An Introduction to Blockchain Development

Joseph Weller

Midwest Instruction and Computing Symposium, Milwaukee School of Engineering, Milwaukee, WI 53202 USA

wellerj@msoe.edu

3 / 11 / 2022

Abstract

This report will provide high-level insight as to what blockchain development is and will evaluate the benefits of learning more about it. The report will begin by briefly providing an overview of what blockchain networks are, how they work, some of their current issues, and the advantages that they provide to modern software developers. It then will narrow in on how developers can write code that is capable of being stored and executed on blockchains. Finally, there will be a discussion on the opportunities that blockchain and decentralized autonomous organizations open up for entrepreneurially-minded engineers, and how they can best take advantage of these opportunities.

Introduction

In today's interconnected world, there are few topics in computer science as far-reaching and rapidly unfolding as the use of blockchain. However, as of recently, the public perception of this technology has been less than positive. From the ridiculousness of NFT digital art marketplaces to the detrimental effects of mining bitcoin, discussing the topic of blockchain has almost become taboo. In this report, I will outline the value of using blockchain technology in modern software development and show the importance of why blockchain will be a tool in most future developers' toolboxes.

Over the course of the past year and a half, I have been fortunate enough to have multiple experiences to learn about what blockchain and its surrounding topics are as well as where they have been shown to be practical. These experiences have covered a wide range, including everything from taking online blockchain development courses, to researching and exploring where blockchain can be integrated at a Fortune 500 company, to creating patent applications for two innovations that use blockchain technology, to developing coursework for an undergraduate blockchain development curriculum for my university. With the observations I have collected throughout these different journeys I hope to be able to provide a perspective on blockchain that is less commonly discussed.

Why Blockchain

As of recently, blockchain and some of its initial use cases, as well as its initial flaws, have been picking up attention. This can quickly be observed by creating a Twitter account and nearly instantly being subjected to the daily battles between users who are climate activists versus the users with NFT monkey profile pictures. One side might assert that the mining of all cryptocurrencies consumes more energy per year than a mid-sized country, while the other side would fight back by claiming that every winter the US consumes just as much energy on Christmas lights. While these fierce debates are currently immensely important, as well as slightly entertaining, the key topics that reside at the core of almost all of them are nothing more than short-term issues.

Calling these important topics of debate short-term issues is no small claim to make. By doing surface-level research into the topic of blockchain, some might say that my glasses are overly rose-tinted and that belittling these problems, which appear to be foundational to blockchains and their ecosystems, is irresponsible. I understand and acknowledge those who hold this opinion, as my judgment was very similar a year ago when I was began approaching the topic of blockchain with caution. However, after methodically performing research on the most common problems brought up when discussing blockchain, I have discovered supporting evidence as to why the majority of them are temporary.

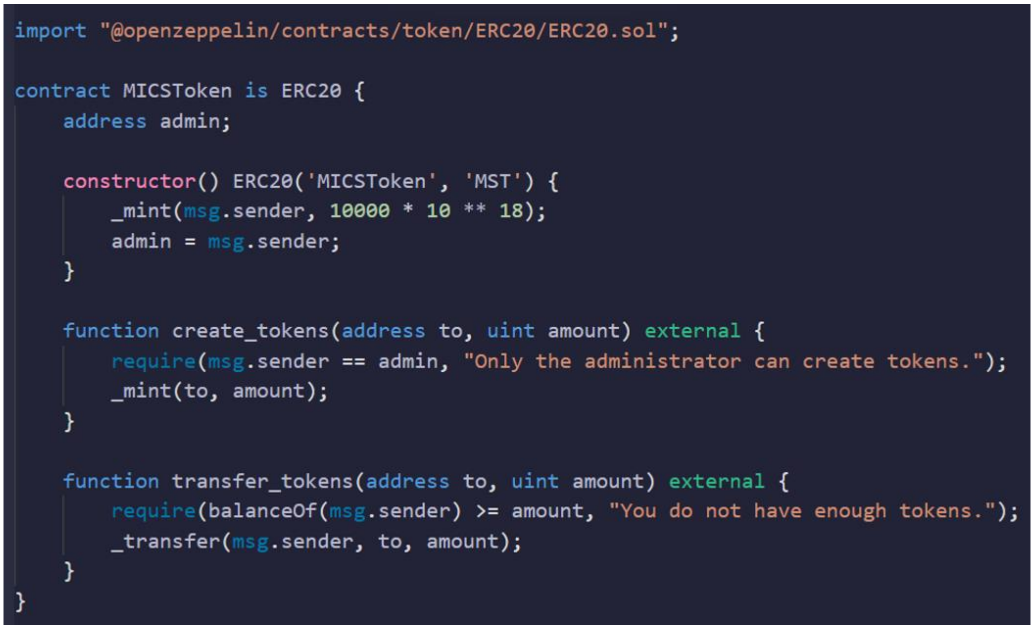
Before dissecting some of the modern dilemmas of blockchain, it is essential to understand what exactly the technology is at a high level. To dive into this topic, I will make use of some of the material which was created as part of my undergraduate curriculum. A good starting point is to define what exactly a blockchain is: a continuously growing list of records, called blocks, that are linked and secured through cryptography. It can essentially be viewed as an immutable append-only linked list. By moving up one layer from blockchain as a data structure, public blockchain networks can be found. Similar to the application layer protocols that the internet was built on, these public networks have no central authority but instead are governed by numerous authorities who maintain control of the network, meaning that they are decentralized. In addition to being decentralized, these networks are also distributed, meaning that numerous parties hold a copy of the underlying blockchain data structure. While there are endless details and complexities involved in creating and securing these networks, these fundamental pieces of insights should be enough to form a foundation for understanding the essentials of blockchain development.

Now that some context has been provided for what blockchains are at a high level, their modern problems can be discussed with a better understanding. One of the most commonly brought up issues when discussing blockchain networks, more specifically cryptocurrencies, is that they are unusable for practical applications, such as being a currency, due to their excessive fees and energy consumption rates. While this argument holds true for the Bitcoin and Ethereum networks as well as a few others, solutions to this problem have been implemented on other networks through the means of alternative consensus mechanisms. Another often mentioned pitfall of blockchain networks is their lack of interoperability, and therefore lack of mass adoption. In the long term, this issue will fade as the blockchain networks with the first-rate fundamentals will outperform their competition, and the networks which are built on inferior fundamentals, which is the majority of them, will be left behind. Other voiced concerns with blockchain include the fraudulence involved in the NFT digital art marketplaces, the privacy risks involved with storing records publicly on blockchain networks, and so on. While I agree that these issues need to be carefully and thoughtfully considered, they are not exclusively bound to the domain of blockchain, meaning that the technology should not be held liable. In time, increased government regulations will reduce the turmoil within the NFT markets, developers will refine ideas such that they do not jeopardize their data's security, and as more and more problems are resolved, the shortcoming of blockchain's lack of commercial adoption will also be resolved.

Smart Contracts and Decentralized Applications

Now that the high-level fundamentals and rationale for using blockchain have been covered, the next question to answer would be how modern developers can work with this technology. The answer to this question primarily comes from two interconnected concepts which are called smart contracts and decentralized applications, dApps for short. These smart contracts can be reduced to being code that can be stored and executed on a blockchain network that has behavior that is similar to a restful API. Depending on which blockchain network a developer prefers to work with, these smart contracts are able to be written in either a network's proprietary programming language, such as with Solidity and the Ethereum network, or by using a software development kit for an existing programming language, such as the Rust SDK for the Solana network or the Python SDK for the Algorand network.

