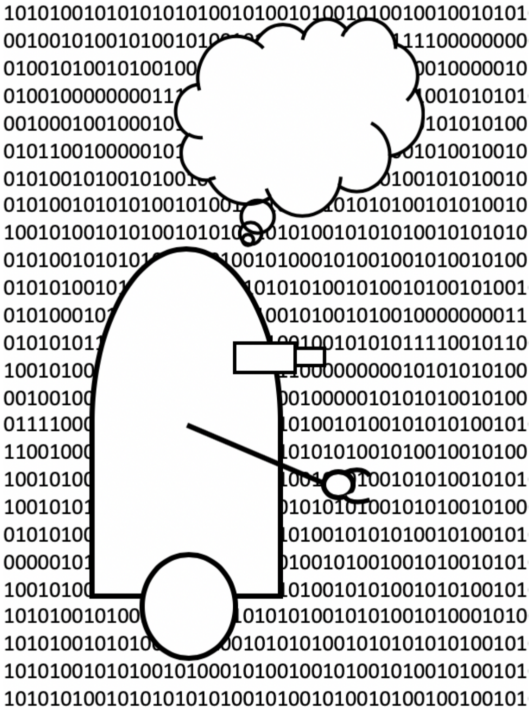
# Algorand Autonomous

Brian Haney[1] Archie Chaudhury[2]



Cryptots[3]

2021 MIT Bitcoin Expo Hackathon
The New Normal

April 3, 2021

[1] bhaney3 [at] nd [dot] edu.
[2] archchaudhury02 [at] gmail [dot] com.
[3] Authors contributed equally to this work, submitting as a team, Cryptots.

Project GitHub: https://github.com/Bhaney44/Cryptots

## Abstract

This project introduces Algorand Autonomous, a smart contract platform, voting mechanism, and a full-stack DAO deployment mechanism. At the technical convergence of blockchain and artificial intelligence, Autonomous Algorand is a platform for technical and financial innovation. The purpose of Algorand Autonomous is to provide investors with a secure financial hub that acts as a DAO and also allows them to upload transactions and financial data to a remote Skynet Database.

## Table of Contents

*Introduction*.......... *4*

*I. Smart Contract*.......... *6*

*II. Voting*.......... *8*

*III. Outcome*.......... *12*

*Conclusion*.......... *17*

*Appendix A. Statistical Results*.......... *18*

# Introduction

Algorand Autonomous creates a more accessible blockchain framework, which updates autonomously according to a consensus validation structure for smart contracts. We used two main technologies Algorand and Skynet to build Algorand Autonomous. Algorand is an open source permissionless blockchain, available under the MIT License. Then, the MIT License grants a license to use the technology, while limiting liability for the copyright holder.[4] Algorand's vision is a global, borderless economy. Algorand is a truly democratic and efficient way to implement a public ledger.[5]

Algorand is a proof-of-stake blockchain, which evolved to improve security and power efficiency across the network by limiting miners to validating transactions proportional to an ownership share.[6] A majority override is a hack which results from competitive advantage in mining. To combat the majority override problem, Algorand developed a proof-of-stake chain, differing from classical blockchains, which use a proof-of-work to validate transactions.

Skynet is a decentralized hosting platform that allows users to upload files and web-apps. Developers can take advantage of Skynet's SKYDB system to create dynamic websites which allow users to change mutable data and have webpages update accordingly. Skynet's goal is to implement a decentralized internet, and its hosting capabilities enable it to do just that.

Skynet's protocol is built on top of SIA, which is a blockchain that was developed primarily for the purpose of storage. This allows users to simply store data on Skynet's servers. This decentralized mechanism also is advantageous in terms of authentication. Users of "SKAPPS" can use one id to log into any webapp that is based in Skynet. We used Skynet and Algorand to address the Algorand Challenge and the Sia Challenge, in doing so, we solved three problems.

**Problem 1** Across blockchain networks, governance is a complex task with multiple stakeholders having several financial interests. The process by which decisions are made among these stakeholders is often centralized with certain authorities. For example, investment funds typically are controlled by a central authority. However, this decision-making framework limits openness for global community development and collaboration, as well as the personal autonomy of network participants.

**Solution 1** The solution is a self-updating Decentralized Autonomous Organization (DAO) with three parts. First, a smart contract platform is made available to DAO users. We explored the Algorand PyTeal Library for contract swaps and developed a novel smart contract software with Tkinter, a Python GUI. Second, a consensus voting mechanism is established to govern the DAO by democratic process. We started with PyTeal voting software and invented a new autonomous consensus protocol. Third, the smart contract platform and voting procedure are deployed through an online interface and DAO platform. We used Skynet to make a website for our project.

[4] The MIT License, Open Source Initiative (2021).
[5] Jing Chen, Silvio Micali, Algorand, 1 (May 26, 2017).
[6] Yossi Gilad, et al., Algorand: Scaling Byzantine Agreements for Cryptocurrencies, 53 (2017).

**Problem 2** One challenge we faced was finding a small problem to solve in the financial industry, to provide a use case. So, we started by developing some smart contract code for Algorand Swaps using PyTeal. In doing so we identified a problem because blockchain swap markets are very volatile. So, we established another problem statement, predict the close price of Algorand tomorrow.

**Solution 2** To start, we aggregated a dataset with financial information for top cryptocurrency chains and hardware technology companies. We then ran statistical analysis on public Algorand data using the Python Pandas Library. Next, we decided to expand the dataset. The dataset now includes public financial data for Apple, Intel, NVIDIA, Bitcoin, Ethereum, and Algorand. We then began construction on a neural network using the TensorFlow Library to make daily price predictions.

