A Proposal for Decentralizing Algorand Governance

By Silvio Micali

This post focuses on an architecture developed by Jing Chen and Silvio Micali, with the contributions of the entire Algorand Research Team.

STATUS QUO

Currently, in Algorand, every account can participate in *consensus*, but not governance. Governance is the power of deciding non-consensus tasks, the funding of grant proposals being an example.

As a next step toward responsible decentralization of the network, we want to introduce a mechanism for any account to not only participate in consensus, but also to govern.

GOAL

The goal of this proposal is decentralizing Algorand governance, and aligning network rewards with such governance. More precisely, it is putting forward the mechanics and incentives guaranteeing a governance that, just like our consensus protocol, is simultaneously decentralized, secure, and efficient.

QUICK SUMMARY

Participation in governance will be voluntary. The accounts that choose to participate, the *governing accounts* or, more simply, the *governors*, lock their tokens for a given amount of time, initially proposed to be one year. Governing accounts shall be rewarded for their work. We anticipate that the rewards earned by governing accounts will be higher than the current network rewards they will replace.[1]

We propose to implement this governance gradually, starting by decentralizing the funding of grant proposals presented to the Foundation.

[1] Network rewards are the additional tokens every Algorand account receives, based on its own stake and the given year, whether or not it participates in consensus.

1. Consensus vs. Governance

Before putting forward our guiding principles for governance participation, let us recall the principles for consensus participation and highlight the differences between these two forms of participation in Algorand.

Consensus

Consensus consists of the process of choosing the next block in the blockchain and verifying that all its transactions are valid. As in other blockchains, consensus is a *purely algorithmic process*. But, in Algorand, consensus enjoys some unique characteristics. Namely, in Algorand, consensus participation is

- *Easy*.
  Indeed, it does not require any significant computational resources and can in fact be performed, in the background, by an ordinary laptop.

  (We have worked hard to guarantee that consensus participation is open to all!)

- *Voluntary*.
  Algorand accounts can freely choose whether to participate in consensus or not.

- *Uncompensated*.
  Since consensus participation requires, in Algorand, only minimal computational effort, it does not require any separate compensation and is not compensated.

- *Unmonitored*.
  Since there is no remuneration attached to consensus participation, there is no need to check whether an account really participates in consensus and such participation is indeed unmonitored.

(Essentially, in Algorand, consensus participation is a “very light form of civic service”.)

The above guiding principles have been in force since the launch of our network. We have procedures in place to approve and deploy upgrades in our consensus protocol. We have already successfully deployed three main consensus upgrades (e.g., layer-1 smart contracts). But we have never changed the characteristic principles informing our consensus protocol. These principles continue to inform the present proposal.

Governance participation, however, is quite different and requires different principles.

Governance

Governance is not algorithmic. It requires *personal attention and time*. Accordingly, we propose that, in Algorand, governance participation is

- *Voluntary*.
  Algorand accounts can freely choose whether to participate in governance or not.

- *Compensated*.
  Governing accounts will be compensated, as discussed in section 3.

- *Monitored*.
  Governing accounts will be monitored to verify real participation in governance.

- *Locked*.
  Governors take decisions that deeply affect the growth of the Algorand ecosystem. To ensure that they will be affected by the votes they cast, governing accounts serve for one year and lock their tokens for the entire year.

  However, as we shall detail in section 5, governors are allowed early withdrawals from their locked accounts, but with precise restrictions and penalties.

  The rewards governors earn, however, are *not* locked! In fact they will be directly deposited to unlocked accounts chosen by the governors themselves.

Let us now explain how these principles inform governance's mechanics and incentives.

2. Governance Mechanics

Governance mechanics specify the voting process by which governing decisions are jointly made. This process involves only the Foundation and the governors, with very distinct roles: namely,

The Foundation *facilitates*, and governors *decide*.

This clarified, different governance mechanics can and should be considered and explored. But all of them should follow the following guiding principles.

- *Voting Sessions.*
  A voting session calls for the governors to vote on a set of items (e.g., a set of grant proposals submitted to the Foundation) providing for each item
  (i) its name/ID,
  (ii) a brief summary,
  (iii) a link to its details, and
  (iv) YES and NO selection options (which result in different postings on the blockchain).

