SORA Token

The Token for Sora Terminal, the KYC Wallet

SORA Token White Paper v2.5.0

Table of Contents

The Death of the Public Sale

The ICO is dead. Where once one could freely sell tokens and cryptocurrencies to the public without fear of litigation, it is now almost impossible to do so. While private sales have filled the funding gap for developing blockchain technologies, the lack of public access to purchase crypto currencies has led to an alarming trend of centralization throughout the blockchain space--many well-funded coins with market caps in the millions of dollars have only tens of holders. At its core, this state of increasing centralization threatens the safety of technologies which take advantage of decentralized trust mechanisms and leads to speculation and price volatility for investors. The validity of any blockchain community is impossible to verify, making it difficult for exchanges and investors to justify the legitimacy behind every project. Where community was once built through public sales, it is now built by third-parties specialized in funnelling fake accounts to project group chats and social media platforms, making it even more difficult for newcomers to build trust in our industry. This issue is the very core reason to why blockchain and digital asset companies are still vulnerable to being classified as fraud. The Sora Terminal and Sora KYC token is an attempt to relish this challenge. The team is built by industry investors, experts, and true believers in hopes of better filtering out fraudulent projects in our industry using data, compliance, and blockchain technology.
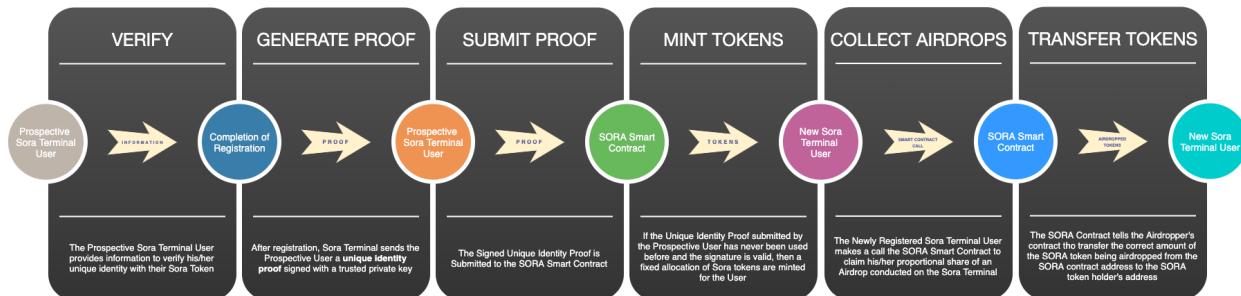
An Imperfect Solution

To grapple with increasing centralization, many projects discovered that through marketing to target audiences including exclusive news and development of the projects, preferential access to future invitations, exclusive deals and simply giving away small amounts of tokens for free to the public ('**Airdropping'**) increases interest and PR, and ultimately leads to both faster and more sustainable growth of one's project. However, projects often find it challenging to locate their specific target audiences for marketing purpose. Airdrops, likewise, also came with their own logistical and legal challenges, and this has made Airdrops an impractical solution for many blockchain projects trying to grapple with solving the centralization problem. At the same time, many investors have shied away from using airdrop results as a benchmark for community health due to concerns about unique address ownership; a single person can own many addresses on a blockchain platform, so while a project's token may be held by 1000 addresses after an airdrop, those 1000 addresses might be owned by only a handful of people, making it difficult to validate the legitimacy of a community.

Community Validation 2.0: Enter Sora Terminal

In an effort to address the challenges of doing marketing and centralization and community airdrops, we introduce the *Sora Terminal*: a distributed KYC wallet system that uses the SORA token, a token that is minted every time a user signs up, to enable the smart contract owners across many blockchains to cultivate their KYC community and distribute a portion of their token to a large network of real verified users. With Sora Terminal, blockchain projects don't need to spend money on costly and increasingly controversial advertising, or worry about the logistical and legal issues of conducting an airdrop on their own; all one needs to do is open the Sora Terminal client website, transfer tokens to the *SORA* smart contract, and watch as a network of engaged blockchain KYC users begin growing the project's KYC community in exchange for their tokens.

