BOLT: Booster of Ledger Technology White paper

2018/05/25 Version 1.1

Blockchain booster、scaling solution、solving the public chain’s lack of bandwith and the private chain’s lack of global consensus

TIDEiSUN Group

TIDEiSun InfiniteChain Co. Ltd.: We are committed to the development of blockchain technology. Our innovations and improvements include the main chain, expansion solutions, side chains, blockchain algorithms, artificial intelligence and business solutions.

Index

BOLT: Booster of Ledger Technology

White paper 2018/05/25 Version 1.1

1. Introduction 4

1.1. Background 4
1.2. Problem 1: Insufficient blockchain bandwidth 4
1.3. Problem 2: Insufficient blockchain payload space 6
1.4. Problem 3: Lack of privacy protection 7
1.5. Problem 4: Expensive and delayed transaction costs 7
1.6. Problem 5: Limited application scenarios 8
1.7. Summary of problems 9

2. Core technology 10

3. Application scenarios 17

Share and rights transactions 17
Asset transactions 17
Bank supervision/legal compliance 17
Blockchain finance 18
Social governance 18
High-speed cryptocurrency payment systems or exchanges 18

4. Development of BOLT and cooperative projects 19

5. Conclusion 21

Appendix A: Reasons for non-scalability of public blockchain 22

Appendix B: Privacy protection in BOLT's ledger booster 24

Appendix C: BOLT crypto-payment booster's contract and operating protocol 25

BOLT

A diversified industry application that overcomes restrictions in blockchain processing and realizes truly decentralized transactions

Ever since Bitcoin implemented a cryptocurrency with decentralized controls in 2009, people have been waiting eagerly for the transformation of social values that decentralization might bring. The industry has been in a rush to explore how the blockchain technology that Bitcoin is based on might be used to realize even greater business efficiency.

BOLT proposes an all-new type of distributed auditing as well as a scaling solution that overcomes bottlenecks encountered thus far by blockchain technology and its implementation in commercial applications. Its advantages include:

- Fast transaction speed: The main and sidechains can operate together to achieve over 10 million transactions per second (TPS).
- Transaction privacy assured: BOLT provides consumer and digital asset provider privacy protection during transaction completion.
- Convergence with centralized business scenarios: BOLT integrates decentralized information equivalence with modern agent models of business.

BOLT has surpassed the blockchain's limits for processing speed and number of transactions. It has solved both the lack of bandwith on the public chain and the lack of global consensus in private chains, establishing a trustworthy decentralized architecture with no speed limits.

BOLT can be applied in public or private blockchain systems with smart contract mechanisms. It has solved the problems impeding traditional blockchain architecture:

- Insufficient blockchain bandwith
- Insufficient blockchain payload space
- High and non-immediate transaction costs
- Lack of privacy protection
- Inability to integrate with existing centralized applications, limited industrial application scenarios

Finally, this white paper gives examples of potential value and business opportunities from the use of blockchain in financial, legal, medical, supply chain scenarios, and exchange. For share trading and asset transactions, BOLT can not only realize even faster transactions but also superior privacy protection, thus truly offering a transaction ecosystem with mutual trust.

BOLT is the best way to bridge centralized systems with the decentralized blockchain.

1. Introduction

This white paper begins by introducing the problems and bottlenecks currently facing the blockchain, then explains the origin, planning, structure, and ecology of BOLT. BOLT supports a decentralized system structure with no limits on speed and number of transactions. It has adopted an off-chain model for transaction processing with a concurrently running main chain and booster, using new decentralized auditing technology to solve the problems of inadequate transaction bandwidth, excessive data volumes, and the lack of privacy protection in conventional blockchains. BOLT creates a completely new and trustworthy fully-featured blockchain structure.

1.1. Background

Bitcoin is a cryptocurrency with decentralized controls that was created in 2009. The blockchain technology it is based on is now widely accepted and used in many industries. Apart from becoming an internationally recognized currency, people are now hoping that this shared value system will enable the development of decentralized applications (Dapp) in each industry based on blockchain technology.

In addition to its use in cryptocurrency transactions, the advantages of blockchains, such as decentralized information verification and resistance to tampering, have been noted in various fields. Key applications include value registry[1], value web[2], and value ecosystem[3]. Industries with related applications include logistics, financial systems, medical records, the collection and verification of data in the Internet of Things (IoT), supply chain management, stocks or options trading, social networking software, electronic patient records, micropayment/mobile payment systems, asset transactions, and distribution of digital products. People are hoping that blockchains will be able to play the role of a trusted machine in the operation of such systems. Keeping a detailed record of related information and solving information asymmetry problems will enable a trusted record to be established. In the use scenarios mentioned above, large amounts of information will need to be recorded on the blockchain.

However, the current state of development in blockchain technology has encountered a bottleneck that, if not solved, will make it difficult to realize the various application potentials of the blockchain. The following will describe and explain these challenges.

1.2. Problem 1: Insufficient blockchain bandwidth

The blockchain's decentralized operation is dependent on Internet users worldwide for its maintenance and use. Any user can therefore use block transactions to exchange cryptocurrency,

[1] Application of distributed ledger to Proof of Existence and Possession (PoEaP).

[2] Value registry, smart contract, domestic payment, international payment, trade finance and capital market.

[3]Applications for non-financial services such as public ledgers where it can provide many kinds of commercial applications.

write smart contracts, or record information. Bitcoin and Ethereum are, however, limited to no more than 7 and 25 TPS[4], respectively. If no technology available can place large numbers of transactions on the blockchain, it will simply not be practical to solve information asymmetry by storing transactions on blockchains.

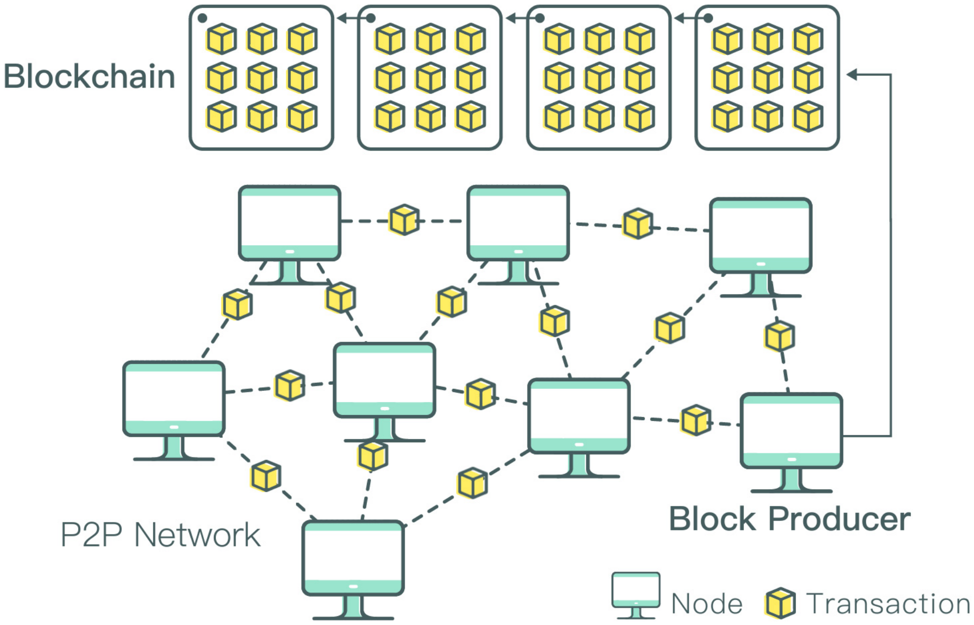


Figure 1: Public Blockchain Consensus Algorithm

As shown in Figure 1, the decentralized operation of the blockchain basically uses a consensus protocol such as Proof of Work (PoW) or Proof of Stake (PoS) to obtain or select a block producer[5] from participating nodes. The block producer then collects transactions through a Peer-to-Peer (P2P[6]) network and records these transactions in a single block within the blockchain using electronic signatures and a hash function[7]. All nodes in the public blockchain participating in the consensus protocol must continuously update any changes to the data in the blockchain as well as obtain transactions that ordinary users want to place on the blockchain. Large amounts of information must therefore be exchanged[8] over P2P networks, thus making it impossible to increase transaction

[4] Yo Banjo, "Ethereum won't scale like you've been told," https://medium.com/@yobanjo/ethereum-wont-scale-like-you-ve-been-told-cae445bef539.

[5] It is also referred to as a “miner.”

