Decentralised Application Project

Introduction

A new breed of applications is being discussed across the world. These types of applications are not owned by anyone, canÕt be shut down, and cannot have downtime. Such new type of applications is named Decentralised Applications (DApps).

Decentralised apps could be either a mobile app or website. However, rather than interacting with a database and file system, they interact with smart contracts, which are snippets of code that could be stored and executed in a blockchain. The aim of this project is to let you design and implement a decentralised application. More details will be given in the later sections.

Decentralised Shopping Cart

One idea for you is to design and implement a decentralised shopping cart. Similar to a classical shopping cart application, the decentralised shopping cart supports operations such as adding items, removing items, calculating prices, searching items, purchasing etc. However, typical online shopping systems may contain extra charge from sellers. More than that, traditional online shopping applications suffer security issues such as DDoS, SQL injection and cross-site scripting attack.

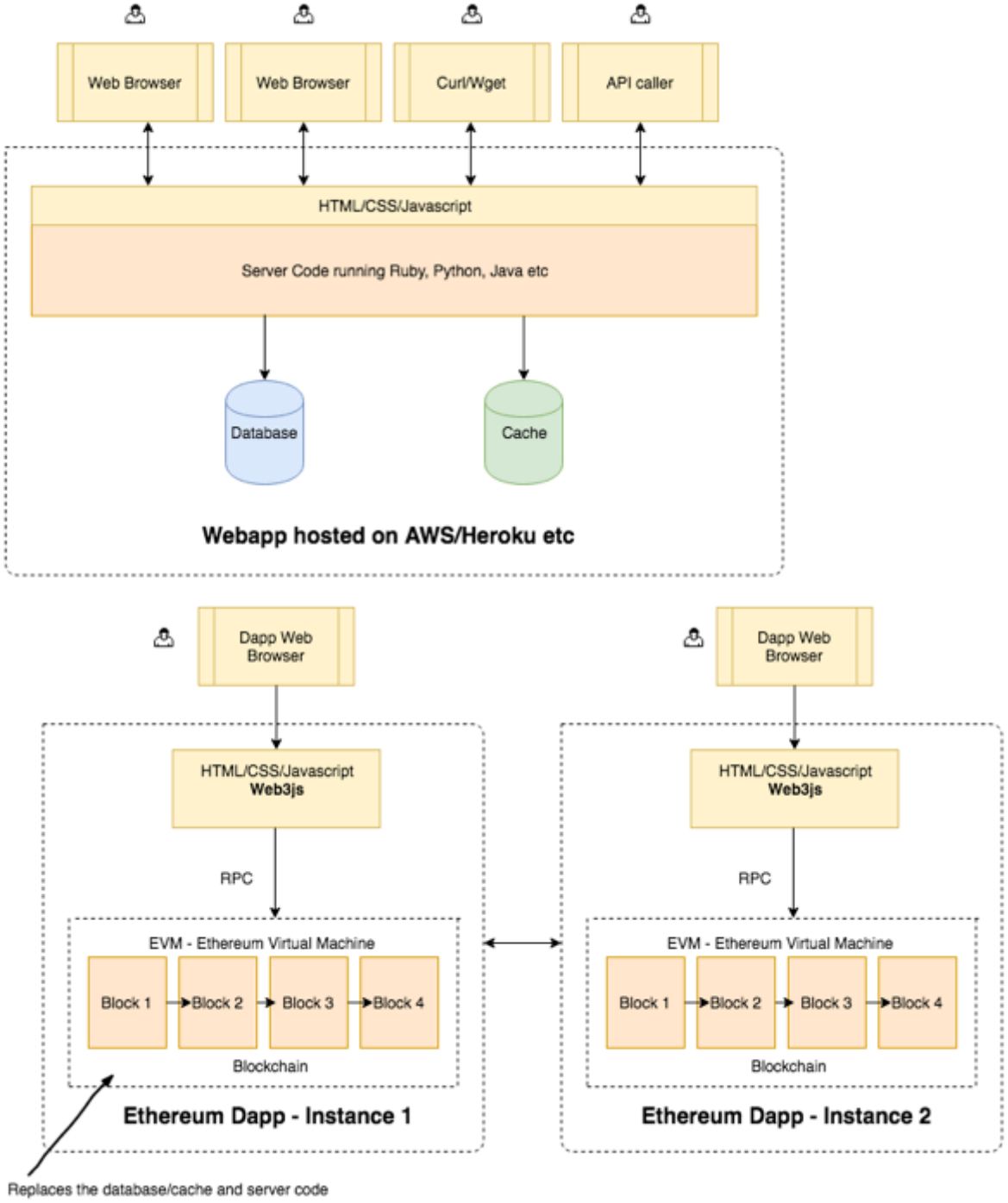
One solution is to make the application decentralised. In a decentralised shopping application, all trades happen directly between buyers and sellers with no middleman to take a cut from each sale. In terms of security, essential components have been distributed using the blockchain. Such property makes it practically impossible and expensive to attack.

