RIF Name Service

Especificação

v1.31-en

Índice

Preâmbulo

Um dos pilares da World Wide Web é o Sistema de Nomes de Domínio (DNS). Este sistema é responsável por criar um mapeamento entre nomes legíveis por humanos e endereços IP numéricos. A Corporação da Internet para Atribuição de Nomes e Números (ICANN) é uma entidade responsável por coordenar a manutenção e os procedimentos de vários bancos de dados relacionados aos espaços de nomes e espaços numéricos da Internet, garantindo a operação da rede. A ICANN faz o trabalho de manutenção técnica real dos registros de zona raiz de DNS.

Esses serviços são o ponto central de confiança e falha[1][2]; eles podem ser colocados offline por ataques distribuídos de negação de serviço (DDoS), e os mapeamentos de domínios podem ser alterados, forçando alterações nos servidores DNS ou falsificando as respostas deles. Além disso, há algumas preocupações com a segurança, como a capacidade de provedores de serviço de Internet (ISPs) de bloquear nomes sem fácil detecção.

Um dos objetivos do RIF Name Service (RNS) é se tornar um sistema descentralizado e seguro, parecido com o DNS. O RNS funciona sobre o RSK Blockchain, herdando assim sua natureza e segurança descentralizadas.

Um obstáculo para a adoção de criptomoedas é a dificuldade de se lidar com endereços. Um endereço típico de Bitcoin, tal como "06f1b66ffe49df7fce684df16c62f59dc9adbd3f", é notoriamente suscetível a erros quando um usuário tenta digitá-lo. Uma sequência tão longa de caracteres também é difícil de memorizar, o que torna a adoção frequente de criptomoedas impraticável.

Em conclusão, o RNS é um serviço descentralizado, que oferece aos usuários um domínio ou um alias legível para humanos que se refere a diferentes recursos (por exemplo, endereços RSK ou Swarm). Uma vantagem do uso de endereços legíveis para humanos é a redução da aparente complexidade da tecnologia blockchain para o usuário final, facilitando seu uso.

Domínios e subdomínios são comprados e vendidos em um mercado aberto. O mecanismo para se obter um domínio pela primeira vez é através de um leilão de Vickrey com lances fechados [3], onde os lances são feitos com RIF Tokens. A prática mostra que traços psicológicos humanos, e não apenas a oferta e a procura, são fatores determinantes em um leilão. O mecanismo de leilão de Vickrey reduz a probabilidade de um proponente pagar um valor excessivo por um item e aumenta a probabilidade de o vendedor conseguir o máximo que ele puder. Quando um usuário vence um leilão, a propriedade do domínio é designada a ele, sendo necessário o pagamento de um aluguel anual para manter essa

propriedade. Detalhes sobre como o aluguel é calculado serão explicados na seção técnica deste documento, intitulada “Registrar”.

As diretrizes iniciais fornecidas neste protocolo podem estar sujeitas a mudanças adicionais, pois as ideias e a arquitetura serão discutidas e aprimoradas pelo ecossistema no futuro.

O Ecossistema RIF

RIF Token

Na economia de RNS tokens, a principal função do RIF token é impedir o armazenamento não utilizado, devido a domínios não utilizados, ou impedir a prática conhecida como “name squatting”. O proprietário do domínio deve ter incentivos para abrir mão da propriedade de um domínio não utilizado. Para que isso aconteça, o proprietário do domínio bloqueia os RIF tokens que serão reembolsados quando o domínio for liberado. Além disso, qualquer pagamento de tarifa ou penalização deve ser feito por meio de RIF tokens, que serão alocados ao Network Resource Pool. O Network Resource Pool existe para subsidiar determinadas despesas de rede que serão necessárias nos primeiros estágios de adoção do RIF Name Service. O objetivo final é permitir a governança descentralizada dos fundos.

RIF Open Standard (RIFOS)

O RIF Explorer, implementado pela RIF Labs, é uma camada de abstração que permite que cada Serviço RIF seja dissociado de uma implementação específica. As implementações específicas são conhecidas na plataforma como provedores de serviço. Essa desassociação permite que cada serviço evolua à medida que novos avanços tecnológicos são disponibilizados.

Para oferecer suporte a diferentes provedores de serviços para cada serviço da Plataforma RIF, o RNS é usado como um mecanismo de descoberta de serviço. Isso permitirá que usuários e desenvolvedores descubram quais provedores de serviços estão disponíveis e como se comunicar com eles. Para saber mais sobre o RIF Explorer, leia o whitepaper correspondente [4].

