# RSS3: A Next-Generation Feed Standard

Natural Selection Labs

***Abstract*—Inspired by the original RSS Standard, this paper presents RSS3, the next-generation feed standard, that aims to support efficient and decentralized information distribution. Furthermore, the RSS3 Protocol is proposed in conjunction with a sophisticated network structure to implement the standard. The result is a fully decentralized network that is flexible, efficient, and extensible.**

## I. Introduction

The original RSS Standard is open, neutral, and decentralized in nature. In the Web 2.0 era, the adoption of RSS dropped significantly as centralized platforms gained traction. However, a centralized platform, where a single trusted party dominates the network, inevitably results in problems such as privacy infringement, profit misappropriation, censorship, algorithm abuse, and data monopoly. Some innovative solutions have been proposed to tackle these problems: 1) Federated networks enable users to choose a trusted center while still supporting communications among centers; 2) Blockchain-based networks allow user data to be distributed across all nodes. To some extent, these solutions manage to solve certain problems, but with the limitations of flexibility, efficiency, and extensibility.

As the Web moves toward more openness and modularization, the pipeline of information flow is evolving to include four decentralized layers including creation, storage, distribution, and rendering. Various protocols or modules are expected to support an interoperable communication that can be fully controlled by users. Existing solutions do not form a fully decentralized information distribution standard that is urgently demanded by the Web.

In this paper, we propose the RSS3 Standard, a next-generation feed standard that enables efficient and decentralized information distribution with flexibility, efficiency, and extensibility. The standard ensures that the network will be financially self-sustained with redundancy and fault tolerance. We then introduce the RSS3 Protocol with a sophisticated network structure to implement the standard. Multiple measures are implemented to ensure the network's stability and security.

## II. glossary

This section includes the glossary Table I contains a list of terms used throughout the paper.

| Term | Definition |
|---|---|
| DAO | Decentralized Autonomous Organization, an automated decision-making organization that is agreed by the network and is not influenced by a central authority. DAO oversees all important decisions. |
| SN | A serving node, which hosts RSS3 Files and responds to file-related requests. DAO sets a limit of files that a single SN can serve. |
| Subgroup | A group of SNs, which consists of a number of SNs. DAO sets a limit of SNs that a single subgroup can have. |
| GI | A global indexer, which orchestrates subgroups, routes client requests, and maintains the network capability. DAO sets the incentive scheme. |
| RN | A relay node, as part of a GI to assist in routing. |
| Epoch | A reference point in time where multiple events are scheduled to be triggered. |
| ER | Epoch round, a period of time between two epoch points where the network operates without any planned events from DAO and GIs. DAO sets the ER interval. |
| SDG | Scalable Dynamic Grouping, the strategy used by GIs to orchestrate subgroups that will be executed at the epoch. |
| MOC | Minimal operational capability, the required number of GIs or SNs in a subgroup, to provide trustworthy services. |
| HBC | Heartbeat checking, a strategy used by nodes to report their working status. |

TABLE I: A glossary table that contains the terms used throughout the paper.

## III. Standard

Compared to its predecessor, RSS3 introduces fundamental changes to create an efficient and decentralized information distribution standard.

### A. Instance

An instance is a collection of cryptography based accounts owned by one cyber existence. Upon registration, the first address submitted will be used as the instance's primary address. Additional accounts can be imported by auto-verification or user actions. All verified accounts will be stored in an RSS3 File.

### B. Asset

An asset refers to a medium of exchange owned by the instance that is digitally generated. An RSS3 network automatically incorporates instance assets from verified accounts, and stores them into the corresponding RSS3 File. DAO decides the rendering of assets in RSS3 Files as there are wide variations in types.

