# RIF Name Service

Especificación

v1.31-es

# Índice

# Preámbulo

Uno de los pilares de la World Wide Web es el sistema de nombres de dominio (DNS). Este sistema es responsable de crear una asignación entre nombres legibles por humanos y direcciones numéricas de IP. La Corporación de Internet para la Asignación de Nombres y Números (ICANN) es una organización responsable de coordinar el mantenimiento y los procedimientos de varias bases de datos relacionadas con espacios de nombres y espacios numéricos de Internet, asegurando el funcionamiento de la red. La ICANN realiza el mantenimiento técnico real de los registros de zona raíz del DNS.

Estos servicios son un punto central de confianza y fracaso[1][2]; pueden ser desconectados por ataques por denegación de servicio distribuido (DDoS) y las asignaciones de dominios pueden modificarse ya sea forzando cambios en los servidores DNS o falsificando respuestas de ellos. Además, existen algunos problemas de seguridad, como que los proveedores de servicios de Internet (ISP) sean capaces de censurar nombres sin que sea fácil detectarlo.

Uno de los objetivos del RIF Name Service (RNS) es ser un sistema descentralizado y seguro similar al DNS. RNS funciona sobre la RSK Blockchain, heredando de este modo su naturaleza y seguridad descentralizadas.

Un problema en la adopción de las criptomonedas es la dificultad para tratar con las direcciones. Una dirección de Bitcoin tiene el siguiente aspecto: “06f1b66ffe49df7fce684df16c62f59dc9adbd3f”, la cual es notablemente propensa a errores cuando un usuario intenta transcribirla. Una secuencia tan larga de caracteres es también difícil de recordar, lo que hace que la adopción frecuente de las criptomonedas sea poco práctica.

En conclusión, el RNS es un servicio descentralizado que les da a los usuarios un dominio o alias legible por humanos que hace referencia a diferentes recursos (por ejemplo, direcciones RSK o Swarm). Una ventaja del uso de direcciones legibles por humanos es la reducción de la aparente complejidad de la tecnología blockchain para el usuario final, lo que facilita su uso.

Los dominios y subdominios se compran y venden en un mercado abierto. El mecanismo para obtener un dominio por primera vez es a través de una subasta ciega de Vickrey, [3] ofertando con RIF Tokens. La práctica ha demostrado que son las peculiaridades psicológicas humanas, y no solo la oferta y la demanda, las que impulsan las subastas. El mecanismo de subasta de Vickrey reduce la probabilidad de que un postor pague de más por un artículo y aumenta la probabilidad de que el vendedor obtenga lo máximo que puede obtener por él. Una vez que un usuario gana una subasta, se le asigna la propiedad del dominio y se paga una renta anual para conservar dicha propiedad. Los detalles de cómo se calcula esta renta se explicarán en la sección técnica del “Registrar” de este documento.

Las pautas iniciales proporcionadas en este protocolo pueden estar sujetas a cambios adicionales, a medida que las ideas y la arquitectura son discutidas y mejoradas por el ecosistema en el futuro.

# El Ecosistema RIF

## RIF Token

En la economía del RNS Token, la principal función del RIF Token es evitar el almacenamiento no utilizado debido a dominios no utilizados, o evitar la usurpación de nombres. El propietario del dominio debe tener incentivos para renunciar a cualquier propiedad de dominio no utilizado. Para lograr esto, el propietario del dominio bloquea los RIF tokens que son reembolsados cuando se liberan los dominios. Además, cualquier pago de tarifa o penalización debe realizarse utilizando RIF Tokens, que serán asignados al Network Resource Pool. El Network Resources Pool existe para subsidiar ciertos gastos de la red que serán necesarios en las primeras etapas de la adopción del RIF Name Service. El objetivo final es la gestión descentralizada de los fondos.

## RIF Open Standard (RIFOS)

RIF Explorer, implementado por RIF Labs, es una capa de abstracción que permite que cada servicio RIF sea desacoplado de una implementación particular. Las implementaciones particulares son conocidas por la plataforma como proveedores de servicios. Este desacoplamiento permite que cada servicio evolucione a medida que se disponga de nuevas mejoras tecnológicas.

Para ser compatibles con diferentes proveedores de servicios para cada servicio de la Plataforma RIF, el RNS se utiliza como un mecanismo de descubrimiento de servicios. Esto permitirá que usuarios y desarrolladores averigüen qué proveedores de servicios están disponibles y cómo comunicarse con ellos. Para entender más sobre el RIF Explorer, lea su informe oficial [4].

