Check for updates

BBCPS: A Blockchain Based Open Source Contribution Protection System

Qiubing Zeng, Xunhui Zhang, Tao Wang(✉), Peichang Shi, Xiang Fu, and Chenhui Feng

College of Computer, National University of Defense Technology, Changsha 410073, China
{zengqiubing18,zhangxunhui,pcshi,fuxiang13,fengchenhui18}@nudt.edu.cn, taowang.2005@outlook.com

Abstract. In the current open source ecosystem, developers rely on internal factors, such as ideology, interesting, and other external factors, such as reputation, learning, to participate in the contribution of open source software. However these things are not enough to support their continuous contribution. Meanwhile, for social coding communities, for example Github, collaborative developers, especially peripheral contributors, they do not receive any intellectual property after participating in the contribution. The lack of effective intellectual property protection and reasonable material incentives restrict developers' participation in open source contributions to a certain extent, which therefore hinders the development of open source ecology. In this paper, we combine the Trustie open source community with blockchain technology by recording developers' contributions and corresponding tokens on blockchain. We design and implement a blockchain based open source contribution protection system, and enhance the enthusiasm of contributors to continuously participate through the transformation of property rights to potential material incentives.

Keywords: Blockchain · Open source · Social coding · Intellectual property protection · Material incentives

1 Introduction

With the development of the open source ecosystem, more and more open source projects have emerged, and more and more developers are participating in the open source contribution. The crowd intelligent development mode in the open source world leads to the rapid iteration of software and the quick bug fix process. Many open source software, such as Mysql, Spark, Tensorflow benefit from the crowd contribution and become successful in the end.

However, there are still some problems with the current open source ecosystem. **Firstly, developers lack of intellectual properties.** In the current open source communities, for example Github, the repository creators in the

© Springer Nature Singapore Pte Ltd. 2020
Z. Zheng et al. (Eds.): BlockSys 2019, CCIS 1156, pp. 662–675, 2020.
https://doi.org/10.1007/978-981-15-2777-7_54

community have all the management rights. For other contributors, especially the peripheral contributors, even though their contributions are recorded in the platform, they cannot gain any corresponding rights in addition to the role modification, which means that the current open source community does not take individual's intellectual property into consideration. **Secondly, developers especially peripheral develpers do not continuously contribute to open source projects.** Relevant research [1] shows that there are many kinds of motivations for developers to contribute to open source software, including internal factors such as ideology, interesting, internalized extrinsic factors such as reputation, learning, and external factors such as career and pay. Among these factors, Har et al. [2] thought that getting paid is the main factor that keeps developers engaged in continuous contributions. However, as of 2019, there lacks of material incentive mechanism in Github[1] that promotes the continuous contribution of contributors. By focusing on the turnover of developers in Github communities, Foucault et al. [3] found that for many popular open source projects, although there was always a large number of external contributors participated in the project contribution, the newly added peripheral contributors tend to stop contributing after a period of time. That is to say, the current open source community is difficult to maintain the continuous activity of new developers to a certain extent.

In order to stimulate developers to contribute continuously, enhance the enthusiasm of developers, promote the emergence of wisdom, Github sets up the sponsor mechanism[2]. Through this mechanism, some contributors can involve in the community's activities by continuously sponsored by others. Although this approach allows developers to continuously participate in the open source project on the basis of material incentives, it does not combine the contributions of developers with open source projects, which means that the sponsors can only sponsor a person rather than sponsor their related activities towards an open source project. And at the same time, this mechanism does not consider the intellectual property protection problem of various kinds of contributors.

Based on the above analysis, we consider combining the collaborative development behavior of developers in the open source ecosystem with intellectual property protection. By recording the contribution of developers in a more secure and tamper-proof manner on blockchain, developers' intellectual properties can be well protected. And by creating the user account and recording users' property right proportion according to different open source projects, the potential material incentives can be formed, which will promote the enthusiasm of developers and enhance their contributions.

