

EduCoin: a Secure and Efficient Payment Solution for MOOC Environment

Lingling Lu¹, Jianhai Chen*(✉)¹, Zijun Tian¹, Qinming He¹, Butian Huang¹, Yang Xiang², Zhenguang Liu³

¹ College of Computer Science
Zhejiang University
Hangzhou, P.R. China 310027
{lulingling, chenjh919, vaynetian,
hqm, butine}@zju.edu.cn

² Swinburne Research
Swinburne University of Technology
Victoria 3122 Australia
yxiang@swin.edu.cn

³ College of Computer and
Information Engineering
Zhejiang Gongshang University
Hangzhou, P.R. China 310018
liuzhenguang2008@gmail.com

Abstract—Nowadays, the rise in popularity of massive open online courses (MOOCs) is significant. However, current online education service heavily relies on a third-party platform, which acts as a middleman to process transactions occurred among system participants. On the one hand, this centralized business model is based on database management system, once compromised, the data consistency and safety of transaction records can no longer be guaranteed. On the other hand, it's more expensive to resolve disputes and possible fraud transactions via a mediator than a direct peer-to-peer trusted payment. Aiming at solving issues regarding security and inefficiencies, we propose a crypto-asset EduCoin which is designed for secure and efficient payment in MOOC environment. Specially, we present a multi-party business model based on permissioned blockchain, Composer smart contracts and network topologies that utilize the three-tiered functional architecture of Fabric to achieve fast payment and trusted audit. By leveraging benchmark tools Caliper and Hitchhiker, we develop typical workloads based on transaction scenarios applying EduCoin. The experimental results demonstrate the practicality of EduCoin and offer insights into the design trade-offs and performance bottlenecks in production. We have released EduCoin for public use.

Keywords—Permissioned Blockchain; Crypto Asset; MOOC; Payment Solution; Performance Evaluation

I. INTRODUCTION

Massive Open Online Courses (MOOCs) are online education service regarded as a popular mode of e-learning across the Internet. Generally, the service providers with copyrighted digital resources make agreements with a trusted third-party platform and authorize it to deliver paid education service to customers. For example Coursera [1], Udacity [2] and edX [3] offer courses from universities and institutions to global users. In these MOOC systems, the organization who runs the platform is a middleman between resource providers and customers. They apply a centralized application structure based on Database Management System (DBMS). In which, the payment records are stored in the database clusters and concurrency control

techniques are employed to order transactions among multiple servers. We observe that the platform often spend a lot of money on dispute resolutions and fraud transaction detection. Furthermore, traditional centralized systems are vulnerable to single-point-failure attacks and take the risk of data leakage [4].

Inspired by permissioned blockchain, an immutable shared ledger maintained by all authenticated system participants and enables developing enterprise-class business applications in a secure and efficient way. We put forward a comprehensive Fabric based payment solution to address the above issues in current DBMS based service consumption process in MOOC environment. Considering its tamper-evident feature and peer-to-peer (P2P) network, we intend to utilize blockchain smart contract to record the agreement among multiple parties which is both technologically and legally valid [5]. Note that introducing blockchain technology in payment transaction in MOOC environment is not trivial since it brings three important challenges as follows: (1) **Performance limitation**. When applying blockchain technology in payment transactions, improper configurations and design trade-offs will limit transaction performance. As a result, user experience will be degraded and system efficiency is reduced. (2) **Access control in the blockchain network**. The most distinguished feature of blockchain is the ability to enable confidential and non-frameability transactions. We need a well established access control mechanism to protect data privacy and prevent unauthorized chaincode executions from happening. (3) **User interface design**. End users consume blockchain service via a user interface which connects through the Fabric client to the peer nodes. Client applications employ distinct user identity management solutions and authentication methods for different security and efficiency considerations.

