RIF Name Service

사양

v1.31-한국어

목차

서문

월드 와이드 웹의 핵심 요소 중 하나는 도메인 네임 시스템(DNS) 입니다. 이 시스템은 사람이 읽을 수 있는 네임과 숫자 IP 주소를 대응시켜 주는 역할을 합니다. 국제인터넷주소관리기구(ICANN)는 네트워크 작동을 보장하기 위해 인터넷의 네임 공간 및 숫자 공간과 관련된 여러 데이터베이스의 유지 관리와 절차를 조정하는 단체입니다. ICANN은 DNS 루트 존 레지스트리의 실제 기술적 유지 관리 작업을 수행합니다.

이러한 서비스는 디도스(DDoS) 공격을 받아 오프라인 상태가 될 수 있으며, DNS 서버에 억지로 변화를 주거나 이 서비스의 회신인 것처럼 위장하면 도메인 매핑이 변경될 수 있으므로 신뢰와 실패를 좌우하는 중요한 부분입니다[1][2]. 또한 쉽게 탐지할 수 없이 네임을 삭제할 수 있는 인터넷 서비스 공급자(ISP)와 같은 몇 가지 보안 우려도 존재합니다.

RIF Name Service(RNS)의 목표 중 하나는 분권형의 안전한 DNS 같은 시스템이 되는 것입니다. RNS는 RSK 블록체인 위에서 작동하므로, RSK 블록체인의 분권화된 성격과 보안을 물려받습니다.

암호화폐 채택을 방해하는 장애물은 주소를 다루는 것이 어렵다는 점입니다. 비트코인 주소의 특징은 다음과 같습니다. "06f1b66ffe49df7fce684df16c62f59dc9adbd3f", 이는 사용자가 입력할 때 오류가 일어나기 쉬운 구조입니다. 이렇게 긴 문자열은 기억하기도 어려우며, 암호화폐 채택을 비현실적으로 만듭니다.

결국 RNS는 사용자에게 사람이 읽을 수 있는 도메인 네임이나 다른 리소스(예: RSK 또는 Swarm 주소)를 가리키는 별칭을 제공하는 분권화된 서비스입니다. 사람이 읽을 수 있는 주소를 사용하면 최종 사용자가 블록체인의 복잡한 기술적 내용을 알 필요 없이 쉽게 사용할 수 있다는 이점이 있습니다.

도메인과 서브도메인은 공개된 시장에서 매매됩니다. 처음 도메인을 획득하려면 RIF 토큰으로 비공개 비커리 경매[3]를 통해 구입합니다. 실제 사례를 보면 단지 수요와 공급의 원칙이 아닌 사람의 심리적 경향에 따라 경매가 진행되는 것으로 나타났습니다. 비커리 경매 방식은 입찰자가 어떤 물품에 대해 과도하게 지불할 가능성을 줄일 뿐만 아니라 판매자가 대가로 얻을 수 있는 최고의 보상을 받을 가능성도 높여 줍니다. 사용자가 경매에 낙찰되면 해당 도메인의 소유권이 낙찰자에게 넘어가며, 이 소유권을 유지하려면 연간 임대료를 지불합니다. 임대료 계산 방법의 자세한 내용은 이 문서의 "등록기관" 기술 섹션에서 설명합니다.

향후 이 생태계는 개념과 아키텍처에 대해 논의하고 개선할 것이므로 이 프로토콜에 제공되는 초기 지침은 변경될 수 있습니다.

RIF 생태계

RIF 토큰

RNS 토큰 경제에서 RIF 토큰의 주된 기능은 미사용 도메인으로 인한 미사용 저장소나 네임 스쿼팅을 방지하는 것입니다. 도메인 소유자에게는 미사용 도메인의 소유권을 포기하는 것에 대한 혜택이 있어야 합니다. 이를 위해 도메인 소유자는 RIF 토큰을 잠그고 도메인이 해제될 때 환불을 받습니다. 또한 모든 요금이나 벌금은 RIF 토큰을 사용하여 지불해야 하며, 이 토큰은 네트워크 리소스 풀(Network Resource Pool)에 할당됩니다. 네트워크 리소스 풀은 RIF Name Service의 초기 채택에 필요한 특정 네트워크 비용의 보조금을 지급하기 위해 존재합니다. 최종 목표는 자금의 분권화된 관리를 촉진하는 것입니다.

RIF 공개 표준(RIFOS)

RIF Labs가 구현한 RIF Explorer는 각 RIF 서비스가 특정 구현에 구속을 받지 않도록 하는 추상화 계층입니다. 이러한 특정 구현은 플랫폼에 서비스 공급자로 알려집니다. 이렇게 분리함으로써 새로운 기술 향상이 실현됨에 따라 각 서비스가 진화할 수 있습니다.

RIF 플랫폼의 각 서비스에 대해 서로 다른 서비스 공급자를 지원하기 위해 RNS를 서비스 검색 메커니즘으로 사용합니다. 이렇게 하면 사용자와 개발자가 이용할 수 있는 서비스 공급자가 누구인지, 그리고 해당 공급자와 소통하는 방법을 찾을 수 있습니다. RIF Explorer에 대한 자세한 내용은 해당 백서를 참조하십시오 [4].