[6] A Peer-to-Peer (P2P) network is an Internet networking system with no central servers, and depends on the exchange of information between peers. It reduces the number of unnecessary nodes in network transmissions and thereby lowers the risk of data loss. Unlike centralized networks with a central server, in a P2P network every peer is a node that also functions as a server. One single node cannot find another node directly; instead, all data must be exchanged through peers.

[7] The hash function is a method for generating a small digital “fingerprint” for any data type. A hash function compresses the message or information, reducing its size and fixing its format. The information is completely re-combined by the function to create a new fingerprint called a "hash value."

[8] Propagating.

bandwidth. Public blockchains are generally considered to have global consensus[9]. A detailed explanation is given in Appendix A.

A "private blockchain[10]" or "consortium blockchain" are methods that attempts to solve the problem of insufficient transaction bandwidth. The number of participating blockchain nodes is limited to facilitate rapid propagation and the use of special consensus protocols (e.g., all types of PoS, BFT, and PoA[11]), which contribute to speeding up the selection of block producers. There is obviously a big credibility gap between private blockchains and public blockchains. The core philosophy of a decentralized system is to reduce the access threshold and remove restrictions on participating nodes so that no monopoly on trust machines can be formed. In a private chain, the smaller number of nodes increases vulnerability to 51% attacks[12] and prevents global consensus.

Some public chain developments intend to increase speed by adopting an architecture with a smaller number of nodes, but that leaves them vulnerable to 51% attacks as well as Distributed Denial-of-Service Attack (DDoS) attacks that result in the stoppage of the entire blockchain.

1.3. Problem 2: Insufficient blockchain payload space

As described in the previous section, in each type of system the public blockchain plays the role of the trust machine. As large amounts of transaction records are pushed onto the blockchain, the amount of data on the blockchain will rapidly increase within a short amount of time. Depending on the consensus model, full nodes of the blockchain must store every block on the blockchain and the transactions they contain. The consensus protocol of Bitcoin, for example, restricts the growth in blockchain capacity to around 70 GB[13] per year. In the absence of such restrictions, the propagation and storage of blocks becomes a major problem. This situation is also known as "blockchain bloat[14]." VISA reported that it generated a total of 92.064 million payment transactions in 2015. If translated into the data structure used for Bitcoin transactions, it would amount to around 2,900 transactions per second and 47TB of storage space. This already far exceeds the hard drive space on an ordinary computer[15].

[9] Bitcoin and Ethereum have between 8,000 – 10,000 participating nodes at any given time, many of which are also mining pool nodes with massive processing power.

[10] Examples includes the Coco architecture proposed by Microsoft and Intel and the hyperledger.

[11] In PoS, representatives are usually chosen to compete for block producer status; since in all types of BFT point to point communication between all nodes is required, it can only have between 20 - 30 nodes; in PoA (Proof of Authority), authority nodes designated in advance are responsible for producing blocks.

[12] A 51% Attack is where control over more than 51% of the nodes gives the controllers the ability to modify blocks or control their production.

[13] Around 300,000 transactions a day with each transaction taking up 700 Bytes. The amount of memory added per year would be 300,000*365*700 Bytes ≒ 70GB.

[14] Some experts warn that Ethereum will soon be hit by this problem （https://read01.com/zh-tw/aKE6A7.html#.WcBzldv3U0o)

[15] https://www.zhihu.com/question/39067000.

1.4. Problem 3: Lack of privacy protection

At the moment, privacy protection in blockchains consists mainly of using a mechanism similar to money-laundering to conceal information about cryptocurrency transactions. There are two main methods: (1) Cryptography accumulator: Used by Zerocoin; (2) CoinJoin: Used by SharedCoins, Dark Wallet, CoinShuffle, the PrivateSend feature of Dash, and JoinMarket. Cryptocurrency transaction information recorded on the blockchain gives no indication of the sender.

The two methods above can only be used for cryptocurrency transactions and so cannot be used for other general transactions or contracts. The popularity of Ethereum's smart contract is due to its ability to handle general transactions or contracts, not just cryptocurrency transactions. These include asset transactions and patent licensing as well as contract, document, and information records. Smart contracts for non-cryptocurrency transactions cannot make use of cryptocurrency's privacy protection technology, thus limiting the system's scope of application.

1.5. Problem 4: Expensive and delayed transaction costs

On the blockchain, each transaction must be chained to the new block by miners. The miners' commission fee therefore makes it difficult to use the blockchain for micropayments (such as buying beverages or taking the bus), which in turn prevents the blockchain from achieving universal use. Miners' packaged blocks need a period of time for confirmation[16] to avoid accidental fork as well, further enhancing the challenges of using cryptocurrency for ordinary payments.

Currently, many companies have been developing cryptocurrency payment systems[17,18,19,20]. Payment via cryptocurrency and tokens will become a financial important operation mode in the future, but existing payment systems and financial models have the following problems:

Centralized deposit payment systems: Cryptocurrency and token payments are first deposited into a centralized account, then, after a period of time, the payment system clears the cryptocurrency and tokens to the recipient's blockchain address from the centralized payment system. This is because since that centralized account is not on the blockchain, it can process information faster. However, since customers' cryptocurrencies are kept in an deposit account controlled by a third party, there are customer concerns about the security and privacy of the system, as well as numerous technical and support issues.

[16] The confirmation time for Bitcoin is at least one hour. For Ethereum, the confirmation time varies from several minutes to many hours.

[17] PayPal：Currently PayPal's system sends the private key of an e-wallet to the recipient. In the wallet is the amount of virtual currency determined in advance as the amount to be paid, thus eliminating the need for the recipient to wait for the transaction to clear the blockchain.

[18] Yamada Denki, Japan's home appliance retail chain, and bitFlyer, a cryptocurrency exchange, launched bitcoin payment services at two stores in Tokyo.

[19] Coinbase：http://blockcast.it/2018/02/11/coinbase-launches-paypal-like-plugging/.

[20] Line：https://www.pixpo.net/fiance/0lHONwv7.html.

Blockchain payment systems: Because all cryptocurrencies and tokens are managed by the participants involved, there are fewer security concerns. However, all transactions and payments must be conducted directly on the blockchain. The speed of payment is limited to that of the blockchain main chain, and the transaction cost is high. This is suitable for a small amount of cases, such as rent payments or loans, but not for most scenarios with micropayment requirements.

A new generation of cryptocurrency trading and payment platforms must now solve the problems of security problems, slow transaction speeds, and high transaction costs. At the same time, the solution must entail the ability for cryptocurrency or token trading to be supervised on the blockchain with a strong global consensus; under such demands, there are many projects that try to solve the problem of transaction costs. The most common solution is to only send the transaction signature to the recipient on transaction channels. Such solutions are convenient for one-on-one payments, such as installment payments, but as the transaction signature is generated when the transaction is sent to the payment network, multi-party transactions are difficult to handle. Channel-type schemes must solve many issues, including centralized hubs' large deposit requirements, networks flooded with channel transactions, and users needing constant online access.

1.6. Problem 5: Limited application scenarios

The decentralization concept has gained acceptance in some circles. The forging and trading of cryptocurrencies, for example, can now be completely implemented using a decentralized model. Nevertheless, human economic activities are influenced by law, habit, legacy systems, and inter-personal relationships. Dispensing completely with centralized operations is impossible. Industries with related applications, as mentioned in Section 1.1, such as: logistics, financial systems, medical records, the collection and verification of data in the Internet of Things (IoT), supply chain management, share or rights transactions, social networking software, electronic patient records, micropayment/mobile payment systems, asset transactions, and distribution of digital products—nearly all of these applications still require a centralized agent or intermediary. If the public blockchain cannot be integrated with industries that have similar centralized operations, the use of the blockchain as the trust machine will be greatly limited.

The following example uses digital product distribution to explain why that is the case. For digital products such as e-books, music, movie rentals, and electronic tickets, the widespread use of the Internet and larger bandwidth has popularized sales over online platforms. To expand their sales channels, the rights-holder will usually commission agents to make sales over the agent's network platform. The agent collects payments from users and maintains a record of accounts. The accounts are then provided to the rights-holder at fixed intervals with details on downloads and corresponding royalties. However, since the accounts are recorded and maintained by the agent, the rights-holder is unable to verify their authenticity. For example, the agent's records may contain accidental omissions or other errors due to bugs in the system. Or the agent may deliberately forge or modify the records to reduce the amount of royalties payable to the rights-holder.