# Introducción

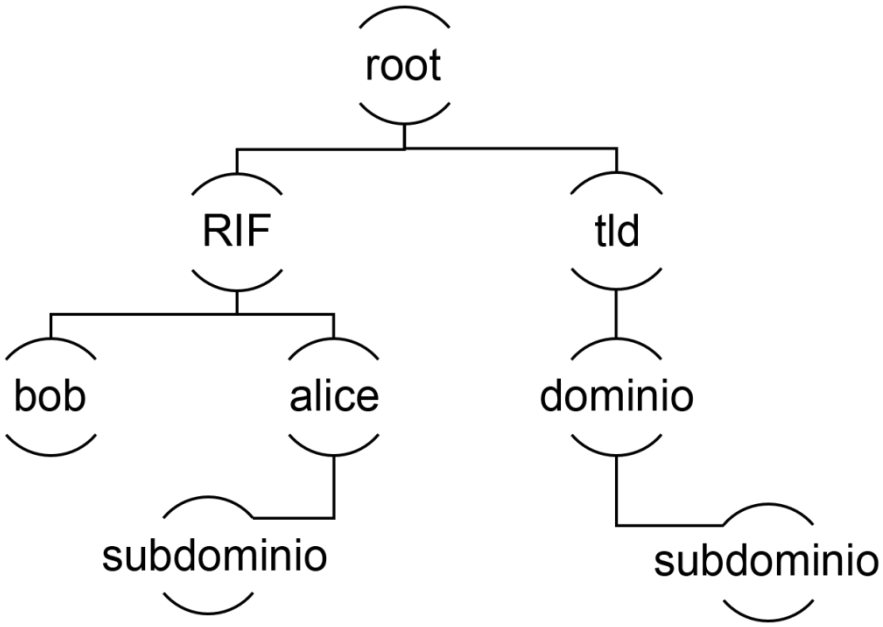
La siguiente sección ofrece una descripción general de los componentes de RNS. Para conocer más detalles, lea la Sección técnica.

## Resumen de componentes

La arquitectura del RNS se basa en el Ethereum Name Service (ENS) que se describe en EIP-137 [5]. Está dividido en tres componentes principales: Registry, Resolver y Registrar.

### Node

En RNS, el nombre de nivel superior se define como "*.rsk*". Los nombres de segundo nivel se llaman dominios. Además, un dominio puede hacer referencia a diferentes recursos a través de subdominios. Un nodo es una parte de un árbol de jerarquía de subdominios. La ruta a la raíz en este árbol es un nombre de dominio. Los componentes de esta ruta están divididos por un punto ("."). Cada nodo almacena un hash de la cadena de nombre de un componente intermedio, calculado mediante el algoritmo de hash explicado anteriormente.




#### Formato de nombre

El formato de nombre RNS debe cumplir con la siguiente sintaxis: "subdomain.domain.rsk". En resumen, los nombres consisten de una serie de etiquetas separadas por puntos, cada una de las cuales se compone de un nivel del árbol de subdominios. Además, el nombre del subdominio debe cumplir con las reglas explicadas en la sección técnica de Formato de nombre.

### Resolver

Los contratos Resolver se encargan de la resolución del nombre de un recurso. Un resolver tiene muchas funciones definidas por el usuario y cada función habilita un tipo diferente de recurso para ser buscado en el mismo nodo.

### Registry

El contrato registry proporciona una asignación simple entre un dominio y su resolver. Todo lo relacionado con la propiedad de un dominio se gestiona en este contrato, incluida la transferencia de propiedad y la creación de subdominios.

### Registrar

El registrar es responsable de la gestión del RNS. Además, es responsable de registrar el nombre de un dominio para un usuario y la única entidad capaz de actualizar el RNS Registry. En caso de migración a otro registrar (el proceso de migración se explica a continuación), puede delegar la propiedad de los subdominios a otros registrars.

Como se explicó anteriormente, al principio solo estará disponible un top-level domain (TLD): ".rsk". Al reducir el número de dominios de top-level domains, podemos centrarnos en el registro de dominios de segundo nivel y posponer la discusión sobre superposiciones de alto nivel con el sistema DNS estándar y otros servicios de nombres.

Para evitar que los dominios se conviertan en spam o sean usurpados, el registrar gestionará el proceso de subasta. Como los pagos de RNS están denominados en RIF Tokens, el registrar interactúa con el contrato ERC 677 RIF Token para efectuar pagos entre cuentas. Inicialmente, solo manejará el dominio ".rsk" y los subdominios de cualquier longitud de caracteres. Esto estará restringido desde el nodo raíz RNS, que está controlado por un contrato de multifirma. Este contrato multifirma será controlado inicialmente por RSK Labs durante un período beta.

### Deed

A fin de evitar el almacenamiento utilizado innecesario debido a dominios no utilizados o para evitar la usurpación de nombres, el propietario del dominio deberá tener incentivos para renunciar a su propiedad sobre ellos. Para lograr esto, el propietario del dominio bloquea los tokens que son reembolsados cuando se liberen los dominios.

## Principales casos de uso

### Registro de un dominio

Hay dos formas en que los usuarios pueden obtener un dominio. La primera es abrir una subasta para el dominio deseado a través del contrato de registrar. Por ejemplo, si ".rsk" es el TLD y un usuario llamado Alice quiere el dominio "alice.rsk", ella puede abrir una subasta para este dominio, hacer una oferta, y si es la más alta, se convertirá en la nueva propietaria del dominio "alice.rsk". Asimismo, si un usuario llamado Bob es el propietario de "bob.rsk" y Alice quiere el subdominio "subdomain.bob.rsk" Bob puede delegar la propiedad del subdominio a Alice sin un proceso de subasta. Una vez que un usuario obtiene un dominio, este debe definir en el contrato del registry el resolver que llegará a una resolución entre el nuevo dominio y el recurso deseado. Si un usuario no establece un resolver, se establece uno por defecto. Este default resolver es el nuevo resolver del padre del dominio

propio.  Luego, el nuevo propietario del dominio debe crear una entrada de resolver con su nuevo dominio. Por ejemplo, si Alice no establece un resolver en la entrada del registry “subdomain.bob.rsk”, se establece el resolver “bob.rsk”.

### Resolución de un dominio

La resolución de dominio es un proceso que verifica si un dominio existe y luego retorna la información asociada a su entrada en el registry. Esta resolución puede usarse en wallets, exchanges o dApps, para manejar dominios sin conocimiento de sus respectivas direcciones. Para hacer esto, primero, asumamos que el dominio “bob.rsk” existe. Para resolver el dominio “bob.rsk”, un usuario debe usar el algoritmo de hash explicado a continuación que devuelve un nodo (por ejemplo, 0x231..de3). Después de eso, dado ese nodo, un usuario debe buscar la entrada del RNS Registry asociada a él. Esta entrada contiene el resolver que, dado el nodo, devuelve el recurso deseado.

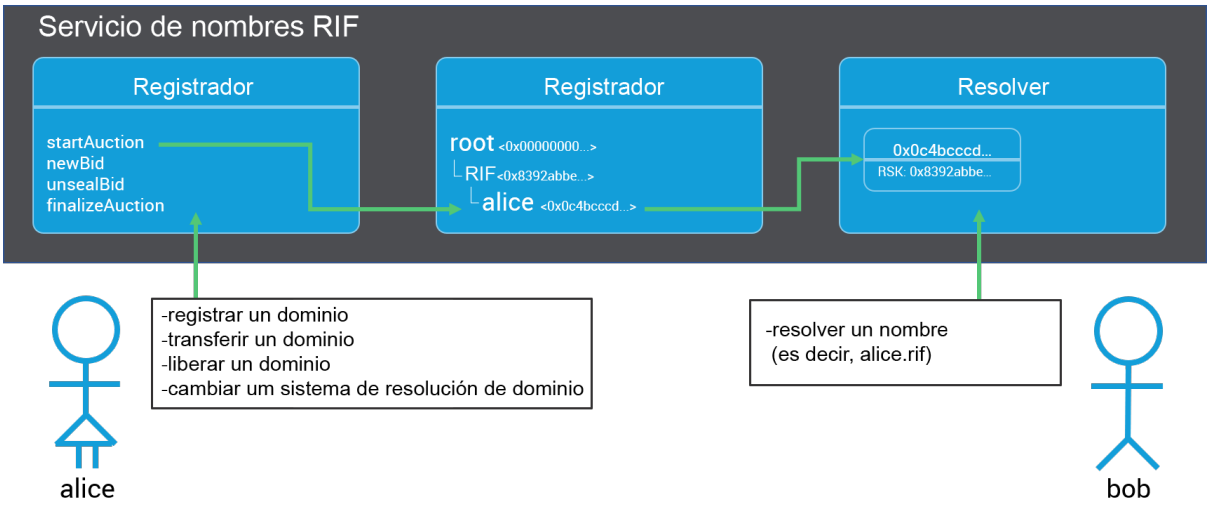
# Resumen técnico

Al principio, RSK Labs utiliza un contrato de registry que maneja la asignación entre el nombre de dominio y su propietario. Cada entrada en el registry hace referencia a un resolver que maneja la resolución entre el dominio de nombre y el recurso deseado.

