DeFi: Blockchain and Cryptocurrencies

Duke CS+ End of Project Presentations (8/6/2021) b
by Dylan Paul, Oum Lahade, Malika Rawal, Rhys Banerjee

Project Leads: Profs. Luyao Zhang, Yulin Liu, Kartik Nayak, Fan Zhang
Graduate mentor: Derrick Adam

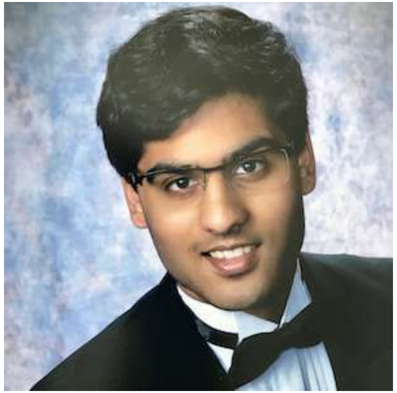
Developers

Dylan Paul

Duke University Class of 2023

Major: Computer Science

Minor: Finance



Oum Lahade

Pratt School of Engineering Class of 2024

Major: Electrical Engineering, Mathematics

Minor: Finance

Malika Rawal

Duke University Class of 2024

Major: Economics and Computer Science

Minor: Finance/ Markets and Management Certification

Rhys Banerjee

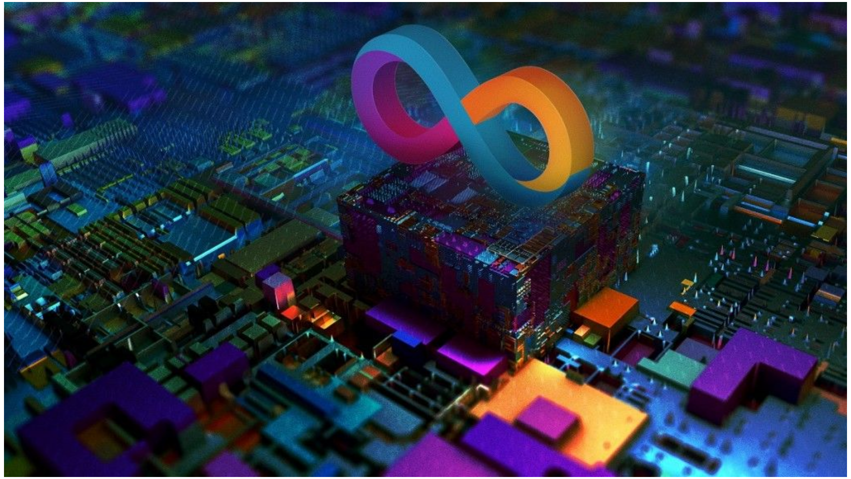
Trinity School of Arts & Sciences Class of 2023

Major: Computer Science

Minor: Mathematics

What is the Internet Computer?

- ❑ Blockchain technology
  - Distributed ledger technology that enables creation of applications that are permissionless, secure, transparent, and borderless
- ❑ Working with the Internet Computer
  - Highly scalable blockchain network compared to Ethereum
- ❑ Utilizing the Internet Computer to create a decentralized banking app comparable to Ethereum
- ❑ Canisters are analogous to smart contracts, which are executable code contracts that run on the blockchain



DeFi Lending Protocols

- **What is DeFi?**
  - De-Fi is a type of finance using blockchain that does not involve any financial intermediaries. Centralized finance systems such as brokerages, exchanges, and banks rely on intermediaries.
- **What opportunities does DeFi present?**
  - DeFi presents a way to improve existing financial services by increasing transparency, scalability, and interoperability as well as making investing more accessible to individuals who may not have been able to participate within the current system
- **What are Lending Protocols?**
  - Lending Protocols allow users to deposit assets into "liquidity pools." The protocol will than take those funds and lend them out to borrowers.
- **What are Borrowing Protocols?**
  - Borrowing protocols are 0% interest rate protocols where users can take out overcollateralized crypto loans by depositing assets and withdrawing a protocol generated stable coin
- **Why do we care?**
  - DeFi poses an interesting solution to the problems of centralized financial systems. By studying DeFi, we can determine the future of finance.