소개

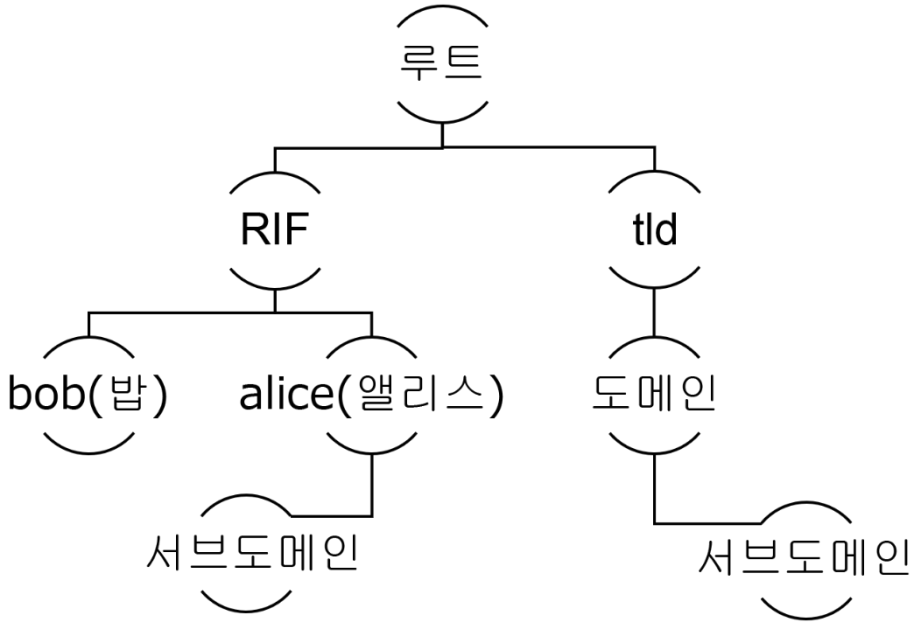
다음 섹션은 RNS 구성 요소의 일반적인 설명을 제공합니다. 자세한 내용은 기술 섹션을 참조하십시오.

구성 요소 개요

RNS 아키텍처는 EIP-137[5]에 기술된 Ethereum Name Service(ENS)에 기반하며, 세 가지 주요 구성 요소 Registry, Resolver, Registrar로 나누어집니다.

노드

RNS에서 최상위 네임은 "*.rsk*"로 정의됩니다. 2단계 네임은 도메인이라 합니다. 또한 도메인은 서브도메인을 통해 서로 다른 리소스를 참조할 수 있습니다. 노드는 서브도메인 계층 트리의 일부입니다. 이 트리의 루트에 대한 경로가 도메인 네임입니다. 이 경로는 점(".")에 의해 서로 다른 구성 요소로 분할됩니다. 각 노드는 중간 구성 요소의 네임 문자열에 대한 해시를 저장하며, 이 해시는 위에서 설명한 네임 해시 알고리즘에 의해 계산됩니다.



네임 형식

RNS 네임 형식은 "서브도메인.도메인.rsk" 구문을 준수해야 합니다. 즉, 네임은 일련의 점으로 구분된 레이블(각각 서브도메인 트리 단계로 구성)로 구성합니다. 또한 서브도메인 네임은 네임 형식 기술 섹션에서 설명한 규칙을 따라야 합니다.

Resolver

Resolver 컨트랙트는 리소스 네임을 확인하는 역할을 합니다. Resolver에는 여러 가지 사용자 정의 기능이 있으며 각 기능을 사용하여 동일한 노드에서 서로 다른 리소스 유형을 가져올 수 있습니다.

Registry

Registry 컨트랙트는 도메인과 해당 Resolver 사이의 간단한 매핑을 제공합니다. 소유권 이동과 서브도메인 생성을 포함하여 도메인 소유권과 관련된 모든 사항은 이 컨트랙트 내에서 관리됩니다.

Registrar

Registrar는 RNS 관리를 책임집니다. 또한 사용자의 도메인 네임의 등록을 담당하며, RNS Registry를 업데이트할 수 있는 유일한 주체입니다. 도메인이 다른 Registrar로 마이그레이션되는 경우(마이그레이션 프로세스는 아래에서 설명), 서브도메인의 소유권을 다른 Registrar에 위임할 수 있습니다.

앞서 설명했듯이, RNS의 초기 단계에는 단 하나의 최상위 도메인(TLD) ".rif"만 사용할 수 있습니다. 최상위 도메인 수를 줄임으로써 2단계 도메인의 등록에 집중할 수 있으며, 최상위 도메인과 표준 DNS 시스템 및 다른 네임 서비스 간의 중복에 대한 논의를 미룰 수 있습니다.

도메인 스팸이나 스쿼팅을 방지하기 위해 경매 과정은 Registrar가 관리합니다. RNS 결제는 RIF 토큰으로 이루어지므로, Registrar는 ERC 677 RIF 토큰 컨트랙트와 상호 작용하여 계정 간의 결제를 실행합니다. 초기에는 ".rsk" TLD만 처리하며, 서브도메인은 어떤 문자 길이라도 상관없습니다. 다만, 다중 서명 컨트랙트로부터 제어되는 RNS 루트 노드의 제한을 받습니다. 이 다중 서명 컨트랙트는 초기 베타 기간 동안에는 RSK Labs에 의해 제어됩니다.

Deed

비사용 도메인으로 인한 불필요한 저장소 사용을 방지하거나 네임 스쿼팅을 방지하기 위해 도메인 소유자는 자신의 소유권을 포기하는 데 대한 혜택을 받아야 합니다. 이를 위해 도메인 소유자는 토큰을 잠그고 도메인이 해제될 때 환불을 받습니다.

주요 사용 사례

도메인 신청하기

사용자가 도메인을 얻을 수 있는 방법에는 두 가지가 있습니다. 첫 번째는 원하는 도메인에 해당하는 Registrar 컨트랙트를 통해 경매를 시작하는 것입니다. 예를 들어, TLD가 ".rsk"이고 앨리스라는 사용자가 "alice.rsk"라는 도메인을 얻으려면, 앨리스는 이 도메인에 대한 경매를 시작하고, 입찰을 한 뒤, 자신의 입찰가가 가장 높으면 "alice.rsk" 도메인의 새로운 소유자가 됩니다. 두 번째 방법은, 밥이라는 사용자가 "bob.rsk"의 소유자이고, 앨리스가 서브도메인 "subdomain.bob.rsk"를 원할 경우, 밥은 경매 과정 없이 앨리스에게 서브도메인 소유권을 위임할 수 있습니다. 사용자는 도메인을 획득한 후 새로운 도메인과 원하는 리소스 사이의 확인을 수행하는 Resolver를 도메인의 Registry 컨트랙트에 정의해야 합니다. 사용자가 Resolver를 설정하지 않으면 기본 Resolver가 설정됩니다. 이 기본 Resolver는 새로운 소유권이 지정된 도메인의 부모의 Resolver입니다. 그 후 새로운 도메인 소유자는 본인의 새로운 도메인을 가지고 Resolver를 만들어야 합니다. 예를 들어 앨리스가 "subdomain.bob.rsk" Registry 항목에 Resolver를 설정하지 않으면 "bob.rsk" Resolver가 설정됩니다.

도메인 확인

도메인 확인은 도메인이 존재하는지 확인하고 해당 Registry 항목과 관련된 정보를 반환하는 절차입니다. 이 확인을 사용하면 지갑, 거래 또는 dApps 내에서 해당 주소를 모르는 상태에서도 도메인을 다룰 수 있습니다. 한 예로 먼저 "bob.rsk"라는 도메인이 존재한다고 가정해 보겠습니다. "bob.rsk" 도메인을 확인하려면 사용자는 아래에서 설명한 대로 노드를 반환하는 네임 해시 알고리즘을 사용해야 합니다(예: 0x231..de3). 그 후 사용자가 해당 노드와 연결된 RNS Registry 항목을 찾아야 합니다. 이 항목은 노드가 주어졌을 때 원하는 리소스를 반환하는 Resolver를 포함합니다.

기술 개요

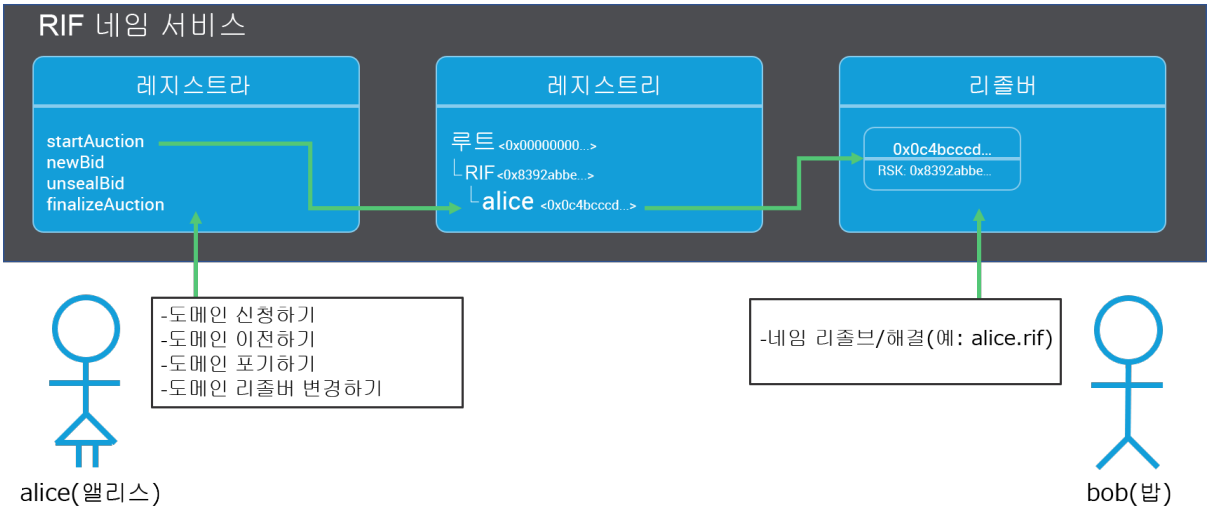
RSK Labs는 도메인 네임과 해당 소유자 간의 매핑을 처리하는 Registry 컨트랙트를 배포합니다. 각 Registry 항목은 네임 도메인과 원하는 리소스 간의 확인을 처리하는 Resolver를 참조합니다.

