BitXHub WHITEPAPER

Inter-Blockchain Technology Platform

V 1.0.0

Hangzhou Qulian Technology Ltd.

October, 2019

Abstract

Inter-blockchain is a new technology which aims at connecting various blockchains. Under the dual requirements of business and technology, interchain technology has become increasingly necessary and inevitable. BitXHub designs a unified interchain messaging protocol IBTP to process, route and validate interchain transactions conveniently. It also devises a pluggable and effective validation engine for interchain transactions. In general, BitXHub devotes to constructing a scalable and robust multi-layer IBTP-based reference implementation platform.

About Details: https://github.com/meshplus/bitxhub

Contributors

Consultant：

Weiwei Qiu

Author：

Caichao Xu, Xiaoyi Wang, Liwei Xia, Yongxing Tao, Yang Yan

Editor:

Shutian Bao

Contents

1 Introduction

1.1 Background

Nowadays blockchain platforms are blooming, but most of the mainstream blockchain systems are still independent and vertical closed. In most business scenarios which are increasingly complex today, they lack a unified interconnection mechanism between chains, which hugely limits the healthy development of blockchain technology and its application ecology. Wherefore, interchain[1] requirements arise.

Inter-blockchain refers to the interoperation of the ledgers by connecting relatively independent blockchains. Interoperation can be divided into asset exchange and information exchange depending on their contents. In terms of asset exchange, some blockchains are still in isolation , the transfers of assets are mainly done by centralized exchanges, which is neither safe nor transparent. In contrast, the information exchange is more complicated due to the data synchronization and interchain calls. At present, the barriers between blockchain applications are extremely high, and the on-chain information cannot be effectively shared.

On the other hand, the single-chain architecture has problems such as insufficient performance and capacity. Due to the restriction of consensus's speed, the execution performance of nodes cannot be linearly extended, which limits the applications of high-throughput and low-latency scenarios. In addition, as the running time increases, the blockchain's storage capacity will also gradually increase, and this data growth rate will even exceed the capacity ceiling of single-chain storage media.

Under the dual constraints of business and technology, the interoperability has received more and more attention, it is a trend that the interchain technology has become more and more necessary and inevitable. As a bridge connecting various blockchains, it is

mainly used to realize asset exchange, information share and service complementation. The rigorous description, formal implementation and common application of interchain protocols will become the key to form a "value internet".

1.2 Definition

We have proposed a universal interchain messaging protocol - IBTP (Inter-Blockchain Transfer Protocol) , owing to requirements of interchain interoperability. Based on the protocol, we have implemented an interchain technology reference implementation called BitXHub, which supports both isomorphic and heterogeneous blockchain transactions including asset exchange, information sharing and service complementation. BitXHub includes three roles: Relay-chain, App-chain and Pier (an interchain gateway). And it has three core functions: universal interchain messaging protocol, heterogeneous transaction validation engine and multi-layer routing, which ensure the security, flexibility and reliability of the platform.

BitXHub is committed to building a scalable, robust, and pluggable inter-blockchain reference implementation, that can provide reliable technical support for the formation of a blockchain internet and intercommunication of value islands.

2 Design Overview

2.1 Architecture

In order to ensure the credible validation and transfer of transactions in heterogeneous blockchains and multi-layer blockchains, we present a TCP/IP-like protocol -- IBTP. Our platform is a comprehensive interchain service realization based on IBTP.

As seen in Fig 2-1, BitXHub shows its overall architecture. BitXHub consists of relay-chain, Pier and app-chain, which is a service platform to handle the interaction among cross-layer heterogeneous chains. The primary components of BitXHub are described as following:

- **Relay-chain**: Relay-chain is devoted to deal with the management of interchain, credible validation and routing for interchain transactions. It is a permissioned blockchain which realizes IBTP validation and routing protocol.
- **Pier**: Pier plays the role of collecting and broadcasting interchain transactions. It is not only utilized between app-chains and Piers, but also used between relay-chains.
- **App-chain**: App-chain takes charge of logic of specific events. It is divided into two types: 1) Isomorphic app-chains (the blockchain supporting BitXHub interchain protocol) have similar blocks, similar storage structure of transaction data, same consensus algorithm and secure algorithm, e.g. Qulian consortium blockchain platform. 2) Heterogeneous app-chains commonly do not match the storage structure of block data, consensuses and secure algorithms supported by BitXHub directly, e.g. Fabric[2], Ethereum[3], etc.

