CS+ Decentralized Finance: Blockchain and Cryptocurrency on the Internet Computer

Meeting Minutes

7/1/2021 10:00 – 11:00 AM EST

Present:

Prof. Luyao Zhang - Lead

Prof. Kartik Nayak – Co-Lead

Prof. Yulin Liu – Co-Lead

Derrick Adam – Graduate Mentor

Dylan Paul – Full-Time Researcher

Urjit Banerjee – Full-Time Researcher

Oum Banerjee – Full-Time Researcher

Malika Rawal – Full-Time Researcher

Tianyu Wu – Research Support

Elliot Ha – Research Support

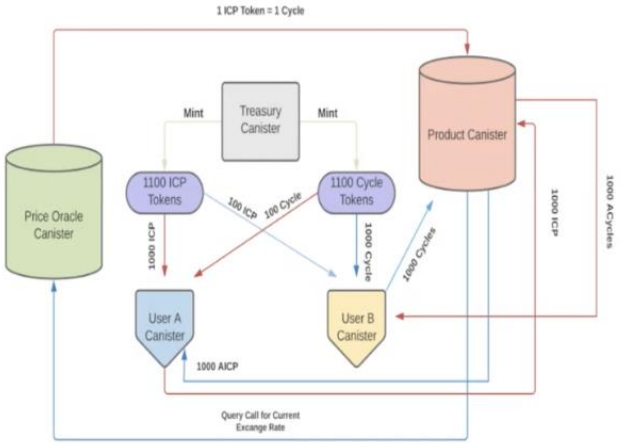
Saad Lahrichi – Research Support

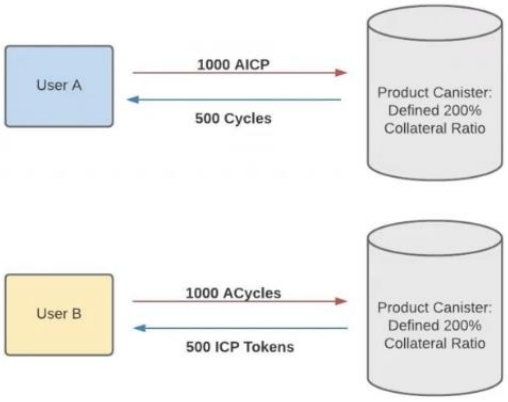
Ray Zhu – Research Support

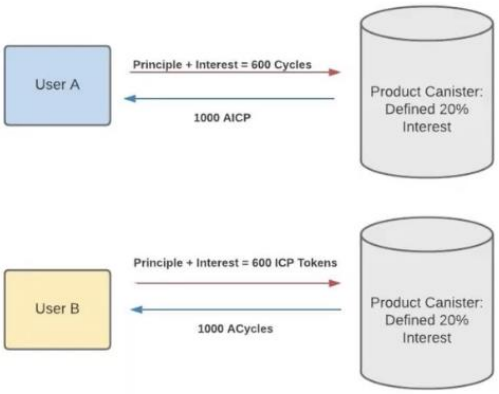
Dylan & Malika:
Team Name: Icy – play on words Internet Computer and it being icy

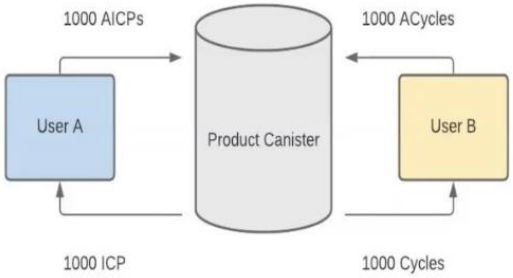
Dylan presented the following:

Team Name: ICy
Update from Week 4: Dylan Paul

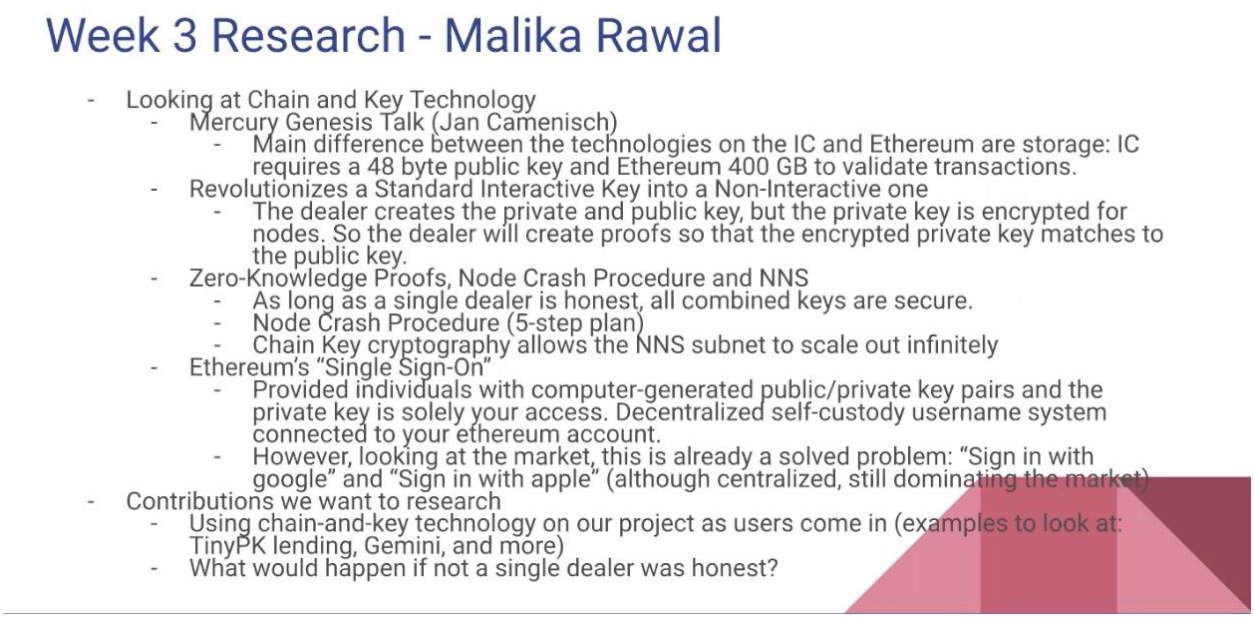








Malika presented:



Prof. Fan:
To verify a transaction, you need a state. How does transaction verification work within 400 GB? The amount of storage is related to the state of the system. This reminds of stateless verification proposed in Ethereum you retrieve whatever state you need on the fly instead of storing it.

Prof. Yulin:
How can you reduce the state from 400 GB to 48 Byte? Do more research on this. DFINITY provides a lot of talks and videos on this topic.

Rhys presented:

Research from Week 3: Rhys Banerjee

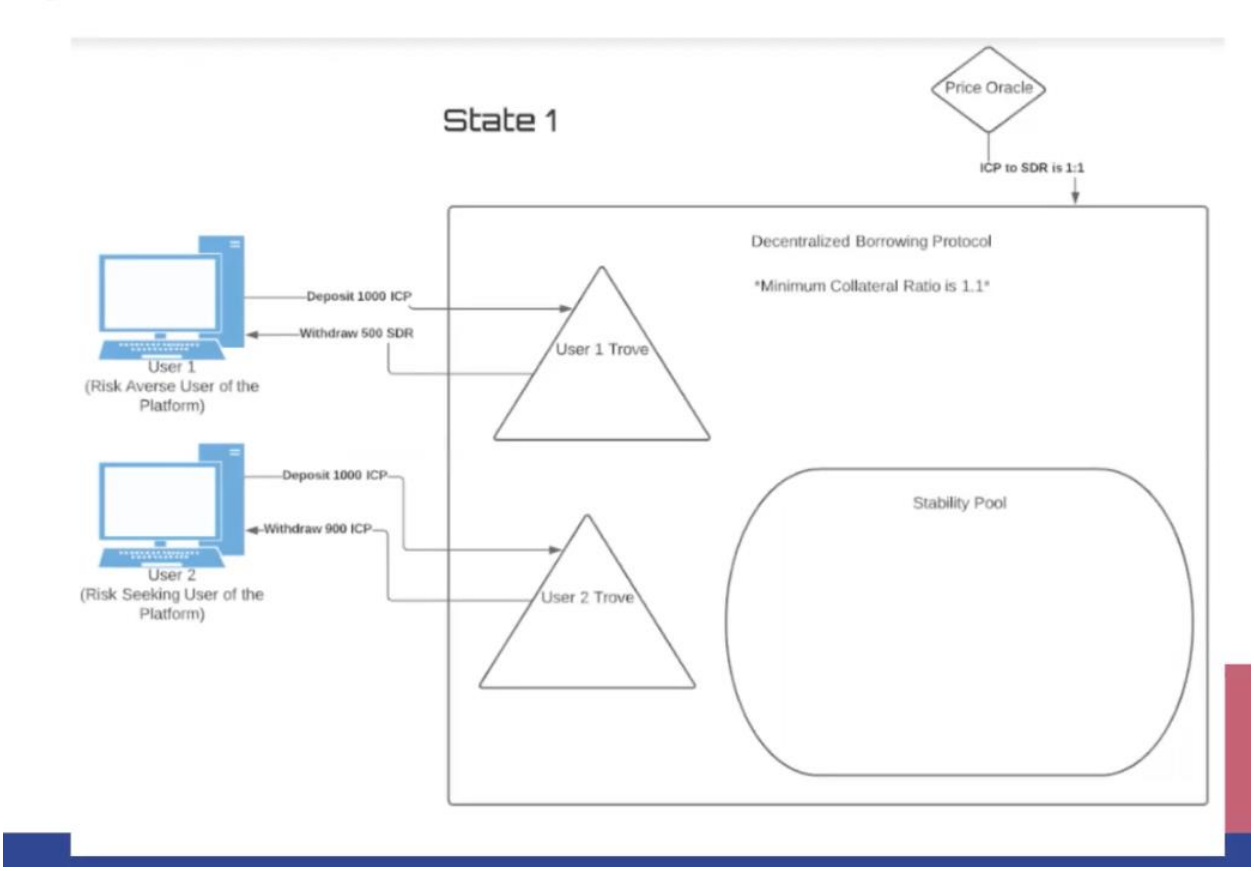
- Topic of Choice: Orthogonal Persistence
- Current Literature:
  - Modern applications of Orthogonal Persistence.
    - This includes attempts to create an orthogonally persistent Java
  - Criticisms of Orthogonal Persistence
    - Other forms of persistence are favorable to Orthogonal Persistence in this argument.
    - Argues in favor of the use of “coarse-grained objects” in software development.
    - Persistence is still orthogonal to type but where there are typically restrictions on what objects can reference what objects.
    - Such systems are often structured around 'fine-grained objects' and 'coarse-grained objects', where coarse-grained objects are used as the units of permissions, locking, transferral and so on.
- Contributions we want to research
  - Orthogonal Persistence in its applications to DeFi and Fintech.
  - How we can use Motoko to utilize orthogonal persistence in our decentralized borrowing protocol.

