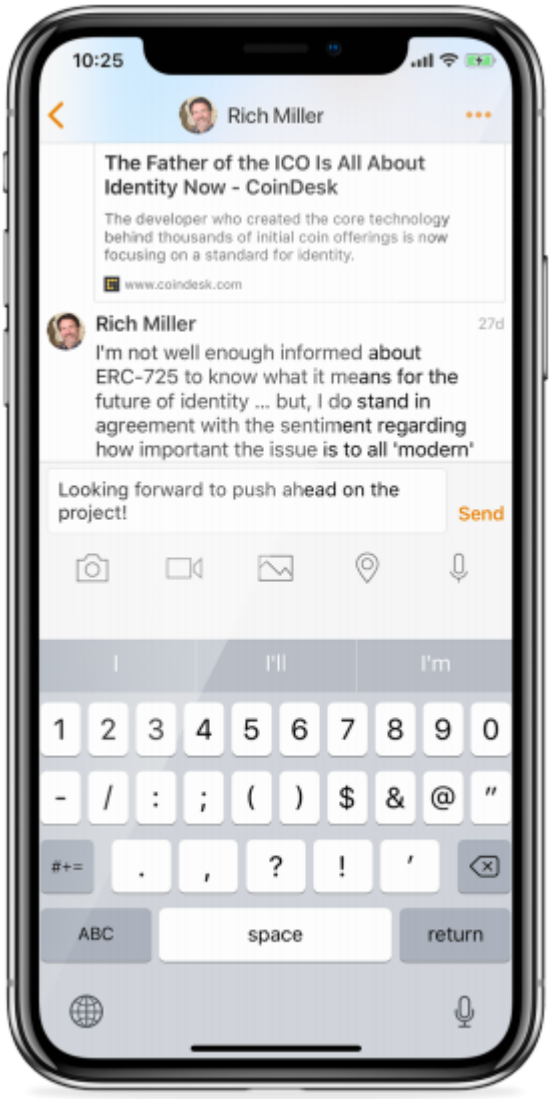
Несмотря на широкий выбор соперничающих между собой мессенджеров, они почти не затрагивают профессиональную и бизнес аудиторию, что представляет огромные возможности для рынка. Профессиональные соцсети идеально подходят для применения Протокола, так как бизнесмены и специалисты в различных сферах деятельности часто взаимодействуют с незнакомыми людьми: клиентами, поставщиками, партнерами, и оказываются в ситуациях, когда взаимодоверие просто необходимо в профессиональных интересах. Используя широко распространенные мессенджеры в деловых целях, пользователи часто испытывают дискомфорт, ведь таким образом их коллеги или будущие

партнеры получают доступ к большому объему персональной информации пользователя.

Находящееся на стадии разработки приложение Hub задаст тон использованию и дальнейшему развитию Протокола. Несмотря на все силы, которые мы вкладываем в развитие Hub сейчас, нашей конечной целью является успешное внедрение Протокола доверия и системы связанных приложений, направленных на создание человеческого доверия между интернет-пользователями.
Главные свойства приложения Hub описаны в следующих частях.

11.1. Сообщества

Основой приложения Hub являются непосредственно хабы – микросообщества, существующие в рамках той или иной индустрии, сферы, круга интересов, локального комьюнити, общего дела. Внутри каждого хаба есть новостная лента, в рамках которой пользователи могут обмениваться публикациями и взаимодействовать так же как в социальных сетях - посредством лайков, комментариев и репостов, одновременно узнавая новых членов сообщества. Внутри хаба также доступна функция создания и управления мероприятиями сообщества. Функции хаба также будут распространяться на обмен документами, вакансиями и мероприятиями, а также различными товарами. Хабы будут обладать очень гибкими механизмами управления: пользователи сами смогут принимать или ограничивать круг участников, а также широту полномочий каждого из них внутри сообщества. Важной целью проекта является предоставление адекватных инструментов для самоуправления сообщества.