Introdução

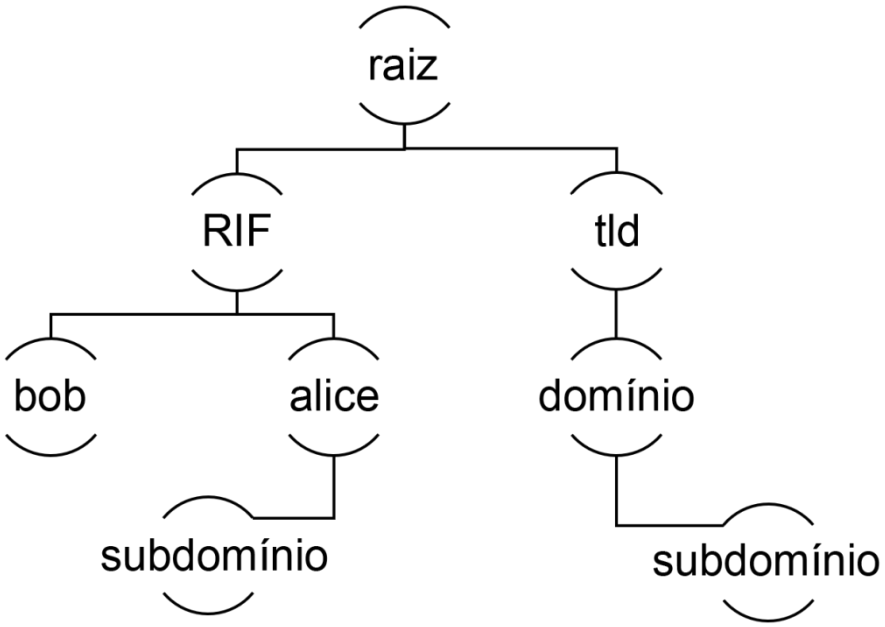
A seção a seguir fornece uma descrição geral dos componentes do RNS. Para obter mais detalhes, leia a Seção Técnica.

Visão geral dos componentes

A arquitetura do RNS baseia-se no Ethereum Name Service (ENS), descrito em EIP-137 [5]. Ela é dividida em três componentes principais: Registry, Resolver e Registrar.

Node

No RNS, o nome de nível superior é definido como "*.rsk"*. Nomes de segundo nível são chamados de domínios. Além disso, um domínio pode apontar para diferentes recursos através dos subdomínios. Um node é uma parte da árvore hierárquica de um subdomínio. Um caminho à raiz dessa árvore é um nome de domínio. Esse caminho é dividido em componentes por um ponto ("."). Cada node armazena um hash (criptografia) da sequência de nomes de um componente intermediário, computado pelo algoritmo de hash de nome explicado acima.



Formato do Nome

O formato do nome RNS deve seguir a seguinte sintaxe: "subdomain.domain.rsk". Em suma, o nome consiste em uma série de rótulos separados por pontos, cada qual consistindo em um nível de árvore de subdomínio. Além disso, o nome do subdomínio deve seguir as regras explicadas na seção técnica Formato do Nome.

Resolver

Os contratos Resolver são responsáveis pela resolução do nome de um recurso. Um resolver tem diversas funções definidas pelo usuário, sendo que cada função permite que um tipo diferente de recurso seja obtido no mesmo node.

Registry

O contrato Registry oferece um mapeamento simples entre um domínio e seu resolver. Tudo relacionado à propriedade de domínio é controlado neste contrato, incluindo a transferência de propriedade e a criação de subdomínios.

Registrar

O Registrar é responsável pela governança do RNS. Além disso, ele é responsável por registrar o nome de um domínio para um usuário e é a única entidade capaz de atualizar o RNS Registry. Caso o domínio seja migrado para outro registrar (o processo de migração é explicado abaixo), ele pode delegar a propriedade de subdomínios a outros registrars.

Conforme explicado anteriormente, apenas um top-level domain (TLD) estará disponível nas fases iniciais do RSK: ".rsk". Ao reduzir o número de top-level domains, podemos nos concentrar no registro de domínios de segundo nível e adiar a discussão sobre sobreposições de nível superior com o sistema DNS padrão e outros serviços de nomes.

Para evitarmos o squatting ou spamming dos domínios, o Registrar controlará o processo de leilão. Como os pagamentos RNS são denominados em RIF tokens, o Registrar interage com o contrato do RIF Token ERC 677 para efetuar os pagamentos entre as contas. Inicialmente, ele controlará apenas o TLD ".rsk" e subdomínios com qualquer número de caracteres. Isso será restringido pelo node raiz do RNS, que é controlado por um contrato de múltiplas assinaturas. Esse contrato de múltiplas assinaturas será inicialmente controlado pela RSK Labs. durante o período beta.

Deed

Para impedir o armazenamento desnecessário devido a domínios não utilizados ou impedir o name squatting, o proprietário do domínio deve ter incentivos para abrir mão da propriedade desses domínios. Para que isso aconteça, o proprietário do domínio bloqueia os tokens que serão reembolsados quando o domínio for liberado.

Principais casos de uso

Registrar um domínio

Há duas formas para os usuários adquirirem um domínio. A primeira é abrir um leilão através do contrato do registrar para o domínio desejado. Por exemplo, se ".rsk" for o TLD e uma usuária Alice quiser o domínio "alice.rsk", ela poderá abrir um leilão para esse domínio, fazer uma oferta e, se for a mais alta, ela se tornará a nova proprietária de "alice.rsk". Uma segunda maneira é, se Bob for o proprietário de "bob.rsk" e Alice quiser o subdomínio "subdomain.bob.rsk", Bob pode delegar a propriedade do subdomínio a Alice sem um processo de leilão. Quando um usuário adquirir um domínio, ele deve definir no contrato de registry do domínio o resolver que será responsável pela resolução entre o novo domínio e o recurso desejado. Se um usuário não definir um resolver, será definido um default resolver. Esse default resolver será o resolver do pai do novo domínio adquirido. O novo proprietário

do domínio deverá então criar uma entrada de resolver com o novo domínio. Por exemplo, se Alice não definir um resolver na entrada de registry “subdomain.bob.rsk”, o resolver “bob.rsk” será definido.

Resolução de domínio

Resolução de domínio é um processo que verifica a existência de um domínio e, em seguida, retorna informações associadas à sua entrada de registry. Essa resolução pode ser usada em wallets, exchanges ou dApps, para lidar com domínios sem conhecimento sobre seus respectivos endereços. Para fazer isso, primeiro vamos supor que o domínio “bob.rsk” exista. Para resolver o domínio “bob.rsk”, um usuário deve usar o algoritmo de hash de nome explicado abaixo, que retorna um node (ex. 0x231..de3). Após isso, dado esse node, o usuário deve buscar a entrada de RNS Registry associada a ele. Essa entrada contém o resolver que, dado o node, retorna o recurso desejado.

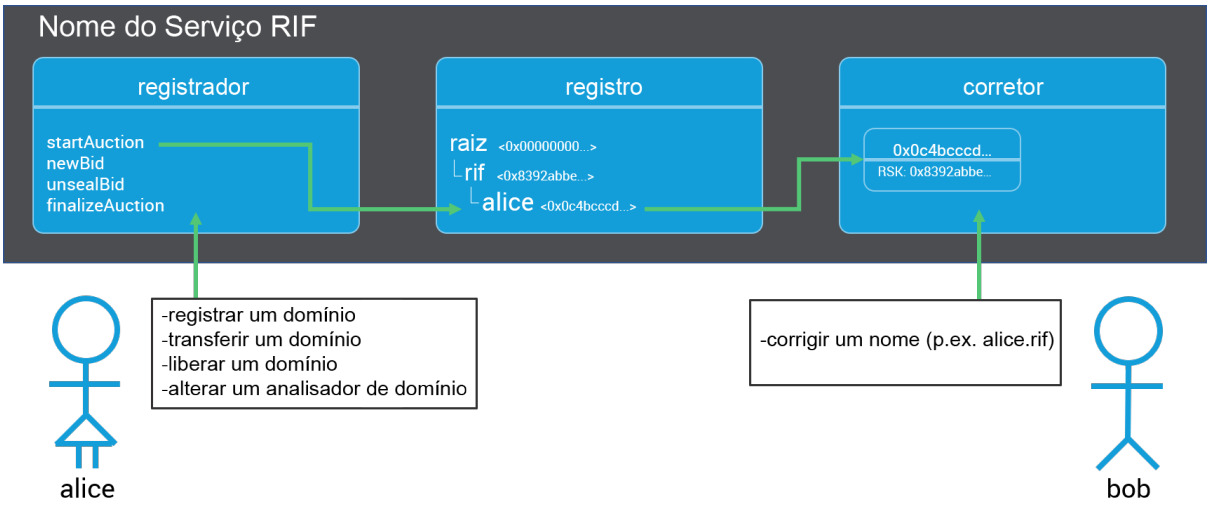
Visão Geral Técnica

A RSK Labs emprega um contrato registry que realiza o mapeamento entre um nome de domínio e seu proprietário. Cada entrada de registry aponta para um resolver, que é responsável pela resolução entre o nome de domínio e o recurso desejado.

