RIF Name Service

Спецификация

v1.31-en

Указатель

Предисловие

Одной из основных опор World Wide Web является система доменных имен (DNS). Эта система отвечает за создание сопоставления имен, удобных для чтения людьми, с IP-адресами. Корпорация по управлению доменными именами и IP-адресами (ICANN) отвечает за координацию обслуживания и процедур работы нескольких баз данных, связанных с пространствами имен и цифровыми пространствами в Интернете, которые обеспечивают работу сети. ICANN проводит фактическое техническое обслуживание реестров корневой зоны DNS.

Эти службы являются центральной точкой доверия и возможных сбоев[1][2]; их можно отключить с помощью распределённых атак «отказ в обслуживании» (DDoS), а сопоставление имен доменов можно изменить с помощью принудительного изменения DNS-серверов или подмены ответов от них. Кроме того, в этой системе есть определенные проблемы с безопасностью. Например, интернет-провайдеры (ISP) могут подвергать имена цензуре, что достаточно сложно заметить.

Поэтому одной из целей RIF Name Service (RNS) является обеспечение децентрализованной и безопасной системы, которая по своим функциям является аналогом DNS. RNS работает на основе RSK Blockchain, тем самым наследуя ее децентрализованный характер и безопасность.

Основная помеха, которая мешает широкому распространению криптовалют — это сложности при использовании адресов. Адрес Bitcoin имеет следующий вид: “06f1b66ffe49df7fce684df16c62f59dc9adbd3f”. При попытке пользователя напечатать его часто возникают ошибки. Такую длинную последовательность символов также трудно запомнить, что делает широкое распространение криптовалюты непрактичным.

В целом, RNS — это децентрализованная служба, которая позволяет пользователям получать удобные для чтения людьми домены или псевдонимы для разных ресурсов (например, адреса RSK или Swarm). Преимуществом использования удобных для чтения человеком адресов является уменьшение видимой сложности технологии блокчейн для конечного пользователя. Это упрощает ее распространение среди пользователей.

Домены и поддомены покупаются и продаются на открытом рынке. В первый раз домен продается с помощью слепого аукциона Викри [3] с использованием токенов RIF Tokens. Практика показала, что результаты аукционов определяют не только спрос и предложение, но и причуды людей. Механизм аукциона Викри снижает вероятность того, что претендент переплатит за какой-либо товар, и повышает вероятность того, что продавец получит за него хорошую цену. Если пользователь выиграл аукцион, он получает право владения доменом и с него снимается годовая арендная плата за

пользование. Мы подробно рассмотрим механизм расчета этой суммы в техническом разделе “Registrar” данного документа.

Первоначальные принципы такой процедуры могут со временем меняться по мере того, как в будущем идеи и архитектура будут постепенно обсуждаться и улучшаться экосистемой.

Экосистема RIF

RIF Token

В "RS token"-экономике основная функция RIF token заключается в том, чтобы предотвратить хранение неиспользуемых доменов или предотвратить незаконный «захват» имен. Владельцу домена нужен стимул, чтобы отказаться от владения неиспользуемым доменом. Для этого владелец домена блокирует RIF tokens, которые будут возвращены после выпуска домена. Кроме того, любые платежи или штрафы должны выплачиваться с помощью RIF Tokens, которые будут распределены в Network Resource Pool. Network Resources Pool существует для субсидирования определенных сетевых расходов, которые понадобятся на ранних этапах распространения RIF Name Service. Конечной целью является децентрализация управления фондами.

RIF Open Standard (RIFOS)

Протокол RIF Explorer, реализованный RIF Labs, представляет собой уровень абстракции, который позволяет каждой службе RIF отделиться от конкретной реализации. Те, кто занимаются реализацией услуг, становятся в рамках платформы поставщиками услуг. Это отделение позволяет каждой службе развиваться по мере появления новых технологий.

Чтобы обеспечить поддержку разных поставщиков услуг для каждой услуги на платформе RIF, RNS выполняет функцию поиска услуг. Это позволит пользователям и разработчикам узнавать о действующих поставщиках услуг и о том, как общаться с ними. Подробное описание RIF Explorer можно найти в соответствующей технической документации [4].