11.2 Обмен сообщениями

Для того чтобы пользователи могли в полной мере получать новые контакты и взаимодействовать друг с другом напрямую, Hub предусматривает встроенную функцию обмена сообщениями. Пользователи смогут использовать сообщения, как для совместной работы, так и для собственного продвижения. Как все современные мессенджеры, Hub будет поддерживать обмен мультимедиа, документами, голосовые, видео и текстовые чаты.

11.3 Денежные переводы

В процессе взаимодействия пользователям также пригодится возможность обмена денежными средствами. Пользователям Hub будет доступна встроенная система перевода денежных средств как на уровне сообщества, так и непосредственно между пользователями. В таблице ниже приведены

уровни транзакций и некоторые примеры:

Уровень	**Пример**
Переводы в пределах сообщества	• Членские сборы • Мероприятия • Платный контент • Рекламные объявления
Переводы непосредственно между пользователями и Сообщения	• Сделки • Выставление счетов • Переводы, не требующие подтверждения • Маркетинговые коммуникации • Переход права собственности

11.4 Безопасность и конфиденциальность

Осознавая особую важность конфиденциальности и безопасности данных пользователей в профессиональном контексте, Hub по умолчанию будет использовать сквозное шифрование как сообщений, так и контента сообщества. Наиболее перспективным в данном случае является Сигнальный протокол [44]. Мы также рассматриваем децентрализацию базы данных приложения. Следует подчеркнуть, что эта технология сейчас только зарождается и быстро развивается. Мы продолжим изучать развитие технологий в этих областях и внедрять усовершенствования в приложение, по мере развития технологий. Возможности приложения Hub будут постоянно обновляться независимо от обновлений Протокола.

11.5 Внедрение Протокола доверия в приложение Hub

Приложение Hub будет служить как потребителем, так и источником репутационных данных, помогая пользователям создавать доверительные отношения с сообществами. Многие активности в сети могут быть представлены в виде Задач, и мы будем внедрять дополнительные типы Задач, по мере масштабируемости Протокола:

Фаза	**Описание**
1. Оценщики доверительности	Каждый пользователь приложения будет иметь репутационные данные. Основа для доверия первоначально будет интегрирована в приложение и будет работать с исходными данными, такими как заявки SSI (идентификаторы) и другие данные из внешних источников.
2. Взаимодействие с сообществом	Первые Задачи для широкого диапазаона возможных действий не будут связаны с платежами, но предполагают Залог с использованием Токенов Hub. Эти действия могут включать такие взаимодействия, как присоединение к сообществу и публикация вакансии (см. ч. 17-18).

3. Взаимодействие с платежами	Будет поддерживаться взаимодействие на уровне сообщества, которое требует платежей, таких членство в сообществе или публикации работы. На этом этапе также будут поддерживаться Токены Hub, используемые в качестве способа оплаты.
4. Переводы непосредственно между пользователями	Поскольку Протокол масштабируется для обработки большего количества транзакций, задачи для одноранговых взаимодействий и транзакций будут доступны для пользователей. Примеры включают трудовые контакты и выставление счетов.
5. Интеграция Хранилища задач	Из Хранилища Задач приложение-Hub предоставит пул, в котором пользователи смогут использовать все разработанные Шаблоны Задач для своих целей.

12. Определение децентрализованной сети доверия

В следующих разделах мы формально определяем структуры данных и Протокол доверия. Мы начинаем с базовых определений, а затем расширяем до двух реалистичных применений протокола: по отправке сообщений и публикации вакансий.

Сначала мы опишем децентрализованную сеть доверия, которая представляет собой множество нод, которые управляют протоколом и хранят репутационные данные. Структура данных, хранимых нодой описана ниже.