그런 다음, 도메인 경매와 공급을 관리하는 Registrar 컨트랙트가 배포됩니다. Registrar는 Deed 컨트랙트를 생성하고 사용자의 입찰 금액을 Deed로 이동합니다. 경매 낙찰자는 Deed의 잔액을 잠그고 이 Deed와 도메인 소유권을 교환합니다. 그런 다음, 시스템은 Registry에 경매 낙찰자를 도메인 소유주로 등록하며, 소유주는 자신만의 Resolver를 설정할 수 있습니다.

또한 소유주는 경매 절차 없이 Registry 컨트랙트를 사용하여 서브도메인을 위임할 수 있습니다.

모든 네임 소유자는 각 네임에 따른 Deed를 가지고 있으며, 각 Deed마다 연간 임대료를 지불해야 합니다. Registry에 임대료가 청구되는 이유는 Registry 컨트랙트 저장소에서의 도메인 스쿼팅과 도메인

스팸을 방지하기 위해서입니다. 임대료를 지불하지 않으면 도메인 소유권은 취소되며, Registrar 컨트랙트 내에서 도메인의 상태는 새로운 경매에 공개됩니다(오픈 상태).



구성 요소

Registry

Registry를 위한 지침과 인터페이스는 RNSIP01[6]에서 설명합니다.

네임 형식

RNS 네임은 다음 구문을 따라야 합니다.

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

즉, 네임은 점으로 구분된 일련의 레이블로 구성됩니다. 각 레이블은 transitional=false 옵션과 함께 UTS46 [7]에 기술된 유효하고 정규화된 레이블어야 STD3AsciiRules = true를 사용해야 합니다. JavaScript 구현을 위해서는 네임을 정규화하고 확인하는 라이브러리[8]를 사용할 수 있습니다.

참고로 네임에는 대문자와 소문자를 사용할 수 있지만, UTS46 정규화에서는 레이블을 해싱하기 전에 레이블의 모든 문자를 소문자로 변환하므로, 대/소문자가 다르지만 철자가 같은 두 네임은 같은 네임 해시를 생성합니다.

라벨과 도메인 길이는 제한이 없지만, 구식 DNS와의 호환성을 위해 레이블을 각각 64문자 이하로 제한하고 RNS 네임을 255문자 이하로 완성하는 것이 좋습니다. 같은 이유로 레이블이 하이픈으로 시작하거나 끝나지 않고 숫자로 시작하지 않는 것이 좋습니다.

네임 해시 알고리즘

RNS는 네임 해시 알고리즘을 사용합니다. 이 알고리즘은 네임의 구성 요소를 반복적으로 해시하여 모든 입력 도메인에 대해 고유한 고정 길이 문자열을 생성합니다.

네임 해시의 출력을 '노드'라 합니다.

네임 해시 알고리즘을 위한 의사 코드는 다음과 같습니다.

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

Resolver

Resolver는 인터페이스입니다. 사용자는 RSK가 제공하는 공용 Resolver 컨트랙트를 사용하거나 본인의 Resolver 컨트랙트를 구현할 수 있습니다. 사용자가 자신의 Registry 항목에 자기만의 Resolver를 직접 설정하지 않으면 해당 부모 도메인 Resolver가 사용됩니다. 그러면 사용자는 부모 도메인 Resolver에 도메인 네임과 원하는 리소스 간의 확인 정보를 등록해야 합니다. Registry와 마찬가지로 Resolver 인터페이스와 사양도 RIFIP01[6]에서 설명합니다.

Registrar

Registrar 인터페이스

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

- 생성자는 RNS Registry, Registrar가 속하는 루트 노드, 그리고 RIF 결제에 사용할 ERC 677 토큰 컨트랙트를 수신합니다.

startAuction(bytes32 _hash) public

- 해시의 상태를 공개에서 경매로 변경합니다.

startAuctions(bytes32[] _hashes) public

- 누구나 여러 개의 해시에 대한 경매를 시작할 수 있습니다. 이 방법을 사용하면 공격자가 무조건 경매에 입찰하는 것을 방지할 수 있습니다. 이 경우 발신자는 해시 하나에 대한 입찰에만 관심이 있지만 제출된 해시의 일부는 더미 해시입니다. 따라서 공격자가 모든 새 경매에 무조건 입찰하려면 비용이 증가합니다. 공개되었지만 입찰되지 않은 더미 경매는 1주일 후 마감됩니다.