The contributions of this paper include the following: (1) we propose a blockchain based open source contribution protection system, called *BBCPS*[3]. (2) we realize the converting of developers' contributions to intellectual properties and record their cumulative interests in the form of account balance. (3) we

[1] https://github.com.
[2] https://help.github.com/en/articles/about-github-sponsors.
[3] http://git.trustie.net/qiubing/chain_creator_nodejs_trustie_fabric2.git.

realize the non-perceived combination of traditional development behavior and blockchain network interaction.

The rest of this paper is organized as follows: Sect. 2 provides the review of the related work. The architecture of *BBCPS* is proposed in Sect. 3. Section 4 describes the core workflows of *BBCPS*. The discussion is introduced in Sect. 5. Finally in Sect. 6, we present the conclusion and future work.

2 Related Work

2.1 Trustie

Trustie[4] is an open source community, which integrates crowd collaboration, resource sharing, runtime monitoring, and trustworthiness analysis into an unified framework [4]. Trustie is mainly designed for university teachers and students and is based on an idea of grouped practice teaching, focusing on reading and maintaining high quality open source software. The Trustie platform supports open source code reading and evaluation, analysis and sharing of open source resources, and collaborative development of practical projects. Trustie can provide goal planning, milestone setting to control the progress of the project work in order to ensure the realization of the target plan. It also provides projects' progress statistics, analysis, notifications, and other mechanisms to ensure consistency and integrity of the code modified between collaboration teams [5].

There are nearly 1,000 teachers and students from more than 150 universities in China who join the Trustie community, as well as researchers from about 25 research institutes and innovation laboratories, including the Beidou open source lab, the first robotic operating system team in China called Micros, the international top robot racing team Nubot and the Guangzhou Supercomputing Center. In addition, more than 200 free software enthusiasts have contributed their own code, development experience and documents to Trustie projects.

2.2 Hyperledger Fabric

Bitcoin is a digital cryptocurrency, which was introduced in 2009 by Nakamoto [6]. The blockchain is its main core technology. Blockchain can be thought of as a public ledger, which is connected by hashes of individual blocks, each of which stores several transaction records [7]. Its core technologies include: encrypted hash, digital signature and consensus protocol. It has the advantages of security, transparency and high anonymity.

Hyperledger fabric is a secure enterprise-level alliance blockchain that supports node management and authentication. Each user can connect to each other through a certificate. Data can be managed securely. Fabric is an implementation of a distributed ledger platform that leverages familiar and proven technologies to run chaincode with a modular architecture that allows for pluggable implementation of various functions [8]. The fabric distributed ledger protocol runs on

[4] https://www.trustie.net/.

peer nodes. The role of peer nodes are divided into two types. one is a validating peer, which is used to reach consensus [9]. The other is a non-validating peer, used to endorse transactions, store blocks, and so on.

2.3 Blockchain in Intellectual Property Protection

In recent years, intellectual property protection issues have received increasing public attention because counterfeit products and trademarks will cost hundreds of billions of dollars a year [10,11]. O'Dair et al. [12] considered about the combination of the music industry and blockchain technology, which can use the blockchain technology to protect the intellectual property of musical works. The Soundchain[5] platform is one of the representatives, which is widely used in Russia to undertake property right protection activities in the music industry.

In the "Industrie 4.0", 3D printing technology is a revolutionary innovation [13]. Researchers think that its manufacturing process is a commodity [14,15] and is vulnerable to copyright issues. Holland et al. [16] combined blockchain technology to commercialize the entire process chains, which effectively prevented intellectual property theft. Once there is any risk on the way, it can be identified and the corresponding solution is provided. The program is currently being developed by a project called SAMPL.

Similar to the music and 3D printing industry, the copyright problem of artwork is also hard to deal with because its rapid spread on the Internet will cause serious negative impact to the creators and the entire industry [17,18]. Zhaofeng et al. [19] proposed a digital copyright management scheme for artworks based on watermark and blockchain, which is highly robust and secure. When the suspicious image data is abused and spread on the network without authorization, the system can be used to track the thief's responsibility.

