

Decentralized Autonomous Investment Club

Leveraging solidity & python for transparent financial automation



CU-NYC Team 6



**Khulika
Rojanakanoksak**



**Phara
Dorisca**



**Femi
Aina**



**Ogo
Nwanyanwu**

Timeline of Team6 Projects

August 2020
Group 3 created

**Back-Testing, MonteCarlo,
Efficient Frontier Securities
Analysis via Python**

August 2020
Begin project 1
planning

August 2020
Present project 1

September 2020
Begin project 2
planning

**Machine Learning
Automated Trading Bot**

October 2020
Present project 2

November 2020
Begin project 3
planning

**Decentralized Autonomous
Investment Club**

December 2020
Present project 3

Project Summary

Our first group project referenced multiple python libraries and implemented back-testing, monte carlo simulations and efficient frontier portfolio analysis to compare style factors across various ETF investments selected by the team.

For the second group project we created a universe of investable stocks, retrieved and formatted fundamental, sentiment and technical data to feed a machine learning algorithm that produced a long-term bias with short-term tactical entries for alpha-generating automated stock investments via our trading bot on Alpaca.

To expand on projects 1 and 2, we are proposing a decentralized autonomous investment club with smart contract automation for governing funds deposits, voting on proposals, fund management, performance tracking and funds disbursement.

Problems



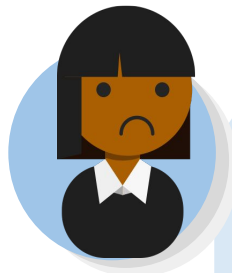
Members

Lack of transparency

- Management of traditional investment club requires central authority

Difficulty organizing groups

- Limited options for alternative funding sources



Admin/Member

High cost to manage capital in investment club

- Collection of funds/fees and distribution of shares, dividends, funds are expensive for small member group

Traditional governance is resource intensive & inefficient

- Centralized trust based system requires extensive compliance to implement

Solutions



Members

Transparent & auditable

- Visibility for investment club process validations

Automate workflows

- Reduce admin costs

Decentralized authority

- Trust-less and open voting governance



Admin/Member

Less costly and more efficient governance

- Eliminates need for centralized compliance

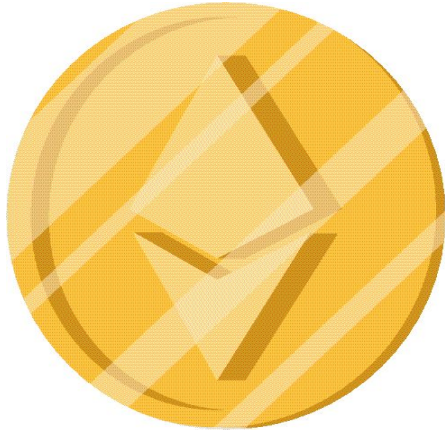
Smart contracts automate workflows

- Codify rules and quickly transact

Secure exchange

- Robust platform with tracking performance and

Project Approach



1

Define and understand problems with organizing a traditional investment club

2

Plan and confirm Decentralized Autonomous Investment Club as solution

3

Research requirements for business layer and design automation to address core use cases

4

Execute development plan via code sprint

5

Test -> Update Design -> Develop and repeat... until Launch!!!

Smart Contracts

- Investors.sol
- Investment_Club_join_acc.sol
- Contract_Ownership.sol
- MembersVote.sol
- SellToken.sol
- InvestmentMintableToken.sol
- MemberInfo.sol