In this paper, to solve the challenges discussed above, we propose EduCoin, a secure and efficient payment solution in MOOC environment, by introducing a multi-party business model based on permissioned blockchain. Specifically, EduCoin is developed based on a three-tiered blockchain architecture which includes business layer of MOOC system,

*Jianhai Chen is the corresponding author (e-mail: chenjh919@zju.edu.cn)

access control and consensus layer and network management layer. In summary, our contributions are:

- 1) To the best of our knowledge, this is the first work to propose a decentralized payment solution in MOOC Environment. Through secure and efficient transactions on Fabric, EduCoin can achieve fast payment and trusted audit.
- 2) Considering the performance limitation, we conducted a comprehensive evaluation of transactions in four typical business scenarios. Our empirical results ensure the practicality of EduCoin network, and also help us understand the bottlenecks to process data workloads and make a balance between security and performance.
- 3) To solve the user interface design problem, we study and compare two kinds of client applications, Web APP and Standard APP, that can offer business service to end users based on the Composer business network. We have released the application and experiment for public use [6].

The remaining sections are organized as follows. We give the background and system model in Section II. We present the design of Fabric solution to EduCoin in Section III, and give the experimental methodologies to run the benchmark of EduCoin network in Sections IV, followed by analysis of experimental results in Section V and related work in Section VI before concluding the paper in Section VII.

II. BACKGROUND AND PROBLEM STATEMENT

A. Digital Assets Based on Blockchain

After the success of Bitcoin [7], many cryptocurrency projects based on the blockchain technology bloom, for example Ethereum (ETH), Ripple (XRP), EOS etc. These cryptocurrencies are usually based on public blockchain, in which there does not exist a trusted authority to authenticate the identities of peer nodes. These cryptocurrencies [8], [9], [10] often change or improve the key elements of Bitcoin blockchain system such as consensus protocol, application scenarios and incentive mechanisms. The values of cryptocurrencies depends on the supply and demand in the digital currency platform and market [11]. Cryptocurrencies based on public blockchain see frequent price movements, therefore are not good choices for enterprise use cases.

Many companies in regulated industries, for example diamond provenance, rewards point management and insurance etc., embark on permissioned blockchain. Hyperledger Fabric [12] is a project hosted by the Linux Foundation, which provides a key-value storage in the permissioned blockchain. In the financial service use cases, a key-value pair where the key can represent your bank account and the value is your account balance. In the traditional business model, you can trust the bank's database will not cheat on you. While in the permissioned blockchain model, you can trust the blockchain key-value storage that maintained by

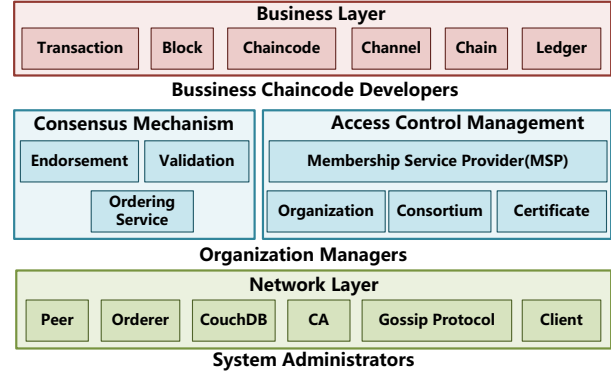


Figure 1. Functional Architecture of Fabric.

consortium of banks is correct. The key component of a blockchain is a distributed ledger that can only be updated by committing transactions in a consensus-driven way among system participants. This builds the underlying trust between multiple parties in the system, and facilitates a truly digital interaction between them [13].

B. Functional Architecture of Fabric

Fabric has a highly modular and configurable architecture. YANG Baohua [14], a member of Hyperledger Fabric contributors, identified the functional architecture of Fabric. Figure 1 shows that, from the perspectives of business chaincode developers, organization managers and system administrators, we can better understand that at different layers of Fabric, we are facing different Fabric components.

- **Business Layer:** Business chaincode developers write chaincode and design transaction flows based on the blockchain business network.
- **Access Control and Consensus Layer:** Organization managers make the plan how organizations reach consensus and configure the access control mechanism correctly.
- **Network and System Management Layer:** System administrators build up the underlying P2P blockchain network, in which peers play different roles and take their corresponding responsibilities.