**Problem 3** The final challenge we faced was to integrate a suitable front-end where users could effectively store their financial data and come back to it later. This was needed because of all of the financial exchanges in our eventual application will be dependent on Algorand's framework. Thus, there needed to be an effective way for users to record their transaction data so that they could use it as a proof of work when they needed it.

**Solution 3** Our solution to this was to use Skynet's SKYDB system. Through integration of the authentication login system, we were able to develop a front end where users could create a secret key and then write a ledger representing all of their transactions. This data will then be saved on Skynet's network so that users can access it later.

# I. Smart Contract

## A. Software

The main programming language used for software development during this project was Python.[7] Moreover, PyTeal is a Python compiler for Algorand's Transaction Execution Approval Language (TEAL).[8] PyTeal has two types of contracts, stateless and stateful. Stateless contracts allow the network to validate the contract as it is delivered to the ledger. Stateful contracts live on the blockchain with global and local storage. We created a GitHub for the Autonomous Algorand Project, licensing the software under the MIT License.[9]

## B. Interface

A smart contract is a computer program which automatically executes, transferring cryptocurrency. In other words, smart contracts are programs that are logically executed on a blockchain without a central oversight. Figure 1 shows and empty user interface (UI) for an Algorand smart contract.

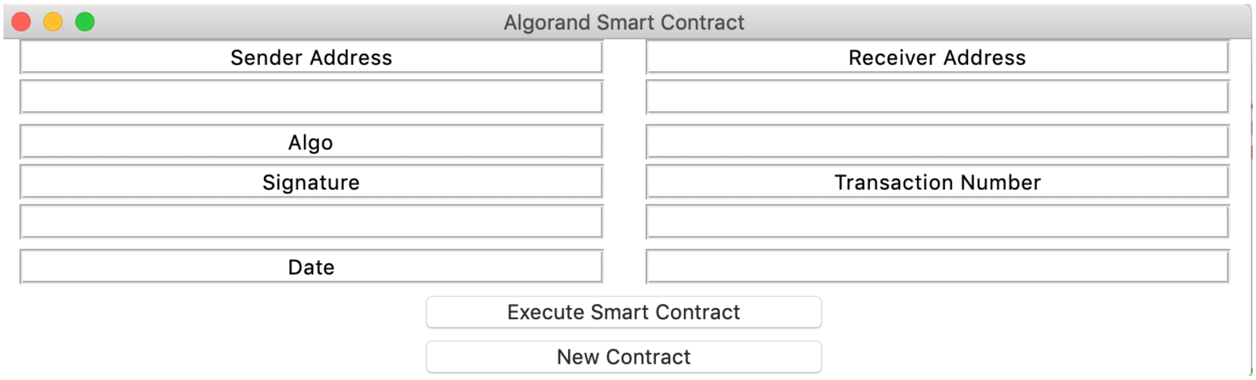
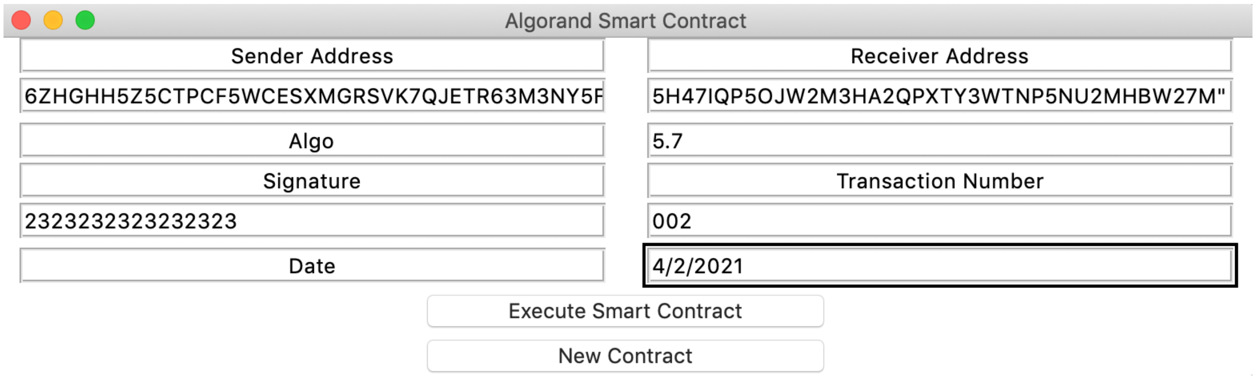


Figure 1

While digital signatures are notoriously expensive on computing resources, mechanisms may be adopted from the Algorand software stack for signatures. For example, Pixel signatures reduce the necessary bandwidth for standard digital signatures in proof-of-stake chain.[10] Figure 2 shows a complete smart contract.

[7] Guido van Rossum, An Introduction to Python (2001).
[8] PyTeal Documentation (2021), https://pyteal.readthedocs.io/en/stable/overview.html.
[9] Cryptots, GitHub (2021) https://github.com/Bhaney44/Cryptots.
[10] Manu Drijvers, et al., Pixel: Multi-signatures for consensus (2019).



|  |  |
|---|---|
| Sender Address | Receiver Address |
| 6ZHGHH5Z5CTPCF5WCESXMGRSVK7QJETR63M3NY5F | 5H47IQP5OJW2M3HA2QPXTY3WTNP5NU2MHBW27M" |
| Algo | 5.7 |
| Signature | Transaction Number |
| 2323232323232323 | 002 |
| Date | 4/2/2021 |

Figure 2

One problem with smart contracts, is that only really skilled developers can execute contracts on blockchain networks, which can delay payment processing. This contract software solves the payment processing problem by making smart contracts simpler and more accessible to new developers and general users.