Community Validation: The Flow

VERIFY	GENERATE PROOF	SUBMIT PROOF	MINT TOKENS	COLLECT AIRDROPS	TRANSFER TOKENS
Prospective Sora Terminal User — INFORMATION →	Completion of Registration — PROOF →	Prospective Sora Terminal User — PROOF →	SORA Smart Contract — TOKENS →	New Sora Terminal User — SMART CONTRACT CALL →	SORA Smart Contract — AIRDROPPED TOKENS → New Sora Terminal User
The Prospective Sora Terminal User provides information to verify his/her unique identity with their Sora Token	After registration, Sora Terminal sends the Prospective User a **unique identity proof** signed with a trusted private key	The Signed Unique Identity Proof is Submitted to the SORA Smart Contract	If the Unique Identity Proof submitted by the Prospective User has never been used before and the signature is valid, then a fixed allocation of Sora tokens are minted for the User	The Newly Registered Sora Terminal User makes a call the SORA Smart Contract to claim his/her proportional share of an Airdrop conducted on the Sora Terminal	The SORA Contract tells the Airdropper's contract tho transfer the correct amount of the SORA token being airdropped from the SORA contract address to the SORA token holder's address

The process from joining the Sora Terminal to claiming tokens to validating a community takes just 6 easy steps:

1. The Prospective User provides information to register on Sora Terminal in exchange for Sora Token.

2. The Terminal sends the Prospective User a unique identity proof signed with the private key.

3. The Signed Unique Identity Proof is Submitted to the SORA Smart Contract running on either the EOS or Ethereum Blockchain.

4. If the Unique Identity Proof submitted by the Prospective User has never been used before and the signature is valid, then a fixed allocation of tokens is allocated for the User from the locked portion of Sora Token owned by the Sora Token contract.

5. The new Sora Terminal user makes a call to the SORA Smart Contract to declare his interest in the project to receive marketing materials including development of the projects, preferential access to future invitations, and a potential token distribution conducted on the SORA smart contract into the Sora Terminal.

6. The SORA Contract tells the project's smart contract to send the marketing materials to the SORA Token holder and transfer the correct amount of the token/fee being distributed from the SORA contract address to the SORA Token holder's address.

Sora Terminal and SORA: A Deeper Look

At its base, SORA smart contract is a permission-less smart contract that lives on the Ethereum and EOS blockchain networks implementing the ERC20 and eosio.token standards respectively. When a blockchain project conducts a community validation into the Sora Terminal using the SORA smart contract, all holders of SORA Tokens can claim a portion of the tokens that are sent to the SORA smart contract. The number of tokens distributed to each individual address in the Sora Terminal will be standardized. The majority of the logic found in the SORA smart contract is dedicated to the four major functional components of the SORA platform: The Airdrop Submissions Process (ASP), the User Verification and Allocation Mechanism (UVAM), the Airdrop Claims Process (ACP), and the [Token Supply] Control Mechanism (FCM). In Table 1 below, you will find a brief summary of these four functions and how they help the SORA team achieve their mission to making airdrops painless for users and blockchain projects alike.

Table 1

Functional Component	Function Description	Benefit to the Platform
Airdrop Submission Process (ASP)	A process by which blockchain projects can submit their token for airdrop distribution using SORA	Makes conducting an airdrop as simple as a smart contract call in one line of solidity, or one button press from the SORA Terminal.
User Verification and Allocation Mechanism (UVAM)	A mechanism by which users can generate a unique proof of identity signed by one of our real name identity verification partners, and verifiable on chain using RSA	Allows the smart contract to ensure both that each address which holds SORA Tokens is a unique person, and that the number of total tokens in circulation is directly proportional to the number of unique users on the network
Airdrop Claims Process (ACP)	A process by which holders of SORA Tokens can claim tokens airdropped on SORA Terminal	Provides a means for distributing tokens which is scalable as the number of SORA Token holders grows
[Token Supply] Control Mechanism (TSCM)	A mechanism which allows the foundation to mint the tokens allocated to each new KYC user.	Encourage early adopters to mint and allows for the ecosystem to be based on a fixed amount of token.

Community Validation Process in Sora Terminal

Summary: A process by which blockchain projects can submit their token for community validation by sending their tokens to the SORA smart contract and then redistributed to KYC users in the Sora Terminal.

Sora Terminal makes community validation easy, and the simple process that encapsulates everything from start to successful airdrop is called the "*Airdrop Submission Process*" or "***ASP***".