Posteriormente, se despliega el contrato del registrar que administra la subasta y el suministro de dominios. Para cada oferta, el registrar crea un contrato de deed y traslada el monto ofertado por el usuario al deed. El ganador de la subasta debe bloquear el saldo del deed e intercambiar el deed por la propiedad del dominio. Luego, el sistema registra al ganador de la subasta en el registry como propietario del dominio y este puede establecer su propio resolver.

Además, el propietario puede delegar subdominios utilizando el contrato de registry sin un proceso de subasta.

Cada propietario de un nombre tiene un deed por cada nombre y debe pagar una renta anual por cada deed. El motivo para cargar las rentas del registry es evitar la usurpación de dominios y el spam de dominio en el almacenamiento de contratos del registry. Si la renta no se paga, la propiedad del dominio se cancela y el estado del dominio en el contrato del registrar se abre a nuevas subastas (el estado abierto).




# Componentes

## Registry

Las pautas y la interfaz para el registry se describen en RNSIP01[6].

### Formato de nombre

Los nombres de RNS deben cumplir con la siguiente sintaxis:

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

En resumen, los nombres están compuestos de una serie de etiquetas separadas por puntos. Cada etiqueta debe ser una etiqueta normalizada válida como se describe en UTS46 [7] con las opciones transitional=false y usar STD3AsciiRules=true. Para implementaciones de JavaScript, hay una biblioteca [8] disponible que normaliza y verifica los nombres.

Tenga en cuenta que, si bien las letras mayúsculas y minúsculas están permitidas en los nombres, el proceso de normalización UTS46 pliega las etiquetas antes de mezclarlas, por lo que dos nombres con mayúsculas y minúsculas diferentes pero con ortografía idéntica producirán el mismo hash.

Las etiquetas y los dominios pueden tener cualquier longitud, pero para que sean compatibles con el DNS legado, se recomienda que las etiquetas estén restringidas a no más de 64 caracteres cada una, y completen los nombres RNS a no más de 255 caracteres. Por la misma razón, se recomienda que las etiquetas no comiencen ni finalicen con guiones, o que comiencen con dígitos.

### Algoritmo de la función hash

RNS usa el algoritmo de la función hash. Este algoritmo mezcla los componentes del nombre de forma recurrente, produciendo una cadena única de longitud fija para cualquier dominio de entrada válido.

El resultado de la función hash se conoce como 'nodo'.

El pseudocódigo para el algoritmo de hash es el siguiente:

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

## Resolver

El resolver es una interfaz. Un usuario puede usar el contrato public resolver provisto por RSK o implementar su propio contrato resolver. Si un usuario no establece su propio resolver para su entrada en el registry, se utilizará el resolver del dominio del padre. Luego el usuario debe registrar en el resolver de dominio del padre la información de la resolución entre el nombre de dominio y el recurso deseado. Al igual que con el registry, la interfaz y la especificación del resolver se describen en el RNSIP01. [6].

## Registrar

### Interfaz del registrar

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

- El constructor recibe un RNS Registry, un nodo raíz al que pertenece el registrar, así como un contrato de ERC 677 token para usar en los pagos RIF.

***startAuction(bytes32 _hash) public***

- Cambia el estado de un hash de abierto a subasta.

***startAuctions(bytes32[] _hashes) public***

- Permite a cualquier persona comenzar una subasta por varios hashes. Este método se puede utilizar para evitar que un atacante oferte a ciegas en las subastas. En este caso, algunos de los hash presentados son hash ficticios, mientras que el remitente solo está interesado en ofertar por uno. Esto aumentará el costo para un atacante que simplemente realiza una oferta a ciegas en todas las subastas nuevas. Las subastas

ficticias que están abiertas pero en las que no se oferta se cierran después de una semana.

