Since people are actively discussing I would like to propose a simple and better solution which does not use SNARKs. SNARKs are in my view grossly overhyped at the moment.

The solution goes as follows:

1. Users submit transactions to the operator, using user indexes instead of user public keys to save space.
2. The operator combines signatures in a block B.
3. The operator submits the block back to every user from this block.
4. The user creates a BLS signature share and sends it back to the operator.
5. The operator waits a while, combines signatures into a multi-signature and

sends B, the multi-signature, and the list of users that signed to the smart contract.

1. The smart contract verifies the signature and saves the block into Ethereum log storage.
2. To enter you simply deposit money, which will post a corresponding log entry. You can exit by passing your coin to someone who wants to enter.
3. To find out how much money anyone has, just follow the on-chain history from the beginning of time.
4. If someone tries to withdraw more than this person has, simply assume that the transaction size is the maximum of what the person has and the transaction value.

Note that this is totally on-chain and no exit required at all. Also there could be any number of operators, concurrently posting.

If you, say, have a billion of indices, (32 byte indices) you can have the price of the index go to infinity Bancor style, and if you release an index, you get your money back.