CKshare is a trusted mold redesign knowledge sharing platform based on private cloud and blockchain technology. It uses the raw data of a private cloud storage mold to record mold data and use a blockchain network to ensure data security and reliability [20]. The platform can be extended to other similar application scenarios.

There are many areas for intellectual property protection using blockchain technology, including electronic media data such as music, images, video, 3D printing, and various product supply chains. However, there is few application focusing on the intellectual property protection of open source contributions.

2.4 Blockchain in Open Innovation

Blockchain technology was originally designed to support cryptocurrency. However it is now increasingly proven to be suitable for many other cases. The definition of open innovation is described as a distributed and reliable solution supported by blockchain technology and derivatives [21].

[5] https://vc.ru/25112-soundchain-ico.

Gitcoin is a distributed application based on the Ethereum blockchain. It is designed to reward people for their contributions to open source and wants to build a world where everyone is far from financial troubles to work. It provides opportunities for developers to work for and get paid from open source projects. Thus the corresponding open source project can attract a large number of excellent developers to participate. Rewarders write the reward rule into the smart contract. Anyone who implements a related function can reward a number of Ethercoins or Bitcoins. After a developer publishes the solution to the task, and the rewarder approves the solution, the smart contract can automatically complete the transaction[6].

One drawback of Gitcoin is that the rewarder needs to manually confirm that the solution submitted by the developer meets the requirements, which not only wastes time, but also leads to non-objective judgments. In order to solve this problem, Król et al. [22] proposed ChainSoft, which refined the process flow by using Travis CI tools and Oracles technology [23]. Then the verification process becomes automatic, and the rewarder does not need to verify the code manually. After the verification is completed, the smart contract will complete the remaining transactions. However, this method requires the rewarder to write a number of test cases in advance, which is difficult to implement.

de la Rosa et al. [24] aims at providing Small and Medium Enterprises (SMEs) with the right open innovation intellectual property protection. A Networked Innovation Room (NIR) was developed using blockchain and smart contract technology. This will reduce the concerns of SMEs about intellectual property.

Blockchain has a large number of applications in open source innovation. We can learn and reference excellent development techniques and ideas, and apply them to open source contribution protection.

3 The Architecture of *BBCPS*

In order to better protect the contribution of collaborative developers in the open source ecosystem and increase their enthusiasm of participating, we propose *BBCPS*. As shown in Fig. 1, it consists of three parts: Client, Trustie, and Fabric Network. Next, we will describe each part in detail.

3.1 Chain Tool

There are a large number of open source projects in the open source world. Developers can participate in the open source projects in a variety of ways, such as: issues, pull request, push, and comment. However the way developers want to modify the code directly can only be done by push or pull request. This involves two identities for developers: core developers and peripheral developers. The core developer can directly push the latest code that has been modified to the repository, and can review and determine whether to merge pull requests,

[6] https://gitcoin.co.

which are sent by peripheral developers. Peripheral developers can only modify the code by sending a pull request, after it is accepted by the core developers and merged into the repository, the code changes will take effect.

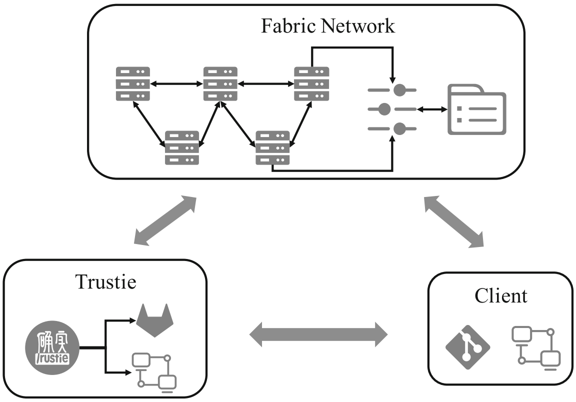

Fig. 1. The architecture of *BBCPS*

In this chapter, we use Chain Tool, also known as Client, to record core information of the push and pull request operations in the block, permanently store all data, and ensure data's authenticity by characteristics of the blockchain. In addition, in order to better stimulate developers to continuously contribute to open source projects, *BBCPS* has designed the token mechanism. Developers can get token rewards for any changes made to a project source code, for example adding a line of code to reward a token. The token value is stored in the world state of the Fabric Network, and modification to state of the world is permanently recorded in block by the blockchain. To this end, the system has designed different token calculation processing mechanisms for core and peripheral developers.