***newBid(bytes32 sealedBid, uint tokenQuantity) public***

- Las ofertas se crean enviando un mensaje al contrato principal con un hash sealedBid (creado con la función shaBid) y una cantidad de tokens. El hash contiene información sobre la oferta, incluido el hash ofertado, el valor de la oferta y una sal aleatoria. Las ofertas no están ligadas a ninguna subasta hasta que se revelan. El valor de la oferta en sí se puede enmascarar por medio de enviar más del valor realmente ofertado. Una vez que finaliza el período de la subasta, este es seguido por un período de revelación de 48 horas. Si se revela una oferta después de este período, podría penalizarse con los tokens ofrecidos. Como se trata de una subasta, se espera que la mayoría de los hashes públicos, como los dominios conocidos y las palabras comunes del diccionario, tengan múltiples postores que eleven el precio. Por último, se crea un deed con un número de tokens y el contrato que los gestiona.

***newBidWithToken(bytes32 sealedBid, uint tokenQuantity,) public***

- Equivalente a newBid. Es útil para llamadas desde contratos ERC 677.

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

- Una función de utilidad que permite una llamada a startAuctions seguida de newBid en una sola transacción.

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

- Una vez que se completa el período de oferta, hay un período de revelación durante el cual se envía el comprobante de propiedad de una oferta. El registrar evalúa estos parámetros usando la función shaBid() para verificar que coincidan con una oferta preexistente sellada. Si la oferta no sellada es la mejor oferta nueva, la mejor oferta anterior se devuelve a su postor.

***cancelBid(address bidder, bytes32 seal) public***

- Cancela una oferta no revelada de acuerdo con las reglas descritas en el calendario de reembolsos a continuación.

***finalizeAuction(bytes32 _hash) public onlyOwner(_hash)***

- Una vez finalizado el período de revelación, se debe llamar a esta función para finalizar la subasta. Una vez que la subasta se cierra, el RNS Registry se actualiza con el mejor postor como el nuevo propietario del nombre subastado.

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

- Actualiza el RNS Registry, transfiriendo la propiedad de un hash de etiqueta a un nuevo propietario.

***releaseDeed(bytes32 _hash) public onlyOwner(_hash)***

- Nueve meses después de la creación del deed, el propietario de un nombre puede llamar a este método para renunciar al nombre y devolverles parte de sus fondos en el deed.

***eraseNode(bytes32[] labels)***

- Permite a cualquier persona eliminar al propietario y los registros del resolver para un subdominio de un nombre que actualmente no es propiedad del registrar. Por ejemplo, para llevar a cero my.example.rsk en un registrar que posee ".rsk", pase una matriz que contenga [sha3('my'), sha3('example')].

***transferRegistrars(bytes32 _hash)***

- Si este registrar ya no es el propietario del nodo raíz en el RNS, esta función transferirá el deed al propietario actual, que debe ser un nuevo registrar. Esta función arroja un error si este registrar aún posee su nodo raíz.

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

- Combina los valores necesarios para una oferta secreta.

***payRent(bytes32 _hash) public***

- Paga el alquiler anual de un dominio.

***payRentWithTokens(bytes32 _hash) public***

- Equivalente a payRent. Es útil para llamadas desde contratos ERC 677.

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

- Acepta transferencias de nodos y cambia su estado para una migración de registrar.

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

- Función necesaria para hacer transferencias con ERC 677.

## Proceso de subasta

La subasta Vickrey es un proceso de cuatro pasos:

- **Open**: el estado predeterminado del dominio.
- **Auction:** subasta iniciada. Hay un período de 72 horas en el que los usuarios pueden presentar sus ofertas selladas. Las ofertas selladas se pueden obtener a través de *shaBid(bytes32 hash, address owner, uint value, bytes32 salt).*
- **Revealed:** después de la subasta, hay un período de revelación de 48 horas. Cada postor revela su oferta y el registrar actualiza la subasta según corresponda. Si no se revelan ofertas, el estado regresa a abierto.
- **Owned:** cuando finaliza el período de revelación, el ganador debe enviar una transacción para finalizar el tiempo de revelación a través del método *finalizeAuction*. Dicho método finaliza la subasta y registra al ganador como el propietario del nombre hash de las subastas.

| **Resultado de la oferta** | **Descripción** | **Pagos involucrados** |
|---|---|---|
| El usuario gana la subasta. | Si un usuario gana la subasta, la segunda oferta más alta quedará bloqueada en su deed (TL) y la diferencia entre esa cantidad y la oferta más alta será reembolsada. Un porcentaje de este monto bloqueado menos el monto de renta anual (Y) se pagará como tarifa (F) cuando finalice la subasta. | T2: Segundo valor más alto<br>Y: Precio de renta anual<br>TL: Monto bloqueado en el deed<br>F: Tarifa<br>T = (T2 - Y)<br>F = T * 0,2<br>TL = T * 0,8 |
| El usuario pierde la subasta. | Si un usuario pierde la subasta porque su oferta no es la más alta, un porcentaje del monto bloqueado (TL) en su deed se pagará como tarifa (F). | TL: Monto bloqueado en el deed<br>F: Tarifa<br>F = 0,05 * TL |