C. System Model

In order to write smart contracts applying to payment appropriately, first we have to understand the transaction workflow of MOOC system. Figure 2 describes entities and their key components in EduCoin framework.

Online education service platform: offers various kinds of courses and registered users in the platform can use EduCoin to consume educational service from different service providers.

Online education service providers: Different online education service providers such as Zhejiang University or Stanford University have abundant course resources. In order

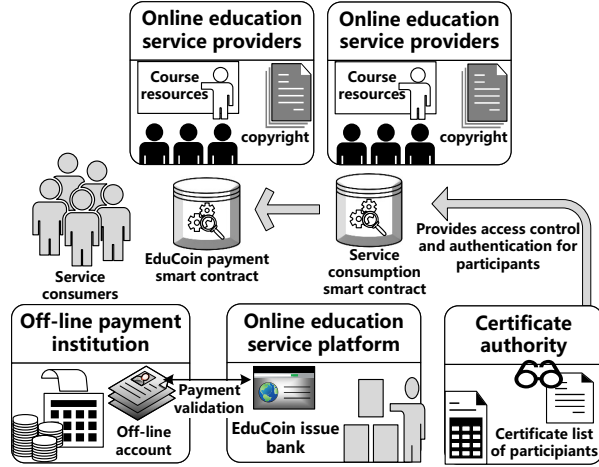


Figure 2. Entities and their key components in EduCoin framework.

to reach more users, they can join the business network. Service providers set the price of a specific course with possession of copyright in EduCoin. Once a user finishes the payment, she have access to the online education service. Providers and platform then receive a percentage of the EduCoin service profits based on their contact.

Service consumers: can purchase EduCoin with legal tender in the platform. The exchange rate between them is fixed by smart contract. In other words, the price of EduCoin depended on the legally enforceable agreement between business parties in MOOC system. What's more the purchase history of service consumers is recorded in the history transaction of blockchain, a distributed ledger trusted by all entities.

EduCoin issue bank: is in charge of issuing and transferring EduCoin to users. The administrator of bank account is authorized to issue EduCoin.

Off-line payment institution: provides the proof of user payment validity to the administrator of EduCoin bank account. Therefore when the purchase payment of EduCoin finishes, correct amount of EduCoin will be successfully transferred to user's account.

Certificate authority: (CA) is responsible for managing certificates of blockchain network members. Fabric component Membership Service Provider (MSP) provides business participants with digital identity validations that can be used in access control over smart contract services and transaction executions in chaincode container.

III. EDUCOIN PAYMENT SOLUTION

In this section we propose EduCoin, our secure and efficient payment solution. It utilizes a three-tiered functional architecture of Fabric to achieve fast payment and trusted audit. We describe each Fabric layer in detail.

A. Business Layer of MOOC System

We choose four typical scenarios during the transaction settlement in MOOC system applying EduCoin. We can simplify the chain ledger as a key-value storage. *UserRecharge*, *RejectUserRecharge* and *TokenTransferU_U* transactions modify the value of a specific key in the ledger. Query transactions search from the Org's CouchDB for the state value of a specific key. The four scenarios are described below in detail.

1) *Scenario #1: User submits UserRecharge transactions:* If a user wants to recharge some EduCoin. She first submits *UserRecharge* transactions. The corresponding smart contract logic creates asset *UserTokenRecharge*, which is recorded in the ledger and contains properties (1) tokenNum, (2) rechargeID, (3) user and (4) confirmBank.

2) *Scenario #2: Administrator submits RejectUserRecharge transactions:* If an administrator finds failed off-line payment of users, she first submits *RejectUserRecharge* transactions. The corresponding *UserTokenRecharge* asset is then updated.

3) *Scenario #3: User submits TokenTransferU_U transactions:* If a user wants to transfer some EduCoin to another user's account. She submits *TokenTransferU_U* transactions. The corresponding smart contract logic checks whether user's account balance is enough and transfer the EduCoin to another user. Their account balance is updated.