- *Voting Deadlines.*
  People need time to think and we recommend that the Foundation gives 30 days to vote. The voting deadline of a session posted on day D is $D + 30$ and only votes posted by that deadline will be counted.

- *Transparency.*
  Session items can be proposed by any member of the community, but must be posted on the blockchain before they can be included in a session.

  The Foundation should take the responsibility of researching beforehand all items requiring a vote in a session; be the first one to vote; and in fact publish its own vote when publishing a session.

- *No censoring.*
  Suppose an item has been posted for 6 months but has not been included in any voting session. Then, governors can force a vote on a session containing that single item by posting on the chain transaction(s) showing that at least, say, 25% of the governing stake supports this move.

- *No spamming.*
  The posting of an item should require a fee greater than the milli algo of an ordinary transaction. In fact, an ordinary transaction is processed algorithmically, while a governance item must also be processed by humans. We must thus prevent that someone could cheaply waste the valuable time of the Foundation and the governors.

- *Voting options.*
  A voting session should have a “Vote with the Foundation” (VWF) option.
  Governing accounts should be free to either
  (1) ignore the VWF option and cast their votes individually; or
  (2) choose the VWF option, so as to match the Foundation’s vote.

- *Vote Validity, Vote Weight, and Outcomes.*
  Needless to say, each voting session should specify what makes a vote valid and how valid votes translate into outcomes. Only governors are allowed to vote, and their votes must have a weight proportional to the number of their locked tokens.

- *Monitoring.*
  Assume, for simplicity only, that the locking period of a governing account always starts on the first day of a month. Now assume that, at the start of any given month M during its locking period, the account has T tokens; and that the governance reward rate it is enjoying is r. Then, at the end of M, the account receives

  - $rT/12$ reward tokens, if it has voted on *all* sessions posted in its locking period and whose voting deadlines are within M, and

  - Half as many tokens, that is, $rT/24$ rewards tokens, otherwise.

  In the latter case, the unreceived $rT/24$ tokens will be transferred to a special account used by the Foundation for rewarding future governors.

  Thanks to the transparency of the blockchain, monitoring governance participation is both easy and unambiguous.

- *Convenience.*
  Governance voting should be made available to mobile wallets.

- *Separate governance keys.*
  Governance votes must be digitally signed by governance-participation keys that are separate from both spending and consensus-participation keys. This enables governing keys to be ephemeral, and expire after a voting session. It also enables a user to keep her locked governing account with a custody provider, while retaining her governance voting power on her wireless wallet.

In sum, the aim of the above guiding principles is simple enough:

The Foundation should facilitate governance, but not control it.

Voting-Type Examples

Different types of voting can and should be considered. Here are just two of them.

- *K-out-of-N voting*
  For concreteness, let us consider an 11-out-of-20 voting session. Such a session calls for a vote on 20 items: for example, 20 grant proposals, each with its own information and YES and NO buttons. A valid vote for such a session consists of selecting (i.e., clicking YES on) exactly 11 of them.

  In such a session,

  (1) the Foundation may actually rank all 20 proposals in its own order of preference, so as to provide more information to the governors, and

  (2) the Foundation's vote is implicit in the provided ordered list: namely, it consists of selecting the first 11 proposals in the list.

  Thus, in a K-out-of-N session, the information provided by the Foundation is richer than just its own vote. A governor, which is sure about selecting only ---say--- 7 proposals, may therefore be helped by the ranked list provided by the Foundation to select 4 additional proposals and cast a valid vote.

  In the above 11-out-of-20 session, rather than exclusively choosing between (a) individually selecting 11 proposals and (b) voting with the Foundation, our envisaged governor may actually choose a third option: (c) selecting the VWT option *and* the 7 proposals it is sure about. By choosing the latter option, our governor automatically selects 11 specific proposals: the 7 it has explicitly selected and the first 4 proposals in the Foundation's ranked list that do not appear among the 7 it has individually selected.

  The outcome of such a session may consist of ---say--- funding the 5 proposals with the highest selection weight. (If a governor has T locked tokens, then it contributes a weight of T to each proposal it selects.)