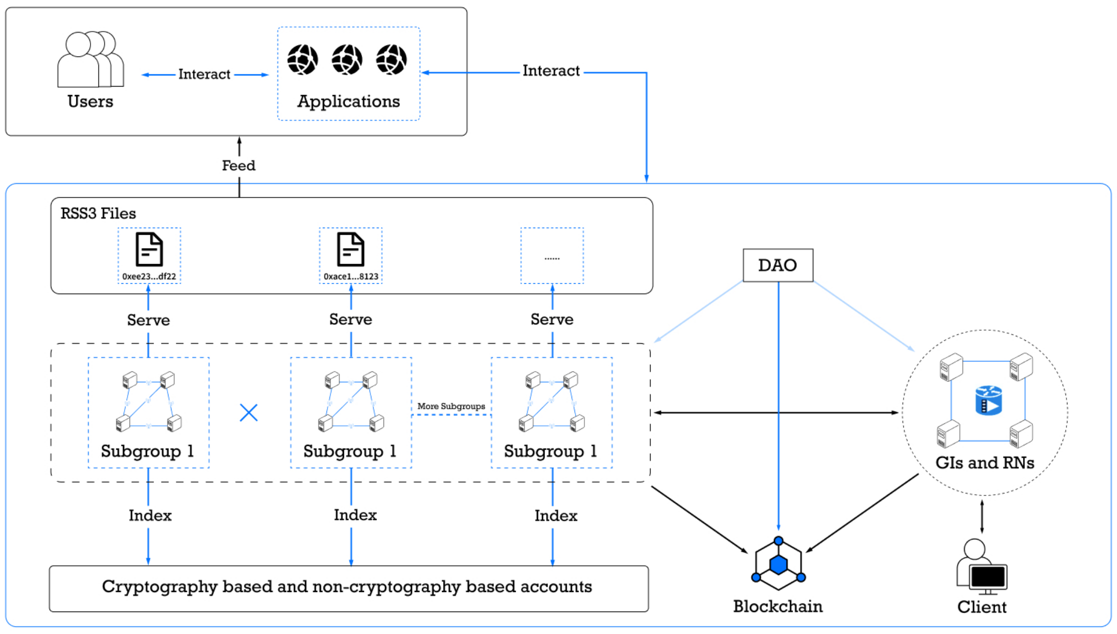

Fig. 1: A typical RSS3 Network architecture.

### C. Link

Ubiquitous linking is the foundation of an open information system. The RSS3 Protocol supports universal links with customized types between RSS3 Objects. Major RSS3 Objects include:

- Instance - a collection of cryptography based accounts
- Asset - a medium of exchange that is digitally generated
- Item - content generated on the network

An internal link is bidirectional which connects two objects within an RSS3 Network, whereas an external link unidirectionally connects an RSS3 Object with an external object. An external link destined for an RSS3 Object may be bidirectional if DAO approves the source.

### D. Governance and Ownership

RSS3 network is equipped with a high Turing completeness. It is capable of handling sophisticated logic such as reviewing smart contracts that define permission and ownership of RSS3 Files.

### E. Activity Feed

In an RSS3 File, an activity feed contains: 1) all activities indexed from the instance's verified accounts; 2) all items generated by the instance or the network; 3) all universal links generated on the RSS3 Network. The feed resembles the original RSS protocol with additional cryptographic information to support data integrity and originality. The design represents a decentralized approach for information distribution.

Based on the standard described above, we present the RSS3 Protocol, available at https://github.com/NaturalSelectionLabs/RSS3. The implementation is by no means a perfect solution to the problems which the web is facing now, but a significant leap forward toward a truly decentralized cyberspace.

## IV. Network

Fig. 1 shows an abstraction of a typical RSS3 Network architecture. In short, an RSS3 file is served by a subgroup which consists of several SNs. A subgroup is managed by GIs and forwards client requests to its SNs. A GI consists of: 1) a set of relay nodes to assist in routing; 2) a set of archive modules to archive the network for failure recovery in the event of a large-scale network crash.

### A. Decentralized Autonomous Organization (DAO)

DAO is responsible for governing the followings:

- GI and SN Elections
- RSS3 Files limit per SN
- DKG key fragments threshold
- Subgroup scaling
- Module upgrade
- Incentive Management
- ER Duration

*1) Election Mechanism:* GIs and SNs are elected members that support the network's essential operations. There are entry requirements for candidates to be qualified for participating in an election: These requirements may include hardware and staking set by DAO. GI elections are held less frequently than SN elections. Each election may produce an abundance of nodes and only the required number of nodes will be immediately placed into work. $S = \{s_1, s_2, ..., s_{\mathcal{S}}\}$ is the set of all SNs, whose number is $\mathcal{S}$. $G = \{g_1, g_2, ..., g_{\mathcal{G}}\}$ and is denoted as the set of all GIs, whose number is $\mathcal{G}$.

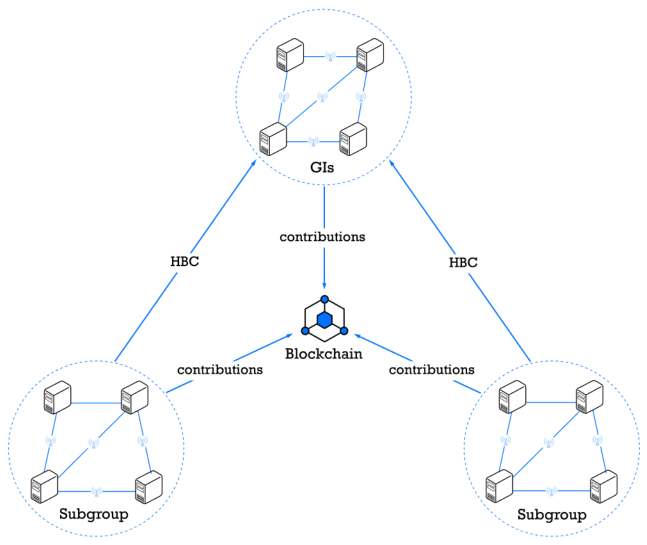

Fig. 2: HBC and Proof on chain.

