Design Principles for Migrating from Traditional Systems to Blockchain Systems

Mohammad Saidur Rahman*, Ibrahim Khalil*, Abdelaziz Bouras¥,
Mohammed Atiquzzaman, Senior Member, IEEE†
*Computer Science and IT, School of Science, RMIT University, Melbourne VIC 3000, Australia
{mohammadsaidur.rahman, ibrahim.khalil}@rmit.edu.au
¥ Qatar University, Qatar
{abdelaziz.bouras}@qu.edu.qa
†School of Computer Science, University of Oklahoma, Norman, OK
{atiq}@ou.edu

Introduction

Blockchain is a distributed database technology that builds on a tamper-proof list of time-stamped transaction records. The data structure of blockchain is a chained list of blocks [1]. Each block contains a hash of the previous block's representation, thus creating the chain. As a result, historical transactions in the blockchain cannot be deleted or altered without invalidating the chain of hashes. With a combination of computational constraints and incentive schemes (e.g. Proof-of-Work (PoW) in Bitcoin [2]) for block creation, the tampering and revision of the information in the blockchain can be prevented.

Blockchain also provides a general-purpose programmable interface. Programs can be deployed and run on a blockchain; such programs are called smart contracts [3]. The result of a smart-contract invocation is stored in public data storage. Smart contracts can express triggers, conditions, and business logic to enable more complex programmable transactions. Codes in these languages are deterministic and rely on a closed-world assumption: the only information that is stored on the blockchain is available in the runtime environment. Smart contract code is deployed with a specific type of transaction that stores the code file in all of the nodes in a Blockchain. The storing of a smart contract code file has to be done using a valid user account. Once the code file for a smart contract is created, it is fixed and cannot be changed [4]. Therefore, the deployment of smart contract code to the blockchain is immutable similar to any other blockchain transaction. Smart contracts offer a way to execute code directly on the blockchain network once they are deployed. Business rules in traditional business process management systems can be expressed as smart contracts. A simple example of a smart-contract-enabled service is the conditional transfer of money if a particular condition is fulfilled.

The design principles of a blockchain-based system are not yet explored. Designing a blockchain-based system from scratch can be considered as a straight forward approach. The components of a business management system need to be identified by a designer. Next, business rules need to be modeled as smart contracts. Finally, the blockchain-based data model is defined to capture business process transactions. There are several challenges of designing a blockchain system from scratch. First, this design method is expensive. The outcome of designing from scratch is a brand new system. A significant amount of workforce requires to be employed for building this new system which costs a lot. Second, an existing system needs to be replaced by the new system, which is time-consuming. Converting existing business rules to smart contracts and relational data models to blockchain data models (i.e., distributed ledger) may take lots of time. Finally, designing from scratch requires expert

knowledge. As a result, many of the existing business management systems would not be able to adopt blockchain technology.

Another approach for designing a blockchain-based system is to integrate an existing business management system with a blockchain-based system [4]. Designing a blockchain-based business management system that would be integrated into an existing traditional system is a challenging task. To state clearly, the data model and business rules of the existing business management system can be converted to the blockchain data model and smart contracts, respectively. A blockchain-based system uses cryptography (hashing and digital signature) heavily, and business rules need to be executed as smart contracts in the blockchain network. Thus, the design approach of a blockchain-based system for migration purposes is different from traditional system design approaches. Existing research work does not thoroughly guide how to migrate data models of existing business systems to blockchain-based systems. Therefore, developing a set of design principles is urgent for transforming an existing business management system to a blockchain-based system.