The code in these smart contracts being stored on the public blockchain network has a couple of unique properties that are imperative to the technology's success. The first property is that once the code written in the smart contract is deployed to the blockchain, it's immutably stored there. This means once code has been developed and audited, it can be permanently stored on the blockchain. This opens up the door for a completely trustless level of transparency between the developers and the users of an application that wasn't attainable before blockchain. Another property that is associated with smart contracts is their indirect promotion of clean and efficient code due to a concept called gas. This property is best noticed through viewing figure 1, which is a smart contract written in the Solidity programing language that creates a fungible token that can be transacted by anyone on the Ethereum network once it has been deployed. While this code technically uses an imported interface that reduces its footprint and allows it to be interoperable with other Ethereum smart contracts, it should be recognized that it produces a sizeable amount of functionality with only 16 lines of easy-to-understand code. Since the code in smart contracts is executed by every node on the Ethereum network, it needs to be as optimized as possible to reduce the amount of gas that it expends.



```
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract MICSToken is ERC20 {
    address admin;

    constructor() ERC20('MICSToken', 'MST') {
        _mint(msg.sender, 10000 * 10 ** 18);
        admin = msg.sender;
    }

    function create_tokens(address to, uint amount) external {
        require(msg.sender == admin, "Only the administrator can create tokens.");
        _mint(to, amount);
    }

    function transfer_tokens(address to, uint amount) external {
        require(balanceOf(msg.sender) >= amount, "You do not have enough tokens.");
        _transfer(msg.sender, to, amount);
    }
}
```

Figure 1: A smart contract written in Solidity that creates a fungible token following the ERC20 standard.

After this smart contract has been created, audited, and deployed to the blockchain, the outside world needs to have some way to interact with it, this is where dApps are introduced. By combining a smart contract, which acts as a back end, and a web application, acting as a front end, a dApp is produced. For example, in figure 1, any web application that wants to interact with the smart contract can use a JavaScript library called Web3JS which connects to the contract through a hash that was generated when the smart contract was deployed. Once the web application is connected to the smart contract, any user who connects their wallet to the web application can call the functions in the smart contract which are marked with the external keyword. While one of the most common ways of interacting with smart contracts is these dApps, other mediums such as APIs can be used as well.

Decentralized Autonomous Organizations

After learning about what blockchain networks are at a high level, and observing how developers can directly interact with them, the next concept to grasp is decentralized autonomous organizations, or DAOs for short. These organizations are community-led and surround a project which has no central authority. The rules of DAOs are transparently laid out in smart contracts and are available for anyone to view. These decentralized communities create and vote upon proposals, such as technical upgrades and treasury allocations, that affect the future trajectory of the underlying project. The advantages of DAOs might not be completely clear at first glance, however, for an entrepreneurial-minded engineer, they bring to light a new opportunity.

When an entrepreneurial developer has a new idea for a project that creates value, they need to evaluate the risk of pursuing that idea. During this evaluation, they need to consider the possibility of their idea scaling and ensure that they are ready to manage the business side of the company in the event that it does scale. For most engineers, even the entrepreneurial type, this sounds like a headache at best and a nightmare at worst, however, DAOs introduce a potential solution to this problem. Instead of splitting their time between engineering the project and planning on how to operate the business side of the project, the developer could instead begin working on the project with the eventual goal of converting it into a DAO in mind.

As an example, one way an engineer could do this is by initially creating and developing a project, while outsourcing labor where needed, and eventually presenting the project to the public. Once the project is ready to be launched, a token directly tied to the use case of the developer’s project would be created and deployed to a blockchain network. The initial distribution of the token would be split amongst the engineer and the team they outsourced and initial investors in the project, and the remainder of the tokens could be distributed to the community involved with the project over a set number of years. This process would be very similar to the distribution of equity seen with traditional companies upon their initial public offerings. Once operations of the project are running smoothly, the developer can begin transitioning the project to be governed by a DAO, with the eventual plan to phase themselves out of working on it entirely such that the project is running fully autonomously. Once the project is completely being managed by the DAO, if the engineer had taken the time to set up the DAO’s infrastructure properly, the project will continue to grow alongside the value of its associated tokens.

Conclusion

The idea space surrounding the problems for which blockchain technology can provide value is currently very under-capitalized. If not only to provide engineers with a new tool to assist them in their entrepreneurial endeavors of exploring this space, learning blockchain can at least grant everyday developers a chance to set themselves apart from their peers by learning an emerging technology that will continue to become even more mainstream in the coming years. Going forward, the need for more developers to traverse this idea space surrounding blockchain is essential to solving the world's future uncharted problems.