newBid(bytes32 sealedBid, uint tokenQuantity) public

- 입찰은 메인 컨트랙트에 sealedBid 해시(shaBid 기능으로 생성됨) 및 여러 개의 토큰이 포함된 메시지를 메인 컨트랙트에 보내면 생성됩니다. 이 해시는 입찰된 네임 해시, 입찰 값, 그리고 무작위 솔트(random salt)를 포함하여 입찰에 관한 정보를 포함합니다. 입찰은 공개되기 전까지 어떤 경매에도 구속되지 않습니다. 입찰 값 자체는 실제 입찰 값보다 더 많은 값을 보내서 위장할 수 있습니다. 경매 기간이 끝나면 48시간 공개 기간이 이어집니다. 이 기간 이후 입찰이 공개되면

제공된 토큰으로 벌금을 물 수 있습니다. 이러한 절차는 경매 절차이므로 알려진 도메인과 일반적인 사전적 단어와 같은 대부분의 공개 해시는 가격을 높이는 여러 명의 입찰자가 있을 것으로 예상됩니다. 끝으로, 여러 개의 토큰과 이를 관리하는 컨트랙트가 포함된 Deed가 생성됩니다.

newBidWithToken(bytes32 sealedBid, uint tokenQuantity,) public

- 이는 newBid와 동일하며, ERC 677 컨트랙트 호출에 유용합니다.

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

- 단일 거래에서 startAuctions에 이어 newBid를 호출할 수 있는 유틸리티 기능입니다.

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

- 경매 기간이 끝나면 입찰 소유권 증명을 제출하는 공개 기간이 있습니다. Registrar는 shaBid() 함수를 사용하여 이러한 매개 변수를 해시하고 이것이 기존의 봉인 입찰과 일치하는지 확인합니다. unsealedBid가 새로운 최고 입찰인 경우, 기존의 최고 입찰은 입찰자에게 반환됩니다.

cancelBid(address bidder, bytes32 seal) public

- 아래 환불 스케줄에서 설명하는 규칙에 따라 공개되지 않은 입찰을 취소합니다.

finalizeAuction(bytes32 _hash) public onlyOwner(_hash)

- 공개 기간이 끝나면 경매를 완료하기 위해 이 함수를 호출해야 합니다. 경매가 닫히고 나면 RNS Registry는 최고 입찰자를 경매된 네임의 새로운 소유자로 업데이트합니다.

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

- RNS Registry를 업데이트하여 레이블 해시의 소유권을 새 소유자에게 이동합니다.

releaseDeed(bytes32 _hash) public onlyOwner(_hash)

- Deed가 생성된 지 9개월 후 네임의 새로운 소유자는 이 메서드를 호출하여 네임을 포기하고 자신의 Deed 자금의 일부를 돌려받을 수 있습니다.

eraseNode(bytes32[] labels)

- 누구나 현재 Registrar에 소유되지 않은 네임의 서브도메인 소유자와 Resolver를 삭제할 수 있습니다. 예를 들어 ".rsk"를 소유하는 Registrar에서 my.example.rsk를 삭제하려면 [sha3('my'), sha3('example')]가 포함된 배열을 전달합니다.

transferRegistrars(bytes32 _hash)

- 이 Registrar가 더 이상 RNS의 루트 노드의 소유자가 아닌 경우, 이 함수는 Deed를 현재 소유자(새 Registrar이어야 함)에게 이동합니다. 이 함수는 이 Registrar가 여전히 해당 루트 노드를 소유하면 오류를 표시합니다.

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

- 비밀 경매에 필요한 값을 해시합니다.

payRent(bytes32 _hash) public

- 도메인의 연간 임대료를 지불하십시오.

payRentWithTokens(bytes32 _hash) public

- payRent와 동일합니다. ERC 677 컨트랙트 호출에 유용합니다.

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

- Registrar 마이그레이션을 위해 노드 이동에 동의하고 해당 상태를 변경합니다.

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

- ERC 677과 함께 이동하는 데 필요한 함수입니다.

경매 프로세스

비커리 경매는 4 단계로 이루어집니다.

- **오픈:** 도메인의 기본 상태입니다.
- **경매:** 경매가 시작되었습니다. 사용자가 자신의 봉인 입찰을 제출할 수 있는 72시간의 시간이 있습니다. 봉인 입찰은 shaBid(*bytes32 해시, 주소 소유자, 단위 값, bytes32 솔트*)를 통해 획득할 수 있습니다.
- **공개:** 경매 이후에는 48시간의 공개 기간이 있습니다. 각 입찰자는 본인의 입찰가를 공개하며 Registrar는 그에 따라 경매를 업데이트합니다. 공개된 입찰이 없으면 상태는 오픈으로 되돌아갑니다.
- **소유됨:** 공개 기간이 끝나면 낙찰자는 *finalizeAuction* 메서드를 사용하여 거래를 제출하여 공개 시간을 완료해야 합니다. 이렇게 해서 경매가 완료되며 낙찰자를 경매 네임 해시의 주인으로 기록합니다.

환불 및 벌금 스케줄

입찰 결과	설명	관련 결제
사용자가 경매에 낙찰됨	사용자가 경매에 낙찰되면 두 번째로 높은 입찰 금액이 해당 사용자의 Deed에서 잠기며(TL) 해당 금액과 최고 입찰 금액 사이의 차액이 환불됩니다. 경매가 끝나면 연간 임대료 금액(Y)에서 이 잠긴금액을 뺀 비율이 수수료(F)로 지불됩니다.	T2: 두 번째로 높은 값 Y: 연간 임대료 TL: Deed에 잠긴 금액 F: 수수료 T = (T2 - Y) F= T*0.2 TL = T*0.8
사용자가 경매에서 낙찰되지 않음	사용자의 제안이 최고 제안이 아니라는 이유로 경매에서 낙찰되지 않은 경우, Deed에 잠긴 금액의 비율이 수수료(F)로 지불됩니다.	TL: Deed에 잠긴 금액 F: 수수료 F = 0.05*TL

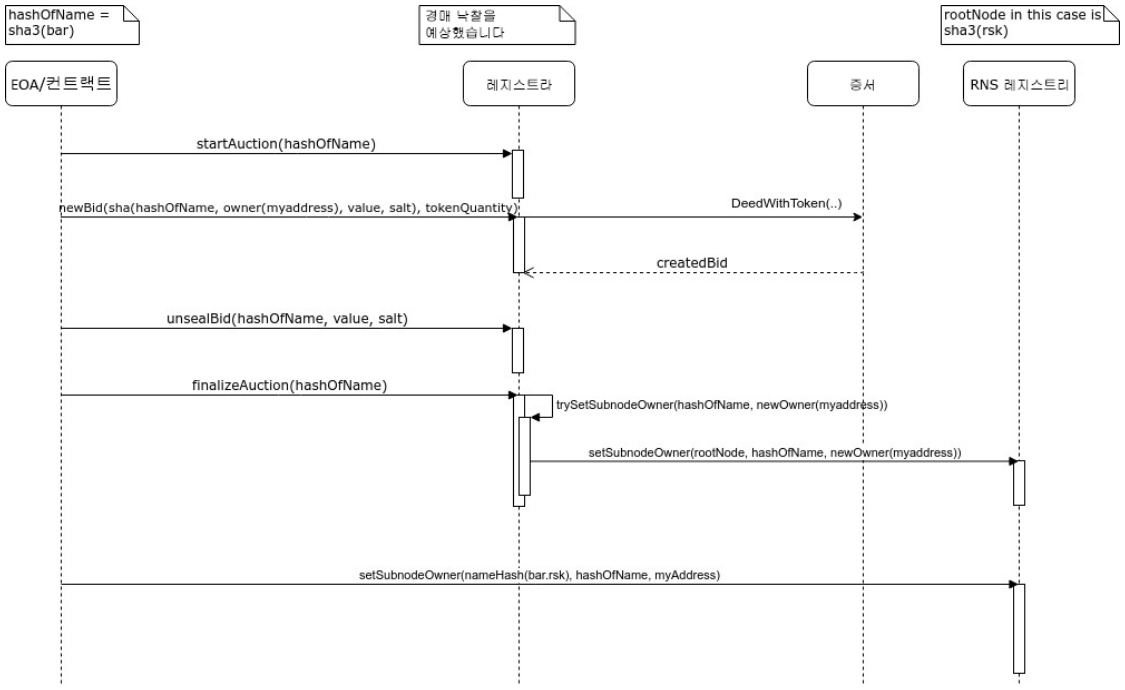
비공개 입찰을 공개하는 기간(공개 기간)입니다. 공개 기간이 시작되기 전에 입찰자가 입찰을 취소하면 입찰 토큰의 99.5%가 환불됩니다. 그런 다음, 공개 기간이 시작되 전에 입찰자가 입찰을 공개하면 거래는 원래대로 돌아갑니다. 이 기간이 만료되면 각 입찰이 결산됩니다. T가 낙찰 금액, T2가 두 번째로 높은 입찰 금액, V가 공개 기간이 끝난 후 공개된 입찰 금액이라고 가정하면 결제와 환불은 다음과 같은 조건으로 진행됩니다.

조건	설명	관련 결제
V > T 일 경우	적시에 공개되었다면 이는 낙찰된 경매임	V*0.2가 수수료로 지불되고 나머지는 환불됨
T > V > T2 일 경우	적시에 공개되었다면 이는 두 번째로 높은 값임	V - T2가 수수료로 지불되고 나머지는 환불됨
그렇지 않으면	공개되지 않았으며 두 번째로 높은 값보다 낮음	V*0.05가 수수료로 지불되고 나머지는 환불됨

공개 기간이 끝난 후 사용자에게는 입찰을 공개할 15일이 주어집니다. 사용자가 여전히 공개하지 않으면 Deed에 완전히 잠긴 토큰이 RNS Network Resource Pool로 전달됩니다. 이 과정은 RIF Token이 영구적으로 잠기는 것을 방지하기 위해 수행됩니다.