To further strengthen the network, a waiting list $S'$ is in place to host idle elected SNs as backup. DAO sets a MOC warning level, $\mathcal{MOC}(m)$, backup SNs will be commissioned once the number of functioning SNs falls below $\mathcal{MOC}(m)$ to avoid subgroup failures.

*2) Module Upgrade:* WebAssembly modules are used by SNs for indexing to ensure that indexing results are cross-platform deterministic. Modules can be contributed by the network participants and are governed by DAO.

*3) Subgroup Scaling:* During each ER, DAO may dynamically adjust the size of subgroups, denoted $\mathcal{N}$, to meet the network's needs while maintaining a sufficient level of decentralization. Hence, the number of subgroups $\mathcal{Y} = \lceil \frac{\mathcal{S}}{\mathcal{N}} \rceil$ is dynamic and decided by $\mathcal{S}$ and $\mathcal{N}$. This adjustment predominately ensures the network's performance and stability.

*4) Incentive Management:* Within each ER, DAO may dynamically adjust the incentive ratio for network participants. Incentive is calculated based on multiple factors, including verified contributions from GIs and SNs on chain , as shown in Fig. 2. All incentives are distributed from an incentive pool described in Sec. V-C. A staking pool is governed by DAO, each candidate will be notified of the level staking required to participate in the election. Slashing occurs when an elected node failed to meet its obligations outlined in Sec. IV.

## *B. Global Indexer (GI)*

GIs are elected by the network participants through DAO. A GI has three major responsibilities: Scalable Dynamic Grouping, Routing and Heartbeat Checking. GIs are assisted by functional nodes listed in Table I, the details are elaborated below in this section.

A GI at any time may announce its intention to quit the network and gracefully do so at the end of the current ER without slashing. If the MOC is reached, DAO will hold a new round of GI election immediately to avoid a potential network failure.

*1) Scalable Dynamic Grouping (SDG):* At the beginning of each ER, SNs are shuffled into different subgroups $S_z; \forall z \in [1, \mathcal{Y}]$ abiding by a modern Byzantine fault tolerant (BFT) algorithm[2, 5, 6, 8], to fundamentally prevent collusion attacks. The organization of subgroups is periodically shuffled by GIs through a consent structure, and broadcast to SNs to initiate the re-organization. Subgroups are scalable to incorporate newly available SNs to extend their capability, up to the limit $\mathcal{N}$ imposed by DAO. The number of subgroups is dynamically scaled by the network's existing load based on the strategy agreed by DAO.

To further elaborate on how a consent structure is agreed to, SDG begins with a random number calculated by collecting signatures from GIs. Based on the distributed key generation protocol Ped-DKG[4] and Threshold BLS Signatures[1], each GI contributes a key fragment as one constituent to form a public and private key pair, the random number is then derived from all the signatures collected, where the DKG key fragments threshold set by DAO is met to guarantee security. SDG proceeds to shuffle SNs and forms a new subgroup structure until not a single subgroup contains $\mathcal{P}$ SNs from the previous ER:

$$\mathcal{P} = \lfloor \frac{\mathcal{N} - 1}{3} \rfloor; \forall \mathcal{N} \in \mathbb{N}^+ \tag{1}$$

GI members will be dynamic, allowing GI to join and leave at the next ER. The change of GI members will affect the generation of distributed random numbers, as it depends on the key fragmentation of each GI member. Proactive secret sharing protocol is adopted[7, 12], which allows a group of GI members to keep the shared key and dynamically transfer the key fragments among different GI members. So as to achieve such a result, that the keys shared among the GI member group fragments are saved in different groups.