4) *Scenario #4: User queries GetUserTokenRechargeUserR:* If a user wants to know the detail of her *UserRecharge* transactions that rejected by administrator, she can queries *GetUserTokenRechargeUserR* with userID.

B. Access Control and Consensus Layer

1) *Access Control in Chaincode:* The following example will help us understand the access control and business interactions when deploying a Composer smart contract on Fabric with two organizations. The organization Org1 is represented by Alice, who is the business network administrator. She can onboard new business participants in Org1. For example, she add some participants. Among them, Jdoe is a service provider, Amy is a bank administrator. The business network administrator for Org2 is Bob. He also onboards new business participants in Org2. Kate is a user and Max is a regulator. Alice, Jdoe and Amy register and enroll with the certificate authority (CA1) of Org1. While Bob, Kate and Max register and enroll with the certificate authority (CA2) of Org2. The certificate of each user is used to authenticate to the business network.

2) *Offer Business Service:* End users like Alice from Org1 and Bob from Org2 can access client application to submit transactions and carry out daily business tasks. We identify two kinds of client applications, Web APP and Standard APP, that can offer business services based on Fabric network to end users.

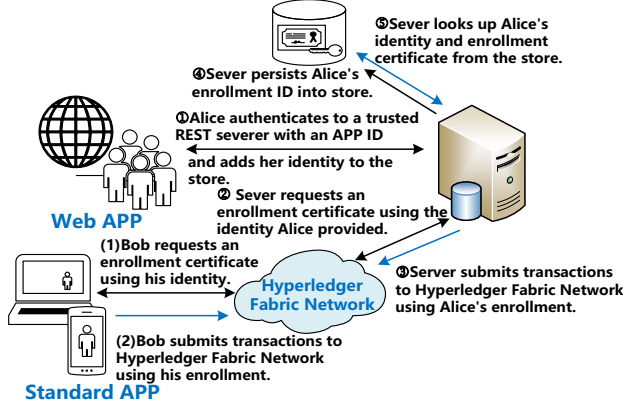


Figure 3. Two kinds of client applications to interact with Fabric network

Web APP: Note that, in the production environment, users can not interact with Fabric network directly. Instead a user authenticates to a trusted REST severer with an APP ID. The severer has the responsibility to manage users' certificates. What's more the REST severer is able to deal with some additional business logic.

Standard APP: If a user is unwilling to give away her certificate and secret key, she can use a standard app. This app stores user's certificate in a PC or a mobile phone, and sends transactions to Fabric network directly. Generally speaking, a PC or a smart phone belongs to a single user and can thus represent her identity.

Figure 3 shows how end users access client applications to interact with Fabric network. If Alice uses a web app, she first authenticates to a trusted REST severer with an APP ID. The server will add her identity to the store. Secondly, sever requests the Fabric network for an enrollment certificate using the identity Alice provided and submits transactions to Fabric network using Alice's enrollment. Finally, sever persists Alice's enrollment ID into the store and can look up her identity and enrollment certificate from the store next time. If Bob uses a standard app, he requests the Fabric network for an enrollment certificate using his identity and submits transactions to using his enrollment.

3) *Consensus and Transaction Endorsement:* The business model of EduCoin is different from token transfer business model in a centralized application based on DBMS. The transactions of EduCoin business have to be recorded in the ledger shared by multiple organizations. Peers of organizations reach consensus on the ledger, which represents their digital interactions and builds the underling trust between multiple parties. Consensus in Fabric is divides into three phrases: *endorsement*, *ordering service* and *validation* [15].