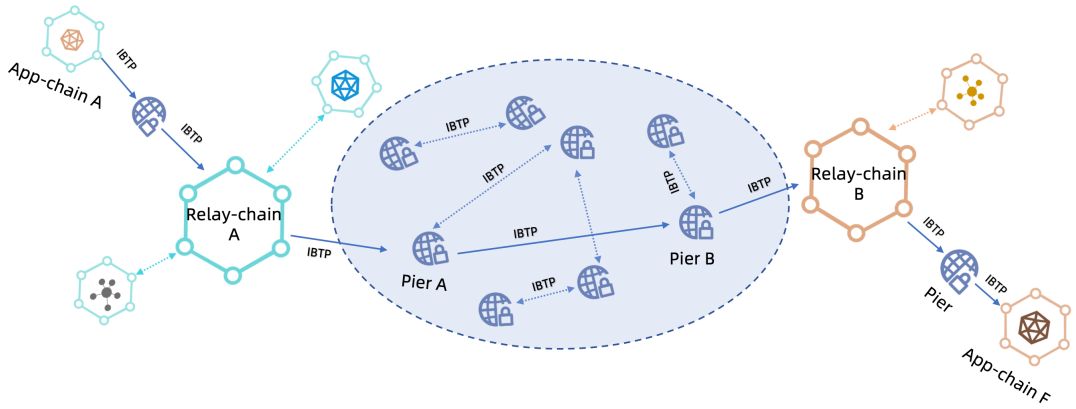

Fig. 2-1 BitXHub Architecture

2.2 Workflow

Due to the limitation of a single relay-chain, we assume that each relay-chain supports 64 app-chains at most. The sepcific number of app-chains will be adjusted dynamically according to the complexity of interchain events. It results in the appearence of many blockchain alliances focusing on relay-chains. Shown in Fig 2-2, it presents the core workflow of handling transactions in BitXHub:

Step 1. App-chain 1 proposes an interchain transaction denoted as ct1;

Step 2. ct1 is sent to relay-chain A connected with app-chain 1;

Step 3. Relay-chain A verifies whether ct 1 comes from app-chain 1 and satisfies the rules registered in A;

Step 3.1 If the process fails, then go to step 4;

Step 3.2 If the process passes, then go to step 5;

Step 4. Roll back the illegal transaction, then go to step 11;

Step 5. Relay-chain A judges whether the receiving chain of ct1 exists in the list of its corresponding app-chains. If it exists, then enter step 6. Otherwise, go to step 7;

Step 6. Send ct1 to receiving chain, then go to step 11;

Step 7. Send ct1 to Pier 1 corresponding to relay-chain A;

Step 8. According to the address of receiving chain corresponding to ct1, Pier 1 queries corresponding Pier 2 in the cluster of Piers by Distributed Hash Table (DHT).

If Pier 2 exists, then go to step 9. Otherwise, go to step 4;

Step 9. Pier 1 transfers ct1 to Pier 2. Then, Pier 2 submits ct1 to its corresponding relay-chain B;

Step 10. Relay-chain B validates the endorsments of ct1 from previous relay-chain. If the endorsments are valid, then go to step 6. Otherwise, go to step 4;

Step 11. Complete the transaction.

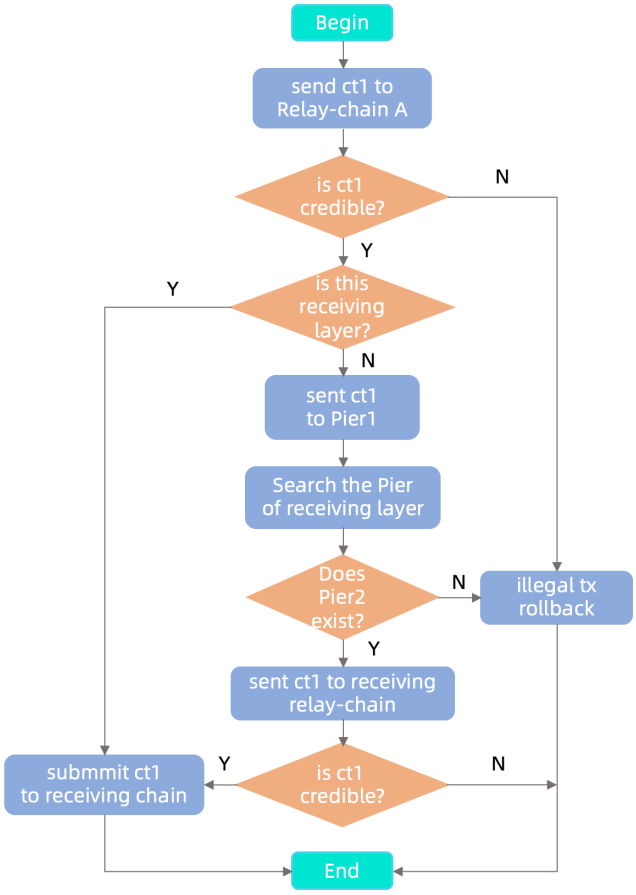

Fig 2-2 BitXHub Workflow

3 Inter-Blockchain Transfer Protocol (IBTP)

The difference of consensus algorithms[4] and signature mechanisms among heterogeneous blockchains leads to different proofs of the existence of transactions. BitXHub designs a unified interchain messaging protocol like TCP/IP in order to process interchain transactions, routing and validation by relay-chains conveniently. It also helps Piers to handle interchain transactions integrally.