Community validation on the *Sora Terminal* platform only requires 3 steps to launch:

1. Transfer the tokens you want to airdrop to the address of the *SORA* smart contract (See ASP.1.1 in figure FLOW.ASP.1)

2. Open the Sora Terminal and submit the "Launch Airdrop" form in the *SORA* client by entering your Ethereum smart contract address or EOS contract name in the GUI. (See ASP.1.1 in figure FLOW.ASP.1)

3. Check the number of increased token holders and activity on https://etherscan.io/.Figure:

FLOW.ASP.1

ASP.1: A third party token project starts an airdrop on the SORA platform

FIGURE: FLOW.ASP.1

[ASP.1.2]
TTPO notifies the SSC smart contract that he wishes to conduct an airdrop and sends it the address of the TTPSC

Third party Token Project's Owner **(TTPO)**

SORA Smart Contract **(SSC)**

[ASP.1.3]
SSC calls the balanceOf function of the TTPSC with its own address to see how many tokens TTPO has given SSC to airdrop

[ASP.1.5]
SSC checks its balance of TTPSC (in wei) is greater than the number of verified users, and if so it increments the global airdrop counter and saves a new airdrop record with the address of TTPSC on chain

[ASP.1.1]
TTPO transfers X wei of his project's tokens to the address of the SSC smart contract by calling the transfer function on the TTPSC

Third party Token Project's Smart Contract **(TTPSC)**

[ASP.1.4]
TTPSC sends the balance of SSC to SSC

Legend

TTPO = Third party Token Project's Owner

SSC = Sora Smart Contract

TTPSC = Third part Token Project's Smart Contract

User Verification and Allocation Mechanism

Summary: A process by which users can join for the Sora Terminal and receive SORA Tokens

Registering for the Sora Terminal is a simple, straightforward process by which users prove that they have not registered for a Sora Terminal account before and receive free Sora Tokens in exchange. This process is called the "User Verification and Allocation Mechanism" or "**UVAM**".

Joining the SORA network is a streamlined and secure process:

1. First, the new user opens the Sora Terminal web client (www.soraterminal.com) and begin by creating a new wallet. Users are welcomed to use 3rd party wallets that we support (including Metamask, Ledger, Trezor and MEW).

2. After creating a wallet, new users will choose to proceed as an individual or as an institution. For individuals, he/she will have to submit his/her basic information, including date of birth, full name, country of origin, email address and phone number. For institutions, he/she will have to provide their companies' basic info and their contact information. This is done to ensure that they are real users (there will also be an option to enter a referral code, see UVAM 2.1). Users will then have to agree to the terms and conditions, as well as privacy policy in exchange for tokens (See UVAM 1.1 and UVAM 2.1 in the figures on the next page).

3. Afterwards, users have to go through KYC verification. This will be done through Paxos. After passing KYC on Paxos, an icon will indicate that your account has been verified. Users will then be able to claim airdrops and gain access the OTC Marketplace. The **RNVP** (Sora Terminal operated *Real Name Verification Provider*) validates the information and produces a hash that is a pure function of the new user's information (every time someone authenticates with the same information, the same hash will be generated) and produces a corresponding signature with the RNVP's private key. (See UVAM.1.2 in FLOW.UVAM.1 on the next page)