### C. Ledger

A public ledger is a tamperproof data sequence that can be read and changed by everyone with access to the ledger. In DAOs, public ledgers have innumerable and compelling uses because they provide a mechanism by which the decentralized organization may exchange and store data.[11]

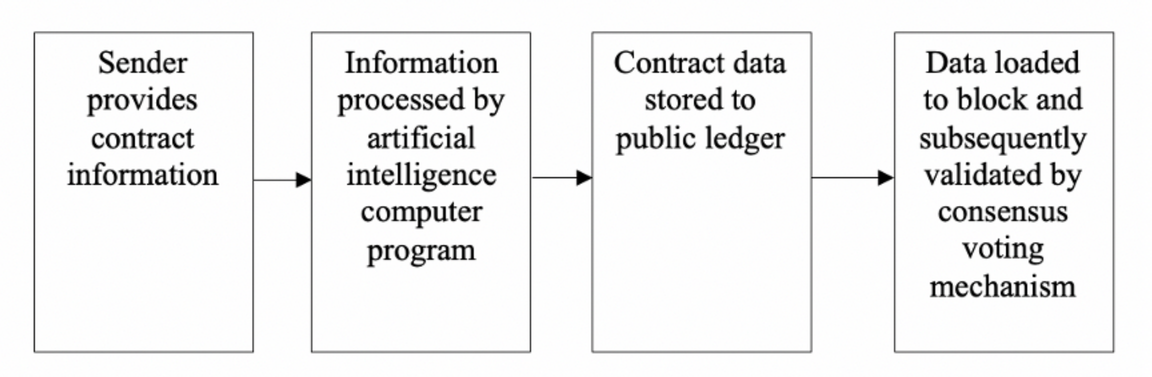

Sender provides contract information
Information processed by artificial intelligence computer program
Contract data stored to public ledger
Data loaded to block and subsequently validated by consensus voting mechanism

Figure 3

Figure 3 is a process by which contracts are communicated to the ledger, and subsequently validated by the consensus voting mechanism. Indeed, Algorand is a democratic way to implement a public ledger because it uses a staking rewards mechanism.

[11] Jing Chen, Silvio Micali, Algorand, 1 (May 26, 2017).

# II. Voting

## A. Staking

The staking mechanism may evolve using Algorand Standard Asset (ASA) voting. ASA's offer a standardized, Layer-1 mechanism to represent any asset on the Algorand blockchain.

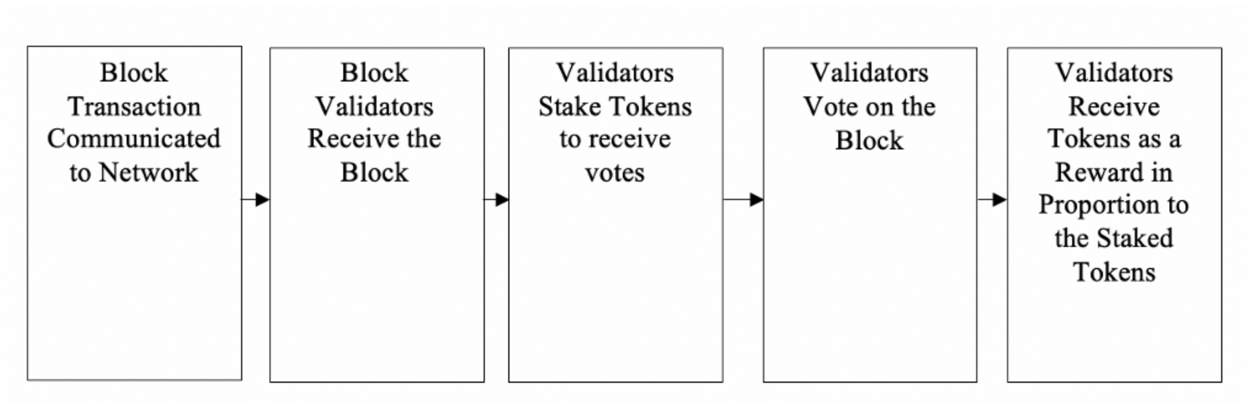

Block Transaction Communicated to Network
Block Validators Receive the Block
Validators Stake Tokens to receive votes
Validators Vote on the Block
Validators Receive Tokens as a Reward in Proportion to the Staked Tokens

Figure 4

Figure 4 is a schematic for block validation where validators on the block receive tokens in proportion to their staked rewards. For example, for every percent of the total stake, a node may get 1000 votes. This will allow widespread participation by a global community. Simultaneously, this will prevent majority override attacks.

### B. Consensus

Algorand provides a democratic consensus mechanism for voting,[12] using a new Byzantine Agreement (BA) protocol to reach agreement among users on the next set of transactions.[13] Voting can be accomplished on the Algorand blockchain using the Python PyTeal library.

