

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

## Meeting Minutes

7/29/2021 10:00 – 11:00 AM EST

Present:

Prof. Luyao Zhang – Lead

Prof. Yulin Liu – Co-Lead

Prof. Kartik Nayak – Co-Lead

Prof. Fan Zhang – 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

Ray Zhu – Research Support

William Zhao – Research Support

Oum & Rhys: Waterpark

**Oum presented:**

Canister ID: renrk-eyaaa-aaaa-aaaa-cal

### Candid UI

Browse and test your API with our visual web interface.

**close\_Trove: (nat) → (text)**

**CALL** **RANDOM**

**collect\_ICP\_from\_Stability\_Account: () → (text)**

**CALL** **RANDOM**

**create\_Account\_User2: (text) → ()**

**CALL** **RANDOM**

() (6.195s)

**create\_Stability\_Account: () → (text, bool)**

**CALL** **RANDOM**

("Success, Stability Account for Rhys created with 0 SDR deposited.", true) (36.261s)

#### OUTPUT LOG

```
> create_Account_User2("Rhys")
()
> create_Trove()
('Success, Trove for Rhys created with 0 ICP deposited and 0 SDR withdrawn.')
> create_Stability_Account()
('Success, Stability Account for Rhys created with 0 SDR deposited.', true)
```

First step for user is to create an account. Then create a trove to deposit ICP tokens and take out SDR tokens. The last part of the initialization is to create a stability account to deposit SDR into the stability pool.

The accounts are granted 10,000 ICP tokens to play around with.

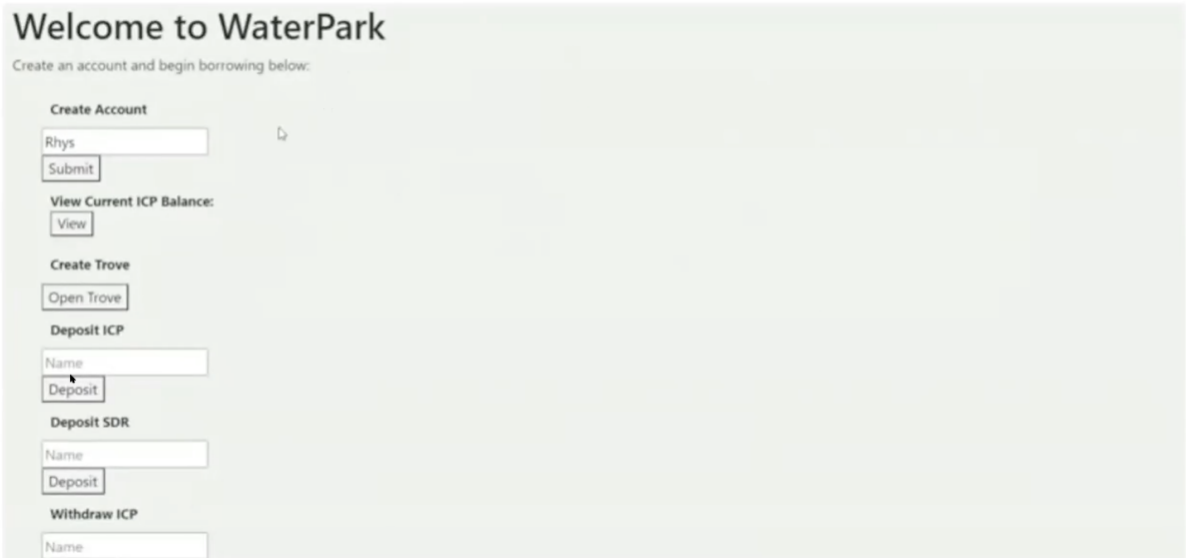
Oum demonstrates how the Candid UI works. For more details you can watch the Zoom recording from 2:30 – 11:30.

### **Prof. Yulin commentary:**

Difference between Internet Computer and Ethereum – on Ethereum you have to send some transactions to trigger the liquidation. On the Internet Computer it will be automatically running, so whenever the collateral ratio falls under the threshold value of 110% it will liquidate the asset automatically without anyone triggering the liquidation.

### **Rhys presented:**

Demonstration from 11:43 – 15:10



The screenshot displays the 'Welcome to WaterPark' interface. At the top, it says 'Welcome to WaterPark' and 'Create an account and begin borrowing below:'. Below this, there are several sections with input fields and buttons:

- Create Account:** A text input field containing 'Rhys' and a 'Submit' button.
- View Current ICP Balance:** A 'View' button.
- Create Trove:** An 'Open Trove' button.
- Deposit ICP:** A text input field containing 'Name' and a 'Deposit' button.
- Deposit SDR:** A text input field containing 'Name' and a 'Deposit' button.
- Withdraw ICP:** A text input field containing 'Name'.

```

graph TD
    OP[Orthogonal Persistence] --> PLSP[Programming Language Support for the Concept]
    OP --> Crit[Criticism]
    OP --> SD[Solution for Data Storage]
    PLSP --> JS((Java support))
    JS --> PJL((PJama Library))
    Crit --> Alt([The alternatives to orthogonal persistence include other forms of persistence, where persistence is still orthogonal to type but where there are typically restrictions on what objects can reference what objects.])
    SD --> WP[WaterPark Application]
    SD --> FOP[First application of Orthogonal Persistence to a DeFi system]
    SD --> CGC([Comparing gas costs between a system with orthogonal persistence and one without it])
    WP --> IC[IC as an alternative to Ethereum]
  
```

Dylan &amp; Malika: ICy

[illegible]

Was unable to implement a transfer function.