In order to minimize the chance of node collusions, at the $e^{th}$ ER, GIs ensure that during SDG:

- The subgroup set $\{S_z\}$ is generated by SDG, feeding all SNs as input based on parameter $\mathcal{P}$ and $\eta^e \in \mathbb{N}^+$:

$$\{S_z^e\} = \mathcal{SDG}(< \{s_1^e, s_2^e, ..., s_{\mathcal{S}}^e\} >; \mathcal{P}, \eta^e, \mathcal{Y}^e) \tag{2}$$

- The intersection of all subgroups is an empty set:

$$\bigcap_{z=1}^{\mathcal{Y}^e} S_z^e = \varnothing \tag{3}$$

- The union of all subgroups is the set of all SNs:

$$\bigcup_{z=1}^{\mathcal{Y}^e} S_z^e = S^e \tag{4}$$

- Any $\mathcal{P}$ SNs in the $z^{th}$ subgroup $S_z^e$ will not be grouped into the same subgroup in $\eta^e$ consecutive ERs:

$$|S_z^e \cap S_v^{e-j}| < \mathcal{P}; \forall z, v \in [1, \mathcal{Y}^e], \forall j \in [1, \eta^e] \tag{5}$$

$$s_{z,i}^e \notin S_z^{e-j}; \forall s_{z,i}^e \in S_z^e, \forall i \in [1, |S_z^e|] \tag{6}$$

where $S_z^e$ denotes the set of SNs in the $z^{th}$ subgroup at the

**Algorithm 1** $\mathcal{SDG}(< S^e >; \mathcal{P}, \eta^e, \mathcal{Y}^e)$

```
1:  while $\mathcal{SDG} \leftarrow fail$ do
2:      for $s^e_{z,i} \in S^e_z; z \in [1, \mathcal{Y}^e]; i \in [1, |S^e_z|]$ do
3:          for $j \in [1, \eta^e]$ do
4:              if $(|S^e_z \cap S^{e-j}_v| < \mathcal{P}) \wedge (s^e_{z,i} \notin S^{e-j}_z)$ then
5:                  $\mathcal{SDG} \leftarrow success$
6:                  return $\{S^e_z\}$
7:              end if
8:          end for
9:      end for
10: end while
```

$e^{th}$ ER, $s^e_{z,i}$ refers to the $i^{th}$ SN in the $z^{th}$ subgroup at the $e^{th}$ ER, and both $\mathcal{P}$ and $\eta$ are set by DAO. More details regarding the procedure of SDG are presented in Algorithm 1.

*2) Robust Routing:* GIs maintain a routing table, which follows the SDG at each ER. GIs are walled by a set of RNs. Following the routing table, RNs accept and forward user requests to the corresponding subgroup.

*3) Heartbeat Checking (HBC):* GIs make the best effort to guarantee a subgroup's MOC, see Eqn. 7 and 8, where $n$ denotes the number of SNs, $\mathcal{W}_s$ is set by DAO, satisfying $\mathcal{C} < \mathcal{W}_s < \mathcal{N}$, and $\mathcal{C}$ is the minimum number of SNs supporting the system's MOC. Once the warning level is breached, GI will start enlisting SNs from the waiting list $S'$. Once the critical level is breached, GI will immediately start a new round of partial SDG to avoid a potential subgroup failure. A partial SDG re-balances current subgroup in order to satisfy MOC without interrupting network operations.

$$\mathcal{C} = \lceil \frac{2\mathcal{N}}{3} \rceil \tag{7}$$

$$\mathcal{MOC}(n) = \begin{cases} safe, & if\ n > \mathcal{W}_s; \\ warning, & if\ \mathcal{C} < n \leq \mathcal{W}_s; \\ critical, & if\ n \leq \mathcal{C}. \end{cases} \tag{8}$$

Furthermore, a new GI election would start immediately as the number of GIs, namely $m$, reaches the warning level $\mathcal{W}_g$, which is set by DAO. See Eqn. 9.

$$\mathcal{MOC}(m) = \begin{cases} safe, & if\ m > \mathcal{W}_g; \\ warning, & if\ m \leq \mathcal{W}_g; \end{cases} \tag{9}$$

### C. Serving Node (SN)

Fig. 3 illustrates the internal architecture of a subgroup.

A list of elected SNs, along with their permitted serving ERs, is governed by DAO. Candidates $s'$ who were elected but not commissioned are placed on a waiting list $S'$, see Sec. IV-A1.

An exit mechanism is in place to permit the departure of SNs from their subgroups during an ER without slashing: 1) the departure must be broadcast to the subgroup to confirm that the MOC level, denoted $\mathcal{C}$, is maintained; 2) the departure must not occur consecutively within $\eta$ ERs, where $\eta$ is set by DAO.