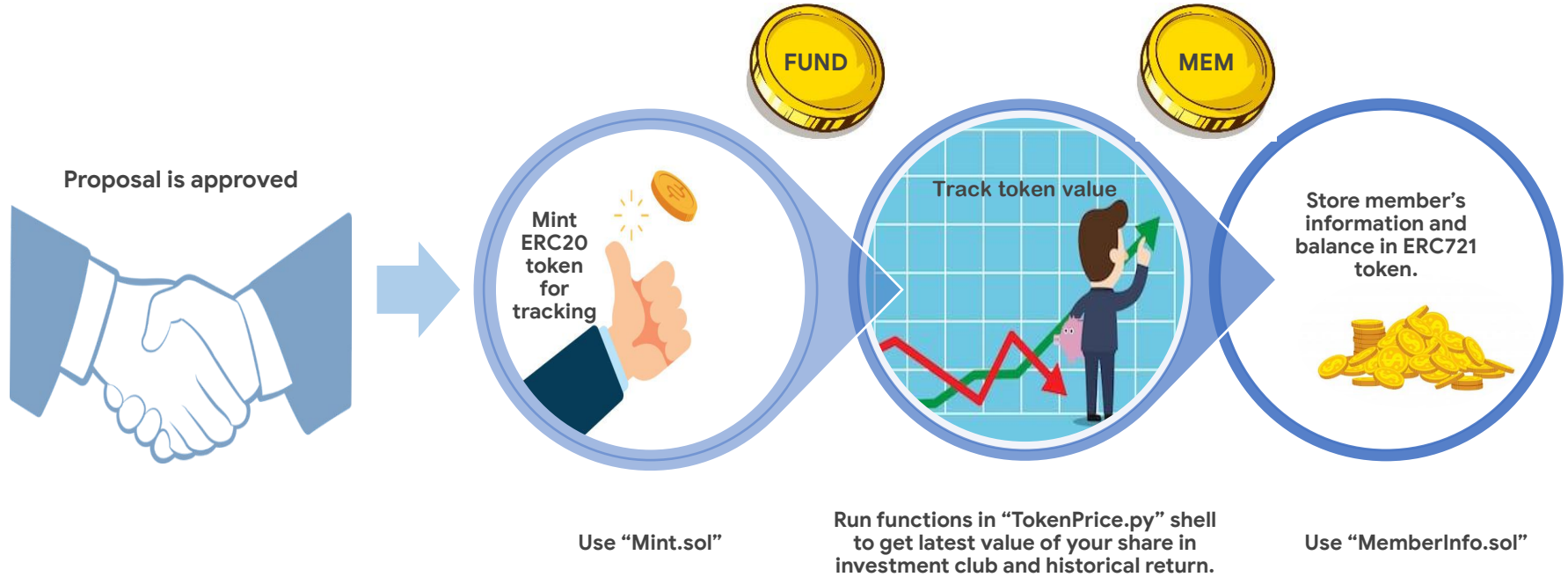
Solidity Code Sample

```
function createProposal(string memory name, uint amount, address payable recipient) external onlyMembers() {
    require(availableFunds >= amount, 'amount too big');
    proposals[nextProposalId] = Proposal(
        nextProposalId,
        name,
        amount,
        recipient,
        0,
        block.timestamp + voteTime,
        false
    );
    nextProposalId++;
}

function vote(uint proposalId) external onlyMembers() {
    Proposal storage proposal = proposals[proposalId];
    require(votes[msg.sender][proposalId] == false, 'members can only vote once for a proposal');
    require(block.timestamp < proposal.end, 'can only vote until proposal end date');
    votes[msg.sender][proposalId] = true;
    proposal.votes += shares[msg.sender];
}

function executeProposal(uint proposalId) external onlyOwner() {
    Proposal storage proposal = proposals[proposalId];
    require(block.timestamp >= proposal.end, 'cannot execute proposal before end date');
    require(proposal.executed == false, 'cannot execute proposal already executed');
    require(((proposal.votes * 100) / totalShares) >= quorum, 'cannot execute proposal with votes # below quorum');
    proposal.executed = true;
}
```

Tracking Process



Tracking Fund Token Value



ERC20 Token

TokenPrice.py



Inception Date:	Nov 30, 2020
Initial club investment:	100 ETH
Total members in club:	4 members
Investment universe:	4 stocks from DowJones30
Stock 1:	Amazon Inc. (AMZN)
Stock 2:	Apple Inc. (AAPL)
Stock 3:	Johnson & Johnson (JNJ)
Stock 4:	Microsoft Inc. (MSFT)

```
token_latest_value('20201130',100, 4, 'AMGN', 'AAPL','JNJ','MSFT')
24.519285000181707

token_return_since_inception('20201130',100, 4, 'AMGN', 'AAPL','JNJ','MSFT')
-1.9228599992731743

token_return_1W('20201130',100, 4, 'AMGN', 'AAPL','JNJ','MSFT')
0.0

token_return_1M('20201130',100, 4, 'AMGN', 'AAPL','JNJ','MSFT')
7.711238929298969
```

Member's Info



ERC721 Token

```
[ { "from": "0x6FE979689c2CB44a744c11A4bF9f94b608F37942", "topic":  
"0xea2873860399415acb372dd6c941321aab08770c5f28bf07c9d9385f806dab7e", "event": "Updatebalance", "args": { "0": "1",  
"1": "5", "2": "", "token_id": "1", "investment_value": "5", "report_uri": "" } } ]
```

Register Member

Owner: contract owner's address

Name: name of registered member

Member_address: member's wallet address

Initial_balance: balance of member's fund

registerMember

owner: 0x17280e566a1ba9ff1ea0FB82cE444A7Db245fF0

name: khulika

member_address: 0x01a316216fe7A4540b2D419499F966d1BA4275

initial_balance: 1

token_uri: 0

transact

View Member's Info

Input is token id

member_info

1

call

0: string: name khulika

1: string: member_address 0x01a316216fe7A4540b2D419499F966d1BA4275

2: uint256: account_balance 1

Update fund token balance

newBalance

token_id: 1

new_value: 5

report_uri: string

transact

member_info

1

call

0: string: name khulika

1: string: member_address 0x01a316216fe7A4540b2D419499F966d1BA4275

2: uint256: account_balance 5

Opportunities to unlock for future functionality

- 100% automation of all workflows to run investment club
 - Internal workflows:
 - Current proposals submitted manually
 - Currently allocation of funds defined by members manually
 - External business layer:
 - Use monte carlo simulations & efficient frontier analysis from project 1 to define investment rules
 - Use security selection and tactical analysis from project 2 to help define monthly investment universe



Thank You