Two Student Projects

ICy - Lending Protocol

WaterPark - Borrowing Protocol

ICy: A Decentralized Lending Protocol for the Internet Computer

Aave - Background

- Decentralized Lending Protocol on the Ethereum Blockchain
  - Allows Users to lend, borrow, and earn interest on Ethereum crypto assets without intermediary's involvement
- Launched November 2017
- Last Closing Price: $259.91
- Total Value Locked: $10,181,720,467
- Lending Pools
  - Connect peers by allowing lenders to place assets into a pool instead of to a specific borrower. Borrowers borrow from the pool instead of a specific lender.
- Borrowing
  - A user must lock up collateral more then the amount borrowed (overcollateralized loans) and must maintain the collateralization ratio.
  - 'aTokens' are given in a 1:1 ratio to a user when they deposit collateral for lending or borrowing (acts as a receipt) and these tokens receive interest.
- Price Oracle, Flash Loans, Arbitrage Opportunities, Increase Leverage in certain positions

Advantages/Disadvantages of Aave

Pros:

- Wide range of DeFi applications and Cryptocurrencies to choose from

- Collateral-Free Flash Loans

- Fixed Fee Flash Loans (Original Amount + Approximately 1% of the amount borrowed)

- Can borrow a different currency and return a different currency

Cons:

- Not beginner friendly, need to know a good bit of technical terms

- 5 minute transaction time

- Overcollateralization, users have to lock up larger amounts of assets then they need to borrow. Not great for ordinary users.

- High transaction fees!

Key Technologies on the Internet Computer

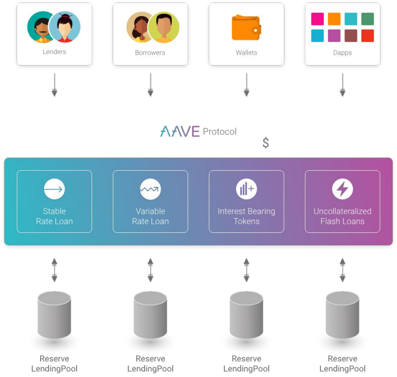
- Chain Key Technology
  - Allows the Internet Computer to finalize transactions in a matter of seconds
  - All nodes used in canisters have secret key shares so that when they are all joined they create a message for the canister to execute.

- NNS (Network Nervous System)
  - Tokenized open governance system that manages the Internet Computer
  - Stores what nodes belong to which subnets, and how to update nodes/what happens when they crash. Can be used for the Price Oracle Canister rather than 3rd party source like Chainlink.

- Reverse Gas Fee Models
  - Canisters pay for computation and ongoing persistence of memory using own gas through cycles.
  - Developers paying for transactions rather than users which allows for more users to access lending protocol features for free.

ICy: A Decentralized Lending Protocol on the Internet Computer

- The ability of users to lend and borrow IC assets with consideration of their collateralization ratio.
  - ICP tokens and Cycles
- Receipts for transactions to the users in terms of AICP and ACycles that earn interest.
- Use of reserve canisters to isolate assets and calculate liquidity.
- Price oracle feed for exchange rate of assets