12.1. Структура данных

$$N := \{TS, \{t_1, .., t_n\}, \{r_1, .., r_n\}\}$$

- N, ноды
- TS, Хранилище задач
- $\{t_1, \ldots, t_n\}$, Журнал задач
- $\{r, \ldots, r\}$, Профили репутации

Каждая полная нода имеет доступ к хранилищу Задач, истории выполненных задач и профилям репутаций пользователей протокола. По мере масштабирования Протокола будет изучаться использование различных методов, чтобы требования к объему памяти нод были выполнимыми.

13. Определение Шаблона задачи

Шаблон задачи - это абстрактный базовый шаблон для всех других Шаблонов Задач. Конкретные шаблоны заданий будут определять требования к Залогам и содержать логику возврата Залога при различных результатах. Кроме того, для всех Заданий потребуются атрибуты метаданных, и они указаны здесь в базовом шаблоне.

13.1. Структура данных

$T := \langle id, desc, source, cost, (u_1,..,u_n), \{u_1 \to s_1,..,u_n \to s_n\}, start, completed, outcome, outcome\ data \rangle$ где:
• T, Шаблон задачи . • id, глобально адресуемый уникальный идентификатор Шаблона • $desc$, короткое описание Шаблона • $source$, клиент, запросивший разработку Задачи • $cost$, количество Токенов Hub к оплате Разработчику Шаблона задачи • $(u_1, \ldots, u_n)$, Пользователи; для идентификации по IDs • $\{u_1 \to ss_n, \ldots, u_n \to s_n\}$, требуемые Залоги ($s_i$) от Пользователей ($u_i$) в Токенах Hub • $start$, Временная метка начала выполнения задачи • $completed$, Временная метка окончания выполнения задачи • $outcome$, пересчитанный результат • $outcome\ data$, переменные результирующие данные

13.2. Протоколы

13.2.1. Instantiate (интерполировать)

Создает частный случай Задачи из Шаблона задачи.

$t := Instantiate(client\ source, (u_1,..,u_n), \{u_1 \to s_1,..,u_n \to s_n\})$, где:
Входные данные: • $client\ source$, клиент Задачи • $(u_1, .., u_n)$, участвующих пользователей; u_0 - это инициатор Задачи • $\{u_1 \to s_1, .., u_n \to s_n\}$, Залог s_i участника u_i Выходные данные: • t, интерполированная задача Условия:

1. *source* := *client source*
2. Хотя бы один пользователь внес Залог в Задачу
3. Все пользователи имеют Залоги в соответствии с требованиями к шаблону
4. Для каждого пользователя u_i Задача t добавлена в репутационный профиль u_i
5. Залоги закладываются до t до открытия Задачи
6. Перенести Токены Hub в размере *cost* Разработчику Задачи

14. Определение Задачи

Задача – это пример Шаблона Задачи. Содержимое Задачи криптографически зашифровано и доступно только участникам.

14.1. Протоколы

14.1.1. Settle (клиринг Задачи)

Протокол Settle вызывается при завершении задачи, чтобы активировать перераспределение Залоговых Вознаграждений участникам Задачи в зависимости от результата. Он специально реализован в не базовом Шаблоне задачи, а в его расширениях.

Settle(*outcome*[, *outcome data*])
Входные данные: • *outcome*, Результат задачи • *outcome data*, дополнительные данные, относящиеся к Результату: • t, Задача Условия: 1. Залоговые Вознаграждения перераспределяются между участниками в соответствии с Итогами Задачи, реализованными в смарт-контракте Шаблона Задачи

15. Определение репутационного профиля

Репутационный профиль - это данные о Задачах Пользователя, в которых он участвовал, с их результатами и типами. Каждый профиль репутации является суверенным. Протокол определяет операции, которые делают доступ к профилю авторизованным и санкционированным.

15.1. Структура данных

$R := \langle u, invalidated, (t_1, .., t_n) \rangle$

- R, репутационный профиль
- u, идентификатор Пользователя
- $invalidated$, 0, если Пользователь считается честным после валидации; 1 в противном случае
- (t_1,., t_n), история задач задач, в которых u является участником

15.2. Протоколы

15.2.1. Map (клиринг Задачи)

Протокол карты предоставляет метод разрешения Пользователю доступа и оценки репутационного профиля другого Пользователя. Карта может использоваться Клиентами для «оценки» репутации данного Пользователя, например, при вычислении метрик доверия или визуализации данных о репутации Пользователя. Во всех случаях доступ должен быть разрешен владельцем профиля и может быть реализован как смарт-контракт. Параметр спецификации запроса ($search\ spec$) действует как фильтр для истории задач одного Пользователя, который сканирует атрибуты запроса и возвращает усеченный репутационный профиль. Владелец профиля должен согласиться как с поисковой спецификацией, так и с видимостью атрибутов профиля для других Пользователей.

$Map(u, search\ spec, result\ attributes)$

Входные данные:

- u, запрашивающий Пользователь
- $search\ spec$, критерии поиска для профиля репутации; дерево запросов сопоставлений атрибутов и критериев
- $result\ attributes$, атрибуты Заданий, которые могут быть возвращены и сделаны видимыми для запрашивающего Пользователя

Выходные данные:

- (t_1,., t_n), список задач, соответствующих спецификации поиска ($search\ spec$), только с доступными атрибутами ($result\ attributes$)

1. Разрешение владельца Репутационного профиля, состоит в том, что пользователь u авторизован для установки параметров поиска и просмотра атрибутов Заданий.

2. В противном случае, протокол вычисляет O:= для каждой задачи t_i в репутационном профиле, собирает для вывода только те атрибуты в атрибутах результата, где t_i соответствует критерию поиска

3. Протокол возвращает O

15.2.2. Attest (Подтверждение)

Реализует доказательство с нулевым знанием (zero-knowledge proof) в репутационном профиле пользователя для запрашиваемого подтверждения с правами владельца. Доказательство с нулевым знанием, реализованное через zk-SNARKs [45], требуют предварительной «доверительной настройки» в зависимости от требуемого подтверждения. Особенности реализации еще будет более детально исследованы. (Доказательство с нулевым знанием в конечном итоге может быть реализовано с помощью zk-STARKs [46], для которого не требуется этап «доверительной настройки».)

$Attest(u, criteria)$
Входные данные: • u, запрашивающий Пользователь • $criteria$, спецификация репутации, которую следует подтвердить; сопоставление атрибутов Задачи с их ожидаемыми значениями Выходные данные: • true, если подтверждение завершается успешно. В противном случае – false. 1. Протокол подтверждает разрешение от владельца Профиля, на то что Пользователь u мог проверить критерии. Если подтверждения нет, то не проверка не получится. 2. Тогда R: = $false$ 3. В противном случае для каждой Задачи t_i в профиле репутации Протокол задает R: = $true$, если t_i соответствует критериям 4. Вывод R

16. Определение Хранилища задач

Хранилище задач появится в будущей версии протокола и будет хранить общедоступные Шаблоны задач для создания Задач с помощью Протокола интерполяции.

16.1. Структура данных

$TS := \{ \langle T_1, r_1 \rangle, \ldots, \langle T_n, r_n \rangle \}$ • TS, Хранилище задач • $\langle T, r \rangle$, набор Шаблонов задач и связанных с ними репутационных профилей

16.1.1. AddUpdate (Обновить)

Данная структура добавляет новый Шаблон задачи или применяет обновление к существующему.

$AddUpdate(T)$
Входные данные: • T, Шаблон задачи Выходные данные: нет 1. Поиск существующего Шаблона T_0, используя идентификатор T 2. Проверка на предмет того, что запрашивающий Пользователь может добавить или обновить шаблон. Если не может, то обновить не получится. 3. В противном случае, если T_0 существует, производится замена T_0 на T 4. если T_0 не существует, то добавляется T

16.1.2. Search (Поиск)

Находит Шаблоны задач в соответствии с заданными критериями поиска.