Core Developer. Core developers are the main contributors to open source projects. In a traditional distributed collaboration environment, core developer can directly push code changes to remote repository after editing locally, and other members can synchronize latest code version through the pull operation. The workflow for core developers in *BBCPS* is different from traditional process because it needs to record key information of code push and update related token value in the blockchain. For a detailed explanation of the core developer's calculation of token, see Sect. 4.2.

Peripheral Developer. There are a large number of peripheral developers in the open source world. They participate in contribution of the open source projects through issues, comments, and pull requests. Although peripheral developers are different from core developers and do not have actual administrative rights to a project, they often provide very innovative ideas and solve difficult problems in the open source projects. It is necessary to record, preserve and reward contributions of the peripheral developers. *BBCPS* takes the pull request contribution into consideration. For a detailed explanation of the peripheral developer's calculation of token, see Sect. 4.3.

3.2 Trustie Platform

Trustie is a support platform for building a community-oriented teaching practice [4]. By integrating Gitlab[7], it primarily provides services for hosting code repository, verifying authenticity of developers' contributions, and distributing transactions to the fabric network. Through Trusite API, one's contributions can be converted into tokens and recorded in related blockchain account permanently. The number of tokens represents the total contribution of developers to the project. Tokens represent intellectual properties of contributors to target projects. Currently, exchanging token number is related to the number of modified code lines while committing.

3.3 Fabric Network

BBCPS uses the Hyperledger Fabric[8] to build a blockchain network. Fabric supports pluggable consensus mechanisms and membership service components. The well-known consensus mechanism are kafka and raft, which have much higher tps than Bitcoin and Ethereum. fabric supports multi-channel, high performance and scalability.

Figure 2 shows the architectural diagram of the Fabric blockchain, divided into three parts. The left side A_1 stands for client application, which is used to interact with the fabric network. Fabric officially supports and provides a large number of SDKs for calling components in the network. The middle part is the core of Fabric network, which consists of channels and certification authorities. Among them, CA_1 and CA_2 are responsible for providing authentication for all members of the entire network. $Channel_1, Channel_2, \cdots, Channel_n$ can be regarded as different chains, which can be created infinitely for each project in the actual development process. The channel internally contains CC and O_1. CC refers to the alliance, and each time a channel is created, a new alliance is created. O_1 is a orderer node, responsible for reaching consensus in the network and acting as a mining block. The right module refers to the component relationship diagram inside $channel_2$. R_1 and R_2 represent two different organizations, and the open source community Trustie can act as one of the organizations, and R_1 and R_2 together form the alliance CC_2. P_1 and P_2 are nodes in two organizations, which can be used to install chain codes, transaction endorsements, storage

[7] https://about.gitlab.com/.
[8] https://www.hyperledger.org/projects/fabric.

