IBM Blockchain: An Enterprise Deployment of a Distributed Consensus-based Transaction Log

Ben Smith and Konstantinos Christidis

1 Extended Abstract

The backdrop of multiple scandals involving the corruption and falsification of our financial and business organizations' records accentuates the need for tamper-resistant data stores [2]. The United States alone has over 10,000 [8] rules and regulations that pertain to the proper storage and maintenance of an organization's data [6]. The Sarbanes Oxley Act [1], for example, include standards around the trustworthiness of data, emphasizing the importance of the audit log of an organization's records. A motivating example of the reason for these rules and regulations is as follows: if a member of an organization with a vested interest in pertinent outcomes can manipulate the organization's data after the fact, he or she can make it appear that an organization had no record of an embezzlement scheme, which limits or eradicates the organization's liability. Situations like these point to a broader notion: the future of the world's economic stability hinges on the ability to trust and believe in the integrity of data that is maintained by our institutions in health care, finance, business, and government.

Blockchain networks can help address this issue of data tampering and falsification. In a blockchain network, peer nodes (hereafter *peers*) run a piece of software that allows operations on a common, shared data store. Each peer maintains a local copy of the data store and a record of transactions with that store, but all modifications to the data store must be proposed to and agreed upon by the network. Peers use public key cryptography to achieve authenticity and non-repudiation of proposed changes to the datastore. Peers must also form a consensus on which transactions to accept, which to discard, and the order of transactions' occurrence. Choosing

Ben Smith
IBM, e-mail: benjsmi@us.ibm.com

Konstantinos Christidis
IBM e-mail: kchrist@us.ibm.com

the right consensus algorithm depends on a number of factors, including network characteristics and performance targets [14, 13].

In a blockchain network, each peer wishing to modify the data store signs their proposed transaction using their private key and then broadcasts. Peers who receive this transaction only relay the transaction after inspecting its syntactical validity and its authenticity (using the transmitting peer's stored public keys). Peers group and package incoming transactions into a set, called a "candidate block," after a prespecified time interval elapses (the *block period*)[1]. Receiving peers validate the candidate block by ensuring that it contains valid transactions, and that it contains the hash digest of the preceding block. If a peer determines the candidate to be valid, it adopts the candidate as the next block in the chain. Each peer then appends the included transactions to the local transaction log and updates its local copy of the data store accordingly. This set of linked authenticated, verified transaction batches is what we refer to as the blockchain.

Blockchain and its use in blockchain networks makes it difficult to repudiate (deny) that a transaction has occurred, or manipulate the history of transactions post hoc. A peer may be able to filter out transactions from their own copy of the ledger, but other peers will reject these post hoc changes, meaning that all other copies on the network will remain untouched.

To use the blockchain network, developers make use of a REST API that allows the deployment of chain code, which is the programmatic articulation of the functions entailed in a transaction. This is often referred to in the literature as a smart contract [11, 12]. For example, if the application using the blockchain needs to record that user A has transferred the ownership of an item to user B, the chain code will contain source code that implements this transaction. This chain code will check, for example, that A has the item to begin with, check that B can receive the item, and then record that A transferred the item to B as a transaction.

IBM is a premiere member in the Hyperledger Project[2], a collaborative effort run by the Linux Foundation whose goal is identifying and articulating goals for a cross-industry standard for blockchain. IBM has contributed open source software[3] for blockchain network peers that is available from GitHub. Additionally, users can obtain a deployed and configured network of peers through IBM's Blockchain Service, which is available from IBM's next generation cloud application development platform, Bluemix[4]. IBM has produced several demos that utilize blockchain technology, available from IBM's blockchain website, and deployable as applications on Bluemix[5].

The blockchain peers in IBM's implementation form a private network where communication is established via HTTP 2.0 and the gRPC protocol [9]. The fact

[1] Note that the order of transactions in the candidate block, as well as the identity of the peer or peers that will propose it to the network depends on the consensus mechanism that the network uses.

