DevOps for Smart Contracts in Blockchains

Ali Asbai

I certify that generative AI, incl. ChatGPT, has not been used to write this essay. Using generative AI without permission is considered academic misconduct

Royal Institute of Technology
May 2023

Abstract

Smart contracts in blockchain have gathered a lot of attention in the recent past. The space has constant innovation, and its application scope is increasing over time. Because of blockchain's inherent immutability, there is an increased weight on software quality and reliability in the space. At the same time, the role of DevOps in software development is to improve quality, reliability, and the rate of delivery. But the absence of low-level programming abstractions and the constant evolution of the platform have made popular design approaches used in traditional software development difficult to implement for smart contract development. This essay explores how the basic principles of DevOps can be adapted for the development of smart contracts. Todays tools and frameworks contain enough features to support the development of blockchain-based applications, and the essay will provide concrete procedures and general thinking to be applied to the continuous integration and continuous delivery pipelines of smart contract applications.

Contents

1 Introduction

DevOps and blockchain are terms gathering much interest in recent years. The role of DevOps in software development is to improve cooperation, effectiveness, and the rate of delivery. By implementing continuous integration, continuous delivery, and continuous verification, teams can minimise the risk of failure, quickly correct any problems, and reduce development time, which leads to quicker delivery of new functionality to the users while maintaining the quality and stability of a system [1]. Blockchain, on the other hand, is a network of nodes maintaining a distributed ledger. This allows shared business practices to be applied through software running on the blockchain to enforce, facilitate and verify the execution of specified terms of smart contracts. Contents of the ledger are immutable. This means smart contracts are not subject to bug fixes or new features, putting more emphasis on software quality and reliability during the development of the contracts, which is an aspect that DevOps can provide valuable support in.

2 Blockchain

Blockchain is a distributed network of nodes that maintain a distributed ledger. The ledger is a historical register of transactions that are timestamped and structured by a mathematical consensus algorithm. Blockchain uses cryptology as well as public key infrastructure to authenticate, validate and process data to establish trust without an intermediary [2].

There are two types of distributed systems in the blockchain space, private and public blockchains. In a public blockchain, anyone can view the network and maintain the ledger. Examples of such blockchains are Bitcoin and Ethereum. In a private blockchain, the network consists of a number of trusted users that maintain the ledger. If anyone new wants to take part in the network and ledger, they would need permission from the owner [3].

Blockchain technology utilizes cryptology in the form of hashing, public key infrastructure and digital signatures to create security, validation and authentication for its users.

The most well-developed use cases of blockchain technology lie within the financial sector. Its application in the sector ranges from many things, but the most obvious application is cryptocurrency. Blockchain technology has some experimental applications in the health sector as well. Within the sector, health data is viewed as its own special case of personal data because of its increased sensitivity. To protect this sensitive data, information sharing within the sector is very limited, resulting in slower diagnosis, more expensive testing and unsafe data transfers between caretakers. G. Akerlof suggests the use of blockchain technology to make patients the sole owners of their health data [4]. Information sharing would become simpler, safer and need less trust in this patient-centred model. Figure 1 summarizes the steps taken from the creation of a transaction til the registration of said transaction in the blockchain.

