Liquid Staking Module
Workshop

@

Gateway to Cosmos 2023, Prague

Meet the speakers

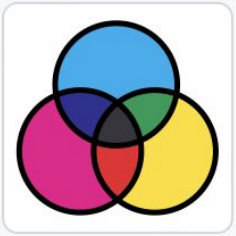
Max Kupriianov
maksim@persistence.one

Jeroen Develter
jeroen@persistence.one

Agenda

1. LSM origins and code layout, future development
2. New transaction and queries overview
3. How delegations are managed after tokenization
4. What is minted after tokenization
5. Difference between a NFT record and share token
6. Transferring liquid shares and rewards to another person
7. **Demo time!**
8. Q/A

LSM Origins



iqlusion

Accelerating cryptofinance: proof-of-stake validator company operating Cosmos Hub and Terra

28 followers · Silicon Valley, CA · https://www.iqlusion.io · hello@iqlusion.io · Verified

Overview · Repositories 71 · Projects · Packages · People 2

liquidity-staking-module Public

Go · 66 · 17 · 8 · 4 · Updated 2 hours ago

LSM’s actual layout

x/auth
x/authz
x/bank
x/capability
x/consensus
x/crisis
x/distribution
x/staking
x/evidence
x/feegrant
x/genutil
x/gov
x/group
…

Forked
LSM-enabled parts:

x/distribution
x/staking

x/stake added transactions

$ persistenceCore tx staking -h

...

- **tokenize**-share
- **redeem**-tokens
- transfer-**tokenize-share**-record
- validator-bond

x/stake added queries

$ persistenceCore q staking -h

...

- **tokenize-share**-record-by-id
- **tokenize-share**-record-by-denom
- **tokenize-share**-records-owned
- all-**tokenize-share**-records
- last-**tokenize-share**-record-id
- total-**tokenize-share**-assets

x/distribution added transactions

$ persistenceCore tx distribution -h

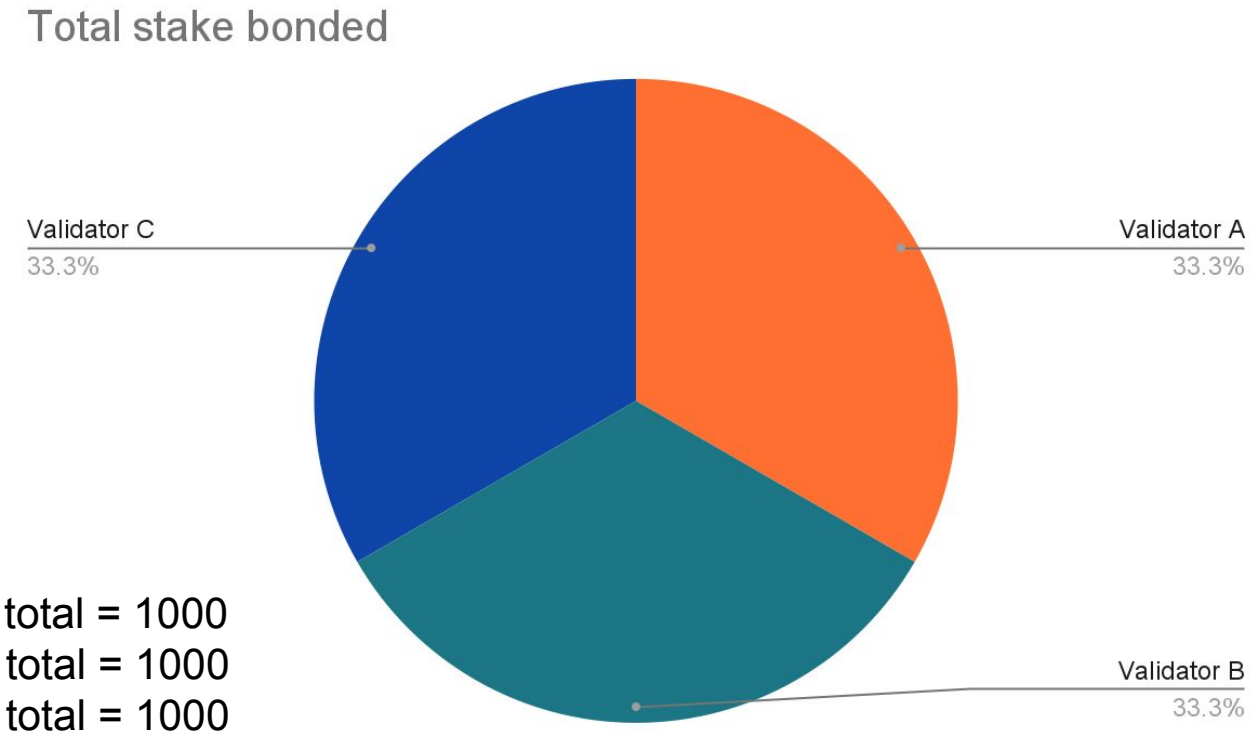
- withdraw-rewards
- withdraw-all-rewards
- set-withdraw-addr
- fund-community-pool
- withdraw-**tokenize-share**-rewards
- withdraw-all-**tokenize-share**-rewards