If Alice from Org1 submits transactions to onboard new Org1 participants Jdoe and Amy. These transactions must be endorsed by both peers form Org1 and Org2. What's more,

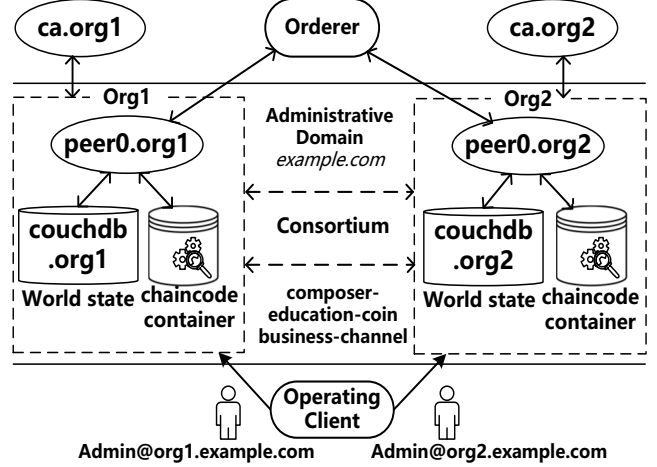


Figure 4. Topology of Educoin business network.

applying blockchain technology can also meet the business security requirements of accountability and non-frameability.

C. Network and System Management Layer

Figure 4 shows the topology of EduCoin business network. The network entities are compromised of an *Orderer* node, two *CA* nodes, two *Peer* nodes, two *CouchDBs* and an *operating client* node. These peer nodes belongs to the same *Administrative Domain* (AD) *example.com*. There are two organizations in this AD, *Org1* and *Org2*. The two organizations form a *Consortium* and join a same business-channel, the *composer-education-coin-business-channel*. Nodes are able to communicate with each other by their corresponding domain names.

Orderer node provides ordering service which is responsible for broadcasting state updates to peers and establishing consensus on the order of transactions. The world state stores the current values of all ledger states. If a key-value pair is recorded in the blockchain and the key-value was updated several times via transactions. The world state only stores the latest value of the key [16].

IV. EXPERIMENTAL METHODOLOGY

In this section we present the experimental methodologies to run the benchmark of EduCoin transactions. We first discuss the experimental tools and environments, then explain evaluation metrics.

A. Experiments Implementation

Hyperledger Caliper [17] is a framework for analyzing various blockchain platforms with performance indicators in success rate, transaction throughput, latency and resource consumption. In this research, we use Fabric v1.1 as the underlying blockchain platform, Composer v0.19 as the chaincode framework and Caliper [17] as the benchmark tool. Besides since Composer REST Sever is able to convert

the Composer model into RESTful APIs, we use Hitchhiker [18] to measure the throughput of the RESTful APIs in query transactions. Combining the benchmark results of Caliper and Hitchhiker, we are able to make a comparison between Standard App and Web App. Since Standard APP uses *composer-client* to interact with Fabric network, the same method Caliper uses. On the other hand Web APP uses RESTful API, the method Hitchhiker applies.

The experiments were run on a 5-node cluster interconnected with 1Gbps networking. Nodes are synchronized with an NTP service. Figure 4 shows the topology of EduCoin business network. The *Orderer* is a solo node. *Org1* and *Org2* each runs on a node with 16GB RAM, 2.0GHz 8-vCPU, 1TB hard drive and Ubuntu-16.04.5 64bit operation system. *CouchDB* runs on a node with 32GB RAM, 2.0GHz 4-vCPU and Ubuntu-16.04.5 64bit operation system.

B. Evaluation Metrics

The output statistics of running an experiment with different configurations can be used to evaluate the EduCoin blockchain against three performance metrics.

- **Throughput:** measured as transactions per second. The Caliper client sends transactions to the blockchain network, it's able to measure the flow rate of all transactions through the test cycle.
- **Latency:** measured as the average time that transactions need to complete and return responses.
- **Transaction Failure Rate:** measured as the percentage of failed transactions versus total number of transactions during a test cycle.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we provide experiments to demonstrate the performance and practicality of EduCoin blockchain system.

A. Throughput and Latency at Peak Performance

Fabric-1 represents the experimental environment where all docker processes of Fabric network entities are running on a single Ubuntu-16.04.5 server. Fabric-3 denotes the environment where the docker processes of Org1 and Org2 run on two separate servers. Fabric-5 expresses the environment where the CouchDBs of each Org run on separate servers.