Введение

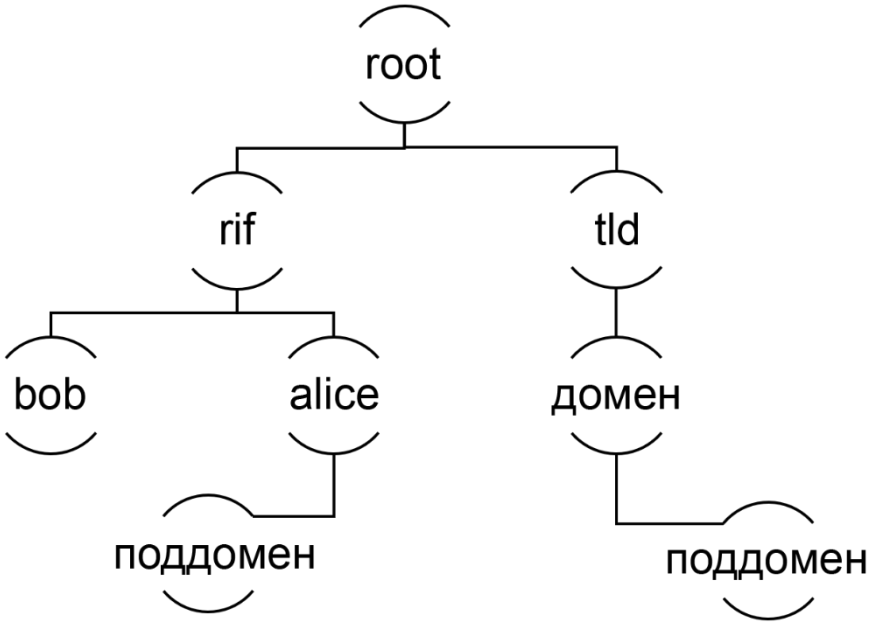
В следующем разделе представлено общее описание компонентов RNS. Подробности см. в Техническом разделе.

Обзор компонентов

Архитектура RNS основана на Ethereum Name Service (ENS), описанной в EIP-137 [5]. Она разделена на три основных компонента: Registry, Resolver и Registrar.

Node

В RNS имя верхнего уровня определяется как “*.rsk”*. Имена второго уровня называются доменами. Кроме того, домен с помощью поддоменов может ссылаться на разные ресурсы. Node является частью дерева иерархии поддоменов. Корневой путь в этом дереве — имя домена. Этот путь делится на компоненты с помощью точки ("."). Каждый node хранит хэш строки имени промежуточного компонента, вычисленный с помощью описанного выше алгоритма.



Формат имени

Формат имени RNS имеет следующий вид: “subdomain.domain.rsk”. Вкратце, имена состоят из серии меток, разделенных точками, каждая из которых является уровнем дерева поддомена. Кроме того, имя поддомена должно соответствовать правилам, описанным в разделе «Формат имени».

Resolver

Контракты Resolver отвечают за предоставление разрешения для имени ресурса. У Resolver есть много пользовательских функций, которые позволяют выбрать другой тип ресурса на том же node.

Registry

Контракт Registry обеспечивает простое сопоставление домена и его Resolver. В этом контракте регулируется все, что связано с владением доменом, включая передачу прав собственности и создание поддоменов.

Registrar

Registrar несет ответственность за управление RNS. Кроме того, он отвечает за регистрацию имен доменов для пользователей, и только он может обновлять RNS Registry. При переходе к другому Registrar (процесс миграции описан ниже) он может делегировать право собственности на поддомены другим Registrars.

Как мы уже говорили, на начальных стадиях RNS будет доступен только один top-level domain (TLD): “.rsk”. Уменьшая количество top-level domains, мы можем сосредоточиться на регистрации доменов второго уровня и отложить обсуждение пересечений доменов верхнего уровня со стандартной системой DNS и другими службами имен.

Чтобы предотвратить рассылку спама и незаконный «захват» доменов, Registrar будет напрямую управлять проведением аукционов. Поскольку платежи в RNS указаны в RIF tokens, Registrar будет взаимодействовать с ERC 677 RIF Token для осуществления платежей между счетами. Первоначально он будет обрабатывать только TLD “.rsk” и поддомены любой длины. Эта работа будет ограничена корневым узлом RNS, который контролируется контрактом, заверенным несколькими подписями. Этот контракт с множеством подписей будет контролироваться в RSK Labs в течение бета-периода.

Deed

Чтобы предотвратить ненужное коллекционирование неиспользуемых доменов или незаконный «захват» имен, владельцам доменов нужно дать стимул избавиться от ненужных доменов. Для этого владелец домена блокирует tokens, которые будут возвращены после выпуска домена.

Основные варианты использования

Регистрация домена

Существует два способа получить домен. В первом случае нужно открыть аукцион для необходимого домена через Registrar. Например, если “.rsk” является TLD, а пользователь Алиса хочет получить домен “alice.rsk,” она может открыть аукцион на этот домен, сделать ставку, и если ставка будет самой высокой, она станет новым владельцем домена “alice.rsk.” Во втором случае, если Боб является владельцем домена «bob.rsk», а Алиса хочет получить поддомен «subdomain.bob.rsk», Боб может передать право собственности на этот поддомен без проведения аукциона. После получения домена ей необходимо в контракте Registry этого домена установить Resolver, который

даст разрешение на передачу домена и ресурсов. Если пользователь не установит Resolver, будет выбран тот, что установлен по умолчанию. Этот Default resolver - Resolver родительского домена нового владельца. Затем новый владелец домена должен создать запись в Resolver для нового домена. Например, если Алиса не установила Resolver на запись в Registry “subdomain.bob.rsk”, то будет установлен Resolver “bob.rsk”.

Разрешение на пользование доменом

Разрешение на пользование доменом - это процесс, который проверяет, существует ли домен, а затем возвращает информацию, связанную с записью в Registry. Такое разрешение может использоваться в кошельках, на биржах или в dApps для обработки доменов, для которых неизвестен соответствующий адрес. Для этого сперва предположим, что существует домен “bob.rsk”. Для получения разрешения для домена «bob.rsk» пользователь должен использовать описанный выше алгоритм хэширования имени, который возвращает node (например, 0x231..de3). После этого, учитывая этот полученный node, пользователь должен найти связанную с ним запись RNS Registry. Эта запись содержит Resolver, указывающую node, который возвращает желаемый ресурс.

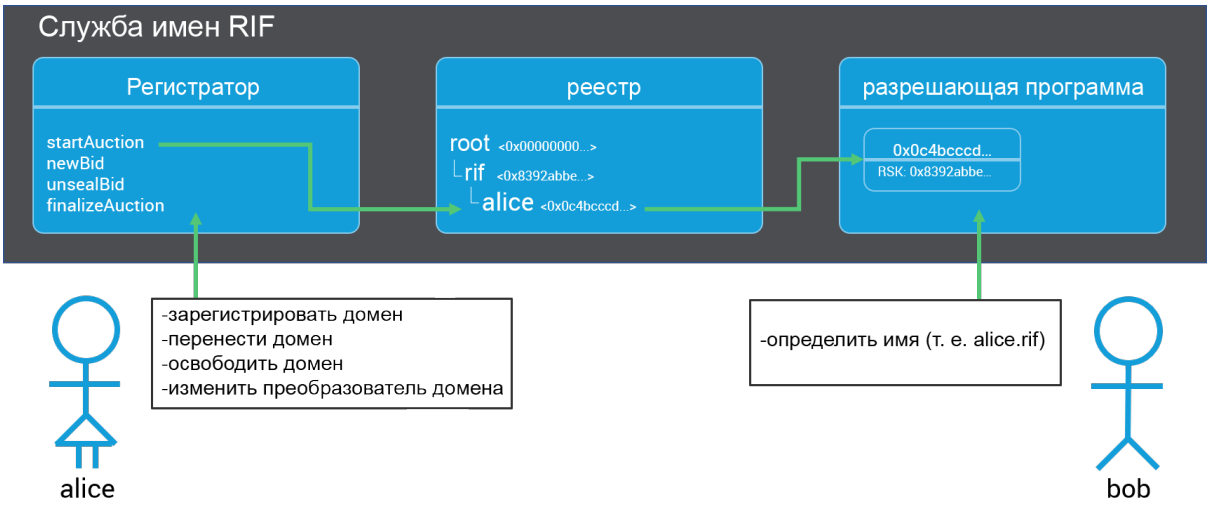
Технический обзор

Сначала RSK Labs создает контракт с Registry, который обрабатывает сопоставление между именем домена и его владельцем. Каждая запись Registry ссылается на Resolver, который проводит обработку имени домена и желаемого ресурса.

Затем используется контракт с Registrar, который управляет аукционом и предоставляет домены. Для каждой заявки Registrar создает контракт Deed и переводит сумму, указанную пользователем, на Deed. Победитель аукциона должен заблокировать баланс Deed и обменять Deed на право собственности на домен. Затем система регистрирует победителя аукциона в качестве владельца домена в Registry, и он может установить собственный Resolver.

Кроме того, владелец может делегировать поддомены с помощью контракта с Registry без проведения аукциона.