Lenders

Borrowers

Wallets

Dapps

AAVE Protocol

$

Stable Rate Loan

Variable Rate Loan

Interest Bearing Tokens

Uncollateralized Flash Loans

Reserve LendingPool

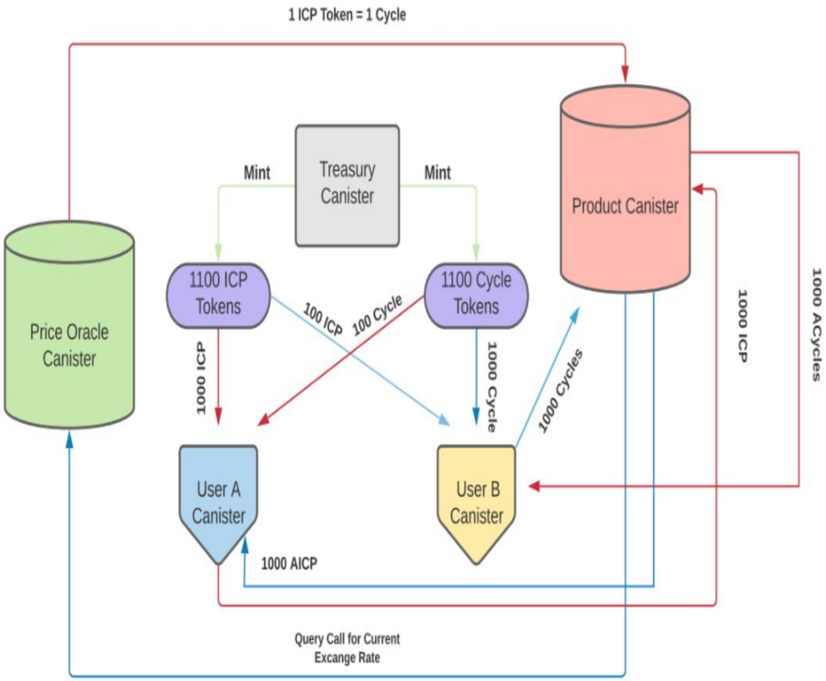
Reserve LendingPool

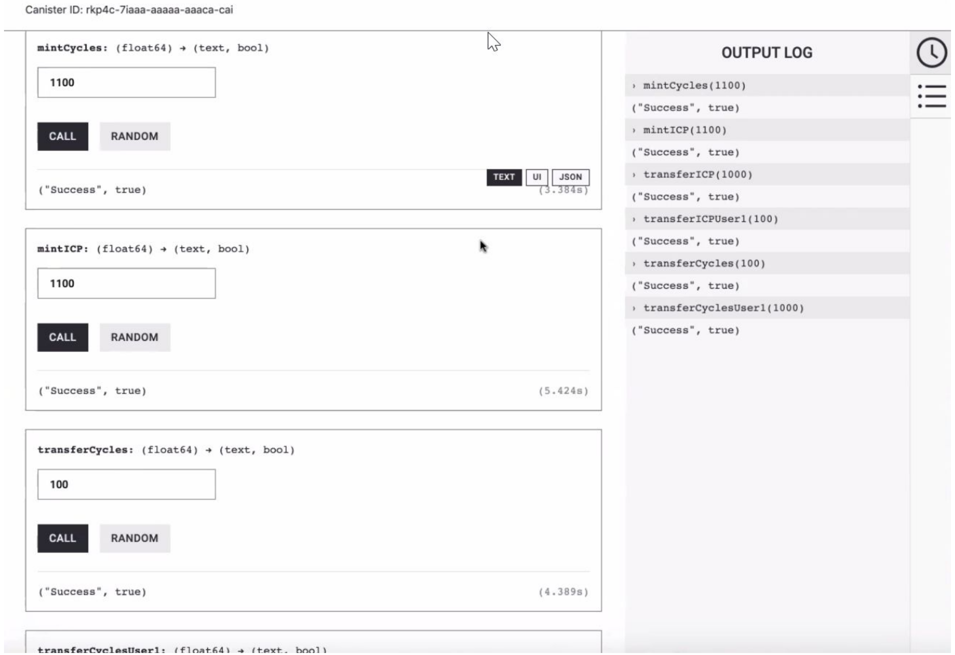
Reserve LendingPool

Reserve LendingPool

ICy Unique Features

- The creation of accounts for users who entrust the product canister with their assets in order to access lending and borrowing.
  - Combats lack of standardization on Internet Computer blockchain as well as lack of wallets.
- Price oracle canister gets the exchange from the NNS.
- There are no transaction fees (gas fees) for users due to reverse gas model.
  - Solves high transaction fee problem faced by users on AAVE which inhibits user access

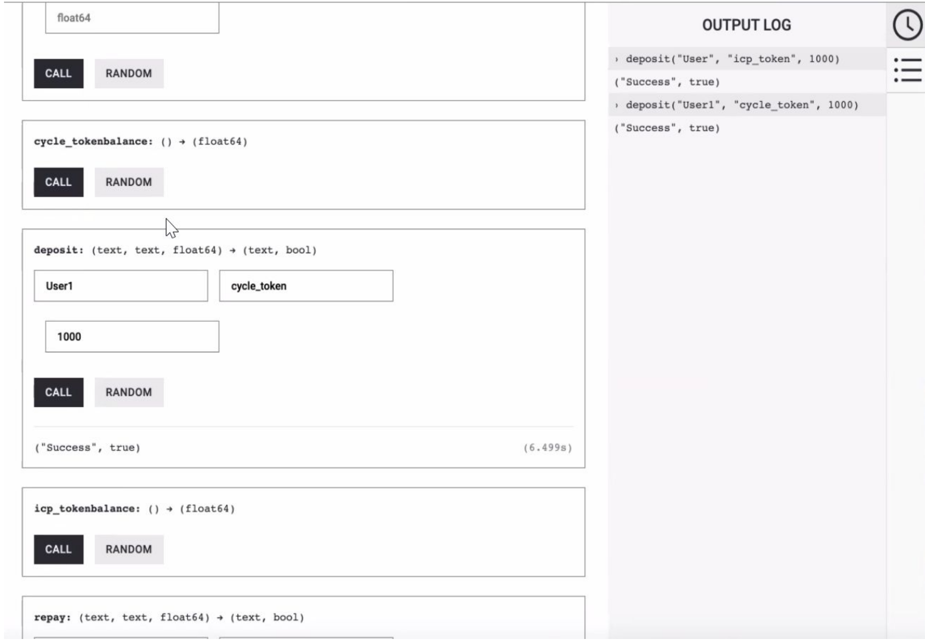




```
Canister ID: rkp4c-7iaaa-aaaaa-aaaca-cai

mintCycles: (float64) → (text, bool)
1100
("Success", true)                                  (3.384s)

mintICP: (float64) → (text, bool)
1100
("Success", true)                                  (5.424s)

transferCycles: (float64) → (text, bool)
100
("Success", true)                                  (4.389s)

transferCyclesUser1: (float64) → (text, bool)
```

```
OUTPUT LOG
› mintCycles(1100)
("Success", true)
› mintICP(1100)
("Success", true)
› transferICP(1000)
("Success", true)
› transferICPUser1(100)
("Success", true)
› transferCycles(100)
("Success", true)
› transferCyclesUser1(1000)
("Success", true)
```