Oum presented:

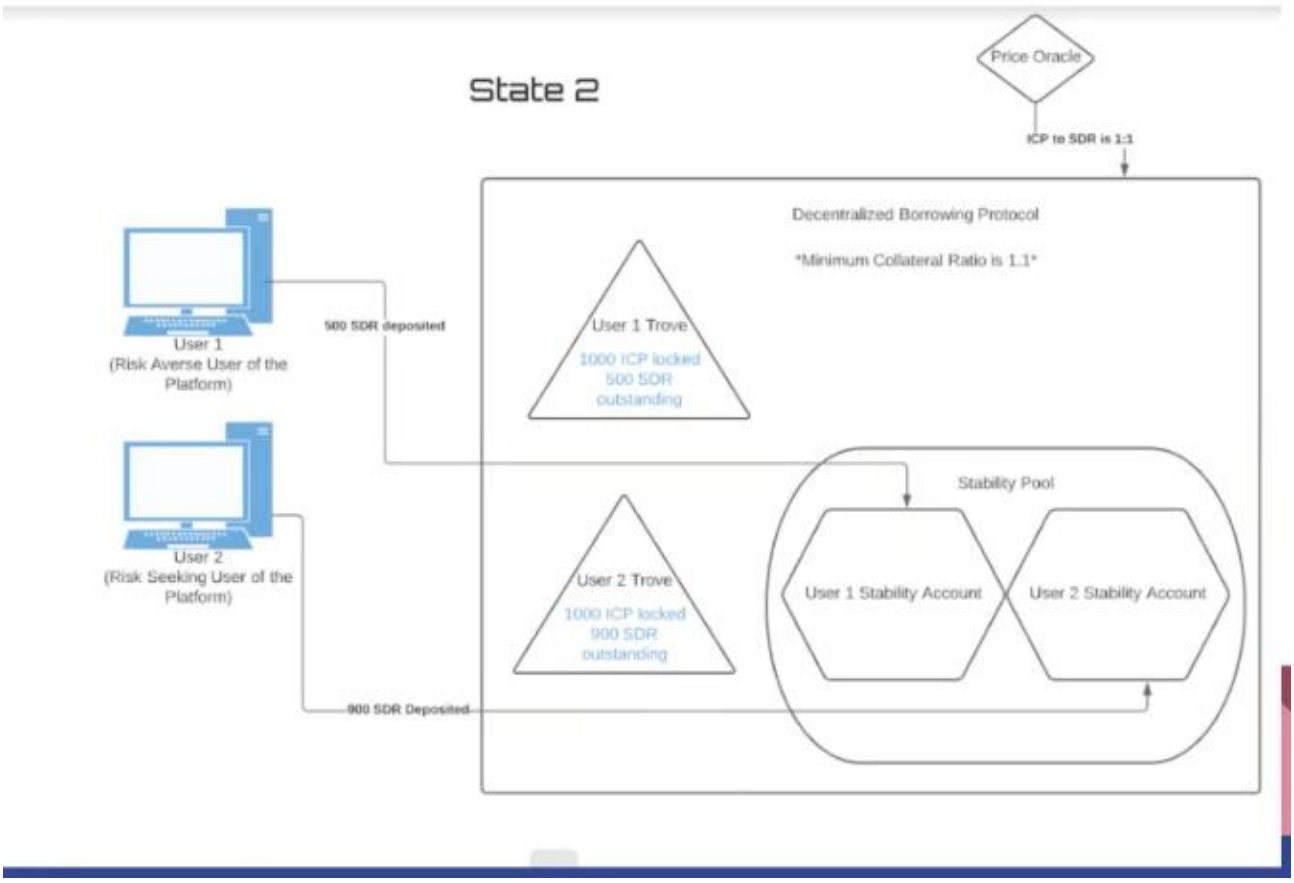
Research from Week 3: Oum Lahade

- Research Question (Last Week): Can Tokenization, NFTs, and Smart-Contracts solve the issue of under collateralization in crypto loans and pave the way for corporate lending via crypto markets?
- Current Literature:
  - Asset Tokenization
    - Deloitte Research
    - EY Research
- Contributions we want to research
  - Utilizing Tokenized assets as collateral in smart-contracts
  - Exploring the “efficiency” of secondary markets for Tokenized Assets