x/distribution added queries

$ persistenceCore q distribution -h

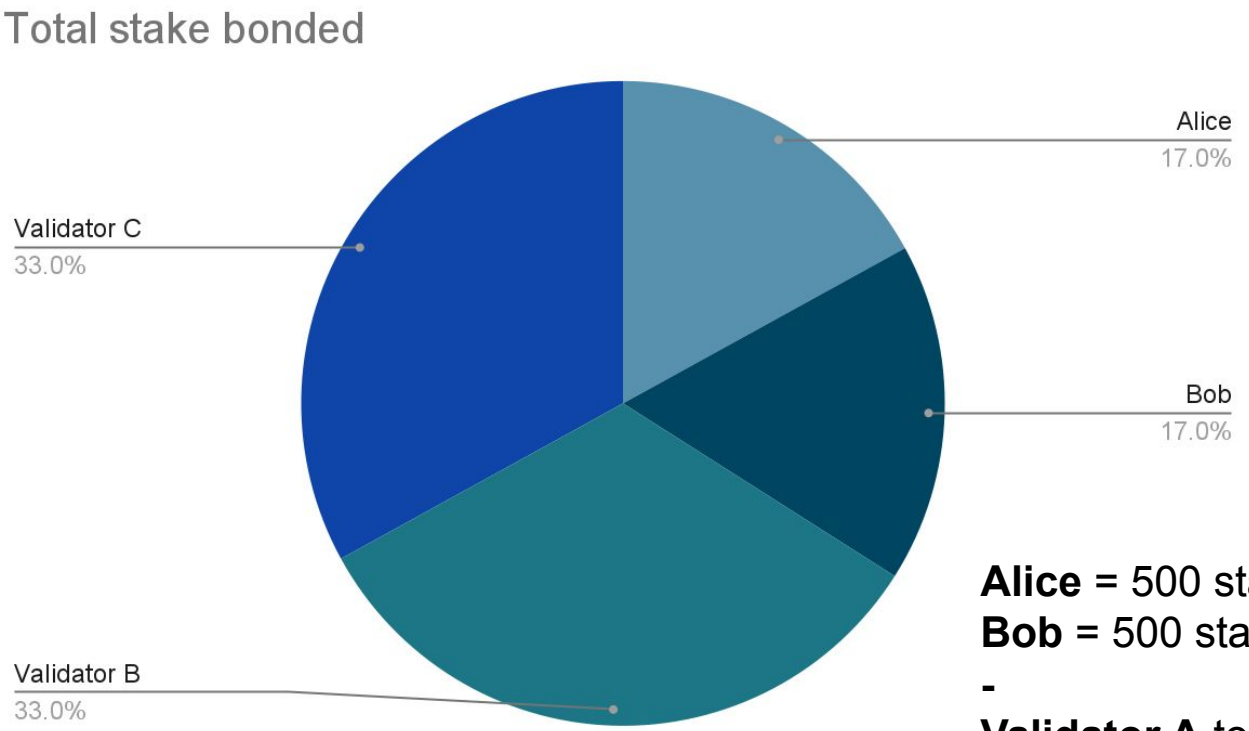
- params
- validator-outstanding-rewards
- commission
- slashes
- rewards
- community-pool
- **tokenize-share**-record-rewards