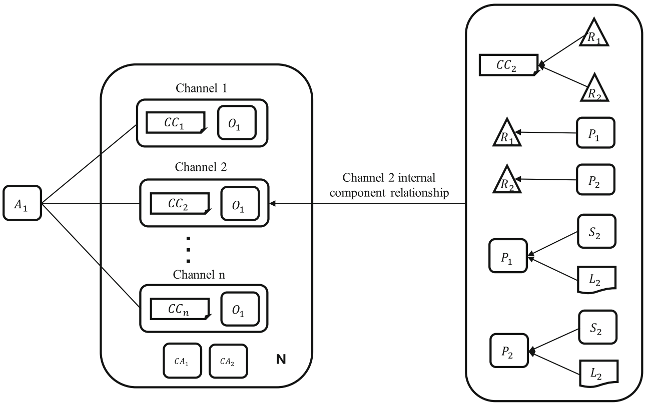

Fig. 2. The architecture of *BBCPS*'s blockchain network

books, etc., including S_2 and L_2 components. S_2 refers to the smart contract, which supports account creation, Token query, Chain push, Chain trustiePush, etc. Each smart contract is instantiated in each channel. L_2 refers to the ledger information, and two nodes P_1 and P_2 store ledger information for each channel, including the block and world state data of the respective projects.

Listing 1.1. *BBCPS* Chaincode

```
package main
var trustieLogger = shim.NewLogger("trustieContract")
type TrustieChaincode struct {}
type Account struct {
Password string
Token    int
}
func (t *TrustieChaincode) Init(stub shim.
    ChaincodeStubInterface) pb.Response {}
func (t *TrustieChaincode) Invoke(stub shim.
    ChaincodeStubInterface) pb.Response {}
func (t *TrustieChaincode) createAccount(stub shim.
    ChaincodeStubInterface, args []string) pb.Response {}
func (t *TrustieChaincode) push(stub shim.
    ChaincodeStubInterface, args []string) pb.Response {}
func (t *TrustieChaincode) trustiePush(stub shim.
    ChaincodeStubInterface, args []string) pb.Response {}
func (t *TrustieChaincode) query(stub shim.
    ChaincodeStubInterface, args []string) pb.Response {}
func main() {}
```

When a developer creates a new git project, the platform can generate a channel for it, and the channel corresponds to the built project. The developer's push and pull request operations on the project will be recorded in the blockchain, and the new token will be updated to the world state to ensure high reliability of the data information.

Chaincode. Chaincode, also known as smart contract, is a program written in Go/Java/Nodejs that implements a predefined interface that allows trusted transactions without third parties, which are traceable and irreversible. This article uses chaincode to write four methods: createAccount, push, trustiePush, and query.

Listing 1.1 shows the BBCPS chaincode, and the detailed code can be viewed on the personnal repository[9]. First of all, the chaincode defines the structure Account, which is used to store the user account password and token value, and implements the predefined interface functions. In addition, the "createAccount" function is used to create an account, and the"query" function is used to query the token value owned by the user. The "push" function is used to record the push data of core developer into the blockchain and update the user's token value in the world state based on the contribution size. The "trustiePush" function is for peripheral developers, it can also record pull request data and update the user's token value.

4 The Workflows of *BBCPS*

There are mainly three workflows in *BBCPS*, namely project creation workflow, code push workflow and pull request workflow.

4.1 Project Creation Workflow

Based on the characteristics of Trustie and Hyperledger Fabric, each time we use the chain tool to create a project, we will create the corresponding repository on Trustie, and generate the related channel in the Fabric Network.

Figure 3 shows the project creation workflow of *BBCPS*. The developer uses the Chain Tool to create the project. Firstly, the user calls the "Chain CreateProject" function by entering the project name to send a http post request to Trustie. Secondly, Trustie gets the request through the Restful API and judges the project name. If it exists, Trustie creates an empty repository for the project, otherwise the project creation fails. Thirdly, Trustie returns the Res. Fourthly, when getting the Res, Chain Tool estimates Res.state. If it indicates success, Chain Tool clones the empty repository automatically, otherwise Chain Tool stops rest operation. Fifthly, Chain Tool executes the "CreateChannel" function using curl command, which can send a request to the Fabric Network for creating a new channel. Finally, the "CreateChannel" function modifies the channel configuration file, instantiates the existing chaincode on the new channel, add organizations to the new channel and returns success, which means that the repository and its related channel are created successfully.

[9] http://git.trustie.net/qiubing/bbcpschaincode.git.