Treasury minting and transferring to user canisters



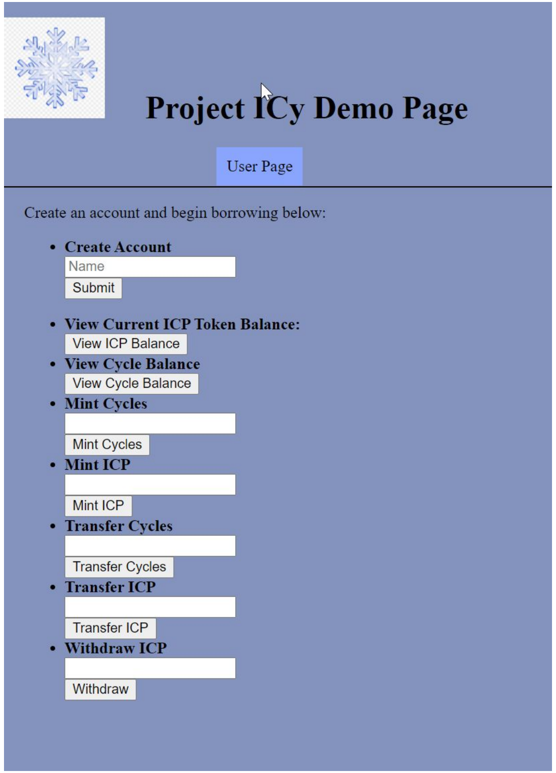
```
float64

cycle_tokenbalance: () → (float64)

deposit: (text, text, float64) → (text, bool)
User1        cycle_token
1000
("Success", true)                                  (6.499s)

icp_tokenbalance: () → (float64)

repay: (text, text, float64) → (text, bool)
```

```
OUTPUT LOG
› deposit("User", "icp_token", 1000)
("Success", true)
› deposit("User1", "cycle_token", 1000)
("Success", true)
```

User canisters depositing to product canister and receiving aCycles and aICP

Candid UI Screenshots

Tab 1/Home Page:

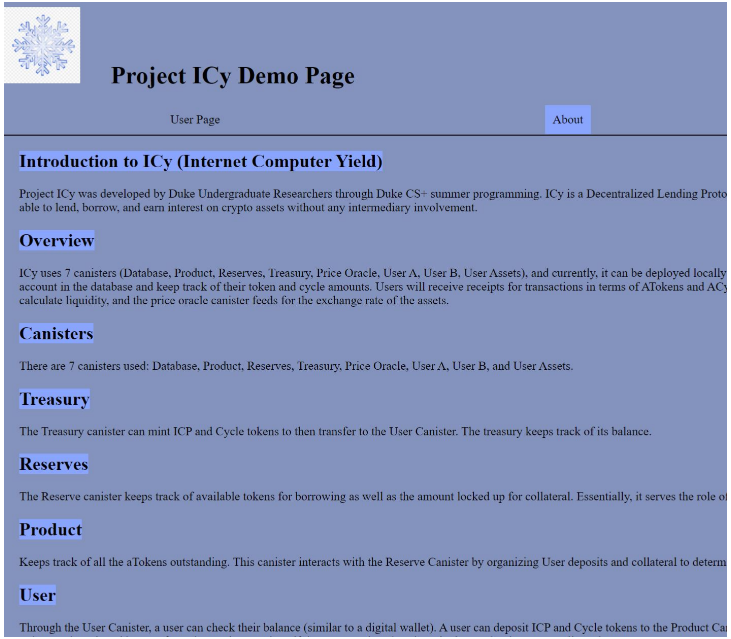


Project ICy Demo Page

Create an account and begin borrowing below:

- **Create Account**
  Name
  Submit
- **View Current ICP Token Balance:**
  View ICP Balance
- **View Cycle Balance**
  View Cycle Balance
- **Mint Cycles**
  Mint Cycles
- **Mint ICP**
  Mint ICP
- **Transfer Cycles**
  Transfer Cycles
- **Transfer ICP**
  Transfer ICP
- **Withdraw ICP**
  Withdraw

Tab 2/About Page:



Project ICy Demo Page

Introduction to ICy (Internet Computer Yield)

Project ICy was developed by Duke Undergraduate Researchers through Duke CS+ summer programming. ICy is a Decentralized Lending Proto
able to lend, borrow, and earn interest on crypto assets without any intermediary involvement.

Overview

ICy uses 7 canisters (Database, Product, Reserves, Treasury, Price Oracle, User A, User B, User Assets), and currently, it can be deployed locally
account in the database and keep track of their token and cycle amounts. Users will receive receipts for transactions in terms of ATokens and ACy
calculate liquidity, and the price oracle canister feeds for the exchange rate of the assets.

Canisters

There are 7 canisters used: Database, Product, Reserves, Treasury, Price Oracle, User A, User B, and User Assets.

Treasury

The Treasury canister can mint ICP and Cycle tokens to then transfer to the User Canister. The treasury keeps track of its balance.

Reserves

The Reserve canister keeps track of available tokens for borrowing as well as the amount locked up for collateral. Essentially, it serves the role of

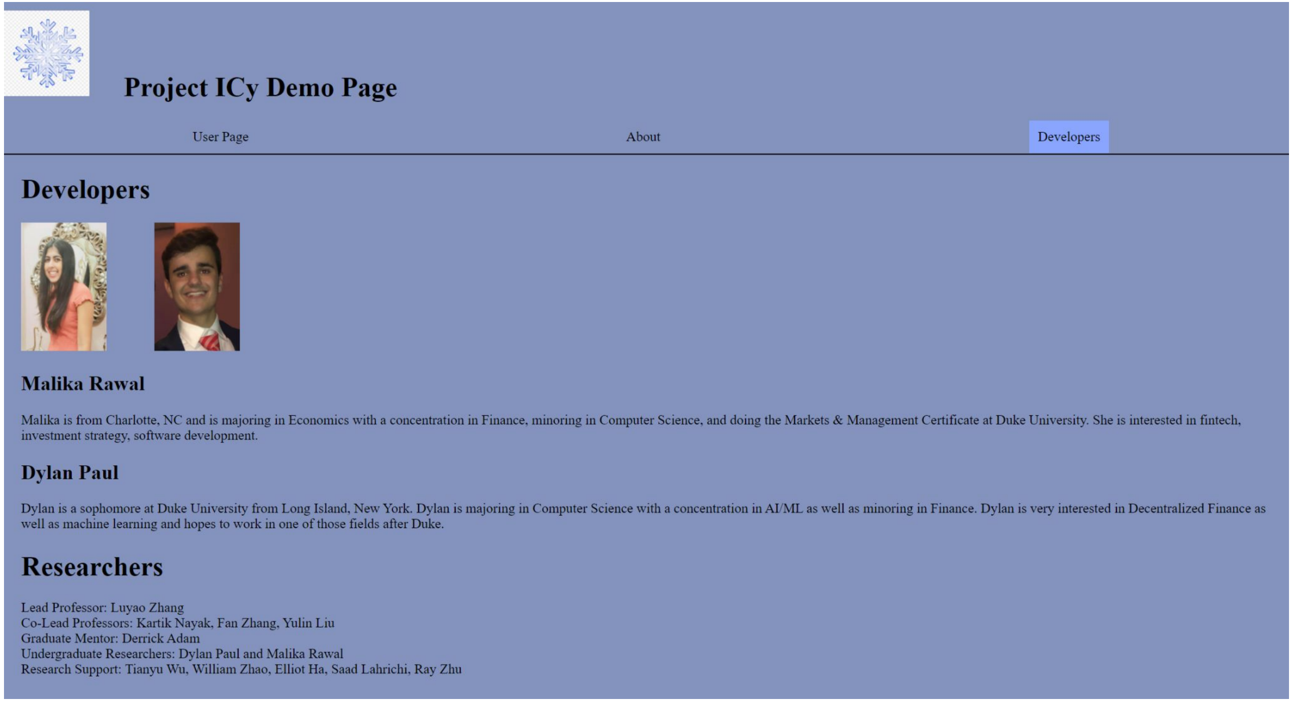
Product

Keeps track of all the aTokens outstanding. This canister interacts with the Reserve Canister by organizing User deposits and collateral to determ

User

Through the User Canister, a user can check their balance (similar to a digital wallet). A user can deposit ICP and Cycle tokens to the Product Ca

Tab 3/ Developers Page



Project ICy Demo Page

Developers

Malika Rawal

Malika is from Charlotte, NC and is majoring in Economics with a concentration in Finance, minoring in Computer Science, and doing the Markets & Management Certificate at Duke University. She is interested in fintech, investment strategy, software development.

Dylan Paul

Dylan is a sophomore at Duke University from Long Island, New York. Dylan is majoring in Computer Science with a concentration in AI/ML as well as minoring in Finance. Dylan is very interested in Decentralized Finance as well as machine learning and hopes to work in one of those fields after Duke.

Researchers

Lead Professor: Luyao Zhang
Co-Lead Professors: Kartik Nayak, Fan Zhang, Yulin Liu
Graduate Mentor: Derrick Adam
Undergraduate Researchers: Dylan Paul and Malika Rawal
Research Support: Tianyu Wu, William Zhao, Elliot Ha, Saad Lahrichi, Ray Zhu

Link to GitHub Repository

https://github.com/Duke-CS-Plus-IC/ICy2.git

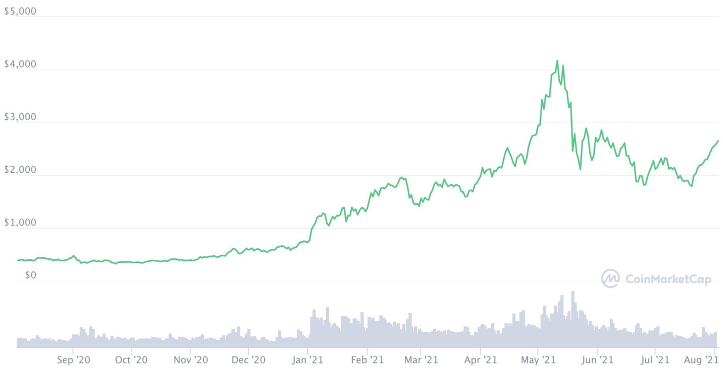
Link to WarpWire Demonstration Video

https://warpwire.duke.edu/w/legFAA/

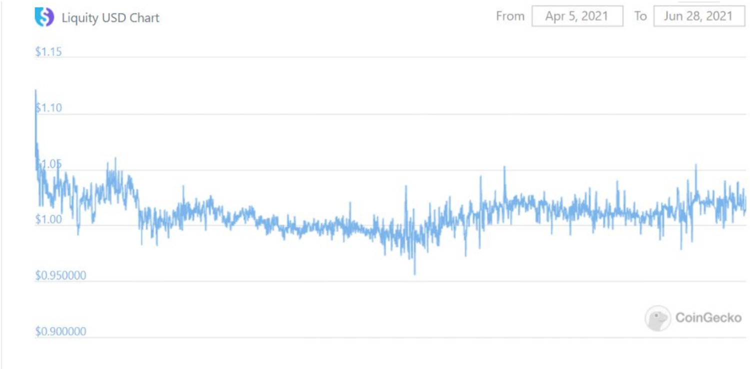
WaterPark: A Borrowing Protocol for the Internet Computer

Background

- The system generates stable coins (SDR) against ICP as collateral.
- Users deposit ICP into individualized troves created for each user.
- These troves are overcollateralized at a rate of 110%
- SDR tokens offer a basis for the creation of stable smart-contracts



Price Volatility Experienced by Ether (ETH)

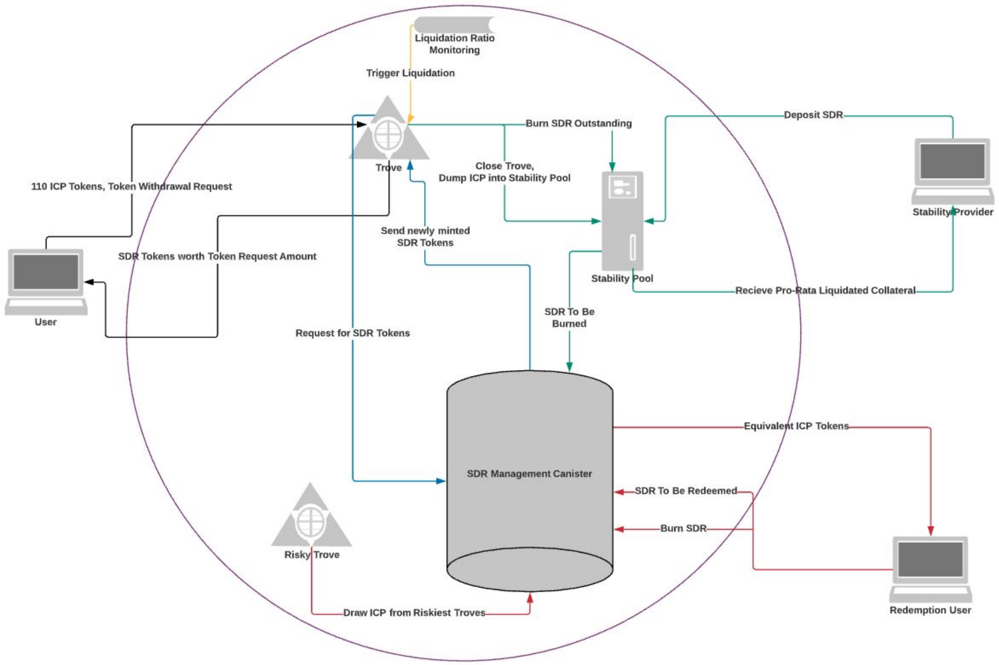


Price Stability Generated by Liquity (LUSD)