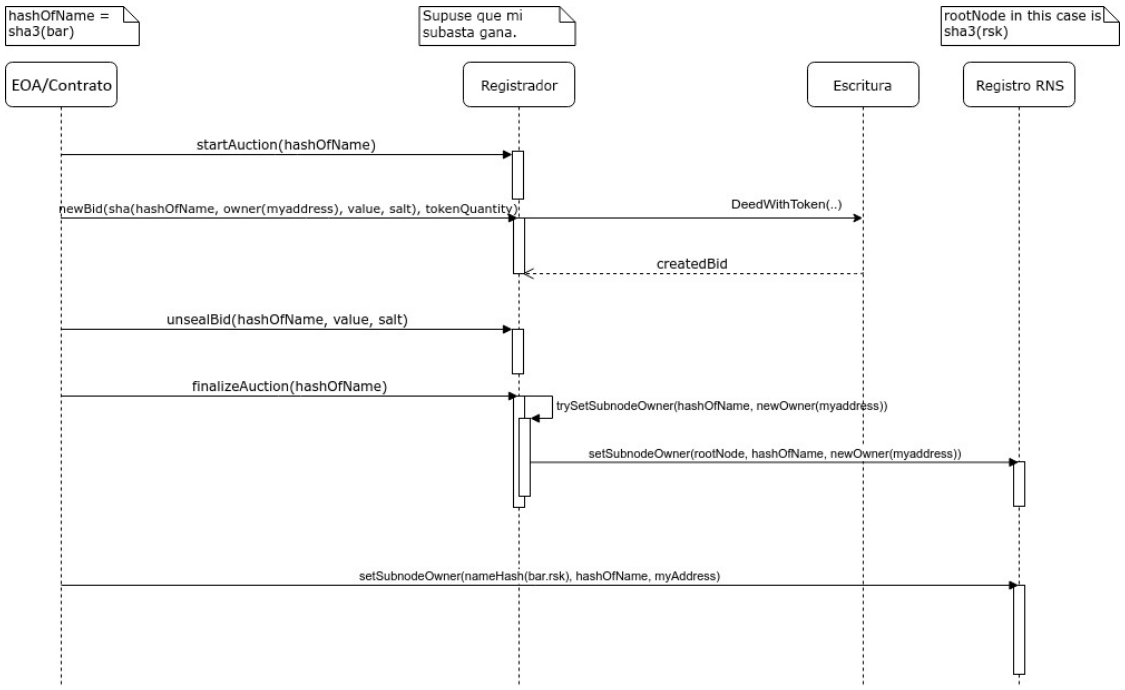
Un período para revelar las ofertas ciegas (período de revelación). Si un postor cancela su oferta antes de que comience el período de revelación, un 99,5 % de los tokens de la oferta son reembolsados. Así, si el postor revela la oferta antes de comenzar el período de revelación, la transacción se revierte. Cuando este período expira, cada oferta se liquida. Supongamos que T es el monto de la oferta ganadora, T2 el segundo monto más alto ofertado y V es una oferta que se revela después de finalización del período de revelación, los pagos y reembolsos se rigen por las siguientes condiciones:

| **Condición** | **Descripción** | **Pagos involucrados** |
|---|---|---|
| Si V > T | Si se hubiera revelado a tiempo, habría ganado. | V * 0,2 se pagará como tarifa y el resto será reembolsado. |
| Si T > V > T2 | Si se hubiera revelado a tiempo, habría sido el segundo valor más alto. | V - T2 se pagará como tarifa y el resto será reembolsado. |
| De lo contrario | No fue revelada y es menor que el segundo valor más alto | V * 0,05 se pagará como tarifa y el resto será reembolsado. |