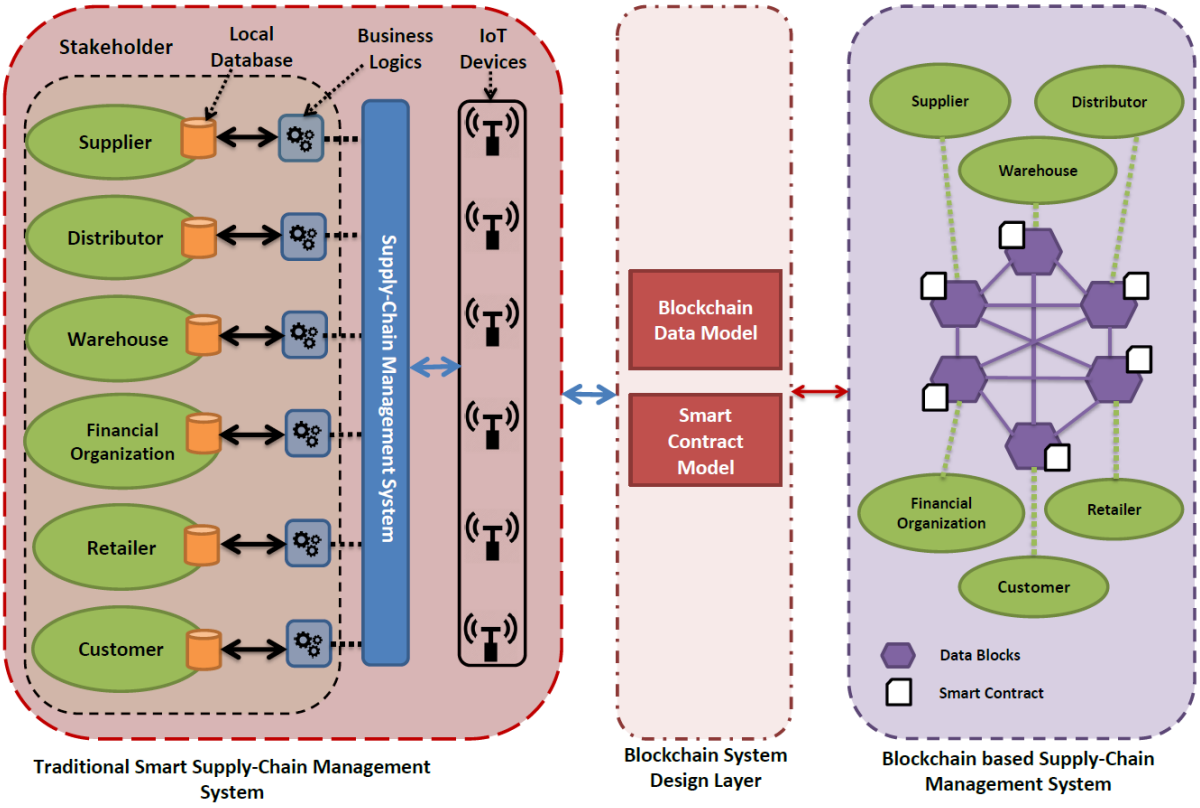

Figure 1: A conceptual representation of the design layer for migrating to blockchain system

What is expected in the design principles?

To discuss design principles in detail, we need to understand the place of designing in the migration process. Figure 1 depicts a conceptual representation of the design layer for migrating to the blockchain system. We use a scenario from the smart supply-chain management systems for simplicity. In the smart supply-chain management systems, the Internet-of-Things (IoT) devices are used to monitor, report, and exchange IoT data for automating the supply-chain process. In general, a traditional smart supply-chain management system, similar to a simple supply-chain management system, consists of four components: stakeholders, individual business logic, local database, and supply chain management system (SCMS). Stakeholders are the participants in the supply chain.

Different stakeholders have different business logic and local databases. We name the business logic of a stakeholder as individual business logic. The individual business logics contain a set of business rules used by the respective stakeholder. The local database of a stakeholder contains the transactions or records of the stakeholder, including IoT data. For the sake of simplicity, we assume that a local database uses a relational data model [5]. The supply chain management system (SCMS) is the platform that connects all of the stakeholders and their IoT devices. For example, SCMS can be hosted in the cloud, and all of the stakeholders interact with the cloud. IoT data are sent to the cloud for storing and to be used by business logic. Lack of transparency and several security threats exists in the cloud, including confidentiality and auditability, due to the centralized nature [6]. On the other hand, a simple blockchain-based supply chain management system may have only two components: stakeholders and blockchain network. The definition of stakeholders is similar to the one stated in the traditional smart supply-chain management system. However, a stakeholder holds a copy of a distributed ledger instead of holding local databases. Moreover, the business logic is converted to smart contracts and distributed among nodes in the blockchain network. The blockchain network contains all of the smart contracts and maintains an immutable distributed ledger for transactions. We name the distributed ledger as the blockchain data model.

Therefore, the migration from the traditional supply chain management system to the blockchain-based supply chain management system is not straightforward. It is necessary to ensure the "error-proof" conversion from the relational data model to a blockchain data model and business logic to smart contracts. As a result, the design layer remains between the traditional and blockchain-based supply chain management system, as depicted in Figure 1. The design principle should offer a set of instructions that allow automatic conversion of a relational data model and business rules into the blockchain data model and smart contracts, respectively.

