Debitable - Decentralized Banking System using Radix

Apratim Shukla, Priyal Soni, Kshitij Sinha, Manjunath Jakaraddi, Rohith Raj Srinivasan
Problem Statement

- **Shortcomings of Traditional banking systems:**
  - Centralized system methodology
  - Inefficient
  - Vulnerable
  - Non-transparent

- **Decentralized Finance (DeFi)** overcomes these shortcomings.
Debitable

- **Full-stack DeFi banking web application**

- **AggieCoin AC**

- Allows users to:
  - View balance
  - Deposit / withdraw funds
  - Borrow / lend ACs
Lending & Borrowing

- Accept lending from lenders
- Allow lenders to get back along with rewards
- Rules preventing debtors from defaulting
- Allow for borrowing non-collateral loans
- Allow borrowers to repay the loan
- No multiple loans at the same time
- Loyalty levels for completing multiple lending and borrowing.
Lending & Borrowing

1. Lend AC Tokens

2. Receive LND Tokens

Lender → XRD Main Pool

XRD Main Pool → LND Main Pool

LND Main Pool
Lending & Borrowing

3. Ask for withdrawal

Lender

4. Receive AC with rewards

XRD Main Pool

3. Ask for withdrawal

LND Main Pool
Lending & Borrowing

1. Ask for a loan
2. Receive the AC
Lending & Borrowing

1. Repay the loan

Borrower → XRD Main Pool
Decentralized Banking System using Radix

Deposit, Withdraw, Borrow and Lend AggieCoin with our DeFi App. Instantaneous, Hassle-free and Secure.
Tech Stack Used

Radix

- Smart contract created using Scrypto (Programming language - Rust)
- FrontEnd - HTML, CSS, JavaScript, jQuery
- Blockchain connected with frontend using AlphaNet (TypeScript)
- Dependencies/Bundles handled using Snowpack in npm
Tech Stack Used (cont)

Ethereum

- Smart contract created using Solidity
- For test-net, using Ganache + Truffle
- FrontEnd - HTML, CSS, JavaScript, jQuery
- Blockchain connected with frontend using JavaScript (Web3.js)
- Dependencies/Bundles handled using Snowpack in npm
# Comparison

<table>
<thead>
<tr>
<th>Radix</th>
<th>Ethereum</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC: ~720</td>
<td>LOC: ~2000</td>
</tr>
<tr>
<td>Language: Scrypto (Rust based)</td>
<td>Language: Solidity</td>
</tr>
<tr>
<td>Asset Oriented Programming (much easier to understand for someone familiar with OOP)</td>
<td>Implementation in Solidity is harder to maintain the code base due to many dependencies and libraries.</td>
</tr>
<tr>
<td>Supports composition of transactions through manifest</td>
<td>Doesn’t support composition of transactions.</td>
</tr>
</tbody>
</table>
Upcoming features for v2.0

- OAuth 2.0 mechanism for secure users’ Authentication verification
- Lending and Borrowing mechanisms to be integrated into the frontend for a full fledged working prototype.
- Introduce real world data (time and dynamic interest rates)
- Add session management for consistency.
- Compose transactions using manifest.
Find our App live here

https://debitable.org/
THANK YOU!