Properties of DApps

According to the dapp white paper, for an application to be considered truly decentralised, it has to:

- Be completely open-source and operate autonomously with no entity in charge of the majority of its currency;
- Have any protocol changes that are designed to make some overall improvement approved by all its users;
- Cryptographically store all of its operational data and records in a public blockchain;
- Use a bitcoin or a currency that is native to its blockchain system so that it can be accessed for use and any future contributions to its value from miners; and
- Generate tokens, or currency, which follows a standard cryptographic algorithm.

One of the most significant features is to use a blockchain for storing data and records instead of a database. The following images show you the difference between the architectures of an app and a Dapp.



Images Source [Ethereum for web developers]

You may find more details about Decentralised Applications from here.

The Ethereum Blockchain

Although a DApp requires a blockchain to manage data, fortunately, we donÕt need to implement the whole blockchain infrastructure by ourselves. **Ethereum** is an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications.

Ethereum is not just a digital currency. It is a blockchain-based platform with many aspects. It features smart contracts, the Ethereum Virtual Machine (EVM) and it uses its currency called ether for peer-to-peer contracts.

In terms of bitcoins, Òsmart contractÓ is a phrase used to describe computer code that can facilitate the exchange of money, content, property, shares, or anything of value. However, in the Ethereum world, developers are allowed to program their own smart contracts, or Ôautonomous agentsÕ. A smart contract is now able to

- Manage agreements between users, say, if one buys insurance from the other
- Store information about an application, such as domain registration information or membership records.
- Provide utility to other contracts (similar to how a software library works)

One of the most widely used programming languages for writing smart contracts are ÒSolidityÓ.

Solidity is a contract-oriented, high-level language for writing smart contracts. It is now the primary language on Ethereum and it will be one of the most important languages to use in this project. You may find more details about the language here.

Tutorials for you to start with

Before starting this project, it is recommended to go through some of the decentralised application tutorials. You may find a lot of tutorials from GitHub. I have listed some tutorials here

- Your First Decentralized Application
- Full Stack Hello World Voting Ethereum Dapp Tutorial
- Building a smart contract using the command line

The following are 2 example projects that you may read through when you are building your own application.

- https://www.openbazaar.org/features/
- https://github.com/brakmic/BlockchainStore

Your tasks

LetÕs back to the aim of this project. You and your team are asked to design and implement a decentralised shopping application. The application should allow customers to buy and sell goods. Some basic functional requirements include:

- post a new item
- update the price, quantity, description of an item
- search items by keywords
- add an item to the shopping cart
- purchase

You should also think carefully about non-functional requirements such as user sign up, log in, logout etc. Please carefully decide which data need to be stored in a blockchain. For big files such as images, you may either consider storing them into a decentralised file system called IPFS or just store them in cloud storages like AWS S3.

Please refer to the next section for possible issues. If you find any other issues, please let us know and we will put them in as well.

Final Demo

Make sure you have implemented all requirements specified in the proposal before the final demo. In your demo, you need to show us a whole picture of your DAppÕs structure. Further more, you should also explain how the tokens are updated in the blockchain. For example, when a seller posts an item, a new token should be generated. In this case, you may print out the new token in text format and further show how this token refers to the item created. Similarly, for any other use cases, show us the project is constructed on the ethereum blockchain.

Common Questions

1. Is it required to download the whole ethereum blockchain to complete this project?

No. Theoretically, any ethereum node obtains a copy of the whole ethereum blockchain which is now 1TB in order to support running a DApp. However, the blockchain is giant and most people wonÕt be happy to waste their disk space keep such ÒuselessÓ data. Hence, we have got some options here. One of them is to install a browser plugin called MetaMask. It is an app that allows you to communicate with the ÒTestNetÓ without downloading the whole blockchain to your computer, but note that the transaction is a bit slower than using a local blockchain.

2. Do I need to buy ethereum coins to complete this project?

No. Although all decentralised application requires cryptocurrency to execute each transaction, you can use the ÒTestNetÓ for free. Again, please install the MetaMaskplugin and try to request some free ÒfakeÓ ethereum coins for your test. Remember to switch to any of the Test Network rather than doing your work on the *Main Ethereum Network*.

3. What are the programming languages required to complete the whole project?

Similar to a classical application, you need some knowledge of HTML, CSS and javascript. More than that, you may also need some knowledge on nodejs and npm because the truffle environment is one of the most popular frameworks for web Dapps, and you need some nodejs skills to run it.
You may also need the Web3.js API to communicate with your smart contract.

Related Resources

1. Decentralized Applications
2. What are Decentralized Applications
3. Ethereum for web developers