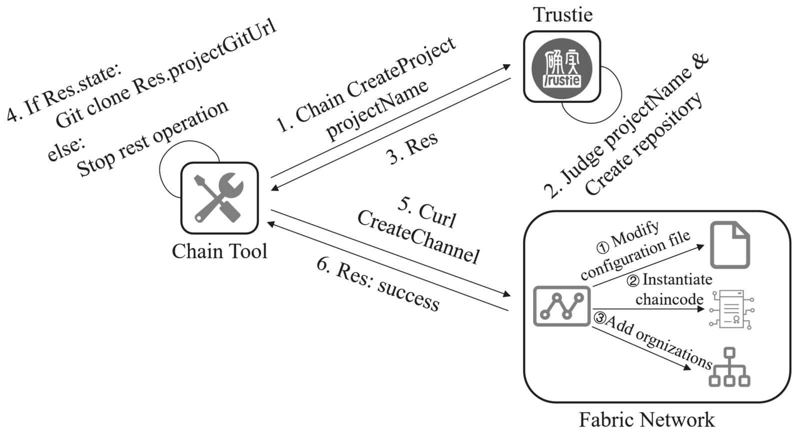

Fig. 3. Project creation workflow of *BBCPS*

4.2 Code Push Workflow

Push is one of the core functions of the platform. It is the basis for the core developers in the open source project to update the local code to the repository in real time, and also records the updated content and corresponding contribution value in the blockchain.

Figure 4 shows the code push workflow of *BBCPS*. The developer uses the Chain Tool to push the latest content. Firstly, the user calls the "Chain Push" function to upload the latest commit information from the local repo to Trustie. Secondly, Trustie updates the repository information and returns result. Thirdly, Chain Tool executes the "Push" function using curl command, which can send a request to the Fabric Network for saving push data. Fourthly, the Fabric Network calls the chaincode to send a request to the Trustie to determine whether the push content actually exists, rather than artificial falsification. Fifthly, after verifying

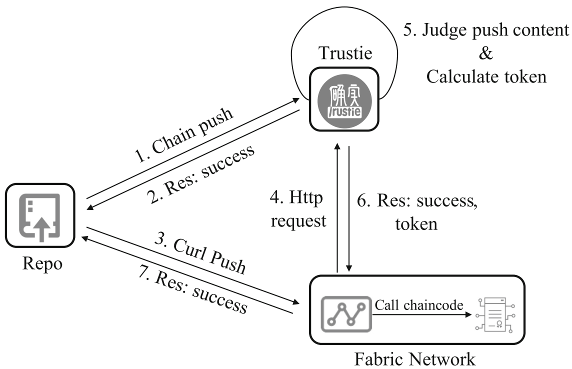

Fig. 4. Code push workflow of *BBCPS*

that the push content is true, Trustie calculates the token value of the content. Finally, Trustie returns the result and the token value. The chaincode records the push content in the blockchain and updates the state of the user account in the world state and returns final result.

4.3 Pull Request Workflow

Peripheral developers can directly participate in the construction of open source projects by using pull request, modify vulnerabilities, write new features, optimize code, and more. Once the pull request is adopted by the core developer, the peripheral developer can get a material incentive that matches the contribution.