$Search(criteria)$
Входные данные: • *criteria*, (критерии) сопоставление атрибутов Шаблонов задач с их желаемыми значениями Выходные данные: • (T_1,., T_n), Шаблоны задач, соответствующие критериям 1. Определение O: = {} 2. Для каждой Задачи T_i в Хранилище задач, которая соответствует *criteria*, добавляет <T_i, R_i> в O 3. Вывод O

17. Пример Шаблона: Коммуникативная Задача

В следующих разделах мы приводим различные примеры полезных Шаблонов задач. Мы будем ссылаться на эти Шаблоны как на Задачи, хотя подразумевается, что они фактически являются Шаблонами задач.

Сообщения - это основной элемент большинства социальных сетей. Обычно сообщения используются для межличностного взаимодействия, но оказываются полезны и для рекламных рассылок и других видов несогласованного взаимодействия. Платный прием сообщений может быть указан Пользователем, если он, например, является консультантом и взимает плату за свои услуги. Коммуникативная Задача представляет собой одно сообщение или поток сообщений между отправителем и получателем. Будучи изложенной в таком виде, Коммуникативная Задача представляет собой частный случай Задачи с частичными Залогами, поскольку для ее исполнения требуется Залог только от отправителя сообщения. Коммуникативная Задача может применяться для более сложных взаимодействий: с ее использованием можно отправлять сообщение двум и более Пользователям и создавать групповые чаты или спонсируемые рассылки, которые достигают сразу многих Пользователей.

Клиент создает Коммуникативную Задачу, которая состоит в передаче сообщения по инициативе Пользователя (отправителя) другому пользователю (получателю). Отправитель обязуется передать свой Залог для отправки сообщения. Платеж, если без него невозможна отправка сообщения получателю, также совершается и при передаче сообщения. Пока Задача не закрыта, получатель и отправитель могут продолжать обмениваться сообщениями, и задача остается активной. Если получатель в конечном итоге отмечает поток как спам, Задача закрывается, и получатель получает Залог, обещанный отправителем. Задача затем закрывается.

Коммуникативная Задача дополняет базовый Шаблон задачи (см. часть 13) атрибутами, подходящими именно для этой задачи. (Важно заметить, что не все формы обмена сообщениями должны быть охарактеризованы как Задачи, а только те, которые могут повлиять на доверие.)

17.1. Структура данных

$$M := T + \ \langle message\ id[, payment]\rangle$$

- M, сообщение
- T, базовый Шаблон задачи
- $message\ id$, ссылка на вне-блокчейн данные (с содержимым отправляемого Сообщения)
- $payment$, платеж получателю при ответе или акцепте контакта; также указывается расчетная валюты, который может быть в Токенах Hub

17.2. Протоколы

17.2.1. Instantiate (интерполировать)

Расширяет базовый Протокол интерполяции из базового Шаблона задачи.

$m := Instantiate((s, r), \{s \to s_S\}, message\ id\ [, payment])$, где:

Входные данные:

• s, идентификатор Пользователя Отправителя

• r, идентификатор Пользователя Получателя

• s_s, Залог отправителя

• $message\ id$, ссылка на вне-блокчейн данные (с содержимым отправляемого Сообщения)

• $payment$, оплата получателя

Выходные данные:

• m, Коммуникативная Задача

Протокол:

1. Устанавливает участников (s, r)

2. Устанавливает Залоги $\{s \rightarrow s_s\}$

3. Задает идентификатор задачи : = "$message$"

4. Сообщение, помещенное в $message\ id$, отправляется от s к r

17.2.2. Accept (Приём)

$Accept()$
Входные данные: • нет Выходные данные: • нет Протокол: 1. Если указано $payment$, эта сумма переносится от s к r

17.2.3. Settle (закрытие Коммуникативной задачи)

Расширение базового протокол Шаблона Задачи, которое предотвращает дальнейшие несанкционированные сообщения между пользователями.