  One good property of K-out-of-N voting is that a rich governing account cannot concentrate its voting weight on a single proposal ---e.g., one that the account itself submitted for funding! If it wants to cast a valid vote for its own proposal it must also vote for another 10 proposals. Accordingly, it might want to invest the time to select 10 more proposals that in its mind are the best ones for the Algorand ecosystem.

- *Greedy Budget Voting*
  K-out-of-N voting works well when each proposal requires roughly the same amount of funding. When this is not the case, we need a voting system that takes budget into consideration. Here is one example of such voting.[2]

  The Foundation specifies a budget and ranks the items. Its vote consists of the highest ranked items that can all be funded without exceeding the budget.

  A governor may either

  (a) choose the VWF option or
  (b) select a set of items that can all be funded without violating the budget.

  In the latter case, the governor's vote is computed automatically by orderly examining the Foundation's list and selecting a non-yet selected item as long as all items selected so far can be funded without violating the budget.

3. Governance Compensation

Having the Foundation dictate the compensation payable to accounts participating in governance would be *centralization, pure and simple*. And it might also be totally counterproductive. By choosing the wrong compensation level, the Foundation may actually *discourage* governance participation.

In keeping with Algorand's philosophy, we propose to have the *governance reward rate* of governing accounts be chosen in a decentralized fashion. More precisely, we propose to have it *decided by the accounts themselves* via a *Dutch auction*, within the parameters decided by the Foundation.

For simplicity, initially assume that Dutch auctions for the governance reward rate are executed once a year, on January 1st. The accounts that win the year's auction are the year's governing accounts, and remain locked throughout the year. As soon as their locking period is over, they are free to participate in the auction of the next year.

At the end of the section, we explain how to handle auctions that start at arbitrary and multiple times in the year. For a more precise description of Dutch auctions, and their implementation on the blockchain, see ChenMicali. Here, we wish to provide only the flavor of what is involved in a Dutch action for the governance reward rate. The following description is descriptive but hopefully conveys how the governance reward rate is "decided by the accounts themselves."

[2] The example is based on the Knapsack problem, and various existing algorithms can be used here.

For each year's auction, the Foundation specifies:

1. the *reward pool* ---i.e., the total amount of governance reward tokens, R, from its own treasury, that it is ready to distribute to the governing tokens during the year.
2. the number of stages of the auction, k, and
3. a matching number of increasing rates between 0 and 1: $r_1 < r_2 < \cdots < r_k$, where $r_1 = R/T$ and T is the number of tokens in circulation at the year's start.

For instance, $R = 8 \cdot 10^8,\ T = 10^{10}, k = 100,\ r_1 = 8\%,\ r_2 = 9\%, \ldots\ ,\ r_{100} = 107\%$.

The rate of the first stage, r_1, is the minimum governance reward rate the auction may produce, and the rate of the last stage, r_k , is the maximum one.

The actual (non-compounded) governance reward rate is decided by the bids the accounts place in the various stages of the auction. Let us thus sketch (a) what bids the accounts may place in a stage and (b) how the placed bids determine the outcome of the auction.

(a) In a stage i, an account may bid a number a of tokens. This bid *"commits to locking a (additional) tokens, if the governance reward rate is greater than or equal to r_i."*

Such a tokens are committed because they (1) cannot be spent or transacted during the auction and (2) are automatically locked for one year if the bid is "winning."

Of course, the bid is valid only if the account still has a tokens available to commit!

(b) The auction terminates at the first stage i in which, letting C_i be the number of tokens committed so far, $r_i \cdot C_i \geq R$.

If no such a stage exists, then the auction terminates at stage k.

Let the auction terminate at stage t. Then,

- the governance reward rate is r_t.
- all the tokens committed so far are locked for the entire year (in a separate account);[3] and
- the accounts owning these locked tokens are the governing accounts.

The termination condition, $r_i \cdot C_i \geq R$, is quite natural. In fact, (1) the Foundation is not willing to use more than R tokens to reward the governing accounts, (2) when the auction terminates at stage t, the reward rate is r_t and (3) by rewarding all C_t locked tokens of the governing accounts at the governance reward rate of r_t obliges the Foundation to spend $r_t \cdot C_t$ tokens by year end.

