Blockchain Democracy

A How To For Smart Contract Developers

Alberto Cuesta Cañada

Follow

The Startup

--

Listen

Share

When you vote for something, how do you know that anything will actually get done? How do you know that promises will be kept?

In this article, I'm going to offer a glimpse into how blockchain can change democracy. With a blockchain democratic process, promises become actions.

I'm not going to say that we can or should do away with politicians and install a technocracy, but I'm going to show how to run a voting system where the proposals get automatically enacted if the voting passes.

You could call it unstoppable democracy.

Conceptual Design

To start, please let me set the scene with two smart contract qualities:

- A smart contract is an immutable program. The rules that are coded in a smart contract cannot be changed. Once deployed, it cannot be stopped either.
- A smart contract can also trigger actions on other smart contracts. For example, a smart contract can trigger another one to release funds to a certain account, or to give someone permission to execute certain transactions.

Applying these concepts we can code a smart contract that runs a fair voting process, according to clear rules everyone can see. In that smart contract we can include a proposal, which is a call to a function in another smart contract.

The voting will happen, no matter what. If the voting passes, the proposal will be enacted, no matter what.

Ethereum and Democracy

Voting is one of the pillars of democracy, and it is as well one of the core building blocks in Ethereum.

It is thought that Vitalik Buterin broke from Bitcoin and proposed Ethereum to create a platform that would allow the implementation of democratic organizations, using the principles we described above.

These blockchain-based democratic organizations are known as Decentralized Autonomous Organizations, or DAOs for short. DAOs are directed by their stakeholders, with rules encoded in a blockchain smart contract, and no central control.

When you vote for something, how do you know that anything will actually get done? How do you know that promises will be kept?

Reading the wikipedia article for DAOs is very interesting. It reveals how early the idea for a DAO was conceived, and how powerful it was. Daniel Larimer (of BitShares, Steem and EOS.IO fame) proposed the concept and implemented it in BitShares as early as 2013. Then you also have The DAO (Ð) which before being hacked managed to attract 14% of all ether in circulation into its investment-focused organization.

However, the demise of "The DAO" didn't mean the demise of "the DAOs". Decentralized Autonomous Organizations are alive and well, now that the vulnerability that killed The DAO is well known and easily avoided.

Vlad Farcas and I started a toy DAO project, because we wanted to learn how to apply the pattern. Through coding a DAO I understood the possibilities for democratic processes in blockchain, and it blew my mind. That's why I'm writing this.

Introduction over, let's dig into the code. How can we do this?

Enacting Smart Contract Proposals

Consider the contract below:

This contract does some low-level magic, but it is not too hard to explain. On deployment it takes the address of another contract and a function call. On calling enact()

it executes the function call on the target contract.

Encoding the proposal can be done with web3j. In javascript the example below a Proposal

is deployed that will mint one ERC20

token when enacted.

web3.eth.abi.encodeFunctionCall

is a bit verbose, but really the only thing that it does is to pack a function signature and arguments in 32 bytes of data.

Its first parameter denotes the signature as a function, called mint

, non-payable, with one address

parameter called account

and another uint256

parameter called amount

. The second parameter assigns values to the parameters as the owner

account defined elsewhere, and the amount

of tokens to mint as 1.

There are much easier ways to have a contract call a function on another contract. So far it is hard to see why we would do things in this overcomplicated way.

Keep on reading, now we are going to make that proposal democratic. Let's see a contract that enacts proposals, but only after a successful vote.

One Token One Vote

You can find this contract in the HQ20 repository, please feel free to toy with it but don't use it as-is for a real-life purpose. To make it easy to understand we didn't close a number of vulnerabilities, for example against flash loan attacks.

In this OneTokenOneVote.sol:

- Votes are tokens from an ERC20 contract, chosen at the time of deployment.
- To vote means to transfer tokens to OneTokenOneVote

using vote()

- A proposal is passed if at any point OneTokenOneVote

holds a proportion higher than the threshold

of all tokens in circulation.

- Once a proposal passes it stays in the passed

state forever.

- Voters can cancel

their votes and retrieve their tokens at any time, but if they want the proposal to pass they should do it after it passed.

- Anyone can trigger the vote counting by calling validate()

. This will make the vote pass if the threshold

is met.

There are several ways of implementing a voting. There are safer ways to vote including requiring a quorum. OneTokenOneVote.sol

was the simplest example we could think of, but it is enough to show the principles of blockchain democracy.

When the voting is deployed, it accepts as a proposal a targetContract

and targetFunction

with its parameters encoded. If the voting passes, the enact()

function can be called by anyone to execute the proposal.

That means that the voting contract includes the action to be taken if the voting passes. There is no possibility to ignore the result of the voting. That is the little bit that was impossible before blockchain, think about that.

Smart Contract Democracy

There is one more twist that we can give to this concept of blockchain democracy. So far we know how to deploy a contract that will execute a voting process and then enact the result.

We could code a contract where all functions could only

be executed if they have been voted so. That is the spirit of a DAO, and it is easier than it sounds.

In the repository we have included a third contract, Democratic.sol, which I find really exciting to use. It allows any contract to hold votes on whether to execute any of its functions.

- Democratic.sol

is designed to be inherited by other contracts, allowing to mark any functions in them as executable only if they have passed a vote. You would do that by using the onlyProposal

modifier.

- Democratic.sol

allows anyone to put a proposal forward for voting. The propose()

function can be used by anyone, with the target function encoded with web3.eth.abi.encodeFunctionCall

.

- The voting tokens will be the same for all proposals, creating a community like with MKR tokens in MakerDAO. Democratic.sol

implements all votings as token-based, but that could easily be changed to account-based.

- The proposals are all stored in a proposals

register, and only proposals created by the same contract can ever execute functions marked as onlyProposal

.

If you think of it, you could use Democratic.sol

and OneTokenOneVote.sol

as the basis for a full democratic system. If you don't find that exciting I don't know what else to tell you.

Conclusion

Blockchain has the potential to change democratic processes by a degree never seen before in our lifetimes.

Using blockchain it is possible to implement unstoppable votings, that no one can avoid being enacted if they pass. As more and more of our world is accessible from the blockchain, the power of democracy will grow.

In this article we have shown how to implement voting procedures that can trigger smart contract executions, and have refined that to produce smart contracts whose functions can only

be executed by democratic processes.

None of this is new in the blockchain ecosystem, these concepts have been studied and implemented since Ethereum was conceived. However, in these contracts we think that we are providing with easy-to-use building blocks to take democracy one step further.