Em seguida, o contrato de registrar que cuida do leilão e do fornecimento de domínios é implementado. Para cada lance, o registrar cria um contrato de deed e transfere o valor do lance do usuário para o deed. O vencedor do leilão deve bloquear o saldo do deed e trocar o deed pela propriedade do domínio. Na sequência, o sistema registra o vencedor do leilão no registry como proprietário do domínio e o vencedor pode então criar seu próprio resolver.

Além disso, o proprietário pode delegar subdomínios usando o contrato de registry sem um processo de leilão.

Todo proprietário de nome possui um deed para cada nome e deve pagar um aluguel anual para cada deed. A razão da cobrança dos aluguéis de registry é evitar o squatting e o spamming dos domínios no armazenamento do contrato de registry. Se o aluguel não for pago, a propriedade do domínio será cancelada e o estado do domínio no contrato do registrar ficará aberto a novos leilões (o estado aberto).



Componentes

Registry

As diretrizes e interface do Registry estão descritas no RNSIP01[6].

Formato do nome

Os nomes RNS devem seguir a seguinte sintaxe:

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

Resumindo, os nomes consistem em uma série de rótulos separados por pontos. Cada rótulo deve ser um rótulo normalizado e válido, conforme descrito em UTS46 [7] com as opções transitional=false e use STD3AsciiRules=true. Para implementações JavaScript, está disponível uma biblioteca [8], que normaliza e verifica nomes.

Observe que, embora letras maiúsculas e minúsculas sejam permitidas nos nomes, a normalização UTS46 processa rótulos com caixas desdobradas antes de elas serem codificadas, portanto, dois nomes com caixas diferentes, mas com grafia idêntica, produzirão o mesmo hash de nome.

Os rótulos e domínios podem ser de qualquer tamanho, mas, para fins de compatibilidade com o DNS legado, recomenda-se que os rótulos sejam restritos a não mais que 64 caracteres cada e que os nomes RNS completos não excedam 255 caracteres. Pelo mesmo motivo, recomenda-se que os rótulos não comecem ou terminem com hifens e que não comecem com dígitos.

Algoritmo de hash de nome

O RNS usa o algoritmo de hash de nome. Esse algoritmo recursivamente criptografa os componentes do nome, produzindo uma sequência única, de comprimento fixo, para qualquer domínio de entrada válido

O resultado de um hash de nome é chamado de 'node'.

O pseudocódigo para o algoritmo de hash de nome é o seguinte:

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

Resolver

O resolver é uma interface. Um usuário pode usar o contrato de public resolver fornecido pela RSK ou implementar seu próprio contrato de resolver. Se um usuário não definir seu próprio resolver para sua entrada de registry, o resolver de domínio pai será usado. Em seguida, o usuário deve registrar no resolver do domínio as informações de resolução entre o nome do domínio e o recurso desejado. Assim como com o registry, a interface e a especificação do resolver estão descritas no RNSIP01 [6].

Registrar

Interface do Registrar

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

- O construtor recebe um RNS Registry, um node raiz ao qual o registrar pertence, assim como um contrato ERC 677 token para usar em pagamentos RIF.

startAuction(bytes32 _hash) public

- Altera o estado de um hash de aberta para leilão.

startAuctions(bytes32[] _hashes) public

- Permite que qualquer pessoa inicie um leilão para um certo número de hashes. Este método pode ser usado para evitar que um invasor dê lances aleatórios em leilões. Neste caso, alguns dos hashes enviados são hashes falsos, embora o remetente esteja interessado em dar lances em um apenas. Isso aumentará o custo para um invasor que deseja simplesmente dar lances aleatórios em todos os novos leilões. Leilões falsos abertos e sem lances serão fechados após uma semana.