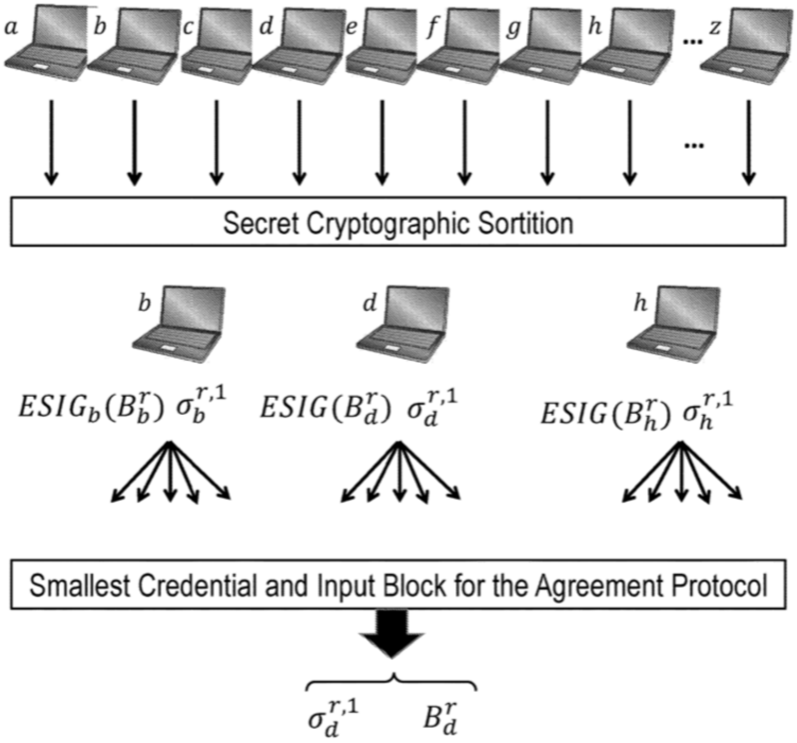

a
b
c
d
e
f
g
h
...
z
...
Secret Cryptographic Sortition
b
d
h
$ESIG_b(B_b^r)\ \sigma_b^{r,1}$
$ESIG(B_d^r)\ \sigma_d^{r,1}$
$ESIG(B_h^r)\ \sigma_h^{r,1}$
Smallest Credential and Input Block for the Agreement Protocol
$\sigma_d^{r,1}$
$B_d^r$

Figure 5[14]

Figure 5 is a drawing from a public patent application for Algorand's process for reaching block validation. The consensus voting mechanism will allow the code to evolve to the will of the network. Indeed, no block can be validated without a majority consensus decision from stakers in the block.

[12] Jing Chen, Silvio Micali, Algorand, 1 (May 26, 2017).

[13] Yossi Gilad, et al., Algorand: Scaling Byzantine Agreements for Cryptocurrencies (2017).

[14] U.S. Patent Application Publication 2019/0147438 to Micali, Distributed transaction propagation and verification system (May 16, 2019).

## C. Rewards

Validators receive rewards in the form of Algo. Figure 6 is a model for an applied validation system for an investing DAO.

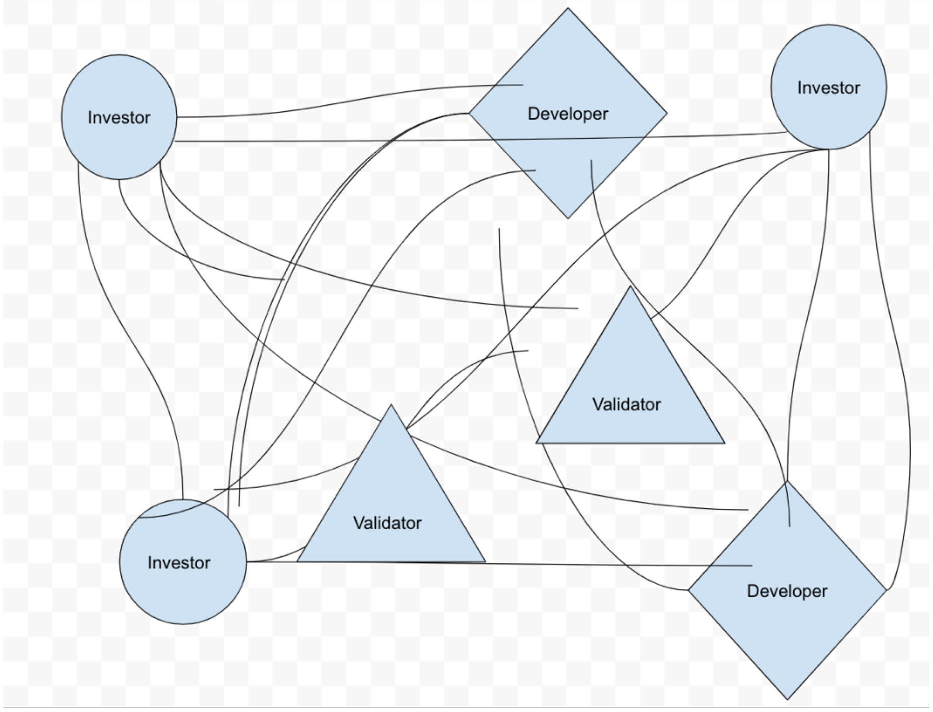


Figure 6

Blocks may be validated at consistent time intervals. During the time intervals, block validators may stake coins, where the total staked coins are aggregated for each block along with a Boolean vote.

$$s \in S$$

$$p = {}^{s_n}/_{S}$$

$$v \in V$$

$$v_n = 0;\ v_i = 1$$

Upon the time interval collapsing, the total coins staked will be summed, along with the total positive votes.