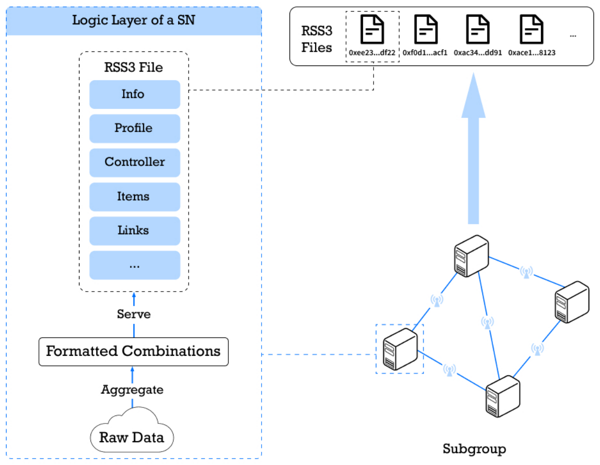

Fig. 3: The internal architecture of a subgroup.

*1) Information Aggregation:* RSS3 files of the entire network are distributed across many subgroups. Each SN serves a number of RSS3 files and handles the requests targeted at these files according to the standard specifications. SNs use WebAssembly modules for indexing, which ensures that indexing results are cross-platform deterministic. SNs begin with indexing data from verified accounts, perform permission and ownership verification, and eventually generate the instance's activity feeds.

*2) Collaboration:* SNs collaborate within of the same subgroup to reach an internal consensus for finalizing the content of RSS3 Files and acknowledging peer contributions for HBC. SNs are obligated to broadcast their HBC back to GIs for next ER's preparation and incentive distribution. Absence from this process will result in the following consequences for the absentees : 1) to be removed from the incentivization of this ER; 2) removed from the next ER's SDG; 3) to be slashed. At the end of each ER, those HBCs are packed and reported on chain, as the proof of contributions.

Algorithm 2 shows the algorithmic abstraction of an ER in the RSS3 Network, and Fig. 4 illustrates the flow of the abstraction.

## V. Incentive

Proper incentive is essential for a decentralized network to provide a satisfactory level of service. To ensure that the incentive scheme is transparent, sustainable, and fair to all network participants, DAO is put in charge to dynamically adjust incentives when required.

Current successful incentive schemes are predominantly found in blockchain-based networks: all nodes are required to verify transactions to be rewarded by "transaction fees". The transaction-based model is proven to be functional in mostly financial-related fields, but is expected to struggle under other settings. Take social media as an example, it is unfeasible to

**Algorithm 2** An algorithmic abstraction of an ER in the RSS3 Network.

```
1:  for each ER $e \in [0, \infty)$ do
2:      init. $\mathcal{G}^e, \mathcal{S}^e, \mathcal{N}^e, \mathcal{W}_s^e, \mathcal{W}_g^e, \eta^e$ by DAO
3:      $\mathcal{Y}^e \leftarrow \frac{\mathcal{S}^e}{\mathcal{N}^e}$
4:      if election then
5:          $G^e \leftarrow \{g_1^e, g_2^e, ..., g_{\mathcal{G}^e}^e\}$
6:      end if
7:      $\{S_z^e\} \leftarrow \mathcal{SDG}(< \{s_1^e, s_2^e, ..., s_{\mathcal{S}}^e\} >; \mathcal{P}, \eta^e, \mathcal{Y}^e)$

8:      for $S_z^e \in \{S_1^e, S_2^e, ..., S_{\mathcal{Y}^e}^e\}$ do
9:          $n \leftarrow |S_z^e|$
10:         switch $(\mathcal{MOC}(n))$
11:         case $n > \mathcal{W}_s^e$:
12:             pass
13:         case $\mathcal{C} < n \leq \mathcal{W}_s^e$:
14:             $S_z^e \leftarrow {s'}_k^e; \forall {s'}_k^e \in {S'}^e$
15:         case $n \leq \mathcal{C}$:
16:             ${S'}^e \leftarrow s_l^e; \forall s_l^e \in \{\complement_{S^e} S_z^e\}$
17:             $\hat{S}_z^e \leftarrow \mathcal{SDG}(< {S'}^e >; \mathcal{P}, \eta^e, \mathcal{Y}^e)$
18:     end for

19:     $m \leftarrow |G^e|$
20:     switch $(\mathcal{MOC}(m))$
21:     case $m > \mathcal{W}_g^e$:
22:         pass
23:     case $m \leq \mathcal{W}_g^e$:
24:         $\hat{G}^e \leftarrow \{\hat{g}_1^e, \hat{g}_2^e, ..., \hat{g}_{\mathcal{G}^e}^e\}$

25:     consensus number $c \leftarrow 0$
26:     for $g_z^e \in G^e \vee \hat{g}_z^e \in \hat{G}^e$ do
27:         if $g_z^e \vee \hat{g}_z^e$ agrees then
28:             $c \leftarrow c + 1$
29:         end if
30:     end for
31:     if $c > \lceil \frac{2\mathcal{G}^e}{3} \rceil$ then
32:         submit proof on chain
33:     end if

34:     init. and adjust $P$ and $\Phi$ by DAO
35:     $P \leftarrow \{p_{nod}, p_{dev}, p_{cre}, p_{con}, p_{dao}\}$
36:     $\Phi \leftarrow \{\alpha, \beta, \gamma, \delta, \epsilon\}$
37:     for $i \in [1, 5]$ do
38:         $P_i \leftarrow \Phi_i \mathcal{I}$
39:     end for
40: end for
```