[2] https://www.hyperledger.org/

[3] https://github.com/hyperledger/fabric

[4] https://console.ng.bluemix.net/catalog/services/blockchain

[5] https://ibm.com/blockchain

that the network is private, or permissioned, means that there is no risk of a Sybil attack [7]. The consensus mechanism is a pluggable component that can be changed according to the network's requirements. Existing implementations include (a) the classic Practical Byzantine Fault Tolerant (PBFT) algorithm [5, 3], which offers a solution to the Byzantine Generals Problem [10] in asynchronous settings, like the Internet, and (b) Sieve [4], an algorithm developed by IBM Research that adds a speculative execution and verification phase to the PBFT protocol to allow the network to reach consensus on the output state of candidate transactions.

Blockchain networks like IBM's can help prevent the post hoc tampering of transaction histories and organizations' data. Effective use of blockchain networks requires that non-trusting (and indeed competitive) organizations agree to participate in a shared network. Each organization can use the REST API described above to create and record transactions on the blockchain that are then validated by the other participating organizations in the network. Consider a consortium of organizations agreed to store their financial holdings on a shared blockchain network–it would be difficult (if not prohibitively expensive) for a paricipant in this network to manipulate transactions stored on the blockchain after the fact. This level of transparency and non-repudiation makes covering up a financial scandal more difficult than it is today. Finally, organizations will be incentivized to participate in blockchain networks in the future because there is no central mediating authority–the network as a whole acts as the validator and purveyor of truth. This presentation will review blockchain concepts and implementation details. Additionally, this presentation will provide information on how to access and utilize IBM's Blockchain Service in Bluemix and provide a demonstration of how blockchain technology can help shape the nature of future business transactions.

References

[1] Act SO (2002) Public company accounting reform and investor protection. Public Law (107-204)

[2] Bratton W (2002) Does corporate law protect the interest of shareholders and other stakeholders: Enron and the dark side of shareholder value. Tulane Law Review 1275

[3] Cachin C, Guerraoui R, Rodrigues L (2011) Introduction to reliable and secure distributed programming. Springer Science & Business Media

[4] Cachin C, Schubert S, Vukolić M (2016) Non-determinism in byzantine fault-tolerant replication. arXiv preprint arXiv:160307351

[5] Castro M, Liskov B, Others (1999) Practical byzantine fault tolerance. In: OSDI, vol 99, pp 173–186

[6] Chan CC, Lee Y, Lam H, Zhang XM (2004) Analytical method validation and instrument performance verification. John Wiley & Sons

[7] Douceur JR (2002) The sybil attack. In: Peer-to-Peer Systems, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 251–260, URL

http://link.springer.com/chapter/10.1007/3-540-45748-8_24

[8] Gerr P, Babineau B, Gordon P (2003) Compliance: the effect on information management and the storage industry. Enterprise Storage Group Technical Report

[9] Google (2016) grpc: A high performance, open source, general rpc framework that puts mobile and http/2 first. URL http://www.grpc.io/

[10] Lamport L, Shostak R, Pease M (1982) The byzantine generals problem. ACM Trans Program Lang Syst 4(3):382–401, URL http://doi.acm.org/10.1145/357172.357176

[11] Szabo N (1994) Smart contracts. URL http://szabo.best.vwh.net/smart.contracts.html

[12] Szabo N (1997) The idea of smart contracts. URL http://szabo.best.vwh.net/smart_contracts_idea.html

[13] Tschorsch F, Scheuermann B (2015) Bitcoin and beyond: A technical survey on decentralized digital currencies. IACR Cryptology ePrint Archive 2015:464

[14] Vukolić M (2015) The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In: Proc. IFIP WG 11.4 Workshop on Open Research Problems in Network Security (iNetSec 2015), URL http://www.vukolic.com/iNetSec_2015.pdf