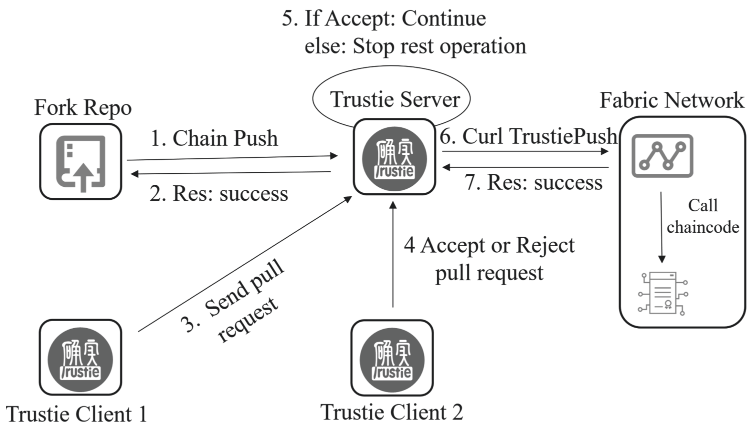

Fig. 5. Pull request workflow of *BBCPS*

Figure 5 shows the pull request workflow of *BBCPS*. The developer uses the Chain Tool and Trustie webpage to send pull request. Firstly, the peripheral developer calls the "Chain Push" function to upload the latest commit information to the forked repository, which is the same as the "git push" command. Secondly, Trustie Server updates the repository information and returns result. The webpage refreshes automatically with the latest submissions. Thirdly, the peripheral developer creates a pull request by using the forked repository as the source repository and the original repository as the target repository. Fourthly, the core developer reviews the pull request, and merges the code changes if it is acceptable, otherwise the core developer rejects the pull request. Fifthly, Trustie Server judges the response from Trustie Client2. If response equals 'Accept', Trustie Server calculates the corresponding contribution value, otherwise Trustie Server stops rest operation. Sixthly, Trustie Server calls Chain Tool to execute the "TrustiePush" function using curl command, which can send a request to the Fabric Network for saving pull request data. Finally, the Fabric Network calls the chaincode to records the pull request content in the blockchain, updates the value of the user account in the world state and returns the process result.

5 Discussion

To theoretically analyze the effectiveness of the Fabric Network, we obtain Trustie's latest pull request and commit data, about 7 times per second. We also use GHTorrent's mysql dump on September 1, 2018 to count the same data from July 25, 2018 to August 25, 2018. The results show that August 14, 2018, the largest amount of data, about 314 times per second. In addition, Fabric's official data indicates that its TPS can reach 3,500 [25]. So, we believe that BBCPS can support the real open source community scenario.

Our work is based on the open source ecosystem. Although we can't ensure that developers commit indiscriminately, if the developer commits maliciously, it will actually hinder the development of the project. Under the open source ecological mechanism, it will lead to a large loss of core and peripheral developers, which will eventually lead to project failure. We can design a visual monitoring mechanism to enhance and improve the supervision of the developer's contribution.

6 Conclusion and Future Work

In this paper, we present a distributed, secure, and trusted platform called *BBCPS* for permanently recording the contributions of developers on blockchain and assigning corresponding potential material incentives. *BBCPS* consists of three parts, Client, Trustie and Fabric Network. By integrating the git tool and fabric SDK operations, the Client can execute the git command without perceiving the fabric network process. Meanwhile, the Trustie integrates the client when merging codes through pull request. For developers, the pull request operation habit is the same as Github. The Fabric Network, which is used to store developers' contributions and property right proportion can protect the intellectual property of open source contributors and motivate developers to contribute continuously. In the future, the work can be extended to address the following issues:

(1) Optimize the material incentive mechanism. Now *BBCPS* only considers the repository changing behaviors, including the "push" and "pull request" operations. It does not take other software process products into consideration, such as issues, comments, code reviews and so on. In addition, the evaluation of code change contributions is too simple, which is judged only by the increase and decrease of the code.
(2) Replace the appropriate consensus mechanism. Currently, the consensus mechanisms "raft" and "kafka" supported by Hyperleger Fabric are non-BPFT. Once there are evil nodes in the network, the system will be paralyzed. In this regard, we will find or design a consensus mechanism that is suitable for the open source ecosystem in the future.

Acknowledgments. The research is supported by the National Natural Science Foundation of China (Grant No. 61772030) and the GF Innovative Research Program.

