NotesExchange: A distributed application built with smart contracts

Xiana Carrera Alonso, Aldán Creo Mariño

Distributed Systems

Abstract

A distributed application (DApp) is a type of application run in more than one computer and in which communication takes place through a network. It is strongly tied to the blockchain and to smart contracts, self-executing contracts where the agreement terms are predefined and that run automatically when their conditions are met. This report aims to detail the development process of creating NotesExchange, a DApp that uses smart contracts to provide a classnotes marketplace and renting system between classmates.

1 The idea

NotesExchange is centered around the idea of a collaborative learning community, where students are able to improve their academic performance, and optimize their time management, by sharing class notes with their classmates, or even other students. NotesExchange provides a marketplace environment, where class notes that have already been taken can be published for their purchase by any other users; and where potential buyers can browse through these offers. Furthermore, there's also the option to delegate a future note taking process for a certain lecture to another student, effectively renting their services to ensure future access to the imparted knowledge.

An application such as the one described poses several implementation challenges, for which a distributed system and smart contracts are a great fit ([1]).

First, a distributed app allows for decentralization, meaning that the system is more resilient and less prone to interference from external sources such as education entities.

Secondly, smart contracts automate the aforementioned digital transactions, freeing the students from periodic, manual update checks with their note takers / renters. They also provide a secure environment where users are certain that the conditions stated in their contracts won't be violated.

Finally, the transparency and accountability component that distributed apps provide should not be overlooked. The recording processes that take place on the distributed ledger allow users to easily verify note authenticity and their previous transaction history.

2 Implementation

The development process involved several components. Some relevant mentions are:

Truffle An environment that allows for fast and easy initialization and deployment of DApps based upon the Ethereum blockchain.

Ganache A private Ethereum blockchain for local development, providing means to simulate a real deployment of the app's smart contracts, while providing test accounts that enable us to not spend real money or pay real gas fees. It is part of the Truffle suite. Both the console and GUI versions were used during development.

MetaMask A browser extension that we use to perform transactions on the blockchain. It can interact with our DApp.

Solidity Used to write the smart contract.

Mocha Used to perform testing, because of its simplicity and intuitive syntax.

Remix The Remix online IDE was used for smart contract testing purposes. However, the app itself was developed in Visual Studio Code.

React, NodeJS, Bootstrap The off-chain source code is written in TypeScript, grounded on NodeJS. React was chosen as the frontend framework, styled using Bootstrap.

The implementation process itself started with the setup of the previous tools. After deciding the main purpose of the app and the features it should implement for completeness, the details were discussed with the TA for the Distributed Systems course, Rodrigo Otoni. Following his approval and after making adjustments with his feedback, the research process began. We went through the Solidity documentation while at the same time reading examples and tutorials. We also learned how to deploy the smart contract and how to connect it to the user interface. At the same time, while the UI was not yet fully implemented, we made use of unit tests and of Remix to manually validate and verify our contract.

3 Challenges

The main challenge faced was the learning curve of the many new tools and technologies used. As we had no previous experience with blockchain technology, Solidity, Truffle, Ganache, etc., there was an un undeniable feeling of overwhelm during the initial steps. However, the documentation for distributed apps is certainly quite complete and well explained, and after a while, we started to gain more confidence, and noticed an improvement in our skills. Nonetheless, we put special emphasis in manual testing to enhance security.

Furthermore, the lack of standardization in the field made it more difficult to adapt to the needed tools ([2]). For example, the differences in the syntax of Solidity between versions can be quite significant, and trying to get inspiration from resources written for older versions was not straightforward in most cases.

On top of that, our team had to communicate remotely, which add some friction to the process. Fortunately, the workload had previously been divided and each one of us had a clear view of what we were aiming for, so we didn't encounter too many problems on that regard. We used a shared repository in GitHub and communicated mainly through private chatting, and a shared Notion workspace where we kept a Kanban board of the tasks to be completed.

4 Conclusion

Although developing a DApp with smart contracts posed challenges, it was deemed a valuable and beneficial project for the skills and amount of knowledge gained in an unexplored field. The final version of NotesExchange meets our initial goals and expectations, and we feel prepared to face more challenging projects related the techologies used in the future. For example, features such a full satisfaction rating of the services provided by note takers, or the implementation of services fulfilled by multiple users, could all serve as examples of plausible research topics.

References

[1] Khan, S.N., Loukil, F., Ghedira-Guegan, C. et al. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Netw. Appl.* 14, 2901–2925 (2021). https://doi.org/10.1007/s12083-021-01127-0

[2] Tian, Z., Tian, J., Wang, Z. et al. International Journal of Intelligent Systems. *Landscape estimation of solidity version usage on Ethereum via version idenficiation* v. 37, issue 1, 450-477 (2021). https://doi.org/10.1002/int.22633