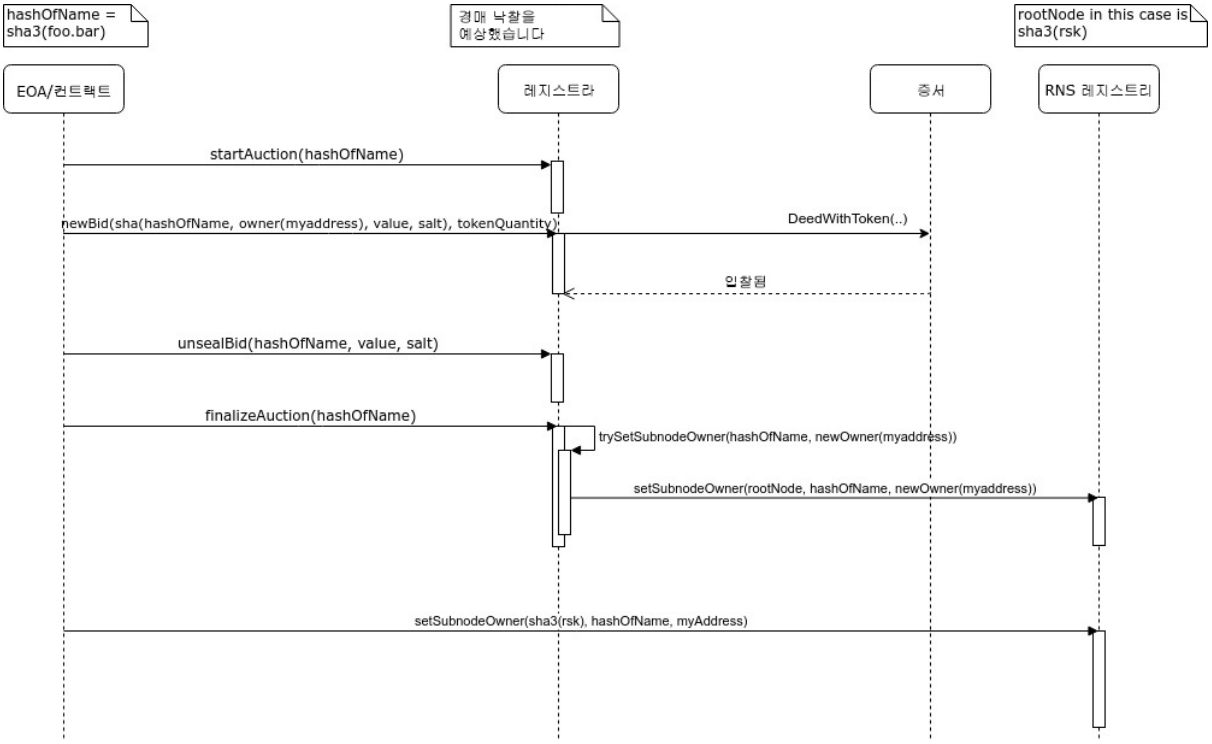
도메인의 고유성

앨리스가 도메인 "bob.alice.rsk"의 소유권을 획득하고 서브도메인 "bob.alice.rsk"의 소유권을 다른 사용자 밥에게 위임한다고 가정합니다. 서브도메인의 소유권을 획득하는 순서는 다음과 같습니다.



이 상호 작용은 악성 사용자 Mallory가 자신이 ".bob"의 정당한 주인이 아닌데도 ".rsk" Registrar의 `sha3("subdomain.bob")` 도메인에 대한 경매를 시작할 수 있음을 암시할 수 있습니다.

이는 경매가 네임 문자열이 아닌 도메인의 네임 해시에 대해 이루어지기 때문입니다. Mallory가 밥이 소유한 "subdomain.bob.rsk" 도메인을 스쿼팅하려 한다고 가정해 보겠습니다. 프로세스는 다음과 같을 것입니다.



그 후 Mallory는 "subdomain.bob.rsk"의 확인을 다른 리소스로 재지정할 의도로 자신의 Resolver 컨트랙트를 사용하여 RNS Registry에 sha3((sha3('rsk'), sha3('subdomain.bob')))에 대한 항목을 설정합니다. 그러나 사용자가 도메인 네임 "subdomain.bob.rif"을 조회하면 (위에서 설명한) 네임 해시 알고리즘이 sha3(sha3('rsk'), sha3('subdomain.bob')) 대신에 sha3(sha3(sha3('rsk'),sha3('bob')), sha3('subdomain'))을 확인합니다. 따라서 네임 해시 알고리즘이 확인한 도메인 네임은 밥이 정의한 네임입니다.

Deed

Registrar에 보내진 모든 RIF Token은 "Deed"라는 별도의 컨트랙트에 저장됩니다. 각 Deed는 특정 네임 해시의 토큰 잔액을 저장합니다. Deed는 입찰이 제출될 때마다 생성됩니다. 그 후 경매가 끝나고 도메인이 등록되면 낙찰 Deed와 입찰이 잠기고 도메인 소유권과 교환됩니다. 낙찰되지 않은 입찰과 관련된 Deed 잔액은 요청할 경우 정당한 주인에게 환불됩니다. Deed는 Registrar와 마찬가지로 토큰 결제를 처리하는 RIF 토큰 ERC 677 컨트랙트를 알고 있습니다.

소유된 네임에 대한 Deed는 해당 네임의 소유자에 의해 다른 계정으로 이동될 수 있으며, 따라서 네임의 소유권과 제어권도 이동됩니다. 이 과정은 Registrar 컨트랙트를 통해 이루어집니다.