In other words, even if the agent placed the account ledgers on the blockchain, the rights-holder would still be unable to verify their authenticity. In scenarios like that above, the blockchain is unable to play the role of the trust machine. It has been suggested that all related transactions should be conducted through cryptocurrency. This method, however, encounters too many limitations: First, such purchases are often micropayments, but with blockchain's overly high transaction costs, there is not enough transaction bandwidth to handle large volumes of micropayments; second, consumers usually pay by ordinary currency or credit card; third, no space is available for some records that are not related to currency transactions. If there is a way to overcome this limitation while also achieving "information symmetry," the goal of decentralization, it will greatly increase the versatility of the blockchain.

1.7. Summary of problems

Based on the explanations above, we can now give a summary of the current problems to be solved: Blockchain transaction speeds are too slow. How can it handle a large number of transactions in a short period of time? How do we fix the problem of transaction costs being too high and not immediate? How can blockchain bloat be avoided when storing large numbers of transaction records on the blockchain? How can the privacy of involved parties be protected when records for transactions other than cryptocurrencies are written to the blockchain? In the real world, the division of labor in the commercial environment means that intermediaries (or agents) cannot be easily replaced. They may also form a wall between digital asset publishers and consumers. So how can intermediaries be retained while providing transparent, reliable, and verifiable consumption records?

These four problems are tightly interrelated. For example, if blockchain's transaction bandwidth is greatly expanded, then the problem of block bloat seems inevitable. Large numbers of transaction records on the blockchain will lead to excessive data storage. If the privacy of the transaction parties could be protected, third parties will not be able to obtain the transaction details. But then the problem caused by dishonest agents, may become even worse. The BOLT proposed in this white paper will completely solve these problems. Please refer to Section 2.

2. Core technology

In this section, we will first explain the operating model of BOLT, then look at how it solves problems from Section1.7.

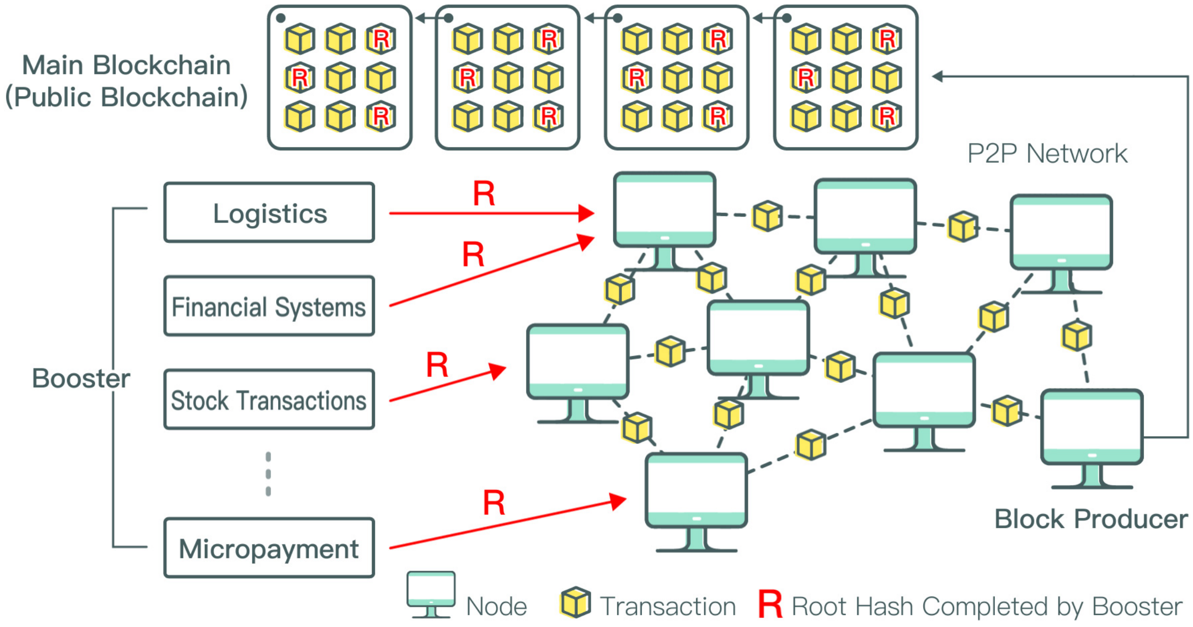


Figure 2: The Operating Model of BOLT

The architecture of BOLT is shown in Figure 2. It is a joint operating model consisting of the main blockchain and several **boosters**. Generally speaking, transactions such as cryptocurrency transactions or individual contract records that do not need to be processed quickly are sent directly into the P2P network and finally linked to the main chain by nodes that have become block producers

High-volume transactions, or those that require centralized matchmaking, however, are first processed on a booster. A hash value is then generated for the transactions, which is then sent to a node in the P2P network and linked to the main chain. The booster runs at a high speed and accumulates a large number of transactions after a certain amount of time. A hash value and corresponding identification code is then generated by the auditing node responsible for the decentralized operation of the booster and then sent to to the main chain.

There are several techniques for running transactions outside of the main chain before adding them to the main chain. After explaining each of these techniques, we will look at how BOLT are different. The first type is “relay-based.” Assets are transferred between the main chain and sidechain before the transaction is conducted on the sidechain[21、22]. The assets are transferred back to the main chain after a certain time. This reduces the number of transactions that take place on the main blockchain. Implementations of this system include BTC-Relay and Rootstock. The problem is how to implement a 2-way peg protocol. For example, the blockchain of Bitcoin is impossible to establish such a relayer.

Another type is “channel-based” and usually referred to as “off-chain.” Lightning-network and Raiden, for example, use off-chain transactions to increase their TPS. In this method there is no need to use nodes to obtain consensus about transactions produced in sidechains. A payment channel is created in advance on the main chain and all the participants to the transactions made over this channel exchange electronically signed information outside of the main chain to indicate that transactions have taken place. A summary of these transactions are then written back to the main chain. However, this approach requires a prepayment in the channel and the Internet connection in real time to avoid receiving transactions from others, conditions that can be difficult to apply.

BOLT uses an agent type (Proxy-Based) system for its boosters. In this type of scenario, the user will commission a platform or agent to assist them in chaining the transaction and submitting non-defective modifications into the consensus system. Thus, the effectiveness of the application is achieved by the underlying data structure. An unlimited number of booster can be generated at any time when needed, making BOLT suitable for solving the problems of real-world scenarios interfacing with the blockchain.

In the features provided by BOLT, we not only increase the bandwidth and solve blockchain bloat and privacy protection issues, but also solve the difficulty of integrating current application systems with decentralized systems. In its operating model, we’ve combined the consistency and global consensus of the main chain with the validity and wide distribution of the booster. The operation of booster avoids having a single point of failure or malicious attack by agent (via the existence of distributed auditing). The operation of booster proposed by BOLT includes the company-patented "Distributed Audit Function[23]”.

Boosters are initiated by industry applications (e.g., transaction agent platforms, professional brokers, investment banks, securities companies, auditors, appraisers, lawyers, toolkit developers). For market and business development, different boosters are managed and operated for individual business types. Boosters must regularly synchronize their information with the main chain to avoid the counterfeiting or tampering of data. The main chain can operate in parallel with multiple boosters to achieve transaction speeds in excess of 10 million TPS.

[21] A SIMPLE EXPLANATION OF BITCOIN “SIDECHAINS” https://gendal.me/2014/10/26/a-simple-explanation-of-bitcoin-sidechains/

[22] How Two New Sidechains Proposals Could Change Bitcoin's DNA https://www.coindesk.com/two-new-sidechains-proposals-change-bitcoins-dna/

[23] Invention name: Distributed Auditing Method, Device and System. This is an international patent owned by the BOLT research team.

The following table compares BOLT with other public and private chains.

	Public Chain	Private Chain	BOLT
Number of nodes	Unlimited (currently about 10k)	Restricted	Unlimited
Consensus	Global	Local	Global
51% Attack Risk	Difficult	Easy	Difficult
Number of transactions per second	7-25	1,000-2,000	> 1,000,000 (main chain + booster)
Blockchain Expansion	Yes	No	No
Privacy	No	Yes	Yes

There are several types of BOLT boosters. The first type of booster processes general transactions for their recording in the blockchain, while the second type processes cash flow transactions with cryptocurrencies or tokens. The third type of booster processes the operation of function executions in smart contract. The following sections will detail each of these in turn.