References

1. von Krogh, G., Haefliger, S., Spaeth, S., Wallin, M.W.: Carrots and rainbows: motivation and social practice in open source software development. MIS Q. **36**(2), 649–676 (2012)
2. Hars, A., Ou, S.: Working for free? Motivations for participating in open-source projects. Int. J. Electron. Commer. **6**(3), 25–39 (2002)
3. Foucault, M., Palyart, M., Blanc, X., Murphy, G.C., Falleri, J.-R.: Impact of developer turnover on quality in open-source software. In: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, pp. 829–841. ACM (2015)
4. Wang, H., Yin, G., Li, X., Li, X.: TRUSTIE: a software development platform for crowdsourcing. In: Li, W., Huhns, M.N., Tsai, W.-T., Wu, W. (eds.) Crowdsourcing. PI, pp. 165–190. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47011-4_10
5. Wang, H.: TRUSTIE: towards software production based on crowd wisdom. In: Proceedings of the 20th International Systems and Software Product Line Conference, pp. 22–23. ACM (2016)
6. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V., et al.: Blockchain technology: beyond bitcoin. Appl. Innov. **2**(6–10), 71 (2016)
7. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557–564. IEEE (2017)
8. Cachin, C.: Architecture of the hyperledger blockchain fabric. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers, vol. 310, p. 4 (2016)
9. Thakkar, P., Nathan, S., Viswanathan, B.: Performance benchmarking and optimizing hyperledger fabric blockchain platform. In: 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 264–276. IEEE (2018)
10. Simpson, T.W., Williams, C.B., Hripko, M.: Preparing industry for additive manufacturing and its applications: summary & recommendations from a national science foundation workshop. Addit. Manuf. **13**, 166–178 (2017)
11. Chow, D.C.K.: International Business Transactions: Problems, Cases, and Materials. Wolters Kluwer Law & Business, Alphen aan den Rijn (2015)
12. O'Dair, M., et al.: Music on the blockchain: blockchain for creative industries research cluster. Middlesex University Report 1, pp. 4–24 (2016)
13. Holland, M., Stjepandić, J., Nigischer, C.: Intellectual property protection of 3D print supply chain with blockchain technology. In: 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), pp. 1–8. IEEE (2018)
14. Yang, S., Tang, Y., Zhao, Y.F.: A new part consolidation method to embrace the design freedom of additive manufacturing. J. Manuf. Process. **20**, 444–449 (2015)
15. Kim, D.B., Witherell, P., Lipman, R., Feng, S.C.: Streamlining the additive manufacturing digital spectrum: a systems approach. Additive manufacturing **5**, 20–30 (2015)
16. Holland, M., Nigischer, C., Stjepandić, J., Chen, C.H.: Copyright protection in additive manufacturing with blockchain approach. Transdiscipl. Eng.: Parad. Shift **5**, 914–921 (2017)
17. Zeng, J., Zuo, C., Zhang, F., Li, C., Zheng, L.: A solution to digital image copyright registration based on consortium blockchain. In: Wang, Y., Jiang, Z., Peng, Y. (eds.) IGTA 2018. CCIS, vol. 875, pp. 228–237. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-1702-6_23

18. Jnoub, N., Klas, W.: Detection of tampered images using blockchain technology (2019)
19. Zhaofeng, M., Weihua, H., Hongmin, G.: A new blockchain-based trusted drm scheme for built-in content protection. EURASIP J. Image Video Process. **2018**(1), 91 (2018)
20. Li, Z., Liu, X., Wang, W.M., Vatankhah Barenji, A., Huang, G.Q.: CKshare: secured cloud-based knowledge-sharing blockchain for injection mold redesign. Enterp. Inf. Syst. **13**(1), 1–33 (2019)
21. Rosa, J., et al.: A survey of blockchain technologies for open innovation, November 2017
22. Król, M., Reñé, S., Ascigil, O., Psaras, I.: ChainSoft: collaborative software development using smart contracts. In: CRYBLOCK 2018-Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Part of MobiSys 2018, pp. 1–6. ACM (2018)
23. Zhang, F., Cecchetti, E., Croman, K., Juels, A., Shi, E.: Town crier: an authenticated data feed for smart contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 270–282. ACM (2016)
24. de la Rosa, J.L., et al.: On intellectual property in online open innovation for SME by means of blockchain and smart contracts. In: 3rd Annual World Open Innovation Conference WOIC (20160
25. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, p. 30. ACM (2018)