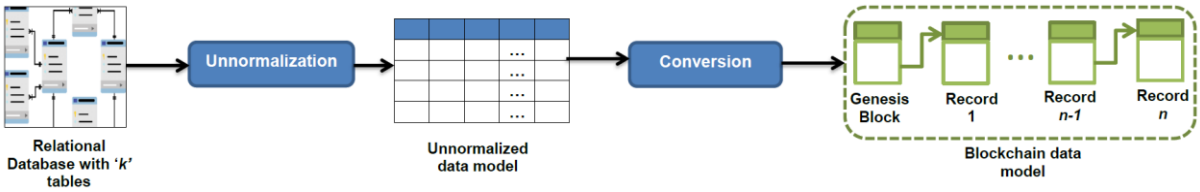

Figure 2: An abstract view of design principles for transforming relational data model to blockchain data model

Designing principles for transforming relational data model to blockchain data model

Figure 2 illustrates the abstract view of design principles for transforming the relational data model to the blockchain data model. We introduce a couple of abstract guiding principles to convert relational model to blockchain data model as follows:

- **Unnormalization of relational data model**: In this step, a database of relational data models with multiple data tables to be unnormalized for generating a single table. The unnormalized form is a simple data model that contains data redundancies. The records in the unnormalized database or single table should be stored in an ordered fashion based on the time of creation. Additionally, each record should provide a unique ID in a single table.
- **Conversion from unnormalized data model to blockchain data structure**: Each record in the single table is considered as a transaction. The unique ID and timestamp data of a record are used to convert the record to a blockchain transaction and added to a block. Finally, a

blockchain is formed. Next, the blockchain is distributed among the nodes in blockchain networks.

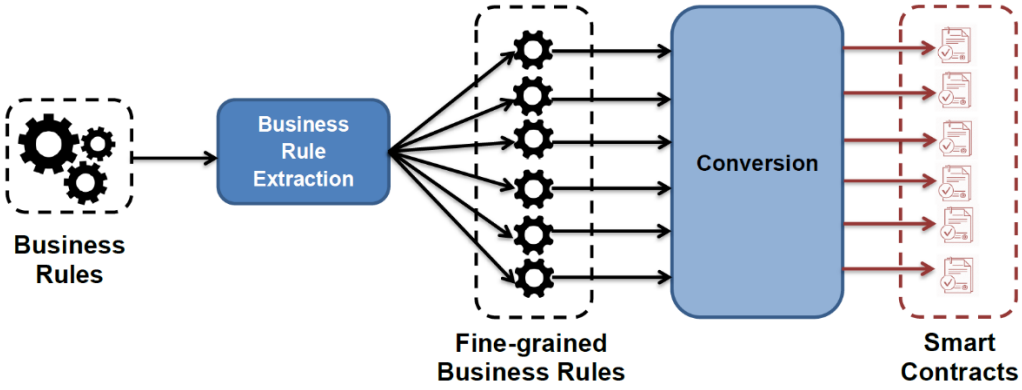

Figure 3: An abstract view of design principles for transforming business rules to smart contracts

Designing principles for transforming business rules to smart contracts

We present a set of principles for transforming business rules to smart contracts in this part. Figure 3 shows an abstract view of design principles for transforming business rules to smart contracts. The design principles are stated below:

- **Business rules extraction**: In this step, the business rules of a stakeholder are analyzed and extracted to form fine-grained business rules. For instance, business rules of the local system of a stakeholder are a combination of multiple conditions. A business rule may be coarse-grained or fine-grained. The business rule extraction process generates multiple fine-grained business rules where each fine-grained business rule deals with only one independent condition.
- **Conversion from fine-grained business rules to smart contract**: The conversion process takes a fine-grained business rule as input and produces a smart contract as output. This conversion process may be automatic or manual. Finally, the smart contracts are replicated among the nodes in blockchain networks.