4. The new user obtains Sora Token and may start to validate project communities, as well as access the Sora OTC Marketplace.
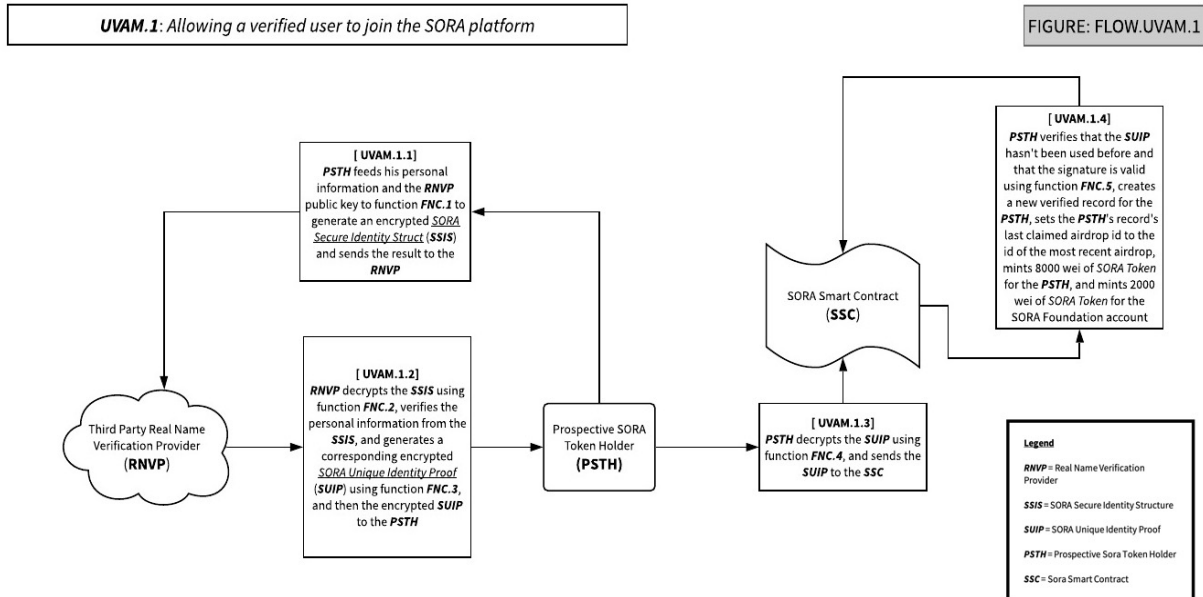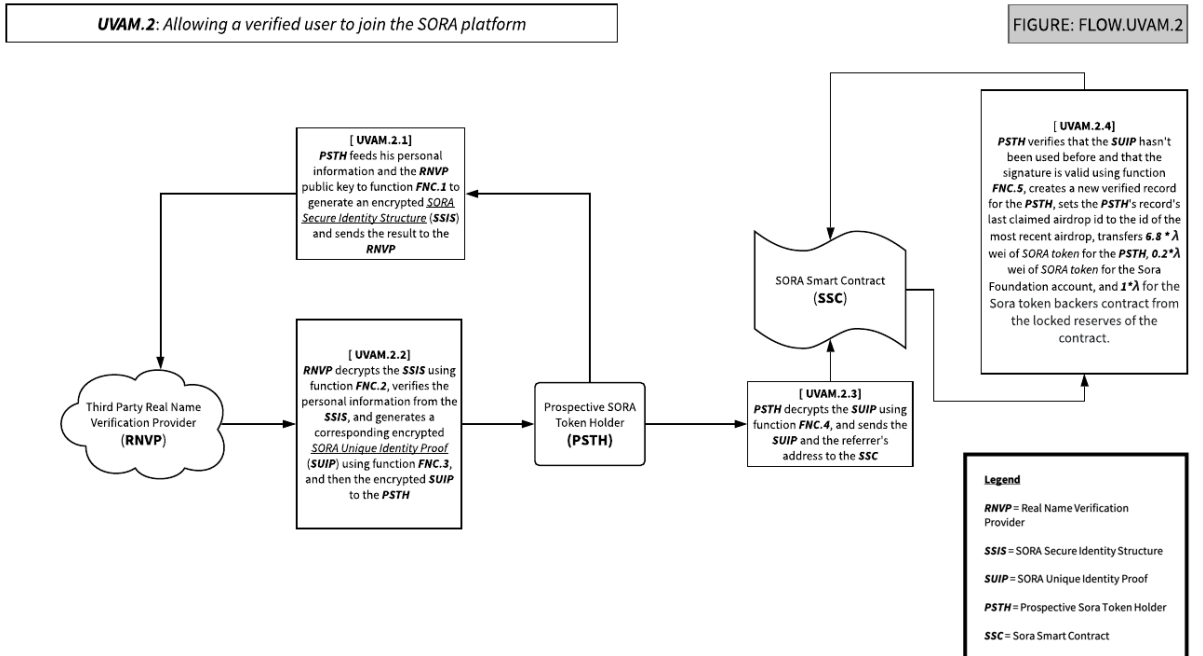
Figure: **FLOW.UVAM.1**

UVAM.1: Allowing a verified user to join the SORA platform

FIGURE: FLOW.UVAM.1

[UVAM.1.1]
PSTH feeds his personal information and the RNVP public key to function FNC.1 to generate an encrypted *SORA Secure Identity Struct* (SSIS) and sends the result to the RNVP

[UVAM.1.2]
RNVP decrypts the SSIS using function FNC.2, verifies the personal information from the SSIS, and generates a corresponding encrypted *SORA Unique Identity Proof* (SUIP) using function FNC.3, and then the encrypted SUIP to the PSTH

Third Party Real Name Verification Provider (RNVP)

Prospective SORA Token Holder (PSTH)

[UVAM.1.3]
PSTH decrypts the SUIP using function FNC.4, and sends the SUIP to the SSC

SORA Smart Contract (SSC)

[UVAM.1.4]
PSTH verifies that the SUIP hasn't been used before and that the signature is valid using function FNC.5, creates a new verified record for the PSTH, sets the PSTH's record's last claimed airdrop id to the id of the most recent airdrop, mints 8000 wei of SORA Token for the PSTH, and mints 2000 wei of SORA Token for the SORA Foundation account

Legend
RNVP = Real Name Verification Provider
SSIS = SORA Secure Identity Structure
SUIP = SORA Unique Identity Proof
PSTH = Prospective Sora Token Holder
SSC = Sora Smart Contract

Figure: **FLOW.UVAM.2 (Same process as UVAM.1, but with a referral code)**

UVAM.2: Allowing a verified user to join the SORA platform

FIGURE: FLOW.UVAM.2

[UVAM.2.1]
PSTH feeds his personal information and the RNVP public key to function FNC.1 to generate an encrypted *SORA Secure Identity Structure* (SSIS) and sends the result to the RNVP

[UVAM.2.2]
RNVP decrypts the SSIS using function FNC.2, verifies the personal information from the SSIS, and generates a corresponding encrypted *SORA Unique Identity Proof* (SUIP) using function FNC.3, and then the encrypted SUIP to the PSTH

Third Party Real Name Verification Provider (RNVP)

Prospective SORA Token Holder (PSTH)

[UVAM.2.3]
PSTH decrypts the SUIP using function FNC.4, and sends the SUIP and the referrer's address to the SSC

SORA Smart Contract (SSC)

[UVAM.2.4]
PSTH verifies that the SUIP hasn't been used before and that the signature is valid using function FNC.5, creates a new verified record for the PSTH, sets the PSTH's record's last claimed airdrop id to the id of the most recent airdrop, transfers $6.8 * \lambda$ wei of SORA token for the PSTH, $0.2*\lambda$ wei of SORA token for the Sora Foundation account, and $1*\lambda$ for the Sora token backers contract from the locked reserves of the contract.

Legend
RNVP = Real Name Verification Provider
SSIS = SORA Secure Identity Structure
SUIP = SORA Unique Identity Proof
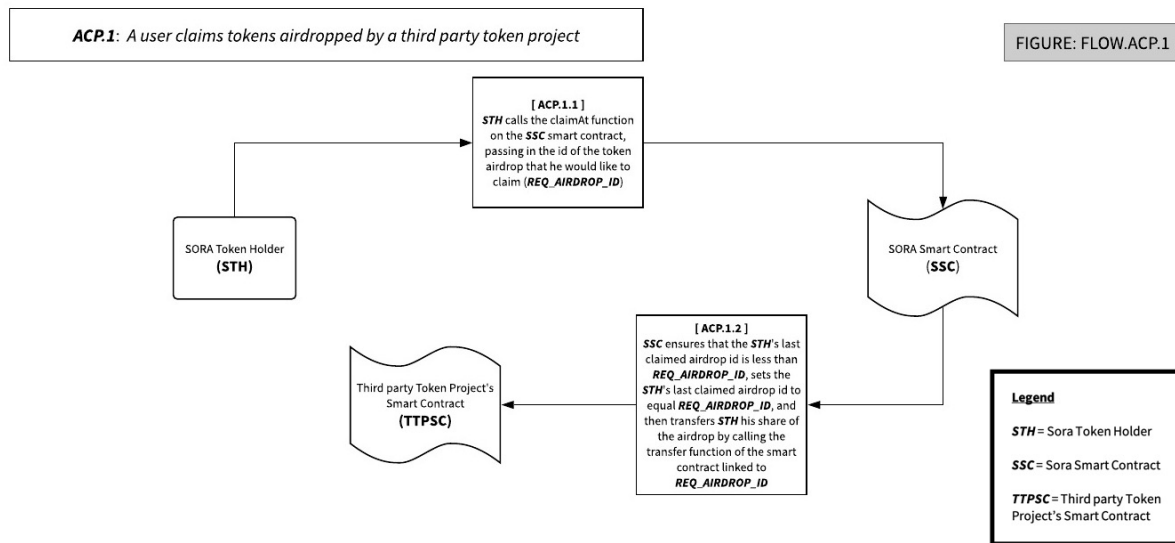PSTH = Prospective Sora Token Holder
SSC = Sora Smart Contract

Airdrop Claim Process

Summary: A process by which SORA Token holders can claim the airdrops conducted on the Sora Terminal.

Sora Terminal makes it easy for users to claim airdrops through a friendly web interface that enables users to select the airdrops they wish to claim and reject the airdrops they do not want to waste gas/cpu on. In addition, upon claiming an airdrop, users will receive a portion of the airdrop fee paid in ETH/EOS to help offset the gas fee/cpu used when claiming the airdrop.