Stake Tokenization: A Primer



Validator A total = 1000
Validator B total = 1000
Validator C total = 1000

Stake Tokenization: A Primer



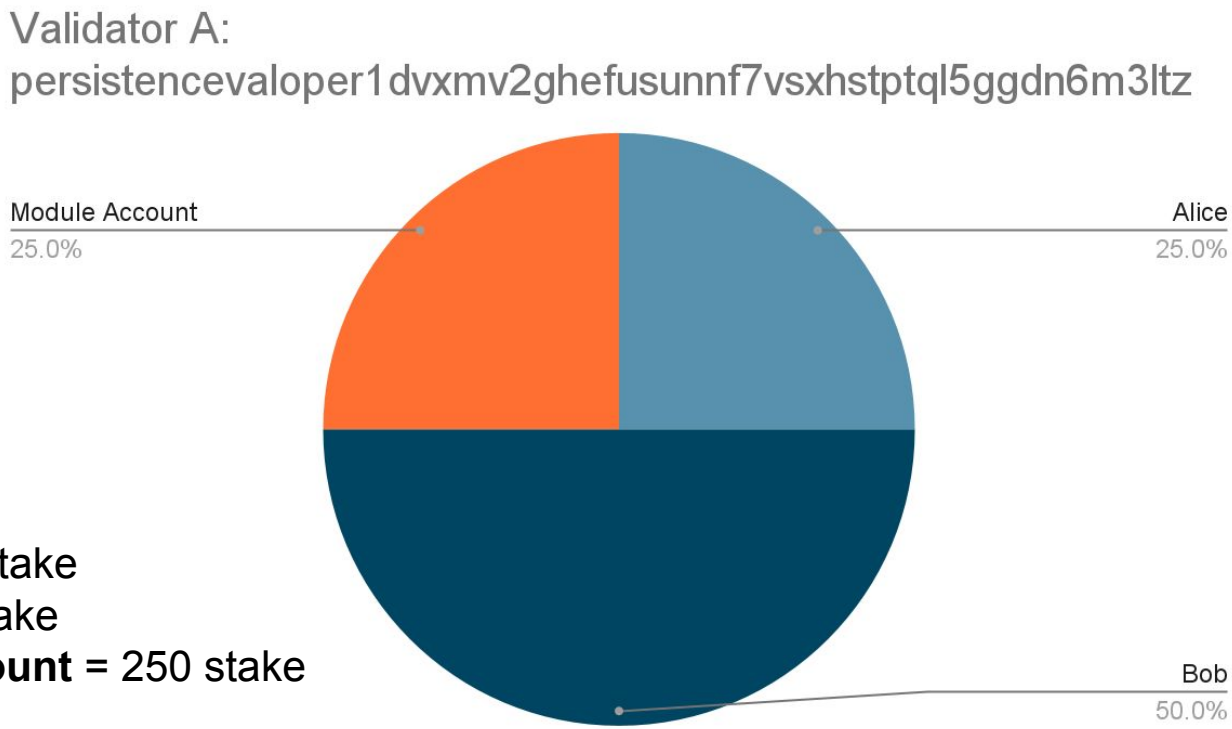
Alice = 500 stake
Bob = 500 stake
-
Validator A total = 1000