경매 종료 후 9개월이 지나면 노드 소유자에게는 연간 임대료를 지불하고 도메인을 1년 더 소유하는 옵션이 주어집니다. 소유자가 1년분의 임대료를 더 지불하고 싶지 않으면 소유권을 포기하고 Deed에 잠긴 잔액을 환불받을 수 있습니다.

도메인 임대료를 지불하려면 사용자는 Registrar의 payRent 함수를 사용할 수 있습니다. 이 함수를 사용하려면 RIF 토큰으로 지불해야 합니다. Deed 생성된 지 9개월 후 소유자가 소유권을 포기하는 경우, 해당 소유자에게는 releaseDeed를 호출하여 잠긴 토큰을 환불받을 수 있는 3개월의 기간이 주어집니다. 이 3개월이 지나면 해당 도메인의 경매 상태는 오픈으로 전환되며 Deed 잔액 전체가 RNS Network Resource Pool로 이동됩니다.

낙찰되지 않은 입찰에 대한 Deed는 여러 메서드를 통해 종료할 수 있으며, 이때 보류된 RIF 토큰은 입찰자에게 반환됩니다.

ERC 677 토큰 컨트랙트

토큰 RIF는 ERC 677 표준을 사용하여 구현됩니다. 연간 임대료나 RNS 입찰에 대한 결제는 RIF 토큰을 사용하여 이루어집니다. 그러므로 ERC 677 RIF 토큰 컨트랙트와 상호작용하는 프로세스는 다음과 같습니다.

- 매개 변수가 3개인 `transferAndCall` 함수에서 서명은 다음과 같아야 합니다.
    - newBid: *데이터* 매개 변수에 대해 shaBid 함수가 생성한 sealedBid와 연결된 서명 0x1413151f를 설정합니다.
    - payRent: *데이터* 매개 변수에 대해 임대료가 지불된 레이블 sha3과 연결된 서명 0xe1ac9915를 설정합니다.

개선 제안

RNS 아키텍처에 속하는 컨트랙트는 새로운 개선 사항을 도입하기 위해 업그레이드될 수 있습니다. 이러한 업그레이드는 커뮤니티 피드백과 RSK Labs의 제안을 기반으로 제공됩니다. 모든 RNS 업그레이드는 하위 버전과 호환됩니다. 요컨대 도메인 소유자가 도메인 소유권을 유지합니다.

서브도메인 관리

사용자가 특정 도메인의 서브도메인을 획득할 수 있는 방법에 두 가지가 있습니다. 도메인 소유자가 Registrar인 경우, 사용자는 해당 도메인의 어떤 서브도메인에 대해서도 경매를 시작할 수 있습니다. 또한 도메인 소유자는 경매 절차 없이 setSubnodeOwner 함수를 사용하여 구입자에게 서브도메인을 위임할 수 있습니다. 마지막 옵션에서는 새로운 소유자가 서브도메인을 더 이상 사용하지 않을 때 Registry 항목을 삭제해도 혜택을 제공하지 않는데, 그 이유는 Deed 컨트랙트가 없으므로 잠긴 값이 없기 때문입니다. 당사는 더 나은 위임 시스템과 공정한 서브도메인 관리 절차를 개발하고자 노력하고 있습니다.

새로운 Registry 구조

Registry 컨트랙트는 노드 소유권에 대해 유효한 유일한 컨트랙트입니다. Registry는 모든 RNS 정보를 저장합니다. 이는 저장소 임대가 실행된 후에는 확장되지 않습니다. 당사는 저장소 임대를 더 공정하게 만드는 대체 Registry 구조를 연구하고 있습니다.

DNS 도메인과 오라클

일반 DNS 주소를 RNS로 마이그레이션할 수도 있습니다. 이때 DNS 주소 소유자는 오라클을 사용하여 본인이 원래 도메인의 정당한 소유자임을 확인함으로써 RNS 내에서 도메인 소유권을 선언할 수 있습니다. ICANN 네임 도메인과 충돌하는 경우, 중재 시스템을 사용하여 분쟁을 해결합니다. 이 중재 시스템은 다른 방식과 함께 오라클을 통해 구현할 수 있습니다.

Resolver 익명성

사용자는 자신의 도메인이 매핑되는 주소를 숨기고자 할 수 있습니다. 암호화된 리소스로 그렇게 할 수 있습니다. 소유자는 오프 체인 통신 채널을 통해 요청할 때 사용자에게 해독 키를 전달할 수 있습니다. 또한 저장된 주소는 도난당할 수 있으므로 발신자는 각 독립된 결제에 대해 새 주소를 생성해야 합니다.

최상위 Registrar 생성하기

RSK Labs는 TLD를 위한 초기 Registrar(.rsk)를 제공했습니다. 앞으로 사용자는 자기만의 Registrar를 배포하는 자기만의 TLD를 생성할 수 있게 될 것입니다. 이때 사용자는 타인이 서브도메인을 획득하게 하는 경매 프로세스(또는 다른 프로세스)를 활성화할 수 있습니다.

참고 문헌

[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