newBid(bytes32 sealedBid, uint tokenQuantity) public

- As licitações são criadas enviando uma mensagem para o contrato principal com um sealedBid hash (criada com a função shaBid) e alguns tokens. O hash contém informações sobre a licitação, incluindo o hash de nome leiloado, seu valor e um sal (de criptografia) aleatório. Os lances não são associados a nenhum leilão até serem revelados. O valor do lance em si pode ser mascarado, por meio de um valor superior ao valor real do lance. Após o término do período do leilão, segue-se um período de 48 horas para a abertura dos lances. Se um lance for aberto após esse período, ele pode ser penalizado com os tokens oferecidos. Como isso é um leilão, espera-se que a maioria dos hashes, como domínios comuns e palavras de dicionário comuns, recebam diversos lances, o que eleva o preço. Por fim, é criado um deed com um número de tokens e um contrato para gerenciá-los.

newBidWithToken(bytes32 sealedBid, uint tokenQuantity,) public

- Equivalente a newBid. É útil para chamadas de contratos ERC 677.

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

- Uma função de utilidade permite que seja feita uma chamada para startAuctions, seguida de um newBid, em uma única transação.

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

- Após o término do período de ofertas, há um período de abertura dos lances, durante o qual é enviada a prova de propriedade de um lance. O registrar criptografa esses parâmetros usando a função shaBid() para verificar a correspondência a um lance lacrado pré-existente. Se o unsealedBid agora for o novo melhor lance, o antigo melhor lance será devolvido ao lançador.

cancelBid(address bidder, bytes32 seal) public

- Cancela um lance não aberto de acordo com as regras descritas na tabela de devolução abaixo.

finalizeAuction(bytes32 _hash) public onlyOwner(_hash)

- Quando o período de abertura dos lances se encerrar, essa função pode ser usada para finalizar o leilão. Após o leilão ser fechado, o RNS Registry é atualizado com o maior lançador como o novo proprietário do nome leiloado.

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

- Atualiza o RNS Registry, transferindo a propriedade de um hash de rótulo a um novo proprietário.

releaseDeed(bytes32 _hash) public onlyOwner(_hash)

- Nove meses após a criação do deed, o proprietário de um nome pode chamar esse método para falsificar o nome e ter uma parte de seus recursos do deed devolvida a ele.

eraseNode(bytes32[] labels)

- Permite que qualquer pessoa exclua os registros do proprietário e do resolver para um subdomínio de um nome que não pertença atualmente ao registrar. Por exemplo, para zerar my.example.rsk em um registrar proprietário de ".rsk", passe uma série contendo [sha3('my'), sha3('example')].

transferRegistrars(bytes32 _hash)

- Se esse registrar não for mais o proprietário do node raiz no RNS, essa função transferirá o deed para o proprietário atual, que deve ser um novo registrar. Essa função emite um erro se o registrar ainda for o proprietário de seu node raiz.

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

- Criptografa os valores exigidos para um lance secreto.

payRent(bytes32 _hash) public

- Paga o aluguel anual de um domínio.

payRentWithTokens(bytes32 _hash) public

- Equivalente a payRent. É útil para chamadas de contratos ERC 677.

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

- Aceita transferências de node e altera seu estado para uma migração de registrar.

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

- Função necessária para fazer transferências com ERC 677.

Processo do leilão

O leilão de Vickrey é um processo de quatro etapas:

- **Open**: estado padrão do domínio.
- **Auction:** Leilão iniciado. Há um período de 72 horas em que os usuários podem enviar seus lances secretos. Os lances secretos podem ser obtidos através de *shaBid(bytes32 hash, address owner, uint value, bytes32 salt).*
- **Reveal:** Após o leilão, há um período de abertura dos lances de 48 horas. Cada lançador revela seu lance, e o registrar atualiza o leilão. Se nenhum lance for revelado, o estado retorna a aberto.
- **Owned:** Quando o período de abertura dos lances termina, o vencedor deve enviar uma transação para finalizar o período de abertura através do método *finalizeAuction*. Isso finaliza o leilão e registra o vencedor como o proprietário do hash de nome leiloado.

Esquema de reembolso e penalização

Resultado do Leilão	**Descrição**	**Pagamentos envolvidos**
Usuário vence o leilão	Se um usuário vencer o leilão, o segundo valor de lance mais alto será bloqueado em seu deed (TL), e a diferença entre esse valor e o maior valor de lance será reembolsado. Uma porcentagem desse valor bloqueado, menos o valor do aluguel (Y), será paga como uma tarifa (F) após o término do leilão.	T2: Segundo valor mais alto Y: Preço do aluguel anual TL: Valor bloqueado no Deed F: Tarifa T = (T2 - Y) F= T*0,2 TL = T*0,8
Usuário perde o leilão	Se um usuário perder o leilão por não ter enviado o lance mais alto, uma porcentagem do valor bloqueado (TL) em seu deed será pago a título de tarifa (F)	TL: Valor bloqueado no Deed F: Tarifa F = 0,05*TL

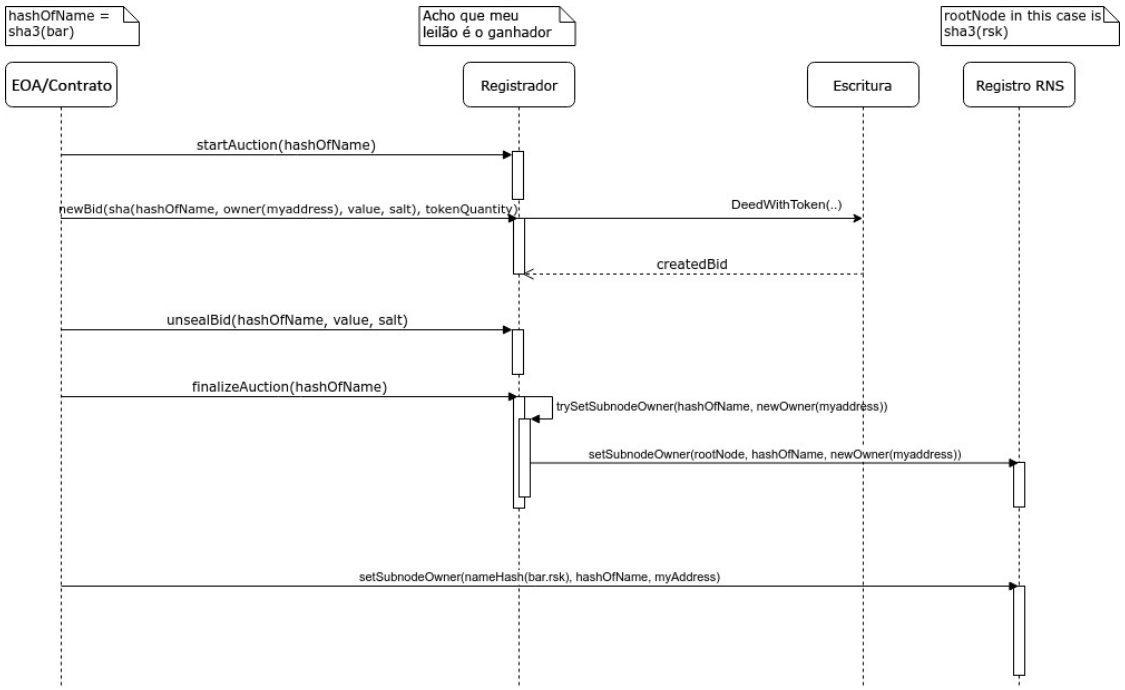
Um período para revelar os lances secretos (período de abertura dos lances). Se um lançador cancelar o lance antes do início do período de abertura, 99,5% dos tokens de lance serão reembolsados. Depois, se o licitante revelar o lance antes do início do período de abertura, a transação será revertida. Quando esse período se encerra, cada lance é liquidado. Digamos que T seja o valor do lance vencedor, T2 seja o segundo maior valor de lance e V seja um lance revelado após o término do período de abertura; os pagamentos e reembolsos serão regidos pelas seguintes condições:

Condição	**Descrição**	**Pagamentos envolvidos**
Se V > T	Se fosse revelado a tempo, ele venceria	V*0,2 será pago a título de tarifa e o restante reembolsado
Se T > V > T2	Se fosse revelado a tempo, seria o segundo valor mais alto	V - T2 será pago a título de tarifa e o restante reembolsado
De outro modo	Não foi revelado e é menor que o segundo valor mais alto	V*0,05 será pago a título de tarifa e o restante reembolsado

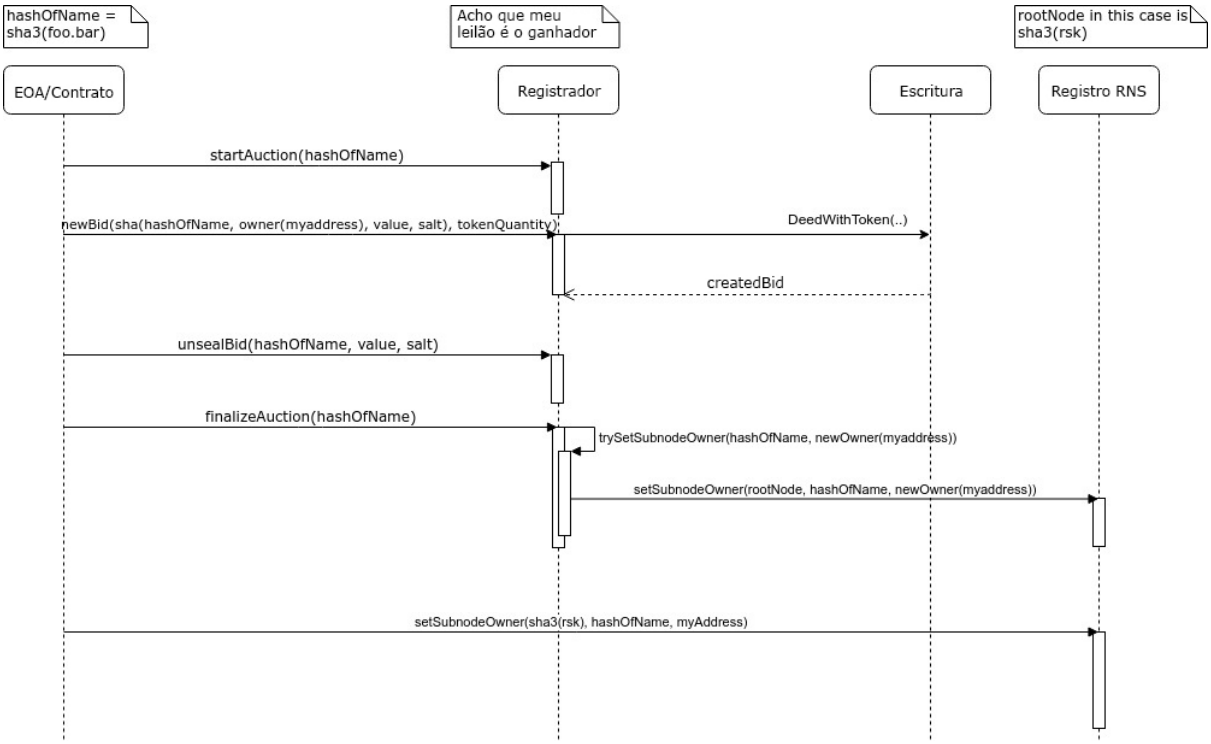
Após o término do período de abertura, o usuário tem 15 dias para revelar o lance. Se o usuário ainda não revelar o lance, todos os tokens bloqueados no deed serão enviados para o RNS Network Resource Pool; isso é feito para evitar que RIF Tokens sejam bloqueados para sempre.

Singularidade de domínios

Supondo que Alice adquira a propriedade do domínio “bob.alice.rsk” e delegue a propriedade do subdomínio “bob.alice.rsk” para outro usuário Bob. A sequência para adquirir a propriedade de um subdomínio seria a seguinte:



Essa interação pode sugerir que um usuário mal-intencionado, Mallory, pode iniciar um leilão para o domínio `sha3(“subdomain.bob”)` no registrar “.rsk”, mesmo que ele não seja o legítimo proprietário de “.bob”. Isso acontece porque os leilões são para o hash de nome de um domínio e não para a sequência de um nome. Vamos supor que Mallory queira ocupar o domínio “subdomain.bob.rsk” de Bob. O processo seria o seguinte:



Posteriormente, Mallory definirá uma entrada no RNS Registry para sha3((sha3(‘rsk’), sha3(‘subdomain.bob’)) usando seu próprio contrato de resolver com a intenção de redirecionar a resolução de “subdomain.bob.rsk” para um recurso diferente. Mas, quando um usuário procura o nome de domínio ‘subdomain.bob.rsk’, o algoritmo de hash de nome (explicado acima) resolverá sha3(sha3(sha3(‘rsk’),sha3(‘bob’)), sha3(‘subdomain’)), em vez de sha3(sha3(‘rsk’), sha3(‘subdomain.bob’)). Consequentemente, o nome de domínio resolvido pelo algoritmo de hash de nome será aquele definido por Bob.

Deed

Cada RIF token enviado ao registrar é armazenado em um contrato separado, chamado de “deed”. Cada deed armazena o saldo de tokens de um hash de nome específico. O deed é criado quando um lance é enviado. Em seguida, com o leilão for concluído e o domínio registrado, o deed vencedor e o lance serão bloqueados e trocados pela propriedade do domínio. Os saldos dos deeds pertencentes aos lances perdedores são reembolsados para seus devidos proprietários mediante solicitação. Como no registrar, um contrato de deed conhece o contrato RIF Token ERC 677 que é responsável pelos pagamentos com tokens.