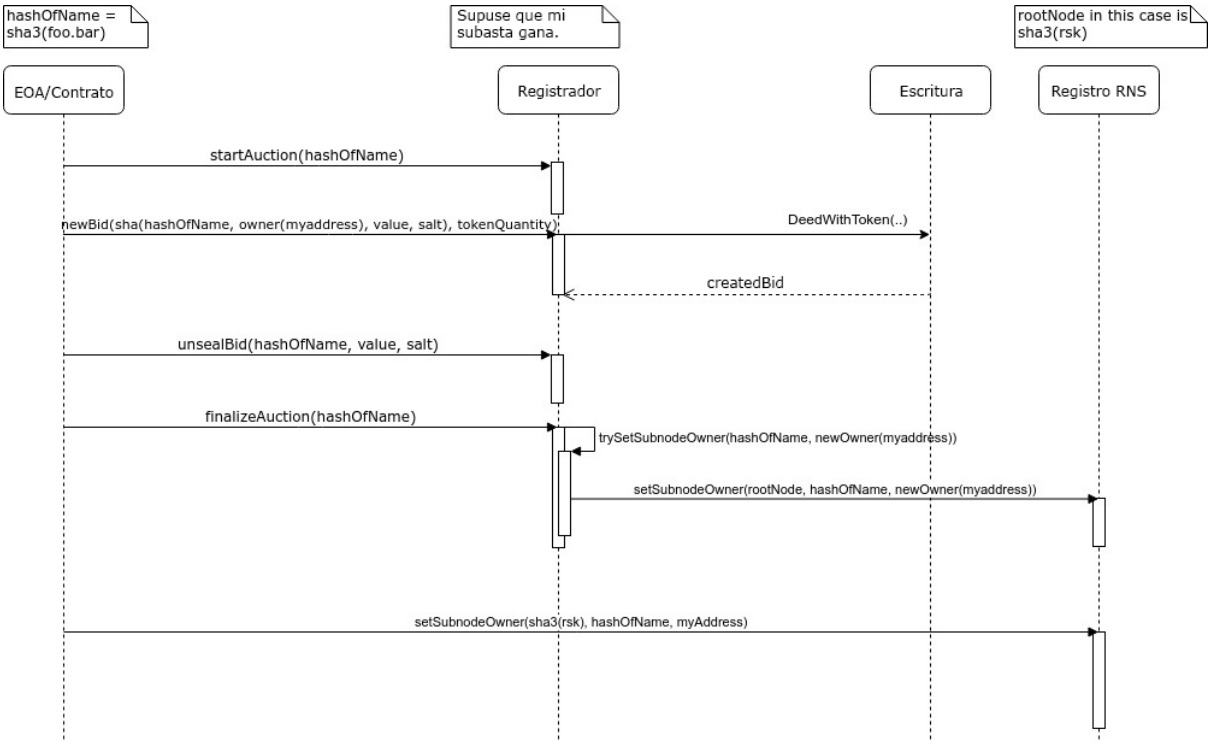
Una vez finaliza el período de revelación, el usuario tiene 15 días para revelar la oferta. Si el usuario aún no la revela, los tokens completamente bloqueados sobre el deed serán enviados al RNS Network Resource Pool. Esto se hace para evitar que los RIF Tokens se bloqueen para siempre.

### Exclusividad de dominios

Supongamos que Alice obtiene la propiedad del dominio “bob.alice.rsk” y delega la propiedad del subdominio “bob.alice.rsk” a otro usuario Bob. La secuencia para obtener la propiedad de un subdominio sería la siguiente:




Esta interacción puede sugerir que un usuario malicioso Mallory puede comenzar una subasta del dominio `sha3(“subdomain.bob”)` en el registrar “.rsk”, incluso si no es el propietario legítimo de “.bob”. Esto se debe a que las subastas son para el hash de un dominio y no para la cadena de caracteres del nombre. Supongamos que Mallory quiere usurpar el dominio “subdomain.bob.rsk” propiedad de Bob. El proceso sería el siguiente:




Posteriormente, Mallory establecerá una entrada en el RNS Registry para el sha3((sha3(‘rsk’), sha3(‘subdomain.bob’)) usando su propio contrato de resolver con la intención de redirigir la resolución de “subdomain.bob.rsk” a un recurso diferente. Pero cuando un usuario busca el nombre de dominio ‘subdomain.bob.rsk’, el algoritmo de hash (explicado anteriormente) resolverásha3(sha3(sha3(‘rsk’),sha3(‘bob’)), sha3(‘subdomain’)) en lugar de sha3(sha3(‘rsk’), sha3(‘subdomain.bob’)). En consecuencia, el nombre de dominio resuelto por el algoritmo de hash será aquel definido por Bob.

## Deed

Cada RIF token enviado al registrar se almacena en un contrato separado, denominado “deed”. Cada deed almacena el saldo del token de un hash específico. El deed se crea cuando se hace una oferta. Luego, cuando se completa la subasta y se registra el dominio, el deed ganador y la oferta se bloquearán e intercambiarán por la propiedad del dominio. Los saldos en los deeds de las ofertas perdedoras se reembolsan a sus legítimos propietarios a solicitud. Al igual que el registrar, un contrato de deed conoce el contrato del RIF Token ERC 677 que maneja los pagos con tokens.

Un deed de un nombre que pertenece a alguien puede ser transferido a otra cuenta por su propietario, transfiriendo así la propiedad y el control del nombre. Esto se hace a través del contrato de registrar.

Nueve meses después de la conclusión de la subasta, el propietario del nodo tendrá la opción de pagar una renta anual, renovando la propiedad sobre el dominio por otro año. Si el propietario no desea pagar otra renta anual, puede optar por renunciar a la propiedad y congelar los fondos en el deed que se le devolvió.

Para pagar la renta de cualquier dominio, un usuario puede usar la función payRent del registrar. La función requiere un pago mediante el RIF token. Si el propietario opta por renunciar a la propiedad, después de nueve meses de la creación del deed, tiene tres meses para llamar a releaseDeed y recibir los tokens bloqueados como reembolso. Después de estos tres meses, el estado de la subasta de ese dominio cambiará a abierto y la totalidad del saldo de los deeds será transferida al RNS Network Resource Pool.

Los deeds para ofertas no ganadoras se pueden cerrar mediante varios métodos, momento en el que se reintegrará al postor cualquier token RIF.

## Contrato ERC 677 Token

El Token RIF se implementa con el estándar ERC 677. Los pagos por renta anual u ofertas en RNS se efectúan con tokens RIF. Por lo tanto, el proceso para interactuar con el Contrato RIF Token ERC 677 es el siguiente:

- En la función `transferAndCall` con 3 parámetros, la firmas deben ser:
    - newBid: Establezca el parámetro *data,* la firma 0x1413151f, concatenada con sealedBid creada por la función shaBid.
    - payRent: Establezca el parámetro *data* la firma 0xe1ac9915, concatenada con el sha3 de la etiqueta por la que se paga el alquiler.

# Propuestas de mejora

Los contratos que pertenecen a la arquitectura RNS se pueden actualizar para introducir nuevas mejoras. Estas actualizaciones se brindan mediante comentarios de la comunidad y propuestas de RSK Labs. Cada actualización de RNS es compatible con versiones anteriores, en otras palabras, la propiedad de los dominios es mantenida por sus propietarios.

## Manejo de subdominios

Existen dos formas en las que un usuario puede adquirir un subdominio de un dominio específico. Si el propietario del dominio es un registrar, el usuario puede iniciar una subasta de cualquier subdominio de ese dominio. Además, el propietario del dominio puede delegar un subdominio al comprador sin pasar por un proceso de subasta a través de la función setSubnodeOwner. Esta última opción no genera ningún incentivo para que el nuevo propietario elimine la entrada en el registry del subdominio cuando ya no esté en uso; esto se debe a que no hay ningún valor bloqueado, porque no hay un contrato de deed. Estamos trabajando en un mejor sistema de delegación y una administración justa de subdominios.

## Nueva estructura de registry

El contrato de registry es el único válido para la propiedad de nodos. El registry almacena toda la información de RNS. Esto no escalará una vez que la renta del almacenamiento haya sido implementada. Estamos investigando una estructura de registry alternativa para obtener una renta de almacenamiento más justa.

## Oráculos y dominios de DNS

También existirá la posibilidad de migrar direcciones DNS regulares al RNS. El propietario de una dirección DNS podría reclamar un dominio en el RNS mediante el uso de oráculos para verificar que es el propietario legítimo del dominio original. En caso de colisión con el dominio de nombre ICANN, se utilizará un sistema de arbitraje para resolver los conflictos. Dicho sistema de arbitraje podría implementarse a través de oráculos y otros métodos.

## Anonimato del resolver

Los usuarios pueden querer ocultar la dirección a la que se asignan sus dominios. Esto se puede hacer con recursos encriptados. El propietario puede transferir la clave de descifrado a un usuario a través de un canal de comunicación fuera de la cadena. Además, las direcciones almacenadas pueden ser encubiertas, por lo que el remitente debe generar una nueva dirección para cada pago independiente.

## Crear un nuevo top-level registrar

RSK Labs proporciona un registrar inicial para TLD (.rsk). En el futuro, los usuarios pueden crear su propio TLD implementando sus propios registrars. El usuario puede habilitar un proceso de subasta (o un proceso diferente) para permitir que las personas adquieran subdominios.

# Referencias

[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