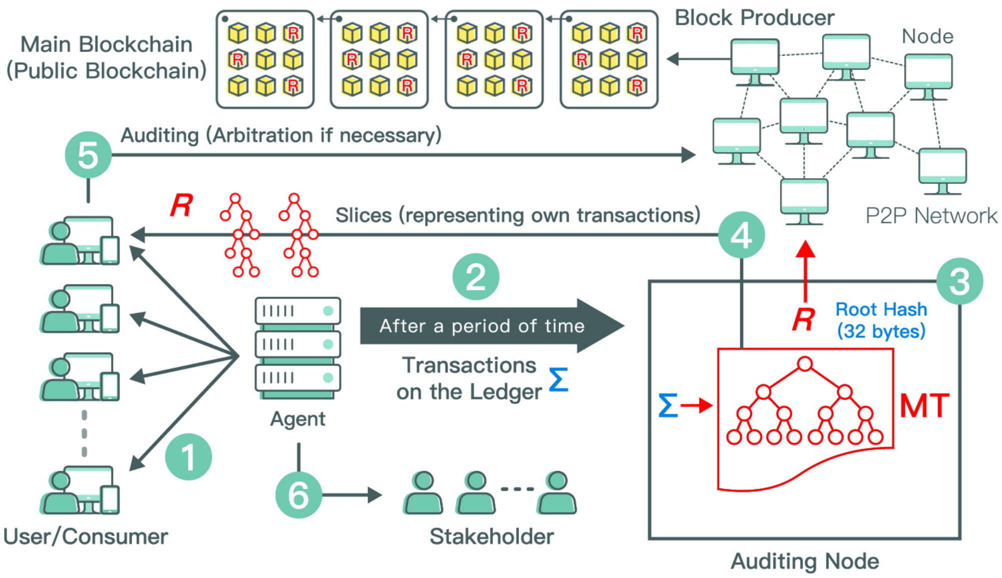


Figure 3: Operation Diagram of BOLT Ledger Booster

2.1.1 BOLT ledger booster[24]

For the operation of the BOLT ledger booster, please refer to Figure 3. A stage in a booster is completed using the following steps:

Step (1)：The agent responsible for initiating the operation of booster starts by conducting a series of transactions with participants (or consumers).

Step (2)：After a certain period of time (End of a stage), the agent sends transactions ∑ to the auditing node.

Step (3)：The audit node uses transactions ∑ to generate an Indexed Merkle Tree (IMT). IMT is also used to generate a root hash value R[25]. R and the corresponding identification tag are sent to the main chain for storing. All participants can use the identification tag from the main chain to obtain R.

Step (4)：Participants are responsible for auditing their own transactions to see if they were correctly placed in the IMT:

- The given root hash value R is used to ask the auditing node to return slices[26] of the participant's own transactions, with each slice representing one such transaction. Since R is anchored to the main chain, an audit of the slice that does not turn up a particular transaction is electronic evidence[27] that the agent did not put their own transaction in the IMT.

Step (5)：Participants send their own audit results to a node within the P2P network:

- If the participant’s audit finds that the agent provided missing or incorrect data, the associated information is signed and then sent to a node for arbitration by block producers. If arbitration finds that the agent made an error, then the participant receives a share of the bond.

Step (6)：The agent pays royalties to the rights-holder. A rights-holder can use R and IMT to verify that a royalty payment is free from error.

An IMT is generated in each stage and is held by the agent. The hash value of the MT must be placed on the main chain; this is implemented as a contract created on the main chain. All of the root hash values generated by each stage are stored in this contract. The distributed ledger fee and bond token are stored and converted using this contract as well.

[24] The SDK had be published in Github. https://github.com/BOLT-Protocol.

[25] Bottom-up recursive hashing of the MT leaf nodes all the way up to root node gives a root hash 32 bytes in length. Addition of the agent’s electronic signature (128 bytes) brings the total length to just 160 bytes.

[26] A slice is a small part of the IMT. An IMT ledger that holds 500,000 transactions takes up at least 300 MB. If other tags are added, then several GB of storage may be needed. A slice contains just 1/100,000th of a full ledger’s data. It can be used to audit an IMT ledger to see if it contains a transaction located within the slice node.

[27] Each transaction can be audited within 1 ms.

The integrity of transactions generated by ledger booster are maintained by all participants. A bond is deposited in advance in the contract for this booster by the agent. Participants and agents both electronically sign their transaction information to realize mutual non-repudiation. In Step (4), multiple participants are involved in auditing the existence and integrity of transactions on this booster. Any omissions or errors found in an agent's transactions are arbitrated by nodes on the main chain. Arbitration is an execution of a specific contract function. If arbitration is passed then the bond is automatically shared among the participants who issued the arbitration; if not, it is refunded to the agent. This boosts the incentive for participants to take part in the audit.

The design of ledger booster is suitable for transaction-based billing. It is not only easy to implement, but also can achieve applications that are not limited by the bandwidth of the main chain, such as copyright registration and privacy protection. See Appendix B for details. However, if you need to use high-speed, low-cost micropayments, the cashflow distributed audit sidechain described in the next section is more suitable to meet the needs of such applications.

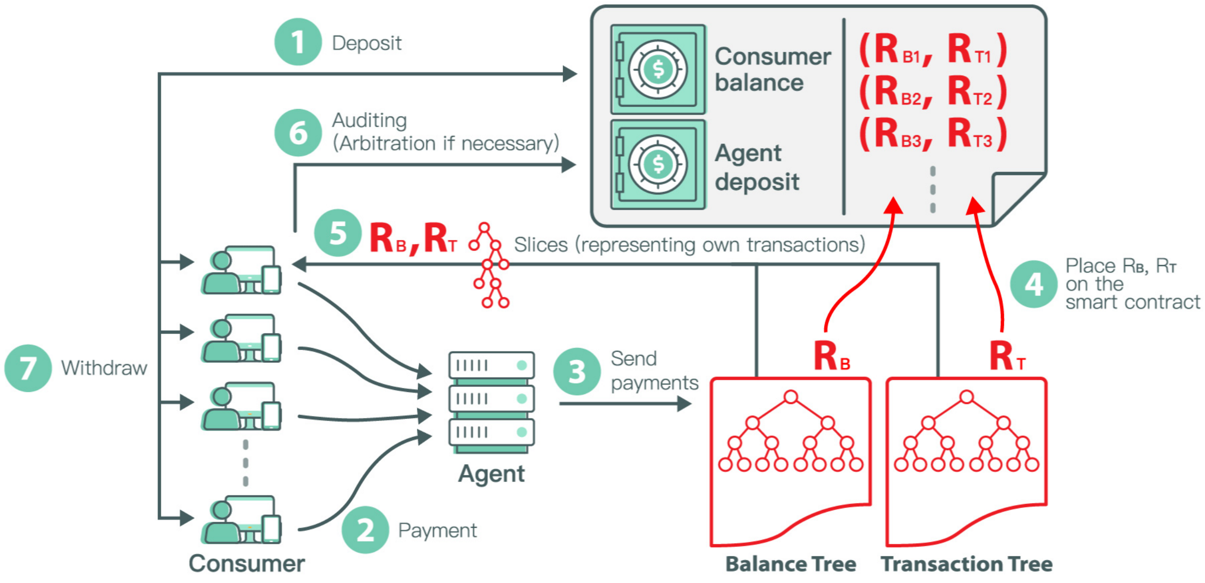


Figure 4: Operation diagram of BOLT crypto-payment booster

2.1.2 BOLT crypto-payment booster

The structure of the crypto-payment booster in BOLT is shown in Figure 4. A contract published to the main chain (referred to as the booster contract) is used to control and record the cash flow exchange of participants in the booster. General cash flow does not need to pass through the main chain, which speeds up the cash flow. A cash flow network can be initiated by a convener who acts as the agent. Each contract corresponds to one booster. A crypto-payment booster can be initiated at any time as necessary. For example, participants can take part in different crypto-payment boosters for an online mall or high-speed cryptocurrency exchange, all based on their requirements.

Users can store their cryptocurrency or tokens into a booster contract and withdraw them at any time. If cryptocurrency or tokens stored in a booster contract enters a booster for trading, then it will be temporarily unavailable for withdrawal. It must be withdrawn from the booster to the booster contract before it can be transferred to another account on main chain block chains.

The agent is responsible for maintaining a record of the consumer's transactions on the booster, the balance tree and receipt tree. The balance tree and transaction tree are all indexed Merkle trees or a data structure that supports the use of slices to verify existence or non-existence of stored records. A balance tree records the balance of cryptocurrencies or tokens transferred by each participant in this booster. The balance is stored in the balance tree using the participant's ID as the index. The balance decreases whenever a participant transfers cryptocurrency or tokens on the booster to other participants in the booster, or transfers cryptocurrency or tokens from the booster to the booster contract. A receipt tree records all transactions and functions similarly to the IMT in the general ledger booster, as shown in Figure 3.

The operation of the crypto-payment booster is similar to the operation of the ledger booster (see Section 2.1.1, Steps (1)-(6) and Figure 5), and is also the agent responsible for launching the operation of booster by beginning a series of transaction activities with participants (or consumers). After the end of the phase, the agent finally puts the two root hash values of balance tree and receipt tree in the contract. Then, all participants are responsible for auditing whether their transactions are correct and placing them in the receipt tree. In addition to this, each participant is ultimately required to audit the their own balance in the balance tree, double-checking the cryptocurrency in the contract and the booster, and the conversion of tokens. The most important thing is to generate cryptographic evidence and send it back to the contract for fraud proofing when it comes to verifying that an agent has an error.

The protocol is patented[28] and the detailed protocol is very complex. Please refer to Appendix C. Each transaction's receipt ($T_{receipt}$) contains a GSN (global sequence number), an integer generated by the agent that increases by 1 after each transaction of a participant is processed. When someone has an error, that number can generate enough cryptographic evidence to send it back for fraud proof.

2.1.3 BOLT contract booster

In addition to general data recording and cryptocurrency transfer, the execution of functions in smart contracts is also an important application in blockchain. Variables and predefined functions are

[28] Invention name: Protocol for Distributed Auditing of Payment Flow. This is an international patent owned by the BOLT research team.

two major parts in a smart contract. An execution of a smart contract always makes state transitions for some of its variables. The functions are executed by miners (block producers) in the main chain, one at a time.

Generally speaking, the BOLT smart contract booster is an extension of its crypto-payment booster (refer to Figure 5). The contract functions and their corresponding validating functions are pre-defined in the main contract. Contract functions are publicly announced. In steps 1 and 2, participants request the agent to execute an contract function by sending a signed request message. The agent executes the function off-chain after receiving this message and then sends back a response message which contains the name of the contract function, attached parameters, and the state transaction caused by function execution. The response message representing a transaction is placed into the transaction tree. States of all variables are stored in state trees. Both transaction and state trees are stored as index Merkle trees.

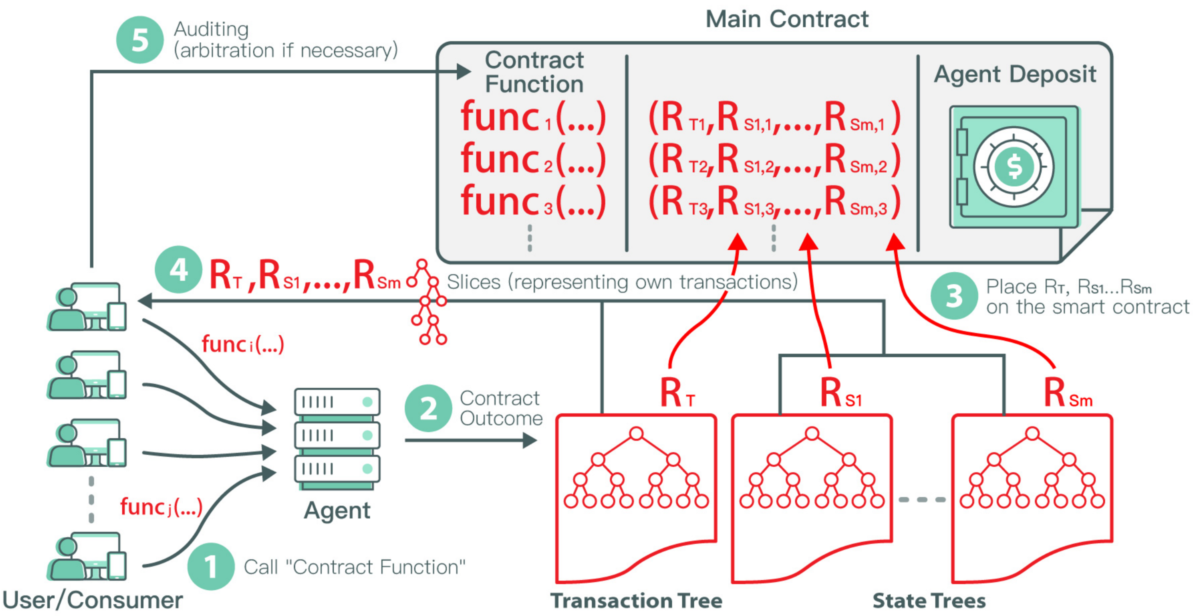


Figure 5: Operation diagram of BOLT smart contract booster

In step 3, after a stage is finished, the agent appends the root hashes of transaction and state trees into the main contract. After that, in step 4, each participant audits to see if their transactions have been properly put into transaction trees. In addition, participants also check if the state transitions caused by function execution are correct. Whenever there is any error, participants can generate a cryptographic proof and send it to the main contract for fraud proof.

3. Application scenarios

The popularity of the blockchain has led to widespread discussion in each industry on the potential value of its applications. The financial field is seeing ongoing developments in transactions and payment, while people in the social field are looking at using blockchains to record activities and build up their reputation; people in the medical field are looking at the advantages of using the blockchain for storing electronic patient records; and in the legal field, blockchain has huge potential in validation, auditing, and smart contract applications.

Share and rights transactions

In the share trading scenario, the need for buyers and sellers to track the transaction of share rights means that transactions will take longer. Companies must expend a great deal of resources on attorneys, auditors, and consultants to review the transactions among investors. With the blockchain acting as the trust mechanism, middlemen such as auditors can be taken out of share transactions. If transactions are linked to the blockchain, buyers can spend less time tracking share rights. However, the massive volume of share transactions is simply too large a load for the transaction bandwidth of current blockchains. BOLT not only offers fast transaction speeds but also emphasizes privacy. Every participant to a transaction can only see the records relevant to them. Benefits include speed and confidentiality. A practical system can be implemented using BOLT ledger booster.

Asset transactions

All assets can be digitized. Once an asset has been digitized, it can be quantified, circulated, bought and sold, and mortgaged to generate huge value. In the future, houses and cars may all become assets on a blockchain. Ownership will be decided using private keys, making the moment of all real estate far easier. The biggest advantage of applying the blockchain to digital assets is that circulation becomes far easier once assets are published to the blockchain.

The BOLT offers many methods for converting digital assets; it also supports transaction applications. During a transaction, all participants receive suitable privacy protection. Non-reputability of the transaction is also guaranteed, allowing participants to trust and rely on one other.

Bank supervision/legal compliance

Banks can use the trust machines of public blockchains to satisfy requirements for internal controls and legal compliance. For example, each transaction or business activity made by a bank employee can be recorded immediately. The transaction serial number and the bank's electronic signature prevent the transaction from being repudiated by the bank. Once all records have been consolidated they are placed on the blockchain. The data is transparent and cannot be tampered with, making post auditing unnecessary.

In comparison, existing internal controls and legal compliance depend on third-party audits. In addition, preliminary internal training, laws and regulations, as well as post audits/verification are required. With BOLT's distributed auditing technology, a bank and its employees' business records are completely transparent. Bank employees can audit the accuracy of their own activity records or of a particular action raised by the bank. The forging of records is also impossible in this use scenario.

Blockchain finance

Using VISA as an example, VISA's global average number of transactions per second is about 2000 (up to 58,000 per second). Combining the VISA trading system with the blockchain's trust mechanism using the traditional public blockchain is very difficult, as you must fix VISA's huge transaction volume in the block. If we use BOLT booster, we can generate tens of thousands of transaction records per second on the sidechain and complete decentralized audits to solve the problem of transaction bandwidth. BOLT provides fast transaction services, and maintains transaction integrity through decentralized audits. This will be the basis for future blockchain applications in finance.

Social governance

In the traditional field, identity authentication, notarization, judicial arbitration, voting, and loan systems all use centralized servers to access data. There is the potential for this data to be counterfeited. One good way to solve this type of problem is to use a blockchain. Blockchains are open, transparent, non-counterfeit, and low in cost. It can therefore be predicted that in the future, such applications will use blockchain technology to solve counterfeiting issues.

BOLT complements blockchain privacy and speed deficiencies, and can accept a large amount of data at the same time. This technology can be used in electronic voting, allowing citizens to confirm their identities, vote quickly, keep the advantages of the blockchain, and make up for the shortages of the blockchain system.

High-speed cryptocurrency payment systems or exchanges

Existing cryptocurrency payment systems or exchanges are faced with the issue of bandwidth for main chain transactions. With the enthusiasm and value of cryptocurrency transactions, the cost of implementing cryptocurrency payments and transactions directly in the main chain has risen, with too many executions often causing congestion. Using BOLT crypto-payment booster to implement similar systems will enable cryptocurrency micropayments and high-speed exchanges.