$Settle(outcome)$
Входные данные: • $outcome$, результат Задачи, один из: $sent \mid read \mid accepted \mid replied \mid spam$ Выходные данные: • m, Коммуникативная Задача Протокол: 1. Если результат Задачи - $spam$, то r получает сумму s_s от s 2. В противном случае, сумма s_s возвращается к s

18. Пример Шаблона Задачи: Опубликовать вакансию

В Задаче Опубликования вакансии работодатель обязуется предоставить Залог и опубликовать вакансию, которая может представлять интерес в определенном сообществе. Кроме того, данная Задача предполагает реферальную программу, состоящую из выплаты части Залогового Вознаграждения за успешную Рекомендацию кандидата на должность. Гарантией качественной рекомендации также выступает Залог, который Рекомендующий прикладывает к своей рекомендации.

Онлайн-рекрутинг и онлайн-консалтинг по вопросам трудоустройства - это многомиллиардная индустрия, но процессы найма по-прежнему трудоемки, малоэффективны и неопределенны. Мы считаем, что мотивация доверенных рефералов, и достоверность резюме для проверки рекомендаций создает хорошее подспорье для доверия на расстоянии. Шаблон задания, который организует процесс рекрутинга с соответствующими стимулами среди всех участников, может привести к повышению надежности, скорости и снижению стоимости этого процесса.

Шаблон «Опубликовать вакансию» позволяет работодателю публиковать требования к соискателю и предложить сообществу реферальную программу. На примере этой Задачи показано, как Шаблон задачи может специализироваться и объединять другие Задачи для соответствия определенным требованиям. В этом случае шаблон задания расширяет шаблон Коммуникативной Задачи и использует сообщения для отслеживания рефералов.

Клиент создает свою копию Задачи от имени работодателя, указывая необходимые параметры для публикации Задачи. Оплата производится получателю, который дал успешную рекомендацию по трудоустройству, или сообществу, после успешной публикации. В любой момент Пользователь может отсылать кандидата (другого Пользователя) к работодателю, который вызывает протокол $AddReferral$, чтобы с помощью сообщения связать рекомендателя с Задачей. После того, как публикация вакансий должна быть закрыта, независимо от того, был ли в итоге найден сотрудник или нет, Клиент вызывает протокол $Settle$ с результатом в Задаче. Если Задача было успешно заполнена, то работодатель решает, что один из рекомендателей должен получить награду, и Залоговое Вознаграждение выплачивается успешному рекомендателю. Если были обнаружены последствия злонамеренных действий, соответствующие Ставки перераспределяются другим участникам. (Подробнее об этом ниже.) Если с Задачей ничего не происходит, Залоги возвращаются первоначальным залогодателям без изменений.

18.1. Структура данных

$J := M + \langle \{r_1, .., r_n\}, job\ title[, salary, ...] \rangle$

- J, Шаблон Задачи по публикации вакансии
- • M, Шаблон Коммуникативной Задачи
- • $\{r_1, .., r_n\}$, набор из Коммуникативной Задачи для привязки

рекомендателей

- • $job\ title$ [, $salary$, ...], атрибуты вакансии

18.2. Протоколы

18.2.1. Instantiate (интерполировать)

$j \coloneqq Instantiate(p, p \to s_{\text{u}}, job\ title, job\ description, bounty[, salary, \dots])$, где:

Входные данные:

• p, идентификатор Пользователя Работодателя

• $p \to s_{\text{p}}$, размер Залога Работодателя

• $job\ title$, название вакансии

• $job\ description$, ссылка на вне-блокчейн данные передается в сообщении $message\ id$

• $bounty$, вознаграждение, сохраненное в сообщении «$payment$»

Выходные данные:

• m, Задача Публикации Вакансии

Протокол:

1. Устанавливает p в качестве участника задачи

2. Устанавливает s_{p} в качестве Залога Работодателя

3. Устанавливает $message\ id$: = $job\ description$

4. Устанавливает $payment$: = $bounty$

18.2.2. AddReferral (добавить рекомендателя)

Добавляет рекомендателя с помощью Коммуникативной Задачи в Задачу Публикации вакансии.