Stake Tokenization: A Primer

$ persistenceCore tx staking **tokenize-share**

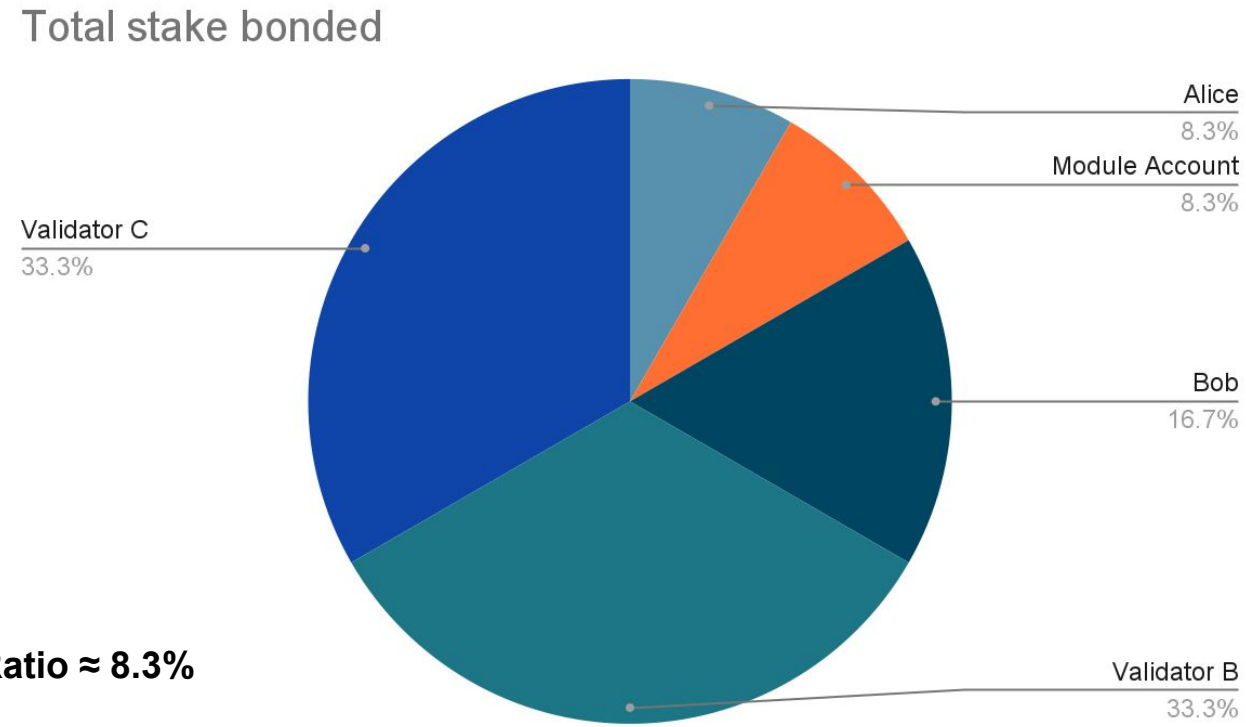
- [validator-addr]
- [amount]
- [rewardOwner]

