Peer to Peer Systems and Blockchains
Academic Year 2020/2021
Final Project

The attempt of a democratic election system

After a few centuries, the folks of Valadilène got tired of allowing a single person at the time to become mayor-in-trial. What if two or more people candidate themselves at the same time? Who goes first?

It was at that time that Sir Daniel Fortesque, one of the bravest frankish knights, proposed to vote not for one person at the time, but to let them compete and then elect one winner. Sir Daniel proposed a joust like in the old times, but no one remembers anymore how to use a lance, neither Sir Daniel himself, who is now a creative gardener.

The new election system is very similar to the mayor confirmation with a small change: the voting **doblon**, which is not useful anymore, is substituted with the **candidate's symbol**. Each candidate associates to themselves a personal public symbol that uniquely identifies them, they create many copies that are sent to the homes of the citizens of Valadilène together with their government plan. No wonder half of the plan is dedicated to how to organize the *Maxi-Delirium*. During the election day, each voter puts inside their envelope, together with their **sigil** and the **soul**, the **symbol** of the candidate they want to elect. The decision is mainly driven by how ~~cool and fun the next *Maxi-Deliriums* will be~~ important the plans to develop and improve Valadilène are.

Similar to the old mechanism, the envelopes are first collected and only afterwards opened and counted by the counting council members. Moreover, those who voted for the losing candidates get their soul back as a refund.

Since the neighboring cities got angry for continuously receiving kicked mayors, now the candidates who lose will stay in Valadilène, but they will serve as butlers and waiters during all the *Maxi-Deliriums* organized by the winning candidate.

However, to make the elections more spicy, the **election plan committee** formed by Angela, Pamela, and Phyllis, three paper saleswomen, will organize a few modifications so that every election is unique and remarkable!

Task

TASK01 - Generalize the smart contract of the final term. Instead of having a single candidate who wins or loses, the contract keeps a list of candidates identified by their symbol, i.e. the Ethereum address: the candidate who gets more soul (cryptocurrency) associated with the votes wins; in case of two, or more, candidates receive the same soul, the candidate with more votes wins; in case of an additional tie, no one wins.

The candidate list is passed to the smart contract at its creation (argument of the constructor). After declaring the winner, all the soul associated with the users who voted for the losing candidates receive their soul back; in case no one wins, all the soul go to the escrow account.

Moreover, the contract must implement one of the following extras proposed by the election plan committee:

<table>
<tr><th colspan="2"><b>Election plan committee</b>
<i>Spicy proposal N. 1001/BIS, Prot. 7829/BIS</i></th></tr>
<tr><td><i>Title</i></td><td>Detailed description</td></tr>
<tr><td><i>Join my side, I give you cookies</i>
from Phyllis</td><td>Each candidate must deposit some soul (> 0) as well. At the end, the soul deposited by the winning candidate is equally sent to all their electors, while the soul deposited by the losing candidate is sent to the winning candidate. In case of no winners, the escrow account takes all. Yes, it looks like buying votes, but where is fun otherwise?</td></tr>
<tr><td><i>Are they real money? Well yes, but actually no</i>
from Angela</td><td>Implement the soul as an ERC20 token. All the operations involving the soul are not performed with cryptocurrency, but instead with ERC20 token transfers. You can send to each voter a certain amount of tokens during a preliminary setup phase. Since ERC20 contracts are very popular and very similar to each other, you can import an ERC20 contract into your project, for example from this repository: https://github.com/OpenZeppelin/openzeppelin-contracts.</td></tr>
<tr><td><i>The Pammerellum</i>
from Pamela</td><td>The candidates either compete alone, or can group together in <i>coalitions</i> of k>1 <i>participants</i>. A voter can either vote for a coalition or a specific participant within the coalition. If the soul of a coalition is at least 1/3 of the total soul invested, all the participants win the election (count the coalition as “mayor in charge”). If two coalitions have more than 1/3 of the soul, the coalition with more soul wins. If no coalition reaches 1/3 of the soul, the election runs as no coalition ever existed (i.e. all single candidates and coalition participants compete with the soul they received).
<i>Note1</i>: a vote for a coalition is not a vote for a participant, and a vote for a participant is not a vote for their coalition.
<i>Note2</i>: if two coalitions have the same soul and >1/3 of the total soul, i.e. a draw case, no one wins and the escrow account rule applies.
<i>Note3</i>: voting for a coalition does not mean voting for the single participants. So if a coalition does not reach 1/3 of the soul, the soul is refunded (i.e. the coalition is considered as losing candidate, but the participants still individually compete).</td></tr>
</table>

TASK02 - Build a DApp, GUI or CLI, of the Valadilène voting system.
TASK03 - Prepare a demo that involves at least 3 candidates (or a mix of candidates and coalitions, depending on the extra implemented).

Submission

The project must be developed individually. The material to be submitted for the evaluation is:

1. The project implementing the Valadilène voting system: smart contracts and front end code.
2. A pdf report explaining:
    a. The structure of the project, and a user manual with the instructions to set it up and try it;
    b. The main decisions made during the implementation of the smart contracts and the front end;
    c. The instructions to execute the demo. The demo can be either automatic, or manual (in the latter case, provide a detailed description of the steps to perform).

The report and the code must be submitted electronically, through the Moodle. The project will be discussed a week after its submission. The discussion of the project consists in the presentation of a short demo, which can be run on the personal laptop, and in a general discussion of the choices made in the implementation of the system.

The oral examination (if required) will regard a review of the topics presented in the course. I recall that the oral examination is waived for the students who have passed both the Mid and Final Term.

Do not hesitate to contact us (laura.ricci@unipi.it, andrealisi.12lj@gmail.com) by e-mail, we will fix a meeting in the Teams room of the course.

“It would be much faster if I could handle the election plan all by myself”
Angela