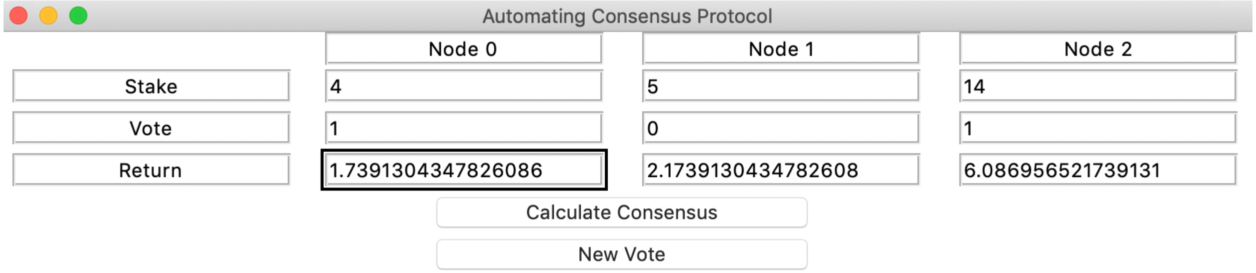
$$T = \sum_{V_i} V$$

$$C = {}^{T}/_{V}$$

The total positive votes will be divided by the total votes, if the product *C* surpasses a 50.0% majority, the block is validated, and rewards are returned to the validators.

$$r \in R$$

$$r_n = R(p)$$



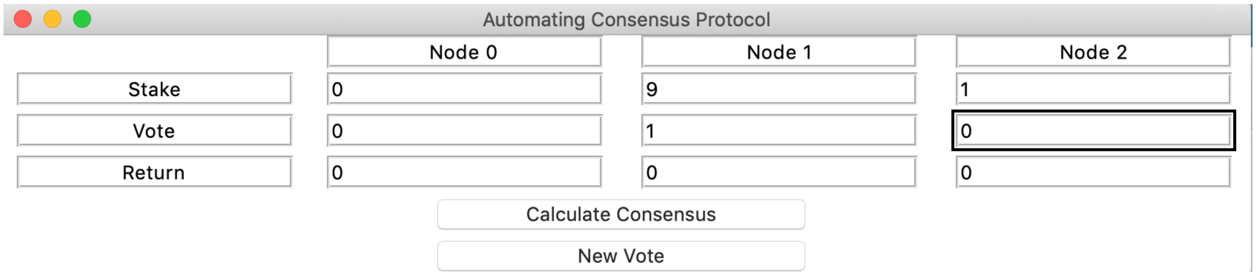
| | Node 0 | Node 1 | Node 2 |
|---|---|---|---|
| Stake | 4 | 5 | 14 |
| Vote | 1 | 0 | 1 |
| Return | 1.7391304347826086 | 2.1739130434782608 | 6.086956521739131 |

Figure 7

If the product does not reach a consensus, the block returns nothing and the chain moves to the next block along with the staked tokens.

$$r_n = 0$$

$$r_i \in R_i$$



| | Node 0 | Node 1 | Node 2 |
|---|---|---|---|
| Stake | 0 | 9 | 1 |
| Vote | 0 | 1 | 0 |
| Return | 0 | 0 | 0 |

Figure 8

Thus, validators are incentivized to validate the block to receive the reward. And the more failed updates, the more incentive to validate the next block for the validators.

# III. Outcome

## A. Deployment

We deployed our application by creating a web-app for Skynet. We did this by using React, along with HTML and CSS. We also used Skynet's SkyDB database to allow users to make their public keys and store transaction data.

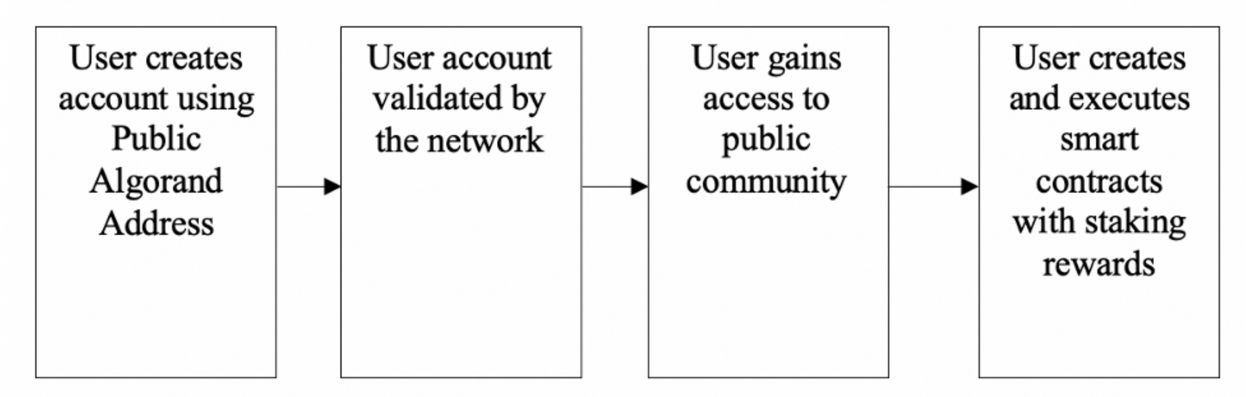
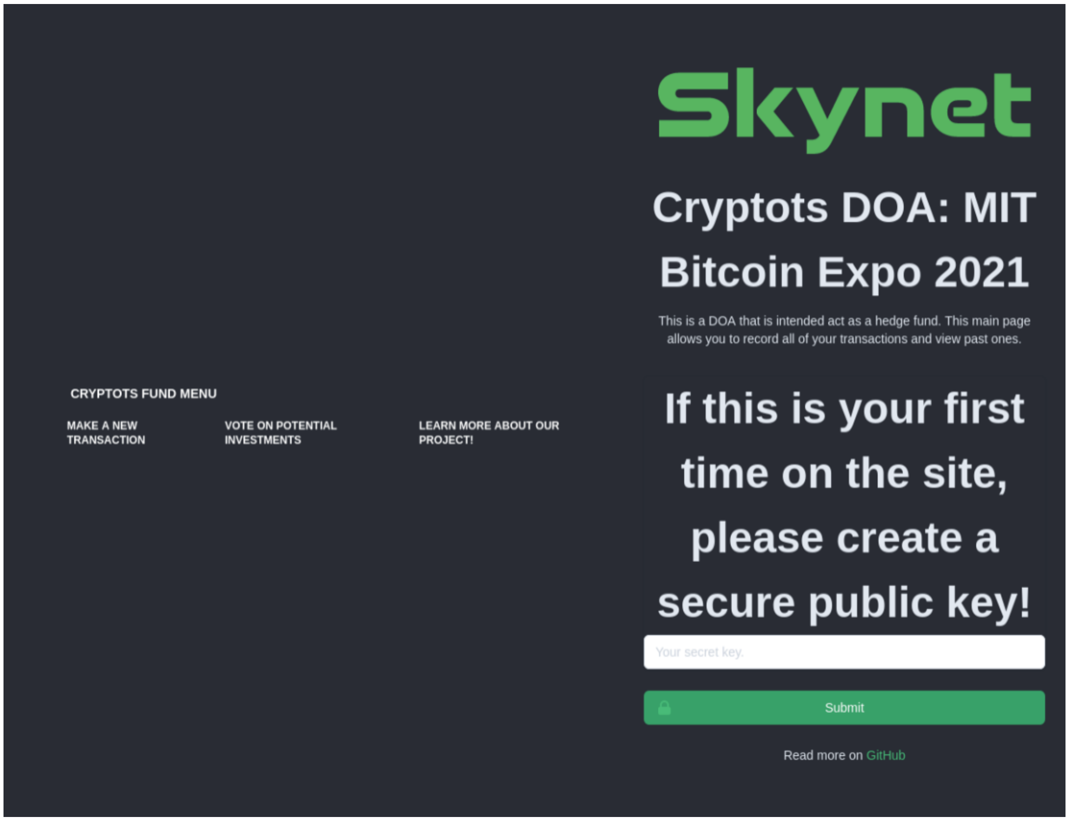