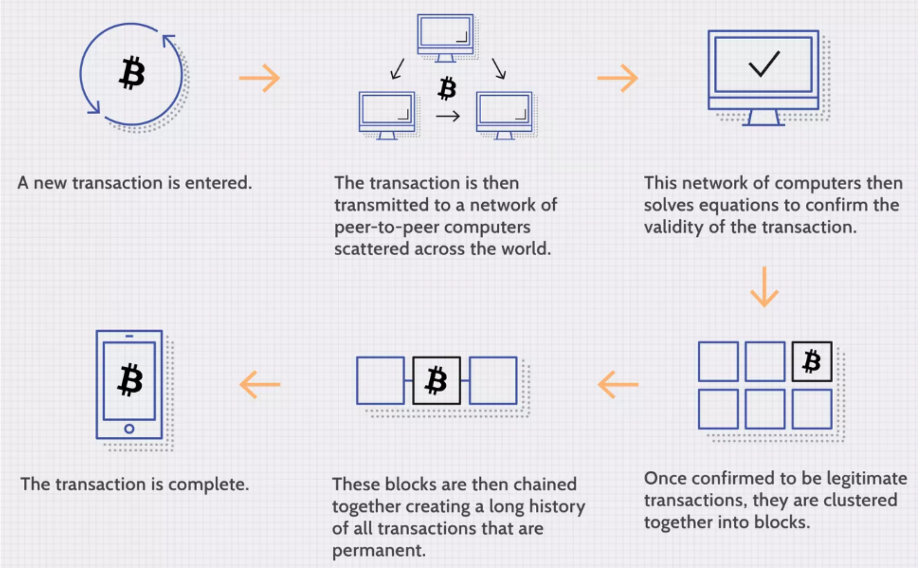
A new transaction is entered.
The transaction is then transmitted to a network of peer-to-peer computers scattered across the world.
This network of computers then solves equations to confirm the validity of the transaction.
Once confirmed to be legitimate transactions, they are clustered together into blocks.
These blocks are then chained together creating a long history of all transactions that are permanent.
The transaction is complete.

Figure 1: An overview of the steps in blockchain

3 Smart contracts

Smart contracts are software that is stored on a blockchain that automatically runs once predetermined conditions are met. Smart contracts are often used for automating the fulfilment of business deals so that all parties involved can be sure of the result without the need for an intermediary.

Smart contracts follow simple if-or-else logic that is written in code and stored on a blockchain. The network of nodes will execute the code of whatever branch in which conditions are fulfilled fully. The code that is executed could be anything from moving money from one account to another to being as simple as sending an email. At the end of code execution, the transaction is registered in the blockchain, which means that it cannot be taken back or altered and that only parties with the correct permissions can view the transaction [5].

Within a smart contract, there can exist as many and as complicated rules and stipulations as necessary to satisfy the parties involved. On top of the rules, the parties need to agree on possible exception handling as well as rules for solving possible conflicts. The smart contract can then be created by a programmer, or an increasingly popular option today is creating smart contracts through public APIs and online tools [5].

At first, glance, applying DevOps concepts to smart contract development does not seem different from applying the same concepts to traditional software development. But some inherent aspects of Blockchain technology need to be taken into consideration when adopting some DevOps principles. The following sections will introduce and adapt some core DevOps principles for smart contract development.

4 Continuous integration

In DevOps, Continuous integration is the practice of automating integration and validation of code changes. The main objective of CI is to improve software quality while reducing the time it takes to deploy software updates. These basic principles and steps can also be applied to smart contract development. Plenty of frameworks for the development of smart contracts provide means for automation and management of repeating tasks. For example, Truffle provides a development environment with an integrated testing blockchain to facilitate the compiling, deploying, testing, and debugging of smart contracts [6].

Once a build is successful, a series of automated tests are run. To be able to test smart contracts, they need to be deployed in a blockchain. This blockchain can be either an existing or temporary customized blockchain. The latter option is preferred to achieve reproducible results easily and to be easier to manipulate the environment. A major challenge with creating temporary test blockchains is the difficulty of populating said blockchain with test data. This issue can be solved in three ways. The first option is to run a specific sequence of transactions and save the resulting state. This way, a consistent test environment is created, and the blockchain can be restored to this state repeatedly, which simplifies automated testing. The second option is to fork the production blockchain. With this approach, one can simulate the same state as the production blockchain within a local environment. The final and simplest option is to use an empty blockchain without any transaction history. This approach is suitable for local testing of software that is independent of test data.

Unit tests are the most common technique to check the correctness of blockchain-based software [7]. For smart contracts, unit tests should at least cover the publicly exposed contract methods. Integration tests on the other hand, are more complex when it comes to smart contract development, since each test can interact with multiple users and contracts at a time. To simplify things, integration testing can be split into two major areas inter-blockchain interaction and smart contract and client interaction. The first of the two refers to the interaction between cooperating smart contracts, while the other refers to the interaction between smart contracts and dependent applications [8].