Table I
THE TRANSACTION THROUGHPUT(#tx/s) IN FABRIC-1, FABRIC-3 AND FABRIC-5

Transaction Type	Fabric-1	Fabric-3	Fabric-5
UserRecharge	9	10	15
RejectUserRecharge	9	10	13
TokenTransferU_U	3	5	7

Cryptographic operations in Fabric are CPU intensive. Consequently, we can observe from Table I and II that as the physical environment improves, Fabric can achieve

Table II
THE TRANSACTION LATENCY(s) IN FABRIC-1, FABRIC-3 AND FABRIC-5

Transaction Type	Fabric-1	Fabric-3	Fabric-5
UserRecharge	13.4	11.17	6.32
RejectUserRecharge	18.8	15.72	5.38
TokenTransferU_U	14.22	10.9	5.94

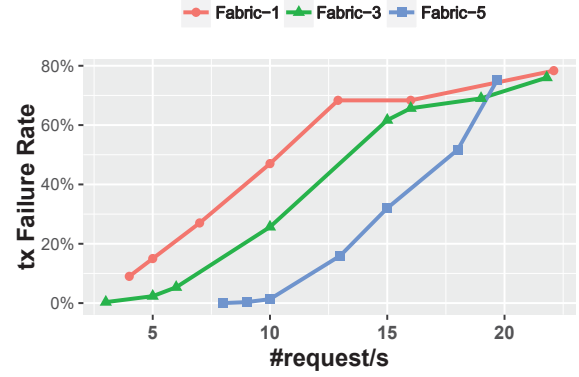


Figure 5. Transaction failure rate of *TokenTransferU_U* with varying request rates in Fabric-1, Fabric-3, and Fabric-5 clusters.

better performance in all evaluation metrics. When we separate Fabric peers and CouchDBs, nodes are more capable to do calculations. The influence of CPU power on the performance of Fabric transaction are consistent with what reported by [15]. Another observation is that as request rate increases, the transaction failure rate of *UserRecharge* and *RejectUserRecharge* always keep zero. In the meanwhile, the transaction failure rate of *TokenTransferU_U* keeps rising.

B. Transaction Failure Rate with Varying Request Rates

Figure 5 plots the transaction failure rate of *TokenTransferU_U*. When request rate increases, the failure of *TokenTransferU_U* is resulted from the *MVCC Validation* of Fabric. Multi-Version Concurrency Control [19] checks whether the read-write set of transactions satisfied the *MVCC* requirements. We observe that if the request rate is far beyond the processing capability of Fabric network, the transaction failure rate will increase accordingly. The transaction failure rate is effected by experimental environment as Fabric-5 achieves better performance because of more powerful CPU calculation ability. In the production environment, to avoid failure of user requests caused by *MVCC*, we may need to (1) write retry logic on the client side or (2) collect the user requests in a queue and send the request in a steady lower rate.

C. Query in Web App and Standard App

Table III shows the query transaction throughput (#tx/s) of web app and standard app in Fabric-1, Fabric-3 and Fabric-5. We observe that as the experimental environment improves, CouchDB is more capable to process query requests.

Specifically, Fabric-5 performs up to 1.33x and 1.44x higher throughput than Fabric-1 and Fabric-3 respectively. Another observation is that the throughput of web app is around 50 tx/s which barely increases in Fabric-1, Fabric-3 and Fabric-5. This behavior indicates that *composer-rest-server* is not as efficient as *composer-client* and limits the performance of application based on EduCoin Fabric network.

Table III
THE QUERY TRANSACTION THROUGHPUT(#tx/s) OF WEB APP AND STANDARD APP IN FABRIC-1, FABRIC-3 AND FABRIC-5

Experimental Environment	Fabric-1	Fabric-3	Fabric-5
Standard App	140	152	202
Web App	50	54	54

VI. RELATED WORK

There has been considerable interest in the development of applications based on blockchain. For permissioned blockchain, [20] designs a physical access control management system based on Fabric and Composer. Our work differs from theirs in that, we do a comprehensive study on business scenarios and benchmark the EduCoin network in a more complex and realistic node topology.

In order to evaluate the performance of these applications, we can either develop typical benchmarks or leverage existing benchmark frameworks. Blockbench [21] is a framework that can be used for analyzing Fabric networks. However Blockbench only supports Fabric v0.6, and the architecture of Fabric has changed a lot since Fabric v1.0. A huge difference is the role of Fabric node. For Fabric version 1.0 and later versions, Fabric nodes are divided into endorsing peer, ordering peer and validating peer, thus Fabric network becomes more complicated. Caliper supports Fabric v1.1 and Composer v0.19, applying a northbound interface (NBI) for users to develop typical workloads to benchmark blockchain networks. Caliper now supports Fabric v1.1, Sawtooth v1.0, Iroha v1.0 and Burrow v1.0 [17].

VII. CONCLUSION

To enable secure and efficient payment for MOOC environment, we propose a crypto-asset EduCoin based on smart contract framework Composer in this paper. It applies characteristics of permissioned blockchain to provide multiple parties in the enterprise ecosystem with a trusted ledger and confidential business transactions. Specially, we utilize the three-tiered functional architecture of Fabric to achieve fast payment and trusted audit. We provide experiments to demonstrate practicality of EduCoin system and discuss the performance bottlenecks.

ACKNOWLEDGMENT

We wish to acknowledge developers of Hyperledger Fabric, Composer and Caliper for their contributions to the

blockchain technologies. This research is financially supported by the National Key R & D Program of China No. 2017YFB1401300 and No. 2017YFB1401301 and Zhejiang Provincial Key R & D Project No. 2019C01055.

REFERENCES

- [1] J. Knox, S. Bayne, H. MacLeod, J. Ross, and C. Sinclair, "Mooc pedagogy: the challenges of developing for coursera," *Recuperado el*, vol. 1, 2012.
- [2] "Udacity, discover your perfect program in our schools," accessed January 4, 2019. [Online]. Available: <https://www.udacity.com/>
- [3] "edX, accelerate your future. learn anytime, anywhere," accessed January 4, 2019. [Online]. Available: <https://www.edx.org/>
- [4] M. R. Moore, "Exploring critical infrastructure single point of failure analysis (spfa) for data center risk and change management," Ph.D. dissertation, Northcentral University, 2019.
- [5] P. De Filippi and S. Hassan, "Blockchain technology as a regulatory technology: From code is law to law is code," *arXiv preprint arXiv:1801.02507*, 2018.
- [6] "Education coin," accessed January 4, 2019. [Online]. Available: <https://github.com/lulinglingcufe/EducationCoin>
- [7] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [8] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016, pp. 45–59.
- [9] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474.
- [10] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 104–121.
- [11] "Coinbase," accessed January 4, 2019. [Online]. Available: <https://www.coinbase.com/>
- [12] "Hyperledger fabric," accessed January 4, 2019. [Online]. Available: <https://github.com/hyperledger/fabric>
- [13] N. Gaur, L. Desrosiers, V. Ramakrishna, P. Novotny, S. Baset, and A. Odowd, *Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer*. Packt Publishing Ltd, 2018.
- [14] Y. Baohua and C. Chang, *The Principle, Design and Application of Blockchain*. China Machine Press, 2017.
- [15] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.
- [16] P. Thakkar, S. Nathan, and B. Viswanathan, "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)*. IEEE, 2018, pp. 264–276.
- [17] "A blockchain benchmark framework to measure performance of multiple blockchain solutions," accessed January 4, 2019. [Online]. Available: <https://github.com/hyperledger/caliper>
- [18] "a restful api test tool," accessed January 4, 2019. [Online]. Available: <https://github.com/brookshi/Hitchhiker>
- [19] C. H. Papadimitriou and P. C. Kanellakis, "Concurrency control by multiple versions," *ACM Trans. Database Syst.*, (United States), vol. 1, 1984.
- [20] V. P. Sara Rouhani and R. Deters, "Physical access control management system based on permissioned blockchain," in *IEEE Blockchain Conference*, 2018.
- [21] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1085–1100.