User creates account using Public Algorand Address
User account validated by the network
User gains access to public community
User creates and executes smart contracts with staking rewards

Figure 9

Figure 9 is a deployment model allowing users to sign up for an account with a public Algorand address. Our deployment was facilitated by Skynet. We intend to deplore our project for public use.

**B. Skynet**

We seek to ensure that all transactions and financial data can be uploaded readily by the user to Skynet. This will enable users to have a secure way to view their financial information and have it securely stored remotely.



Cryptots DOA: MIT Bitcoin Expo 2021

This is a DOA that is intended act as a hedge fund. This main page allows you to record all of your transactions and view past ones.

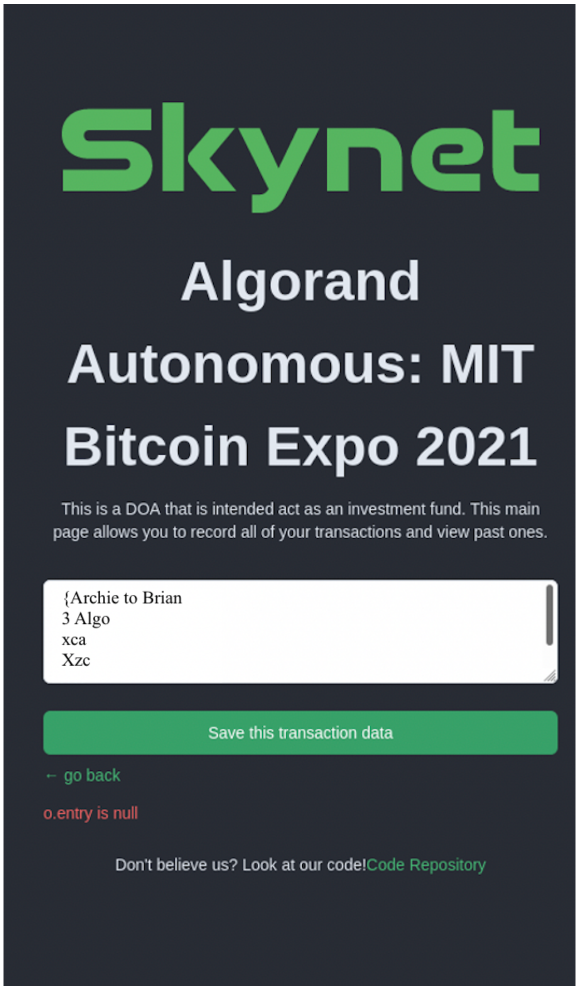
If this is your first time on the site, please create a secure public key!

Read more on GitHub

Figure 10

Users need to first create a secret key to access the transaction base, and then they will be able to upload any transaction-based data to the Skynet servers. This provides users with an additional advantage. Because our web-app takes advantage of the Skynet Login system, users who use our site will be to log into any Skynet web app using the same credentials.

As such, we can provide investors with a decentralized method to invest in both securities and cryptocurrencies. Our goal is to enable users to retain the security that comes with pooling while having independence.



Skynet

Algorand Autonomous: MIT Bitcoin Expo 2021

This is a DOA that is intended act as an investment fund. This main page allows you to record all of your transactions and view past ones.

```
{Archie to Brian
3 Algo
xca
Xzc
```

o.entry is null

Don't believe us? Look at our code!Code Repository

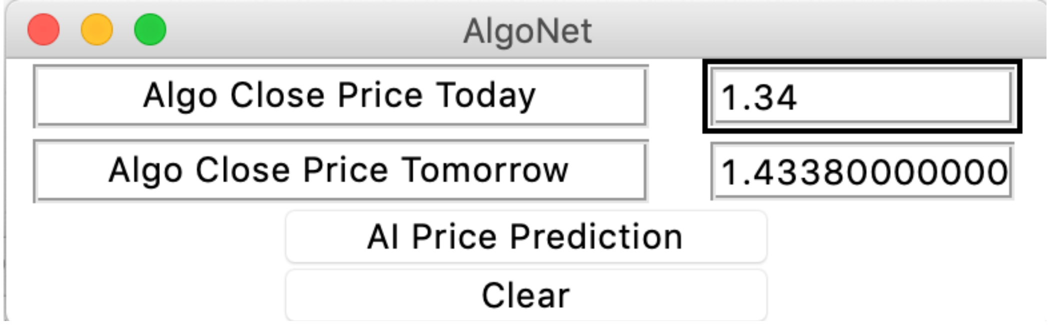
Figure 11

Unlike other investment funds that organized as a DAO, we allow all users to have a say in the investment process.  Our competitive advantage is AlogNet, a neural network for predicting price changes in Algo. This neural network will be directly available to users in the future, which will thus allow them to see when and where they should invest using Algo.

### C. Algonet

One challenge we faced was finding a good problem to solve in the financial industry. We developed some smart contract code for Algorand Swaps using PyTeal. In doing so we identified a problem because blockchain swap markets are very volatile. So, we established another problem statement, predict the close price of Algorand tomorrow.

Algonet, a neural network for predicting daily market volatilities is our solution. AlogNet is a neural network for predicting price changes in Algo.



Algo Close Price Today 1.34

Algo Close Price Tomorrow 1.43380000000

Figure 12

To start, we aggregated a dataset with financial information for top cryptocurrency chains and hardware technology companies. The dataset now includes public financial data for Apple, Intel, NVIDIA, Bitcoin, Ethereum, and Algorand. We then began construction on a neural network to make daily price predictions.

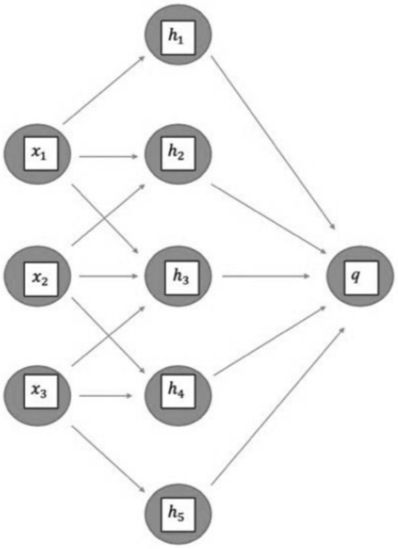


Figure 13

A neural network is a method for generalizing to make predictions.[15] Every neural network has an input layer and an output layer. The depth of the model is defined by the number of layers between the input and output layer. Each layer of hidden neurons acts as a feature extractor by providing analysis of slightly more complicated features. Feature extraction is a method of dimensionality reduction—decreasing input attributes—allowing the observable manifestation of hidden features. The later neurons extract hidden features by combining the previous features of a slightly larger number of neurons. Finally, the output layer observes the whole input to produce a final prediction.

Going forward, we intend to expand and develop the dataset with hardware technology company stock data and cryptocurrency volatility data from Yahoo Finance. The majority of the time spent with deep learning system development is during the pre-processing stage.[16] One way the AlgoNet could control volatilities in Algorand price is by publicly predicting price Algo. Or, by leveraging price predictions to attract investment for the purpose of returning capital gains.

[15] Eugene Charniak, Introduction to Deep Learning 8-9 (2018).
[16] John D. Kelleher, Brenden Tierney, Data Science 97 (2018).