Stake Tokenization: A Primer



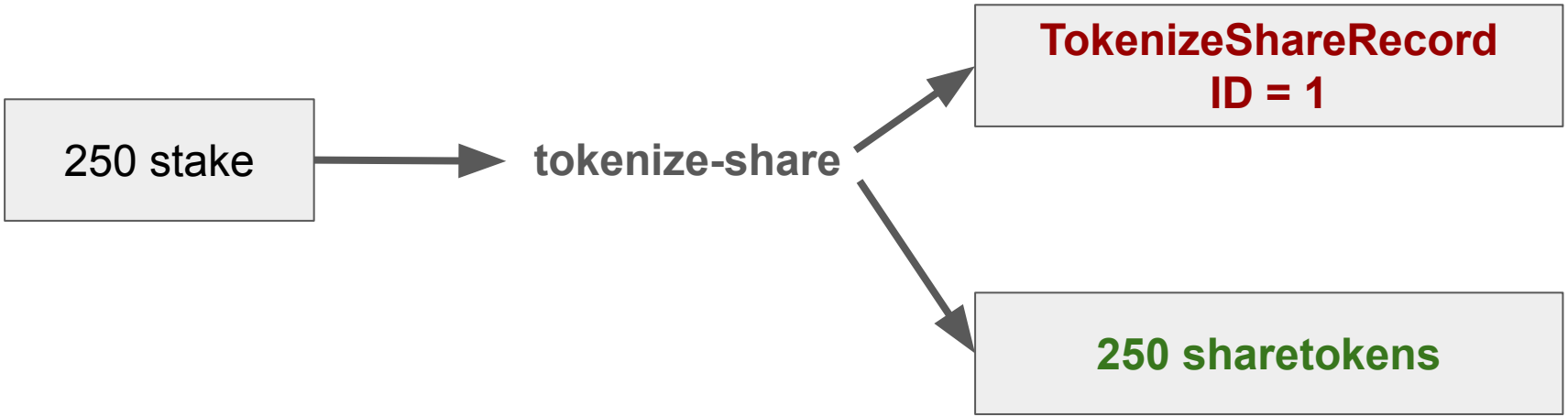
Alice = 250 stake
Bob = 500 stake
Module Account = 250 stake
-
Validator A total = 1000

Stake Tokenization: A Primer



Liquid Stake Ratio ≈ 8.3%

Stake Tokenization: Record and Share Token



Stake Tokenization: Records

TokenizeShareRecord
ID = 1

1. Kinda NFT
2. Accumulates Rewards
3. Transferrable!

Stake Tokenization: Records

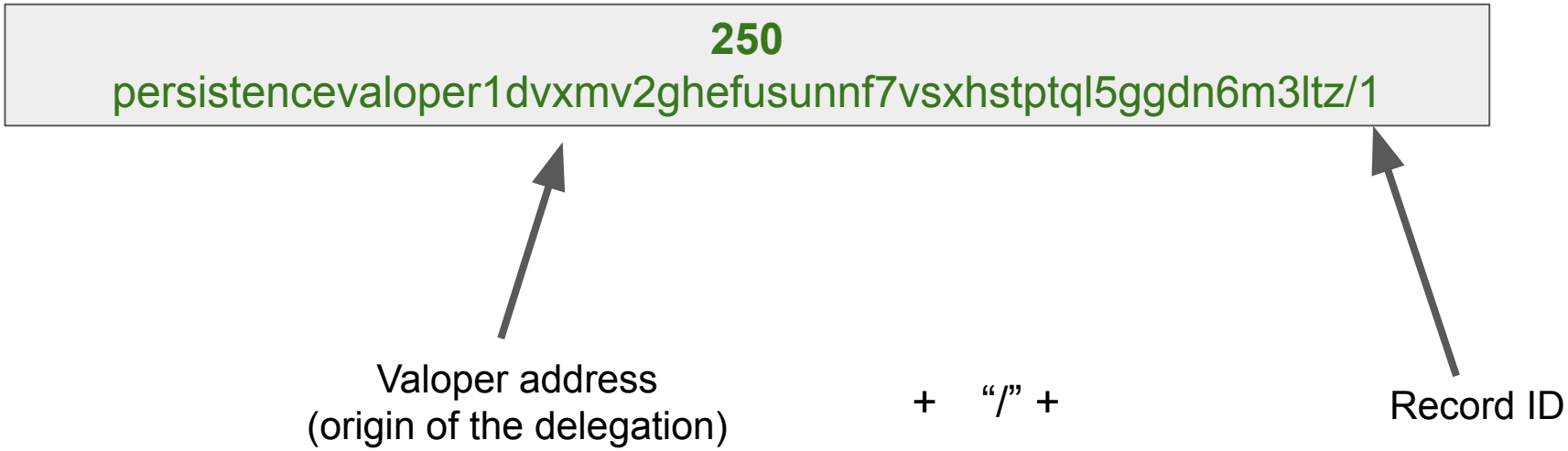
TokenizeShareRecord
ID = 1

1. Kinda NFT
2. Accumulates Rewards
3. Transferrable!

Available Actions:

- **transfer**-tokenize-share-record
- withdraw-tokenize-share-**rewards**
- tokenize-share-**record-by-id**
- tokenize-share-**record-by-denom**
- tokenize-share-**records-owned**
- **all**-tokenize-share-records
- **last**-tokenize-share-record-id

Stake Tokenization: Share Tokens



Stake Tokenization: Share Tokens

250
persistencevaloper1dvxmv2ghefusunnf7vsxhstptql5ggdn6m3ltz/1

1. Fungible
2. Managed by x/bank
3. Can see in the wallet
4. **MsgSend** compatible!

Stake Tokenization: Share Tokens

250
persistencevaloper1dvxmv2ghefusunnf7vsxhstptql5ggdn6m3ltz/1

Available Actions:

- tx staking **redeem**-tokens
- tx bank **send**

Stake Tokenization: Transferring Stake

TokenizeShareRecord
ID = 1

250 sharetokens

MsgTransferTokenizeShareRecord

- **transfer**-tokenize-share-record

+

MsgSend

- tx bank **send**

Demo time!

Getting it ready

https://github.com/persistenceOne/lsm-workshop-gateway23

1. Setting up local client (**README.md**)
2. Copy-pasteable steps (**DEMO.md**)

Explorer

https://lsm-devnet.core-1.dev

Step 1: Check your delegations

$ persistenceCore q staking delegations YOUR_ADDRES

```
delegation_responses:
- balance:
    amount: "10000000"
    denom: stake
  delegation:
    delegator_address: persistence1mn2d9z62l9zqaz3gtz7hrfwg50sclrr7agrkmm
    shares: "10000000.000000000000000000"
    validator_address: persistencevaloper1dvxmv2ghefusunnf7vsxhstptql5ggdn6m3ltz
```

Step 2: Tokenize your delegation

$ persistenceCore tx staking tokenize-share

VAL_ADDRESS **10000000stake** YOUR_ADDRES

Checking Tx status:
$ persistenceCore q tx TX_ID

Step 3: Check bank balance

$ persistenceCore q bank balances YOUR_ADDRESS

Shall have stake tokens and share tokens with a long denom!

Step 4: Check all delegators progress of tokenizing

$ persistenceCore q staking total-tokenize-share-assets

Step 5: Check all records owned

$ persistenceCore q staking tokenize-share-records-owned YOUR_ADDRES

Step 6: Check record’s accumulated rewards

$ persistenceCore q distribution tokenize-share-record-rewards RECORD_ID

Record ID must be from your tokenized position, see the list from previous step!

Step 7: Transferring half of the shares to Max

$ persistenceCore tx bank send YOUR_ADDRESS

MAX_ADDRESS **5000000**VAL_ADDR/YOUR_RECORD_ID

Use RECORD_ID from your own denom!

Example: persistencevaloper1dvxmv2ghefusunnf7vsxhstptql5ggdn6m3ltz**/5**

Checking Tx status:
$ persistenceCore q tx TX_ID

Step 8: Check Max’ delegations

$ persistenceCore q staking delegations MAX_ADDRESS

Amount of directly delegated amount should grow.

Step 9: Transferring the record to Max

$ persistenceCore tx staking transfer-tokenize-share-record

RECORD_ID MAX_ADDRESS

Use RECORD_ID from your own list!

Example: **5**

Checking Tx status:
$ persistenceCore q tx TX_ID

Step 10: Redeem the remaining half of shares

$ persistenceCore tx staking redeem-tokens

5000000VAL_ADDR/YOUR_RECORD_ID

Use RECORD_ID from your own list!

Example: **5**

Checking Tx status:
$ persistenceCore q tx TX_ID

Last Step

$ persistenceCore q staking total-tokenize-share-assets

Should be zero!

$ persistenceCore q staking all-tokenize-share-records

Should be empty!

Q/A