Каждый владелец имени имеет Deed для каждого имени и должен платить годовую арендную плату за каждое Deed. Плата за нахождение в Registry берется для того, чтобы предотвратить незаконный «захват» и спамминг доменов при хранении контракта Registry. Если арендная плата не выплачена, право собственности на домен аннулируется, а домен в контракте Registrar становится открытым для новых аукционов (состояние «Открыт»).



Компоненты

Registry

Рекомендации по работе и интерфейс Registry описаны в документе RNSIP01[6].

Формат имени

Имена RNS создаются по следующему принципу:

```
<domain> ::= <label> | <domain> "." <label>
<label> ::= any valid string label per [7]
```

Вкратце, имена состоят из серии меток, разделенных точками. Каждая метка должна быть допустимой унифицированной меткой, как это описано в UTS46 [7], с опцией «transitional=false», и использовать «STD3AsciiRules=true». В Javascript доступна библиотека [8], которая нормализует и проверяет имена.

Обратите внимание: поскольку в именах можно использовать буквы верхнего и нижнего регистра, процесс нормализации UTS46 сбрасывает метки регистров перед их хэшированием, поэтому у двух имен, которые отличаются только регистром, будут одинаковые хэш-имена.

Метки и домены могут иметь любую длину, но для совместимости с системой DNS рекомендуется, чтобы длина метки не превышала 64 символа, а полное имя RNS не превышало 255 символов. Поэтому рекомендуется, чтобы метки не начинались или не заканчивались дефисом, а также не начинались с цифр.

Алгоритм хэширования имен

Служба имен RNS использует алгоритм хэширования имен. Этот алгоритм рекурсивно хэширует компоненты имени, создавая уникальную строку с фиксированной длиной для любого допустимого входного домена

Вывод хэш-имени называется 'node'.

Псевдокод для алгоритма хэш-имени выглядит следующим образом:

```
def namehash(name):
  if name == '':
    return '\0' * 32
  else:
    label, _, remainder = name.partition('.')
    return sha3(namehash(remainder) + sha3(label))
```

Resolver

Resolver - это интерфейс. Пользователь может использовать контракт Public resolver, предоставляемый в RSK, или реализовывать свой собственный контракт Resolver. Если пользователь не установил свой собственный Resolver для записи Registry, то будет использован Resolver родительского домена. Затем пользователь должен зарегистрировать в Resolver родительского домена информацию разрешения между именем домена и желаемым ресурсом. Интерфейс и технические характеристики Resolver и Registry описаны в документе RNSIP01 [6].

Registrar

Интерфейс Registrar

constructor(RNS _rns, bytes32 _rootNode, uint _startDate, ERC677 tokcAddr)

- Конструктор получает RNS Registry, корневой node, которому принадлежит Registrar, а также контракт для токена ERC 677, который используются для платежей RIF.

startAuction(bytes32 _hash) public

- Статус хэша меняется на «Открыт для аукциона».

startAuctions(bytes32[] _hashes) public

- Это позволяет любому открыть аукцион для нескольких хэшей. Этот метод позволяет предотвратить ситуацию, когда злоумышленник делает ставку вслепую, чтобы победить на аукционе. В этом случае некоторые из представленных хэшей являются фиктивными, хотя создатель аукциона

заинтересован в торгах только за один из них. Это увеличивает затраты для тех, кто просто делает ставки вслепую на всех новых аукционах. Открытые фиктивные аукционы, по которым нет торгов, закрываются через неделю.

newBid(bytes32 sealedBid, uint tokenQuantity) public

- Для создания заявки необходимо отправить сообщение в основной контракт, где указан хэш sealedBid (созданный с помощью функции shaBid) и количество токенов. Хэш содержит информацию о ставке, включая предложенное хэшированное имя, сумму ставки и случайный модификатор. До открытия заявки не привязаны к какому-либо аукциону. Ценность самой ставки можно замаскировать, отправив сумму больше, чем реальная ставка. После окончания периода аукциона начинается 48-часовой период его раскрытия. Если заявка раскрыта после этого, на владельца накладывается штраф на сумму предложенных токенов. Поскольку это аукцион, мы предполагаем, что для большинства общедоступных хэшей, например, известных доменов и часто используемых слов, будет несколько участников, которые будут повышать цену. Наконец, создается Deed с несколькими токенами и контрактом, который ими управляет.

newBidWithToken(bytes32 sealedBid, uint tokenQuantity,) public

- Эквивалент newBid. Очень удобно делать ставки из контрактов ERC 677.

startAuctionsAndBid(bytes32[] hashes, bytes32 sealedBid, uint tokenQuantity) public payable

- Сервисная функция позволяет открывать startAuctions и делать newBid одновременно.

unsealBid(bytes32 _hash, uint _value, bytes32 _salt) public

- После этого начнется период раскрытия, в течение которого владельцы подтверждают права собственности на заявку. Registrar хэширует эти параметры с помощью функции shaBid(), чтобы убедиться, что они соответствуют уже существующей закрытой ставке. Если unsealedBid является новой наилучшей ставкой, старая наилучшая ставка возвращается участнику торгов.

cancelBid(address bidder, bytes32 seal) public

- Отмена нераскрытой ставки проводится в соответствии с правилами, описанными в графике возмещения.

finalizeAuction(bytes32 _hash) public onlyOwner(_hash)

- После окончания периода раскрытия необходимо использовать эту функцию для завершения аукциона. После закрытия аукциона в RNS Registry в качестве владельца будет указан автор самой высокой ставки.

transfer(bytes32 _hash, address newOwner) public onlyOwner(_hash)

- Обновление RNS Registry передает право собственности на хэш-метку новому владельцу.

releaseDeed(bytes32 _hash) public onlyOwner(_hash)

- Через девять месяцев после создания Deed владелец имени домена таким способом может отказаться от имени и вернуть часть средств, вложенных в Deed.

eraseNode(bytes32[] labels)

- Позволяет любому удалить записи владельца и Resolver на поддомен имени, на которое сейчас нет собственника в Registrar. Например, для обнуления «my.example.rsk» в Registrar, которому принадлежит «.rsk», передается массив, содержащий [sha3(‘my’), sha3(‘example’)].

transferRegistrars(bytes32 _hash)

- Если Registrar больше не является владельцем корневого node RNS, эта функция передаст Deed текущему владельцу, который должен стать новым Registrar. Эта функция определяет, владеет ли этот Registrar корневым node.

shaBid(bytes32 hash, address owner, uint value, bytes32 salt) public pure returns (bytes32)

- Хэширование значений требуется для обеспечения безопасности ставки.

payRent(bytes32 _hash) public

- Выплата годовой арендной платы за домен.

payRentWithTokens(bytes32 _hash) public

- Эквивалент payRent. Очень удобно делать ставки из контрактов ERC 677.

acceptedRegistrarTransfer(bytes32 _hash, DeedWithTokens _deed, uint _registrationDate) public pure

- Подтверждение переносов node и изменение их статуса для миграции Registrar.

tokenFallback(address _from, uint _value, bytes _data) public

- Функция, необходимая для переноса с помощью ERC 677.

Проведение аукциона

Аукцион Викри состоит из четырех составляющих:

- **Open**: состояние домена по умолчанию.
- **Auction:** Аукцион открыт. Пользователям дается 72 часа, чтобы разместить свои закрытые ставки. Закрытые ставки могут быть получены через *shaBid(bytes32 hash, address owner, uint value, bytes32 salt).*
- **Reveal:** После завершения аукциона идет 48-часовой период раскрытия. Каждый участник торгов открывает свою ставку, а Registrar обновляет аукцион

в установленном порядке. Если ставки не раскрыты, аукцион возвращается к статусу «Open».

- **Owned:** Когда период раскрытия закончится, победитель должен провести операцию, чтобы завершить раскрытие с помощью функции *finalizeAuction.* После этого аукцион завершается, и победитель становится владельцем хэша имени.

График выплат и штрафов

Результат ставки	**Описание**	**Соответствующие платежи**
Пользователь выиграл аукцион	Если пользователь выиграл аукцион, сумма второй по величине ставки будет заблокирована по Deed (TL), а разница между выигравшей ставкой и второй по величине ставкой будет возвращена. Процент от этой заблокированной суммы минус годовая сумма аренды (Y) выплачивается в качестве сбора (F) после окончания аукциона.	T2: Вторая по величине ставка Y: Стоимость аренды в год TL: Сумма, заблокированная по Deed F: Плата T = (T2 - Y) F= T*0,2 TL = T*0,8
Пользователь проиграл аукцион	Если пользователь проигрывает аукцион, потому что его ставка не самая высокая, процент от заблокированной суммы по Deed (TL) оплачивается в качестве сбора (F)	TL: Сумма, заблокированная по Deed F: Плата F = 0,05*TL

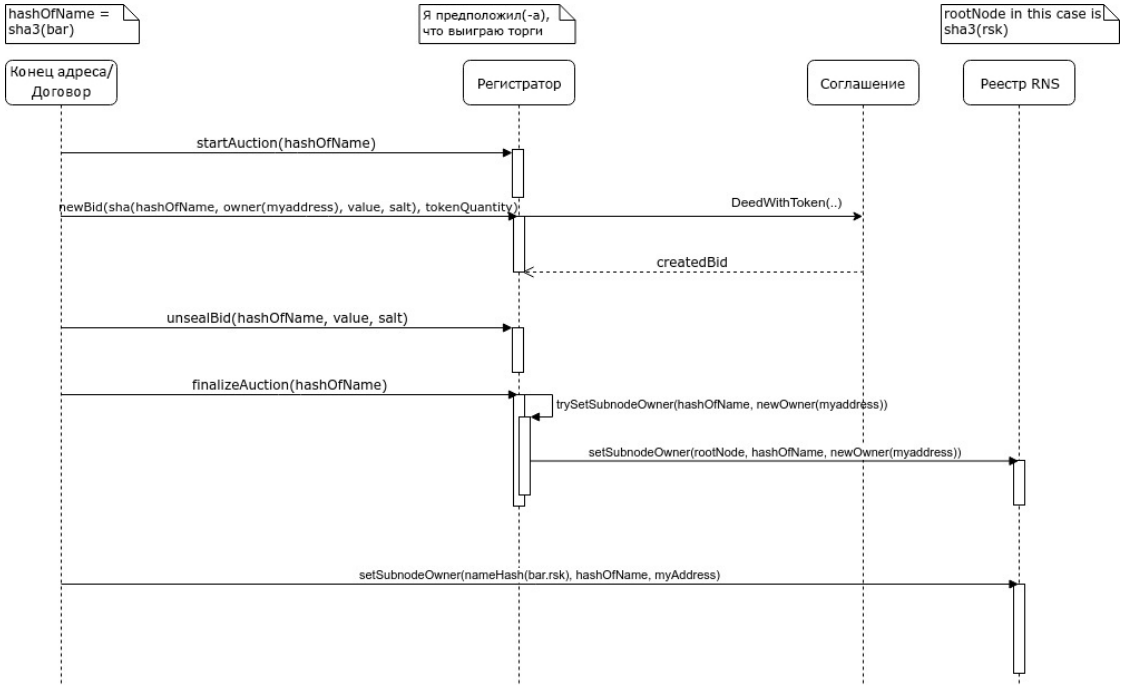
Затем идет период для раскрытия ставок вслепую (период «Reveal»). Если участник торгов отменяет ставку после начала периода «Reveal», 99,5% суммы предложенных токенов возвращаются. Затем, если участник торгов раскроет ставку до начала периода «Reveal», операция будет отменена. После этого будут определены все ставки. Пусть T — сумма выигравшей ставки, T2 — сумма второй ставки, а V — сумма ставки, которая была раскрыта после завершения периода «Reveal». В этом случае выплаты и возмещение будут проводиться следующим образом:

Условие	**Описание**	**Соответствующие платежи**
Если V > T	Если бы ставка была раскрыта вовремя, она бы победила	V*0,2 — вычитается как сбор, остальная сумма возвращается
Если T > V > T2	Если бы ставка была раскрыта вовремя, она стала бы второй по величине ставкой	V - T2 вычитается как сбор, остальная сумма возвращается
В противном случае	Ставка не была раскрыта, и она меньше второй по величине ставки	V*0,05 — вычитается как сбор, остальная сумма возвращается

После периода «Reveal» у пользователя есть 15 дней, чтобы раскрыть ставку. Если пользователь так и не раскроет ставку, все заблокированные токены по Deed будут отправлены в RNS Network Resources Pool. Это делается для того, чтобы предотвратить постоянную блокировку RIF Tokens.

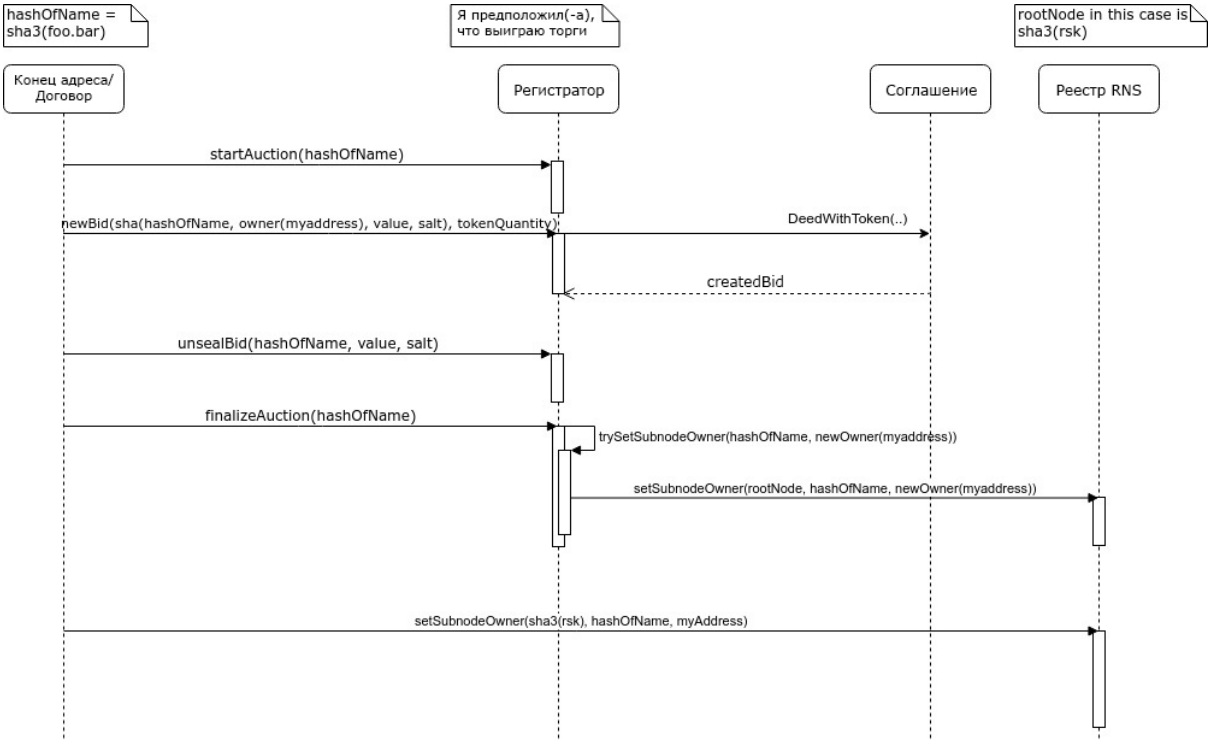
Уникальность доменов

Предположим, что Алиса получила права на домен «bob.alice.rsk» и делегирует право собственности на поддомен «bob.alice.rsk» другому пользователю, Бобу. Последовательность получения права собственности на поддомен будет следующей:



При этом злоумышленница Мэллори может открыть аукцион для домена `sha3("subdomain.bob")` в Registrar ".rsk", даже если она не является законной

владелицей “.bob.” Это возможно потому, что аукционы относятся к хэш-имени домена, а не к строке имени. Предположим, Мэллори хочет забрать себе домен «subdomain.bob.rsk», принадлежащий Бобу. Процесс будет следующим:



После этого Мэллори разместит запись в RNS Registry для sha3((sha3(‘rsk’), sha3(‘subdomain.bob’)), используя свой собственный контракт Resolver с целью перенаправления разрешения на “subdomain.bob.rsk” для другого ресурса. Но когда пользователь ищет имя домена «subdomain.bob.rsk», алгоритм хэша имени (объясненный выше) выполняет действие sha3(sha3(sha3(‘rsk’),sha3(‘bob’)), sha3(‘subdomain’)) вместо sha3(sha3(‘rsk’), sha3(‘subdomain.bob’)). Следовательно, доменное имя, разрешенное алгоритмом хэша имени, будет закреплено за Бобом.

Deed

Каждый RIF token, отправленный в Registrar, хранится в отдельном контракте под названием “Deed”. В каждом Deed хранится баланс токенов для определенного хэша имени. Deed создается при подаче ставки. Затем, после завершения аукциона и регистрации домена, Deed и ставка победителя будут заблокированы, и их обменяют на владение доменом. Остатки по Deed для проигравших ставок возвращаются их законным владельцам по запросу. Как и у Registrar, в контракте Deed указан контракт RIF Token ERC 677, который обрабатывает токены.

Владелец Deed может передать имя домена другому лицу, таким образом передавая право собственности и управления этим именем. Для этого составляется договор с Registrar.