$AddReferral(m)$, где:

Входные данные:

• m, Сообщение

Выходные данные:

• нет

Протокол:

1. Устанавливает, что получатель m является создателем Задачи Публикации вакансии
2. Устанавливает, что сообщение m еще отправлялось в рамках этой Задачи Публикации вакансии
3. Добавляет m в Задачу Публикации вакансии

18.2.3. Settle (закрытие Задачи публикации вакансии)

Данный протокол вызывается, когда публикация вакансии уже закрыта и оценивается, следует ли выплачивать вознаграждение рекомендателю.

$Settle(outcome[, outcome\ data])$, где:
Входные данные: • *outcome*, исход, один из *not placed* | *placed* • *outcome data*, если *outcome* является *placed*, то *outcome data* является идентификатором рекомендателя r, который должен получить Залоговое вознаграждение Выходные данные: • нет Протокол: 1. Если исходный результат и успешный рекомендатель r существуют, переводит $bounty$ рекомендателю r.

18.3. Диспуты

При закрытии Задачи публикации вакансии может возникать несколько спорных сценариев (диспутов). В данной части описано, как можно разрешить споры, и, что более важно, как Залоги могут быть справедливо перераспределены, чтобы обеспечить надлежащие стимулы для участников Задачи.

Рассмотрим ситуацию, когда Работодатель получает рекомендации высокого качества и отмечает их как неподходящие, в попытке оставить Залоги Рекомендателей себе. В аналогичном сценарии Работодатель нанимает одного из рекомендованных, но пытается не выплатить награду Рекомендателю.

Мы утверждаем, что в обоих случаях спор может быть разрешен с использованием арбитра, который может быть менеджером сообщества, где была размещена вакансия. Арбитр может изучить детали диспута, и вынести профессиональное суждение. Если они решают в пользу Рекомендателей, Работодатель теряет свой Залог, который распределяется среди всех Рекомендателей. Когда же оказывается прав Работодатель, он получают Залог от «неправдивого» Рекомендателя (-ей) (со всеми другими «правдивыми» Рекомендателями, возвращающими свои ставки).

19. Дальнейшая работа

19.1. Разработка систематики Задач для эффективной оценки доверительности

По мере увеличения количества Шаблонов Задач, актуальность в их категоризации для более точной оценки доверительности только возрастает. Организация систематики традиционно проводилась в централизованных системах. Мы обдумываем децентрализованное решение задачи, где Шаблоны Заданий могут быть созданы и в конечном итоге эффективно распределены по категориям.

19.2. Расчет рекомендуемых Залогов

Стимулы к доверительному поведению в децентрализованной системе будут хорошо работать при правильном определении значений Залогов для различных Задач. Соответствующее исследование экономических моделей и их применение не только поможет пользователям определить оптимальную сумму Залога в Задачах, но и будет способствовать полномасштабному распространению Протокола.

20. Заключение

Интернет был разработан как глобальная открытая сеть на основе протокола для обмена информацией. Социальные сети и мессенджеры стали одними из самых массовых и надежных применений Интернета. К сожалению, по мере роста этой сети неспособность пользователей создавать доверительные отношения с незнакомцами привела к множеству серьезных ограничений, которые препятствуют будущему экономическому развитию Интернета.

Концепции неизменной базы данных и децентрализованной архитектуры позволяют создать доверие-ориентированный способ взаимодействия, который может значительно повысить экономическую ценность для пользователей, которые взаимодействуют с другими пользователями через Интернет.