While unit and integration tests alone are not enough to uncover potential code vulnerabilities, combining unit and integration testing with static code analysis is therefore common. The most basic form of static analysis can be performed using a linter and can help to improve the code quality and remove minor syntax errors and structural problems. However, more advanced tools can be used to detect more severe security vulnerabilities. Such tools examine generated bytecode to identify potential vulnerable patterns and errors that may occur during software runtime [7].

5 Continuous delivery

In DevOps, continuous delivery is the practice of always building software that is ready for production. Generally, software updates are delivered through deployment pipelines to the production environment, or the production environment is supplied with an infrastructure-as-code approach. The same core concepts and steps can also be applied to smart contract development. Since deploying smart contracts is a rather rare and critical operation, having control over when builds are released to production is desirable [8]. This can be achieved by a manual approval process being implemented before release. Due to the immutable nature of a blockchain, the sole method of upgrading an existing contract is by deploying a new version of the same contract. This process is more difficult than it initially sounds since it entails manually migrating state information from the old contract as well as informing users of the new contract address. To avoid this issue, a proxy pattern [9] combined with the delegate-call mechanisms [10] can be used to replace smart contracts while preserving their address and state. Using these concepts, one can develop a solution where users directly interact with a proxy, responsible for managing state information and delegating transactions (using delegate-call) [10] to and from other upgradeable contracts that hold the relevant logic. A dynamic forwarding mechanism can be employed to enable the proxy to forward any call for any function to the logic contract [9].

After a build has been deployed, ensuring that everything is functioning correctly becomes the most crucial task. In the case of deploying smart contracts on a blockchain, it becomes necessary to verify the deployed contract to establish trust. This process is completed by verifying that the uploaded source code matches the code compiled on the blockchain [8].

After the smart contract has been deployed and validated successfully, continuous monitoring is needed to ensure optimal performance and dependability of the contracts. By continuously monitoring the behaviour of smart contracts, erroneous activity can be spotted and dealt with quickly. A useful tool for monitoring the general metrics of a blockchain is a block explorer, which is an analytics platform that allows users to look up real-time data for transactions in the blockchain [8]. Figure 2 summarizes some of the steps that can be taken in each stage of the DevOps process.

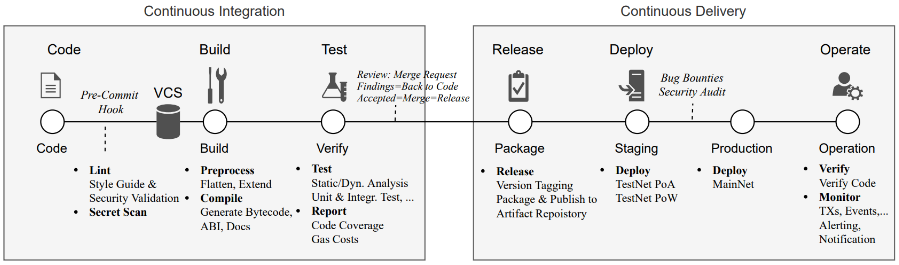

Figure 2: An overview of the stages in DevOps for smart contracts [8]

6 Reflection

Teams that develop smart contracts face massive challenges that can all be reduced with the use of DevOps, along with the correct tools and frameworks. But because of blockchain's decentralized nature as well as its immutability, some adaptions have to be made to the core practices of DevOps to fit smart contract development better. One of these adaptions is the increased focus on testing, especially static code analysis. During the work of this paper, the importance of static code analysis tools during the development of smart contracts was made abundantly clear. There is another shift in testing focus when it comes to unit and integration testing. Where the emphasis on traditional unit testing is decreased during smart contract development while there is a clear increase in integration testing, this could be a result of the fact that methods in smart contracts are publicly called and can, a lot of time, be dependent on each other, shifting the focus towards integration testing.