Через девять месяцев после заключения аукциона владелец node может оплатить годовую арендную плату, продлив владение доменом еще на один год. Если владелец не хочет платить годовую арендную плату, он может отказаться от собственности и получить средства, замороженные по Deed.

Для оплаты аренды любого домена пользователь может воспользоваться функцией payRent для Registrar. Эта функция оплачивается RIF token. Если через 9 месяцев после составления Deed владелец предпочел отказаться от имени, у него есть три месяца, чтобы выполнить releaseDeed и вернуть заблокированные токены. По окончании этих трех месяцев статус аукциона для этого домена меняется на «Open», а весь баланс Deed будет перенесен в RNS Network Resource Pool.

Deeds для невыигравших ставок можно закрыть разными способами, при этом RIF tokens будут возвращены участникам торгов.

Контракт на ERC 677 Token

Токен RIF использует стандарт ERC 677. Выплата арендной платы или ставок на RNS проводится с помощью RIF tokens. Таким образом, процесс взаимодействия с контрактом ERC 677 RIF Token заключается в следующем:

- В функции `transferAndCall` присутствуют 3 параметра, в которых нужна подпись:
    - newBid: Задает параметр *data* подписи 0x1413151f, объединенный с параметром sealedBid, который создан функцией shaBid.
    - payRent: Задает параметр *data* подписи 0xe1ac9915, объединенный с параметром метки sha3, для которого выплачивается арендная плата.

Предложения по улучшению

Контракты, которые принадлежат архитектуре RNS, можно модернизировать для внедрения новых улучшений. Такие обновления опираются на отзывы сообщества и предложения RSK Labs. Все обновления RNS имеют обратную совместимость, то есть право собственности на домены сохраняется за их владельцами.

Управление поддоменом

Существует два способа получения пользователем поддомена указанного домена. Если владелец домена является Registrar, пользователь может начать аукцион для любого поддомена в этом домене. Также владелец домена может делегировать поддомен покупателю без прохождения процесса аукциона с помощью функции setSubnodeOwner. В последнем варианте у нового владельца нет стимула удалить запись в Registry поддоменов, когда он перестанет им пользоваться. Раз нет заблокированной суммы, то нет и контракта Deed. Мы стремимся создать более совершенную систему делегирования и справедливого управления поддоменами.

Структура нового Registry

Контракт Registry действует только для владельца node. Registry хранит всю информацию RNS. Масштабирование прекратится после реализации аренды хранилища. Мы ищем альтернативную структуру Registry, чтобы сделать аренду более справедливой .

Домены DNS и Oracles

Новая система позволит проводить миграцию обычных DNS-адресов в RNS. Владелец DNS-адреса сможет запросить домен в RNS с помощью Oracles, чтобы убедиться, что он является законным владельцем исходного домена. Решать конфликты, возникшие из-за совпадений с доменными именами ICANN, будет система арбитража. Она может быть реализована с помощью Oracles, а также другими методами.

Анонимность Resolver

Пользователи могут пожелать скрыть адреса своих доменов. Это можно сделать с помощью шифрования ресурсов. Владелец может передать ключ дешифрования пользователю по запросу через канал связи вне сети. Кроме того, сохраненные адреса могут быть скрытыми, поэтому отправителю придется создавать новый адрес для каждого независимого платежа.

Создание нового Top-Level Registrar

RIF предоставил первоначальный Registrar для TLD (.rsk). В будущем пользователи смогут создавать собственные TLD, развертывая свои собственные Registrars. Пользователи также смогут давать разрешение на проведение аукционов (или других процессов), чтобы другие люди могли приобретать поддомены.

Ссылки

[1] M. Ali, R. Shea, J. Nelson, M J. Freedman, "Blockstack: A New Internet for Decentralized Applications" (2017) https://blockstack.org/whitepaper.pdf
[2] "Namecoin FAQS" https://namecoin.org/docs/faq/
[3] "Vickrey Auction" https://en.wikipedia.org/wiki/Vickrey_auction
[4] "RIF Explorer" https://docs.rifos.org/rif-explorer-specification-en.pdf
[5] N. Johnson, "Ethereum Domain Name Service" (2016) https://github.com/ethereum/EIPs/blob/master/EIPS/eip-137.md
[6] J. Len, "Registry and Resolver of RNS" (2018) https://github.com/rnsdomains/RNSIPs/blob/master/IPs/RNSIP01.md
[7] M. Davis, M. Suignard "UTR46" http://unicode.org/reports/tr46/
[8] NPM Library https://www.npmjs.com/package/idna-uts46