Designed in a way such that you have to add funds to your account before you can start using it.

Canister ID: rrkah-fqaaa-aaaaa-aaaaq-cai

## Candid UI

Browse and test your API with our visual web interface.

aCycle\_balance: () → (float64)

CALLRANDOM

aICP\_balance: () → (float64)

CALLRANDOM

addFunds: (text, text, float64) → (text, bool)

Usercycles

1000

CALLRANDOM

borrow: (text, float64) → (text, bool)

textfloat64

CALLRANDOM

### OUTPUT LOG

openAccount()  
("Success", true)

### Prof. Yulin Commentary Internet Computer vs Ethereum vs Cosmos vs Polkadot:

Ethereum is transferring from 1.0 to 2.0, so it now has proof of stake and 64 shards. Now Ethereum only has one chain. Ethereum 2.0 is proof of stake not proof of work anymore. The main chain becomes a beacon chain. Then there are 63 parallel chains. This is how Ethereum scales up. It gets more parallel chains.

Cosmos is similar it has a hub, like the beacon/main chain of Ethereum. There are many zones attached to the hub. This is how Cosmos scales up. It's very similar to Ethereum in that there's a main chain and then parallel chains.

Polkadot is also similar. It has a special name for the main chain called, the Relay Chain and there are many slots. There are auctions for the slots. You can create your own subchain, but there are a fixed number of slots for you to beat. Polkadot is very similar to Ethereum 2.0. It's

proof of stake. It has a fixed number of slots. Ethereum has 64 and Polkadot has a fixed number. Gavin Wood, the founder of Polkadot, was the cofounder of Ethereum with Vitalik.

Ethereum, Cosmos, and Polkadot are all of similar architecture. For example, you have first generation Bitcoin and then Litecoin copied and pasted the Bitcoin code and changed some parameters, such as increase the total supply from 21 million to 84 million and reduce the block time. Dogecoin is also first generation Bitcoin.

Cosmos and Polkadot fall into the second generation of blockchain.

DFINITY belongs to the third-generation with scalability throughput. The architecture is different from the second and first generations. Scalability is infinite so you can get a subnet whenever there is more demand.

Generation 1 is a simple transaction network.

Generation 2 you can add smart contracts and DeFi.

The difference between generation two and three is can you deploy large scale applications with millions of users. It's not possible on Ethereum 2.0 or Polkadot. You have to have some off-chain solution. With DFINITY it's possible. DFINITY is a bit more centralized in the sense that, in Ethereum everyone can be a node provider. You don't need permission from the network. You can download the code, install the client, and run as a node provider.

For DFINITY or the Internet Computer, you have to have a professional data center and apply as a node provider. You have to apply through the DFINITY foundation. By the end of next year or early next year, there's a network neural system where you can apply to be a node provider by submitting your application to the ns and there's on-chain voting. Then you can be a node provider. In that sense, it's not as permissionless as Ethereum. The benefit is that only the professional data centers can join which faster and more efficient with high throughput and bandwidth and low latency.

**Prof. Nayak:**

DFINITY is a permission blockchain. There's a lot of hype in the permissionless setting. Academics are more plentiful in permissionless work compared to the permission blockchains. That is why there are currently more articles, videos, and public material about something like Ethereum versus the others.

**Prof. Yulin:**

For DFINITY there's no standard yet to create ERC-20, but you can create your own.

Ethereum is very decentralized in that it has thousands of nodes distributed all over the world, but 70% of these nodes are hosted on AWS. In that sense, it's centralized. DFINITY is planning to host some Ethereum and Bitcoin nodes. The Internet Computer is creating a new internet protocol. If AWS, Azure, or Google Cloud can host blockchain node, then DFINITY can do that to.

**Prof. Luyao:**

Next week is the symposium.

**Symposium Timeline:**

Time	Discussion Topic	Presenter/Facilitator
10:00-10:05	Opening Remarks	Luyao Zhang, Xinyu Tian and Zesen Zhuang
10:05-10:20	ICy Presentation	Derrick Adam, Dylan Paul Malika Rawal
10:20-10:35	WaterparkPresentation	Derrick Adam, Urjit Banerjee Oum Lahade
10:35-10:55	Panel Discussions	Yulin Liu, Kartik Nayak, Fan Zhang, Luyao Zhang
10:55-11:00	Concluding	Xinyu Tian and Zesen Zhuang