A System View of Financial Blockchains

Conference Paper · March 2016

DOI: 10.1109/SOSE.2016.66

4 authors, including:

Wei-Tek Tsai
Arizona State University
424 PUBLICATIONS **6,004** CITATIONS

SEE PROFILE

A System View of Financial Blockchains

Wei-Tek Tsai* & Robert Blower
Digital Society & Blockchain Laboratory
Beihang University
Beijing, P. R. China

*Arizona State University
Tempe, AZ 85287, USA

Yan Zhu
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing 100083 P. R. China

Lian Yu
School of Software & Microelectronics
Peking University
Beijing, 102600. P. R. China

Abstract: This paper presents system-related issues for blockchain (BC) for financial applications. This paper first presents the design of a BC without consideration of any application scenarios, and issues such as performance, security, and scalability lead to specific BC designs. Sample BC scenarios are analyzed and these lead to additional BC designs. Specifically, two new kinds of BCs emerge: TBC (trade blockchain) for storing information at the transactional level, and ABC (account blockchain) for storing account information. Splitting traditional BCs into these two BCs allow one to optimize the system with respect to scalability and privacy.

1. Introduction

Blockchain (BC) has received significant attention recently as major financial institutions in the world announced that they will consider BCs in their operations. For example, R3 CEV and the Linux Foundation have announced projects with many financial institutions as well as technology companies such as IBM, Intel, Cisco involved. Furthermore, many central banks including the Australia, China, South Korea, Singapore, UK, US Central Banks have announced their projects to look at the adoption digital currencies or BC-based accounting ledgers.

Despite the early stage of development, financial institutions believe that BC technology can significantly reduce the complexity of bank processing and replace expensive database and middleware-processing applications. In addition, BC technology also supports fast multi-entity transaction settlement and clearing, and enhances fraud prevention and anti-money laundering protection. These opportunities have motivated many financial institutions to embrace BC hoping to increase banking efficiency and reduce cost at a time when profitability is under real pressure from growing information technology (IT) and operational costs, and falling revenue.

A BC is often viewed as a supporting component for a cryptocurrency. Furthermore, when BC technology is mentioned, it is often associated with a set of concepts such as cryptocurrency, encryption algorithms, peer-to-peer (P2P) protocol, voting mechanisms such as mining, and distributed ledgers. Are all these features needed if a BC is to be used for banking applications? When a BC is used, it needs to be integrated within existing banking processes initially and thus many system integration and scalability issues will be encountered. This paper examines some of these issues, and discusses their implications with respect to the BC designs as well as the design of banking applications.

While numerous organizations have voiced their support for BCs, some also expressed their reservations. For example, Euroclear recently released a white paper supporting the potential of BCs while also expressing reservations. It concluded that a number of issues needs to be addressed before BC technology will have widespread acceptance. The report also remarked that existing techniques such as Central Securities Depository (or CSD) may also perform the same functions as BCs. Other institutions also expressed their reservations including DTCC [DTCC 2016, Higgins 2016] and Ripple Insight [Liu 2016].

This paper focuses on system issues such as software and hardware that may enable BCs to be integrated in financial applications. Our contributions of this paper are as follows:

a) *Presents a system point of view of BCs*. When BCs are applied to banking applications, new features need to be incorporated such as using high-speed networks instead of P2P networks, and fast consensus protocols should be used rather than the mining process used in common BCs. This is done in Section 2. Our laboratory has developed a prototype BC using these concepts.
b) *Put forward ways that BCs can be used at scale and speed*. Many reservations on BCs came from the absence of defined operational architecture,

that partly came from the currently limited BC demonstrations in real applications. Furthermore, current BC designs have features that are not suitable for banking applications. By removing or modifying BC features to cater for these limitations, the future BC environment will enable true financial disruptions as many have envisioned earlier. This paper presents new BC designs to support financial applications based on published potential banking application scenarios. These features include making BCs to focus on only one aspect of operations only, i.e., trading or account, but not both. In system theory as well as software engineering, one subsystem or module is best to handle one function only. Once the functionality is isolated, these BCs can be optimized and scaled easily.

This paper is organized in the following manner: Section 2 reviews the BC requirements and their corresponding design issues; Section 3 presents sample BC scenarios for financial applications; Section 4 present new BC designs to support financial transactions; and Section 5 concludes this paper.

2. Requirements and their Designs

2.1. Financial System Requirements

Most financial systems need these attributes:

High throughput and low latency performance: For example, stock trading systems need to execute or record transactions at a high rate such as 100K TPS (transactions per second). The UK Chief Scientific Adviser recommends that "The blockchains used should be high-performance, low-latency and energy efficient." [UK 2016]

Security and Privacy: Financial systems must have these features otherwise they cannot be used.

Compliance: Financial operations must be monitored carefully to prevent any breaches of money laundering, and it should have anti-frauds controls to ensure compliance with exchanges.

Reliability and persistence: Any outage of technology may results in significant financial loss for financial institutions, and given the volume of transactions, the economy as a whole. For example, if a stock-trading system failed during the market hours, hundreds of millions of dollars would be lost immediately. Financial systems can be SIPS (Systemically Important Payment Systems) and the failure of SIPS such as RTGS (Real-Time Gross Settlement) will cause the whole economic of a country to halt. Federal Reserve System (for the US), TARGET2 (for inter-bank payments in Europe), and STEP2 (European clearing system) are SIPS.

The needs for resilience in systems is a common theme for financial systems and financial institutions. For example, Bank of International Settlement (BIS) published a list of requirements for digital currencies that include business model sustainability, security, scalability, efficiency, cost, usability, cross-border reach, privacy, marketing, and reputation as key system requirements [BIS 2015].

2.2. Current Blockchain Design

Currently, a BC typically has the following features:

Encryptions and cascaded encryption: Blocks are encrypted in a cascaded manner, i.e., the encryption result of the previous block will be used in the encryption of the current block. Thus, if any one changes a block, any subsequent blocks will produce different encryption results. Many of cryptocurrency systems are both ledgers and transaction platforms such as Bitcoin, and in these systems blocks are encrypted but transaction information is open to the public. Anyone did any transaction will leave a trace known to the world.

Timestamps: Each data item in a BC will have a timestamp.

P2P network: all the nodes participating in a BC are connected in a P2P network.

Mining: Each node will maintain the distributed ledger and it does this by a mining mechanism in which computation is performed. In return for their computation to maintain the ledger, the system returns certain amounts of cryptocurrency as a reward to the node.

Digital currency: If the ledger of a BC tracks an asset, the ledger can be used to issue a digital currency and perform financial transactions in that currency [Martin 2014]. Martin explains that it is the distributed ledger, maintained in a cryptocurrency, that creates the digital currency, not the mining mechanism. He further explains that such currency mechanisms were developed initially in the 16th century in Europe except at that time ledgers were stored in banks not in a digital media. Mining is one of many consensus protocols that can be used to maintain the consistency of distributed ledgers, and a slow protocol. PBFT (Practical Byzantine Fault Tolerance) is another protocol often used in BCs [Castro 1999].

Multiple independent copies: All participating nodes in a BC will contain the complete ledger with all the blocks in the BC.

These features are in the first generation of cryptocurrency Bitcoin, as well as in the 2nd generation cryptocurrency such as Ripple (ripple.com), BitShares (bitshares.org), and Ethereum (ethereum.org). At this time, new BCs are being designed each day. For example, the Hyperledger project started by the Linux Foundation has not released their design yet; Hydrachain, a private BC derived from Ethereum, has different design [Tsai 2016]; BeihangChain is a new design of a private BC.

2.3. Analysis

If one examines the requirements for financial systems and current BC designs, there is a significant gap between them. Many BCs operate at a rate that is far from being applicable to many of today's financial processes. Another important issue is the impact of *regulation* because many cryptocurrencies have largely not addressed the regulation issue.

The BIS report quoted lists key regulatory actions for digital currencies: 1) information/moral suasion such as public warning; 2) specific stakeholder regulation such as regulation of digital currency administration or exchanges; 3) interpretation of existing regulations such as explanation of how digital currency may be regulated by laws of a specific country; 4) overall regulations such as regulatory bodies to oversee the related operations; and 5) prohibitions such as banning retail transactions by some digital currencies.

The European Banking Federation (EBF) made key recommendations concerning the application of cryptocurrency to banking:

"1) Conduct a joint assessment by both government and industry participants on the opportunities and impact of crypto-technologies; 2) Build a comprehensive regulatory approach to crypto-technologies to help overcome uncertainty for legitimate users; 3) Make transactions subject to the same regulatory standards (ref. Anti-Money Laundering/Anti-Terrorist Financing)" [EBF 2015].

The EBF report also stressed the importance of taking a system approach to address these issues.

2.4. Financial Blockchains

These regulation, security, privacy, and performance requirements lead to the need for new BC designs that focus more on the need ensure resilience, timeliness, and transparency to regulation. .

Encryption and timestamps: These are great features for financial BCs. But financial BCs need to protect the privacy of transactions performed or recorded at BCs. Thus financial BCs will need additional security and privacy mechanisms.

P2P networks: P2P networks by their nature fault-tolerant and make use of novel communication technology. These kinds of networks can be used in a variety of systems and applications such as a cloud platform. For example, a P2P network can be used to track copies of data so any failure of a component will not cripple the whole system or lose data, and this has been used by Amazon Dynamo [Hastorun 2007]. But a significant drawback is that it is difficult to monitor and control P2P applications as operations may be autonomous and decentralized.

Furthermore, one motivation for P2P networks was to avoid government regulations, and irregularities such as copyright infringement and security leaks, which have been reported regarding to these kinds of networks. Shawn Fanning, who pioneered the P2P network Napster, stated that his reason for developing Napster was that "Participating users establish a virtual network, entirely independent from the physical network, without having to obey any administrative authorities or restrictions." Thus, the design goal of a P2P network (to avoid all regulations) is in a direct contradiction to a principal design goal for any financial systems (compulsory regulation).

Furthermore, each node in a P2P network serves as a client as well as a server, and the performance of a P2P network is inherently slower than a regular network. While a P2P network is fault-tolerant. This this feature comes with the price of low performance. Moreover the multipath connection inherent in a P2P network creates barriers to regulation. Given the obvious issues of compliance and privacy protection, this feature needs to be excluded.

Mining: Once the P2P network is excluded, the mining mechanism that rewards those nodes that offer computation and storage to maintain distributed ledgers also needs to be removed as it is now irrelevant. For banking applications, participating banks maintain those ledgers, and thus miners and the mining process are not needed. Instead, consensus protocols can be used instead to achieve the same functionality across ledgers.

Digital currency: Central banks are concerned with systemic risk in payments and in ensuring the highest operational aspects are maintained [BIS 2015]. However, as the European Banking Association (EBA) stated a distributed ledger may, or may not, result in a digital currency, it may contains digital references to assets such as immediately available cash liquidity, stocks and bonds instead. In these circumstances, distributed ledgers enable settlement to occur through the consensual reallocation of the balance. Accordingly, it is possible to issue a digital currency, settlement may occur with other types of digital assets such as tokens representing fiat currency, something of value such as airline mileage, or community tokens for charity. A BC with these assets may engage in a variety of financial applications, not just digital currencies, in settlement of such as foreign exchange, remittance, real-time payments, documentary trade, and asset servicing [EBA 2015, Pick 2015].

Multiple independent copies: If only one copy of the ledger is maintained, this becomes a centralized system. For example, currently commercial banks need to go to their central banks for clearing across the central bank's accounts. This is efficient and can be regulated easily as central banks can process these efficiently. However, this creates rigidity that results in siloes of liquidity and collateral that are neither transportable nor interchangeable. Accordingly economies, banks, or companies operating across the world are vulnerable to shocks where a failure to have sufficient liquidity in one place can result in a contagion. The central-bank system must be reliable and trustworthy enough, as any failure or unlawful activities can have serious consequences as these systems are SIPS. Currently a CSD is an example centralized system that offer real-time settlement and efficient operations [Euroclear 2016].

Making multiple copies of a ledger will add significant cost for communication, computation, and storage due to the need to run consensus protocols and storing data multiple times. These are expensive operations as numerous transactions will be performed in financial systems. Furthermore, many of these protocols need to be executed in a sequential manner, thus they will not be efficient even they are given additional processors and/or bandwidth.

However, multiple copies will increase system reliability and security. For example, Byzantine protocols normally can tolerate the failure of 1/3 of nodes in a BC before the system will fail [Castro 1999]. Figure 1 shows the average number of years for a BC to fail with a given probability of a node and the number of nodes in the BC. The horizontal axis specifies the numbers of nodes, and vertical axis the number of years for more than 1/3 of the nodes to fail once.

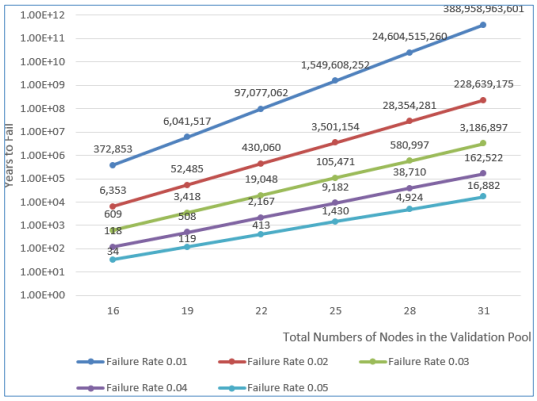

Figure 1: Reliability Analysis

Let *n* be the number of nodes in the system, p the discrete probability distribution of the failure rate of each node, and the years for more than 1/3 of the nodes to fail once can be calculated as

$$\text{Years} = 1/(\sum_{k=\frac{n}{3}+1}^{n} C_n^k p^k (1-p)^{n-k})/365$$

A BC with 16 nodes and node failure rate 0.01 per day will take about 373,000 years for the system to fail once; and when there are 31 nodes, it takes about 389 billion years for the system to fail once. Thus, a BC does not need many nodes to have reasonable reliability.

Table 1 summarizes the discussion between the original BC design versus financial BC design.

Table 1: Summary of BC Designs

	Original BCs	Financial BCs
Encryption and cascaded encryption	Yes, but transaction information is public	Yes, and transaction information is private
Timestamps	Yes	Yes
Multiple independent copies	Yes, usually all the participating nodes in the P2P network	Yes, need to choose the number of nodes
Network	P2P networks	High-speed networks
Mining	Yes	Consistency protocols
Digital Currency	Yes	Yes but not absolute

As one increases the number of nodes, the system will become more reliable, but at the same time the system will slow down. Any consensus protocols will need a node to broadcast its status or transactions to other nodes, and thus for each block creation, $O(n^2)$ messages will be generated.

Thus one has a tradeoff between these attributes. If a BC has more nodes, it will be more reliable, but the system will operate at a lower speed. This issue will be addressed further in Section 4 once application scenarios are better understood as this feature will also affect application architecture.

2.5. Hydrachain

Hydrachain is a private BC developed based on Ethereum, a public BC. It uses PBFT [Csstro 1999] to maintain the consistency of ledgers at nodes in the BC rather than a mining mechanism [Tsai 2016], and its speed is much faster than those that use the mining process. Figure 2 shows the Hydrachain architecture.

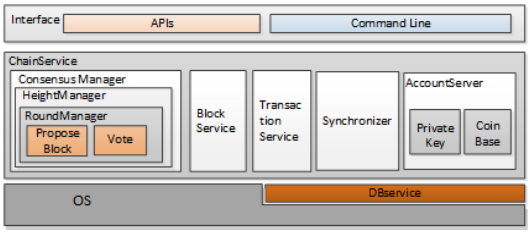

Figure 2: HydraChain Architecture

2.1 BeihangChain

Beihang University and Peking University have jointly developed a private chain BeihangChain, and the architecture is shown in Figure 3.

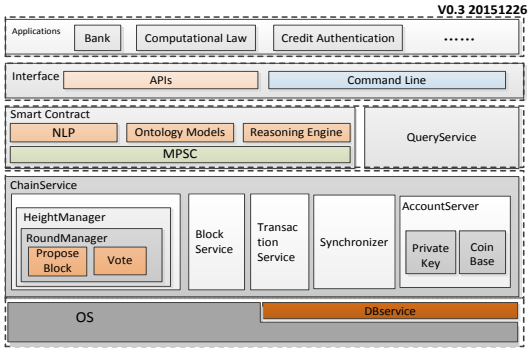

Figure 3 BeihainChain Architecture

In BeihangChain, Byzantine voting and data collection are carried out concurrently to speed up the process, and thus it has a unique block creation process. Furthermore, not only blocks are voted on, but individual transactions are also voted on too. This will allow transaction data to be collected while blocks are being voted on. To ensure security, block creation results are voted on to identify any participating nodes compromised. Due to these three rounds of voting, more messages will be generated (each needs $O(n^2)$ messages), but due to concurrent operations, speed can be improved. In some configurations, BeihangChain reached 24K TPS. These are obtained without

hardware optimization, load balancing, data re-organization, or asynchronous operations. These will further improve the speed significantly.

Table 2 shows various features in Hydrachain and BeihangChain.

Table 2: HydraChain and BeihangChain

Aspects	**HydraChain**	**BeihangChain**
Block creation	A leader creates a new block	Any node can create new blocks
Voting	On each new block; Byzantine voting	On each transaction, new blocks, and block voting results; Byzantine voting
Voting Failure Handling	Vote again if it received 1/3 of votes	Each transaction will have at least *n* (currently 5) chances to get vote in
Transaction Data Encryption	Yes	Yes
Processing	Sequential operations	Simultaneous voting and data collection
Leader selection	Round robin	Multiple strategies
Reputation systems	No	Yes to identify cheating nodes
Speed	1K TPS	12K TPS

3. Application Scenarios

BCs can be used in a variety of scenarios, EBA lists four scenarios [EBA 2015] and it can be examined how these will affect BC designs.

Scenario 1 Foreign Exchange: Each region can have a gateway, and the gateway will hold digital assets that have collateral in a fiat currency or securities systems. These participating gateways will form a BC so that foreign exchange transactions can be made at these gateways.

Scenario Analysis: This means a collection of gateways will serve as nodes in a BC, each keeping a copy of the shared ledger. When two trading gateways make a transaction, all participating gateways will update their copies of the ledger. This means that each region needs to update its information on a needed basis by uploading the account information involved in transactions to gateways, and transaction data need to be downloaded from gateways to appropriate accounts in an appropriate bank. For example, an Australian gateway may keep track of a transaction between a German account and a US account even though it has no direct interest in this transaction. After the transaction is settled, the German gateway needs to copy data back into the account in a German bank, and the same for the US gateway to update a US bank, but the Australian node may also be needed in the event of a failure to reconcile or unavailability of the "receiving" or "sending" nodes.

Scenario 2 Real-Time Payment: Instead of having a central bank to deal with payments from commercial banks, a collection of nodes in a BC can clear payments on a continuous basis.

Scenario Analysis: This scenario implies that all the participating commercial banks will publish and share their account data, and act on them. However, given the size of account bases this sharing may be related to the bank's own accounts, not client accounts. In this case, the sharing will be enormous and clients may object to this design. For example, Alice has accounts in Bank A, Bob in Bank B. As Bank A and B participate in a BC, Bank A will contain data about Bob's information, and Bank B will contains Alice's information. In addition to the large storage requirements, computation will be an issue due to large numbers of accounts and transactions.

An alternative scenario will be similar to the Foreign Exchange scenario. Each bank will store its account information, and only those involved will be moved to the BC. This means that participating banks will send/receive significant number of messages.

These two alternative scenarios need to compete with the original design where a central database (such as a CSD) maintained by a central authority, and all participating commercial banks will access the same database for settlement and clearing.

Scenario 3 Documentary Trade: This is related to trade finance and it involves open account, letters of credit, and consignment.

Scenario Analysis: A pilot project is being conducted by DBS and Standard Chartered Bank to experiment this approach by working on the invoice part of trade transactions. According to the report, this scenario is supposed to help in detecting potential frauds. Specifically, they stated that "The flaw in the current system is easy to see. Currently, trade finance transactions only exist between a borrower and a bank… that would reveal every recorded transaction between the ledger's participants." [TodayOnline.com 2015]

However, to do this, it means that every bank that may be involved with trade financing need to be on the BC. But this is likely not feasible unless only major players are involved. This also means that every participating bank will know at least some details about every trade financing done. On the other hand, this mean an object-oriented design can be used as each asset involved in financing may have an digital identification (ID), and this ID may be open to all the banks and other participants involved in trade finance.

Scenario 4 Asset Servicing: This involves creation of assets, enablement of trading between partners, and liquidation of positions.

Scenario Analysis: This will be a complex scenario as many parties will be involved with different kinds of assets.

In summary, different scenarios call for different BCs. If a BC holds only one kind of data or asset, then it is easier to optimize the BC design with respect to the asset held. For example one BC for cash account information, one for security holding information, one for trade financing information, one for collateral information, and one for real-time transactions.

4. Blockchain Designs

The scenarios indicated some BCs are mainly for *trading*, while others for *account* information.

TBC (Trading Blockchain): This kind of BCs will store information useful for trading. For example,

if account 123 from Bank A will trade with account 456 from Bank B, this TBC will store account information about accounts 123 and 456 for transactions and settlement.

ABC (Account Blockchain): This kind of BCs will hold account information such as cash balance, stock portfolios, and derivative contracts.

In the early BC designs, a BC is both a TBC and ABC at the same time, but this will complicate the design especially if the application scenario is already complex such as trade-financing applications.

4.1. TBC Operations

A TBC will store only information necessary to carry out trades and settlements only, and it will not store all the account information for all the participating banks. It will use the following process to carry out trades. Assuming without loss of generality, only two banks A and B are involved in a TBC:

- Bank A will use an authenticated method to copy the account data from Bank A ledger (an ABC) into the TBC. Likewise for Bank B. Effectively, both banks guarantee that the data uploaded are accurate, and any subsequent participations, e.g., from a Central Bank or Clearing House, will only add to security of the underlying data.
- The TBC will perform or record the transaction depending on if the matching is done at the TBC or elsewhere, and settle the transaction using the data stored in the TBC.
- The TBC will use an authenticated method to copy the transaction data back into Bank A's ABC ledger, and it will guarantee the data copied are correct. Likewise, the TBC will copy the transaction data into Bank B's ABC ledger.

Furthermore, after the transactions are over, the data in the TBC will be marked as "expired" indicating that the data are no longer available for transaction. Thus, the data in a TBC has a *timed* life, thus "TBC" can also mean "timed blockchain". The records stored in a TBC are permanent and not changeable like regular BCs, but the data will be useful for transactions for a period only. Expired data on a TBC are still useful for validation.

The data stored in a TBC can also be encrypted so that only participating banks can see the data, for example banks other than A or B cannot see the data.

This design also match well with the scenario described by Marc Robert-Nicoud CEO of Clearstream, ".. Alternatively, access to blockchain database can also be configured to participants' needs in a permissioned network. This customised access means that in post-trading, banks could be given access to a blockchain while the underlying client data could only be seen by the relevant banks and by all regulators." [Robert-Nicoud 2015]

This design has advantages:

1) **Optimization**: a TBC does not need to keep most of its blocks online once those blocks are not useful for trading. Only the most recent blocks will contain data useful for transactions. For example, any blocks that are one month old may be stored elsewhere to make room for high-speed execution.

Furthermore, a TBC may maintain multiple tracks of blocks, for example, one for those accounts with short tenure (such as those day-trade accounts), and another for those accounts with long transaction tenure (such as those smart contracts that can be triggered as long as accounts have money). This will lead ways to optimize data storage and performance.

Furthermore, old blocks can be processed in the background to produce indexing, and data can reorganized and saved into a backend database to speed up any analysis or queries. As a TBC needs to cross validate its data with the data stored in the participating ABCs to maintain their consistency, such indexing will speed up this process.

2) **Regulatory Enforcement**: A node participating in a TBC may be a regulatory agent, and they may inspect the data, and depending on requirements, they will have rights to participate in the voting. For example, for high-value transactions, a regulatory agency may have rights to stop a transaction if the transaction is suspected to have an issue. Otherwise, the regulatory nodes may just watch over transaction trading and record the data once it is done. The regulatory checks can be performed before the transaction, during the transaction, and after transaction.

3) **Privacy**: This design will ensure that only those banks that need to see data can see data, and data will be available for the needed time only. This design is consistent with the Windhover principle [Clippinger 2014] where individuals can keep their privacy while regulators can perform legitimate auditing and enforcement.

4) **Messages**: In addition to normal BC operations, a TBC needs to copy data from ABCs before transactions, and copy back the data after transactions to ABCs. These messages can be stored in these BCs and they can be used for analysis and failure recovery if necessary.

5) **Scalability** and **Locking**: Potentially, any group of financial institutions (ABCs) can start a TBC and share their transactions, and thus multiple TBCs can be formed. In this case, trading speed can be improved as a bank may involve transactions at multiple trading exchanges (TBCs), splitting the trading workloads among multiple exchanges associated with multiple TBCs.

However, a locking mechanism will be needed to ensure transaction integrity. For example, if Bank A participates in two TBCs, TBC_1 and TBC_2, and thus Bank A may place two trades at these two TBCs simultaneously, this will result in a *double spending* problem. This problem can be solved by placing a lock on specific accounts at ABC_A for those risked balances. When the relevant TBC results are returned to the ABC_A, the lock will be released.

Thus, accounts in an ABC will have at least two components, say *total-balance* and *risk-capital* (or *risk-asset*). An account can have one total-balance, but multiple *risk-capital*. Each risk-capital can be used in

one TBC transaction only, and there will be a lock on each risk-capital. The lock can be encrypted so that only the associated TBC can release the lock. An account may be involved in multiple transactions at different TBCs with multiple *risk-capitals*. For example, a client may engage in smart contracts to trade multiple stocks at different TBCs while withdraw money from her saving accounts at the same time.

4.2. ABC Operations

An ABC stores account information within a financial institution or a family of institutions. For example, multiple branches of a bank as well as third-party agents can host an ABC, and within the ABC, information is shared. In this way, bank employees will have difficulties to modify the account information illegally without being caught.

An ABC can have multiple designs, for example, the account information can be stored in a hash for O(1) retrieval , or in a Merkle Patricia Tree where the most recent data can be easily retrieved.

In addition to *copy-from*, *copy-to*, and *lock* (and unlock) operations (described in Section 4.1), an ABC is also scalable. An initial ABC may maintain one chain of account information, and as the account number grows, the BC can no longer handle the workload for high performance. The BC can be split into two or more sub-chains. For example, chain 1 at block *k* is split into 2 chains, sub-chains 1-1 and 1-2, starting at block *k+1*. Each sub-chain 1-1 (and 1-2) with the original chain together form an integrated BC, thus effectively one has two BCs with a common root. The common root can be easily duplicated so that two new BCs are formed. In this case, each child BC can be hosted by different processors, and a load balancer or dispatcher commonly used in cloud computing can be used to split the workload. In this case, an automated and orderly BC evolution can be developed as each child BC can continue to be split its own child BC to accommodate new accounts. The splitting can be done in an incremental manner, or done in a tree manner [Tsai 2013]. In this case, performance at these ABCs can be maintained in an orderly manner.

According to queuing theory, there is a tradeoff between throughput and delay. Thus, a bank may decide to split an ABC once the BC delay becomes unacceptable, the BC will be split trading a lower throughput for a low latency. Figure 4 illustrates this where a BC is split into multiple parallel BCs, each handling a subset of accounts only. In this way, the BC performance can be scaled when the workload increases. This is similar to horizontal partition in databases, but this is now done for BCs.

A financial institution can perform optimization by placing most active accounts in certain BCs to be supported by powerful machines for differential treatment. One can achieve this by creating a new account in a high-speed ABC with a reference to the old account in the original ABC to keep the complete history of the account. Furthermore, high-speed BCs will have a low size limit before it will be split, and when the size exceeds the limit, it will be split into multiple child ABCs to be hosted on different machines to keep the workload balanced.

These features will allow BCs eventually to serve as a BaaS (BC-as-a-Service) with those common SaaS features such as scalability, fault-tolerance, and dynamic provisioning [Tsai 2012, Tsai 2014].

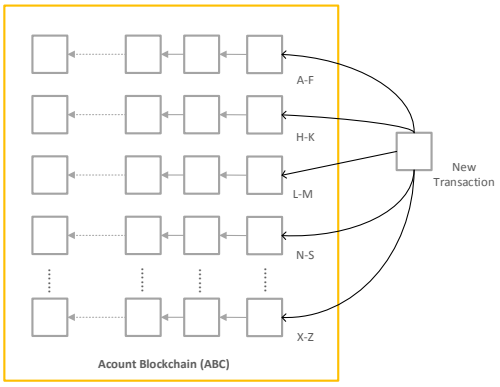

Figure 4 Scalability Diagram

5. Conclusion

This paper describes issues related to using BCs for financial applications. Financial systems need to have high throughput, low latency, high reliability, high security and privacy, and strict regulatory enforcement, but the current BCs have low throughput, high latency, low privacy, and are without a comprehensive regulatory framework. In fact, some original BC features were selected to avoid regulatory restriction or enforcement. These need to be addressed in financial BC designs.

Furthermore, while many high-level scenarios have been described as potential use cases, the implications of these scenarios to BC design have not been thoroughly addressed. Many innovative BC design can be developed for financial applications, but they need to be assessed according to the same standard rules for financial systems on performance, resilience, security, cost, reliability, and scalability.

6. Acknowledgement

This work is supported by National Key Laboratory of Software Environment at Beihang University, National 973 Program (Grant No. 2013CB329601) and National Natural Science Foundation of China (Grant No. 61472032).

7. References

[BIS 2015] Bank for International Settlements, Committee on Payments and Market Infrastructure, "Digital Currencies," Nov. 2015. http://www.bis.org/cpmi/publ/d137.pdf

[Castro 1999] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," Proc. of ACM Operating System Principles, 1999.

[Clippinger 2014] John H. Clippinger and David Bollier, editors, *From Bitcoin to Burning Man and Beyond: The Quest for Identity and Autonomy in a Digital Society*, Off the Commons Books, 2014.

[DTCC 2016] DTCC, "Embracing Disruption: Tapping the Potential of Distributed Ledgers to Improve the Post-Trade Landscape," Jan. 2016, file:///Users/tsai/Downloads/DTCC-Embracing-Disruption.pdf

[EBA 2015] Euro Banking Association (EBA), EBA Working Group on Electronic and Alternative Payments, "Cryptotechnologies, a Major IT Innovation and Catalyst for Change," May 11, 2015. https://www.abe-eba.eu/downloads/knowledge-and-research/EBA_20150511_EBA_Cryptotechnologies_a major IT innovation v1 0.pdf

[Euroclear 2016] Euroclear and Oliver Wyman, "Blockchain in Capital Market," white paper, Feb. 2016, http://www.automatedtrader.net/Files/z/Euroclear_Wyman_REPORT.pdf

[EBF 2015] European Banking Federation (EBF), "Driving the Digital Transformation: The EBF Blueprint on Digital Banking and Policy Change," Sept. 8, 2015, http://www.ebfdigitalbanking.eu/EBFDB_3.html

[Hastorun 2007] Deniz Hastorun , Madan Jampani , Gunavardhan Kakulapati , Alex Pilchin , Swaminathan Sivasubramanian , Peter Vosshall , Werner Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," Proc. of SOSP, 2007.

[Higgins 2016] Stan Higgins, "DTCC Report Cautions Building Blockchain Hype," Coindesk.com new, Jan. 25 2016. http://www.coindesk.com/dtcc-report-blockchain-hype/

[Liu 2016] A. Liu, "2016 Will be the Year that You Realize that You do not need the Blockchain," Ripple Insight, Jan. 8, 2016 https://ripple.com/insights/2016-will-be-the-year-you-realized-you-dont-need-the-blockchain/

[Martin 2014] F. Martin, "Bitcoin is Pointless as a Currency, but it could Change the World anyway," Wired, March 31, 2014. http://www.wired.com/2014/03/bitcoin-currency_martin/

[Pick 2015] L. Pick, "EBA Crypto Report Envisages Multiple Use Cases Scenarios," May 13, 2015. http://www.financemagnates.com/cryptocurrency/news/eba-crypto-report-envisages-multiple-use-cases-scenarios/

[Standard 2015] Standard Chartered & DBS Work on Blockchain Tech for Trade Finance, https://www.cryptocoinsnews.com/standard-chartered-dbs-work-on-blockchain-tech-for-trade-finance/

[Robert-Nicoud 2015] Marc Robert-Nicoud, "Is Blockchain Technology Really the Solution to All Our Problems?" TMI (Treasury Management international) Issue 239, Dec. 15, 2015, page 1. http://www.clearstream.com/blob/79108/7a8bff9b09f9dd52258c0947a4e21b7b/fintech-mrn-1512-data.pdf

[Singleton 2015] Andy Singleton, "Third Generation Blockchain – Big-Time Trading, Messaging, and Ledger," Sept. 8, 2015. https://medium.com/@andysingleton/the-third-generation-of-blockchain-tech-will-mix-and-match-with-real-world-systems-93b6cc3b1eb9#.e2fdhs3fn

[Steinmetz 2005] R. Steinmetz and K. Wehrle, "What Is This "Peer-to-Peer" About?" Springer Berlin Heidelberg, 2005, pp. 9-16.

[Tsai 2012] W. T. Tsai, Y. Huang, X. Bai, and J. Gao, "Scalable Architecture for SaaS," Proc. of IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORC), 2012, pp. 112-117.

[Tsai 2013] W. T. Tsai, G. Qi, and Z. Zhu, "Scalable SaaS Indexing Algorithms with Automated Redundancy and Recovery Management," Int J Software Informatics, Volume 7, Issue 1, 2013, pp. 63–84.

[Tsai 2014] W. T. Tsai, X. Bai, and Y. Huang, "Software-as-a-Service (SaaS): Perspectives and Challenges," Science China Information Sciences, Vol. 57, No. 5, 2014, pp. 1-15.

[Tsai 2016] W. T. Tsai, L. Yu, C. J. Hu, Y. F. Yao, and G. N. Li, "Hydrachain: Design of A Private Blockchain," Technical Report, Jan. 2016, Beihang University.

[Todayonline 2015] Todayonline.com, "DBS, StanChart Work on Distributed Ledger for Trade Finance," Dec. 18, 2015, http://www.todayonline.com/business/dbs-stanchart-work-distributed-ledger-trade-finance

[UK 2016] UK Government Chief Scientific Adviser, "Distributed Ledger Technology: Beyond Block Chain, " Government Office for Science, 2016, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf