DESIGNING CONTINUOUS AUDIT ANALYTICS AND FRAUD PREVENTION SYSTEMS USING EMERGING TECHNOLOGIES

by

YUNSEN WANG

A Dissertation submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

In partial fulfillment of requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

written under the direction of

Dr. Alexander Kogan

and approved by

____

____

____

____

Newark, New Jersey

May, 2018

ABSTRACT OF THE DISSERTATION

Designing Continuous Audit Analytics and Fraud Prevention Systems Using Emerging Technologies

By YUNSEN WANG

Dissertation Director:
Dr. Alexander Kogan

This dissertation consists of three essays that design and evaluate the continuous audit analytics and fraud prevention systems using three emerging technologies (i.e., the blockchain, in-memory cloud computing, and deep learning). The first essay designs a framework of Blockchain-based Transaction Processing System using the homomorphic encryption and zero-knowledge proof mechanisms. Furthermore, this study develops a prototype of the designed system to demonstrate its applications in real-time accounting, continuous monitoring, and fraud prevention. Although the simulation tests show the Blockchain-based Transaction Processing System consumes more computational overhead than the conventional database-based ERP system, the blockchain should be considered as a promising technology for future accounting and auditing practice.

The second essay introduces the database architecture that manages data in main physical memory and columnar format. This essay proposes a conceptual framework for applying the in-memory columnar database system to support high-speed continuous audit analytics. Moreover, this study develops a prototype and conducts the simulation tests to evaluate the proposed framework. The test results show the high efficiency and effectiveness of the in-memory columnar database relative to the conventional ERP system regarding the computational time and the storage volume. Furthermore, the deployment of the in-memory columnar database to the cloud shows great promise of applying the in-memory columnar database for continuous audit analytics.

The third essay designs a continuous fraud detection system based on modified deep learning technology. Specifically, this essay builds an accounting layer on top of the deep learning architecture to process financial data for predicting the fraudulent financial statements. A prototype is developed to evaluate the prediction accuracy of the proposed design. The test results show the deep learning-based continuous fraud detection system provides high prediction accuracy relative to the existing studies of financial statement fraud detection.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Dr. Alexander Kogan, who has provided strong encouragement, valuable guidance, and full trust all along my doctoral journey. You will always be the mentor that leads me to become a serious researcher and a finer person. I would also like to express my sincere gratitude to Dr. Miklos A. Vasarhelyi, who has given me continuous support, warm caring, and insightful advice. Your academic spirit will always inspire me to become a better scholar. I am very grateful to Dr. Hussein Issa. Your constructive feedback and suggestions have perfected my research work over the years. I would also like to express my sincere gratitude to Dr. Graham Gal, your insightful comments and valuable support have always been beneficial to my dissertation and helped improve it in many aspects. Also, I would like to give special thanks to Dr. Dan Palmon for your continuous support and helpful suggestions that have motivated me in becoming a better accounting educator.

I am indebted to my dear parents, Zhiguo Wang and Maobin Zhu, and my lovely wife, Tiffany Chiu. Your unconditional love, whole-hearted support, and continuous faith in me have enabled me to discover my potential, achieve my goals, and fulfill my dream. From the bottom of my heart, thank you all for making my achievement possible.

I owe thanks to my dearest colleagues and friends, especially to Victoria Chiu, Feiqi Huang, He Li, Yue Liu, Jun Dai, Pei Li, Qi Liu, Ting Sun, Yan Li, Zhaokai Yan, Abdulrahman Alrefai, and Paul Byrnes, your warm friendship and academic support will always be cherished.

This dissertation would not have been possible without the help and support of my dissertation advisor, my committee members, my friends, and my family. Thank you all for standing by me throughout this journey!

TABLE OF CONTENTS

DESIGNING CONTINUOUS AUDIT AND FRAUD PREVENTION SYSTEMS USING EMERGING TECHNOLOGIES

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

The emerging technologies are changing today's business environment, re-engineering business processes, and redefining numerous aspects of accounting and auditing procedures. They are bringing a new wave of upgrades for continuous auditing (CA) research and practice. The CA is defined as "*a methodology for issuing audit reports simultaneously with, or a short period after, the occurrence of the relevant events*" (Kogan et al. 1999). It has been a critical element in internal audit practice (Alles et al. 2006) and the key component of overall corporate governance (Kuhn and Sutton 2010). Due to the increased frequency of periodic audits, CA not only automates the tests of details and analytical procedures but also provides the continuous assurance and continuous monitoring (CM) for financial reporting and internal controls.

Traditional CA systems utilize two different approaches: the embedded audit modules (Groomer and Murthy 1989) and the control and monitoring layer (Vasarhelyi and Halper 1991). The approach of embedded audit modules inserts the CA functions in the auditee's enterprise information systems, while the control and monitoring layer extracts the auditee's data to be processed in the auditor's systems. The extant research of CA has focused on the algorithms and implementations of the rule-based CA systems; however, few studies have been conducted on designing the intelligence-based CA systems using recent technology innovations. This dissertation thesis contributes to the CA literature by proposing an integrated architecture for continuous audit analytics and fraud prevention consisting of three system components based on three emerging technologies (i.e., the blockchain, in-memory columnar computing and deep learning).

The blockchain is one of the most disruptive and promising emerging technologies, and it appears to have the potential for significantly affecting the accounting and auditing fields. Essentially, blockchain is a freely open and publicly shared database that keeps track of transactions and protects data from tampering (Lansiti and Lakhani 2017; Yermack 2017; Dai and Vasarhelyi 2017). Once a transaction is committed, it is practically irreversible and immutable (Nakamoto 2008). Blockchain technology provides a method to share a database among the participants even if they do not trust each other, and it creates a marketplace to transfer assets based on a peer-to-peer network without a central authority. However, one of the challenges impeding the adoption of blockchain is that firm's managers are concerned about the financial privacy and business secrets because all participants in a public blockchain have a full copy of every transaction. This concern led to the development of private blockchains in which only authorized parties can read records and create transactions. Although a private blockchain provides a relatively closed, secure business environment, it loses data transparency and public participation, which could limit the function of resisting tampering because the managers have full control over the private blockchain. Therefore, the private blockchain's immutability cannot be guaranteed if management can retroactively manipulate the transaction data for personal gain.

To apply blockchain for accounting and auditing and preserve its privacy and confidentiality, the first essay proposes a framework design - a blockchain-based accounting information system (Bb-TPS) - using zero-knowledge proof (ZKP). The ZKP is a cryptographic method by which one party can prove to the other parties that the initiated transaction is valid without releasing any sensitive information. For example, a transaction initiator can prove to a transaction verifier that his/her transaction is valid without releasing the identity of the trading partners and transaction amounts. Besides ZKP, this essay shows how to configure homomorphic encryption (an encryption algorithm allowing computations to be done on encrypted data) and permission-management schema in Bb-TPS. In a nutshell, this proposed system can provide real-time accounting and continuous monitoring services, prevent transaction fraud and deliver guaranteed privacy protection.

In the era of big data, audit profession is starting to leverage the emerging data analytic techniques (e.g., deep learning, process mining) to examine financial data, evaluate internal control effectiveness, and detect fraudulent transactions. To apply audit analytic techniques to examine an auditee's business data, an auditor needs to extract the full population of transactions periodically (e.g., purchase orders, invoice, receipts) from the client's Enterprise Resource Planning (ERP) system. The higher the frequency of the data access is, the timelier the financial and audit report will be; however, the more computing and communication resources it will consume (Pathak, Chaouch, and Sriram 2005). An alternative solution is to embed audit modules in ERP systems (Groomer and Murthy 1989); however, the queries that select data from the operational database and perform complex audit analytics can easily overload ERP systems and disrupt the regular transaction processing (Chaudhuri and Dayal 1997). Therefore, to conduct real-time and continuous audit analytics using computationally costly artificial intelligence (AI) algorithms, it is necessary to build a high-speed and high-volume data processing infrastructure to support continuous audit analytics.

The in-memory columnar database system is such an infrastructure that supports high-speed data analytics using main memory as the primary storage (Garcia-Molina and Salem 1992; Plattner 2009). The second essay introduces the architecture of the modern in-memory columnar database system and proposes a design of applying the new database for high-speed continuous audit analytics. Furthermore, it conducts a simulation test to measure the computational overhead in comparison with a conventional relational database system.

Financial statement fraud could lead to severe consequences for companies and auditors. For example, the Securities and Exchange Commission (SEC) would file charges against the fraud companies and their top executives with misleading investors, and then the executives could face punishment by jail, fines, or probation. The auditors who engaged in financial statement fraud may also face fines and damaged reputations. Therefore, if a company realizes that it filed a materially misleading financial statement, it has to restate the previous financial statements immediately. The financial misstatement may be caused by unintentional cleric errors or intentional earnings manipulation, that is, financial statement fraud.

In order to help auditors and management detect financial statement fraud and reduce fraud risk in a real-time and continuous manner, this study proposes a continuous fraud detection system to identify a company's abnormal financial performance using deep learning algorithms. The proposed system would be able to predict whether a financial statement engages in fraud schemes and further predict the possibility of a specific type of fraud. Based on the prior research on financial statement fraud detection, this study designs and prototypes the Deep Learning-based Continuous Fraud Detection System (DL-based CFDS) using the fraud and nonfraud sample during 1992 and 2012. It uses the fraud and nonfraud sample during 2012 and 2016 to validate the prediction accuracy of the DL-based CFDS. Moreover, instead of constructing a large number of complex prediction variables (e.g., ratios, accrual-family variables), this study designs an accounting layer that transfers the account balance values to log values. The deep neural networks can intelligently generate various input variables in the first layer and further process the inputs to the hidden layers. The evaluation of a prototype shows that the DL-based CDFS achieves high prediction accuracy relative to existing financial statement fraud detection methods, but didn't find that further partition of fraud type could improve prediction accuracy.

This dissertation consists of three essays that design and evaluate a comprehensive framework of continuous audit analytics and fraud prevention system using three emerging technologies. The first essay creates a blockchain-based business ecosystem, whereby the second essay builds enterprise information systems using the in-memory columnar database architecture for individual organizations. The third essay aggregates financial data from the new enterprise information systems to perform financial statement fraud prediction using deep learning algorithms. The first essay employs the blockchain technology to propose an infrastructure for continuous assurance of transaction data (transaction level). The second essay uses cloud-based in-memory computing system to support continuous data processing and aggregating (balance level). The third essay leverages deep learning algorithms for continuous fraud detection and prevention (reporting level). Among the six chapters of the thesis, chapter one introduces the background of continuous auditing and emerging technologies. Chapter two provides prior literature on continuous auditing, financial statement fraud as well as the three emerging technologies (i.e., the blockchain, cloud-based in-memory database, and deep learning).. The three essays are included in chapter three, four and five, respectively. The last chapter concludes the dissertation by providing a summary of findings and future research implications.

In summary, this dissertation contributes to the accounting literature by proposing a comprehensive architecture of continuous audit analytics that consists of three system layers using three cutting-edge emerging technologies (i.e., the blockchain, cloud-based in-memory computing, and deep learning). The Blockchain-based Transaction Processing System is designed and created to support the continuous test of management assertions on the transaction level. Cloud-based In-Memory Columnar Database Architecture is proposed and evaluated to support the continuous data aggregation and analytics to test the management assertions on the account balance level. The Deep Learning-based Continuous Fraud Detection Systems is designed and prototyped to support continuous financial statement fraud detection on the financial reporting level. The objective of the proposed architecture is to upgrade the traditional rule-based continues auditing systems to intelligence-based continuous audit analytics systems.

Chapter 2: Literature Review

The CA is defined as "*a methodology for issuing audit reports simultaneously with, or a short period of time after, the occurrence of the relevant events*" (Kogan et al. 1999). Due to the increased frequency of periodic audits, CA not only automates the test of details and analytical procedures but also provides the continuous assurance and continuous monitoring (CM) for financial reporting and internal controls. This section will review the prior literature on continuous auditing and financial statement fraud detection as well as the recent innovations of emerging technologies.

1.1 Continuous Auditing

Groomer and Murthy (1989) and Vasarhelyi and Halper (1991) pioneered the two database-driven approaches to design the architectures of continuous auditing (CA) systems: the embedded audit modules (EAM) and the control and monitoring layer (CML), respectively. The main difference between EAM and CML is the approach to configure the CA functions: EAM inserts the CA functions in the auditee's enterprise information systems and CML extracts the auditee's data and processes the data using CA functions in the auditor's systems. The CA functions will identify a transaction as an exception if it violates some predefined business rules and triggers an automatic alarm to the auditor. Since the early works of Groomer and Murthy (1989) and Vasarhelyi and Halper (1991), the research field of continuous auditing has flourished with numerous advances, ranging from the architectural aspects of CA (Kogan et al. 1999; Woodroof and Searcy 2001; Murthy and Groomer 2004; Kuhn and Sutton 2010) to the practical implementation of EAM in ERP (Debreceny et al. 2005) and the pilot implementations of CA and continuous monitoring (CM) in Siemens and HSP (Alles et al. 2006; Alles et al. 2008). The feasibilities and economics of CA have been examined in prior studies (Alles et al. 2002; Alles et al. 2004).

Kogan et al. (2014) designed architecture of continuous data level auditing systems based upon continuity equations. In the discussion of detecting business process irregularities, they introduced a clear distinction between "exceptions" and "anomalies." The "exceptions" are defined as "*those [detected irregularities] that are violations of deterministic business process rules*" and the "anomalies" as "*those [detected irregularities] that are significant statistical deviations from the steady state of business process behavior*." Based on a proprietary dataset from the internal audit department of a multinational consumer products company, Issa and Kogan (2014) employed an ordered logistic model to develop a method to identify and prioritize anomalies for further investigation.

The mainstream of the prior literature has been focused on the mechanisms, implementations, and feasibilities of CA; however, few studies have been conducted on supporting to perform continuous data analytics and artificial intelligence algorithms in an automatic and real-time manner. This dissertation thesis contributes to the CA literature by proposing a highly integrated architecture of continuous audit analytics and fraud prevention based on three emerging technologies (i.e., the blockchain, cloud in-memory computing and deep learning). The proposed architecture consists of three system components that provide continuous assurance for the financial data in transaction, balance and reporting levels, respectively.

1.2 Financial Statements Fraud Detection

Accounting research on financial statement fraud and Accounting and Auditing Enforcement Releases (AAERs) includes testing hypotheses grounded in the literature of earnings management (Summers and Sweeney, 1998; Beneish, 1999; Sharma, 2004) and corporate governance (e.g., Beasley, 1996). The early research of financial statement fraud dates back to 1980s (Elliott and Willingham, 1980). Feroz, Park, and Pastena Feroz et al. (1991) documented the AAERs affecting the stock price. Beasley Beasley (1996) examined the association between the board of the director composition and financial statement fraud. With fewer proportions of outside members on the board of directors supervising a firm's management (Beasley, 1996), it is more likely that the management uses discretion to manage the firm's accruals and earnings, or even aggressively commits to financial statement fraud.

Therefore, numerous measures for earnings management are created to indicate the risk of financial misstatement and fraud, such as earnings persistence (e.g., Richardson et al., 2005), abnormal accruals and accruals models (e.g., Jones, 1991; Dechow et al., 1995; Dechow and Dichev, 2002; Kothari et al., 2005), and earnings smoothness (e.g., McInnis, 2010). Beneish (1999) matched the sample of fraud to non-fraud by SIC code and year and created an index consisting of seven ratios to indicate the likelihood of an earnings overstatement. Dechow et al. (2011) applied predictors identified in the prior literature (e.g., accrual quality variables, financial ratios, employment and order backlog, and stock price related variables) and developed a measure, the F-score, to assess the risk of financial misstatement and corporate fraud. In order to add more information for predicting fraud risk, Brazel et al. (2009) examined nonfinancial measures (e.g., facilities growth) and suggested that these measures could be used to predict financial statement fraud. In order to evaluate the predictive power of the extent accrual-based earnings management measures to detect financial statement fraud, Jones et al. (2008) conducted an empirical analysis comparing ten measures (e.g., discretionary accruals, accrual quality) derived from popular accrual models and found that only the accrual estimation errors (Dechow and Dichev, 2002) and their modifications have the ability to predict fraud and non-fraudulent restatements of earnings.

Another stream of financial statement fraud detection research is grounded in the literature of data mining and machine learning (e.g., Green and Choi 1997; Cecchini et al. 2010; Perols 2011; Perols et al. 2015). Early work by Green and Choi (1997) develops a financial statement fraud detection model using a neural network classifier that performs relatively well. The more recent research employs other additional classification techniques, such as Support Vector Machine, Logistic Regression and many other machine learning ensemble algorithms (Cecchini et al. 2010; Perols 2011), which improves the performance of fraud prediction. Perols (2011) uses six statistical and machine learning models in detecting financial statement fraud and shows that logistic regression and support vector machine perform well relative to an artificial neural network. In addition to financial variables, text-mining techniques are used to detect financial statement fraud. Humpherys et al. (2011) extracts MD&A textual data from 10-Ks and uses Naïve Bayes and decision tree algorithms to identify fraudulent financial statement. To solve the problems of fraud data rarity and large dimensionality, Perols et al. (2015) develop three data preprocessing methods (i.e., observation under-sampling, variable under-sampling and fraud type partition) to improve prediction performance of the best current fraud classification techniques. The result shows that peer firms matching and misstatement prediction based on different types could improve prediction accuracy (Perols et al. 2015).

1.3 Emerging Technologies

1.3.1 Blockchain

Bitcoin is a peer-to-peer electronic currency system first introduced by Satoshi Nakamoto in 2008. Based on cryptographic algorithms (e.g., digital signature and hash function), Bitcoin uses transaction history to prove ownership and public confirmation to prevent double spending. The transaction history is recorded into an ongoing chain of blocks and shared to all users along the peer-to-peer network (Nakamoto 2008). Each of the recent transactions is publicly announced and confirmed; then the confirmed transactions are written into a new block to be added to the end of the blockchain. As it is practically impossible to rebuild the entire blockchain by one or several dishonest users, the immutable blockchain will serve as the proof of all transactions included in the blockchain.

Besides Bitcoin, many cryptocurrencies follow the similar protocol of blockchain and add new features to distinguish themselves from the others, such as Ethereum, Litecoin, and Zcash. In the crowded virtual currency market, blockchain is the core technology that supports exchange and circulation. Ethereum is not only a kind of cryptocurrency but also a platform for running smart contracts encoded in blockchain (Buterin 2014). A smart contract is a computerized self-executing protocol that enforces the execution of a predefined contract in a real-time manner (Szabo 1994). Before the emergence of the blockchain, a smart contract relied on a trusted third party to program the terms into a contract. The Ethereum project enables all users to deploy and use smart contracts in a decentralized business network. Zcash is a privacy-preserving cryptocurrency initiated by Zerocoin Electric Coin Firm (ZECC) to protect the user's private information. Zcash protocol employs a zero-knowledge proof (ZKP) schema to enable the public to verify a transaction without having to know the details and therefore prevents sensitive data exposure. Using the same ZKP schema, Kosba et al. (2016) create a privacy-preserving protocol, named Hawk, to protect private information for smart contract users.

1.3.2 Cloud-based In-memory Columnar Computing

With the recent breakthroughs in hardware technology and cloud computing (Mell and Grance 2011), the abundant availability of main memory and wide-bandwidth network allow storing big data in the primary storage for fast access by processors. Moreover, new storage products, such as solid-state drives (SSD), provide faster and more reliable alternatives for secondary storage. In a conventional DBMS, data resides permanently in hard disk and will be loaded into main memory when needed, while in the modern in-memory database system (IMDB), data resides permanently in main physical memory. As multi-core CPUs can directly access data in main memory, IMDB has a better response time and transaction throughputs (Plattner 2009).

In contrast to conventional row-oriented storage, the values for each attribute are stored contiguously in the column-oriented storage; therefore, its compression efficiency is usually 4 to 5 times that of row-oriented storage (Abadi, Madden and Ferreira 2006). Moreover, a complicated analytical query could be fast responded to as data aggregation in columnar storage outperforms row-oriented storage, especially in the case of a large number of data items. By applying columnar storage schemas to IMDB, the new in-memory columnar database would be superior to row-oriented IMDB with regards to memory consumption.

1.3.3 Deep learning

The early research of Artificial Neural Network (ANN) dates back at least to the 1940s (McCulloch and Pitts 1943). Neuroscientists build models of human brain's neural network to understand nervous activity, which was then adopted by computer engineers to create better computer systems. The essential elements of ANNs are neurons (or units), their connections and the weights that are assigned to the connections.

In general, an ANN model consists of many layers, and each layer is a string of neurons. Any ANN model has at least two layers – the input layer and output layer. The input layer takes the input variables, and the output layer provides prediction results. All layers between the input layer and the output layer are called hidden layers. An ANN model with a single hidden layer. In this model, x_i , y_j and z represent neurons, $X = (x_1, x_2 \dots x_i)$, $Y = (y_1, y_2 \dots\ y_j)$ and Z represent input layer, hidden layer and output layer, respectively. Neurons in input layer are fully connected with neurons in the hidden layer, and neurons in the hidden layer are fully connected with neurons in output layer. W_{ij} and V_j represent the weights of these connections. In the early and simple ANNs, data move in one direction forward from input layers through the hidden layer and to the output layer, which is called multilayer feedforward network (Hornik et al. 1989). When input data are fed to the input layer, each neuron (e.g., x_1) of the input layer performs a doc product with the input data and the corresponding weight (e.g., W_{11}), adds the bias (e.g., b), applies an activation function (e.g., ReLu) and passes the result to the next layer. All multilayer feedforward networks need activation functions (Hornik et al. 1989; Leshno et al. 1993) to approximate any functions in the computer systems.

Based on conventional ANNs, deep neural networks (DNNs) are designed using modern Graphic Processing United (GPU)-based computers to perform more sophisticated tasks, such as computer vision, speech recognition, and natural language processing. The weights and biases are critical parameters to determine a proper DNN. To accurately assign weights and biases across layers is a crucial objective for building a DNN (Schmidhuber 2015). Backpropagation is the most widely used algorithm for training ANNs and DNNs (Riedmiller and Braun 1993). In supervised learning (Møller 1993), a training dataset has both inputs and outputs. After feeding inputs and receiving the predicted outputs, the difference between predicted outputs and the given outputs could be used to design an error function. To minimize the error, DNNs repeatedly compute the influence of each weight on the error function and adjust each weight through stochastic gradient descent (Riedmiller and Braun 1993; Bottou 2010). Backpropagation stops until convergence is reached or errors have been optimally minimized.

To adjust the fully connected ANNs for processing image data, the computer scientists simplify the ANN architecture by eliminating many unnecessary connections within layers; instead, they create filters to collect local features of an image. DL is designed to process images represented by pixels in red, blue, and green three dimensions. Through the layers of convolution and pooling, the data of an image are transformed to the outcome or the label value. The CNN provides with opportunities to analyze data in significant volume and dimensions. Comparing to traditional ANNs, the DNNs bring better capacity and higher efficiency to train a machine, which doesn’t require the system designers to have relevant domain knowledge before training a machine. The DNN has many new features, such as computer vision and machine memory, and it even allows a machine to play games with itself. Those features are achieved by Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Reinforcement Learning (RL) for computer vision, machine memory, and self-playing mechanism, respectively.

An RNN is developed for sequential data, such as text, speech, and video (Pineda 1987; Lukoševičius and Jaeger 2009). It adds a circuit from the hidden layer's output to its input. In this case, data at time t_1 moves from the input layer to the output layer through the hidden layer and come back to the hidden layer by concatenating data at time t_2. The basic structure of a standard RNN, an unfolded RNN and an RNN example – Long Short-Term Memory Model (LSTM) (Gers et al., 2000). A CNN is developed for image recognition and computer vision (Lawrence et al. 1997; Krizhevsky et al. 2012). The architecture of CNN, which provides two different layers - the Convolution layer, and the Pool layer, to collect the feature of the image and shrink the feature size, respectively. Thanks to the GPU implementation which is efficient at matrix and vector multiplications, it can speed up the learning rate of CNN by at least a factor of 50 (Schmidhuber 2015). The combination of DNN and reinforcement learning (RL) creates machine's another ability that it can automatically play games and even learn "shortcut" from experience. Based on an objective function Q-value function and dynamic programming the machine follows existent rules, takes actions, and gets a reward from the environment (Sutton and Barto 1998; Mnih et al. 2015). To solve the problem of overfitting, Srivastava et al. (2014) provides a simple way that removes neurons from the layers during training to improve model generalization

Chapter 3: Designing Confidentiality-Preserving Blockchain-based Transaction Processing Systems

3.1 Introduction

The topics of continuous financial disclosure and publicly shared databases have been discussed ever since the 1970s (Pastena 1979). Today's business ecosystem demands information sharing and data communication to improve trading efficiency and effectiveness. However, there is a trade-off between transparency and confidentiality: the more information is shared, the more transparent the business will be, and the more potential for business secrets (e.g., pricing strategy, trading partner information, business process details) and confidentiality to be compromised. The trade-off between information transparency and data confidentiality is one of the main tension points of today's business: the cooperation versus the competition (Bengtsson and Kock 2000).

The blockchain is one of the most disruptive and promising emerging technologies, and it appears to have the potential for significantly affecting the accounting and auditing fields. Essentially, blockchain is a freely open and publicly shared database that keeps track of transactions and protects data from tampering (Lansiti and Lakhani 2017; Yermack 2017; Dai and Vasarhelyi 2017). Once a transaction is committed, it is practically irreversible and immutable unless the majority of the blockchain users collude[1] (Nakamoto 2008). Blockchain technology provides a method to share a database among the

[1] The most infamous potential risk is known as "51% attack". In hypothetical, a group of blockchain users who control more than 50% of the network's computing power would be able to reverse the completed transactions and alter transaction history. http://www.coindesk.com/ahead-bitcoin-halving-51-attack-risks-reappear/

participants even if they do not trust each other, and it creates a marketplace to transfer assets based on a peer-to-peer[2] network without a central authority.

The faster speed, lower cost, and more accurate bookkeeping systems attract investment from venture capitalists, multi-national bankers, and attention from regulators. Nasdaq announced in December 2015 that issuers were able to make securities transactions on its private blockchain (Nasdaq 2015). Sydney Stock Exchange (SSX)'s first blockchain prototype was launched in May 2016, which is "their first step toward an instantaneous settlement-and-transfer-upon-trade" exchange platform (Rizzo 2015). Meanwhile, the exploration of blockchain applications by audit firms could improve audit efficiency and effectiveness (PwC 2016, Deloitte 2016, EY 2016, KPMG 2016). The convergence of accounting and blockchain technology shows great promise for reducing redundant manual effort, increasing the speed of transaction settlement, and preventing financial reporting fraud. Furthermore, it could drastically change the way of corporate finance and governance just as the 1933 and 1934 Securities and Exchange Acts did (Yermack 2017).

However, one of the challenges impeding the adoption of blockchain is that firm's managers are concerned about their financial confidentiality and business secrets because all participants in a public blockchain have a full copy of every transaction. This concern led to the development of private blockchains in which only permitted parties[3] can read records and create transactions. Although a private blockchain provides a relatively closed, secure business environment, it sacrifices data transparency and public participation, which could limit its tamper resistance because the managers have full control over the private

[2] A peer-to-peer system is a network that allows data to be sent from one participant to another without going through a central authority.

[3] For example, if an organization or a select number of organizations own a private blockchain, only the employees within these organizations are allowed to participate in the blockchain transactions.

blockchain. Therefore, the tamper resistance of private blockchain cannot be guaranteed if management can manipulate the transaction data for personal gain retroactively.

There is a trade-off between information transparency and data confidentiality such that people choose to use a private blockchain regardless of majority consensus[4] or use a public blockchain under the risk of a confidentiality breach. The more nodes[5] are added to a network, the more reliable the data are, and the less confidential the blockchain is. To apply blockchain for accounting and auditing and preserve its confidentiality, this study proposes a framework design - a Blockchain-based Transaction Processing System (Bb-TPS) - using zero-knowledge proof (ZKP). The ZKP is a cryptographic method by which one party can prove to the other parties that the initiated transaction is valid without releasing any sensitive information. For example, a transaction initiator can prove to the transaction a verifier that his/her transaction is valid without releasing the identity of the trading partners and transaction amounts. Besides ZKP, this paper shows how to configure homomorphic encryption[6] (Gentry 2009) and permission-management schema in Bb-TPS. In a nutshell, this proposed system can provide real-time accounting and continuous monitoring services, prevent transaction fraud, and deliver guaranteed confidentiality protection.

The remainder of the study is organized as follows. Section 2 introduces the blockchain and zero-knowledge proofs, Section 3 proposes a framework of applying blockchain technology to design real-time accounting and continuous monitoring systems,

[4] Majority consensus should be reached in order to build an irreversible and immutable blockchain.

[5] In the peer-to-peer network system, every participant using a computer to access the network is called a node.

[6] Homomorphic encryption is an encryption algorithm that allows complex mathematical operations to be done on encrypted data.

Section 4 provides a prototype of the framework and evaluates the performance, and Section 5 concludes the paper and discusses future studies.

3.2 Motivation and Literature Review

3.2.1 The Dilemma: Transparency and Confidentiality

The objective of Nakamoto's (2008) Bitcoin protocol is to create an online payment system without needing a trusted central authority to prevent fraudulent transactions. Based on peer-to-peer network architecture, this protocol allows all users to get involved in updating (i.e., initiating a new transaction) and maintaining (i.e., mining a new block) the shared database (i.e., blockchain). Therefore, all users have access to all transaction details including sender, recipient and amount. Although the Bitcoin protocol uses cryptographic algorithms (e.g., hash function) to anonymize a user's information, it is still vulnerable to privacy attacks. While the direct disclosure of crypto-wallets' personally identifiable information (PII) is not harmful since the information is sanitized, the transactional level details could allow inferences to be made (Gal 2008).

"*The unprecedented transparency of transactions sits uneasily with the privacy needs …*" because business would favor information security and privacy (Shubber 2016). For example, if a firm voluntarily discloses all the transactions on the blockchain, its rivals have the chance to spy on the firm's activities and steal its business secrets. The objective of adopting blockchain is to reduce the cost of information integrity protection and increase the speed of transaction settlement. However, public disclosure of all transactions is too much of a security and privacy risk. Therefore, some firms want to deploy the blockchain protocol within a secure and closed network, which is the so-called private blockchain. A private blockchain is based on the blockchain protocol that allows only permitted parties to have access to all the transactions (Yermack 2017)[7].

However, there will inevitably be a central authority that maintains the private blockchain and manages the permissions of participants, which concentrates the operational risk in a single or several points of failure and loses the primary function of blockchain – decentralization. More severely, if a dishonest manager is in charge of the central authority, s/he is capable of retroactively manipulating the private blockchain for personal gain. The irreversibility and tamper resistance could not be guaranteed by private blockchain if the central authority is corrupt.

The dilemma of adopting blockchain in accounting and auditing is to find the trade-off between information transparency and confidentiality. With more participants in a blockchain, more business data would be publicly shared; however, business confidentiality and secrets would be protected to a lesser extent. The purpose of this paper is to find a solution to enabling the adoption of blockchain for accounting practice and ensuring only permitted parties (e.g., auditors and regulators) can view the transaction data.

3.2.2 Bitcoin and Smart Contracts

Bitcoin is a peer-to-peer electronic currency system first introduced by Satoshi Nakamoto in 2008. Based on cryptographic algorithms (e.g., digital signature[8] and hash function[9]), Bitcoin uses transaction history to prove ownership and public confirmation to prevent double spending. The transaction history is recorded into an ongoing chain of

[7] Alternatively, instead of using private blockchains, firms may share certain information on the public blockchain relying on the standard encryption procedures and public key infrastructure to protect confidentiality.

[8] Digital signature is a cryptographic scheme for demonstrating the authenticity of digital messages.

[9] Hash function is a function with special properties that maps data of arbitrary size to that of fixed size.

blocks and shared with all users along the peer-to-peer network (Nakamoto 2008). Each of the recent transactions is publicly announced and confirmed; then the confirmed transactions are written into a new block to be added to the end of the blockchain. As it is practically impossible to rebuild the entire blockchain by one or several dishonest users, the immutable blockchain will serve as the proof of all transactions included in the blockchain.

In Jan 2009, Satoshi Nakamoto mined the first 50 Bitcoins and sent 10 Bitcoins to a developer, Hal Finney (Peterson 2014), and by February 2017 there were 16 million Bitcoins in circulation among an estimated 510,000 users[10]. Besides Bitcoin, there are many cryptocurrencies that follow a similar protocol of blockchain while adding new features to distinguish themselves from the others, such as, Ethereum[11], Litecoin[12] and Zcash[13]. In the crowded virtual currency market[14], blockchain is the core technology that supports exchange and circulation.

Ethereum is not only a kind of cryptocurrency but also a platform for running smart contracts encoded in blockchain (Buterin 2014). A smart contract is a computerized self-executing protocol that enforces the execution of a predefined contract in a real-time manner (Szabo 1994). Before the emergence of the blockchain, a smart contract relied on a trusted third party to program the terms into a contract. The Ethereum project enables all users to deploy and use smart contracts in a decentralized business network. Zcash is a privacy-preserving cryptocurrency initiated by Zerocoin Electric Coin Firm (ZECC) to

[10] Data available at: https://blockchain.info/charts/total-bitcoins.
[11] https://www.ethereum.org. Accessed 5/23/17 10:20PM.
[12] https://litecoin.org. Accessed 5/23/17 10:21PM.
[13] https://z.cash. Accessed 5/23/17 10:22PM.
[14] https://coinmarketcap.com. Accessed 5/23/17 10:28PM.

protect the user's private information. Zcash protocol employs a zero-knowledge proof (ZKP) schema to enable the public to verify a transaction without having to know the details and therefore prevents sensitive data exposure. Using the same ZKP schema, Kosba et al. (2016) create a privacy-preserving protocol, named Hawk, to protect private information for smart contract users.

3.2.3 Colored Coins, Sidechains, and Private Blockchain

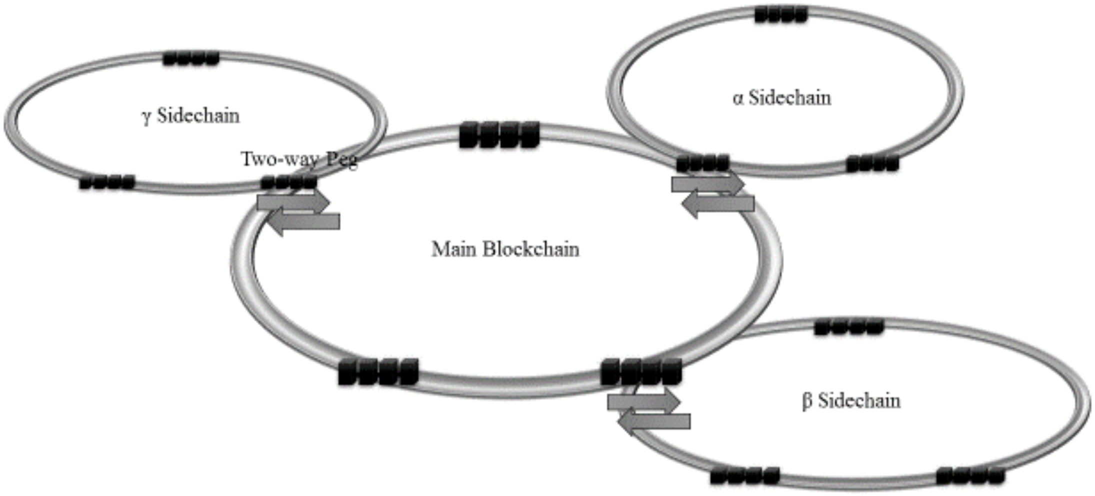
In general, every digital coin in blockchain is a token. The concept of the colored coin[15] refers to a class of digital coins for "representing and managing real-world assets" in the blockchain. A coin color dictionary could be distributed on the blockchain so that all the participants will use the same color consistently. For example, a yellow coin represents a Bitcoin or a U.S. dollar, a green coin represents raw materials, and an orange coin represents a finished product. Therefore, to map real-world business activities onto blockchain network needs a massive amount of digital coins. Besides, an increasing number of blockchains are created to support the circulation of colored coins.

How could these blockchains communicate with each other? For example, if a firm and its suppliers have different blockchains to register their assets as colored coins, how could the firm and its suppliers send each other colored coins across those blockchains? Back et al. (2014) proposes the "pegged sidechains" to enable Bitcoins and other blockchain coins to be transferred between multiple blockchains, and makes it easy for the multiple blockchains to interoperate with each other. The "pegged sidechains" technique creates a channel in which different coins can be exchanged, which gives existing users access to new blockchain protocols using the coins they already own. For example, a

[15] https://en.bitcoin.it/wiki/Colored_Coins. Accessed 5/23/17 10:30PM.

Bitcoin user can give up two Bitcoins in exchange for 60 Zcashs, and vice versa. As shown in Figure 1, there are four blockchains among which α, β, and γ Sidechains need to communicate with the Main Blockchain. For example, when a β Sidechain user wants to transfer several β coins from β Sidechain to the Main Blockchain user, s/he can send the β coins to a unique address where his/her β coins are locked up. In return, s/he would receive some Main Blockchain coins. The number of coins will be based on the conversion rate between the β coin and the Main Blockchain coin. After receiving the Main Blockchain coin, s/he can transfer the coins to another user on the Main Blockchain.

Figure 1: Blockchain and Sidechains



Source: Back et al. 2014

In technical terms, sidechains are separate from but would be interoperable with the main blockchain. Any technical failure or malicious attack on a sidechain will not affect the main blockchain, so it would provide a safe platform to design, customize and test private sidechains without affecting the main blockchain's core code. The permissions control of a private blockchain would concentrate operational risk in a single or several points of failure and "might charge monopolistic rents to network users or fail to treat them evenhandedly" (Yermack 2017). Furthermore, the private blockchains are relatively easy to be manipulated by corrupt central authorities.

3.2.4 Homomorphic Public Key Encryption

The fact that every user has a full copy of transaction history is too much of a security risk for both organizational and individual users. Blockchain data anonymization, such as encrypting or sanitizing personally identifiable information (PII)[16], is necessary to preserve a user's privacy and to stimulate participation. Encryption, such as public key cryptosystems (ElGamal 1985), is a process to encode data that only authorized parties can access. For example, if Alice wants to send secret messages to Bob, Bob needs to generate a pair of keys: a public key (to be disseminated widely) and a private key (to be kept private by Bob). Alice applies Bob's public key to encode the secret messages and send to Bob through a channel which could be intercepted by hackers. Even though the hackers extract the encrypted messages, without Bob's private key they cannot understand what Alice sends to Bob and only Bob can decrypt Alice's messages.

Homomorphic encryption is a type of encryption algorithm that allows for computations to be done on encrypted data (Gentry 2009). For example, Alice sends Bob two messages: one is a number eight, and the other is a number nine. The data is encrypted so that eight become thirty-three and nine becomes fifty-four. The encrypted numbers are

[16] Three commonly used models to sanitize PII are k-anonymity (Sweeney 2002), l-diversity (Machanavajjhala et al. 2007) and t-closeness (Li, Li, and Venkatasubramanian 2007). A k-anonymized dataset needs to have at least k records for each combination of attribute values; beyond k-anonymity, an l-diversity dataset generalizes or suppresses the attribute values; and a t-closeness dataset treats each attribute's value based on the data distribution.

added together resulting in 87 and sent through the channel. Bob only needs to decrypt 87 using his private key to provide the final answer 17. A blockchain user could encrypt the transaction amount and calculate the balance, tax or interest based on the encrypted amount without releasing any transaction data.

3.2.5 Zero-Knowledge Proofs

By encrypting the transaction data, blockchain can provide security protection and privacy-preserving business ecosystem (Kozlowski 2016). However, how can the public verify or confirm a transaction if they do not know the transaction details? ZKP is a scheme that allows one party to prove to another party a given statement is true without revealing any information. For example, the sender can use ZKP to prove that s/he has indeed transferred a certain amount of Bitcoins to the recipient even if s/he does not disclose the recipient and amount. The concept of ZKP was first conceived in a paper introducing an interactive proof system (Goldwasser, Micali, and Rackoff 1989), which is an abstract machine that models the computation as an exchange of messages between two parties. Blum et al. (1988) proved that if a common random string is shared between a prover and a verifier, the prover can convince the verifier that a specific statement is true without interacting with the verifier.

In October 2016, Zcash was launched as an innovative cryptocurrency for its property of high-level privacy. Zcash enables the user to "*make direct payments to each other with a vastly more efficient cryptographic protocol that also hides the amount of the payment, not just its origin*." The underlying protocol that preserves transaction privacy is a recent technical innovation called zk-SNARKS (Ben-Sasson et al. 2014), which is the acronym for zero-knowledge Succinct Non-Interactive Argument of Knowledge.

Zk-SNARK is an efficient variant of ZKP protocol by which the other users in a blockchain can verify the ownership of Bitcoin while revealing no information to the public (Ben-Sasson et al. 2014). To create a Zcash transaction, the sender first encrypts the transaction details (e.g., sender, recipient, and amount), then generates a short proof[17] using zk-SNARK and announces the transaction and the short proof in the blockchain. Any Zcash user can apply a verification key to verify the short proof without having to interact with the sender. For the smart contract, Kosba et al. (2016) use ZKP to define a security-protection and privacy-preserving smart contract protocol called Hawk, and both Zcash and Hawk possess high efficiency in cryptographic computation.

3.2.6 Continuous Monitoring and Real-Time Accounting

The emergence of blockchain creates new opportunities and challenges for continuous monitoring (Groomer and Murthy 1989; Vasarhelyi and Halper 1991) and real-time accounting (Rezaee, Ford, and Elam 2000; Yermack 2017). Since Vasarhelyi and Halper (1991) initially developed the first practical continuous audit systems, the continuous monitoring research has flourished with numerous advances, such as innovations of audit analytics (Kogan et al. 2014; Issa and Kogan 2014) and implementation of continuous monitoring (Alles et al. 2006) and continuous monitoring (Alles, Kogan, and Vasarhelyi 2008). Furthermore, the applications of continuous monitoring in ERP systems (Kuhn Jr and Sutton 2010) and XML (Murthy and Groomer 2004) and the economics of continuous assurance (Alles, Kogan, and Vasarhelyi 2002) have been examined in prior studies.

[17] A short proof consists of a single message sent from a prover to a verifier. In order to assure the zero-knowledge proof is non-interactive, short and therefore uploadable to blockchain, it is necessary to have an initial setup to generate a common random string shared between the prover and the verifier. https://z.cash/technology/zksnarks.html. Accessed 9/25/17 11:57 PM.

The goal of continuous monitoring research is to design automatic systems for highly efficient and real-time auditing. Blockchain could serve as such a technology whereby continuous monitoring system could be built to reduce cost and improve efficiency. Blockchain could be used to record and present real-time transaction data, which enables the auditors and audit systems to conduct substantive testing continuously. Besides transaction occurrence and accuracy, the assessment of other information such as rights and obligations, completeness and cutoff could be timely and efficiently conducted on Blockchain, which serves as one of the tools to automate and improve audit quality. Also, the irreversibility and tamper resistance could ensure the integrity of financial data and prevent tampering and fraud. Although the application of blockchain in auditing is still in its infancy, it seems to be promising for financial data sharing with high-level security and privacy based on the mathematical and cryptographic encryption and digital signature mechanisms (Kosba et al. 2016).

3.3 Blockchain for Accounting: A Privacy-Preserving Design

3.3.1 Colored Coins and Assets Tokenization

Blockchain provides a public, freely open ledger for recording the ownership of a wide range of assets, from stocks, bonds, real estate and automobiles to luxury handbags and priceless works of art (Yermack 2017). Furthermore, the government is exploring the use of blockchain for public records, such as birth certificates, driver licenses, and university degrees. With a tracking device (e.g., GPS, RFID[18]), a real-world asset can be

[18] Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects.

mapped onto a blockchain network and be represented by a colored coin or a token. The process of binding a class of real-world assets to a token is called asset tokenization. Table 1 shows examples of assets that can be tokenized, including currencies, tangible assets, and intangible assets. For example, U.S. Dollar can be encoded into blockchain as tokens for currency circulation; a building can be recorded in blockchain as a token for registration and trading, and copyrights can be programmed into blockchain as tokens for registration.

Table 1: Examples of Assets Tokenization

Virtual Currencies	Tangible Assets	Intangible Assets
Bitcoin Ethereum U.S. Dollar Euro China Yuan Credit Card Limits	Building Land Machinery Equipment Oil Reserves Inventory Work-in-process Office Supplies	Patent Copyright Trademark Licensing Agreement Stock shares Debt securities Financial derivatives

In a securities market, the blockchain provides a solution to the problem of delay in settlement. For example, stock options can be executed in smart contracts once predefined conditions (i.e., exercise prices) have been met. The execution of a smart contract immediately initiates a transaction that the call (put) option owner sends (receives) “cash” coins and receives (sends) “stock” coins. The coin mining process[19] involves a large number of network nodes (miners), which facilitates the transaction verification and speeds up the change of asset’s ownership. Just like Bitcoin, U.S. dollar can be issued in blockchain by the Federal Reserve System in the form of a “U.S.-dollar” coin. The online transfer of a “U.S.-dollar” coin represents a real-world transaction that a person pays

[19] Just like Bitcoin mining, colored coin mining is a procedure conducted by a group of blockchain nodes to verify and confirm transactions as well as generate new blocks to be added to a blockchain. Once a new block is added to the blockchain, the transactions the block contains will be no longer modifiable. Therefore, the transfers of the asset ownership are completed and these transactions are settled. On average, a new block is generated every 10 minutes, which means a transaction of asset transfer could be completed in around 10 minutes.

another person in cash. The “U.S.-dollar coin” will never be lost or stolen because every single dollar’s ownership is registered in the blockchain.

Similarly, intangible assets (e.g., technology patent, music copyright, and software license) can be easily programmed into blockchain represented by a colored coin. For example, when an audit firm wants to use analytical software to examine a client’s transactions, it can purchase the audit software simply by sending “U.S.-dollar” coins and receiving “software-license” coins. Shortly, “copyright” coins could be deemed as evidence in the court. For tangible assets (e.g., land, building), if the titles are encoded in blockchain or programmed into an Ethereum smart contract, the process of buying and selling a piece of land can be simplified as the seller sends a “land” coin to the buyer. Besides, many other types of properties can be recorded (tokenized) into the blockchain, such as equipment and motor vehicles, which could create a trusted and secure trading environment and reduce disputes, fraud, and inefficiency of real-world transactions.

However, asset tokenization is not an easy task due to the heterogeneous nature of assets. Each asset has its complexities, description, properties and transaction rules. For example, bonds can be classified as bonds with zero coupon, convertibles or floating rate notes. There are thousands of securities and derivatives in the capital market, which makes it impossible to use only one or several coins to describe investment activities. If every asset is tokenized in one blockchain, the scalability can render blockchain infrastructure technically problematic. In addition, some types of intangible assets are not easy to tokenize, such as brand loyalty and workforce competence.

Wide implementation of Bb-TPS could start from many small blockchain networks in local business ecosystems formed by regulators, business associations and companies along the supply chains. Then, a set of small blockchains could be connected via the sidechain technique (Back et al. 2014). Asset tokenization could also start with issuing tokens linked to high-value real-world assets, such as automobiles. As technology improves and tokenization cost decreases, many less-expensive assets will be gradually taken into consideration for digitizing property records. The audit of the tokenization process becomes crucial because the first ownership represents the starting point of the provenance of an asset. The first owner of a token should be responsible for the existence and integrity of the corresponding real-world asset that the token links to, and a detailed audit must be performed on the asset's tokenization process. Once an audited asset is hashed to the blockchain, the blockchain will then automatically keep track of the asset for its useful life. Moreover, a complete audit is also necessary when a Bb-TPS environment is set up and initiated. In the near future, as a large number of real-world objects have had coins associated with them, the tamper resistance and irreversibility of blockchain will become the core functionalities for auditing and fraud prevention.

3.3.2 Real-Time Accounting and Continuous Reporting

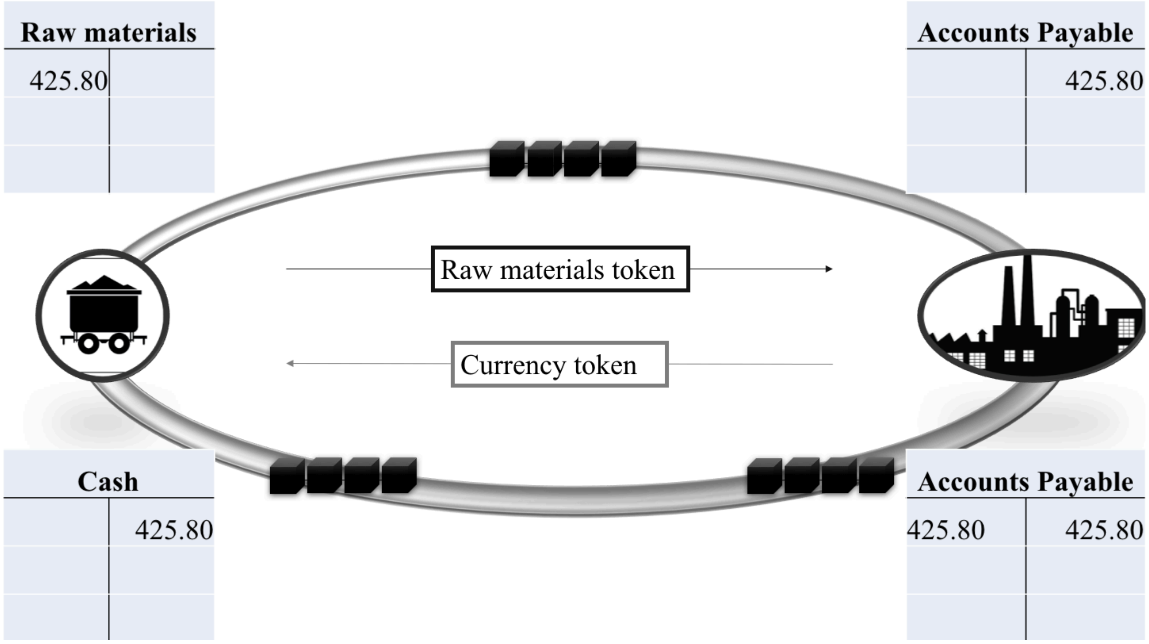
The deployment of an enterprise information system, such as SAP, increases the speed of business integration by connecting a firm with outside trading partners. Blockchain technology brings another wave of upgrading the management information systems and the business ecosystems. Based on blockchain technology, this paper proposes a design of Blockchain-based Transaction Processing System (Bb-TPS) and creates a prototype to demonstrate the functionality of Bb-TPS.

For example, in a blockchain-based business ecosystem, assets and resources (e.g., raw materials, inventory of finished goods, employee labor) have been defined and tokenized in the blockchain. A supplier sends “raw materials” coins along with the raw materials to a manufacturer and will receive “cash” coins as payment. After the raw materials are turned into finished goods, the firm sells the products to a customer by sending a “product” coin along with the product and receiving “cash” coins as payment. Alternatively, the contractual terms, such as cash on delivery (COD), can be encoded in a smart contract that enforces the rule that the customer makes a payment at the time of product delivery. The financing and investment activities can be represented as a bank sends “loan” coins to the firm or the firm collects “cash” coins from the stock market and distributes “stock-share” coins. The governmental agencies collect and refund “tax” coins, and employees can also use tokenized labor time to calculate and request salaries.

This example shows how the Bb-TPS supports and records transactions in a business ecosystem in an automatic and real-time manner, which reduces the systematic duplication of manual effort and improves fiscal accuracy and efficiencies. In addition to traditional double-entry accounting systems, blockchain provides shared transaction records that link the trading parties’ journal entries and further facilitates inter-organizational collaboration. Figure 2 and Figure 3 show the examples of inter-organizational transactions.

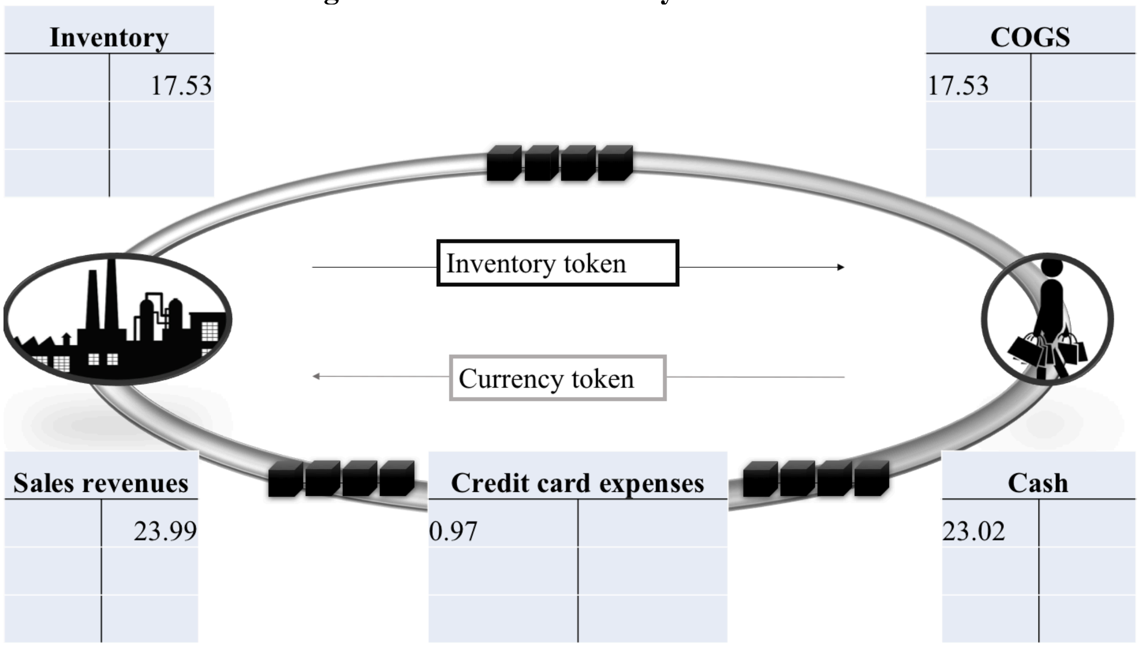
In the procurement-to-payment cycle, a supplier sends “raw-materials” coins (e.g., wood) to a manufacturing firm M, which triggers four journal entries for two economic entities. For simplicity, we focus on firm M’s accounting processes. When firm M receives the “raw- materials” coins, Bb-TPS automatically generates two electronic entries with the same amount of which one debits the raw materials account and the other one credits accounts payable. After firm M received the raw materials and checked the quality and quantity of the goods, it could disburse “cash” coins to the supplier, which generates two more electronic entries of which one debits accounts payable, and the other one credits cash.

Figure 2: Procurement-to-Payment Cycle in Bb-TPS



In the order-to-cash cycle, firm M sells finished goods (e.g., a table) to a customer, which triggers four journal entries for two economic entities. When sending customer the “table” coins, Bb-TPS automatically generates two electronic entries with the same amount of which one debits cost of goods sold, and the other one credits inventory. Then, the customer pays for the table with a credit card, which generates another three electronic entries of which one debits cash, one debits credit card expenses, and the third one credits sales revenue.

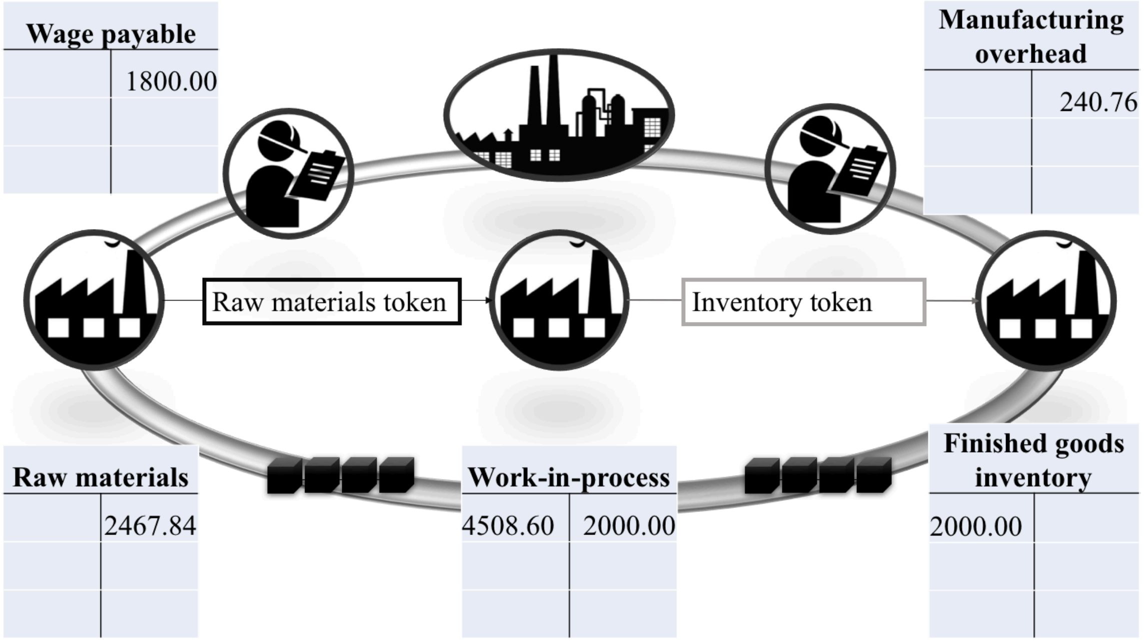
Figure 3: Order-to-Cash Cycle in Bb-TPS



In addition to inter-organizational transactions, Bb-TPS can also be a communication platform within a firm. Suppose firm M is a conglomerate firm whose subsidiaries form a complete industry value chain. For example, firm M could be a manufacturer that has many subsidiaries operating in different but inter-locked sectors from petroleum exploration and refinement to gas retail. Bb-TPS would link headquarter and subsidiaries and would record intra-organizational transactions.

From raw materials to finished goods, Bb-TPS needs to keep records of the manufacturing process of a factory. In this process, more than one type of coins is consumed and transformed into another type of coin. For example, “raw-materials” coins are transformed to “finished-product” coins. In Figure 4, subsidiary A of firm M provides raw materials to another subsidiary B, and subsidiary B delivers finished goods to subsidiary C. For simplicity, we focus on the subsidiary B’s accounting processes. When subsidiary B receives the “raw-materials” coins, Bb-TPS automatically generates two electronic entries coins with the same amount of which one debits raw materials, and the other one credits accounts payable. Then, subsidiary B consumes raw materials, labor, and manufacturing overhead and produces work-in-process, which generates four electronic entries of which three credit raw materials, wages payable and manufacturing overhead and one debits work-in-process. After the assembly process, Bb-TPS credits the work-in-process and debits finished goods inventory. In the end, the final products are delivered from subsidiary B to subsidiary C, which generates four electronic entries: two credit finished goods inventory and sales revenue and the other two debit COGS and accounts receivable.

Figure 4: Production Line in Bb-TPS



In the manufacturing process, "raw-materials" coins are transformed into "work-in-process (WIP)" coins, and then "finished-goods" coins. To present the production process in which raw materials are transformed into finished products, subsidiary B first needs to identify the ownership of "raw materials," "labor" and "manufacturing-overhead" coins and retire those coins to be turned. Retiring a coin refers to disposing of or destructing a real-world asset. Technically, a blockchain user will send the coin that s/he decided to dispose of to a specific wallet address. The unique wallet will permanently lock up the coins and keep them out of circulation. After subsidiary B retires “raw materials,” “labor” and “manufacturing-overhead” coins, it could issue new “WIP” coins that would further be converted to “finished-goods” coins by binding them to the physical products with sensors.

In the near future, if all economic entities use digital currency as their exchange medium, register all assets in Bb-TPS, and tokenize every product by digital coin, all business transactions of this business ecosystem will be posted on a public blockchain and recorded permanently with time stamps, which prevents them from being altered ex-post. In this case, auditors and regulators could aggregate a firm’s transactions into financial reports[20] (e.g., income statement and balance sheet) at any time. The financial statement consolidation could be efficiently and continuously conducted through inter-organizational nettings of digital coins and depreciation schedule or inventory revaluation could be executed based on fair market value by smart contracts. Most importantly, continuous reporting on Bb-TPS allows the public to rely less on quarterly or annual reports and demotivates management from manipulating transactions, such as backdating, to positively impact management compensation stocks or options. Shareholders will have more trust in financial data integrity, and the increased trust could result in saving some social costs of

[20] The blockchain is a single-entry system which keeps record of the event of asset transfer. A “single-entry bookkeeping system in a blockchain” is introduced to replace the traditional double-entry system (Simon, 2016), while another group of cryptographers provides discussions about a “triple-entry” system (Grigg, 2005). The definition of a “triple-entry” accounting system used by blockchain-accounting practitioners is different from Ijiri (1986)’s definition. The blockchain based “triple-entry” system is an enhancement to the traditional double-entry system where all accounting entries are recorded by a third entry into the blockchain. In contrast to traditional accounting where the trading partners independently book debit and credit to their own accounts, blockchain’s shared transaction records link the journal entries of trading parties, which provides additional assurance in auditing.

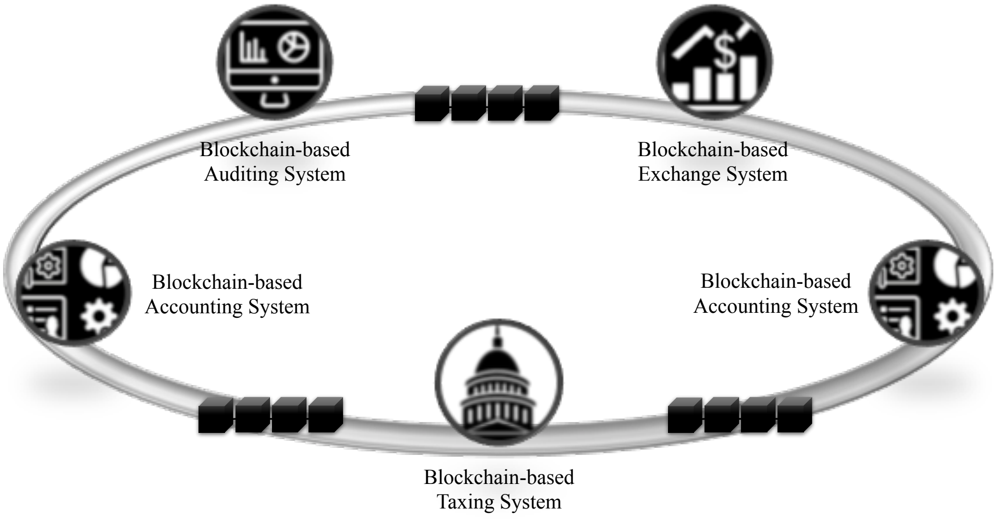
mistrusting company’s management, i.e., reducing the cost of auditing which exceeds $50 billion per year (Yermack 2017).

3.3.3 Blockchain Neutrality

It should be noted that Bb-TPS should be conceived as a neutral and independent infrastructure that underpins business event recording. Similar to net neutrality (Jordan 2009), blockchain neutrality represents the argument that the blockchain ledger applies to different accounting treatments or even different accounting standards (i.e., GAAP and IFRS). The generic neutrality of Bb-TPS design requires the separation of event recording and accounting treatment (e.g., period-end adjustments/closing), even if it is technically possible to encode accounting rules in blockchain (e.g., an aging method can be programmed in a smart contract for doubtful accounts). In general, blockchain serves as a neutral shared database that keeps transaction records per se, while journalizing and adjustments are processed on top of the Bb-TPS infrastructure using enterprise information systems. In this case, the Bb-TPS provides the transaction level assurance for the full population and presents the facts of business events, while the individual enterprise information systems could merely perform aggregation, materialization, and adjustment to prepare financial statements.

This arrangement also allows auditors to aggregate individual transactions and verify financial reports based on accounting standards. Figure 5 shows the Bb-TPS-neutrality principle: Bb-TPS is used for recording transactions per se, and each economic entity has free choice of building the accounting, auditing or taxing systems on top of Bb-TPS. Blockchain only contains the transaction data, while on top of that ERP systems can automatically extract the data from blockchain, aggregate the transactions and provide information for use in producing financial statements.

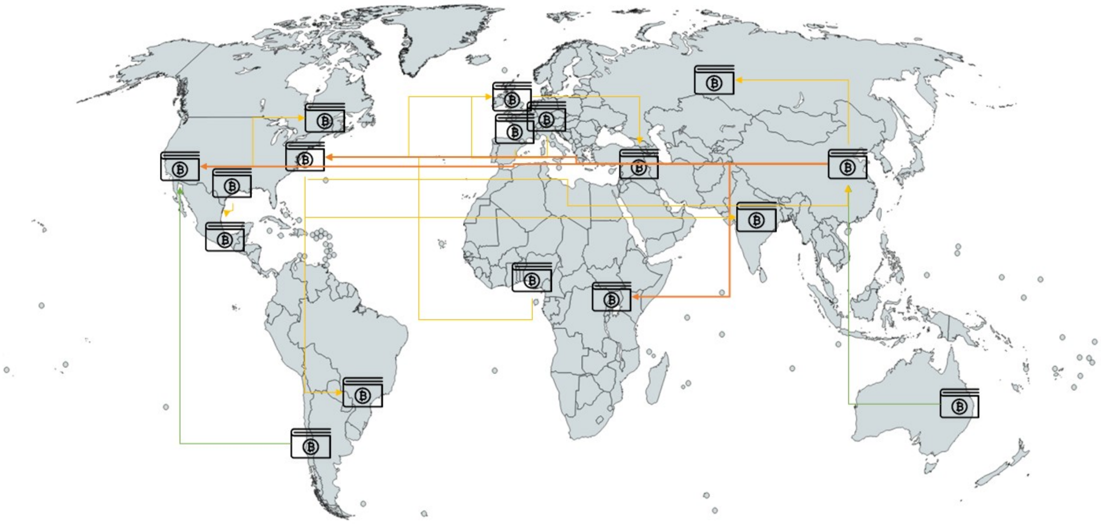
Figure 5: The Neutrality of Bb-TPS


Blockchain-based Auditing System
Blockchain-based Exchange System
Blockchain-based Accounting System
Blockchain-based Accounting System
Blockchain-based Taxing System

3.3.4 Continuous Monitoring, Fraud Prevention, and Auditor's Role

When a business transaction occurs in Bb-TPS, the business event represented by transferring a digital coin from one entity to another will be announced publicly and recorded in real-time. The Bb-TPS continuously adds transactions to the blockchain and shares the blockchain with all users; therefore, auditors are able to obtain a full copy of his/her client's transaction data. The real-time availability of transaction data makes it possible for auditors to monitor firm's global assets continuously. Figure 6 shows a Blockchain-based Continuous Monitoring System (BbCMS) that conducts real-time tracking of firm's assets. If an auditor wants to confirm a client's accounts receivable with its customers or accounts payable with suppliers, the auditor only needs to collect the relevant sales or procurement transaction data from the blockchain and perform analytical procedures (e.g., matching transaction amount to accounts receivable or payable). Besides, the automatic confirmation scheme reduces the duplication of work for the reconciliation and improves audit efficiency.

Figure 6: Global Blockchain based Continuous Monitoring System



Fraud prevention is an essential issue in the audit practice. In general, fraud is defined as either intentional embezzlement of assets from a firm or intentional misstatement of financial reports to mislead stakeholders. Bb-TPS can trace the movement of firm's tokenized assets; therefore, it can proactively deter asset (e.g., cash or inventory) misappropriation. Physical controls are still needed to safeguard physical goods. Bb-TPS is used to identify possible misappropriation and concealment of assets or liabilities at the ownership level. The continuous monitoring based on blockchain makes it hard for managers to engage in earnings management or even commit to fraud, such as "big bath" and "cookie jar reserve" scheme. For example, if the blockchain presents the fact that a high rate of return happens immediately after an abnormal high-volume year-end sale, the blockchain-based continuous auditing systems (BbCAS) could raise the alarm and send a message to the auditor about the potential "channel-stuffing" fraud and revenue recognition concern. Besides, the transaction data from blockchain could serve as high-quality audit evidence as 100% transactions have been verified once occurred.

Bb-TPS delivers real-time data and provides real-time reporting, which makes quarterly and annual reporting less critical and therefore lowers management's incentives to manipulate reported earnings periodically. Even if accrual earnings management shifts to real earnings management (Cohen, Dey, and Lys 2008), the unprecedented transparency still enables the analysts and auditors to use their judgment to identify whether management has made suboptimal or myopic decisions.

If a firm has engaged in a financial reporting fraud scheme, Bb-TPS keeps the records of all relevant transactions, which could deliver valid evidence showing the possible accounting irregularities. For example, the criteria of revenue recognition can be encoded in a blockchain-based continuous monitoring system (BbCAS) using smart contracts to ensure that all conditions have been met before recognizing sales revenue. If a firm's sales revenue is overstated using "channel stuffing" or "round tripping," the BbCAS raises the alarm about the suspicious transactions and indicates the type of fraud scheme. Thus, Bb-TPS can prevent managers from cooking the books or tampering with the data, such as creating fictitious transactions or backdating sales contracts or option compensation. Also, related-party transactions could be self-disclosed; therefore, suspicious transfer of assets implying conflicts of interest can be spotted instantly. In order to defeat securities exchange fraud, Bb-TPS makes managerial ownership so transparent that insider buying or selling can be detected automatically in real time. Corporate voting using Bb-TPS could be more transparent, accurate and fast (Yermack 2017). Furthermore, the transparency reduces opportunities for corruption or bribery behavior between regulators and firm's management.

However, Bb-TPS cannot automatically detect every fraud scheme. Some frauds involving complex transactions still need the auditors to brainstorm, exercise professional skepticism, and investigate the transaction details.

3.3.5 Internal Control and Permissions Management

When a firm enters into a transaction selling goods to a customer, the firm puts in "goods" coins and sends the "goods" coin together with physical goods to the customer. Once the transaction is announced but before it is confirmed, Bb-TPS will apply built-in rules to check: (1) whether the sender has sufficient balance of the "goods" coins to make the transfer, (2) whether the coins to be sent have valid trace back to previous transactions, and (3) whether the coins have been double spent. All the checks ensure the sender sends the valid and unspent coins to the recipient. After a set of transactions has been confirmed, a blockchain miner collects those transactions and solves proof-of-work[21] to generate a new block to be added to the blockchain.

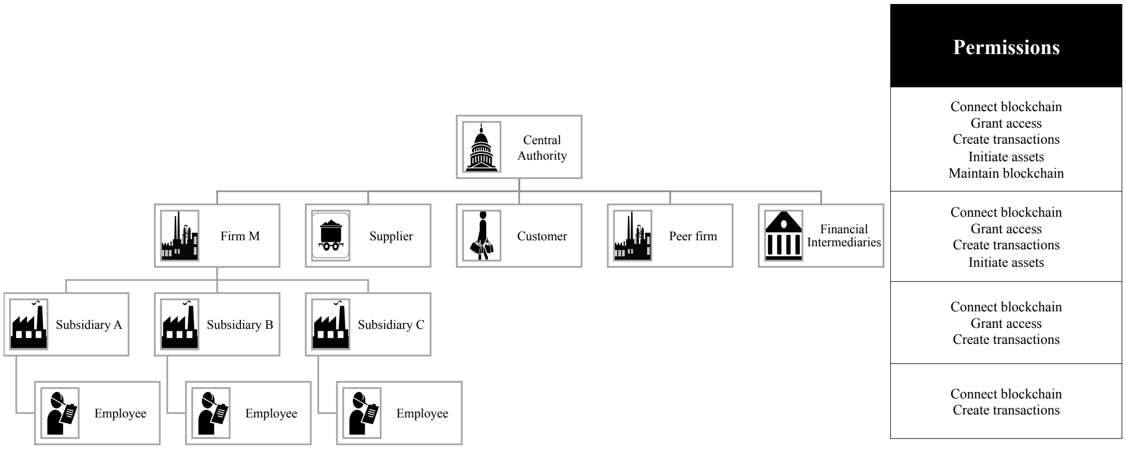
On top of the built-in rules, a firm's internal controls can be added to Bb-TPS or smart contracts. It is conceivable that smart contract can be used to automate the enforcement of business rules compliance. The add-on controls can efficiently reduce firms' business risk and fraud risk. As Bb-TPS shows the movement of a tokenized asset, it prevents managers or employees from embezzling firms' resources, such as cash or

[21] A proof of work is a cryptographic data puzzle which require a computationally costly and time-consuming effort to produce the result which is easy for others to verify. In a blockchain, producing a proof-of-work is a random process which requires miners to do a lot of trials before a valid proof of work is generated. A description is available at: https://en.bitcoin.it/wiki/Proof_of_work. Accessed 5/23/17 10:32PM.

inventory. Bb-TPS with access controls only allows authorized users to read or create transactions. For example, only the shipper has permissions to transfer “goods” coins to a customer under a certain credit limit, and only the procurement staff can purchase raw materials from suppliers. In addition, Bb-TPS can properly separate job duties when sufficient competent staffs are available at the entity. For example, the accountant and the treasurer will have different access to the coins in Bb-TPS. In order to ensure product quality, the Bb-TPS monitors the production line from raw materials, work-in-process, to finished goods. Besides, a lot of duplicated work for bank reconciliation can be reduced. For example, the accountant does not have to have concerns about the timing difference between the bank and firm’s records caused by deposits in transit or outstanding checks, and the risk of receiving an insufficient funds check could be avoided because of the built-in check schemes. A properly designed Bb-TPS can precisely and dynamically control who can connect to the blockchain, initiate transactions and mine the blocks.

Developing an irreversible blockchain for a business ecosystem requires active participation from many groups, such as firm’s management, suppliers, and customers, stakeholders and regulators. However, as an increasing number of parties are brought into Bb-TPS, confidentiality becomes a critical issue because every participant can see each other’s transactions. So, some firms choose to deploy private blockchains for internal use. In this case, the firm has the majority computational power over the private blockchain, which makes it possible to rewrite or falsify transaction records. We propose a solution – permissions management - by assigning miner role to a trusted party or managing the permissions to access the private blockchain. Figure 7 shows the hierarchy of permissions management schema in a private blockchain.

Figure 7: Hierarchy of Permissions Management



For example, a firm can build a blockchain network with its trading partners, stakeholders, and regulators in which only regulators are allowed to confirm transactions, generate blocks and grant permissions to new users. Furthermore, in order to remove central authorities, Bb-TPS can allow more than one sophisticated user to grant access and mine blocks. For example, a long-tenured user who has created a large number of transactions in Bb-TPS could participate in mining blocks and maintaining the blockchain, or it could grant access to a limited number of new users. As long as the honest users (who have no intention of perpetrating fraud) outnumber the dishonest users (who have the intent of perpetrating fraud), this network will guarantee the data integrity of transactions in Bb-TPS.

3.3.6 Blockchain Confidentiality and Homomorphic Encryption

Private blockchain creates a closed business ecosystem but has the possibility of losing data integrity, while public blockchain ensures data irreversibility but probably loses data confidentiality. Encryption, one type of which is called homomorphic encryption, provides a cryptographic solution that enables firms' secret information to be sealed. Homomorphic encryption allows computations to be done on encrypted data without first having to decrypt it.

Figure 8: Homomorphic Encryption in Bb-TPS

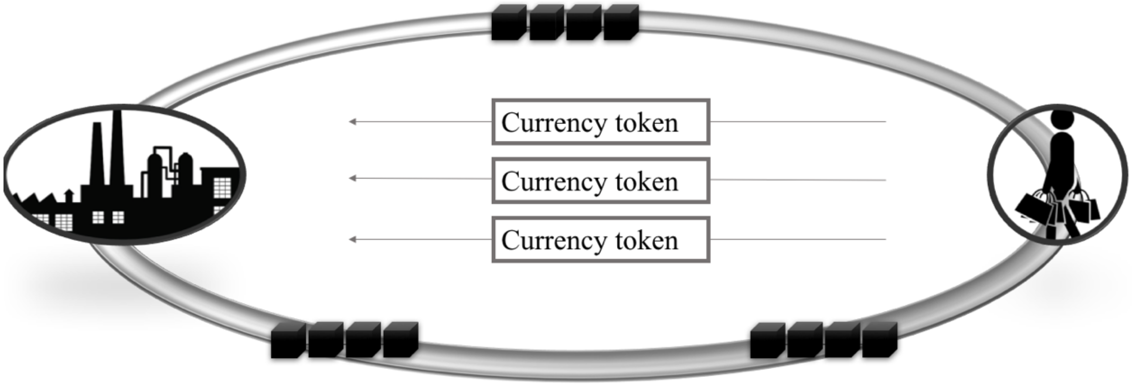


Figure 8 and Table 2 show an example of how homomorphic encryption protects a Bb-TPS user's privacy. In Table 2, there are at least three blockchain-based transactions. Each of them has a Transaction Hash (transaction unique identifier), Block Hash (indicating which block contains this transaction), Received Time (indicating when this transaction completed), Inputs Address (indicating the sender's address), Total Inputs (indicating how much the sender transfers), Outputs Address (indicating the recipient's address), Total Outputs (indicating how much the recipient receives) and Transaction Fee (indicating the cost this transaction). If a customer, named Alice, buys a luxury handbag and pays for it using an installment plan in Bb-TPS, she can send "cash" coins to the handbag store every month. As a hash function anonymizes Alice's cryptowallet address, her personally identifiable information (i.e., name) will never be disclosed in Bb-TPS.

Table 2: Transaction Details

Transaction Hash	Block Hash	Received Time	Inputs Address	Total Input	Outputs Address	Total Output	Transaction Fee

				s		s	
feff4ca52e4	00d520	2017-01-24 14:20:33	1NVatZC	0.6613	1PeA5LS	0.6612	0.0002
erjhfff12nc	00ff310	2017-02-24 16:12:33	1NVatZC	0.2621	1PeA5LS	0.2621	0.0001
eep098ujnf	0000h3	2017-03-24 20:00:09	1NVatZC	0.1876	1PeA5LS	0.1876	0.0001
…	…	…	…	…	…	…	…

However, if a node is a frequent user having a long transaction history in Bb-TPS (e.g., a multinational company), the sophisticated data mining and pattern recognition techniques will let some other parties (e.g., a competitor) infer the connections between the user's identity and its cryptowallet addresses. This type of "network analysis" for blockchain (Reid and Harrigan 2013) could even probably recognize the frequent user's transaction patterns, trace its trading partners and link the cryptowallet address to the IP address. Then, the user's privacy would be at risk once a hacker breaks into the IP address and steals the user's assets. Even if a user is able to apply anti-tracking mechanisms (e.g., creating many anonymous wallets) to prevent privacy attack, the big data of high-frequency trades will expose the user to malicious attackers and compromise its confidentiality.

To reduce exposure risk, Bb-TPS uses homomorphic encryption to encode sensitive transaction information into unreadable ciphertext. In the above handbag-shopping example, Alice can choose to encrypt her transaction details using her private key. Then, the encrypted transaction will be written into Bb-TPS in human-unreadable format (Table 3).

Table 3: Example of Encrypted Transaction Details

Transact	Bloc	Receiv	Inputs	Total	Outpu	Total	Transactio

ion Hash	**k Hash**	**ed Time**	**Address**	**Inputs**	**ts Address**	**Outputs**	**n Fee**
feff4ca52e4	00d520	2017-01-24 14:20:33	1NVatZC	hrnPgua6vm	1PeA5LS	TUC5VpcQAv	5XXPhf+ku6U
erjhfff12nc	00ff310	2017-01-24 16:12:33	1NVatZC	eI+IYf/YZb	1PeA5LS	eI+IYf/YZbM	IBieGxOyZKk
eep098ujnf	0000h3	2017-01-24 20:00:09	1NVatZC	NuJeseiM1i	1PeA5LS	NuJeseiM1iY=	IBieGxOyZKk
…	…	…	…	…	…	…	…

As shown in Table 3, not only the input and output addresses but also the input and output amount and transaction fees have been encrypted and recorded in the blockchain, which makes it difficult for an intruder to steal secrets based on the limited disclosures (i.e., block hash, transaction hash and timestamps).

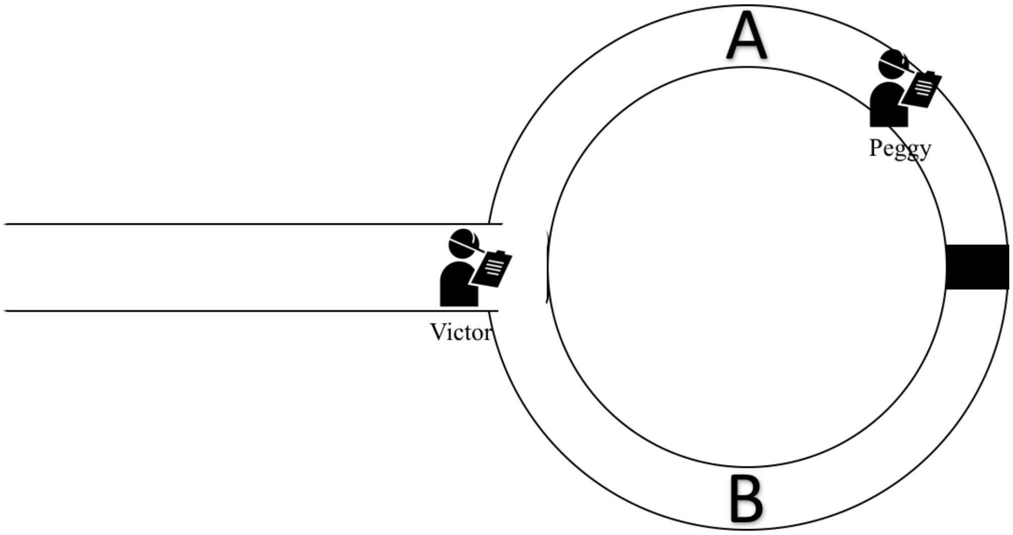
Furthermore, with homomorphic encryption, we can set up mathematical operations (e.g., additions, multiplications, quadratic functions) between two ciphers. In Alice’s case, although the transaction amount is encrypted, Bb-TPS can still calculate the account balance by aggregating the encrypted transaction amounts. In addition, by aggregating all transactions and offsetting the accounts receivable and accounts payable between two subsidiaries, Bb-TPS could deliver a consolidated statement automatically in real time. The utilization of homomorphic encryption enables a secure and confidentiality-preserving Bb-TPS with guaranteed confidentiality.

3.3.7 Blockchain Confidentiality and Zero-knowledge Proofs

Since the transaction details (i.e., sender, recipient, and amount) could be hidden using encryption, the problem is how the other Bb-TPS users can verify the transaction before it is written in the blockchain. For example, if only the trading partners have access to transaction details, the other Bb-TPS users would not be able to trace the asset transfers and identify the owner of that asset. In order to deliver a certain level of transparency while preserving transaction confidentiality, this paper uses zero-knowledge Succinct Non-interactive Argument of Knowledge (zk-SNARK) to create a confidentiality preserving and transaction verifiable Bb-TPS.

Zk-SNARK uses a variant of zero-knowledge proofs mechanism. The basic idea of the zero-knowledge proof is that one party (prover) convinces another party (verifier) that its statements are true without revealing the content of that statement (Rackoff and Simon 1991). For example, a person named Peggy claims that she knows the secret to open a door in a cave shaped like a circle (shown in Figure 9). Peggy wants to prove to a person named Victor that she knows the secret; however, she is not allowed to tell the secret to Victor. First, they decide to label the upper and lower paths A and B. Then, Peggy randomly takes A or B path and walks in while Victor waits outside without seeing which path Peggy takes. Next, Victor calls the path where Peggy has to show up. If Peggy walks in the same path as Victor calls, she can quickly return to the entrance; if Peggy does not walk in the path Victor calls, she needs to go through the door and walk out to the entrance using another path. Suppose that Peggy indeed knows the secret, she could either use the secret to open the door or simply returns using the same path as she entered. However, if Peggy lies, she can only show up on the path which she uses to enter and fails on the other path if Victor picks it. For every trial, the dishonest Peggy will have 50% chance to guess it right. If repeating the trials 50 times, Peggy’s chance to guess all correct paths would be impossibly small.

Figure 9: Example of Zero-Knowledge Proof



The above example shows the standard interactive zero-knowledge proof, and a non-interactive zero-knowledge proof is a method by which no interaction between the prover and verifier is necessary. Based on prior work on non-interactive zero-knowledge proofs and recent breakthroughs of zk-SNARKs, Ben-Sasson et al. (2014) create a publicly shared ledger with strong privacy.

By adopting zk-SNARK, the encrypted transactions in Bb-TPS can be verified without first decrypting the details. In this scheme, the other Bb-TPS users need to verify the following three statements: (1) “the sender has a valid source of that coin s/he is about to transfer”, (2) “the coin has not been double spent”, (3) “the input amount balances out the output amount”. In order to prove the three statements, a sender uses zk-SNARK to generate a short proof (which could be quickly processed by the other users) and releases the verification key while encrypting transaction details. Then, s/he announces the transaction in the form of ciphertext. The other users, such as block miners, work together to verify the three statements by validating the short proof and verification key, but they will not be informed about any content of the three statements. Once the verification result is valid, the encrypted transaction is ready to be written in a new block and added to the blockchain. Therefore, all business transaction data could be encrypted and confirmed based on the zk-SNARK scheme[22], and none of the transaction details would be publicly disclosed. On the other hand, if a user wants to disclose the transaction information voluntarily, s/he still has an option to announce the transaction details. If the user chooses to shield the information such as sender's address, recipient's address and transaction amounts, s/he must generate a zk-SNARK short proof and share it with the public for verification.

3.4 Prototyping and Evaluation

3.4.1 Basic Infrastructure

In order to test the proposed framework of design, this paper develops a prototype of the Blockchain-based Transaction Processing System (Bb-TPS) using the core code from the Multichain[23] platform. Multichain.com is an open source platform for blockchain applications. It helps quickly build applications on blockchains and shared ledgers, and it also provides the functions of permissions management, assets issuance and data sharing. Based on the Multichain platform and four Windows servers at Rutgers CAR-Lab[24], this study creates a four-node blockchain network and tests the performance of the proposed Bb-TPS framework.

[22] For the technical details, please refer to Ben-Sasson, Chiesa, Tromer and Virza. 2015. Succinct Non-Interactive Zero-Knowledge for a Von Neumann Architecture. Available at: https://eprint.iacr.org/2013/879.pdf. Accessed 5/23/17 10:33PM.

[23] http://www.multichain.com. Accessed 5/23/17 10:34PM.

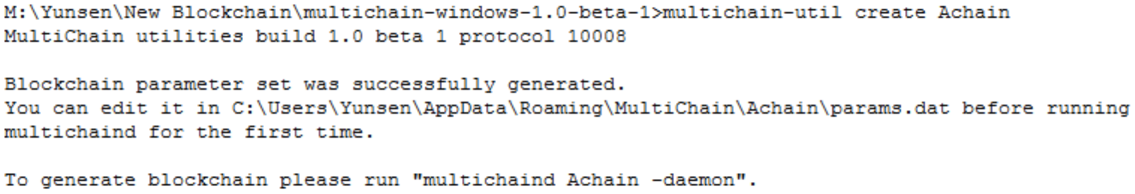
[24] http://raw.rutgers.edu/carlab.html. Accessed 5/23/17 10:35PM.

First, this study created a new blockchain named "Achain" and initiated it on Server 92. Second, the other three servers (i.e., Server 109, Server 116 and Server 117) are used to connect to the "Achain." There are three subprocesses to complete the connection: 1) the other three servers send requests for the connection permissions; 2) Server 92 grants connect (or together with send, receive and mine) permissions to the other three servers; 3) the other three servers use their public and private keys to connect to the "Achain". Finally, these four servers form a private blockchain network that serves as the infrastructure of Bb-TPS. Figure 10 shows the steps of creating and initiating the "Achain" and connecting the nodes to it[25].

[25] This prototype focused on the demonstration of Blockchain based Transaction Processing System without discussing the reward mechanisms for block mining. The authors understand the mining incentives are an important parameter in blockchain design, however our main objective is to apply this technology to bookkeeping and fraud prevention instead of encouraging participation. We hope this will stimulate further discussion of this issue.

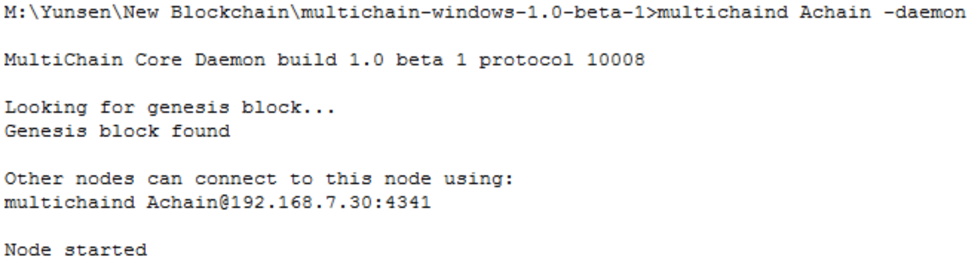
Figure 10: Creation and Initiation of "Achain"

Figure 10-1. Creating "Achain"



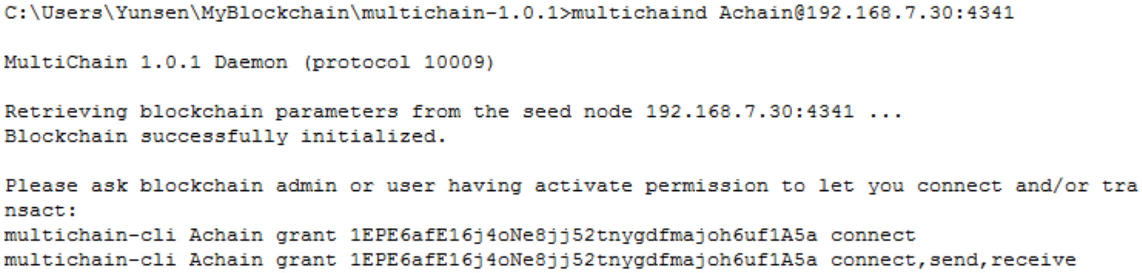
```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-util create Achain
MultiChain utilities build 1.0 beta 1 protocol 10008

Blockchain parameter set was successfully generated.
You can edit it in C:\Users\Yunsen\AppData\Roaming\MultiChain\Achain\params.dat before running
multichaind for the first time.

To generate blockchain please run "multichaind Achain -daemon".
```

Figure 10-2. Initiating "Achain"



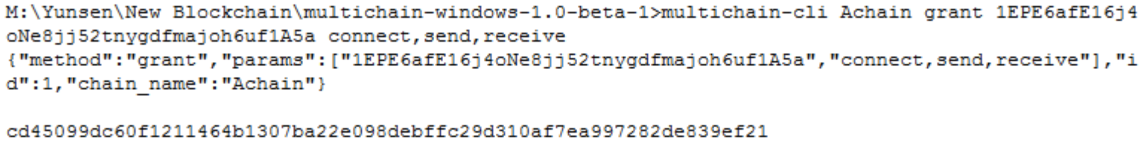
```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichaind Achain -daemon

MultiChain Core Daemon build 1.0 beta 1 protocol 10008

Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind Achain@192.168.7.30:4341

Node started
```

Figure 10-3. Requesting to access to "Achain."



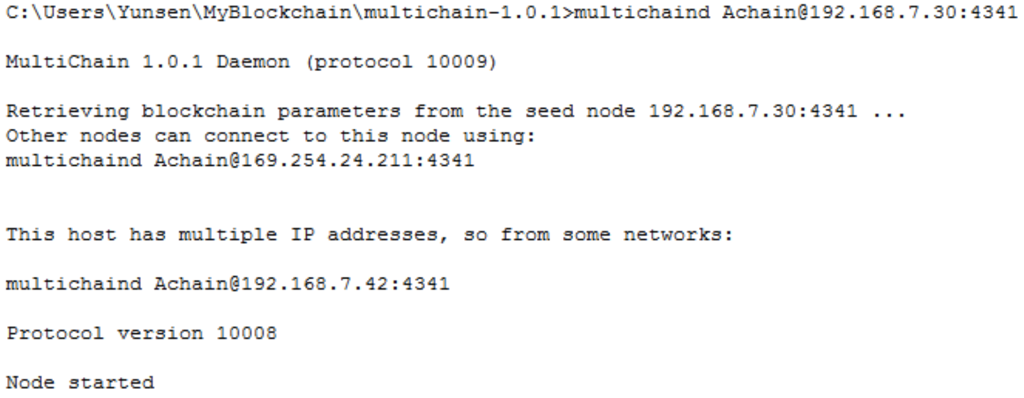
```
C:\Users\Yunsen\MyBlockchain\multichain-1.0.1>multichaind Achain@192.168.7.30:4341

MultiChain 1.0.1 Daemon (protocol 10009)

Retrieving blockchain parameters from the seed node 192.168.7.30:4341 ...
Blockchain successfully initialized.

Please ask blockchain admin or user having activate permission to let you connect and/or tra
nsact:
multichain-cli Achain grant 1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a connect
multichain-cli Achain grant 1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a connect,send,receive
```

Figure 10-4. Granting access to "Achain."



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain grant 1EPE6afE16j4
oNe8jj52tnygdfmajoh6uf1A5a connect,send,receive
{"method":"grant","params":["1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a","connect,send,receive"],"i
d":1,"chain_name":"Achain"}

cd45099dc60f1211464b1307ba22e098debffc29d310af7ea997282de839ef21
```

Figure 10-5. Connecting "Achain"

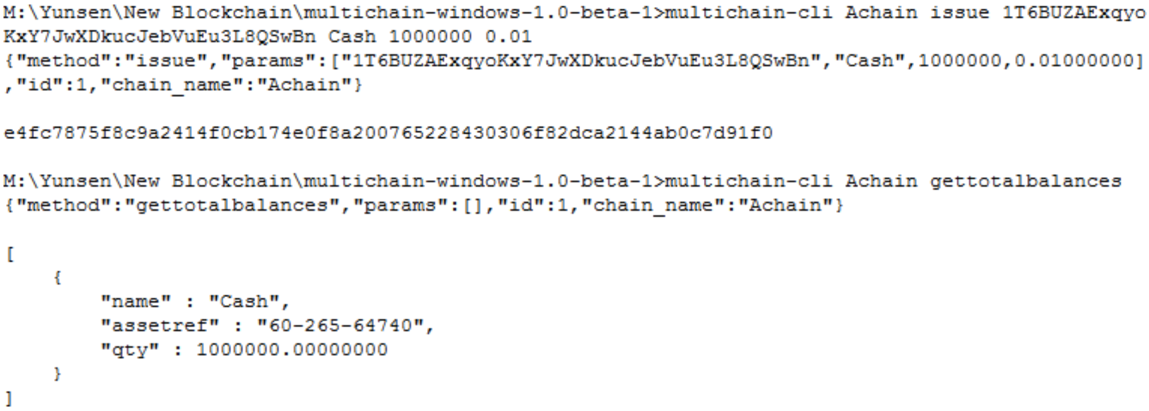


```
C:\Users\Yunsen\MyBlockchain\multichain-1.0.1>multichaind Achain@192.168.7.30:4341

MultiChain 1.0.1 Daemon (protocol 10009)

Retrieving blockchain parameters from the seed node 192.168.7.30:4341 ...
Other nodes can connect to this node using:
multichaind Achain@169.254.24.211:4341


This host has multiple IP addresses, so from some networks:

multichaind Achain@192.168.7.42:4341

Protocol version 10008

Node started
```

After creating “Achain” and connecting the nodes to it, this study simulated the processes of issuing and transferring assets in the blockchain. As shown in Figure 11, first we checked the issue permissions and established that only Server 92 has the permission to issue new assets. Then, 1,000,000 new assets “Cash” were issued with the unit of 0.01, and 100 “Laptop” assets were tokenized with the unit of 1. Thus, the simulation of assets issuance and tokenization was completed.

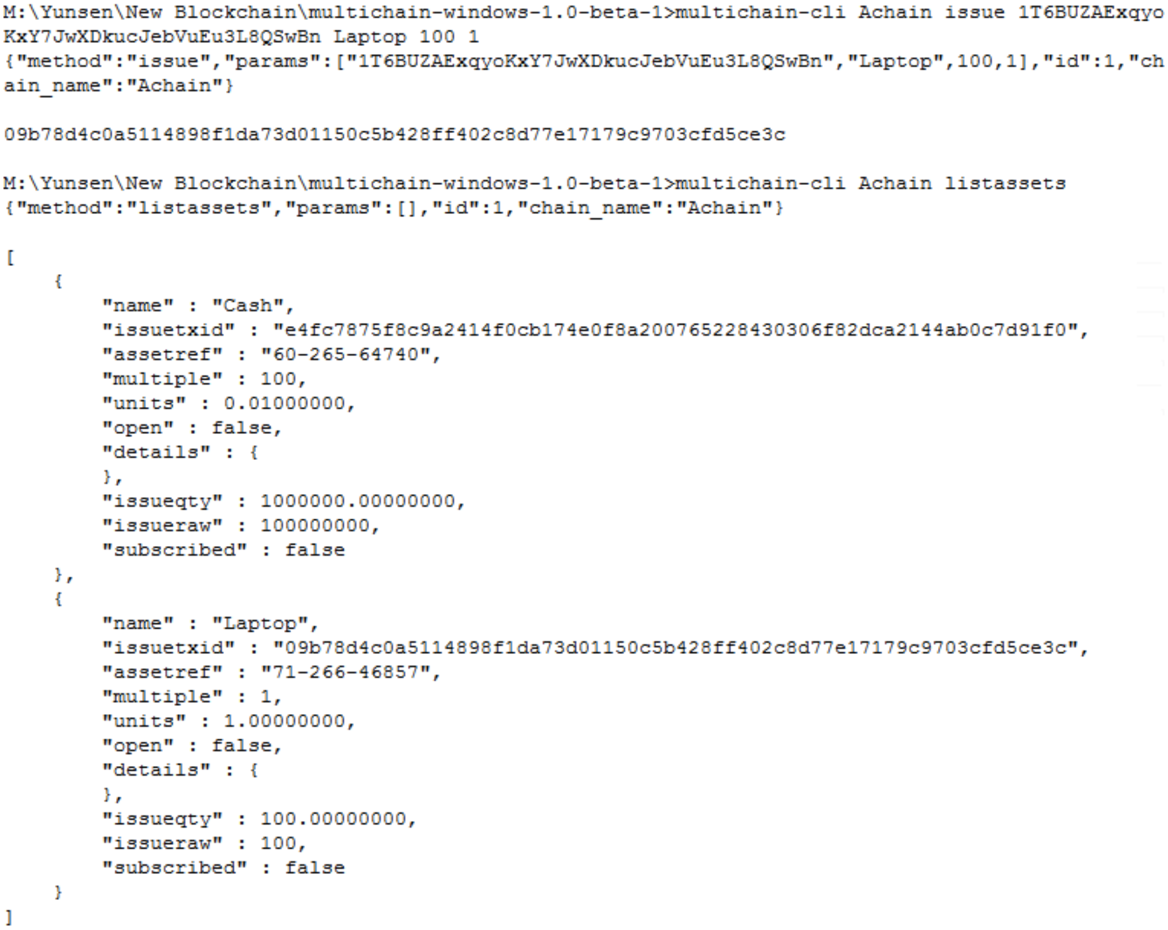
Figure 11: Assets Issuance and Tokenization

Figure 11-1. Coin Issuance - Cash



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain issue 1T6BUZAExqyo
KxY7JwXDkucJebVuEu3L8QSwBn Cash 1000000 0.01
{"method":"issue","params":["1T6BUZAExqyoKxY7JwXDkucJebVuEu3L8QSwBn","Cash",1000000,0.01000000]
,"id":1,"chain_name":"Achain"}

e4fc7875f8c9a2414f0cb174e0f8a200765228430306f82dca2144ab0c7d91f0

M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain gettotalbalances
{"method":"gettotalbalances","params":[],"id":1,"chain_name":"Achain"}

[
    {
        "name" : "Cash",
        "assetref" : "60-265-64740",
        "qty" : 1000000.00000000
    }
]
```

Figure 11-2. Assets Tokenization - Laptop



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain issue 1T6BUZAExqyo
KxY7JwXDkucJebVuEu3L8QSwBn Laptop 100 1
{"method":"issue","params":["1T6BUZAExqyoKxY7JwXDkucJebVuEu3L8QSwBn","Laptop",100,1],"id":1,"ch
ain_name":"Achain"}

09b78d4c0a5114898f1da73d01150c5b428ff402c8d77e17179c9703cfd5ce3c

M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain listassets
{"method":"listassets","params":[],"id":1,"chain_name":"Achain"}

[
    {
        "name" : "Cash",
        "issuetxid" : "e4fc7875f8c9a2414f0cb174e0f8a200765228430306f82dca2144ab0c7d91f0",
        "assetref" : "60-265-64740",
        "multiple" : 100,
        "units" : 0.01000000,
        "open" : false,
        "details" : {
        },
        "issueqty" : 1000000.00000000,
        "issueraw" : 100000000,
        "subscribed" : false
    },
    {
        "name" : "Laptop",
        "issuetxid" : "09b78d4c0a5114898f1da73d01150c5b428ff402c8d77e17179c9703cfd5ce3c",
        "assetref" : "71-266-46857",
        "multiple" : 1,
        "units" : 1.00000000,
        "open" : false,
        "details" : {
        },
        "issueqty" : 100.00000000,
        "issueraw" : 100,
        "subscribed" : false
    }
]
```

3.4.2 The Prototype of Blockchain-based Transaction Processing System

This study simulated the process of asset tokenization. As there are many types of assets, for simplicity they are divided into 1) cash coin, 2) divisible assets and 3) indivisible assets. For each issuance of new assets, Bb-TPS will automatically create a reference ID that can be linked to a real-world asset. For example, a building as an indivisible asset can be encoded in blockchain by binding its physical address to the corresponding token's reference ID. The issuer's address is also linked to the reference ID, which provides a proof of the first ownership. In this case, the indivisible assets could be encoded in blockchain one by one, while the tokenization of divisible assets needs to be done in a batch. For example, a factory creates a batch of inexpensive handbags that could be encoded in blockchain sharing the same reference ID. Cash coin is another type of divisible assets, which however could only be issued by central bank authorities. Table 4 shows the process of tokenization and the tokenized assets in "Achain." For the asset retirement, the Bb-TPS provides a specific wallet address where the users can send the tokens. Then, those tokens are out of circulation.

Table 4: Assets Tokenization Simulation

Asset Name	Reference ID	Issuer Address
Cash coin	213-267-35883	18FWJdyLDLdQU59EEg8UQHE3RZWAh6JCoxpnco
Handbags	192-266-38671	1RzU4qqiyaH6y6jDDQSJtAKNDqviJNgvn5Tiqe
Building 1 WP	784-643-67849	1HpDPEK8pnegtXVKdkzN67nEtwsMmmCs2JFsuv
…	…	…

In Table 4, there are at least three assets tokens. Asset Name describes the real-world names of the tokenized assets, Reference ID indicates the unique identifiers of the tokenized assets, and Issuer Address shows who initially encoded an asset into blockchain.

After simulating the cash coin and inventory – handbags, a transaction of the order-to-pay cycle was demonstrated as follows (Figure 12). First, we checked the firm M’s cash coin and inventory balance. As shown in Figure 12-1 and Figure 12-2, firm M has 19574.2 cash coins and 6 handbags, while its supplier has 425.8 cash coins and 14 handbags. Then, firm M ordered 6 handbags from the supplier, and the supplier shipped the goods immediately (Figure 12-3), which completed the transfer of the 6 handbags’ ownership from the supplier to firm M. As soon as firm M received the goods in the warehouse, it immediately disbursed the payment of 425.8 cash coins to the supplier (Figure 12-4). Finally, we checked the ending balance of both accounts after the procurement transaction (Figure 12-5 and Figure 12-6) and finished the procurement transaction.

<INSERT Figure 12 HERE>

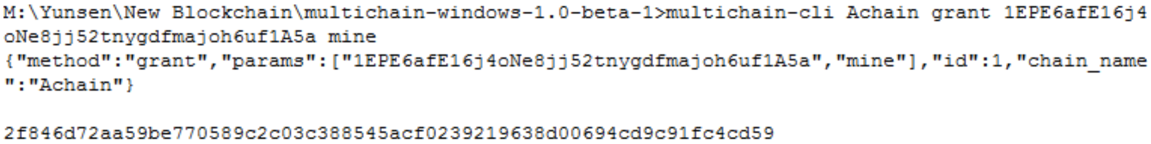
Figure 12: Order-to-Pay Transaction

3.4.3 Simulation of Permissions Management

If a user has been actively participating in Bb-TPS for an extended period, some of the “senior” users can consider promoting this user to be a maintainer or blockchain miner. As shown in Figure 13-1, a user was granted the permission of mining, and the user can participate in confirming and writing transactions to the blockchain. In Figure 13-2, this user is listed in the mining permission list, and it can start to mine the new blocks.

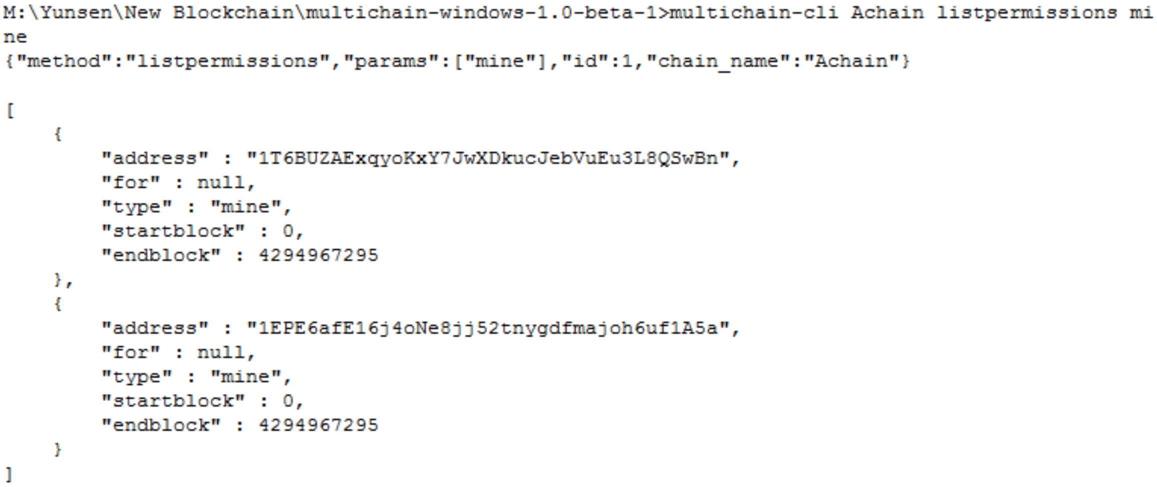
Figure 13: Granting Block Mining Permission

Figure 13-1. Granting Block Mining Permission



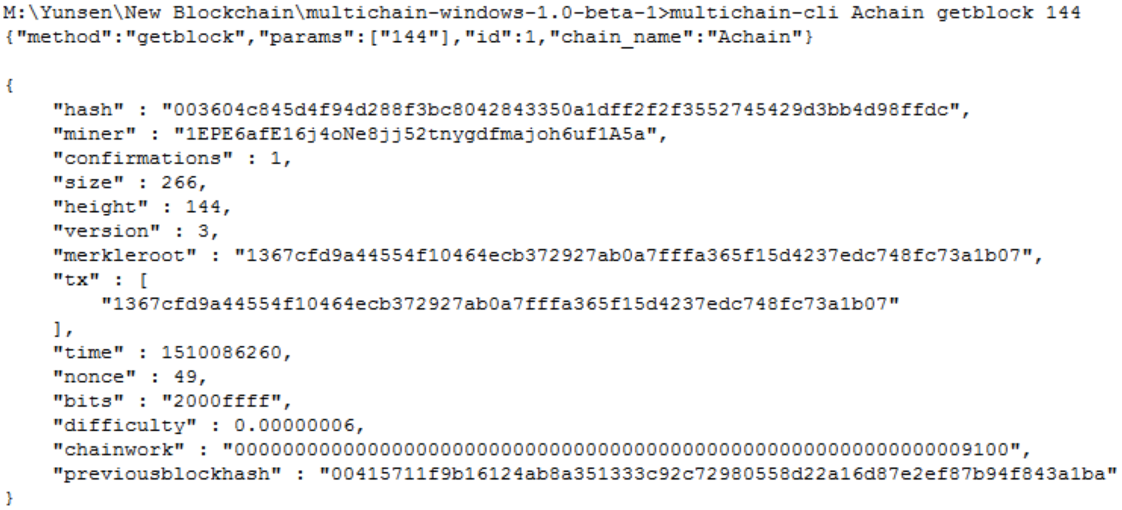
```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain grant 1EPE6afE16j4
oNe8jj52tnygdfmajoh6uf1A5a mine
{"method":"grant","params":["1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a","mine"],"id":1,"chain_name
":"Achain"}

2f846d72aa59be770589c2c03c388545acf0239219638d00694cd9c91fc4cd59
```

Figure 13-2. Mining Permission List



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain listpermissions mi
ne
{"method":"listpermissions","params":["mine"],"id":1,"chain_name":"Achain"}

[
    {
        "address" : "1T6BUZAExqyoKxY7JwXDkucJebVuEu3L8QSwBn",
        "for" : null,
        "type" : "mine",
        "startblock" : 0,
        "endblock" : 4294967295
    },
    {
        "address" : "1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a",
        "for" : null,
        "type" : "mine",
        "startblock" : 0,
        "endblock" : 4294967295
    }
]
```

Figure 13-3. Block Mined by One Node



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain getblock 144
{"method":"getblock","params":["144"],"id":1,"chain_name":"Achain"}

{
    "hash" : "003604c845d4f94d288f3bc8042843350a1dff2f2f3552745429d3bb4d98ffdc",
    "miner" : "1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a",
    "confirmations" : 1,
    "size" : 266,
    "height" : 144,
    "version" : 3,
    "merkleroot" : "1367cfd9a44554f10464ecb372927ab0a7fffa365f15d4237edc748fc73a1b07",
    "tx" : [
        "1367cfd9a44554f10464ecb372927ab0a7fffa365f15d4237edc748fc73a1b07"
    ],
    "time" : 1510086260,
    "nonce" : 49,
    "bits" : "2000ffff",
    "difficulty" : 0.00000006,
    "chainwork" : "0000000000000000000000000000000000000000000000000000000000009100",
    "previousblockhash" : "00415711f9b16124ab8a351333c92c72980558d22a16d87e2ef87b94f843a1ba"
}
```

Figure 13-3 shows a new block mined by a node. In this block, it lists the block parameters: hash, miner, height, nonce[26], difficulties[27] and mined time. The transaction records are recorded in the blocks and impossible to be changed in the future.

3.4.4 Simulation of Automatic Confirmation

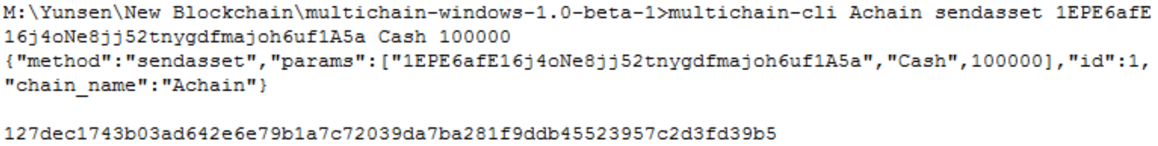
Figure 14-1 shows that in the order-to-pay cycle, a customer made a payment to firm M for goods received and Figure 14-2 shows the payment details. All users of Bb-TPS automatically confirmed this payment transaction, and the payment details can be used as evidence by the auditors. Furthermore, all the transaction details including goods shipment and payment between the customer and firm M can be collected and calculated in real time, and the accounts receivable, and payable between firm M and the customer can be easily calculated automatically.

[26] A nonce is a value that sets the hash of the block containing many leading zeroes.

[27] Difficulties is a measure of how difficult it is to find a nonce for a given target hash containing leading zeroes.

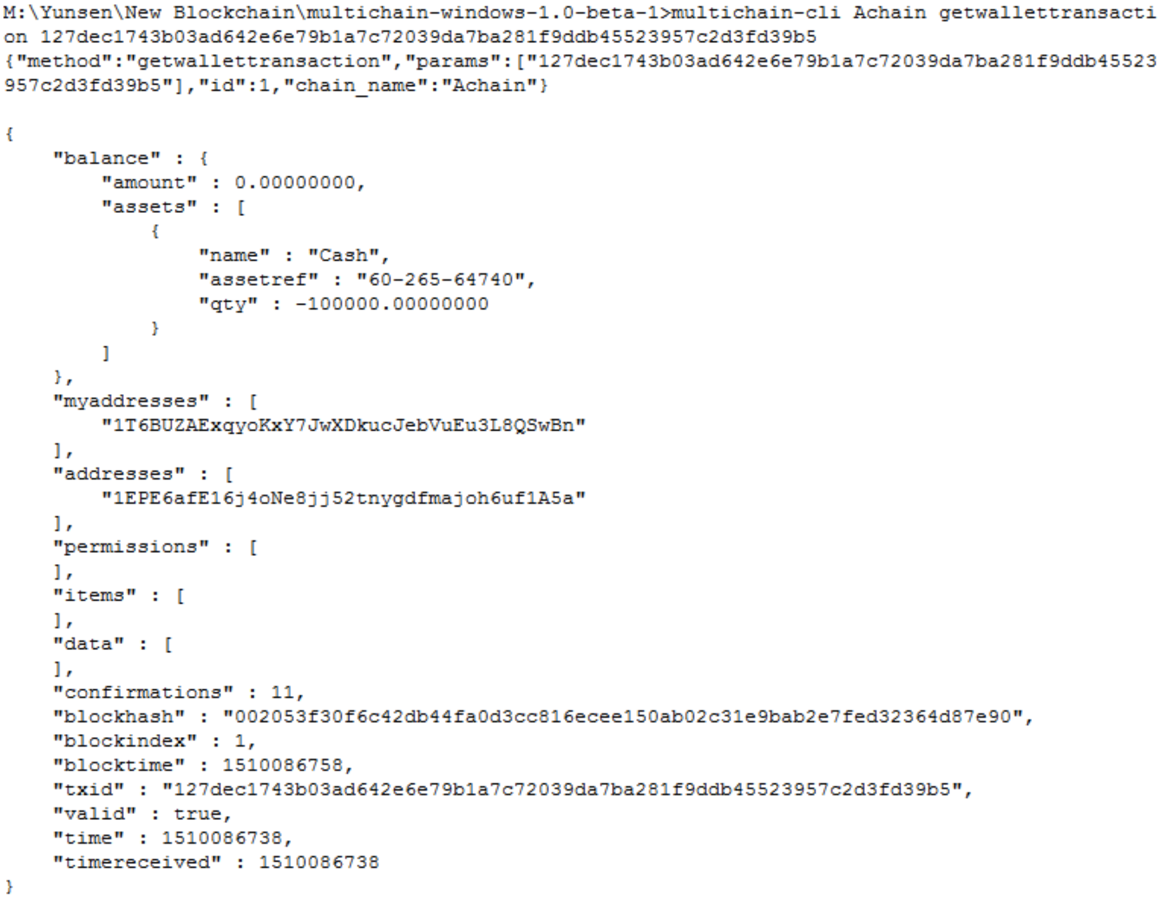
Figure 14: Simulation of Automatic Payment Confirmation

Figure 14-1. Payment for Handbag Delivery



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain sendasset 1EPE6afE
16j4oNe8jj52tnygdfmajoh6uf1A5a Cash 100000
{"method":"sendasset","params":["1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a","Cash",100000],"id":1,
"chain_name":"Achain"}

127dec1743b03ad642e6e79b1a7c72039da7ba281f9ddb45523957c2d3fd39b5
```

Figure 14-2. Payment Details



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain getwallettransacti
on 127dec1743b03ad642e6e79b1a7c72039da7ba281f9ddb45523957c2d3fd39b5
{"method":"getwallettransaction","params":["127dec1743b03ad642e6e79b1a7c72039da7ba281f9ddb45523
957c2d3fd39b5"],"id":1,"chain_name":"Achain"}

{
    "balance" : {
        "amount" : 0.00000000,
        "assets" : [
            {
                "name" : "Cash",
                "assetref" : "60-265-64740",
                "qty" : -100000.00000000
            }
        ]
    },
    "myaddresses" : [
        "1T6BUZAExqyoKxY7JwXDkucJebVuEu3L8QSwBn"
    ],
    "addresses" : [
        "1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a"
    ],
    "permissions" : [
    ],
    "items" : [
    ],
    "data" : [
    ],
    "confirmations" : 11,
    "blockhash" : "002053f30f6c42db44fa0d3cc816ecee150ab02c31e9bab2e7fed32364d87e90",
    "blockindex" : 1,
    "blocktime" : 1510086758,
    "txid" : "127dec1743b03ad642e6e79b1a7c72039da7ba281f9ddb45523957c2d3fd39b5",
    "valid" : true,
    "time" : 1510086738,
    "timereceived" : 1510086738
}
```

3.4.5 Computational Overhead: Blockchain vs. Database

The current accounting information systems (e.g., Enterprise Resource Planning) are designed on the basis of relational databases and database management systems (DBMS). Blockchain can be an alternative to keep business transactions records with high information integrity and low transmission cost. In order to compare the computational performance of relational databases and blockchain, this study conducts an experiment to simulate a large number of transactions and measure the computational overhead (i.e., computational time and data size) of the blockchain and database. This study uses the SQLite database to simulate transaction recording (SQLite is a C library that provides a lightweight disk-based database). Sqlite3 is the Python module that serves as the interface for the SQLite database. The transactions were automatically generated on Server 92, while Server 109 played the role of the trading partner. At the start of generating transactions, we monitored the account balance of the "trading partner" server and measured the time taken to complete these transactions in the blockchain, and also measured the time of recording these transactions in the SQLite[28] database on Server 92.

Table 5: Computational Overhead: Blockchain vs. Database

	Blockchain		**Database**	
Number of Transactions	**Computational Time (s)**	**Record Size (byte)**	**Computational Time (s)**	**Record Size (byte)**
1	3.71	570	0.00	201
10	19.70	5700	0.01	2010
100	193.07	57000	0.01	20100
1000	1927.13	570000	0.02	201000

Table 5 shows the comparison of computational overhead between the blockchain and database. This study simulated 1,000 transactions and recorded the corresponding computational time (in seconds) from initiation of the first transaction to completion of the last transaction in both blockchain and SQLite database. Table 5 also shows the data size of the records in both blockchain and SQLite database. We find that to record the same number of transactions the blockchain system needs to consume more computational overhead than the database system. Although more computational time is consumed to complete recording transactions in the blockchain than in the database, it increases linearly

[28] https://docs.python.org/3.6/library/sqlite3.html. Accessed 5/23/17 10:36PM.

in the number of transactions and does not seem to be cost-prohibitive. It is anticipated that with the rapidly improving information technology and decreasing computation cost, the blockchain scalability and computational resources should not be a big concern for accounting and auditing applications.

Blockchain provides an infrastructure for accounting information systems to keep records of business transactions with high information integrity and low transmission cost. However, to achieve the function of irreversibility and tamper resistance we need to trade in computational resources and runtime. However, it is expected that blockchain could be the most promising technology in accounting and fraud prevention in the near future.

3.5 Conclusion, Discussion and Future Research

The blockchain is a promising technology for real-time accounting and continuous monitoring. Essentially, it is a publicly shared database that keeps records of all transactions ever executed within. Based on cryptographic algorithms (e.g., digital signature and hash function), the blockchain protocol can guarantee data integrity that makes it impossible to tamper with the transaction history. The property of irreversibility and tamper resistance could be applied in auditing for continuous monitoring and fraud prevention. However, to successfully deploy blockchain in enterprise information systems and achieve high-level data tamper resistance requires a large number of participants who would have access to the full copy of every transaction. It is necessary to find a trade-off between the benefit of information sharing and the cost of weakening confidentiality, which motivates this paper to find a solution for protecting private information in a public blockchain.

Based on the recent technical innovation of zero-knowledge proofs, this paper proposes a design of the Blockchain-based Transaction Processing System (Bb-TPS) and demonstrates its functionalities of real-time accounting, continuous monitoring and permission management using a prototype (Figure 18). Furthermore, Bb-TPS uses the zk-SNARK scheme and homomorphic encryption to provide high-level confidentiality-preserving mechanisms. Finally, the comparative computational performance of blockchain and relational database in transaction recording is evaluated and discussed.

The accounting-blockchain convergence shows great promise for improving information integrity, decreasing transmission cost, increasing the speed of transaction settlement, and preventing fraudulent transactions. Furthermore, using zero-knowledge proofs and homomorphic encryption ensures data tamper resistance while preserving data confidentiality. Therefore, the deployment of blockchain enables the improvement of efficiency and effectiveness of accounting and audit practice. The limitation of this research is that it doesn’t specify the details of the block mining and rewarding mechanisms as well as the implementation of zk-SNARK for Bb-TPS. Although at the current stage the computational overhead of blockchain is still significant compared to that of the relational database, it is expected that technology improvements will result in cost reductions allowing blockchain to become a widely utilized infrastructure for enterprise information systems and continuous monitoring systems.

Chapter 4: Cloud-based In-memory Columnar Database Architecture for Continuous Audit Analytics

4.1 Introduction

In the era of big data, audit profession is starting to leverage the emerging data analytic techniques (e.g., deep learning, process mining) to examine financial data, evaluate internal control effectiveness, and detect fraudulent transactions. For example, process mining of event logs enables auditors to assess the weakness of internal control systems (Jans et al. 2014; Chiu et al. 2017); machine learning and deep learning algorithms provide tools to evaluate the riskiness of financial statement fraud (Perols et al. 2015). With high efficiency and high effectiveness, audit analytics have been recognized as the necessary tools for analytical procedures in the modern audit practice.

In order to apply audit analytics to examine a client's business data, an auditor needs to periodically extract the full population of transactions (e.g., purchase orders, invoice receipts) from the client's Enterprise Resource Planning (ERP) system. In order to examine such high-volume transactions continuous auditing systems (Vasarhelyi and Halper 1991) are designed to extract transaction data from ERP systems systematically, test every transaction entry, and report exceptions or anomalies in close to real time (Alles, Kogan, and Vasarhelyi 2008; Kogan et al. 2014). The higher the frequency of the data access is, the timelier the financial and audit report will be; however, the more computing and communication resources it will consume (Pathak Chaouch and Sriram 2005).

An alternative solution is to embed audit modules in ERP systems (Groomer and Murthy 1989) and provide external auditors with direct access to the transaction data from in the operational database. However, the queries that select data from the operational database and perform complex audit analytics can easily overload ERP systems and disrupt the regular transaction processing (Chaudhuri and Dayal 1997). Therefore, in order to conduct real-time and continuous audit analytics based on artificial intelligence (AI) algorithms that are computationally costly, it is necessary to build a high-speed and high-volume data processing infrastructure to support continuous audit analytics.

The in-memory columnar database system is such an infrastructure that supports high-speed data analytics using main memory as the primary storage (Garcia-Molina and Salem 1992; Plattner 2009). Thanks to the fast improvement of computer and information technology, a new generation of ERP[1] based on the in-memory columnar database (e.g., SAP S/4 HANA cloud ERP[2] is starting to affect today's practice of continuous auditing and audit analytics. For example, SAP S/4 HANA is designed based on the in-memory columnar database to enable high-speed applications of AI and machine learning algorithms and textual analysis. The in-memory columnar database can potentially provide speedy performance regarding data storage and access (i.e., write and read) and data analytics (i.e., deep learning, regression, classification and clustering).

This study introduces the architecture of the modern in-memory columnar database system and proposes a design of applying the new database for high-speed continuous audit analytics. In order to evaluate the performance of the in-memory columnar database system for continuous audit analytics, this study conducts a simulation test to measure the computational overhead in comparison with a conventional relational database system.

[1] https://www.sap.com/products/erp/s4hana-erp.html. Accessed 7/19/17 10:02PM.

This study is organized as follows. The second section motivates this research and explains the need for applying in-memory columnar databases for continuous audit analytics. The third section reviews the prior studies of continuous auditing. The fourth section describes the existing architecture of ERP and data warehousing. The fifth section introduces the techniques of in-memory computing and columnar storage, and the sixth section proposes an artifact of applying the in-memory columnar database system for continuous audit analytics. The seventh section creates a prototype and evaluates the performance of in-memory computing and columnar storage. The eighth section deploys this system on cloud computing. Finally, the last section discusses further research opportunities and summarizes the contributions of this research.

4.2 Motivation and Literature Review

4.2.1 Motivation

ERP systems have become the core of modern enterprise infrastructure, as they provide business event recording, message communication and data storage of a company. In general, an ERP system is mainly designed for two data processing tasks: online transactional processing (OLTP) and online analytical processing (OLAP). OLTP is the task that processes details of the individual transaction, such as order entry and banking transactions; and OLAP is a decision-support task that processes summarized and aggregated data for complex queries. A company's transaction data is continuously generated in the OLTP system and periodically extracted out for OLAP or to be loaded into an auditor's external database. An auditor performs data analytics on the company's business and financial data and provides independent assurance that "the financial statements are presented fairly in all material respects." The assurance quality primarily relies on data quality and extraction frequency. The higher the frequency of extraction is, the better the information quality will be, and the higher the audit quality the auditor can provide.

However, the extraction process cannot be performed continuously because too frequent extractions would be prohibitively expensive and disturb the regular transaction processing (Chaudhuri and Dayal 1997). Real-time reporting and continuous audit analytics require the immediate availability of transaction data to support real-time decision-making. The need for timeliness and speed when conducting complex audit analytics (e.g., AI, deep learning) motivates this study to search for information technology-based breakthroughs. By utilizing in-memory columnar databases for continuous audit analytics, the periodical extraction process can be improved to become real-time and continuous. Therefore, fast data aggregating for generating financial reports and performing audit analytics can significantly improve audit efficiency and effectiveness. Furthermore, keeping the original data in the company's data center would make it easier to better protect information security as opposed to extracting that data out. This study will introduce the in-memory columnar database system and propose an artifact of applying it for high-speed continuous audit analytics.

4.2.2 Continuous Auditing, Audit Analytics and Artificial intelligence

Since Vasarhelyi and Halper (1991) initially developed the first practical continuous audit systems, the research of continuous auditing has progressed with numerous advances, such as innovations of continuity equations (Kogan et al. 2014) and exceptions and anomalies detection (Issa and Kogan 2014). There are many studies that document the implementations of continuous monitoring (Alles et al. 2006) and continuous auditing (Alles, Kogan, and Vasarhelyi 2008). Furthermore, the applications of continuous auditing in ERP systems (Kuhn Jr and Sutton 2010) and using XML (Murthy and Groomer 2004) have been examined, and the economics of continuous assurance (Alles, Kogan, and Vasarhelyi 2002) has been discussed in prior studies.

The goal of continuous auditing is to design an automatic and real-time systematic auditing system to provide auditing with high efficiency and at low cost. The in-memory columnar database system brings new opportunities for continuous auditing in terms of high volume data and high-speed computing. Furthermore, in-memory columnar databases provide the most efficient and effective infrastructure for fast book closing, data aggregating and timely continuous financial and audit reporting. Although the application of in-memory columnar databases in auditing is still in its infancy, it provides great promise for modern audit practice, especially for real-time and continuous audit analytics with guaranteed data security.

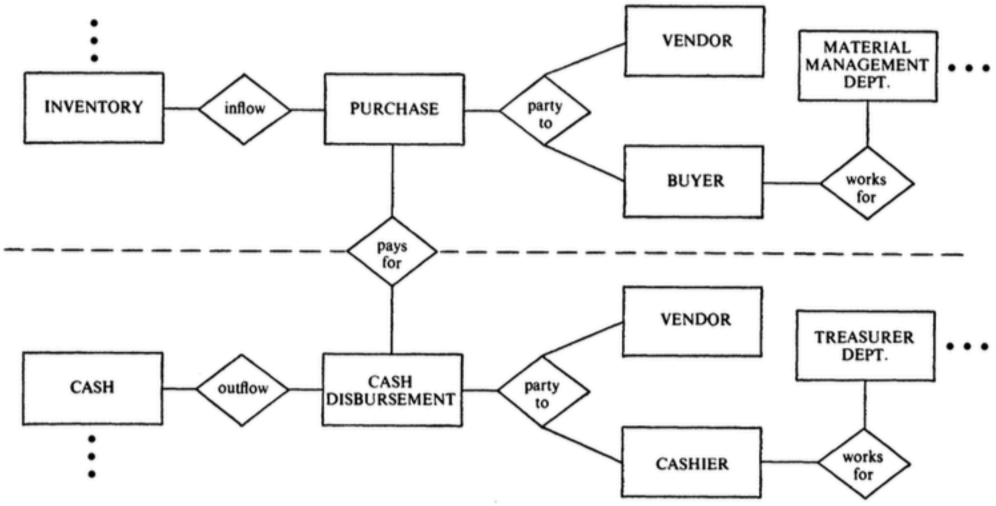
4.2.3 Database Systems and Enterprise Resource Planning

Relational databases were invented in computer science in 1970, and have had the attention in the information technology domain for decades (Selinger et al. 1979). They have been widely implemented as the essential infrastructure to support management information systems. Initially motivated by managers' need for timely access to a company's business situation, such as the inventory level of finished products and raw materials, information system engineers used relational databases to create inventory management systems, sale, and marketing systems and supply chain management (SCM) systems. In order to create an efficient channel for internal personnel communications and/or external supplier or customer communications, they designed the human resources (HR) systems, customer relationship management (CRM) systems, and procurement management systems. All systems within an enterprise are integrated into a complex ERP system (O'Leary 2000), and each functional system serves as a software module, such as CRM module, HR module and accounting and finance module to improve data quality and integrity

Resource-Event-Agent (REA) ontology (McCarthy 1982) is an ISO[3] standard framework (ISO/IEC 15944-4:2015) based on relational databases and used to design accounting information systems. McCarthy (1982) designs the REA ontology to describe business events and model the business event data storage. Figure 1 shows the framework of REA ontology. Business objects are categorized as either resource, event or agent. Resource represents any enterprise resource, such as inventory, cash; event represents all the business transaction events, such as purchase, cash disbursement; an agent represents either the external or internal entity associated with an enterprise's business activities. As shown in Figure 15, in a procure-to-pay cycle, the business events (i.e., purchase order created, goods receipt and invoice payment) are located in the middle and are connected to the business resources (i.e., inventory and cash) and business agents (i.e., vendor, procurement agent, and cashier).

[3] https://www.iso.org/standard/67199.html. Accessed 7/21/2017 12:02 AM.

Figure 15: An Example of Accounting Database Design – REA Ontology

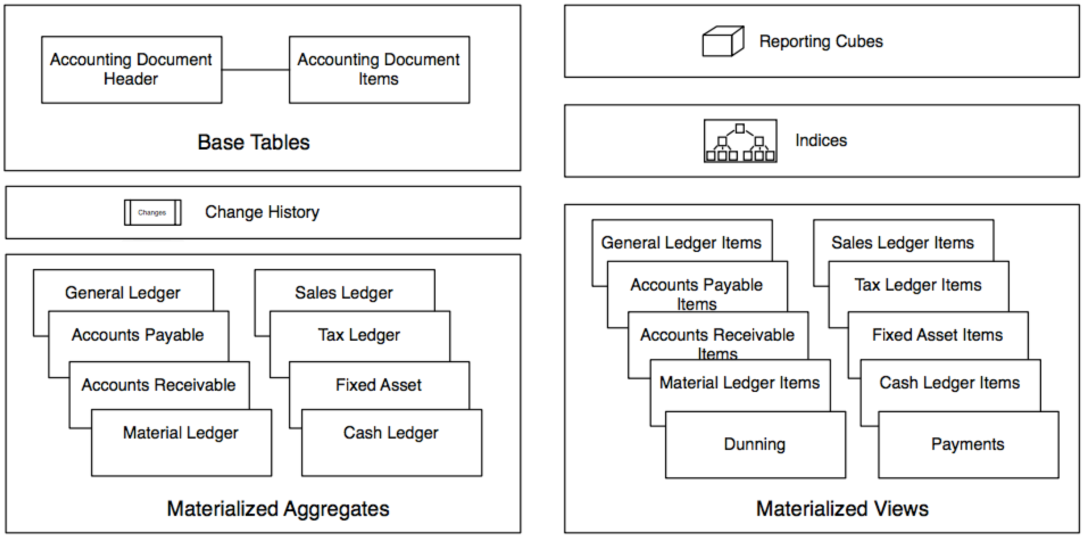


Source: McCarthy 1982

The REA model has been widely adopted in designing accounting database systems. It provides the structure of the base tables storing event-oriented business data about enterprise resources, business events, and relevant agents. Since historical data is logged and archived, it provides a traceable source for data analysis and audit purposes. REA-based accounting database systems capture the wide range of economic event instances, yet they are not designed to store the double-entry accounting data associated with journals or ledgers, such as debit, credit, receivables, or other account balances. Instead, these elements are derived from the REA-based accounting database systems. The process of deriving these elements from accounting database systems is called conclusion materialization (McCarthy 1982). For generating a complete financial report from an ERP system, it is necessary to create the base tables to store the unfiltered transaction data and master data, to create event log tables to record historical data, and to create conclusion materialization tables to store the derived accounting artifacts, such as ledgers and accruals.

Figure 16 shows the example of an extant ERP system that includes base tables, historical logs, materialized files and analytical data warehouse (to be explained in the next section).

Figure 16: An Example of Extant ERP System

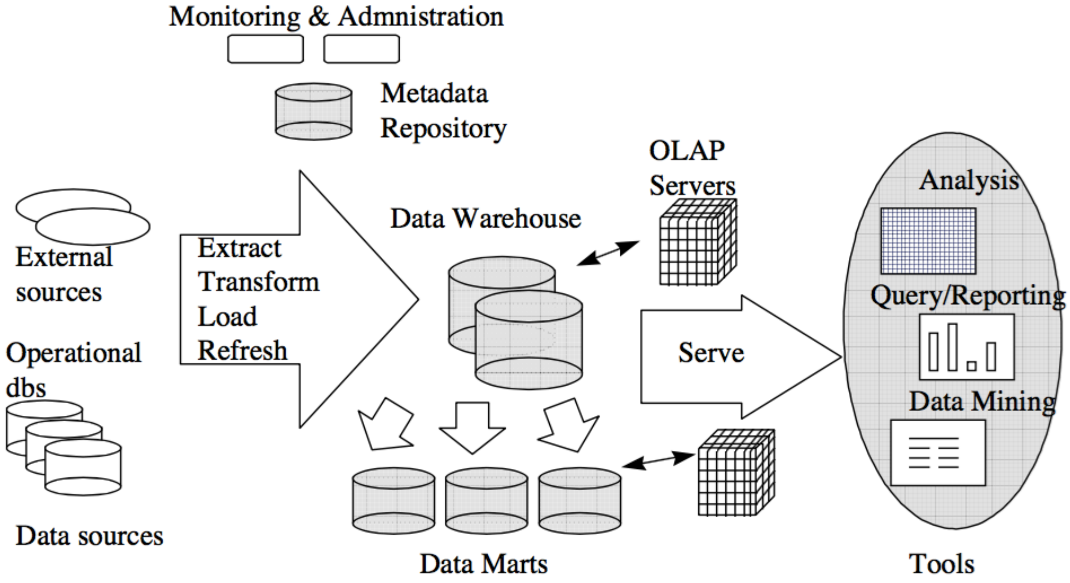


Source: Plattner 2009

4.2.4 Operational Database and Data Warehouse

In order to support ad-hoc complex queries for data analytics, a data warehouse is designed and implemented separately from the operational database (i.e., base tables and conclusion materialization tables). Figure 17 shows the data warehouse that is separated from the operational database to perform more flexible on complex analytical queries. The data warehousing mechanisms work as follows: first, business transaction data are copied and extracted from the operational database to be combined with external data sources (e.g., big data, social media, RFID, weather data); second, the data is transformed and loaded into the data warehouse; third, if there are multiple divisions in a company, the data could be disaggregated into data marts to support different divisions' decision making needs; fourth, the data stored in the warehouse are organized as multi-dimensional arrays of values, called data cubes; and last, it is necessary to refresh and update the data stored in the warehouse periodically. To sum up, data warehousing is designed to extract and integrate data from multiple sources and store them as data cubes to support complex analytical queries.

Figure 17: Data Warehousing Architecture



Source: Chaudhuri and Dayal 1997

The operational database is designed and optimized to support OLTP workloads while the data warehouse is designed and optimized to support OLAP workloads. Therefore, "*to execute complex OLAP queries against the operational databases would result in unacceptable performance*" (Chaudhuri and Dayal 1997). For example, if an auditor directly and continuously uses the transaction data on all POS machines to predict a company's revenue, the computation of revenue estimations will create a significant additional burden for the operational database and even interrupt the regular transaction data processing.

Table 6 shows the difference between OLTP and OLAP. With more complex business needs, there is more focus on OLTP database design, but less care about OLAP.

OLTP tuples are arranged in rows which are stored in blocks, so indexing does not work well when the number of requested tuples increases. OLAP warehouse organizes data in star schemas, which is a popular optimization to compress attributes (columns) with the help of dictionaries. More recently, the use of columnar databases for analytics has become quite popular (Plattner 2009).

Table 6: Online Transaction Processing and Online Analytical Processing

	Online Transaction Processing (OLTP)	**Online Analytical Processing (OLAP)**
Data Characteristics	Detailed Individual records	Summarized Consolidated Historical
Storage Requirement	Consistency Recoverability	1. Orders of magnitude larger than operational databases 2. From several operational databases over an extended period 3. Tools needed for extracting, cleaning and loading data into a data warehouse

The advantage of creating a data warehouse is to reduce the computational burden on the operational database and maintain its data integrity. Moreover, data from external sources can be integrated and used for audit analytics. If indexing or locking mechanisms are not configured properly, performing audit analytics directly on the operational database could corrupt the business transaction processing and damage the whole ERP system. However, the disadvantages are 1) it creates additional management of extracting, transforming and loading data and 2) creates data redundancy in the warehouse. It is necessary to store the original transaction data in the base tables only once and remove the data warehouse as an intermediary to reduce data redundancy and improve data analytics efficiency

4.3 Technology Breakthroughs and Continuous Audit Analytics

4.3.1 In-Memory Database System

The fast improvement of ERPs is enabled by two major information technology breakthroughs: high performance Massively Parallel Processing (MPP) and unprecedented growth in the amount of main memory. MPP is the core technology that supports big data processing and analysis. In order to process complex queries on high-volume data promptly, MPP breaks up the large dataset into manageable chunks and assigns them to many processors (Batcher 1980). Then, it combines the results from all assigned processors through a communication interface. With the fast processing technique based on multi-core CPUs, it becomes straightforward to deal with large datasets and execute complicated queries very fast.

In Von Neumann architecture, the main memory is the primary storage, and the magnetic disk is the secondary storage (Von Neumann 2012). In a conventional database management system (DBMS), data resides permanently in the secondary storage (i.e., hard disk). When the disk data is needed, it will be first cached into the primary storage (i.e., main memory) to be fast accessed by CPUs. Table 7 shows the different properties of main memory and magnetic disk. In general, data stored in main memory could be accessed and processed orders of magnitude faster than data stored on magnetic disk. However, main memory has much less capacity to store data than magnetic disk[4], and it requires uninterruptable power supplies to maintain the integrity of stored data, which is called storage volatility.

[4] For example, a regular server system has 64 GB main memory and 4 TB hard disk drive.

Table 7: Differences between Main Memory and Magnetic Disk

	Main Memory	Magnetic Disk
CPUs Access Time	Direct access	Indirect access
Processing Time	Faster in orders of magnitude	Slower in orders of magnitude
Power Supplies Requirement	Uninterruptable power supplies	None
Storage Arrangement	Not block-oriented storage	Block-oriented storage
Data Access method	Random access	Sequential access
Storage Capacity	Smaller in orders of magnitude	Larger in orders of magnitude

With the recent breakthroughs in hardware technology and cloud computing (Mell and Grance 2011), the vast availability of main memory and wide-bandwidth network allow storing big data in the primary storage for fast access by processors. Moreover, new storage products, such as solid-state drives (SSD), provide faster and more reliable alternatives for secondary storage. In a conventional DBMS, data resides permanently in hard disk and will be loaded into main memory when needed, while in the modern in-memory database system (IMDB), data resides permanently in main physical memory. As multi-core CPUs can directly access data in main memory, IMDB has a better response time and transaction throughputs (Plattner 2009).

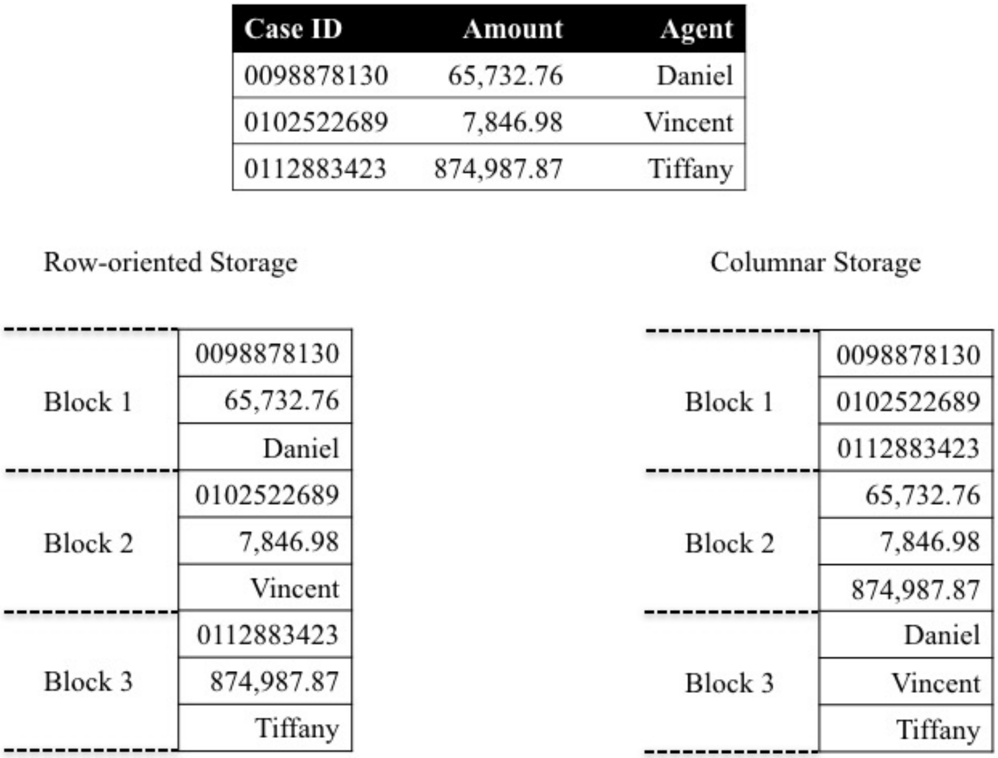
In order to solve the storage volatility problem when the power is shut off, IMDB writes snapshots of main memory to disk at frequent intervals. In between snapshots, IMDB keeps a log of the changes to various secondary storage devices (e.g., magnetic disk, flash drive, SDD). If the power is interrupted, IMDB will be able to quickly recover the last snapshot and the data change log in order to ensure data consistency. Therefore, an IMDB is also configured with hard disk drives (HDD) that, in this case, are used only for transaction logging and snapshots for data backup. Although for a conventional DBMS and modern IMDB a given dataset still has two copies in both main memory and magnetic disk, DBMS and IMDB have different optimization schemas for the database design[5] (Garcia-Molina and Salem 1992), which distinguishes IMDB from DBMS in terms of high-speed data access and processing.

4.3.2 Row-oriented Storage and Columnar Storage

Although DBMS is created to store data tables by rows, an increasing number of recent database systems including IMDBs and DBMSs are configured to store data tables by columns. Figure 18 shows the comparison between row-oriented databases and columnar databases. A relational database system stores the data that is represented as two-dimensional tables arranged by rows and columns. For example, a table has three attributes/columns (i.e., Case ID, Amount and Agent) and has three records or rows. In the physical implementation, the values in all cells are serialized and stored in fixed-size hard disk blocks. Row-oriented storage refers to the arrangement in the hard disk that all attributes of a record are stored contiguously in a block, while column-oriented storage refers to the arrangement that all values of an attribute are stored together in a block.

[5] Random data access in memory will be much faster than the indexed sequential data access in hard disk.

Figure 18: Row-oriented Storage and Column-oriented Storage



Case ID	Amount	Agent
0098878130	65,732.76	Daniel
0102522689	7,846.98	Vincent
0112883423	874,987.87	Tiffany

Row-oriented Storage

Block 1	0098878130
	65,732.76
	Daniel
Block 2	0102522689
	7,846.98
	Vincent
Block 3	0112883423
	874,987.87
	Tiffany

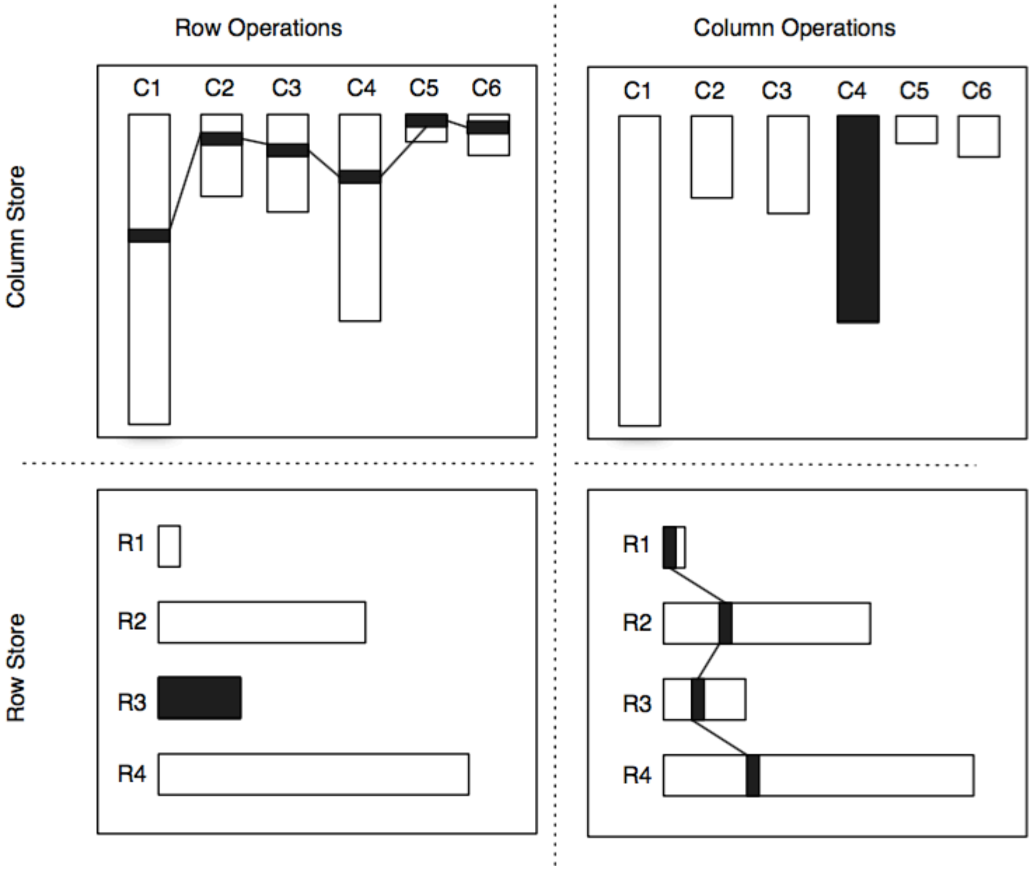
Columnar Storage

Block 1	0098878130
	0102522689
	0112883423
Block 2	65,732.76
	7,846.98
	874,987.87
Block 3	Daniel
	Vincent
	Tiffany

In contrast to conventional row-oriented storage, the values for each attribute are stored contiguously in the column-oriented storage; therefore, its compression efficiency is usually 4 to 5 times that of row-oriented storage (Abadi, Madden and Ferreira 2006). Moreover, a complex analytical query could be fast responded to as data aggregation in columnar storage outperforms row-oriented storage, especially in the case of a large number of data items. Join operations can be directly performed on columns (Abadi et al. 2009). Moreover, many new join algorithms were introduced for columnar storage, such as the “FlashJoin” (Tsirogiannis et al. 2009) and the “invisible join” (Abadi et al. 2009), which achieved substantial performance advantages of columnar storage over row-oriented storage. By applying columnar storage schemas to IMDB, the new in-memory columnar database would be superior to row-oriented IMDB with regards to memory consumption.

Apart from the arrangement of data storage, the data operation is either in a row or column styles. Figure 19 shows the data operation by row or column on both row storage and column storage. For column storage, if a query is to select all attributes of a record (row operation), it needs to visit all hard disk blocks; if a query is to select all values of an attribute (column operation), it only needs to visit a hard disk block. On the other hand, for row storage, if a query is to select all attributes of a record (row operation), it only needs to visit a hard disk block; if a query is to select all values of an attribute, it needs to visit all hard disk blocks.

Figure 19: Data Operation by Row and Column

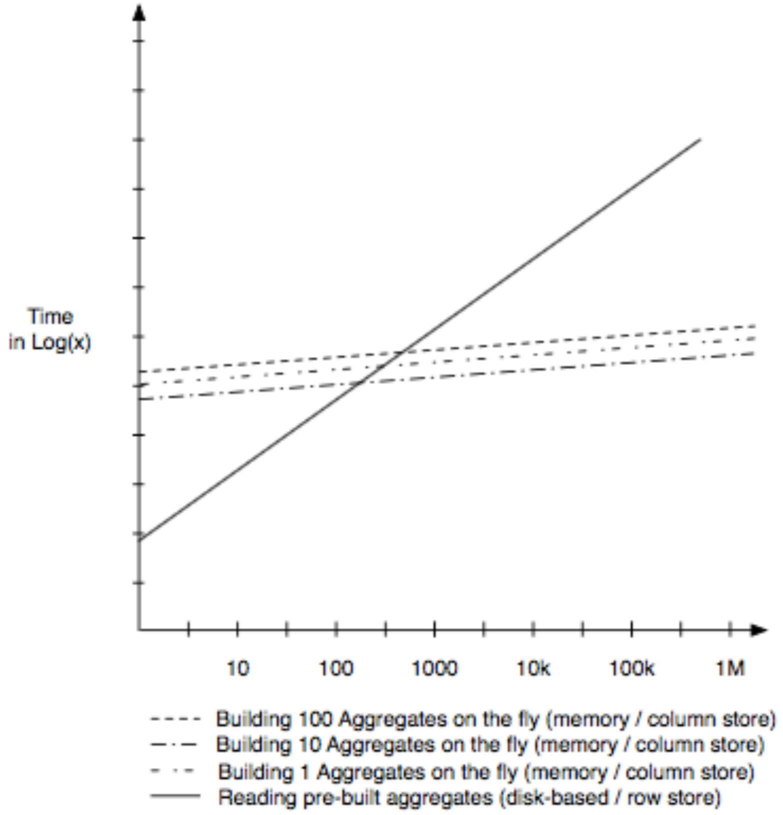


Source: Plattner 2009

The columnar database is a read-optimized and analytics-optimized system for OLAP applications, and the row-stored database is a write-optimized and transactions-optimized system for OLTP applications. Usually, operational database stores indexed data by rows on disk to be cached in main memory, and data warehouses use star schemas or column storage to compress and aggregate attributes. Although the columnar database is row-operation (e.g., inserting and updating) costly, recent findings show that in the main memory columnar database works best for multi-core CPUs and update-intensive applications. Many DBMS vendors (e.g., Microsoft[6]) have enabled columnar indexes that allowed to perform real-time data analytics on an OLTP workload. *"Having all data in main memory greatly improves the data update performance"* of column stores (Plattner 2009). As the business and accounting data is usually required to use insert-only approach, the in-memory columnar database should also be usable as an operational database. As shown in Figure 20, the computation of data aggregation is much faster for the in-memory columnar database than for conventional disk-based row-store database.

[6] https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview. Accessed 4/3/2018 11:23 PM.

Figure 20: Computational Performance of In-memory Columnar Database



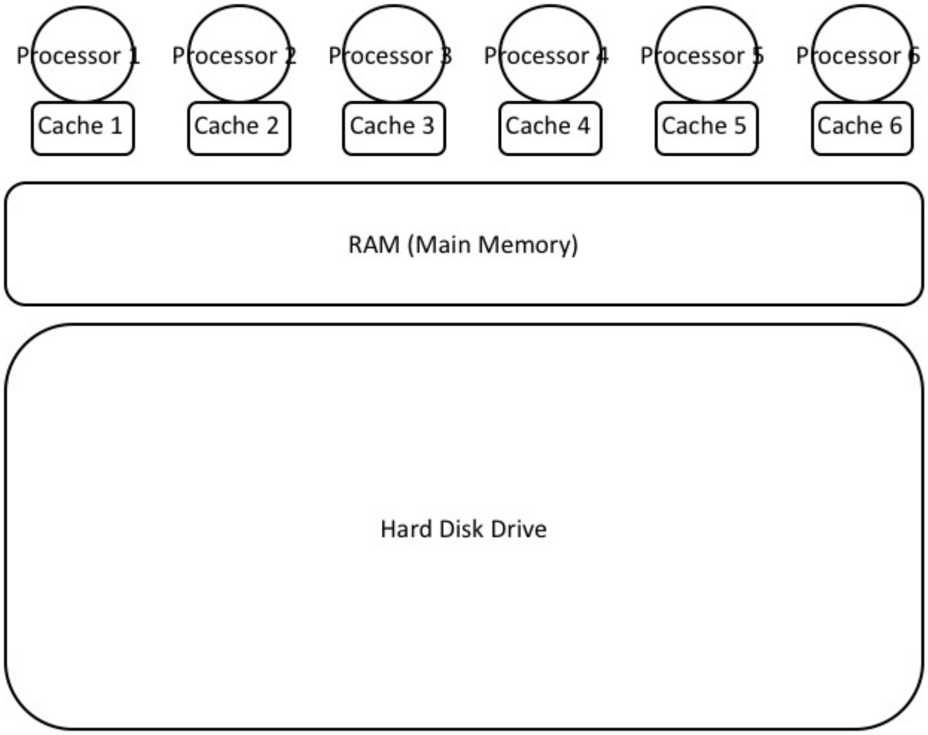
Source: Plattner 2009

4.4 Design: In-Memory Columnar Database for Continuous Audit Analytics

Instead of separating audit analytics from the ERP system, utilizing in-memory columnar database system enables auditors to build analytical applications directly on top of the transaction data for real-time and continuous audit analytics. It was a suboptimal design for conventional DBMS because the OLAP could overwhelm operational database and disturb regular OLTP applications. However, it will be ideal to create an application layer of audit analytics based on the new infrastructure for high-speed data access and aggregation. Columnar data is stored in a highly compressed format in order to improve the efficiency of memory usage and to speed up the data transfer from main memory to processors. Moreover, it also tremendously increases the speed of aggregation for data analytics.

Figure 21 shows the underlying hardware architecture for utilizing in-memory columnar database systems. This hardware architecture is configured with multi-core CPUs and large amounts of main memory together with various secondary storage devices, such as HDD, SSD and flash storage. Furthermore, cluster configuration schema allows to group many hardware systems to scale to orders of magnitude and to execute in parallel and main-memory-centric environments (Weyerhauser et al. 2008).

Figure 21: Hardware Architecture



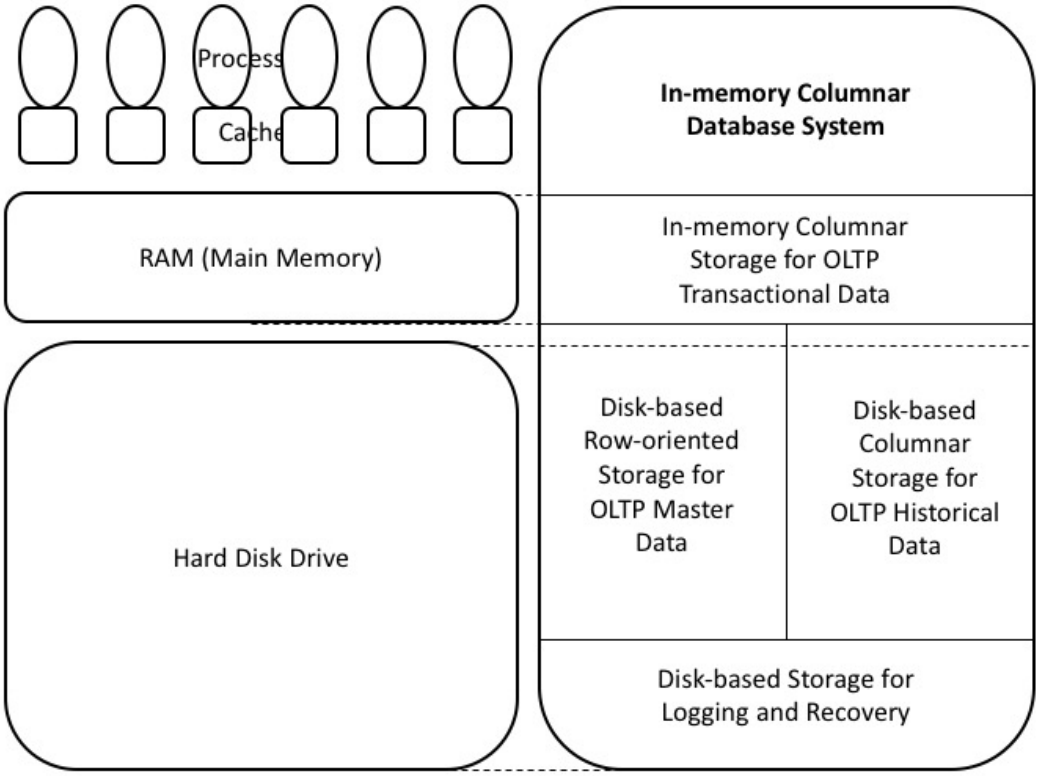
Some types of data would be visited more frequently than some other types to write in or read out the data in an ERP. For example, current transaction data would be queried more often than historical data. Table 8 shows the level of data access frequency. Transaction data such as purchase data and cash disbursement data would be more frequently visited than the master data stored in resource and agent tables. The master data would be more frequently visited than the historical data, such as event log. Therefore, transaction data should be stored in a columnar format to benefit from high compression rate and the highly optimized access for aggregation queries (Farber et al. 2011). Master data and historical data with less frequent access should be placed on the hard disk by rows or columns.

Table 8: Level of Data Access Frequency

Data Types	**Data Example**	**Access frequency**
Transaction data	Purchase data Cash disbursement data	High
Master data	Inventory Cash Vendor Customer Employee	Medium
Historical data	Event logs Last year's mater data The year before last year's transaction data	Low

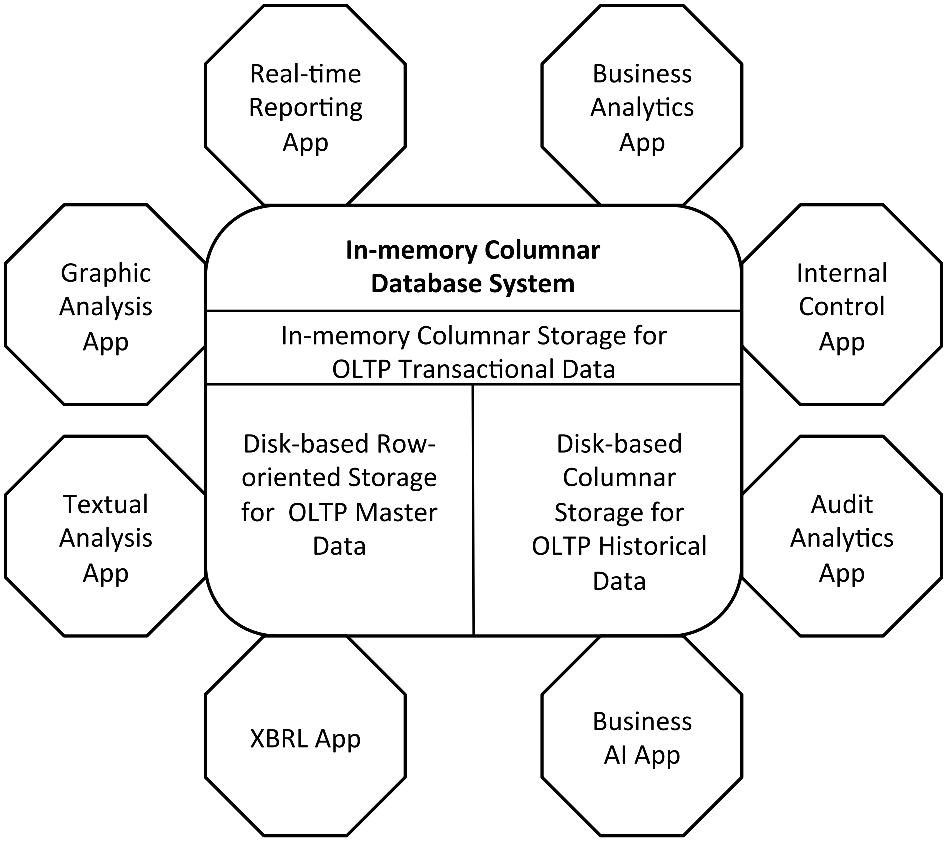
After building the hardware foundations and analyzing the data access frequency, this study designs an architecture of the in-memory columnar database. The main memory will be used to store the OLTP transaction data in the form of columnar storage, and the secondary storage will be used to store the OLTP master data in the form of row-oriented storage for update-intensive applications and to store the OLTP historical data in the form of columnar storage for high-rate of compression. Logging and recovery files for data backup should also be stored in the secondary storage.

Figure 22: Software Architecture


Process
Cache
RAM (Main Memory)
Hard Disk Drive
In-memory Columnar Database System
In-memory Columnar Storage for OLTP Transactional Data
Disk-based Row-oriented Storage for OLTP Master Data
Disk-based Columnar Storage for OLTP Historical Data
Disk-based Storage for Logging and Recovery

Transaction data will be generated automatically once a business event occurs and then stored permanently in the main memory in the form of columnar storage. As shown in Figure 23, in the application-to-database round trips continuous audit analytics can be designed as an application layer that communicates with the "single fact" database by continuously sending complex data analytical queries and receiving in response the result sets from the in-memory columnar database.

Figure 23: Proposed Design


Real-time Reporting App
Business Analytics App
Graphic Analysis App
In-memory Columnar Database System
Internal Control App
In-memory Columnar Storage for OLTP Transactional Data
Textual Analysis App
Disk-based Row-oriented Storage for OLTP Master Data
Disk-based Columnar Storage for OLTP Historical Data
Audit Analytics App
XBRL App
Business AI App

The higher the level of computing complexity and the higher the frequency of the queries are, the higher the needs for high-speed data access and processing will be. Table 9 shows the computing complexity levels of different audit analytical techniques. The higher the level of computational complexity, the higher the frequency of data aggregation will be, and companies and auditors will get more benefit from using the in-memory columnar database. With highly compressed storage and efficient data processing, this architecture could allow big data applications to support sophisticated machine learning and deep learning algorithms on high volume and unstructured datasets (e.g., video, graph and text).

Table 9: Computational Complexity Level of Audit Analytics

Computational Complexity Level	**Examples of Data Analytics Algorithms**	**Complexity Measurement by numeric operations**
Level One	Search Summation Balancing Netting	Simple logic operations Addition Subtraction Multiplication
Level Two	Transaction aggregation Financial reporting generation Segregation of duty check Three-way match	Complex logic operations Multidimensional addition Multidimensional subtraction Vector multiplication
Level Three	Descriptive statistics Hypothesis testing Basic visualization	Multidimensional operations
Level Four	Linear regression Clustering Classification	Matrix multiplication Matrix optimization High-dimensional optimization
Level Five	Multivariate time series models (e.g., MVAR) Deep learning (e.g., ConvNet, RNN) Text mining Advanced visualization	High-dimensional optimization Stochastic gradient descent Dynamic programming Natural language processing Graphic processing

As all operational data resides in main memory, conclusion materialization can be performed on the fly immediately with high efficiency. In this case, no data warehouse needs to be created separately from operational databases because selection, aggregation, and analysis can be performed very efficiently in main memory. The redundancy and complexity of extant ERP systems are reduced to a couple of base tables, which drastically shrink the number of tables that constitute the “single fact” database. To sum up, storing data in main memory substantially speeds up data processing. Also, columnar storage provides high-rate compression to save main memory usage and fast aggregation in response to complex analytical queries.

4.5 In-memory Columnar Database Performance Evaluation

This study builds a prototype and evaluates the computational performance by conducting a simulation test to validate the proposed design of utilizing the in-memory columnar database for continuous audit analytics. This study uses R software[7] as the primary platform, "SQLite"[8] library as the DBMS for disk-based or in-memory row-oriented storage simulation and "MonetDB"[9] library as the DBMS for disk-based columnar storage simulation.

Due to the lack of open-source software implementing an in-memory columnar database, this study is not able to directly compare the performance of proposed in-memory columnar database with the conventional disk-based row-oriented database. However, this study is able to simulate and compare the performance of three different database systems: 1) disk-based row-oriented database (the infrastructure for conventional ERP), 2) in-memory row-oriented database, and 3) disk-based columnar database (the infrastructure for new storage on disk).

First, this study creates some artificial transactions to be stored in those three databases. The number of artificial transactions varies from 100,000 to 5,000,000, and each transaction is represented by five numeric attributes. After creating the tables storing the transaction data, the tables will be immediately stored on disk or in memory. This study measures the sizes of two groups of databases on disk (i.e., disk-based row-oriented database and disk-based columnar database). As shown in Figure 24, the sizes of these disk-based databases grow linearly in the number of transactions; however, the storage size

[7] https://www.rstudio.com. Accessed 8/12/2017 1:26 PM.
[8] https://www.sqlite.org. Accessed 8/12/2017 1:27 PM.
[9] https://www.monetdb.org/Home. Accessed 8/12/2017 1:28 PM.

of the disk-based columnar database grows relatively faster than a disk-based row-oriented database. An explanation is that MonetDB is a relatively new database system that might be less optimized in terms of storage size on disk. Another explanation is that the compression mechanism of columnar storage should work in the main memory instead of on disk.

Figure 24: Comparison of Storage Size

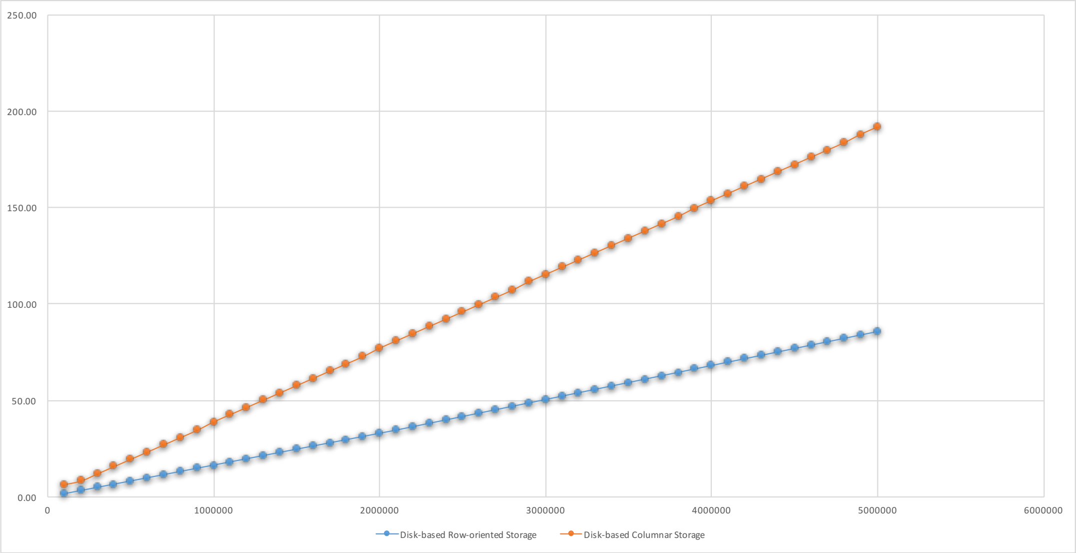


Figure 25 shows the comparison of computation time among the three different database systems (i.e., in-memory row-oriented database, disk-based row-oriented database and disk-based columnar database). This study measures the computation time of querying every database for aggregated information, such as the sum, average and mean of each attribute[10]. As shown in Figure 25, both computation time of the disk-based row-oriented databases and in-memory row-oriented database grow linearly in the number of transactions. Meanwhile, disk-based columnar oriented databases perform much better as it increases much slower than the other two databases, which shows higher efficiency in

[10] The number of artificial transactions simulated increases from 100,000 to 5,000,000, and each transaction is represented by attributes of numeric values, and then this study measures the computation time of querying each database for aggregated information, such as the sum, average and mean of each attribute

terms of aggregation queries[11]. There are some spikes in the computation timeline, which may be because the experiment was conducted on a shared server. Therefore the CPUs might not be able to serve only this analysis.

Figure 25: Comparison of Computation Time

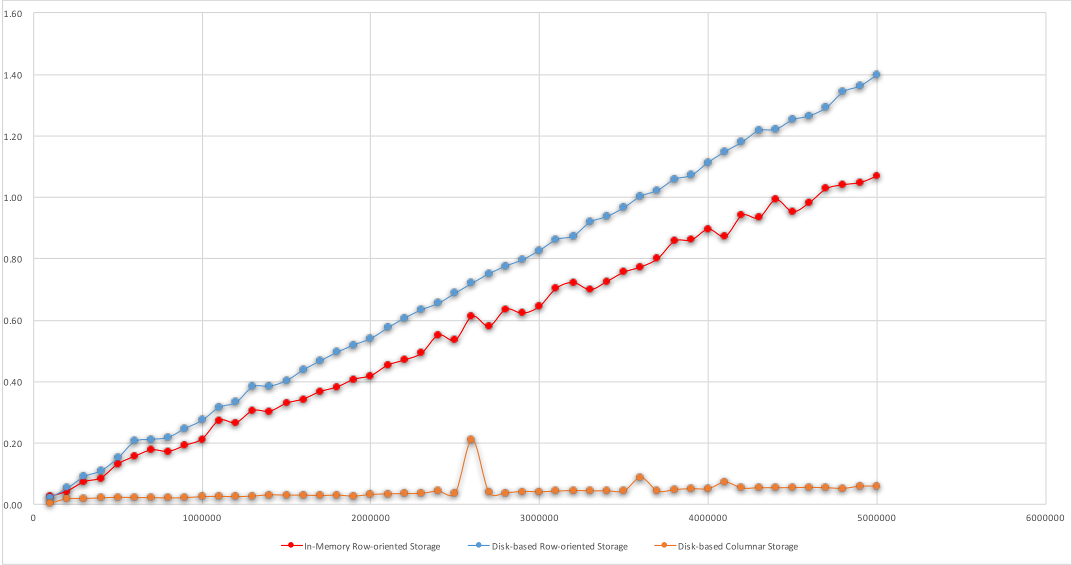


Table 10 shows the experimental data for the three database systems from 100,000 transactions to 5,000,000 transactions.

[11] It should be noted that many "spikes" for the disk-based columnar storage line are caused by the fact that the experiment was conducted on a shared server, therefore the CPUs might not be able to serve only this analysis.

<INSERT Table 10 HERE>

Table 10: Computational Complexity Level of Audit analytics

To sum up, to store data in main memory substantially speeds up data processing. In addition, columnar storage provides fast aggregation in response to complex analytical queries.

Table 11: Simulation Test Comparing Database Performance

	Columnar database	Row-oriented database
In-memory computing	/	Medium
Disk storage computing	High	Low

4.6 Deployment of Cloud Computing

In-memory computing is highly efficient but relatively expensive comparing to disk-based computing because the capacity of hard disks still grows much faster than that of main memory. Cloud computing could be a viable alternative for deploying in-memory columnar databases by start-up or seasonal companies because large amounts of main memory are available and scalable in the cloud thus providing cost flexibility. Moreover, cloud vendors provide professional, up-to-date maintenance, which reduces the cost of a company for hiring (few and expensive) experts in in-memory columnar databases.

In the age of big data, cloud computing serves as a necessary infrastructure that has been widely deployed in many industries. According to SAP, *"Over 90% of businesses are already using cloud technology in a public, private, or hybrid cloud environment."*[12] It can be viewed essentially as a successful application of general outsourcing theory. Cloud computing provides on-demand and high-quality applications and services by centralizing data storage, computing, and transmission (Armbrust et al. 2010). Various cloud models

[12] https://www.sap.com/solution/cloud.html. Accessed 7/19/17 10:15PM.

can be classified either as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS). In the IaaS model, a vendor (like Amazon Web Services) provides on-demand service from their large shared pool of configurable computing resources. In the PaaS model, the cloud provider delivers a computing platform, typically including an operating system (most of which are Windows or Linux), a programming language execution environment, a database, and a Web server. In the SaaS model, the clients have access to various applications. IaaS lets companies "rent" computing resources such as servers, networks, storage, and operating systems on a pay-per-use basis. PaaS provides a cloud platform and tools to help developers build and deploy cloud applications. SaaS is a way of delivering applications over the Internet.

Cloud computing offers an economical way for many companies to deploy the in-memory columnar database for ERP. This research proposes implementing in-memory columnar databases on cloud computing to facilitate automatic and continuous audit analytics. Table 12 shows three different options to deploy the in-memory columnar database (i.e., on the public cloud, on hybrid cloud, and on company's private cloud) as well as the different features and applications of the three options.

Table 12: Cloud vs. On premise vs. Hybrid

	Public Cloud	**On-Premise**	**Hybrid Cloud**
Definition	Open for public usage	On a private network protected by a firewall	Hybrid both cloud and on-premise solutions
Features	Continuous updating Continuous development Dynamic business environment High frequency Faster Innovation	Data security protection	The hybrid strategy of in-memory computation and hard drive computation

Applicable for	SMEs: More lightweight Greater Agility The ability to scale as they grow	Large Enterprise: Greater control Not interested in changing	Seasonality of business nature

Even if with the improvement of hardware the cost of main memory drops unprecedentedly quickly, it is still more expensive to configure a database system with large main memory (i.e., 2 - 10 TB) than that with traditional disk or even SSD. A single SAP HANA system can scale up to 6 TB, and a HANA cluster consisting of a set of connected systems can scale to more than 112 TB[13]. Thanks to the data compression capability of in-memory databases, a 10 TB HANA system can store as much as 50 to 100 TB data from a conventional DBMS, "*which could represent all the credit card transactions for a top 10 bank for 10 years or more.*"[14]

Due to the difficulty of overcoming the hurdle of significant upfront fixed cost, the deployment on the cloud could be cost-effective for a firm to gain fast access to the in-memory database systems, which is budget affordable and easy to implement. The cloud subscribers pay less for upfront investment and shift cost from a capital expenditure to operating expense. Moreover, a cloud solution is maintained by many IMDB specialists for operation and update. It will save expenditures for hiring IMDB specialists on site. For example, a company with $1 billion in revenue is likely to have 50-plus applications running at a time[15]. With cloud access, the company relies on the cloud infrastructure and

[13] https://blogs.saphana.com/2014/09/09/the-sap-hana-faq/#How_big_can_a_SAP_HANA_database_grow_Does_it_scale. Accessed 8/12/17 10:38PM.

[14] https://blogs.saphana.com/2014/09/09/the-sap-hana-faq/#What_happens_if_the_power_goes_out. Accessed 8/12/17 10:39PM.

[15] http://thevarguy.com/blog/best-both-worlds-memory-database-meets-cloud. Accessed 8/12/17 10:42PM.

service to manage the business data. Deployment on the cloud would be the best solution for Small and Medium Size Enterprises (SMEs) or large firms starting investing in IMDB.

4.7 Conclusion

This study introduces the recent breakthroughs (i.e., MPP, large capacity of main memory) in information technology domain and the new in-memory columnar database systems. It proposes a design of the highly efficient and effective in-memory columnar database system architecture for continuous and real-time audit analytics. Furthermore, this study creates a prototype and conducts a simulation test to measure the computational performance in comparison with a conventional relational database to evaluate the performance of the proposed artifact. The simulation test shows great promise of using the in-memory columnar database for continuous audit analytics with high efficiency and effectiveness, which is a key departure from the architecture of the extant ERPs. Compared to a traditional ERP system, the in-memory columnar database architecture will provide high-speed data access and aggregation (i.e., 10 times query performance over traditional row-oriented storage[16]). Therefore, it will massively speed up the process of audit analytics, and significantly improve the efficiency of memory usage. Due to the high efficiency provided by this architecture, a modern ERP system can remove the redundant tables of conclusion materialization and utilize only a small number of base tables to handle the OLTP workload.

[16] https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview. Accessed 4/3/2018 11:23 PM.

The contributions of this study are four-fold. It is the first study that applies the new emerging database system architecture to support continuous audit analytics. It validates the high efficiency and effectiveness of using the in-memory columnar database for complex analytical queries by prototyping a proof-of-work to compare the computational performance of IMDB versus conventional DBMS and columnar database versus row-oriented database regarding computing time and storage size. This study demonstrates that the more complex the audit analytic algorithm is, the higher the requirements of the in-memory column database will be and demonstrates the deployment of the public, private and hybrid cloud.

Due to the lack of availability of open source in-memory columnar database software, this study could not directly measure its performance in comparison with conventional DBMS. In future research, we plan to conduct more simulation tests to measure the computational performance by directly comparing the in-memory columnar database with the conventional DBMS in the context of big data and high-volume main physical memory.

Chapter 5: Adding an Accounting Layer to Deep Neural Network: Designing a Deep Learning-based Continuous Fraud Detection System

5.1 Introduction

Financial statement fraud could lead to severe consequences for companies and auditors. For example, the Securities and Exchange Commission (SEC) would file charges against the fraud companies and their top executives with misleading investors, and then the executives could face punishment by jail, fines, or probation. The auditors who engaged in financial statement fraud may also face fines and damaged reputations. Therefore, if a company realizes that it filed a materially misleading financial statement, it has to restate the previous financial statements immediately. The financial misstatement may be caused by unintentional cleric errors or intentional earnings manipulation that is defined as financial statement fraud.

In order to help auditors and management detect financial statement fraud and reduce fraud risk in a real-time and continuous manner, this study proposes a continuous fraud detection system to identify a company's abnormal financial performance using deep learning algorithms. The proposed system would be able to predict whether a financial statement engages in fraud schemes and further predict the possibility of a specific type of fraud. Based on the prior research on financial statement fraud detection, this study designs and prototypes the Deep Learning-based Continuous Fraud Detection System (DL-based CFDS) using the fraudulent and non-fraudulent peers' financial statements during 1992 and 2012. It then uses the fraudulent and non-fraudulent peers' financial statements during 2012 and 2016 to validate the prediction accuracy of the DL-based CFDS. Moreover, instead of constructing a large number of complex prediction variables (e.g., ratios, accrual-family variables), this study designs an accounting layer that transfers the account balance values to log values. The deep neural networks are able to intelligently generate various input variables in the first layer and further process the inputs to the hidden layers. The evaluation of a prototype shows that the DL-based CDFS achieves high prediction accuracy relative to existing financial statement fraud detection methods, but didn’t find the further partition fraud types improving prediction accuracy.

This study is organized as follows. The second section motivates this research and reviews the prior studies of continuous auditing, financial statement fraud, and fraud detection algorithms. The third section proposes a framework for applying the deep learning for continuous fraud detection. The fourth section creates a prototype and evaluates the performance of the deep-learning based continuous fraud detection system. The last section discusses the contribution, limitation, and further research opportunities.

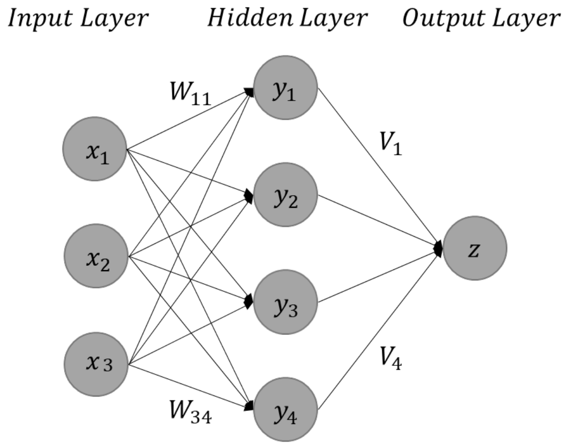
5.2 Motivation and Literature Review

In order to improve prediction accuracy for financial statement fraud detection research, this study attempts to use the deep neural network (DNN) as the basis to build prediction model. However, the modern deep neural network layers (i.e., ConvNet) do not fit in the data format of financial statements. It is necessary to adjust the deep learning (DL) algorithms for a financial statement. Before applying DL algorithms for financial statement fraud detection, this research looks into the computer science field and provides a detailed explanation of deep neural network and reinforcement learning.

5.2.1 Deep Neural Network and Reinforcement Learning

The early research of Artificial Neural Network (ANN) dates back at least to the 1940s (McCulloch and Pitts 1943). Neuroscientists build models of human brain's neural network to understand nervous activity, which was then adopted by computer engineers to create better computer systems. The essential elements of ANNs are neurons (or units), their connections and the weights that are assigned to the connections.

Figure 26: Artificial Neural Network



$$y_j = g\left(\sum_i W_{ij}x_i + b\right) \qquad z = g\left(\sum_i V_j y_j + b\right)$$

$$where\text{:}\ g(a) = \max(a, 0)$$

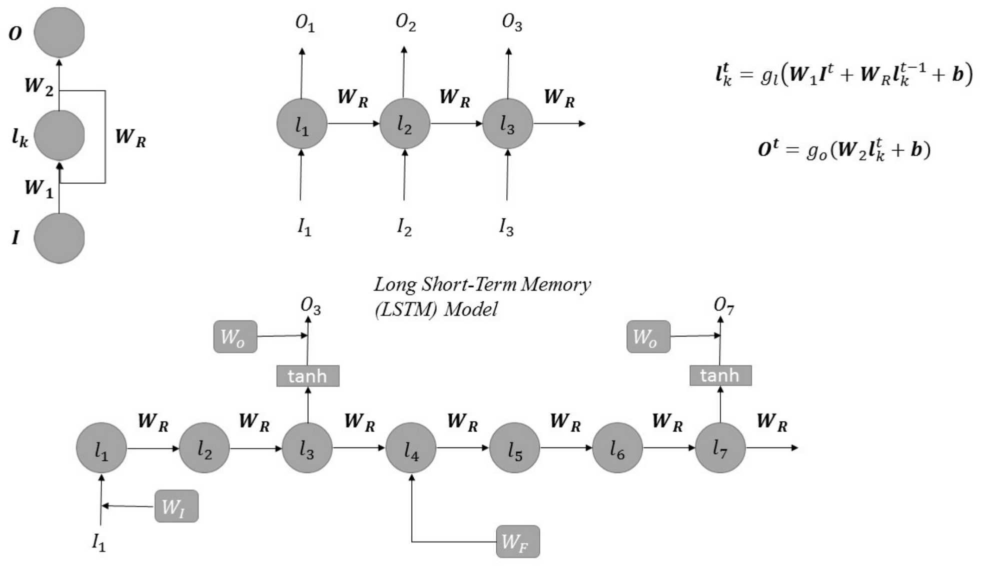
In general, an ANN model consists of many layers, and each layer is a string of neurons. Any ANN model has at least two layers – the input layer and output layer. The input layer takes the input variables, and the output layer provides prediction results. All layers between the input layer and the output layer are called hidden layers. Figure 26 shows an ANN model with a single hidden layer. In this model, x_i, y_j and z represent neurons, $X = (x_1, x_2 \dots x_i)$, $Y = (y_1, y_2 \dots\ y_j)$ and Z represent input layer, hidden layer and output layer, respectively. Neurons in input layer are fully connected with neurons in the hidden layer, and neurons in the hidden layer are fully connected with neurons in output layer. W_{ij} and V_j represent the weights of these connections. In the early and simple ANNs, data move in one direction forward from input layers through the hidden layer and to the output layer, which is called multilayer feedforward network (Hornik et al. 1989). When input data are fed to the input layer, each neuron (e.g., x_1) of the input layer performs a doc product with the input data and the corresponding weight (e.g., W_{11}), adds the bias (e.g., b), applies an activation function (e.g., ReLu) and passes the result to the next layer. All multilayer feedforward networks need activation functions (Hornik et al. 1989; Leshno et al. 1993) to approximate any functions in the computer systems. The activation function of the model in Figure 26 is $g(a) = \max\,(0, a)$, or Rectified Linear Unit (Relu) (Vinod, and Hinton 2010).

Based on conventional ANNs, deep neural networks (DNNs) are designed using modern Graphic Processing United (GPU)-based computers to perform more sophisticated tasks, such as computer vision, speech recognition, and natural language processing. Just as ANN, the weights, and biases are critical parameters to determine a proper DNN. To accurately assign weights and biases across layers is a crucial objective for building a DNN (Schmidhuber 2015). Backpropagation is the most widely used algorithm for training ANNs and DNNs (Riedmiller and Braun 1993). In supervised learning (Møller 1993), a training dataset has both inputs and outputs. After feeding inputs and receiving the predicted outputs, the difference between predicted outputs and the observed outputs could be used to design an error function. To minimize the error, DNNs repeatedly compute the influence of each weight on the error function and adjust each weight through stochastic gradient descent (Riedmiller and Braun 1993; Bottou 2010). Backpropagation stops until convergence is reached or errors have been optimally minimized.

To adjust the fully connected ANNs for processing image data, the computer scientists simplify the ANN architecture by eliminating many unnecessary connections within layers; instead, they create filters to collect local features of an image. DL is designed to process images represented by pixels in red, blue, and green three dimensions. Through the layers of convolution and pooling, the data of an image are transformed to the outcome or the label value. The CNN provides with opportunities to analyze data in big volume and dimensions. Comparing to traditional ANNs, the DNNs bring better capacity and higher efficiency to train a machine, which doesn't require the system designers to have relevant domain knowledge before training a machine. The DNN has many new features, such as computer vision and machine memory, and it even allows a machine to play games with itself. Those features are achieved by Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Reinforcement Learning (RL) for computer vision, machine memory, and self-playing mechanism, respectively.

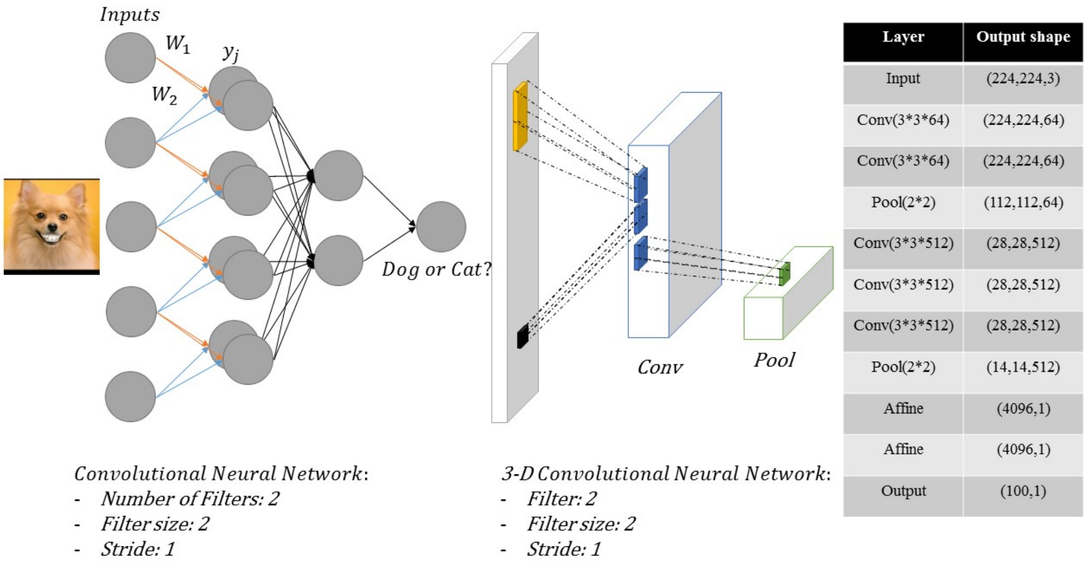
RNNs are developed for sequential data, such as text, speech, and video (Pineda 1987; Lukoševičius and Jaeger 2009). It adds a circuit from the hidden layer's output to its input. In this case, data at time t_1 moves from the input layer to the output layer through the hidden layer and come back to the hidden layer by concatenating data at time t_2. Figure 27 shows the basic structure of a standard RNN, an unfolded RNN and an RNN example – Long Short-Term Memory Model (LSTM) (Gers et al., 2000). The RNN is developed for sequential data, such as text, speech, and video. We simplify an ANN by creating only one neuron for each layer. The RNN adds a circuit from the hidden layer's output to its input. In this case, data at the time t_1 moves from the input layer to the output layer through the hidden layer and come back to the hidden layer by concatenating data at time t_2. By unfolding the circuit, the RNN is displayed in time sequences. Therefore, the RNN can recall the results from the previous time. The LSTM model is an example of RNN in which the memory cells can be tuned to flush, add to or get from the RNN's memory.

Figure 27: Recurrent Neural Network



The CNN is developed for image recognition and computer vision (Lawrence et al. 1997; Krizhevsky et al. 2012). Figure 28 shows the architecture of CNN, which provides two different layers: the Convolution layer and Pool layer, to collect the feature of the image and shrink the feature size, respectively. Usually, an image is represented by pixels in red, blue and green three channels. Through the layers of convolution and pooling, the data of an object are processed, and the output shows the image classification result. The CNN provides the ability to analyze data in big volume and high dimensions. Thanks to the GPU implementation which is efficient at matrix and vector multiplications, it can speed up the learning rate of CNN by at least a factor of 50 (Schmidhuber 2015).

Figure 28: Convolutional Neural Network



Layer	Output shape
Input	(224,224,3)
Conv(3*3*64)	(224,224,64)
Conv(3*3*64)	(224,224,64)
Pool(2*2)	(112,112,64)
Conv(3*3*512)	(28,28,512)
Conv(3*3*512)	(28,28,512)
Conv(3*3*512)	(28,28,512)
Pool(2*2)	(14,14,512)
Affine	(4096,1)
Affine	(4096,1)
Output	(100,1)

The combination of DNN and reinforcement learning (RL) creates machine's another ability that it can automatically play games and even learn "shortcut" from the experience. Based on an objective function Q-value function and dynamic programming, the machine follows existent rules, takes actions, and gets a reward from the environment (Sutton and Barto 1998; Mnih et al. 2015). In order to solve the problem of overfitting, Srivastava et al. (2014) provide a simple way that removes neurons from the layers during training to improve model generalization.

5.2.2 Financial Statements Fraud and Earnings Management

The accounting research on financial statement fraud and Accounting and Auditing Enforcement Releases (AAERs) includes testing the hypotheses grounded in the literature of earnings management (Summers and Sweeney 1998; Beneish 1999; Sharma 2004) and corporate governance (e.g., Beasley 1996). The early research of financial statement fraud dates back to 1990s when Feroz, Park, and Pastena (1991) documented the AAERs affecting stock price and when Beasley (1996) examined the association between the board of the director composition and financial statement fraud. With fewer proportions of outside members on the board of directors supervising a firm's management (Beasley 1996), it is more likely that the management uses discretion to manage the firm's accruals and earnings, and even aggressively commits to financial statement fraud. Therefore, many measures for earnings management are created to indicate the risk of financial misstatement and fraud, such as earnings persistence (e.g., Richardson et al. 2005), abnormal accruals and accruals models (e.g., Jones 1991; Dechow et al. 1995; Dechow and Dichev 2002; Kothari et al. 2005), and earnings smoothness (e.g., McInnis 2010). Beneish (1999) matches the sample of fraud to nonfraud financial statements by SIC code and year and creates an index consisting of seven ratios to indicate the likelihood of an earnings overstatement. Dechow et al. (2011) using predictors identified in the prior literature (e.g., accrual quality variables, financial ratios, employment and order backlog, and stock price related variables) developed a measure, the F-score, to assess the risk of financial misstatement and corporate fraud., Brazel et al. (2009) examined nonfinancial measures (e.g., facilities growth) and suggested they could be used to predict financial statement fraud to add more information for predicting fraud risk.

In order to evaluate the predictive power of the extent accrual-based earnings management measures to detect financial statement fraud, Jones et al. (2008) conducted an empirical analysis comparing ten measures (e.g., discretionary accruals, accrual quality) derived from the accrual models and found that only the accrual estimation errors (Dechow and Dechiev 2002) and its modification have the ability for predicting fraud and non-fraudulent restatements of earnings.

5.2.3 Fraud Detection, Data Mining, and Audit Analytics

Another stream of financial statement fraud detection research is grounded in the literature of data mining and machine learning (e.g., Green and Choi 1997; Cecchini et al. 2010; Perols 2011; Perols et al. 2015). Early work by Green and Choi (1997) develops a financial statement fraud detection model using a neural network classifier that performs relatively well. The more recent research employs other additional classification techniques, such as Support Vector Machine, Logistic Regression and many other machine learning ensemble algorithms (Cecchini et al. 2010; Perols 2011), which improves the performance of fraud prediction. Perols (2011) uses six statistical and machine learning models in detecting financial statement fraud and shows that logistic regression and support vector machine perform well relative to an artificial neural network. In addition to financial variables, text-mining techniques are used to detect financial statement fraud. Humpherys et al. (2011) extracts MD&A textual data from 10-Ks and uses Naïve Bayes and decision tree algorithms to identify fraudulent financial statement. In order to solve the problems of fraud data rarity and large dimensionality, Perols et al. (2015) develop three data preprocessing methods (i.e., observation under-sampling, variable under-sampling and fraud type partition) to improve prediction performance of the best current fraud classification techniques. Vandervelde et al. (2008) illustrate the effectiveness of considering the relations between accounts under cross-sectional and temporal analysis, which can lead to fewer type 2 error. Peer firms matching and misstatement prediction based on different types could also improve prediction accuracy (Perols et al. 2015).

In order to apply the deep learning to financial statement fraud detection, this research makes some adjustment of DNNs by adding a new accounting layer before inputs being processed to hidden layers. The functionality of the accounting layer is to transfer the account balance values to log values. Therefore, the various combination of inputs will represent the various log of ratios that generated by the accounting layer. This study uses AUC as the measure for prediction accuracy of the designed Deep Learning-based Continuous Fraud Detection System (DL-based CFDS).

5.3 The framework of Deep Learning-based Continuous Fraud Detection System

5.3.1 Framework Overview

The objective of this research is to design a deep learning-based financial statement fraud detection system that can be easily implemented by academia and practitioners. This system can be used in various phases of an audit engagement. For example, an engagement team could use it for audit risk assessment. In audit planning stage, the system could be used to evaluate the risk of financial statement fraud and a certain fraud category, which could help auditors properly allocate limited audit resource. DL-based CFDS can be integrated into analytical procedures to identify unusual financial statements and predict the likelihood of fraud and fraud categories.

Figure 29: Deep Learning-based Continuous Fraud Detection System

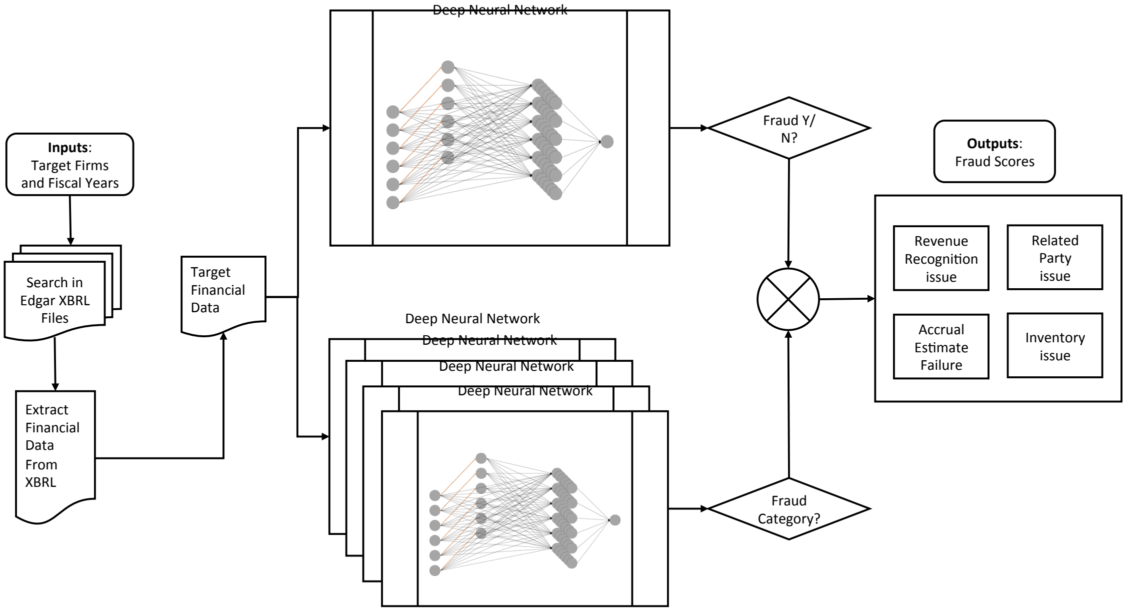


Figure 29 shows the framework of the Deep Learning-based Continuous Fraud Detection System. To estimate the fraud risk of a financial statement, an auditor could just put in the CIK code and the fiscal year end s/he is interested. DL-based CFDS will automatically connect the SEC Edgar filings system and download the corresponding XBRL files of that financial statement. Then, DL-based CFDS will extract financial data from the XBRL files and process them into a set of deep neural networks for classification and prediction. The auditor can either use the previously trained deep neural networks or build these deep neural networks by herself/himself. The same XBRL data will be processed into multiple deep neural networks, one of which is used to determine whether this financial statement is a fraudulent financial statement and then calculate the possibility of fraud. The other deep neural networks are used to determine whether this financial statement involves in a particular fraud category, such as revenue recognition issue, related party transaction issue, and inventory related issue, and then calculate the corresponding possibility of a specific fraud category.

To build and train the center classifiers within a DL-based CFDS, this study follows recent financial statement fraud detection research (i.e., Dechow et al. 2011; Cecchini et al. 2010; Perols 2011; Perols et al. 2015) that selects a set of critical financial variables from balance sheet, income statement and cash flow statement, and uses a GPU-based deep learning library in Python programming language – Keras – to build the deep neural networks. In this study, the criteria to select financial variables is that whether a variable contains more than 10% missing values. If it does, it will be dropped from the dataset. Table 13 shows 88 input variables included in this study, such as total assets, total liabilities, total current assets, and total current liabilities. Then, the 88 variables are processed through an accounting layer that applies logarithm transformation converting the 88 variables to another 88 variables. For example, if a firm's total asset is 100 million dollars, the accounting layer will create a new variable, 4.61-million-dollar log total asset. After the logarithm transformation through the accounting layer, the combination of original 88 variables and the new 88 log variables will be input into the deep neural networks.

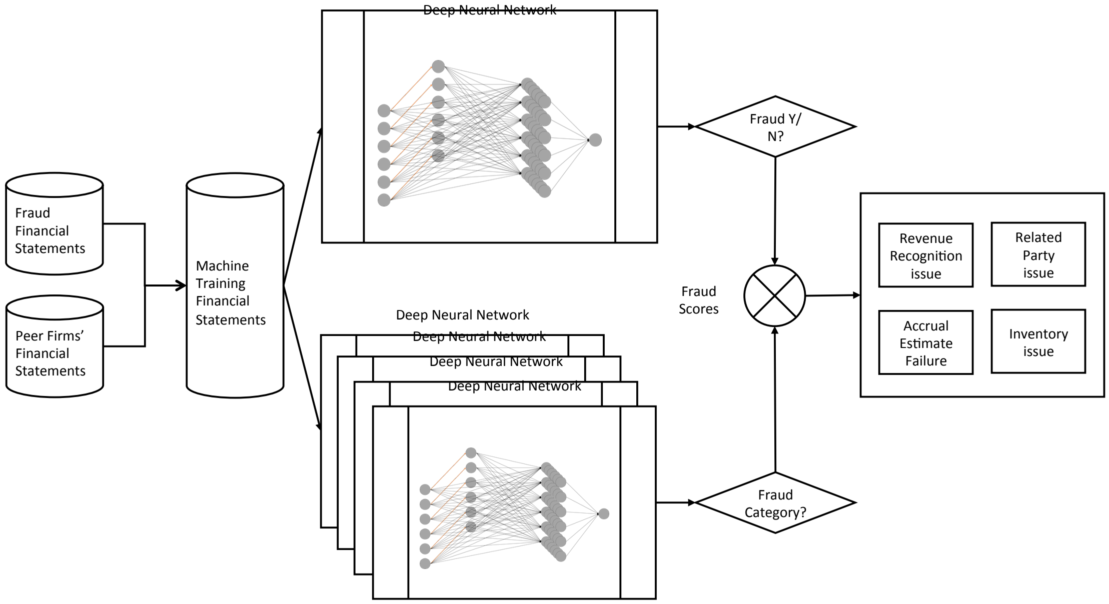
<INSERT Table 13 HERE>

Table 13: Inputs Variables

The function of accounting layer is not only transferring the account values to log values, but also providing a mathematically convenient method to generate financial ratios. For example, if all variables are log transformed, calculating log value of a ratio that is α divided by β could be simply log α minus log β. Besides, the coefficients of log α and log β could represent powers of the numerator and denominator of a ratio. For those account values equal zero, the accounting layer would convert them into zero, while for those account value is less than zero, the accounting layer would convert them into the opposite value of the log of negative account values.

Figure 30 shows the training processes of a DL-based CFDS. This study creates a large training data warehouse by collecting the fraudulent financial statements and the peers' financial statements based on industry and fiscal year. Then, the financial data in training data warehouse are processed into the center classifiers (a set of deep neural networks) using a GPU-based Deep Learning library in Python, Keras. Several DL-based classifiers are trained using backpropagation algorithm and "Adam" optimizer. After a large number of epochs of training, a set of DL-based core classifiers are built among which one is used to predicting whether a financial statement involves fraud and provide the possibilities of fraud and the others are used to predict whether a financial statement involves a specific type of fraud and provide the possibilities of a specific fraud type. Therefore, to train many DL-based core classifiers, a full sample is partitioned into several subsets based on fraud categories.

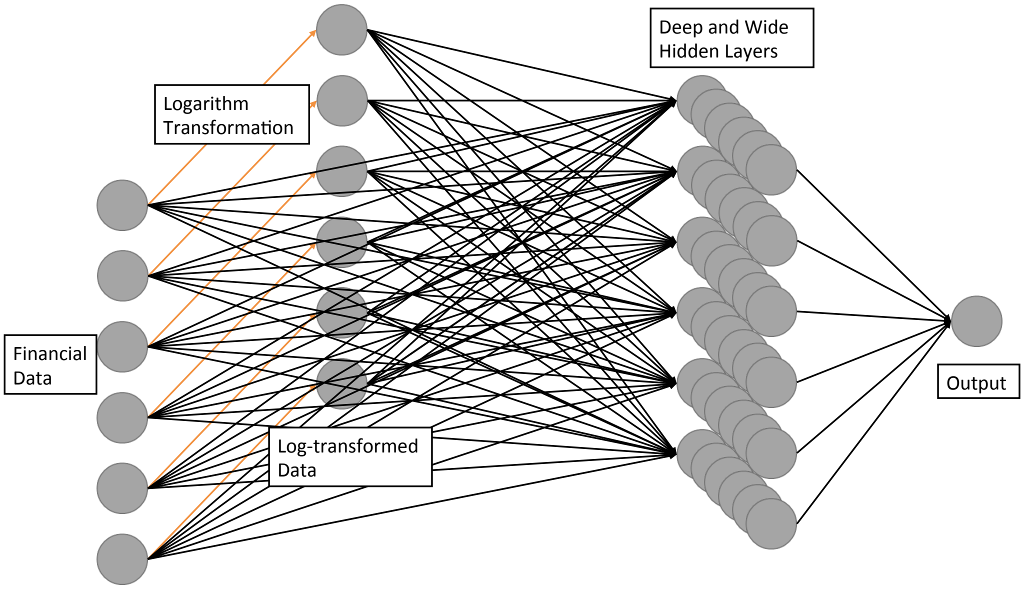
Figure 30: Training Deep Learning-based Continuous Fraud Detection System



5.3.2 Center Classifier

The center of the Deep Learning-based Continuous Fraud Detection System is a set of Deep Learning-based classifiers. As shown in Figure 31, each deep neural network is a collection of connected nodes called "artificial neurons." These artificial neurons are arranged in the form of multiple layers, and the artificial neurons in a layer are connected to those in the layers on the left or right. The connections between artificial neuron layers can transmit signals from one to another, and the artificial neuron that receives the signals can process it and then signal the artificial neurons connected to it. The process of an artificial neuron receiving a signal from a previous one and then transmitting it to the next one requires an activation function to activate the signal.

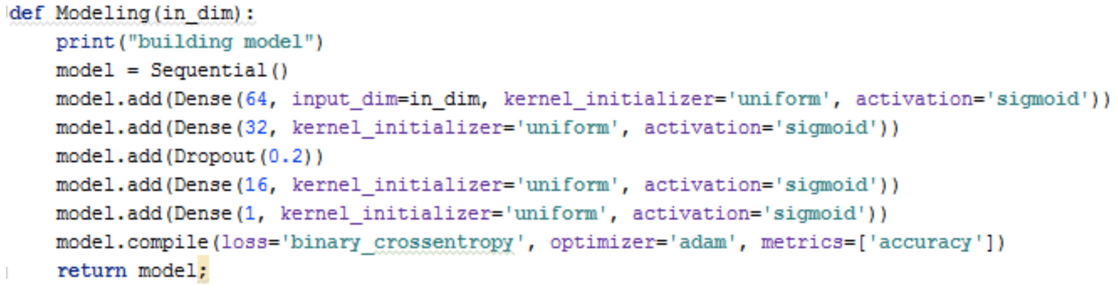
Figure 31: Training Deep Learning-based Continuous Fraud Detection System



An activation function converts an input signal of an artificial neuron to an output signal. The output signal will be used as an input signal in the next layer of the deep neuron network. Technically, each artificial neuron in an artificial layer is assigned a weight. When a vector of inputs whose dimension is the same as the number of neurons in the layer is processed through the layer, it will aggregate the product of inputs and their corresponding weights and then apply an activation function to the sum to get an output as an input for the next layer. The attribute of activation functions is to increase the degree of nonlinearity when transferring inputs to outputs. In many cases, the reason why deep learning is superior to linear regression is that it has more power to learn complicated and nonlinear functional mapping from data. Without activation function, a deep learning model would be just a linear regression model. The most popular activation function includes Sigmoid (or Logistic), Tanh (or Hyperbolic tangent) and ReLu (or Rectified Linear Units). Since logistic regression is widely-used in accounting research field, this study decides to use the Sigmoid function as the activation function for all layers.

Based on rule of thumb and our thousands of trials, as shown in Figure 32 all DL-based center classifiers are six-layer neuron networks in which: 1) the first layer (input layer) consists of 176 neurons because the input data has 176 attributes (88 account values and 88 log account values); 2) the second, third and fifth layer have of 64, 32 and 16 neurons, respectively; 3) the fourth layer is a dropout layer; and 4) the sixth layer (output layer) has only one neuron to provide results, which can be either true (fraud or certain fraud type) and false or numeric value of possibilities (fraud scores). This research adds a logarithm transformation layer that transforms the original account value to log value to adjust deep learning for accounting data. The combination of original value and log value is processed through the hidden layers.

Figure 32: Deep Learning-based Center Classifier



```
def Modeling(in_dim):
    print("building model")
    model = Sequential()
    model.add(Dense(64, input_dim=in_dim, kernel_initializer='uniform', activation='sigmoid'))
    model.add(Dense(32, kernel_initializer='uniform', activation='sigmoid'))
    model.add(Dropout(0.2))
    model.add(Dense(16, kernel_initializer='uniform', activation='sigmoid'))
    model.add(Dense(1, kernel_initializer='uniform', activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model;
```

The goal of training a deep neuron network is to find the optimal weights for each artificial neuron whereby the deep neuron network can map inputs to outputs with smallest errors accurately. A popular training algorithm is a backpropagation that mainly includes five steps: 1) it randomly assigns weights for each neuron; 2) based on the assigned weights and activation functions, it uses inputs to calculate final outputs through layers; 3) given an error function, e.g., binary cross entropy, it computes the difference between the calculated outputs and the given outputs; 4) it uses the error to compute gradient at each layer backwards from the output layer to the input layer; 5) it uses the gradient to update weights for each neuron. The same steps will repeat many times until the error is optimally minimized. This study uses one of the gradient descent optimization algorithms “Adam” as the optimizer for training these DL-based center classifiers.

5.4 Prototyping and Evaluation

5.4.1 Data

The fraud financial statements are collected from AuditAnalytics – Non-Reliance Restatements Dataset. From 1992 to 2016, 275 firms issued in total 286 restatements due to financial statement fraud. After dropping the observations that (1) miss Compustat data, (2) have less than one-year fraud period, (3) are in the industry of Finance, Insurance, Real Estate, the final fraud financial statement fraud consists of 184 firm-year observations, which is comparable to prior studies (i.e., in Perols et al. (2017)’study there are 51 fraudulent firm-year observations). This study selects the peer firms that never filed any restatements with the same fiscal year and sic two digital codes as fraud firms to construct the nonfraud financial statements from peer firms. In total, the final nonfraud sample consists of 28,536 firm-year observations. Therefore, the prior probability of fraud is 0.6 percent, which is also comparable to prior studies (e.g., 0.3 percent in Perols et al. (2017) and 0.6 percent in Bell and Carcello (2000)). Table 14 shows the details of data collection and preprocessing. In total, there are 28,720 firm-year observations in the sample dataset.

Table 14: Data Collection and Preprocessing

	# firm-year Observations
Restatements filed between 1992 and 2016	17214
Fraud sample:	
Restatements due to financial statement fraud between 1992 and 2016	286
Less: merge Compustat data between 1992 and 2016	(80)
Less: fraud period is less than one year	0
Less: financial industry (SIC code 6000~6799)	(22)
The final sample of fraudulent financial statements	184
Nonfraud sample:	
Firm-year financial statement between 1992 and 2016	250145
Less: financial industry (SIC code 6000~6799)	(70555)
Less: firms that have ever filed any restatement	(79277)
Less: match peers with the same fiscal year and sic code	(71777)
The final sample of nonfraud peers' financial statements	28536

The fraud firm-year sample can be further partitioned by fraud categories. Table 15 shows the number and percent of each fraud category. The revenue recognition issues, foreign and related party transaction issues and accounts/loans receivable issues are the top three fraud categories with 89, 73 and 53 instances during 1992 and 2016. It should be noted that the total percentage is greater than 100% because every fraud sample could involve in more than one fraud schemes or categories. For each fraud category, a DL-based center classifier is built and trained to predict whether a financial statement involves a specific type of fraud and how possible for this fraud type.

Table 15: Fraud Types and Fraud Schemes

Fraud Categories	N	Percent
Revenue recognition issues	89	48.37%
Foreign, related party, affiliated, or subsidiary issues	73	39.67%
Liabilities, payables, reserves and accrual estimate failures	53	28.80%
Accounts/loans receivable, investments & cash issues	53	28.80%
Inventory, vendor and/or cost of sales issues	52	28.26%
Foreign, subsidiary only issues (subcategory)	51	27.72%
Expense (payroll, SGA, other) recording issues	44	23.91%
PPE intangible or fixed asset (value/diminution) issues	29	15.76%
Acquisitions, mergers, disposals, re-org acct issues	22	11.96%
Tax expense/benefit/deferral/other (FAS 109) issues	22	11.96%
Debt, quasi-debt, warrants & equity (BCF) security issues	20	10.87%
Deferred, stock-based and/or executive comp issues	18	9.78%
Capitalization of expenditures issues	17	9.24%
Lease, SFAS 5, legal, contingency and commitment issues	16	8.70%
Intercompany, investment in subs./affiliate issues	16	8.70%
Acquisitions, mergers, only (subcategory) acct issues	14	7.61%
Consolidation issues incl Fin 46 variable interest & off-B/S	13	7.07%
Depreciation, depletion or amortization errors	11	5.98%
Financial derivatives/hedging (FAS 133) acct issues	11	5.98%
Fin Statement, footnote & segment disclosure issues	11	5.98%
Unspecified (amounts or accounts) restatement adjustments	9	4.89%
PPE issues - Intangible assets, goodwill only (subcategory)	9	4.89%
Intercompany, only, (subcategory) - accounting issues	8	4.35%
Lease, leasehold and FAS 13 (98) only (subcategory)	7	3.80%
Gain or loss recognition issues	5	2.72%
Consolidation, foreign currency/inflation (subcategory) issue	5	2.72%
Cash flow statement (SFAS 95) classification errors	4	2.17%
X - Audit or auditor related restatements or nonreliance	4	2.17%
EPS, ratio and classification of income statement issues	3	1.63%
Balance sheet classification of assets issues	3	1.63%
X - Audit(or) consent re opinion in f/s issues (subcategory)	3	1.63%
Deferred, stock-based options backdating only (subcategory)	3	1.63%
Pension and other post-retirement benefit issues	3	1.63%
Y - Registration/security (incl debt) issuance issues	2	1.09%
Y - Loan covenant violations/issues	2	1.09%
Restatements made while in bankruptcy/receivership	2	1.09%
X - Audit(or) inability to rely on Co reps (subcategory)	2	1.09%

Table 16 shows the distribution of the fraud and the nonfraud financial statements by fiscal year (Panel A) and industry (Panel B). Panel A shows that the top four fiscal years for fraud instances are 2000, 1998, 1999 and 2001, which have 23, 16, 15 and 15 fraud instances. In general, the number of financial statement fraud increases during 1992 to 2000, while the number of financial statement fraud decreases during 2000 to 2016. Notably, after 2012 the number of financial statement fraud decreases from 7 to 1. Panel B shows that the top four industries for fraud instances are Business Services, Electronic and other Electrical Equipment and Components, Chemicals and Allied Products, and Electric, Gas and Sanitary Services, which have 44, 14, 11 and 11 fraud instances.

Table 16: Distribution by Fiscal Year

	Fraud		**Nonfraud**	
Fiscal Year	**N**	**Percent**	**N**	**Percent**
1992	1	0.54%	89	0.31%
1993	1	0.54%	398	1.39%
1994	4	2.17%	1123	3.94%
1995	2	1.09%	203	0.71%
1996	5	2.72%	1758	6.16%
1997	7	3.80%	1823	6.39%
1998	16	8.70%	2698	9.45%
1999	15	8.15%	2683	9.40%
2000	23	12.50%	2778	9.74%
2001	15	8.15%	1882	6.60%
2002	8	4.35%	1418	4.97%
2003	10	5.43%	1225	4.29%
2004	9	4.89%	1053	3.69%
2005	8	4.35%	896	3.14%
2006	8	4.35%	1197	4.19%
2007	6	3.26%	690	2.42%
2008	6	3.26%	749	2.62%
2009	7	3.80%	725	2.54%
2010	5	2.72%	578	2.03%
2011	6	3.26%	1026	3.60%
2012	7	3.80%	725	2.54%
2013	6	3.26%	1137	3.98%
2014	5	2.72%	984	3.45%
2015	3	1.63%	358	1.25%

2016	1	0.54%	340	1.19%
Total	184	100.00%	28536	100.00%

5.4.2 Prototype Evaluation Result

The prototype of DL-based Continuous Fraud Detection System is created using Python language on a Windows Server with Intel®Xeon® CPU E5-2687W 0 @3.10 GHz, RAW 64.0GB and Disk 7.4TB. It includes DL-based center classifiers, XBRL data fetcher and a data warehouse and preprocessor, which are readily available to implement. This study uses data from AuditAnalytics and Compustat to simulate the process of training and testing DL-based center classifiers. Based on the idea that machine learns from the past and makes predictions for future, we divide the dataset into two parts: (1) data before 2012 are training data and (2) data from or after 2012 are holdout testing data. The reason we choose the fiscal year 2012 as the cutoff is to construct the training and testing pools as the ratio of 9:1.

The prior studies use many different methods to build models and make predictions. In the 1990s, Beneish (1999) uses probit regression and Green and Choi (1997) use neuron network to build prediction models for financial statement fraud. The holdout test AUCs are 0.49 and 0.47 (tested by Cecchini 2010), respectively. This study also follows the prior studies using AUC as the measure of prediction accuracy. In general, 0.5 AUC means random guess, which is a starting point for machine learning. Dechow et al. (2011) use accrual-family variables and other financial and nonfinancial data as input variables and logistic regression as the model to predict material financial misstatement, while Cecchini et al. (2010) uses financial kernel as input variables and support vector machine as the model to predict financial statement fraud, which greatly improve the AUCs to 0.58 and

0.59 (tested by Perols et al. (2015)), respectively. Besides advanced modeling methods and additional input variables, Perols et al. (2015) use undersampling of observations and variables improving the AUC to 0.73.

Based on the prior studies, this paper continues seeking for methods to improve prediction AUC. To obtain better prediction accuracy, first, we use a large number of ratios as the input variables. Second, we calculate the increase rate of ratios, abnormal increase rate of ratios based on time-series moving average and abnormal increase rate of ratios based on industry median. Then, we combine the basic ratios, the percentage change of ratios and abnormal percentage change of ratios and process them into DL-based center classifiers. The results show that the AUC does not improve significantly. Then, we apply the accounting layer that uses logarithm transformation to directly convert account values to log values for deep neural network processing. The reason we use accounting layer is to let the deep neural network construct ratio-family variables based on backpropagation and gradient descent. Furthermore, based on fraud categories, various DL-center classifiers are trained to predict a specific type of financial statement fraud. In summary, as shown in Table 17, the DL-based center classifiers together with accounting layer improve the AUC to 0.76 level, despite that some of the DL-based center classifiers for center types do not work well. Also, to predict whether a financial statement is a fraud the DL-based classifiers can also provide the possibilities of fraud.

Table 17: Design Evaluation and Comparison Result

Studies	Algorithm	AU C
Beneish (1999)	Probit Regression	0.49
Green and Choi (1997)	Neural Network	0.47
Dechow et al. (2011)	Logtistic Regression	0.58
Cecchini et al (2010)	Support Vector Machine	0.59
Perols et al. (2016)	Partition on Fraud Types	0.68
Perols et al. (2016)	Observations Undersampling	0.73
This study	Logistic Regression	0.63
This study	Decision Tree	0.62
This study	Random Forest	0.65
This study	Multilayer Perception	0.67
This study	Support Vector Machine	0.58
This study	Ada Boost	0.63
This study	Gaussian NB	0.62
This study	Quadratic Discriminant Analysis	0.51
This study	Deep Learning (with accounting layer)	0.76
This study	Deep Learning (with accounting layer) for specific fraud category-revenue recognition	0.74
This study	Deep Learning (with accounting layer) for specific fraud category - related party transaction issue	0.61
This study	Deep Learning (with accounting layer) for specific fraud category - Liabilities, payables, reserves and accrual estimate failures	0.71
This study	Deep Learning (with accounting layer) for specific fraud category - Accounts/loans receivable, investments & cash issues	0.68

This prototype does not use accrual-family variables or financial kernels. A simple accounting layer can be used to increase the nonlinearity and complexity of inputs to models. It should be noted that the prediction accuracy improves slowly over the years. The main problem is there is too much noise between the inputs and outputs, which means the signals absorbed by deep neural network do not necessarily signal the outcome. Regarding functional mapping the deep neural network should outperform the linear regression models because the nonlinearity and complexity generated within deep neural networks can easily map the inputs to outputs. However, a generalization of the functional mapping is retained by the generalization of the relations between inputs and outputs. In account field, financial statement fraud is human judgment. Deep learning could have detected financial anomalies, but it could be rationalized by credible evidence, which is called false positive; while deep learning that couldn't detect some anomalies could be the real false negatives that were not detected in the past financial statements, which is called false negative.

5.5 Discussion, Limitation and Future Research

This study modifies the recent breakthroughs in deep learning for audit analytics and financial statement fraud detection. By designing a framework of deep learning - based continuous fraud detection systems, this essay demonstrates the functionalities of financial statement fraud detection and evaluates the performance of the framework by creating a prototype using fraud and restatement data. The evaluation test shows improved prediction accuracy compared with existent financial statement fraud detection algorithms, which enables researchers and practitioners to continuously calculate the fraud possibility when evaluating the likelihood of financial statement fraud. Furthermore, it provides auditors and regulators an architecture of continuous fraud detection to conduct analytical procedure for fraud detection.

Compared to existing studies this paper includes more fraud sample and provides improved financial statement prediction models based on deep learning algorithms. The accounting layer specification simplifies the inputs construction process relative to prior studies, such as accrual-family variables and financial kernels. Furthermore, the framework of deep learning-based continuous fraud detection systems provided in this study is relatively easy to adopt and implement for accounting researchers and practitioners. We think our study provides the highest prediction accuracy in predicting financial statement fraud.

Although this study does not find that partition fraud types can significantly improve prediction accuracy, the result is consistent with the study by Perols et al. (2015). The false negatives and false positives are still the limitations of this type of studies. We think the deep neural networks are sensitive enough to learn the complicated relationship between financial data and the fraud outcome, but the reasons why there are still many false positives are: (1) the detected anomalies can be rationalized outside of financial statement, (2) the detected anomalies could turn out to be other accounting irregularities, such as, bankruptcy, and default, while the reasons for false positives could be there were many undetected fraudulent financial statement in the training sample. In general, to identify whether a financial statement engages in a fraud scheme is a combination of anomaly detection task and human judgment task, we might not be able to solely rely on a financial statement to determine whether a firm engages in a fraud scheme. DL-based CFDS provides a conceptual framework and physical prototype to help auditors find material financial anomalies when conducting audit analytics.

Since we think the deep learning algorithms are sensitive to learn the complex relations between financial data and accounting irregularities, our future research will expand the scope from financial statement fraud detect the other accounting anomalies detections, such as bankruptcy, financial distress, and, internal control weakness. To further improve prediction accuracy, more financial and nonfinancial data could be included as inputs to signal the accounting anomalies, and the more simulated fraudulent financial statement could be created to increase the training sample.

Chapter 6: Conclusion and Future Research

In summary, this dissertation contributes to the accounting literature by proposing a comprehensive architecture of continuous audit analytics that consists of three system layers using three cutting-edge emerging technologies (i.e., the blockchain, cloud-based in-memory computing, and deep learning). The Blockchain-based Transaction Processing System is designed and created to support the continuous test of management assertions on transaction level. The Cloud-based In-Memory Columnar Database Architecture is proposed and evaluated to support the continuous data aggregation and analytics to test the management assertions on account balance level, and Deep Learning-based Continuous Fraud Detection Systems is designed and prototyped to support continuous financial statement fraud detection on financial reporting level. The first essay creates a blockchain-based business ecosystem, whereby the second essay builds enterprise information systems using the in-memory columnar database architecture for individual organizations. The third essay aggregates financial data from the new enterprise information systems to perform financial statement fraud prediction analysis using deep learning algorithms. The objective of the proposed architecture is to upgrade the traditional rule-based continues auditing systems to intelligence-based continuous auditing systems. The performance of the three systems is evaluated through prototyping and simulation tests. The following summarizes the preliminary findings and future work of the three essays.

Based on the recent technical innovation of blockchain technology and zero-knowledge proofs, the first essay proposes a design of the blockchain-based accounting information system (Bb-TPS) and demonstrates its functionalities of real-time accounting, continuous monitoring and permission management using a prototype. Furthermore, Bb-

TPS uses the zk-SNARK scheme and homomorphic encryption to provide high-level privacy-preserving mechanisms. Finally, the comparative computational performance of blockchain and relational database in transaction recording is evaluated and discussed. The accounting-blockchain convergence shows great promise for improving information integrity, decreasing transmission cost, increasing the speed of transaction settlement, and preventing fraudulent transactions. Therefore, the deployment of blockchain enables the improvement of efficiency and effectiveness of accounting and audit practice. The limitation of this research is that it does not specify the details of the block mining and rewarding mechanisms as well as the implementation of zk-SNARK for Bb-TPS. Although at the current stage the computational overhead of blockchain is still significant compared to that of the relational database, it is expected that technology improvements will result in cost reductions allowing blockchain to become a widely utilized infrastructure for enterprise information systems and continuous audit systems.

The second essay introduces the recent breakthroughs (i.e., MPP, large capacity of main memory) in information technology domain and the in-memory columnar database systems. It proposes a design of the highly efficient and effective in-memory columnar database system architecture for continuous and real-time audit analytics. In order to evaluate the performance of the proposed artifact, this study creates a prototype and conducts a simulation test to measure the computational performance in comparison with a conventional relational database. The simulation test shows great promise of using the in-memory columnar database for continuous audit analytics with high efficiency and effectiveness, which is a crucial departure from the architecture of the extant ERPs. The contributions of this study are four-fold. It is the first study that applies this emerging database system architecture to support continuous audit analytics. It validates the high efficiency and effectiveness of using the in-memory columnar database for complex analytical queries by prototyping a proof-of-work to compare the computational performance of IMDB versus conventional DBMS and columnar database versus row-oriented database regarding computing time and storage size. This study demonstrates that the more complex the audit analytic algorithm is, the higher the requirements of the in-memory column database will be. It also demonstrates how to deploy the in-memory columnar database to the public, private, and hybrid cloud. Due to the lack of availability of open source in-memory columnar database software, this study could not directly measure its performance in comparison with conventional DBMS. In future research, we plan to conduct more simulation tests to measure the computational performance by directly comparing the in-memory columnar database with the conventional DBMS in the context of big data and high-volume main physical memory.

This study modifies the recent breakthroughs in deep learning for audit analytics and financial statement fraud detection. By designing a framework of deep learning - based continuous fraud detection systems, this essay demonstrates the functionalities of financial statement fraud detection and evaluates the performance of the framework by creating a prototype using fraud and restatement data. The evaluation test shows improved prediction accuracy compared with existent financial statement fraud detection algorithms, which enables researchers and practitioners to continuously calculate the fraud possibility when evaluating the likelihood of financial statement fraud. Furthermore, it provides auditors and regulators an architecture of continuous fraud detection to conduct analytical procedure for fraud detection. Compared to existing studies this paper includes more fraud sample and provides improved financial statement prediction models based on deep learning algorithms. The accounting layer specification simplifies the inputs construction process relative to prior studies, such as accrual-family variables and financial kernels. Furthermore, the framework of deep learning-based continuous fraud detection systems provided in this study is relatively easy to adopt and implement for accounting researchers and practitioners. We think our study provides the highest prediction accuracy in predicting financial statement fraud.

BIBLIOGRAPHY

Abadi, D., S. Madden, and M. Ferreira. 2006. Integrating compression and execution in column-oriented database systems. Paper read at Proceedings of the 2006 ACM SIGMOD international conference on Management of data.

Alles, M., G. Brennan, A. Kogan, and M. A. Vasarhelyi. 2006. Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. *International Journal of Accounting Information Systems* 7 (2):137-161.

Alles, M. G., A. Kogan, and M. A. Vasarhelyi. 2002. Feasibility and economics of continuous assurance. *Auditing: A Journal of Practice & Theory* 21 (1):125-138.

———. 2008. Putting continuous auditing theory into practice: Lessons from two pilot implementations. *Journal of Information Systems* 22 (2):195-214.

Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica. 2010. A view of cloud computing. *Communications of the ACM* 53 (4):50-58.

Back, A., M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille. 2014. *Enabling blockchain innovations with pegged sidechains*. Available at: http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains.

Batcher, K. E. 1980. Design of a massively parallel processor. *IEEE Transactions on Computers* 29 (9):836-840.

Beasley, M. S. 1996. An empirical analysis of the relation between the board of director composition and financial statement fraud. *Accounting review*:443-465.

Bengtsson, M., and S. Kock. 2000. "Coopetition" in business Networks—to cooperate and compete simultaneously. *Industrial marketing management*29 (5):411-426.

Ben-Sasson, E., A. Chiesa, D. Genkin, E. Tromer, and M. Virza. 2013. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In *Advances in Cryptology–CRYPTO 2013*: Springer, 90-108.

Ben-Sasson, E., A. Chiesa, E. Tromer, and M. Virza. 2014. Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture. Paper read at USENIX Security Symposium.

Blum, M., P. Feldman, and S. Micali. 1988. Non-interactive zero-knowledge and its applications. Paper read at Proceedings of the twentieth annual ACM symposium on Theory of computing.

Boyan, J. A. 2002. Technical update: Least-squares temporal difference learning. *Machine learning* 49 (2-3):233-246.

Buterin, V. 2014. A Next-Generation Smart Contract and Decentralized Application Platform. *white paper*.

Cecchini, M., H. Aytug, G. J. Koehler, and P. Pathak. 2010. Detecting management fraud in public companies. *Management Science* 56 (7):1146-1160.

Chaudhuri, S., and U. Dayal. 1997. Data warehousing and OLAP for decision support. *ACM Sigmod Record* 26 (2):507-508.

———. 1997. An overview of data warehousing and OLAP technology. *ACM Sigmod Record* 26 (1):65-74.

Cohen, D. A., A. Dey, and T. Z. Lys. 2008. Real and accrual-based earnings management in the pre-and post-Sarbanes-Oxley periods. *The accounting review* 83 (3):757-787.

Dai, J., and M. A. Vasarhelyi. 2017. Toward Blockchain-Based Accounting and Assurance. *Journal of Information Systems* 31 (3):5-21.

Dechow, P. M., and I. D. Dichev. 2002. The quality of accruals and earnings: The role of accrual estimation errors. *The accounting review* 77 (s-1):35-59.

Dechow, P. M., W. Ge, C. R. Larson, and R. G. Sloan. 2011. Predicting material accounting misstatements. *Contemporary accounting research* 28 (1):17-82.

Dechow, P. M., R. G. Sloan, and A. P. Sweeney. 1995. Detecting earnings management. *Accounting review*:193-225.

ElGamal, T. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory* 31 (4):469-472.

Elliott, R. K., and J. J. Willingham. 1980. *Management fraud: Detection and deterrence*: Petrocelli Books New York, NY.

Ernst, D., P. Geurts, and L. Wehenkel. 2005. Tree-based batch mode reinforcement learning. *Journal of machine learning research* 6 (Apr):503-556.

Ernst & Young LLP. 2016. *Blockchain reaction: Tech plans for critical mass*. Available at: http://www.ey.com/gl/en/industries/technology/ey-blockchain-reaction-tech-plans-for-critical-mass.

Färber, F., S. K. Cha, J. Primsch, C. Bornhövd, S. Sigg, and W. Lehner. 2012. SAP HANA database: data management for modern business applications. *ACM Sigmod Record* 40 (4):45-51.

Gal, G. 2008. Query issues in continuous reporting systems. *Journal of Emerging Technologies in Accounting* 5 (1):81-97.

Garcia-Molina, H., and K. Salem. 1992. Main memory database systems: An overview. *IEEE Transactions on knowledge and data engineering* 4 (6):509-516.

Gentry, C. 2009. *A fully homomorphic encryption scheme*: Stanford University.

Glancy, F. H., and S. B. Yadav. 2011. A computational model for financial reporting fraud detection. *Decision Support Systems* 50 (3):595-601.

Goldwasser, S., S. Micali, and C. Rackoff. 1989. The knowledge complexity of interactive proof systems. *SIAM Journal on computing* 18 (1):186-208.

Graham, J. R., C. R. Harvey, and S. Rajgopal. 2005. The economic implications of corporate financial reporting. *Journal of Accounting and Economics*40 (1-3):3-73.

Grigg, I. 2005. *Triple Entry Accounting*. Available at: http://financialcryptography.com/mt/archives/000501.html.

Groomer, S. M., and U. S. Murthy. 1989. Continuous auditing of database applications: An embedded audit module approach. *Journal of Information Systems* 3 (2):53-69.

Iansiti, M., and K. R. Lakhani. 2017. The truth about blockchain. *Harvard Business Review* 95 (1):118-127.

Ijiri, Y. 1986. A framework for triple-entry bookkeeping. *Accounting review*:745-759.

Issa, H., and A. Kogan. 2014. A predictive ordered logistic regression model as a tool for quality review of control risk assessments. *Journal of Information Systems* 28 (2):209-229.

Jans, M., M. G. Alles, and M. A. Vasarhelyi. 2014. A field study on the use of process mining of event logs as an analytical procedure in auditing. *The accounting review* 89 (5):1751-1773.

Jordan, S. 2009. Implications of Internet architecture on net neutrality. *ACM Transactions on Internet Technology (TOIT)* 9 (2):5.

Kogan, A., M. G. Alles, M. A. Vasarhelyi, and J. Wu. 2014. Design and evaluation of a continuous data level auditing system. *Auditing: A Journal of Practice & Theory* 33 (4):221-245.

Kosba, A., A. Miller, E. Shi, Z. Wen, and C. Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. Paper read at Security and Privacy (SP), 2016 IEEE Symposium.

KPMG International. 2017. *Blockchain Accelerates Issuance Transformation*. Available at: https://assets.kpmg.com/content/dam/kpmg/xx/pdf/2017/01/blockchain-accelerates-insurance-transformation-fs.pdf.

Kuhn Jr, J. R., and S. G. Sutton. 2010. Continuous auditing in ERP system environments: The current state and future directions. *Journal of Information Systems* 24 (1):91-112.

Lagoudakis, M. G., and R. Parr. 2003. Least-squares policy iteration. *Journal of machine learning research* 4 (Dec):1107-1149.

Li, N., T. Li, and S. Venkatasubramanian. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. Paper read at Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference.

Machanavajjhala, A., J. Gehrke, D. Kifer, and M. Venkitasubramaniam. 2006. l-diversity: Privacy beyond k-anonymity. Paper read at Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference.

McCarthy, W. E. 1982. The REA accounting model: A generalized framework for accounting systems in a shared data environment. *Accounting review*:554-578.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Murthy, U. S., and S. M. Groomer. 2004. A continuous auditing web services model for XML-based accounting systems. *International Journal of Accounting Information Systems* 5 (2):139-163.

Nakamoto, S. 2008. *Bitcoin: A peer-to-peer electronic cash system*. Available at: https://bitcoin.org/bitcoin.pdf.

Nasdaq. 2015. *Nasdaq Linq Enables First-Ever Private Securities Issuance Documented with Blockchain Technology*. Available at: http://ir.nasdaq.com/releasedetail.cfm?releaseid=948326.

O'Leary, D. E. 2000. *Enterprise resource planning systems: systems, life cycle, electronic commerce, and risk*: Cambridge university press.

Ormoneit, D., and Ś. Sen. 2002. Kernel-based reinforcement learning. *Machine learning* 49 (2-3):161-178.

Pastena, V. 1979. Some evidence on the SEC's system of continuous disclosure. *Accounting review*:776-783.

Pathak, J., B. Chaouch, and R. S. Sriram. 2005. Minimizing cost of continuous audit: Counting and time dependent strategies. *Journal of Accounting and Public Policy* 24 (1):61-75.

Perols, J. 2011. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory* 30 (2):19-50.

Perols, J. L., R. M. Bowen, C. Zimmermann, and B. Samba. 2016. Finding needles in a haystack: Using data analytics to improve fraud prediction. *The accounting review* 92 (2):221-245.

Peter, M., and G. Timothy. 2011. The NIST definition of cloud computing. *NIST special publication* 800:800-814.

Peters, J., and S. Schaal. 2006. Policy gradient methods for robotics. Paper read at Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference.

Peterson, A. 2014. Hal Finney received the first Bitcoin transaction. Here's how he describes it. *The Washington Post*.

Piscini, E., J. Guastella, A. Rozman, and T. Nassim. 2016. Blockchain: Democratized trust: Distributed ledgers and the future of value: Deloitte University Press.

Plattner, H. 2009. A common database approach for OLTP and OLAP using an in-memory column database. Paper read at Proceedings of the 2009 ACM SIGMOD International Conference on Management of data.

PwC. 2016. *What's next for blockchain in 2016*. Available at: https://www.pwc.com/us/en/financial-services/publications/viewpoints/assets/pwc-qa-whats-next-for-blockchain.pdf.

Rackoff, C., and D. R. Simon. 1991. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. Paper read at Annual International Cryptology Conference.

Reid, F., and M. Harrigan. 2013. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*: Springer, 197-223.

Rezaee, Z., W. Ford, and R. Elam. 2000. Real-time accounting systems. *Internal Auditor* 57 (2):62-62.

Riedmiller, M. 2005. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. Paper read at European Conference on Machine Learning.

Riedmiller, M., T. Gabel, R. Hafner, and S. Lange. 2009. Reinforcement learning for robot soccer. *Autonomous Robots* 27 (1):55-73.

Rizzo, P. 2016. *Sydney Stock Exchange Completes Blockchain Prototype*. Available at: https://www.coindesk.com/sydney-stock-exchange-blockchain-prototype/.

Selinger, P. G., M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. 1979. Access path selection in a relational database management system. Paper read at Proceedings of the 1979 ACM SIGMOD international conference on Management of data.

Shubber, K. 2016. Banks find blockchain hard to put into practice. *Financial Times*.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529 (7587):484-489.

Simon, G. 2016. *Blockchain and the Introduction of Single-Entry Bookkeeping*. Available at: https://medium.com/@Loyyal/blockchain-and-the-introduction-of-single-entry-bookkeeping-5e5b201db09.

Summers, S. L., and J. T. Sweeney. 1998. Fraudulently misstated financial statements and insider trading: An empirical analysis. *Accounting review*:131-146.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3 (1):9-44.

Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10 (05):557-570.

Szabo, N. 1994. Smart contracts. *Unpublished manuscript*.

Vasarhelyi, M. A., and F. B. Halper. 1991. The continuous audit of online systems. Paper read at Auditing: A Journal of Practice and Theory.

Von Neumann, J. 2012. *The computer and the brain*: Yale University Press.

Yermack, D. 2017. Corporate governance and blockchains. *Review of Finance* 21 (1):7-31.

APPENDICES

Table 10: Computational Complexity Level of Audit analytics

	Time (s)			Size (MB)		
# of transactions	In-Memory Row-oriented Storage	Disk-based Row-oriented Storage	Disk-based Columnar Storage	In-Memory Row-oriented Storage	Disk-based Row-oriented Storage	Disk-based Columnar Storage
100000	0.03	0.02	0.01	\	1.63	6.58
200000	0.04	0.05	0.02	\	3.28	8.42
300000	0.07	0.09	0.02	\	4.93	12.17
400000	0.09	0.11	0.02	\	6.58	15.92
500000	0.13	0.15	0.02	\	8.23	19.67
600000	0.16	0.21	0.02	\	9.88	23.42
700000	0.18	0.21	0.02	\	11.53	27.17
800000	0.17	0.22	0.02	\	13.18	30.92
900000	0.19	0.25	0.02	\	14.83	34.67
1000000	0.21	0.28	0.03	\	16.47	39.05
1100000	0.27	0.32	0.03	\	18.12	42.80
1200000	0.27	0.33	0.03	\	19.77	46.55
1300000	0.31	0.38	0.03	\	21.42	50.30
1400000	0.30	0.39	0.03	\	23.07	54.05
1500000	0.33	0.40	0.03	\	24.71	57.80
1600000	0.34	0.44	0.03	\	26.36	61.55
1700000	0.37	0.47	0.03	\	28.01	65.30
1800000	0.38	0.50	0.03	\	29.66	69.05
1900000	0.41	0.52	0.03	\	31.31	72.80
2000000	0.42	0.54	0.03	\	32.96	77.17
2100000	0.45	0.58	0.04	\	34.61	80.92
2200000	0.47	0.61	0.04	\	36.37	84.67
2300000	0.49	0.63	0.04	\	38.13	88.42
2400000	0.55	0.66	0.04	\	39.89	92.17
2500000	0.54	0.69	0.04	\	41.65	95.92
2600000	0.61	0.72	0.21	\	43.42	99.67
2700000	0.58	0.75	0.04	\	45.18	103.42
2800000	0.64	0.78	0.04	\	46.94	107.17
2900000	0.62	0.80	0.04	\	48.70	111.55
3000000	0.65	0.83	0.04	\	50.46	115.30
3100000	0.70	0.86	0.04	\	52.22	119.05
3200000	0.72	0.87	0.05	\	53.98	122.80
3300000	0.70	0.92	0.05	\	55.75	126.55

3400000	0.73	0.94	0.05	\	57.51	130.30
3500000	0.76	0.97	0.05	\	59.27	134.05
3600000	0.77	1.00	0.09	\	61.03	137.80
3700000	0.80	1.02	0.04	\	62.79	141.55
3800000	0.86	1.06	0.05	\	64.56	145.30
3900000	0.86	1.07	0.05	\	66.31	149.67
4000000	0.90	1.11	0.05	\	68.08	153.42
4100000	0.87	1.15	0.07	\	69.84	157.17
4200000	0.94	1.18	0.05	\	71.60	160.92
4300000	0.93	1.22	0.06	\	73.36	164.67
4400000	0.99	1.22	0.06	\	75.13	168.42
4500000	0.95	1.25	0.06	\	76.89	172.17
4600000	0.98	1.27	0.06	\	78.65	175.92
4700000	1.03	1.29	0.06	\	80.41	179.67
4800000	1.04	1.34	0.05	\	82.17	183.42
4900000	1.05	1.36	0.06	\	83.93	187.80
5000000	1.07	1.40	0.06	\	85.69	191.55

TABLE 13: Inputs Variables

Variable Name	Description
aco	Current Assets Other Total
acox	Current Assets Other Sundry
act	Current Assets - Total
ao	Assets - Other
aox	Assets - Other - Sundry
ap	Accounts Payable - Trade
at	Assets - Total
bkvlps	Book Value Per Share
caps	Capital Surplus/Share Premium Reserve
capx	Capital Expenditures
capxv	Capital Expend Property, Plant and Equipment Schd V
ceq	Common/Ordinary Equity - Total
ceql	Common Equity Liquidation Value
ceqt	Common Equity Tangible
che	Cash and Short-Term Investments
cogs	Cost of Goods Sold
csho	Common Shares Outstanding
cstk	Common/Ordinary Stock (Capital)
cstke	Common Stock Equivalents - Dollar Savings
dcpstk	Convertible Debt and Preferred Stock
dcvt	Debt - Convertible
dd1	Long-Term Debt Due in One Year
dlc	Debt in Current Liabilities - Total
dltis	Long-Term Debt Issuance
dltr	Long-Term Debt Reduction
dltt	Long-Term Debt - Total
do	Discontinued Operations
dp	Depreciation and Amortization
dpact	Depreciation, Depletion and Amortization (Accumulated)
dpc	Depreciation and Amortization (Cash Flow)
dv	Cash Dividends (Cash Flow)
dvc	Dividends Common/Ordinary
dvp	Dividends - Preferred/Preference
dvt	Dividends - Total
ebit	Earnings Before Interest and Taxes
ebitda	Earnings Before Interest
fopo	Funds from Operations Other
gp	Gross Profit (Loss)
ib	Income Before Extraordinary Items

ibadj	Income Before Extraordinary Items Adjusted for Common Stock Equivalents
ibc	Income Before Extraordinary Items (Cash Flow)
ibcom	Income Before Extraordinary Items Available for Common
icapt	Invested Capital - Total
invt	Inventories - Total
itcb	Investment Tax Credit (Balance Sheet)
lco	Current Liabilities Other Total
lcox	Current Liabilities Other Sundry
lct	Current Liabilities - Total
lo	Liabilities - Other - Total
lse	Liabilities and Stockholders Equity - Total
lt	Liabilities - Total
ni	Net Income (Loss)
niadj	Net Income Adjusted for Common/Ordinary Stock (Capital) Equivalents
nopi	Nonoperating Income (Expense)
nopio	Nonoperating Income (Expense) Other
np	Notes Payable Short-Term Borrowings
oiadp	Operating Income After Depreciation
oibdp	Operating Income Before Depreciation
pi	Pretax Income
ppegt	Property, Plant and Equipment - Total (Gross)
ppent	Property, Plant and Equipment - Total (Net)
pstk	Preferred/Preference Stock (Capital) - Total
pstkc	Preferred Stock Convertible
pstkl	Preferred Stock Liquidating Value
pstkn	Preferred/Preference Stock - Nonredeemable
pstkr	Preferred/Preference Stock - Redeemable
pstkrv	Preferred Stock Redemption Value
re	Retained Earnings
recco	Receivables - Current - Other
rect	Receivables Total
rectr	Receivables - Trade
revt	Revenue - Total
sale	Sales/Turnover (Net)
seq	Stockholders' Equity - Total
spi	Special Items
sstk	Sale of Common and Preferred Stock
tstk	Treasury Stock - Total (All Capital)
tstkn	Treasury Stock Number of Common Shares
txdb	Deferred Taxes (Balance Sheet)

txdc	Deferred Taxes (Cash Flow)
txditc	Deferred Taxes and Investment Tax Credit
txp	Income Taxes Payable
txt	Income Taxes - Total
wcap	Working Capital (Balance Sheet)
xi	Extraordinary Items
xido	Extraordinary Items and Discontinued Operations
xidoc	Extraordinary Items and Discontinued Operations (Cash Flow)
xopr	Operating Expenses Total

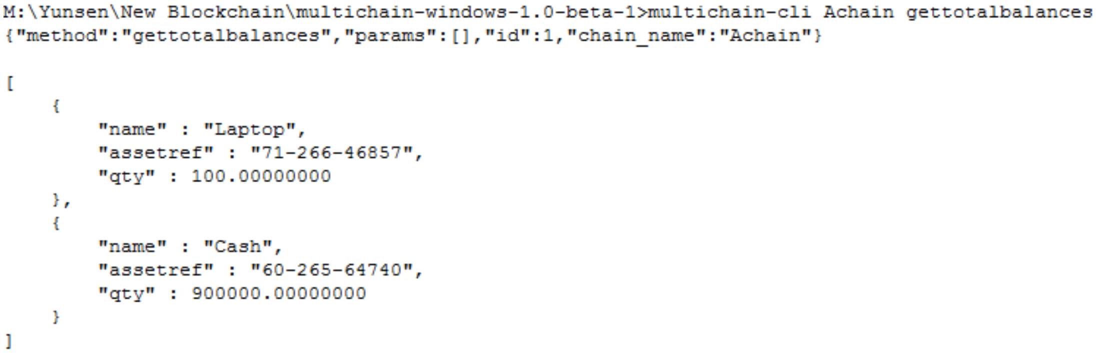
Table 16
Panel B: Distribution by SIC Industry Classification

Industry	SIC Code	Fraud		Nonfraud	
		N	Percent	N	Percent
Metal Mining	10	2	1.09%	125	0.44%
Oil and Gas Extraction	13	5	2.72%	448	1.57%
Mining and Quarrying of Nonmetallic Minerals, Except Fuels	14	1	0.54%	10	0.04%
Heamy Construction, Except Building Construction, Contractor	16	1	0.54%	10	0.04%
Construction - Special Trade Contractors	17	1	0.54%	7	0.02%
Food and Kindred Products	20	5	2.72%	406	1.42%
Textile Mill Products	22	1	0.54%	13	0.05%
Apparel, Finished Products from Fabrics & Similar Materials	23	5	2.72%	167	0.59%
Furniture and Fixtures	25	1	0.54%	16	0.06%
Paper and Allied Products	26	4	2.17%	156	0.55%
Printing, Publishing and Allied Industries	27	1	0.54%	24	0.08%
Chemicals and Allied Products	28	11	5.98%	4080	14.30%
Rubber and Miscellaneous Plastic Products	30	2	1.09%	52	0.18%
Stone, Clay, Glass, and Concrete Products	32	1	0.54%	42	0.15%
Primary Metal Industries	33	5	2.72%	322	1.13%
Fabricated Metal Products	34	3	1.63%	162	0.57%
Industrial and Commercial Machinery and Computer Equipment	35	9	4.89%	1894	6.64%
Electronic & Other Electrical Equipment & Components	36	14	7.61%	2647	9.28%
Transportation Equipment	37	5	2.72%	280	0.98%
Measuring, Photographic, Medical, & Optical Goods, & Clocks	38	5	2.72%	1447	5.07%
Motor Freight Transportation	42	2	1.09%	75	0.26%
Transportation by Air	45	1	0.54%	32	0.11%
Communications	48	10	5.43%	1991	6.98%
Electric, Gas and Sanitary Services	49	11	5.98%	1596	5.59%
Wholesale Trade - Durable Goods	50	3	1.63%	355	1.24%
Wholesale Trade - Nondurable Goods	51	7	3.80%	473	1.66%
General Merchandise Stores	53	1	0.54%	23	0.08%
Food Stores	54	3	1.63%	76	0.27%
Home Furniture, Furnishings and Equipment Stores	57	1	0.54%	33	0.12%

Eating and Drinking Places	58	2	1.09%	75	0.26%
Miscellaneous Retail	59	3	1.63%	96	0.34%
Business Services	73	44	23.91%	10437	36.57%
Health Services	80	7	3.80%	390	1.37%
Educational Services	82	1	0.54%	15	0.05%
Engineering, Accounting, Research, and Management Services	87	4	2.17%	340	1.19%
Nonclassifiable Establishments	99	2	1.09%	221	0.77%
Total		184	100.00%	28536	100.00%

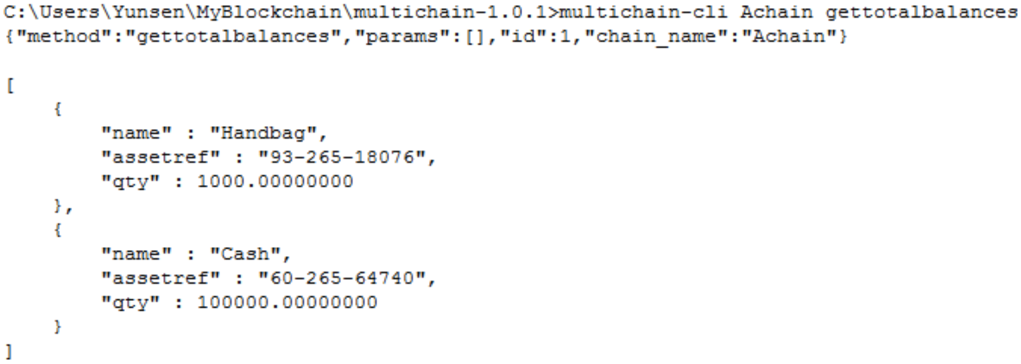
FIGURE 12: Order-to-Pay Transaction

Figure 12-1. Firm M’s Balance before Procurement



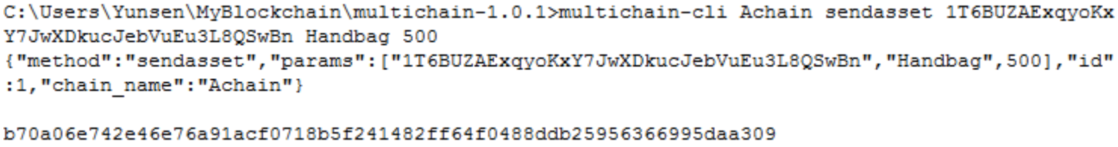
```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain gettotalbalances
{"method":"gettotalbalances","params":[],"id":1,"chain_name":"Achain"}

[
    {
        "name" : "Laptop",
        "assetref" : "71-266-46857",
        "qty" : 100.00000000
    },
    {
        "name" : "Cash",
        "assetref" : "60-265-64740",
        "qty" : 900000.00000000
    }
]
```

Figure 12-2. Supplier’s Balance before Procurement



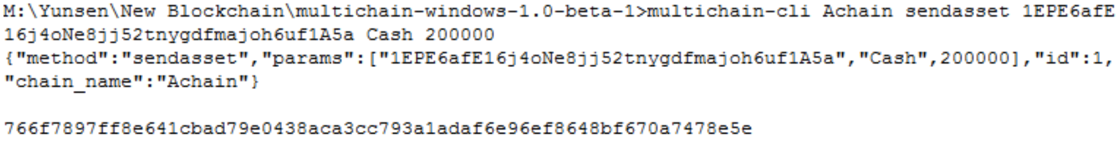
```
C:\Users\Yunsen\MyBlockchain\multichain-1.0.1>multichain-cli Achain gettotalbalances
{"method":"gettotalbalances","params":[],"id":1,"chain_name":"Achain"}

[
    {
        "name" : "Handbag",
        "assetref" : "93-265-18076",
        "qty" : 1000.00000000
    },
    {
        "name" : "Cash",
        "assetref" : "60-265-64740",
        "qty" : 100000.00000000
    }
]
```

Figure 12-3. Goods shipment



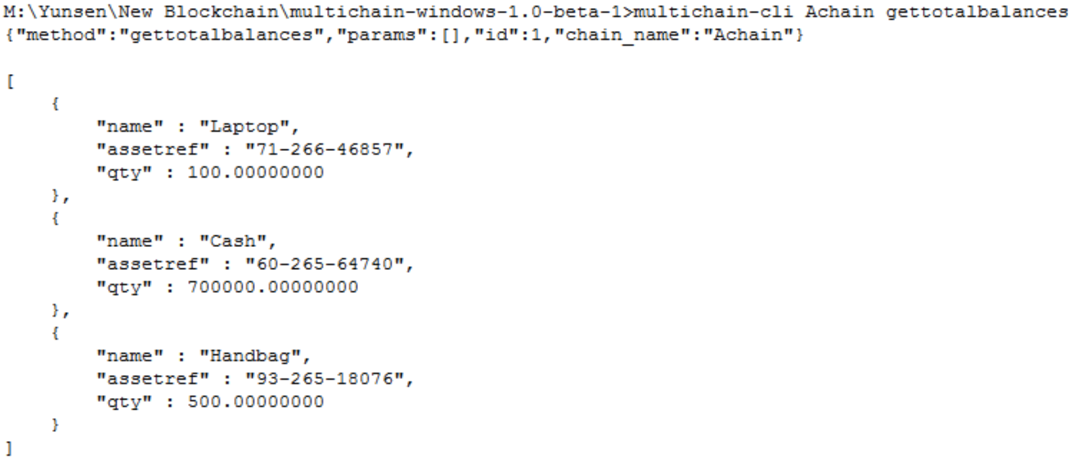
```
C:\Users\Yunsen\MyBlockchain\multichain-1.0.1>multichain-cli Achain sendasset 1T6BUZAExqyoKx
Y7JwXDkucJebVuEu3L8QSwBn Handbag 500
{"method":"sendasset","params":["1T6BUZAExqyoKxY7JwXDkucJebVuEu3L8QSwBn","Handbag",500],"id"
:1,"chain_name":"Achain"}

b70a06e742e46e76a91acf0718b5f241482ff64f0488ddb25956366995daa309
```

Figure 12-4. Disburse payment



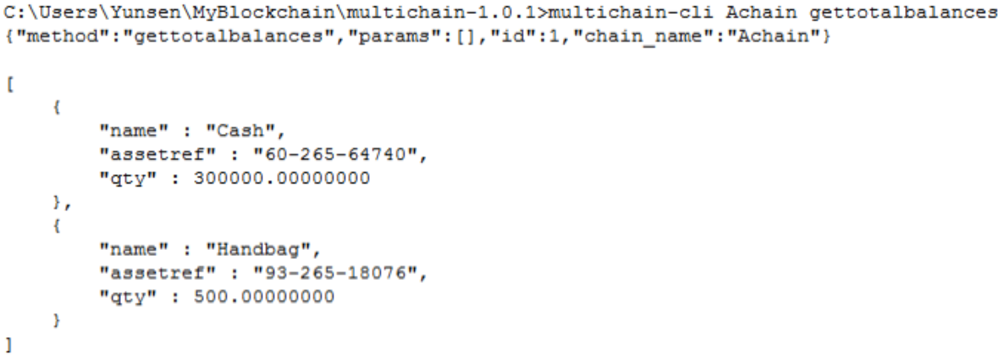
```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain sendasset 1EPE6afE
16j4oNe8jj52tnygdfmajoh6uf1A5a Cash 200000
{"method":"sendasset","params":["1EPE6afE16j4oNe8jj52tnygdfmajoh6uf1A5a","Cash",200000],"id":1,
"chain_name":"Achain"}

766f7897ff8e641cbad79e0438aca3cc793a1adaf6e96ef8648bf670a7478e5e
```

Figure 12-5. Firm M’s balances after procurement



```
M:\Yunsen\New Blockchain\multichain-windows-1.0-beta-1>multichain-cli Achain gettotalbalances
{"method":"gettotalbalances","params":[],"id":1,"chain_name":"Achain"}

[
    {
        "name" : "Laptop",
        "assetref" : "71-266-46857",
        "qty" : 100.00000000
    },
    {
        "name" : "Cash",
        "assetref" : "60-265-64740",
        "qty" : 700000.00000000
    },
    {
        "name" : "Handbag",
        "assetref" : "93-265-18076",
        "qty" : 500.00000000
    }
]
```

Figure 12-6. Supplier’s balances after procurement



```
C:\Users\Yunsen\MyBlockchain\multichain-1.0.1>multichain-cli Achain gettotalbalances
{"method":"gettotalbalances","params":[],"id":1,"chain_name":"Achain"}

[
    {
        "name" : "Cash",
        "assetref" : "60-265-64740",
        "qty" : 300000.00000000
    },
    {
        "name" : "Handbag",
        "assetref" : "93-265-18076",
        "qty" : 500.00000000
    }
]
```