Inherited Features

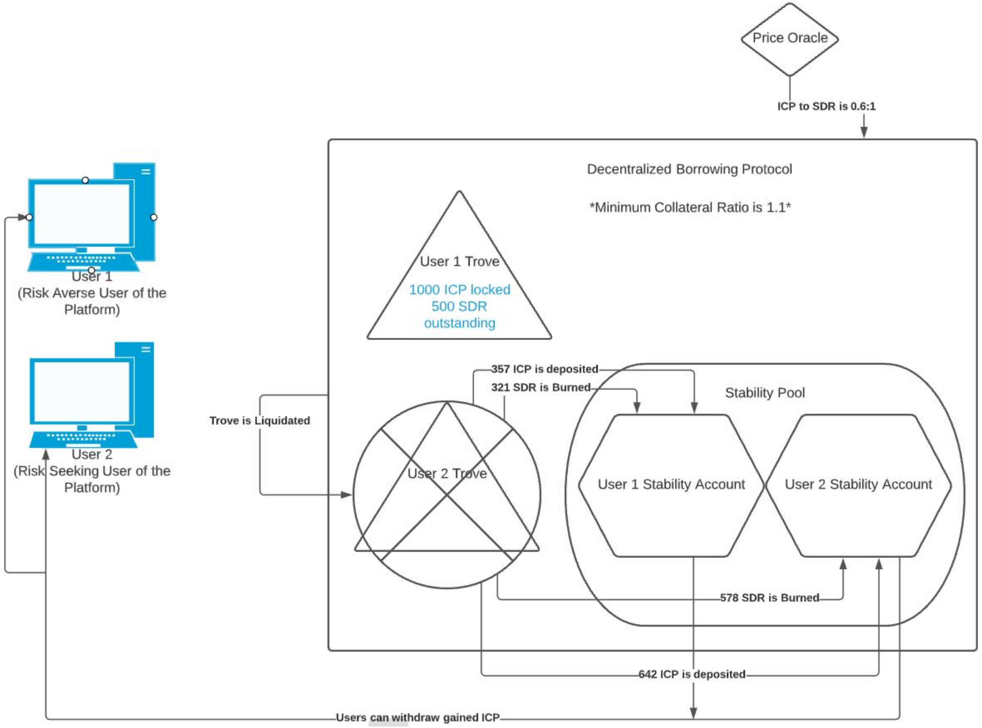
- General Architecture
  - Troves
  - Stability Pool
  - ERC-20 Style Coin
  - Liquidation Mechanism
- WaterPark also utilizes the LUSD stable coin's general stability mechanism
  - Downwards price pressure from borrowing protocol
  - Upwards price pressure from redemption mechanism

Decentralized Borrowing Protocol

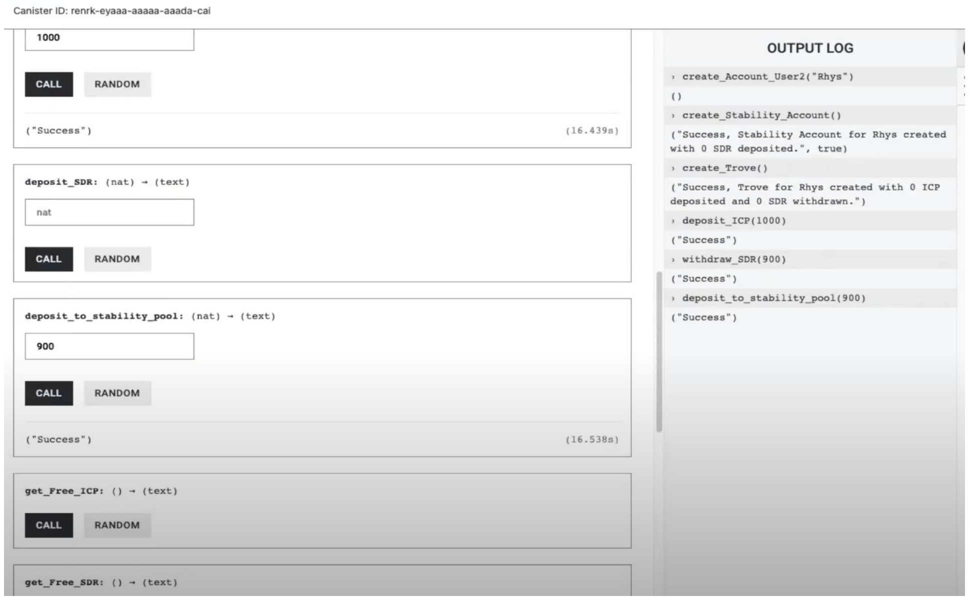

Liquidation Ratio Monitoring
Trigger Liquidation
Burn SDR Outstanding
Deposit SDR
Trove
Close Trove, Dump ICP into Stability Pool
110 ICP Tokens, Token Withdrawal Request
Stability Provider
Send newly minted SDR Tokens
SDR Tokens worth Token Request Amount
Stability Pool
Recieve Pro-Rata Liquidated Collateral
User
SDR To Be Burned
Request for SDR Tokens
Equivalent ICP Tokens
SDR Management Canister
SDR To Be Redeemed
Burn SDR
Risky Trove
Redemption User
Draw ICP from Riskiest Troves

Differing Features

- ERC-20 style stable coin pegged to SDR (IMF's Special Drawing Rights)
- WaterPark uses trove-like stability accounts to distribute rewards for users that deposit to the stability pool
- WaterPark stores all data on-chain on the Internet Computer network
- Front End Applications
- Dynamic Scaling of Canisters by the NNS



Front-End

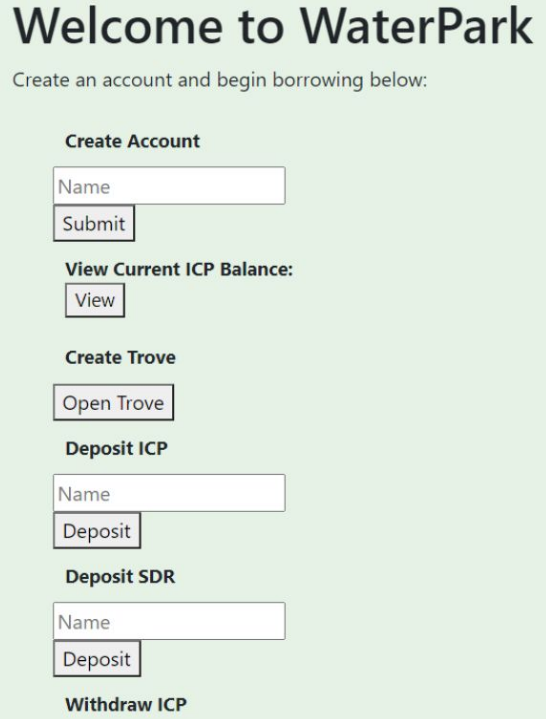


Canister ID: renrk-eyaaa-aaaaa-aaada-cai

```
1000
("Success")                                            (16.439s)

deposit_SDR: (nat) → (text)
nat

deposit_to_stability_pool: (nat) → (text)
900
("Success")                                            (16.538s)

get_Free_ICP: () → (text)

get_Free_SDR: () → (text)
```

```
› create_Account_User2("Rhys")
()
› create_Stability_Account()
("Success, Stability Account for Rhys created
with 0 SDR deposited.", true)
› create_Trove()
("Success, Trove for Rhys created with 0 ICP
deposited and 0 SDR withdrawn.")
› deposit_ICP(1000)
("Success")
› withdraw_SDR(900)
("Success")
› deposit_to_stability_pool(900)
("Success")
```

User's perspective to create an Account, stability account, and trove along with deposit and withdrawal features



Welcome to WaterPark

Create an account and begin borrowing below:

Create Account

Name

View Current ICP Balance:

Create Trove

Deposit ICP

Name

Deposit SDR

Name

Withdraw ICP

HTML and JavaScript Front-end Home page

Resources

GitHub Repository

https://github.com/Duke-CS-Plus-IC/Waterpark

WarpWire Demonstration Video

https://warpwire.duke.edu/w/X-cFAA/

Research Proposals

Orthogonal Persistence (Rhys and Oum)

- Data storage on the Internet Computer is much more efficient and cheaper than data storage on the Ethereum network
- Question: how do differences in persistence alter the storage efficiency of WaterPark?
- Question: How does Dfinity's implementation of orthogonal persistence scale compared to existing decentralized data storage solutions?

Gas Models (Dylan)

- Analyzing and comparing transaction fee mechanisms or models on two blockchains: the gas model on Ethereum and the reverse gas model on the Internet Computer, and then determining how these models affect efficiency, scalability, and sustainability of applications on these blockchains

On-Chain Governance and Liquid Democracy (Malika)

- A blockchain technology's form of governance is incredibly important because as there is an increased scale of users, the system needs to be more cautious with approving unknown protocols since there is more at stake financially wise.
- By analyzing the differences between the governance protocols on the IC and Ethereum, we can discover potential future implications and possibilities for additional change.

Thank you!