Cryptonomica Corporation On The Blockchain

Dividend Distribution and Bonds

Security Tokens - The Multi-Trillion Use Case

- Regular stocks and shares have no functionality
- Security tokens are digital programmable stocks/shares on the blockchain
- Smart contracts provide extra functionality: voting, governance, vesting, dividends
- No justification for paper shares when enhanced functionality exists

Corporation on the Blockchain

- Shares are conceptually similar to ERC20 tokens
- 1 share = 1 vote when voting for proposals
- Regular (e.g. monthly) dividend distribution in ETH
- How does it happen?

Paying Out Dividend - Simple?

- We cannot do a naive loop over all the shareholders. What if there are too many shareholders and we run out of gas?
- We have determined experimentally that we can Ethereum loops are bounded by around 300 iterations. More than that and we run out of gas.

Token Holders still need their dividends

Pull - A possible solution?

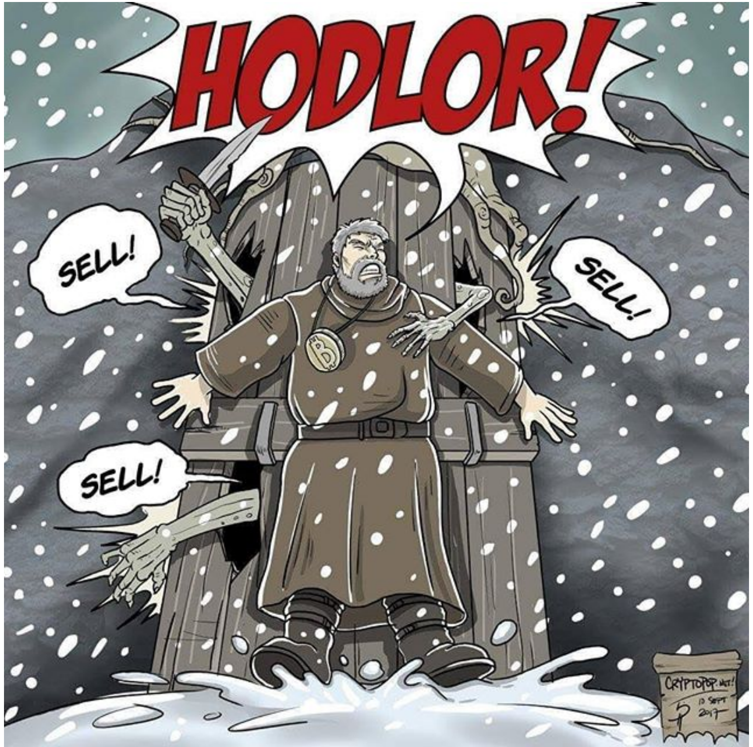
- Shareholders can withdraw their dividends instead of us distributing it.
- Too different from the traditional dividend distribution
- The distribution task should be on the corporation

Deferred Dividends

- Send a dividend to a shareholder by piggybacking on shareholder’s interactions with our contract (e.g. token transfer).
- The heavy gas burden for the full distribution is broken down into small chunks
- Keep an internal ledger of dividends paid

Deferred Dividends

What if some shareholders just HODL and never interact with our contract?



Dodge the throws

- A shareholder contract might throw (revert) in its default fallback payable function and break our whole distribution
- We need a way to handle these cases

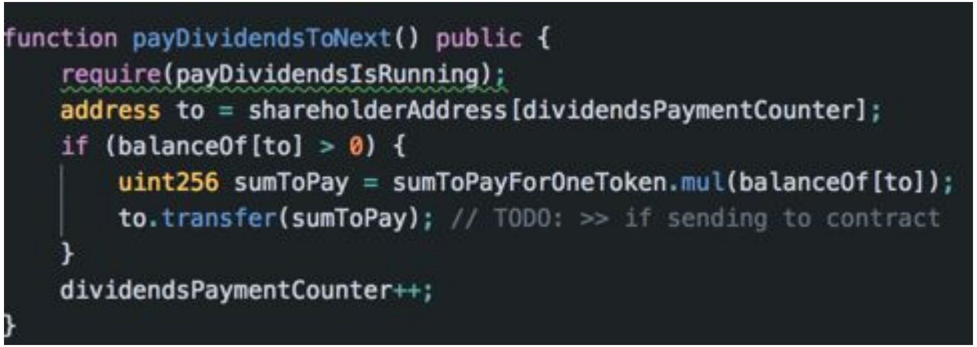
Public or Private?

Public Or Private?

- We make the dividend distribution method public
- Any shareholder or member of the public can distribute when it’s time
- Corporation owners cannot stall!

Our Solution

- Smart contract pays a dividend to one shareholder at a time
- Dividend payout function is publicly callable



```
function payDividendsToNext() public {
    require(payDividendsIsRunning);
    address to = shareholderAddress[dividendsPaymentCounter];
    if (balanceOf[to] > 0) {
        uint256 sumToPay = sumToPayForOneToken.mul(balanceOf[to]);
        to.transfer(sumToPay); // TODO: >> if sending to contract
    }
    dividendsPaymentCounter++;
}
```

- Backend application imports a shareholders registry from the contract and iterates over the list, calling the dividend payment function

Decentralized?

- Anyone can import the shareholders registry from the smart contract
- Anyone can call the dividend payment function
- No “magic” in our server code, so any entity can run it
- Dividend payout doesn’t depend on the contract owner

Bonds

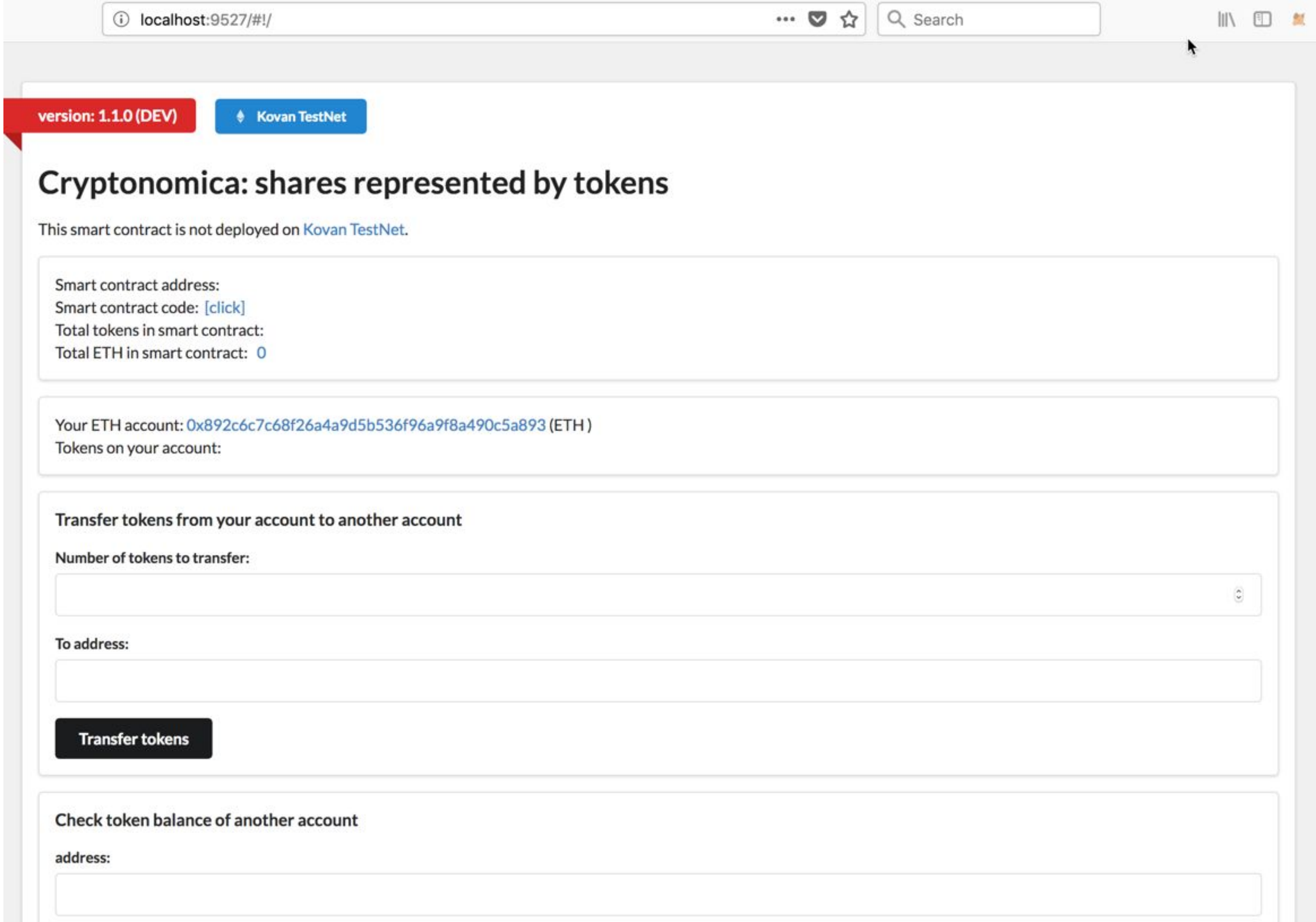
- Another incentive for shareholders.
- A shareholder can send his tokens to the smart contract address (i.e. burn his tokens) and receive his share of the smart contract’s ETH balance.
- Token holder is guaranteed to get his share of the corporation’s profits (ETH balance) at any point in time.
- Side effect: The corporation can also support the token price by sending ETH to the smart contract address.

Implementation

Github: https://github.com/Cryptonomica/dappathon-tlv

- Smart contracts for dividend distribution and bonds
- Web Interface that interacts with these contracts

Local Deployment: http://10.20.212.9:9527



Cryptonomica: shares represented by tokens

This smart contract is not deployed on Kovan TestNet.

Smart contract address:
Smart contract code: [click]
Total tokens in smart contract:
Total ETH in smart contract: 0

Your ETH account: 0x892c6c7c68f26a4a9d5b536f96a9f8a490c5a893 (ETH)
Tokens on your account:

Transfer tokens from your account to another account

Number of tokens to transfer:

To address:

Check token balance of another account

address: