Peer to Peer Systems and Blockchains
Final Project
Academic Year 2017/2018

Development of a Dapp for COBrA

1 Goal of the Project

The final project consists in an extension of the final term assigned in this academic year whose specification is available at the link `https://elearning.di.unipi.it/course/view.php?id=118`, on the Moodle page of the course. The final term required to implement and test (virtually via Remix) the smart contracts of the COBrA system, a system for fair COntent Trade on the BlockchAin. This project requires:

- to extend the COBrA system with additional functionalities.
- to write the frontend of the COBrA Dapp.
- to deploy and test the implemented system on a real blockchain.

2 Extending COBrA

The student is asked to extend COBrA in several directions. First, a customer feedback system must be defined enabing each customer using a content to rate it. Leaving feedback is voluntary and not mandatory.

Content feedback is expressed as a numerical tuple where each element corresponds to the score chosen by the customer to rate the content in a given category. It is up to the students to decide content categories on their own, as long as at least three categories are defined. Examples of categories could be: "appreciation of the content", representing how much the customer enjoyed the content and "price fairness", representing how fair the requested price is considered compared to the content. Customers are prompted automatically by the Dapp to leave feedback as soon as they have consumed the content (NOT when they are granted access to the content). The system should be designed so that only eligible customers are allowed to leave feedback on each content.

The following new functions must be added to COBrA:

- *GetMostRated(x,y)*: returns the content with highest rating for feedback category y (or highest average of all ratings if y is not specified).
- *GetMostRatedByGenre(x,y)*: returns the content with highest rating for feedback category y (or highest average of all ratings if y is not specified) with genre x.

- *GetMostRatedByAuthor(x,y)*: returns the content with highest rating for feedback category y (or highest average of all ratings if y is not specified) with author x.

Modify the COBrA payment system to allow content creators to define the price that will need to be paid by customers to access that content. Furthermore each content is rewarded by the *Catalog* each n views obtained (n chosen by the student) with a base value equal to the price required times the content average rating over the maximum rating.

OPTIONAL: the student can study strategies that malicious content publishers could adopt to leverage this payment scheme either to decrease other authors' revenues or to increase theirs. Discuss if such strategies are economically rational.

3 COBrA Dapp: implementation

The student is requested to develop a COBrA Dapp which must include two components: the blockchain-based backend and the web-based frontend. The backend is implemented through a set of smart contracts (in part already implemented in the final term assignment), while the frontend must be implemented from scratch. To interact with the blockchain it is suggested to use the web3j (`https://web3j.io/,https://web3j.io/articles/web3j%20article%20-%20Java%20Magazine%20JanuaryFebruary%202017.pdf`) library from JAVA to communicate with a *geth* Ethereum client.

The Dapp provides two different interfaces, one for the customers and one for the authors, the interfaces may share some functionalities, since an author can also be a customer.

The frontend asynchronously receives, through a set of callbacks, a set of events emitted by the smart contracts. Students are requested to define these set of callbacks, which must include at least those triggered by following notifications:

- *Feedback activation*: notifies that it is possible to submit feedback for a certain content. It is mandatory to wait for the content to be actually consumed before feedback can be left.
- *Access grant* : notifies to a customer that they have access to a content (either buyed from themselves or gifted by an other user), or that they have access to all contents in case of a premium subscription.
- *New content pubblication* notifies that a new content for a certain author or genre is available. Users need to be able to request the system to notify them on some authors or genres they specify they are interested in.

Do note that, since web applications are not the main topic of the course, the user interface of the frontend of the Dapp will not be taken into account for grading. As long as it is working, a basic GUI is all that is required. Still blinding colours, fixed 1024x720 windows and other examples of unusable UIs will not be appreciated.

4 COBrA deploy

The COBrA Dapp should be implemented on a real blockchain, in particular, it must be deployed on the Ropsten Ethereum testnet. Students can obtain the needed test ether form a faucet as explained during the course. All deployed contracts are required to have a suicide operation to relieve the testnet after the experiments. Furthermore, students are advised to perform their trials on a local testnet started by them and only test the final project on Ropsten to ease the burden on the global testnet.

5 Project Submission Rules

The project must be developed individually. Students should submit

- the backend: the code of the Solidity smart contracts. Each smart contract must be commented in the report.
- the frontend code (developed in JAVA or in any other chosen language), as well as the instructions to compile, deploy and run the Dapp.
- a brief report including the commented smart contracts, the main project choices, a list of operations to test that the Dapp is working as intended.

Do note that the final project will be tested on localhost (Linux machine) with a running geth client connected to Ropsten. The report and the code must be submitted both electronically, through the Moodle, and at the reception desk of the Department of Computer Science.

The project will be discussed a week after its submission. The discussion consists in the presentation of a short demo and a general discussion of the mid term, of the final term and of of the choices made in the implementation of the project and of the developed code.

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

Do not hesitate to contact us by e-mail (laura.ricci@unipi.it) or during the question time, Thursday 15.00 PM-18.00 PM.