Scalability and Analytics

Scalability is an important aspect that needs to be considered during the design phase. The scalability depends on the type of blockchain network that is used for deployment. For example, a consortium blockchain can be configured to be more scalable. To make it clear, an optimal number of participants in the transaction verification process can be determined to improve the performance. Scalability can also be improved by reducing the amount of data included in a transaction. Data can be divided into two types [7]: *on-chain* and *off-chain*. The on-chain data contains items that are required for validating a transaction and should be stored on the blockchain. On the other hand, the off-chain data contains raw data such as transaction-related attributes, documents, images, and smart contract addresses. A cloud-based data storage system can be used for storing off-chain data. Analytics on blockchain transactions can play an important role in determining on-chain and off-chain data items. Though blockchain data analytics is used to determine scams in cryptocurrency [8], it can also be used for dynamically determining on-chain and off-chain data items based on the system scope.

Conclusion

Similar to other systems development processes, such as service systems or cloud-based systems, the blockchain-based system development process requires a set of guiding principles. The migration of blockchain-based system from an existing system is difficult as the blockchain-based systems use cryptography heavily for providing traceability and immutability. Therefore, a set of sound design principles are necessary when migrating to the blockchain system from the traditional system. Our proposed abstract design principles would guide the blockchain system designers and ease the migration process.

Acknowledgement

This work is part of the NPRP11S-1227-170135 project. The authors would like to express their gratitude to the QNRF (Qatar Foundation) for its support and funding for the project activities.

References

[1]. M. Swan, "*Blockchain: Blueprint for a New Economy"*, O'Reilly Media, 2015.

[2] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[2] S. Omohundro, "Cryptocurrencies, smart contracts, and artificial intelligence", AI Matters, vol. 1, no. 2, pp. 19-21, 2014.

[3] Rahman, M. S., Khalil, I., Mahawaga Arachchige, P. C., Bouras, A., & Yi, X., "*A Novel Architecture for Tamper Proof Electronic Health Record Management System using Blockchain Wrapper"*. In Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure (pp. 97-105). 2019.

[4] Delmolino K., Arnett M., Kosba A., Miller A., Shi E. (2016) Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. In: Clark J., Meiklejohn S., Ryan P., Wallach D., Brenner M., Rohloff K. (eds) Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science, vol 9604. Springer, Berlin, Heidelberg

[5] Codd, E. F., "*A relational model of data for large shared data banks*". Communications of the ACM, 13, 377-387, 1970.

[6] M. Armbrust, A. Fox, R. Griffith et al., "*A view of cloud computing*", Commun. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[7] Eberhardt J., Tai S. (2017) On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In: De Paoli F., Schulte S., Broch Johnsen E. (eds) Service-Oriented and Cloud Computing. ESOCC 2017. Lecture Notes in Computer Science, vol 10465. Springer, Cham.

[8] Spagnuolo, M., Maggi, F., & Zanero, S. (2014, March). Bitiodine: Extracting intelligence from the bitcoin network. In International Conference on Financial Cryptography and Data Security (pp. 457-468). Springer, Berlin, Heidelberg.

Mohammad Saidur Rahman is a Ph.D candidate in Computer Science at RMIT University. His research interests are in privacy enhancing technologies, cryptography, and blockchain. His recent research interests include integration of traditional systems with blockchain.

Ibrahim Khalil is an associate professor in School of Computer Science and IT, RMIT University, and Melbourne, Australia. Ibrahim obtained his Ph.D. in 2003 from the University of Berne in Switzerland. Before joining RMIT University Ibrahim also worked for EPFL and University of Berne in Switzerland, Osaka University in Japan, and Silicon Valley based companies. His research interests are in scalable efficient computing in distributed systems, privacy enhancing technologies, cryptography, and blockchain.

Abdelaziz Bouras is a Professor with the Computer Science and Engineering Department, Qatar University, Doha, Qatar, which he joined as a Chair of the ictQatar (Ministry). He was a Professor with the University of Lyon (France) where he leaded a research team on Information and Decision Systems. His current research interests focus on blockchain, distributed systems for lifecycle engineering, including ontologies and fuzzy approaches for lifecycle modeling and intelligent products.

Mohammed Atiquzzaman received the MS and PhD degrees in electrical engineering and electronics from the University of Manchester, United Kingdom. He currently holds the Edith Kinney Gaylord Presidential professorship in the School of Computer Science at the University of Oklahoma. He is the editor-in-chief of Journal of Networks and Computer Applications, founding editor-in-chief of Vehicular Communications and has served/serving on the editorial boards of various IEEE journals and co-chaired numerous IEEE international conferences including IEEE Globecom. His research interests are in communications switching, transport protocols, wireless and mobile networks, satellite networks, and optical communications. He is a senior member of the IEEE.