The UX for claiming an airdrop on Sora Terminal is simple using the web client, but the process relies on a carefully designed process behind the scenes:
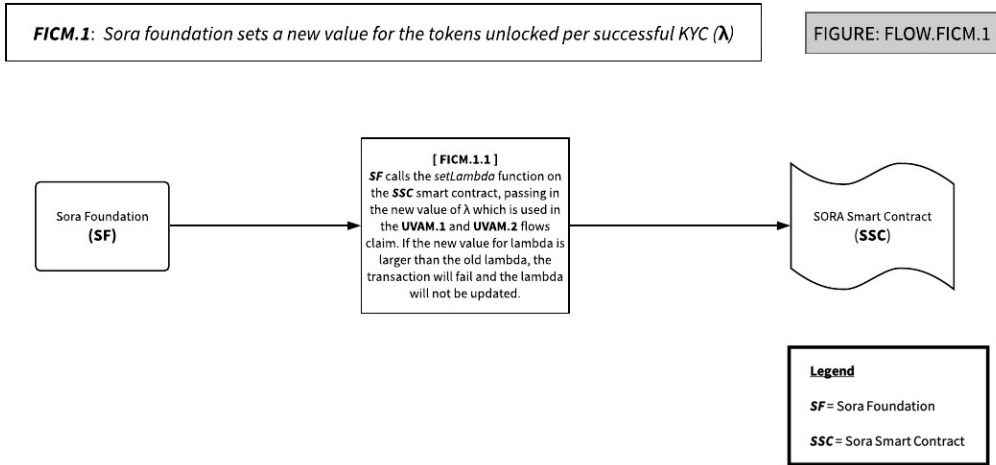
Figure: *FLOW.ACP.1*

ACP.1: *A user claims tokens airdropped by a third party token project*

FIGURE: FLOW.ACP.1

[ACP.1.1]
STH calls the claimAt function on the *SSC* smart contract, passing in the id of the token airdrop that he would like to claim (*REQ_AIRDROP_ID*)

SORA Token Holder
(STH)

SORA Smart Contract
(SSC)

[ACP.1.2]
SSC ensures that the *STH*'s last claimed airdrop id is less than *REQ_AIRDROP_ID*, sets the *STH*'s last claimed airdrop id to equal *REQ_AIRDROP_ID*, and then transfers *STH* his share of the airdrop by calling the transfer function of the smart contract linked to *REQ_AIRDROP_ID*

Third party Token Project's Smart Contract
(TTPSC)

Legend

STH = Sora Token Holder

SSC = Sora Smart Contract

TTPSC = Third party Token Project's Smart Contract

Foundation Control Mechanism

Summary: A process by which the Sora Foundation can slowly decrease the amount of Sora Token issued per KYC verification so as to maintain a fixed token supply cap from the start.

It is important for all token communities for holders to know that the number of airdrop tokens received by them, if claimed, will not be decreased significantly through the increase of number of Sora Token, so Sora token is structured as a fixed cap market token. To make this possible it is necessary for the Sora Foundation to be able to decrease the amount allocated for each new KYC verification through the smart contract.

The control mechanism allows the Sora Foundation to set the lambda (λ) parameter seen in the UVAM.1 and UVAM.2 processes via calls to the EOS/Ethereum smart contracts. While the Sora Foundation can decrease the contract's lambda value, it can never increase the number of tokens allocated per new KYC, meaning that the lambda parameter can only decrease overtime, and any changes to lambda will only serve to help existing token holders.

Figure: ***FLOW.FICM.1***

FICM.1: *Sora foundation sets a new value for the tokens unlocked per successful KYC (λ)*

FIGURE: FLOW.FICM.1

Sora Foundation (SF)

[FICM.1.1]
SF calls the *setLambda* function on the *SSC* smart contract, passing in the new value of λ which is used in the **UVAM.1** and **UVAM.2** flows claim. If the new value for lambda is larger than the old lambda, the transaction will fail and the lambda will not be updated.

SORA Smart Contract (SSC)

Legend
SF = Sora Foundation
SSC = Sora Smart Contract

References:

1. Thomas, Stefan. "The Death of the ICO (And 4 Other 2018 Predictions)." The Death of the ICO (And 4 Other 2018 Predictions) - CoinDesk, CoinDesk, 27 Dec. 2017, www.coindesk.com/death-ico-4-2018-predictions/.