Another distinct hesitancy to fully automate the deployment of software to the production environment. This is another effect of blockchain's immutable nature, making it more difficult to update smart contracts once they have been deployed. Therefore, it is more practical to automate the DevOps pipeline only as far as continuous delivery goes while maintaining a manual deployment strategy accompanied by a manual check of contracts before deployment. While continuous deployment for smart contract deployment is theoretically possible, it is, for the mentioned reasons, very rarely used today [8].

7 Conclusion

The DevOps approach has been applied to a broad range of software development branches in the recent past. While its core principles can be applied to the development of smart contracts, they need adaptation. To figure out how to realise these changes, papers from state-of-the-art users of DevOps in blockchain-based applications have been studied. The work results come in the form of concrete procedures and general thinking to be applied to the continuous integration and continuous delivery pipelines of smart contract applications.

The key takeaways are that today's frameworks and tools can establish CI/CD solutions for blockchain-based software development. The challenges instead lie in choosing the right tools and frameworks for each specific application as well as in implementing the correct testing strategy for each specific use case.

For further reading about the topic, we recommend [7], [8] and [11] in the reference list.

8 References

[1] What is DevOps? - DevOps Models Explained [Internet]. Amazon Web Services, Inc. 2023 [cited 2023 May 1]. Available from: https://aws.amazon.com/devops/what-is-devops/

[2] Mougayar W. The business Blockchain: promise, practice and application of the next internet technology [Internet]. 1st edition. New Jersey: Wiley; 2016. [cited 2023 May 1]. Available from: https://books.google.se/books?hl=en&lr=&id=X8oXDAAAQBAJ&oi=fnd&pg=PR9&dq=The+Bsiness+Blockkedja:+Promise,+Practice,+and+Application+of+the+Next+Internet+Technology+df&ots=jfVTzZP3y&sig=0dOOFZ1SlXvnX4HvBGUTPE8Jz8&redir_esc=y#v=onepage&q=The%20Business%2Blockkedja%3A%20Promise%2C%20Practice%2C%20and%20Application%20of%20the%20Next%20Internet%20Techology%20pdf&f=false

[3] Tapscott D, Tapscott A. Blockchain Revolution: how the technology behind bitcoin is changing money, business and the world [Internet]. 1st edition. Portfolio; 2016. [cited 2023 May 1]. Available from: https://itig-iraq.iq/wp-content/uploads/2019/05/Blockkedja_Revolution.pdf

[4] Akerlof G. The Market for lemons: quality uncertainty and the market mechanism. The Quarterly Journal of Economics. 1970; volym (84). p 488–500.

[5] What are smart contracts on blockchain? [Internet]. IBM. 2023 [cited 2023 May 3]. Available from: https://www.ibm.com/topics/smart-contracts

[6] Overview - Truffle Suite [Internet]. Trufflesuite.com. 2020 [cited 2023 May 3]. Available from: https://trufflesuite.com/docs/truffle/

[7] Understanding the software development practices of blockchain projects [Internet]. ACM Conferences. 2018 [cited 2023 May 4]. Available from: https://dl.acm.org/doi/10.1145/3239235.3240298

[8] Wöhrer M, Zdun U. DevOps for Ethereum Blockchain Smart Contracts [Internet]. [cited 2023 May 4]. Available from: https://eprints.cs.univie.ac.at/7141/1/document.pdf

[9] Proxy Patterns [Internet]. OpenZeppelin. 2018 [cited 2023 May 6]. Available from: https://blog.openzeppelin.com/proxy-patterns/

[10] Klinger P, Nguyen L, Bodendorf F. Upgradeability Concept for Collaborative Blockchain-Based Business Process Execution Framework. Blockchain – ICBC 2020 [Internet]. 2020 [cited 2023 May 6];127–41. Available from: https://link.springer.com/chapter/10.1007/978-3-030-59638-5_9

[11] Wöhrer M, Zdun U. Design Patterns for Smart Contracts in the Ethereum

Ecosystem. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (Smart-Data) [Internet]. 2018 Jul [cited 2023 May 7]; Available from: https://ieeexplore.ieee.org/document/8726782