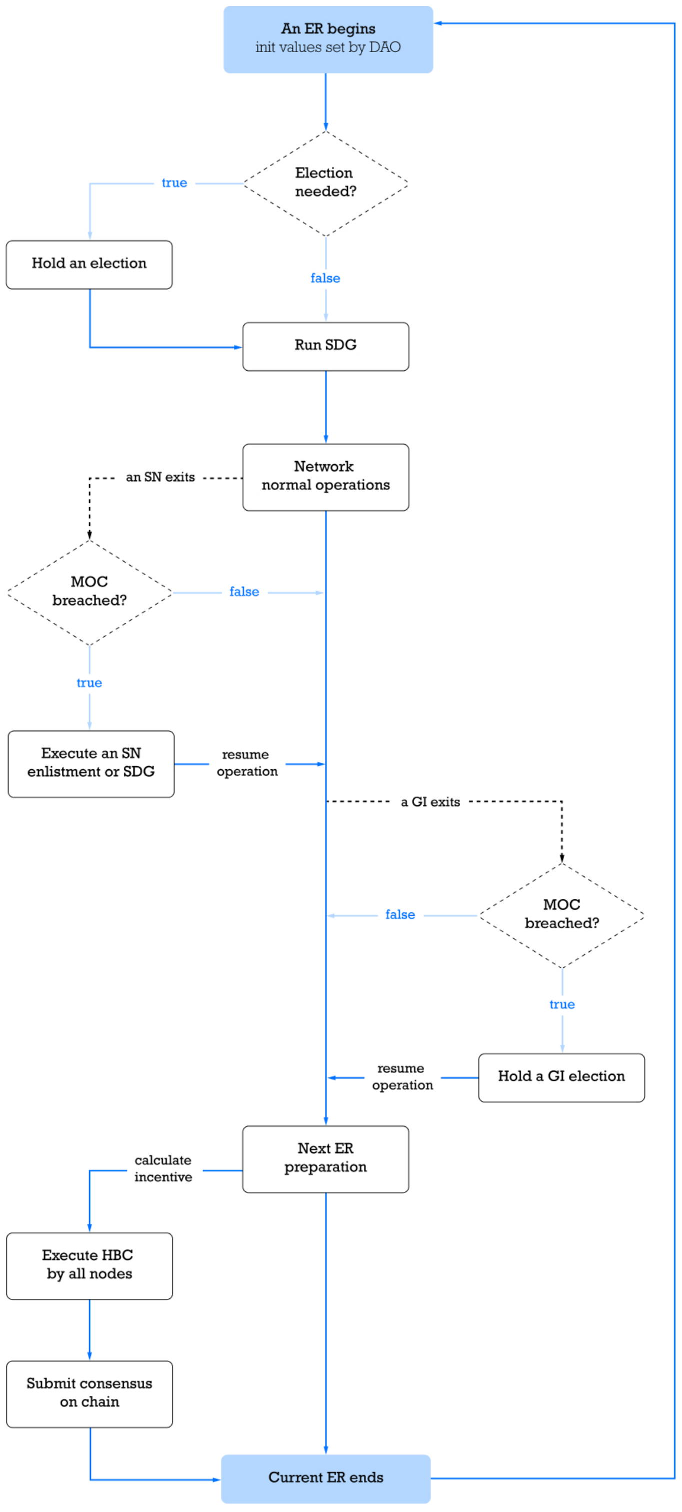

Fig. 4: A flowchart illustrating the abstraction of an ER in the RSS3 Network.

charge users for interactions performed. Nor should developers be paying for the cost of maintaining a decentralized network.

### A. *Incentivization*

The RSS3 Network, on the other hand, will be rewarding network participants with the profit of the network generated from advertising, value-added services, social economic activities, etc.

- **Stage 1: System Incentivization** At the beginning, the network is rewarded by the system to encourage adoption. The system reward gradually decreases and is inversely proportional to network adoption.

- **Stage 2: Hybrid Incentivization** As network adoption increases, profits are expected to be generated by activities such as advertisements, value-added services and other related economic activities. The proceeds will be distributed in the form of network rewards, which also offsets the system rewards.
- **Stage 3: Self-Sustained Incentivization**: All rewards are now fully transitioned into network-generated rewards, the system will no longer incentivize the network.

Incentives will be distributed to network participants including node hosts, developers, content creators, special contributors and DAO.

### B. Staking and Slashing

Upon election, GIs and SNs are required to join the staking pool governed by DAO. Staking demonstrates the participant's commitment toward maintaining the network.

Consequently, slashing as a means to maintain the network's stability occurs when malicious or erroneous behaviors are detected, in addition to situations described in Section IV-B and IV-C.

### C. Incentive Pool

An incentive pool $\mathcal{I}$ is introduced to dynamically balance the incentives distributed to network participants, see Eqn. 10, 11, and 12, where the proportions set $\Phi = \{\alpha, \beta, \gamma, \delta, \epsilon\})$ is decided by DAO; The participants set $P = \{p_{nod}, p_{dev}, p_{cre}, p_{con}, p_{dao}\}$ includes node hosts, developers, content creators, special contributors and DAO respectively. This will ultimately foster a self-adjusting market and avoid skewed incentives in the distribution: Incentive increases as the demand for a role increases and vice versa. A proper allocation of the incentive pool greatly motivates the network participants in improving all aspects of the network.

$$sum(\alpha, \beta, \gamma, \delta, \epsilon) = 1 \tag{10}$$

$$\begin{cases} p_{nod} = \alpha\mathcal{I} \\ p_{dev} = \beta\mathcal{I} \\ p_{cre} = \gamma\mathcal{I} \\ p_{con} = \delta\mathcal{I} \\ p_{dao} = \epsilon\mathcal{I} \end{cases} \tag{11}$$

$$\mathcal{I} = p_{nod} + p_{dev} + p_{cre} + p_{con} + p_{dao} \tag{12}$$

## VI. Scalability

As a network grows, the performance bottleneck often results in a slow processing speed and a higher transaction cost. In previous sections, subgroup scaling (Sec. IV-A3) and SDG (Sec. IV-B1) are described as measures to maintain network's availability and usability, they are also designed to improve both storage and communication efficiency.

### A. Storage Efficiency

Storage efficiency is critical for a decentralized network. First of all, RSS3 Files are lightweight by design (since they contain the metadata only). DAO limits the number of RSS3 Files hosted by an SN and dynamically increases the number of subgroups through subgroup scaling. This strategy provides a sufficient level of data redundancy while maintaining storage efficiency.

### B. Communication Efficiency

In any decentralized network, an increase in efficiency leads to a reduction in fault tolerance. Extensive research has been done to ensure that the RSS3 Network is able to maintain an ideal equilibrium. As opposed to a blockchain based network[10, 14], where a global consensus is required, an RSS3 Network only requires each subgroup to reach an internal consensus. This significantly reduces the communication complexity. As the number of RSS3 Files increases, DAO dynamically scales the network through the election mechanism and subgroup scaling in conjunction with SDG performed by GIs, to further reduce communication complexity for subgroups.

GIs need to reach an internal consensus from time to time - such a consensus only requires an aggregated signature to reduce communication complexity. Furthermore, state-of-the-art BFT algorithms are implemented to maximize communication efficiency[9, 13].

More subgroups will inevitably overwhelm GIs. On top of scaling through the election mechanism, this is additionally mitigated by increasing the number of RNs to take the workload of client request handling (where no consensus is needed) off GIs' shoulders. DAO further imposes a limit on the maximum number of GIs to improve communication efficiency.

## VII. Vulnerability

### A. Collusion

In an RSS3 Network, RSS3 Files are hosted by subgroups of SNs for redundancy and fault tolerance. Although the subgroup design creates the bedrock for potential collusive behaviors, the design itself also acts as the first layer to prevent those behaviors since an internal consensus is required to process all client requests. A client request, as illustrated in Fig. 1, is forwarded from GIs to SNs and returned back to GIs consequently. GIs then return the most consistent result back to the client. Furthermore, a collusion will not have any impact on the result when the number of malicious or erroneous nodes, denoted $\mathcal{M}$, where $\mathcal{M} \leq \mathcal{P}$.

The probability of a collusion taking place, though slim, still exists in theory. GIs minimize this probability through breaking potential $\mathcal{M}$ via SDG, as described in Sec. IV-B1. SNs within the same subgroup are less likely to be grouped together in the future to prevent possible collusion.