21. Благодарности

Авторы хотели бы поблагодарить следующих людей, которые своими предложениями внесли ценный вклад в написание этой работы: Кен Фромм (Ken Fromm), Кен Келлер (Ken Keller), Фред Крюгер (Fred Krueger), Николай Орешкин (Nikolai Oreshkin), Алекс Пун (Alex Poon), Майк Прайнс (Mike Prince) и Кайл Ванг (Kyle Wang).

22. Список литературы

1 https://www.ted.com/talks/rachel_botsman_the_currency_of_the_new_economy_is_trust 2 http://www3.weforum.org/docs/WEF_Realizing_Potential_Blockchain.pdf
3 http://blogs.teradata.com/international/how-much-data-we-create-daily/
4 https://en.wikipedia.org/wiki/Dunbar%27s_number

5 https://en.wikipedia.org/wiki/Robin_Dunbar
6 https://twitter.com/NickSzabo4/status/917474578640732160
7 https://www.lifewire.com/how-many-email-users-are-there-1171213
8 http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf
9 https://www.lifewire.com/how-many-emails-are-sent-every-day-1171210
10 https://www.wired.com/2014/05/sharing-economy-fico/ - см. комментарий Monroe Labouisse об Airbnb
11 https://www.linkedin.com/
12 https://www.csoonline.com/article/3036072/social-networking/the-rise-of-linkedin-fraud.html
13 https://www.linkedin.com/pulse/why-so-many-fake-data-scientist-bernard-marr
14 https://3qdigital.com/socialmedia/linkedin-fake-profile-heaven#.WinVjVQ-fUI
15 https://books.google.de/books?id=myAAyj0hmq8C, c. 95
16 https://web.stanford.edu/~leinav/pubs/AR2016.pdf
17 http://www.fico.com/en/products/fico-score
18 https://en.wikipedia.org/wiki/Sesame_Credit
19 https://medium.com/learning-machine-blog/the-time-for-self-sovereign-identity-is-now-222aab97041b
20 http://oneworldidentity.com/identity-industry-landscape/
21 https://oauth.net/2/
22 http://openid.net/
23 http://identity.foundation/
24 https://www.uport.me/
25 https://sovrin.org/
26 https://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf
27 http://presnick.people.si.umich.edu/papers/postcards/PostcardsFinalPrePub.pdf
28 https://www.ted.com/talks/rachel_botsman_the_currency_of_the_new_economy_is_trust
29 https://en.wikipedia.org/wiki/Performance_bond
30 http://mason.gmu.edu/~atabarro/PrivateProvision.pdf
31 https://en.wikipedia.org/wiki/Free-rider_problem
32 http://marginalrevolution.com/marginalrevolution/2013/08/a-test-of-dominant-assurance-contracts.html
33 https://www.cato-unbound.org/2017/06/07/alex-tabarrok/making-markets-work-better-dominant-assurance-contracts-some-other-helpful
34 https://en.wikipedia.org/wiki/Assurance_contract
35 https://ethereum.stackexchange.com/questions/2402/what-exactly-is-the-nothing-at-stake- problem

36 https://www.cryptocompare.com/coins/guides/what-are-atomic-swaps/

37 https://www.civic.com/

38 https://en.wikipedia.org/wiki/Sybil_attack

39 https://www.cs.ucsb.edu/~ravenben/publications/pdf/reputation-ecrj10, c. 246

40 https://nymity.ch/sybilhunting/pdf/Levine2006a.pdf

41 http://ualr.edu/computerscience/files/2014/01/Paper-6.pdf

42 https://pdfs.semanticscholar.org/d877/826ef2db3e7b3d955ca4b7265123be62154f.pdf, c. 2 43 https://www.cs.ucsb.edu/~ravenben/publications/pdf/reputation-ecrj10.pdf, c. 246

44 https://en.wikipedia.org/wiki/Signal_Protocol

45 https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/

46 https://eprint.iacr.org/2018/046