# Conclusion

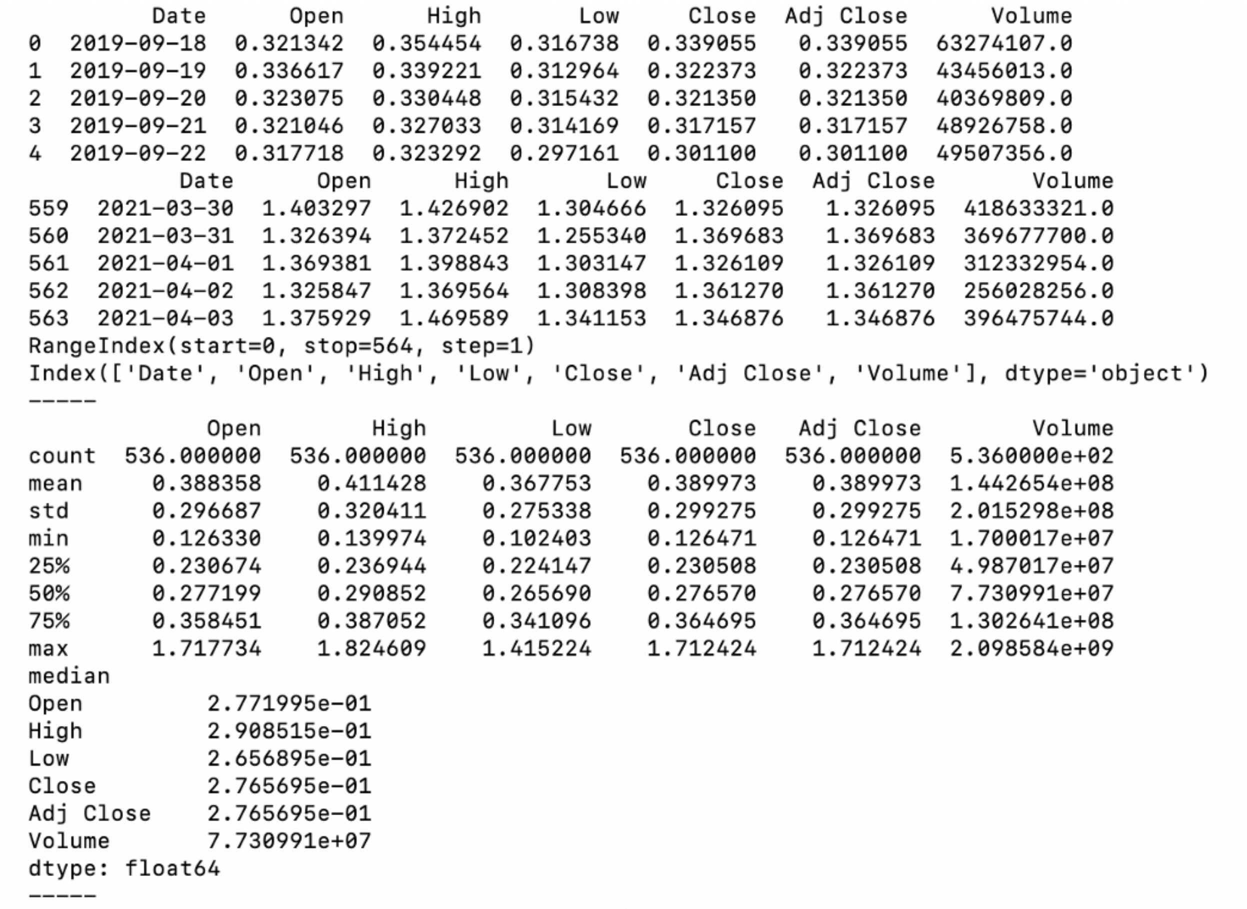
In the digital age, code is law. As one scholar notes, "Cyberspace has an architecture; its code — the software and hardware that defines how cyberspace is — is its architecture."[17] Blockchain networks and traditional organizations have always had problems in governance. This was especially true in financial institutions, where a small number of individuals often made large-scale decisions on the behalf of multiple stakeholders. To remedy this, we offer a decentralized organization that allows users to have an equal say in where their money is invested and allows them to record their transaction data securely.

Across blockchain networks, governance is a complex task with multiple stakeholders having several financial interests. The process by which decisions are made among these stakeholders is often centralized with certain authorities. For example, investment funds typically are controlled by a central authority. However, this decision-making framework limits openness for global community development and collaboration, as well as the personal autonomy of network participants. The solution is a self-updating Decentralized Autonomous Organization (DAO) with three parts.

First, a smart contract platform is made available to DAO users. We explored the Algorand PyTeal Library for contract swaps and developed a novel smart contract software with Tkinter, a Python GUI. Next, a consensus voting mechanism is established to govern the DAO by democratic process. We started with PyTeal voting software and invented a new autonomous consensus protocol. Third, the smart contract platform and voting procedure are deployed through an online interface and DAO platform. So, we used Skynet to make a website for our project and host a place to deploy our minimum viable product. We hope to eventually make the Skynet website the place of all of our cryptographic transactions.

[17] Lawrence Lessig, The Code in Law, and the Law in Code (December 15, 2000).

# Appendix A. Statistical Results



```
         Date      Open      High       Low     Close  Adj Close      Volume
0  2019-09-18  0.321342  0.354454  0.316738  0.339055   0.339055  63274107.0
1  2019-09-19  0.336617  0.339221  0.312964  0.322373   0.322373  43456013.0
2  2019-09-20  0.323075  0.330448  0.315432  0.321350   0.321350  40369809.0
3  2019-09-21  0.321046  0.327033  0.314169  0.317157   0.317157  48926758.0
4  2019-09-22  0.317718  0.323292  0.297161  0.301100   0.301100  49507356.0
           Date      Open      High       Low     Close  Adj Close       Volume
559  2021-03-30  1.403297  1.426902  1.304666  1.326095   1.326095  418633321.0
560  2021-03-31  1.326394  1.372452  1.255340  1.369683   1.369683  369677700.0
561  2021-04-01  1.369381  1.398843  1.303147  1.326109   1.326109  312332954.0
562  2021-04-02  1.325847  1.369564  1.308398  1.361270   1.361270  256028256.0
563  2021-04-03  1.375929  1.469589  1.341153  1.346876   1.346876  396475744.0
RangeIndex(start=0, stop=564, step=1)
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
-----
             Open        High         Low       Close   Adj Close        Volume
count  536.000000  536.000000  536.000000  536.000000  536.000000  5.360000e+02
mean     0.388358    0.411428    0.367753    0.389973    0.389973  1.442654e+08
std      0.296687    0.320411    0.275338    0.299275    0.299275  2.015298e+08
min      0.126330    0.139974    0.102403    0.126471    0.126471  1.700017e+07
25%      0.230674    0.236944    0.224147    0.230508    0.230508  4.987017e+07
50%      0.277199    0.290852    0.265690    0.276570    0.276570  7.730991e+07
75%      0.358451    0.387052    0.341096    0.364695    0.364695  1.302641e+08
max      1.717734    1.824609    1.415224    1.712424    1.712424  2.098584e+09
median
Open         2.771995e-01
High         2.908515e-01
Low          2.656895e-01
Close        2.765695e-01
Adj Close    2.765695e-01
Volume       7.730991e+07
dtype: float64
-----
```