Unlike blockchain-based networks, the collusion in the RSS3 Network is not economically profitable, which eliminates the primary motivation behind collusion[11]. In the event of a collusion, owners can correct tampered data at any time and report SNs for malicious behaviors. Collusive SNs lose their incentives and seats in the network. Since staking is required, slashing also serves as a deterrent to potential colluders.

## B. Redundancy

Subgrouping provides redundancy as SNs: 1) maintain RSS3 Files assigned collaboratively, effectively creating $\mathcal{S}$ copies for redundancy; 2) are with best efforts distributed across the globe to provide geo-redundancy and sustain a high availability.

During the election, DAO selects GIs from multiple regions to provide geo-redundancy.

## C. Disaster Recovery

To recover from an unlikely event of a complete network failure, where more than $\mathcal{R}_g$ GIs or $\mathcal{R}_s$ SNs have failed to stay functional, see Eqn. 13, DAO members may operate archive modules that constantly take snapshots of the entire network. GIs are also encouraged to operate archive modules.

$$\mathcal{R}_{(g \vee s)} = \lfloor \frac{(\mathcal{G} \vee \mathcal{S})}{3} \rfloor ; \forall (\mathcal{G} \vee \mathcal{S}) \in \mathbb{N}^+ \tag{13}$$

## D. Sybil Attack

Sybil attacks may occur in any network, where a disproportionate level of network resources are allocated to attackers, affecting the network's availability. Research shows that there is currently no universally applicable solution to Sybil attacks[3]. In the RSS3 Network, such an attack does not generate any financial gains. Furthermore, identity validation is implemented to minimize the chance of Sybil attacks. The network also increases the economic costs for such an attack.

## VIII. Conclusion

We present the RSS3 Standard, a next-generation feed standard designed for information distribution. The standard enforces strong data ownership by including asset verification and on chain governance, and furthermore equips the feed with native support for interoperable interactions. The standard naturally enables flexible recommendation and searching implementation. We then propose the RSS3 Protocol that is fully conformed to the standard. In comparison to the original RSS, RSS3 is a refined and evolved protocol that supports complicated application scenarios.

To further illustrate the robustness of the standard, we demonstrate an RSS3 Network that implements multiple state-of-the-art measures to ensure data consistency and redundancy. This is achieved through: 1) a sophisticated network architecture that guarantees the network usability and stability; 2) a curated incentivization scheme that encourages all network participants to contribute toward the network's development. The orchestration has the potential to provide a truly decentralized network, that is, on top of all the benefits provided by existing solutions, with flexibility, efficiency, and extensibility.

**At the heart of Natural Selection Labs, we firmly believe in the freedom of information distribution: No organizations or authorities shall prohibit the free exercise of the right of people to create, store, and distribute their information.**

## References

[1] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, Lecture Notes in Computer Science, pages 514–532, Berlin, Heidelberg, 2001. Springer.

[2] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, 20(4):398–461, November 2002.

[3] John R. Douceur. The Sybil Attack. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, volume 2429, pages 251–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[4] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure Applications of Pedersen's Distributed Key Generation Protocol. In Marc Joye, editor, *Topics in Cryptology — CT-RSA 2003*, Lecture Notes in Computer Science, pages 373–390, Berlin, Heidelberg, 2003. Springer.

[5] Jae Kwon. Tendermint: Consensus without Mining. page 11.

[6] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.

[7] Sai Krishna Deepak Maram, Fan Zhang, Lun Wang, Andrew Low, Yupeng Zhang, Ari Juels, and Dawn Song. CHURP: Dynamic-Committee Proactive Secret Sharing. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 2369–2386, New York, NY, USA, November 2019. Association for Computing Machinery.

[8] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The Honey Badger of BFT Protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 31–42, New York, NY, USA, October 2016. Association for Computing Machinery.

[9] Henrique Moniz. The Istanbul BFT Consensus Algorithm. February 2020.

[10] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. page 11, 2008.

[11] Thibault Schrepel. Collusion By Blockchain And Smart Contracts. *SSRN Electronic Journal*, 2019.

[12] David Schultz, Barbara Liskov, and Moses Liskov. MPSS: Mobile Proactive Secret Sharing. *ACM Transactions on Information and System Security*, 13(4):34:1–34:32, December 2010.

[13] Chrysoula Stathakopoulou, Tudor David, Matej Pavlovic, and Marko Vukolić. Mir-BFT: High-Throughput Robust BFT for Decentralized Networks. June 2019.

[14] Gavin Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger. page 32, 2014.