[3] More precisely, all the tokens committed so far are locked only if the inequality of the termination condition actually is an equality. Else, all the tokens committed in the first t-1 stages are locked, and one must use some tie-breaking rule (e.g., first come first serve) to determine which of the tokens committed in stage t are actually locked.

Example

As before, let $k = 100,\ r_1 = 8\%,\ r_2 = 9\%, \dots\ ,\ r_{100} = 107\%$; let the auction terminate at stage 13; and assume that an account x with 900 tokens placed only three bids: the first committing 100 tokens in stage 1; the second committing 200 tokens in stage 5; and the third committing another 200 tokens in stage 10. Note that all these three bids are valid.

Then, the auction sets the governance reward rate to be $r_{13} = 20\%,$ and account x becomes a governing account with 500 locked tokens. (The other 400 tokens of x continue to remain unlocked, in a separate account controlled by x.)

The governance participation of account x will be monitored. If x indeed participates, then it will, throughout the year, receive $500 \cdot 20\% = 100$ tokens from the reward pool by the year's end.

Account x should be happy about the auction outcome: by its first bid, it declared to be happy to lock 100 of its tokens should the governance reward rate be 8% or more. The governance reward rate turned out to be more 2.5 times higher! By the same reason, account x is also happy about its second and third bids.

From Yearly to Pipelined Dutch Auctions

For simplicity, we have been assuming running a single governance-reward-rate auction per year.

It is important, however, to be able to run such auctions more frequently (e.g., once a month) while keeping the locking period of governing accounts to be one year.

The reason for keeping one-year locking periods is to continue to ensure that governing accounts have substantial "skin in the game." And the reason to run such special auctions more frequently is that new tokens enter the Algorand ecosystem all the time. We must thus allow these tokens to participate in governance as soon as possible, without having to wait for the next year.

We must also be cognizant, however, that running several Dutch auctions a year, may complicate the strategic thinking of the bidding accounts.

For instance, an account that has locked its tokens by winning the Dutch auction of, say, February 1st, may regret having done so once it realizes that the Dutch auction of, say, April 1st yielded a much higher governance reward rate. Faced with these strategic complexities, the accounts may react in unforeseeable and counterproductive ways.

To avoid such regrets, Jing Chen and I have architected a way to enable an account that has already locked its tokens in a prior auction to participate to any later auction it wants and, if it wins, to *"seamlessly switch"* some or all of its already locked tokens to the new locking period and new governance reward rate.

We call these so architected auctions *pipelined Dutch auctions*. For a more precise description of such auctions see ChenMicali .

Pipelined Dutch auctions enable the Foundation to call a governance-reward-rate auction essentially at any date.

For instance, at the start of every month, the Foundation may

(1) Compute the number of tokens, T, that have entered into circulation in the past month
(2) Create a reward pool consisting of $R = T \cdot r$ tokens, where r is the reward rate chosen by the Foundation for the entire year; and
(3) Run a governance-reward-rate auction with the created reward pool (and with whatever number of stages and stage rates it wants.)

4. Early withdrawals from Governing Accounts

To encourage locking for participating in governance, we permit early withdrawals from governing accounts, but at the following conditions.

Governors may withdraw cumulatively at most 90% of their locked tokens within a 30-day period. Moreover, for each withdrawal, a governor g

(1) pays a penalty equal to, say, 10% of the withdrawn tokens, and
(2) loses all its rewards for the month in which the withdrawal occurs.

Half of g's penalty will enrich the reward pool of the next scheduled auction, and the other half will be distributed at the end of the locking period to the remaining governors, according to their stake at that time. The same goes for rewards lost by g.

These penalties and losses discourage g from withdrawing from its locked account. Moreover, they encourage over governors, who receive a significant portion of g's penalties and losses, to continue participating in governance: the more withdrawals the more additional rewards for the governors that continue keeping their tokens locked. Finally, the Foundation can use its portion of g's losses and penalties to call for new governance auctions, so as to recruit unlocked tokens to participating in governance.

5. Network Rewards vs. Governance Compensation

Today, all Algorand accounts, whether or not participating in consensus, receive the network reward per token. In 2020, the total number of network reward tokens amounted to a (non-compounded average) rate of 6%. According to the original plan, such an annual network reward rate was going to 0 within the next 3.5 years.