  - Exploring the question of anonymity in regards to large loans collateralized via Tokenized Assets (i.e. is there a way to conduct this in a completely anonymous way; is it right to do so, or should trust be involved?)
  - Regulatory Implications of this type of lending (especially in regards to anonymous debtors) (at the end of the day, Tokenized Assets functions a lot like a security)
- Code Update
  - Graphics (Next Few Slides)
  - MVP Complete
  - Next Steps:
    - Play with ICP tokens, create SDR stablecoin, redemption / price - stability

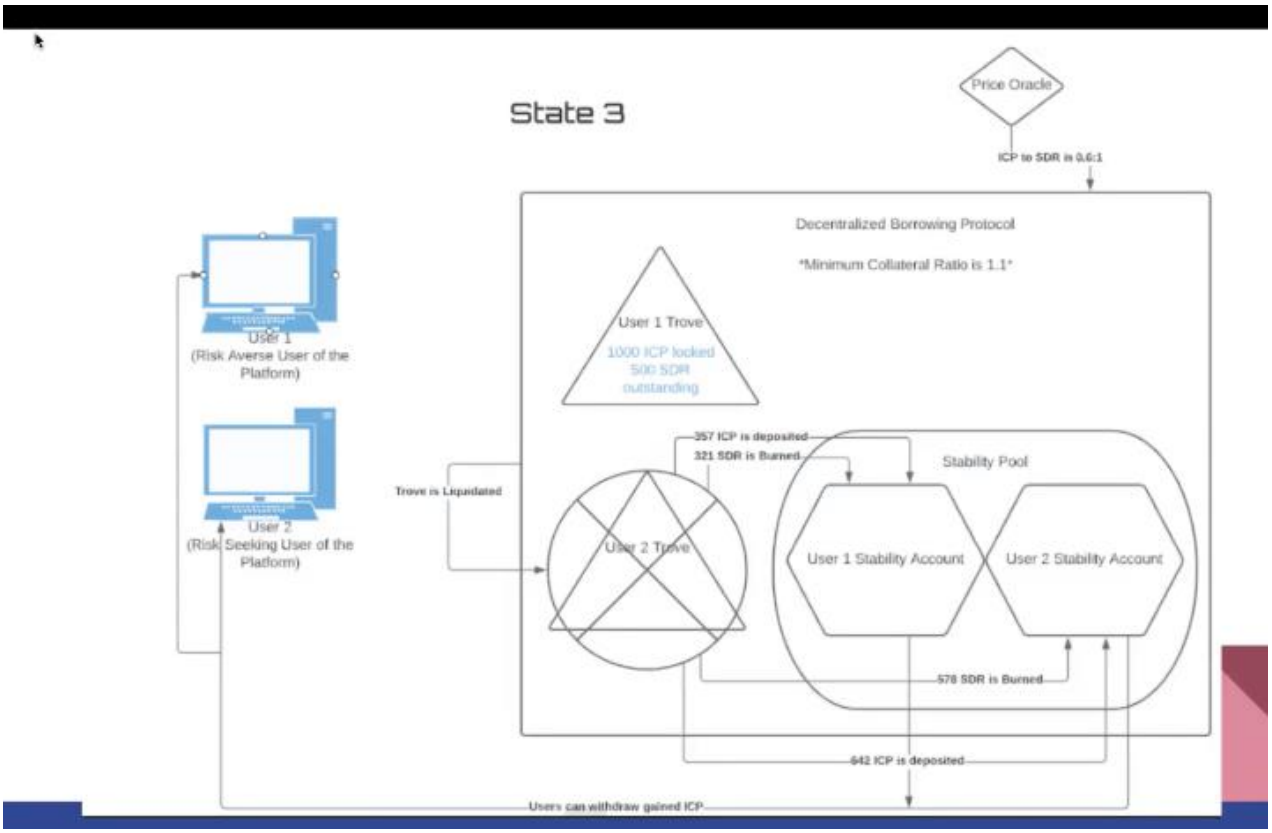
Diagram from user perspective:



There is minimum collateral ratio of 1.1. The risk averse user will deposit 1000 ICP inside the trove and then withdraw 500 SDR from the trove. The risk seeking user will deposit 1000 ICP and then withdraw 900 SDR (just above the minimum collateral ratio).



Here is user 1 and 2 depositing SDR they took out of into the stability pool.



The ICP to SDR ratio dropped to 0.6:1. The product canister will recognize that user 2's trove is in default because its collateral ratio is below 110%. That trove will be liquidated. The proportional amount of SDR outstanding will be burned from user 2's stability account. The 321 SDR for user 1 burned. 357 ICP is deposited into user 1's stability account. 642 ICP is deposited into user 2's stability account.

Prof. Zhang:
Dylan and Oum have very general research questions. Both would need to identify a key technology on the IC that can facilitate your research question.

Dylan Paul:
Ideas for a research question: 1) gas model: Ethereum (gas fee model) vs DFINITY (reverse gas model)
2) flash loans – risks posed on a platform like Ethereum and how that can be implemented differently

Prof. Yulin research topic commentary: gas fee research - DFINITY has canisters pay and Ethereum has users pay. Study how this affects your applications and how it affects users. Deploy Liquity and Ave on the IC. Figure out a revenue model for your canister, so it charges some ICP tokens, but eventually convert all the tokens you receive as profit to cycles because the canister needs to pay cycles.

For flash loans – Liquity – as a borrower you deposit/pledge some ICP tokens in Liquity and then you borrow some SDR tokens. Then you spend these SDR tokens. Some days later, you want to redeem your collateral, so you need to get SDR from the market. By now, you don't have any money, so how do you get that? You could borrow the SDR token from Ave or now it's called Icy. As a Liquity user, you could use a flash loan from ICy and borrow SDR token, pay them back, and get your ICP token collateral and then exchange ICP to SDR and then pay SDR back. You can do all these transactions in one transaction. It allows you to pay back your debt without buying from market some SDR tokens. *flash loan is more complicated, but Dylan, you can do gas fee research

Another potential research question/area to work on in the future:
Figure out the algorithmic interest rate for Aave, now Icy, is fixed at 20%. That's not correct. When you have many depositors and borrowers, you need to figure out an algorithmic interest rate model. You can borrow this from Aave which has a very standard interest rate model.

DFINITY has several test nets, called sodium testnet.

Prof. Zhang: 3 ways to solve ICP problem:

1) For frontend deploy through Fleek. For backend, not possible right now
2) Once ready for the application and created a letter id, they will send us some ICP for us education funds to deploy to the main chain.
3) If we need more, they need to take some time to create an educational session which includes the ICP to help us finish the last stage. To do that, we need to at least have an application and research done before the end of the CS+ project.

Research topics for students:
Oum: Orthogonal Persistence
Malika: Orthogonal Persistence
Dylan: Reverse Gas Fee
Rhys: Chain Key Technology

Prof Yulin:
*Liquity team needs to find a name

Liquity is a borrowing platform, but the key component is the stability of its stable coin. Liquity has a stability pool mechanism to guarantee the price parity. That's one way to keep the price pegged to the US dollar. That is another research question you can think about – price stability.