4. Development of BOLT and cooperative projects

The BOLT team has acquired several international patents and is currently applying for several national patents:

(1) Distributed Auditing Method, Device, and System

(2) Method for Auditing Cloud Access in Real Time

(3) Protocol for Distributed Auditing of Payment Flow

BOLT boosters have completed Proof of Concept programming and operational testing. At present, we have negotiated cooperative projects with several companies. For confidentiality reasons, company names will be represented by initials until products are ready to be officially announced.

BOLT's large billing functions assist Japanese-listed e-commerce companies in building a blockchain time-based pricing model. Time-based entertainment e-commerce companies have in the past been difficult to achieve because of unequal information among agents, users, and content providers. The actual cost of keeping bills on the public chain is very high, and BOLT ledger booster can achieve large-scale and fast recording on the blockchain within the time needed for auditing, making information symmetry possible. At present, it has officially signed up with the company. At the same time, the TIDEiSun Group also uses BOLT's booster technology to create various applications such as blockchain advertising and blockchain copyright registration. Related businesses are TideFinTech and TideXMedia.

The other cooperative project is a well-known international securities and futures exchange. Its goal is to record all kinds of transactions such as equity, stocks, and futures on the blockchain, shorten the tracking time of securities dealers and customers, and increase the accuracy and credibility of related queries. However, the current blockchain does not have enough transaction bandwidth to record the huge number of transactions. BOLT's booster technology, in addition to providing a large number of transaction records, pays more attention to privacy. Each trader can only see their own relevant records according to their own authority, quickly and privately. At present, this project has almost completed PoC, and is about to enter product implementation and release.

Company K designs a mobile app with millions of users around the world, and has recently completed round A financing. They currently plan to implement a payment system for copyright registration and use on the Ethereum backbone, and has signed with the BOLT team to use the BOLT ledger booster to provide their clients with a digital media rights registration system on the Ethereum backbone and the BOLT cashflow booster to implement its ERC20 token instant copyright usage payment system.

Company C is an IT company that serves a government Ministry of Education and has successfully implemented many software contracts for the ministry. BOLT will assist them in providing the Ministry of Education a blockchain qualification registration system on the public chain (Ethereum). Company C will soon sign up to use BOLT's ledger booster to complete this project and kickstart its future blockchain-related projects.

Company H plans to enter the fund-related asset exchange, investment and investor matching business, and has successfully passed the Regulatory Sandbox to become a financial innovator in its country. This year, it plans to record its transactions on the public chain for investors to inspect in order to promote investor trust in its stewardship of their assets. It will implement BOLT's ledger booster in its first step for its blockchain system and BOLT's cashflow booster in its second step to establish a corresponding trading system using ERC20 tokens for exchange.

Company G is a pioneer in the business point integration field, and is currently seeking to use ERC20 tokens to establish a decentralized business point platform. Unfortunately, the Ethereum main chain has insufficient bandwith and long, delayed confirmation times. BOLT's cashflow booster will assist them in designing and implementing this platform.

Company J is a Japanese startup organized by several Japanese game and technology companies. Their product is a contract game on Ethereum, with a completed whitepaper, that is currently in its funding period. Because of Ethereum's insufficient bandwith, the company has recently been investigating various solutions. After being introduced to BOLT's technology and functions, the company has decided to pay for the use of all three BOLT boosters.

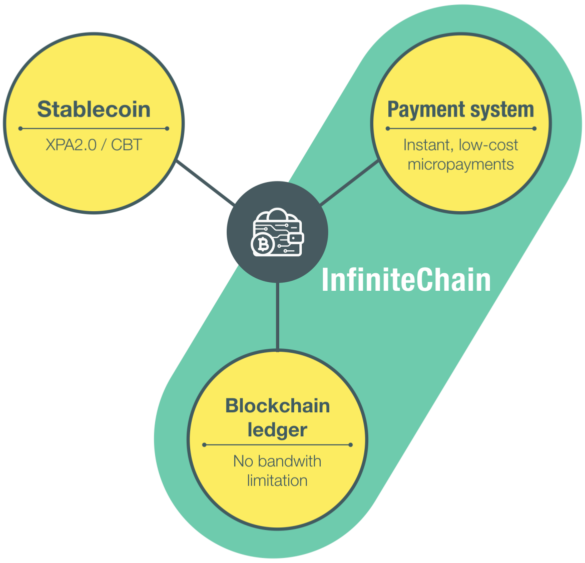
Company I's main product is its high-end encrypted communication app. Its app uses Bluetooth communication to carry out electronic signature and encryption services. Currently, its customers are seeking a method to integrate blockchain payment functions into its products. In addition to the main chain transfer functions expected by general blockchain wallets, the company is also expected to provide high-speed, low-cost payment methods. It is in talks to use BOLT's cashflow booster to successfully design and implement this function.

This is only a representative sample of the companies that are in agreement with BOLT, with further companies not listed for reasons of limited space.

5. Conclusion

The development of decentralized application systems based on blockchain technology is beginning to enter our society, but experts have pointed out the basic technology bottleneck. Of the technical problems, consensus issues, and economic models raised in Section 1.8, if any one problem cannot be solved, we will lose the dream of using a blockchain as a trust machine and the blockchain will only become a cryptocurrency minting and trading platform, or can only be used by a few application scenarios.

At present, TideiSun Group is actively developing TideFinTech, which plans a complete blockchain and cryptocurrency ecosystem. The key technologies include stablecoin on the blockchain, a real-time, low-cost micropayment system, blockchain record not limited by bandwith, and the Wallet app for integration services, all of which are shown below. BOLT realizes the basic technology of an instant, low-cost micropayment system and accountless system within limited bandwidth.



The BOLT development team confirms that the decentralized system breaks current inequality and allows participants to participate in the activities with global peace of mind, and proposes thorough solutions to the blockchain's problems with practical technologies. Related technologies have been implemented in an embryonic form to assess their effectiveness and feasibility. The development of the next decentralized application system and blockchain will certainly be led the contribution of BOLT's development team.

Appendix A: Reasons for non-scalability of public blockchain

The transaction speed of a blockchain is generally expressed as the number of transactions that can be stored to the block per second (Transactions per Second, or TPS). This can also be referred to as the "transaction bandwidth." It is basically the average number of blocks generated by the blockchain per second multiplied by the average number of transactions encapsulated per block. The formula is:

Transactions per Second = (average number of blocks generated per second) × (average number of transactions per block)

Bitcoin uses PoW to randomly select block producers, so the average number of blocks generated per second on the blockchain is quite small. In practice, a block is on average generated only once every 10 minutes. Bitcoin's blockchain is therefore throttled to around 7 TPS. Most of the later blockchain developments trying to overcome this speed constraint adopted a PoS consensus protocol to determine block producers. Basically, a node that wants to become the next producer must compare its stake against other competitors. According to the PoS consensus protocol, the node with the highest stake becomes the next block producer. Unlike PoW, there is no need to compete on the level of computing power, and thus PoS is faster. As shown in Figure 1, a node that wants to become the next block producer receives disseminated transaction data from the P2P network. Once it is selected by the consensus protocol, a block is generated and disseminated to other nodes over the P2P network.

For the PoS consensus protocol to work, all competitors must know the stake of all other competitors. These competitors are scattered all around the world and communicate with each other through the network. Even if they know who the competitors are, they usually have to query the stake of other competitors from the previous blocks on the blockchain. It usually takes several seconds to determine who the next block producer will be. In other words, the average number of blocks produced per second on the blockchain is less than 1. The average number of transactions encapsulated per block is also limited by the node's network bandwidth and the data transmission speed of the P2Pnetwork.

In addition, the block producer must verify[29] all of the transactions to be encapsulated in the block , which also takes time. Once a block producer has been decided through PoS consensus, the producer is usually given a time limit for generating a block, usually this amounts to several seconds. Ethereum is currently limited to around 25 TPS. Experts have already voiced their doubts over Ethereum's push to have more than 100 Dapps running online at full speed. Their reasoning is that

[29] To prevent double spending or illegal transaction.

they think this will overload the Ethereum blockchain[30] . From this we can conclude that if all web users are allowed to participate in a “public blockchain,” speed constraint will remain a problem, since all participating net users must exchange information via P2P. However, the large number of participants and the open nature of public chains means that it is the most trusted.

[30] Yo Banjo, "How Etheroll and other Dapps will kill Ethereum," https://medium.com/@yobanjo/how-etheroll-and-other-dapps-will-kill-ethereum-e973d8e1c465.

Appendix B: Privacy protection in BOLT's ledger booster

The privacy of digital asset providers is protected in two ways. The first is that all transactions stored in the Indexed Merkle Tree (IMT) are encrypted using the public key of the digital asset provider. When the consumer audits one of his or her transactions, they can use the public key of the digital asset provider to encrypt the transaction data, and then compare the encrypted data against the data stored in the IMT. When the digital asset provider wishes to conduct an audit, the entire IMT ledger can be presented to the digital asset provider. But the digital asset provider can only see and audit data that can be decrypted with their private key, so the privacy of the other data is protected. This method offers a sound security mechanism, but each asymmetric decryption operation is time consuming (approximately 22 ms). The digital asset provider must attempt to decrypt each transaction stored in the IMT; only those that can be decrypted are transactions related to that digital asset provider[31]. If the IMT contains a large number of transactions this becomes very time-consuming. For example, if there are 100,000 transactions in the IMT, the audit will take more than 150 seconds to complete[32]. If there are 1,000,000 transactions, it will take more than one hour. It may then be a bad idea for the digital asset provider to use a device with low processing power, like a mobile phone, to conduct audits. Please note that this is not a problem for the consumer, since the consumer only audits the small number of transactions related to them.

The other way is to generate an indexed Merkle tree for each digital asset provider (referred to as the Digital Asset Provider Indexed Merkle Tree, or SubMT for short). All of a digital asset provider's transactions are stored in the SubMT without encryption. The root hash of all SubMTs are then used to generate another MT (referred to as the Main Indexed Merkle Tree, or MainMT for short). The root hash of the MainMT is announced on the blockchain by the validator node. When consumers verify the integrity or existence of one of their transactions: (1) The validator node first presents a slice from the SubMT of relevant digital asset providers to the consumers for verification; (2) The consumer then verifies whether the root hash for this SubMT exists in the MainMT.

When the digital asset provider wishes to audit his own transactions[33], the validator node presents the SubMT of the digital asset provider and the MainMT. The digital asset provider then confirms that his or her SubMT appears in the MainMT and is not duplicated. Then, all the digital asset provider has to do is check all of the transactions in his or her SubMT. Since the SubMT contains only his or her own transactions, the transaction details of other digital asset providers will not be visible. Privacy is therefore protected. Auditing only your own transactions means that even devices with low processing power, such as mobile phones, can be used for auditing with no problem.

[31] Transactions that cannot be decrypted are those related to other digital asset providers.

[32] This processing time does not include the traversal time for IMT.

[33] Open source program for announcement.

Appendix C: BOLT crypto-payment booster's contract and operating protocol

The structure of BOLT crypto-payment booster is as shown in Figure 6. A contract is used to control and record the cash flow exchanges of participants. General cash flow does not need to pass through the main chain, which speeds up cash flow. A cash flow network can be initiated by a convener acting as the agent. Each contract corresponds to one sidechain. BOLT crypto-payment booster can be initiated at any time as necessary. For example, participants can take part in different sidechains for an online mall or cryptocurrency exchange, all based on their requirements.

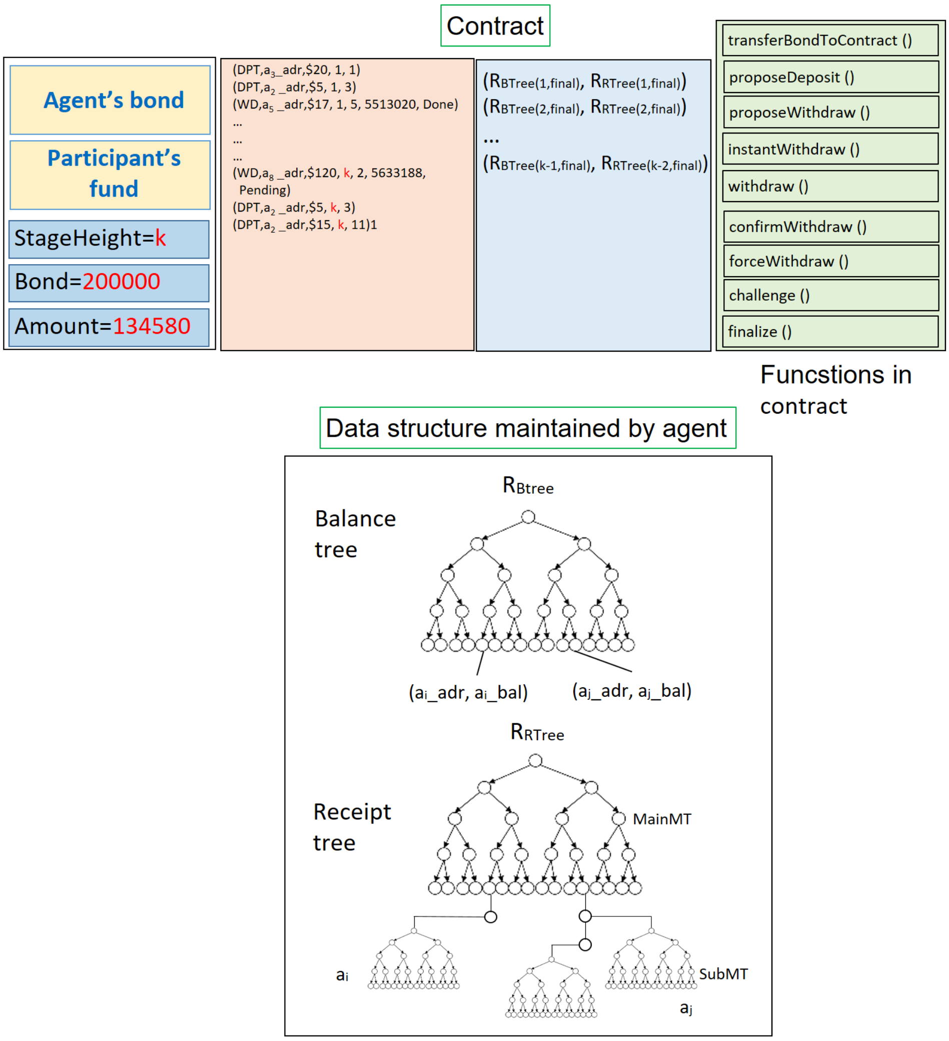


Figure 6: Architecture of BOLT crypto-payment booster

Users can store their cryptocurrency or token[34] into a sidechain contract and withdraw them at any time. If cryptocurrency or tokens stored in a contract enters a sidechain for trading, then it will be

[34] Tokens are cryptocurrencies in the main chain.

temporarily unavailable for withdrawal. It must be withdrawn from the sidechain to the contract before it can be transferred to another account on the main block chain.

Refer to Figure 6. The “Agent’s bond” and “participant’s fund” are tokens stored in the contract. Before starting an operating of an BOLT crypto-payment booster, the agent should execute the function transferBondToContact() to store enough bond, i.e., agent’s bond. Tokens which will be exchanged between participants are first stored in participant’s fund. “Bond” and “Amount” are two contact variables which represents the total agent’s bond and participant’s fund. The operating of an BOLT crypto-payment booster is consisted of a serial of stages. In each end of a stage, clearing of cash flow should be performed to make sure the agent works correctly. The contract's "StageHeight" variable indicates the order number of the current stage.

The agent is responsible for maintaining a record of the consumer's transactions on the sidechain, the balance tree and receipt tree. The balance tree and receipt tree are all indexed Merkle trees or a data structure that supports the use of slices to verify existence or non-existence of stored records.

- The balance tree stores some key-value pairs, a pair is (a_i_adr, a_i_bal), which a_i_adr is the sidechain participant a_i's address; a_i_bal is the balance of tokens deposited into the sidechain by a_i. This is stored in the balance tree using a_i_adr as the index. The balance decreases whenever the participant transfers tokens within the sidechain to other participants in the sidechain, or transfers tokens from the sidechain to the contract.
- The receipt tree holds a record of transactions. The data structure is similar to the description given in Appendix B, and is as shown in Figure 6. MainMT is an indexed Merkle tree where the sidechain participant address is used as the index. Under that, a hash value is used to link to other indexed Merkle trees, referred to as SubMT. Each SubMT holds transactions related to a certain user. Basically, all transaction records for incoming and outgoing tokens as well as transfers from other participants are stored in this SubMT.

Participants can engage in the following four types of transactions:

- Remittance transaction: Remittance transaction: Transfer your own funds recorded in the balance tree of the sidechain to the funds of another user recorded on the balance tree.
- Deposit transaction: Transfer funds from your own contract to the balance tree of the sidechain.
- Instant withdrawal transaction: Transfer your own small-sized funds recorded in the balance tree of the address in the main chain instantly.
- Withdrawal transaction: Transfer your own funds recorded in the balance tree of the address in the main chain.

The operating protocols for the each of the four transaction types will be discussed below.

Remittance transaction protocol (a_i wants to transfer X Token units in the sidechain to a_j)

Step 1: a_i sends a transaction request T_{rmit}=((Remittance, LSN, a_i_adr, a_j_adr, X, SH), $SIG_{PK(a_i)}$) to the agent. LSN（Local sequence number）is a random number generated by a_i that will not repeat. a_i_adr is the address of a_i. X is the token unites to be remitted. SH is the current stage height. $SIG_{PK(a_i)}$ is a digital signature which signed T_{rmit} by a_i.

Step 2: Let Sender_balance = a_i_bal-X and Receiver_balance = a_j_bal+X。The agent goes to the balance tree and changes (a_i_adr, a_i_bal) to (a_i_adr, Sender_balance), and (a_j_adr, a_j_bal) to (a_j_adr, Receiver_balalance).

Step 3: The agent sends $T_{receipt}$ = ((T_{rmit}, Sender_balance, Receiver_balance, GSN), $SIG_{PK(Agent)}$) to a_i and places it in the receipt tree. The GSN (Global Sequence Number) is an agent-generated integer that starts from 0 and increases by 1 each time a transaction is processed. $SIG_{PK(Agent)}$ is a digital signature which signed $T_{receipt}$ by the agent.

Deposit transaction protocol (a_i wants to transfer X Token units to the balance tree in the sidechain)

Step 1: Participant a_i invokes the contract function proposeDeposit(DPT, a_i_adr, X, SH, LSN). A log = [DPT, a_i_adr, X, SH, LSN] will be recorded in the contract. DPT means it is a deposit log. X is the token unites to be deposited. SH is the current stage height.

Step 2: The contract triggers a proposeDeposit event and the log produced in previous step will be sent to the agent.

Step3: Let Balance = a_i_bal+X. The agent goes to the balance tree and changes (a_i_adr, a_i_bal) to (a_i_adr, Balance).

Step 4: The agent sends $T_{receipt}$ = ((DPT, LSN, SH, a_i_adr, Balance, GSN), $SIG_{PK(Agent)}$) to a_i and places it in the receipt tree.

Instant Withdraw transaction protocol (a_i wants to move X Token units from the sidechain balance tree to his address in main chain via the agent)

Step 1: a_i sents transaction request $T_{InstantWithdraw}$=((LSN, a_i_adr, X, SH), $SIG_{PK(a_i)}$) to the agent.

Step 2: Let Balance = a_i_bal-X. The agent goes to the balance tree and changes (a_i_adr, a_i_bal) to (a_i_adr, Balance).

Step 3: The agent sends $T_{receipt}$ = (($T_{InstantWithdraw}$, Balance, GSN), $SIG_{PK(Agent)}$) and places it in the receipt tree.

Step 4: a_i invokes the contract function instantWithdraw($T_{receipt}$). A log [WD, a_i_adr, X, SH, LSN] will be recorded in the contract and X tokens is transfer to a_i_adr in the main chain.

Withdraw transaction protocol （(a$_i$ wants to move X Token units from the sidechain balance tree to his address in main chain by invoking a function in the contract)

Step 1: a$_i$ invokes a contract function proposeWithdraw(WD, a$_i$_adr, X, SH, LSN). A log = [WD, a$_i$_adr, X, SH, LSN, BKH, pending] will be recorded in the contract. “pending” means the withdraw request has not be processed by the agent. If the request does not processed by a specified period, a$_i$ can invoke forceWithdraw(log) to transfer X tokens to its address in the main chain.

Step 2: The contract triggers a proposeWithdraw event and the log produced in previous step will be sent to the agent.

Step 3: Let Balance = a$_i$_bal-X. The agent goes to the balance tree and changes (ai_adr, ai_bal) to (ai_adr, Balance).

Step 4: Let $T_{receipt}$ = ((WD, LSN, SH, a$_i$_adr, Balance, GSN), $SIG_{PK(Agent)}$)). The agent invokes confirmWithdraw($T_{receipt}$). Log generated in step 1 is changed to[WD, a$_i$_adr, X, SH, LSN, BKH, Granted]. $T_{receipt}$ is sent to a$_i$ and placed in the receipt tree.

Step 5: a$_i$ can invoke a contract functionwithdraw([WD, a$_i$_adr, X, SH, LSN, BKH]). X tokens is transfered to a$_i$_adr in the main chain and log is step4 is changed to [WD, a$_i$_adr, X, SH, LSN, BKH, Done].

Sidechain operations are divided into many stages. At the start of the k-th stage, the root hashes of the balance tree and receipt tree are assumed to be $R_{BT(k,init)}$及$R_{RTree(k,init)}$[35]. An agent processing N transactions from different participants during a stage will change the root hash value of the balance tree and transaction tree with each processed transaction. Assuming that the order of changes is as follows:

$$R_{BTree(k,init)} \rightarrow R_{BTree(k,1)} \rightarrow R_{BTree(k,2)} \rightarrow R_{BTree(k,3)} \rightarrow \ldots \rightarrow R_{BTree(k,final)}$$

The root hash of the transaction tree will change after each process, so we can assume that it eventually becomes $R_{RTree(k,final)}$. At the end of the k-th stage, the agent calls Finalize(). The function sends $R_{BTree(k,final)}$ and $R_{RTree(k,final)}$ back to the contract. The StageHeight is also incremented by 1.

Agent Operated Auditing and Cryptographic Evidence

The agent follows the protocols for calling functions in the contract and carrying out the remittance, deposit, instant withdraw, and withdraw transaction protocols. The sidechain can then process large amounts of cash flow transactions between participants without consuming any of the main chain's transaction bandwidth. The agent may, however, experience errors or malicious transferring of the tokens placed in the contract by participants to other parties for their own benefit. The fraud proof mechanism for the sidechain is described below.

As a prevention measure from agents cheating for their own gain, the agent must first deposit a bond in the contract. Audits can be performed by any participant, and if a mistake by the agent is discovered during an audit, the cryptographic evidence they possess can be used to call the

[35] A receipt tree contains no $T_{receipt}$.

Fraud_proof() function in the contract to obtain the bond placed by the agent.

As the distributed ledger sidechain in Figure 3 of the white paper shows, any participant can perform distributed auditing using the $R_{RTree(final)}$ placed in the contract:

- If a participant finds that his/her own transaction was not placed in the receipt tree, then the cryptographic evidence they are holding can be used to call the challenge() function in the contract to obtain the bond placed by the agent. Such a setup prevents the agent from not adding funds transferred by a participant to the receiver's key-value pair in the balance tree.
- Each participant can obtain the transactions related to them using $R_{RTree(k,final)}$. Deposit transactions, withdraw transactions and remittance transactions from other people can be found in the participant's own SubMT. Remittance transactions to other people's accounts can be found in the Receiver's SubMT. As every $T_{receipt}$ in the transaction records has a GSN, a participant can sort his/her own transactions by their GSN. Now, assume that a participant can find n related transactions in the receipt tree.

  $$T_{receipt(1)} \rightarrow T_{receipt(2)} \rightarrow T_{receipt(3)} \rightarrow T_{receipt(4)} \rightarrow T_{receipt(5)} \rightarrow \ldots \rightarrow T_{receipt(n)}$$

  First, the participant uses the root hash of the balance tree from the previous stage to extract the slice for his/her own key-pair and establish the balance in the sidechain at the start of this stage. Next, check the balance in $T_{receipt}$ after each transaction. The balance in $T_{receipt(n)}$ should be identical to the participant's balance in $R_{BTree(k,final)}$. If there is an error in $T_{receipt(j)}$ then the two neighboring transactions $T_{receipt(j-1)}$ and $T_{receipt(j)}$ are used as the cryptographic evidence to call the challenge() function in the contract and obtain the agent’s bond.
- If a participant executing withdraw() finds that the participant’s fund is insufficient, the agent must have conspired with other participants or a mistake was not caught during the auditing process. withdraw() will then transfer cryptocurrency or tokens from the agent bond to account address specified by the participant. A sufficient reserve must exist in the agent bond before the next operation of the sidechain can be initiated.

We briefly introduce the operation of BOLT crypto-payment booster in this appendix. For details, refer to the white paper of BOLT crypto-payment booster.