The Foundation, however, is planning to change the network reward plan to cover seven years with a corresponding to an (approximate) annual rate of 8% in 2021, down to 1% in 2028. That is, for 2021, the Foundation plans to distribute R_{2021}network reward tokens, where $R_{2021} = 8\% \cdot T_{2021}$and $T_{2021}$is the number of tokens in circulation at the start of 2021. And similarly for the years 2022 through 2028.

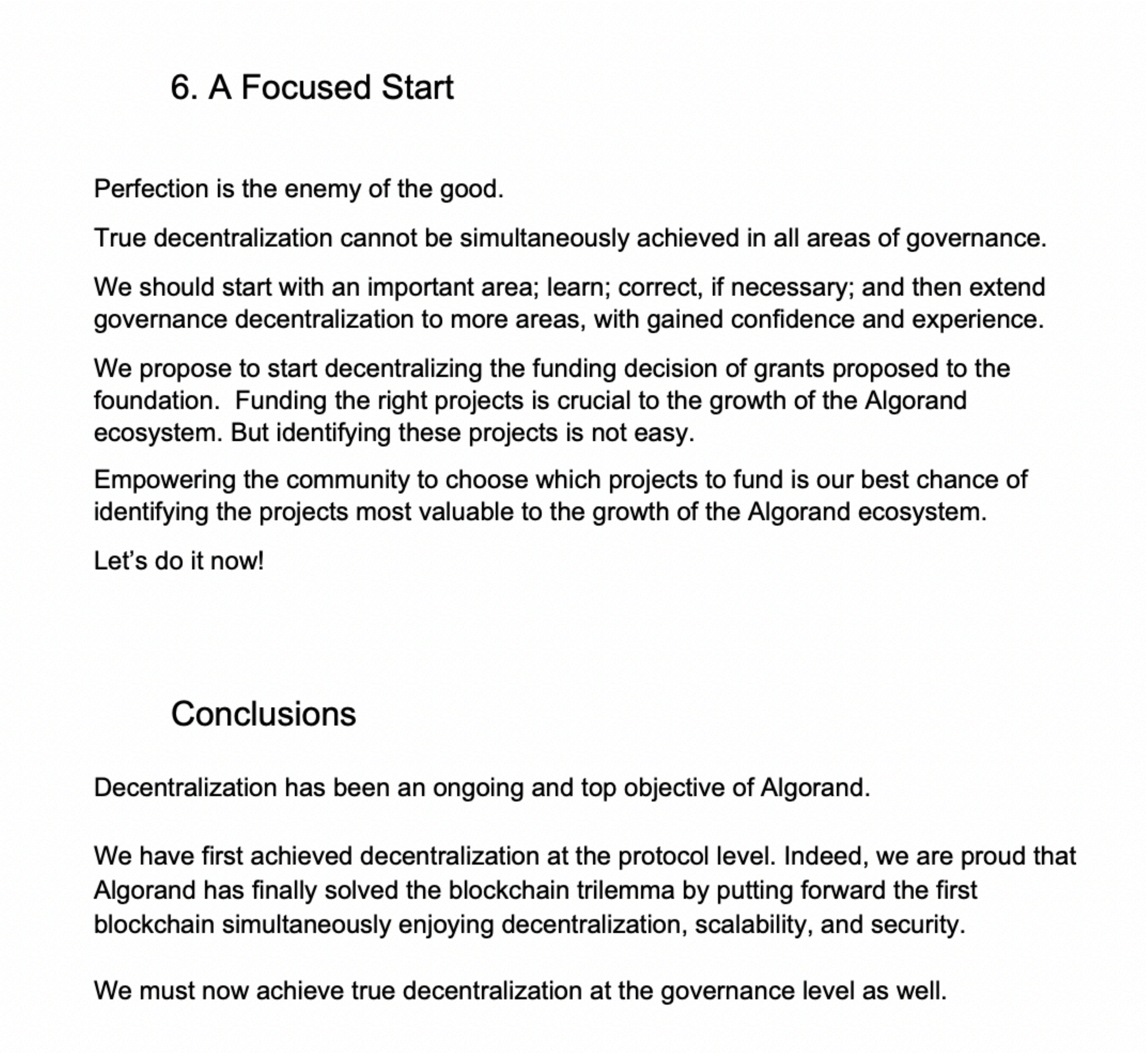
What would happen if the Foundation used the same number of reward tokens, but for governance rather than network rewards? It would set the governance reward rate for 2021 via a Dutch auction run on January 1st with a reward pool of R_{2021}tokens.

In such an auction, the reward rate of stage 1 is, by construction, $r_1 = 8\%$.
Thus, *each account may guarantee itself a governance reward rate of 8%* on all or some of its tokens, so long as it is willing to lock those tokens and participate in governance.

Indeed, all the account has to do is to commit all those tokens in stage 1. By doing so, the account guarantees that its bid is a winning bid, even if all other accounts commit all their tokens in stage 1. More generally, no matter how the 2021 auction is played by all other accounts, the resulting governance reward rate will be at least 8%. Indeed, 8% is the reward rate of stage 1, and reward rates increase with the stages.

Of course, the final governance reward rate enjoyed by our account may be strictly higher than 8%. Such a higher rate occurs whenever not all accounts commit all of their tokens in stage 1 of the 2021 auction.

For instance, assume that, collectively, the accounts do not wish to lock $T/2$ of their tokens, that is, half of the tokens in circulation. Then, these tokens will never be committed in the 2021 auction. Accordingly, no matter how the auction is played, the resulting governance reward rate is at least 16%. In fact, the auction cannot end at any stage s whose reward rate is $r_s < 16\%$. Indeed, even if all the other $T/2$ tokens were committed by such a stage s, since $R =$ 8% T, we must have $\frac{T}{2} \cdot r_s < R$.



6. A Focused Start

Perfection is the enemy of the good.

True decentralization cannot be simultaneously achieved in all areas of governance.

We should start with an important area; learn; correct, if necessary; and then extend governance decentralization to more areas, with gained confidence and experience.

We propose to start decentralizing the funding decision of grants proposed to the foundation. Funding the right projects is crucial to the growth of the Algorand ecosystem. But identifying these projects is not easy.

Empowering the community to choose which projects to fund is our best chance of identifying the projects most valuable to the growth of the Algorand ecosystem.

Let’s do it now!

Conclusions

Decentralization has been an ongoing and top objective of Algorand.

We have first achieved decentralization at the protocol level. Indeed, we are proud that Algorand has finally solved the blockchain trilemma by putting forward the first blockchain simultaneously enjoying decentralization, scalability, and security.

We must now achieve true decentralization at the governance level as well.

JOIN THE CONVERSATION!

Discuss this proposal for decentralized governance and more on the Algorand Governance Forum and Algorand’s Governance Discord Channel.

JING CHEN | Head of Theory Research and Chief Scientist

Jing is an Assistant Professor in the Computer Science Department at Stony Brook University. She is also an Affiliated Assistant Professor in the Economics Department and an Affiliated Member of the Stony Brook Center for Game Theory. Her main research interests are distributed ledgers, game theory, and algorithms. Jing received her Bachelor and Master degrees in Computer Science from Tsinghua University, and her PhD in Computer Science from MIT. She did a one-year postdoc at the Institute for Advanced Study, Princeton. Jing received the NSF CAREER Award in 2016.

SILVIO MICALI | Founder, Algorand

Silvio Micali has been on the faculty at MIT, Electrical Engineering and Computer Science Department, since 1983. Silvio's research interests are cryptography, zero knowledge, pseudorandom generation, secure protocols, and mechanism design and blockchain. In particular, Silvio is the co-inventor of probabilistic encryption, Zero-Knowledge Proofs, Verifiable Random Functions and many of the protocols that are the foundations of modern cryptography.

In 2017, Silvio founded Algorand, a fully decentralized, secure, and scalable blockchain which provides a common platform for building products and services for a borderless economy. At Algorand, Silvio oversees all research, including theory, security and crypto finance.

Silvio is the recipient of the Turing Award (in computer science), of the Gödel Prize (in theoretical computer science) and the RSA prize (in cryptography). He is a member of the National Academy of Sciences, the National Academy of Engineering, the American Academy of Arts and Sciences and Accademia dei Lincei.

Silvio has received his Laurea in Mathematics from the University of Rome, and his PhD in Computer Science from the University of California at Berkeley.