Um deed de um nome que tenha um proprietário pode ser transferido para outra conta pelo seu proprietário, transferindo, assim, a propriedade e o controle do nome. Isso é feito pelo contrato do registrar.

Nove meses após a conclusão de um leilão, o proprietário do node terá a opção de pagar um aluguel anual, renovando a propriedade do domínio por mais um ano. Se o proprietário não quiser pagar mais um aluguel anual, o proprietário pode optar por renunciar à propriedade e receber o reembolso dos fundos congelados no deed.

Para pagar o aluguel de qualquer domínio, um usuário pode usar a função payRent do registrar. A função exige um pagamento por RIF token. Se o proprietário optar por renunciar à propriedade, nove meses após a criação do deed, ele terá três meses para chamar a releaseDeed e receber o reembolso dos tokens bloqueados. Após esse período de três meses, o estado do leilão desse domínio mudará para aberto e o saldo total do deed será transferido para o RNS Network Resource Pool.

Os deeds para lances não vencedores podem ser fechados por diversos métodos, quando todos os RIF tokens retidos serão devolvidos ao lançador.

Contrato ERC 677 Token

O RIF Token é implementado com a Norma ERC 677. Os pagamentos para aluguéis anuais ou lances no RNS são feitos com RIF tokens. Portanto, o processo para interagir com o contrato RIF Token ERC 677 é o seguinte:

- Na função `transferAndCall` com 3 parâmetros, a assinatura deve ser:
    - newBid: Ajustar parâmetro *data* na assinatura 0x1413151f, concatenada com o sealedBid criado pela função shaBid.
    - payRent: Ajustar parâmetro *data* na assinatura 0xe1ac9915, concatenada com o sha3 do rótulo para o qual o aluguel é pago.

Propostas de melhoria

Os contratos que pertencem à arquitetura RNS podem ser melhorados para introduzir novas melhorias. Essas atualizações são fornecidas com base no feedback da comunidade e nas propostas da RIF Labs. Cada melhoria RNS tem retrocompatibilidade, ou seja, os domínios permanecem com seus proprietários.

Gestão de subdomínios

Há duas formas de um usuário adquirir um subdomínio de um domínio específico. Se o proprietário do domínio for um registrar, o usuário pode iniciar um leilão de qualquer subdomínio daquele domínio. Além disso, o proprietário do domínio pode delegar um subdomínio ao comprador sem passar por um processo de leilão, usando a função setSubnodeOwner. Essa última opção não gera incentivo para o novo proprietário excluir a entrada do registry do subdomínio quando este não estiver mais em uso, pois não há nenhum valor bloqueado, já que não há contrato de deed. Estamos trabalhando para desenvolver um sistema aprimorado de delegação, assim como um procedimento de gestão de subdomínios justo.

Nova estrutura do Registry

O contrato de registry é o único válido para propriedade de nodes. O registro armazena todas as informações do RNS. Isso não será dimensionado depois que o aluguel de armazenamento for implementado. Estamos pesquisando uma estrutura de registry alternativa para definir um aluguel de armazenamento mais justo.

Domínios DNS e oráculos

Haverá também a possibilidade de migrar endereços comuns DNS para o RNS. Um dono de endereço de DNS poderá reivindicar um domínio no RNS por meio de oráculos para verificar se ele é o legítimo proprietário do domínio original. No caso de colisão com o domínio de nome ICANN, um sistema de arbitragem será usado para resolver conflitos. Esse sistema de arbitragem pode ser implementado através de oráculos, assim como outros métodos.

Anonimato do Resolver

Os usuários podem querer esconder o endereço apontado por seu domínio. Isso pode ser feito com recursos de criptografia. O proprietário pode transferir a chave de descriptografia a um usuário mediante solicitação por meio de um canal de comunicação externo. Além disso, os endereços armazenados podem ser invisíveis, de modo que o remetente deve gerar um novo endereço para cada pagamento independente.

Criando um novo Top-Level Registrar

A RSK Labs forneceu um registrar inicial para TLD (.rsk). No futuro, os usuários poderão criar seus próprios TLD usando seus próprios registrars. O usuário pode ativar um processo de leilão (ou um processo diferente) para permitir que as pessoas adquiram subdomínios.

Referências

[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