3.1 Structure

IBTP sets some primary data fields to ensure its generality and flexibility. The details are shown below:

Table 3-1 IBTP Data Structure

Arguments	Descriptions
From	ID of sending chain
To	ID of receiving chain
Version	Version of protocol
Index	Index of interchain transaction
Payload	Encoded content used by interchain
Timestamp	Timestamp of interchain events
Proof	Proof of inter-chain transactions
Extra	Self-defined fields

"From" and "To" represent the unique ID of sending chain and receiving chain respectively. The ID is generated when app-chain is registered to relay-chain. "Index" is the index of interchain transactions. It is the foundation of sequently packaging interchain transactions on relay-chain , which means the interchain transactions proposed by the same blockchain can keep ordered. "Version" denotes the version infomation of IBTP.

"Payload" is the encoded content of interchain transaction, which supports encryption and is decided by requirements of app-chains. "Timestamp" is a time stamp of inter-field transaction, whcih is expressed by height of block usually.

"Proof" stores the existence validation and legality infomation of interchain transactions. It also provides specific validation information for validation engine of relay-chains. "Proof" will be different according to the specific characterists of app-chains. The rules and methods can be loaded into interchain validation engine flexibly.

Finally, this protocol provides an expand field "Extra", which is utilized by specfic requirement of app-chain flexibly.

3.2 Proof of Interchain Transaction

The "Proof" in IBTP, which reflects the main difference among heterogeneous blockchains, confirms the existence and validity of transactions. The following paragraphs describe the construction of "Proof" in both isomorphic app-chains and heterogeneous app-chains.

3.2.1 Isomorphic App-chain

The following explains how IBTP is constructed in isomorphic app-chains. The structure of MerkleTree[5] in blockchain is described in 4.1.

Figure 3-1 shows the structure of IBTP used by isomorphic app-chain.

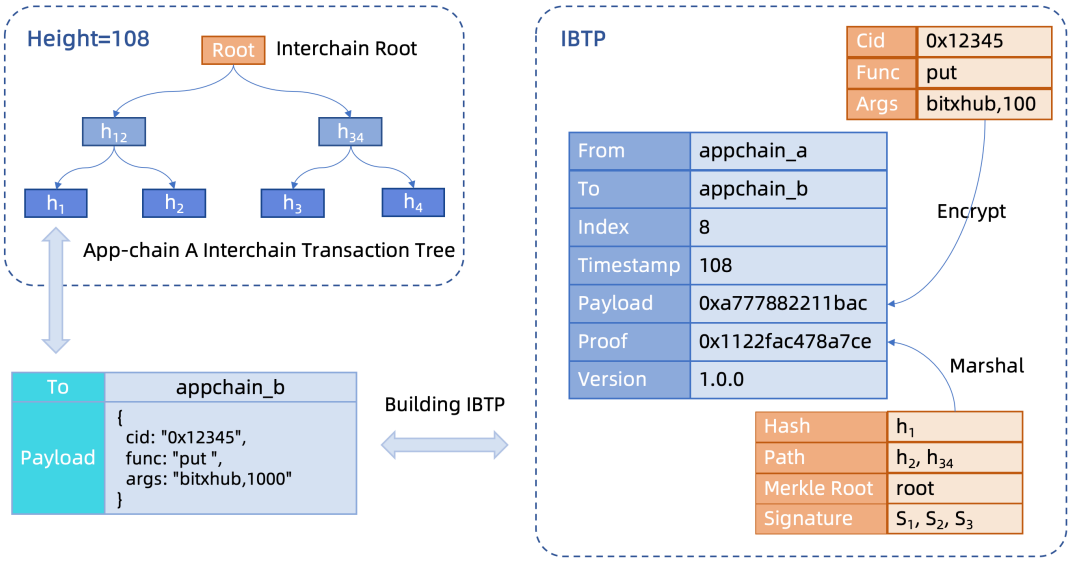

Fig. 3-1 IBTP in Isomorphic App-chain

Take the first interchain transaction in block 108 as an example. As shown in Figure 3-1, this transaction need invoke the smart contract located on the address "0x12345" of app-chain B. The function name is "put" and the argument is "bitxhub, 1000". In this case, the "Proof" provides the information similar to SPV (Simplified Payment Verification). The "Proof" contains "Hash", "Path", "Merkle Root" and "Signature". The "Hash" is the hash of transaction content. The "Path" shows the hash of SPV path. The "Merkle Root" is the hash of final root. And the "Signature" indicates the signed root hash.

3.2.2 Heterogeneous App-chain

According to the consensus algorithm, there are two kinds of heterogeneous app-chains: one is the blockchain where all transactions are immediately confirmed and endorsed by validators, such as Hyperledger Fabric, the other one is the blockchain where no validators exist and transactions cannot be immediately determined, like Bitcoin[6] and Ethereum.

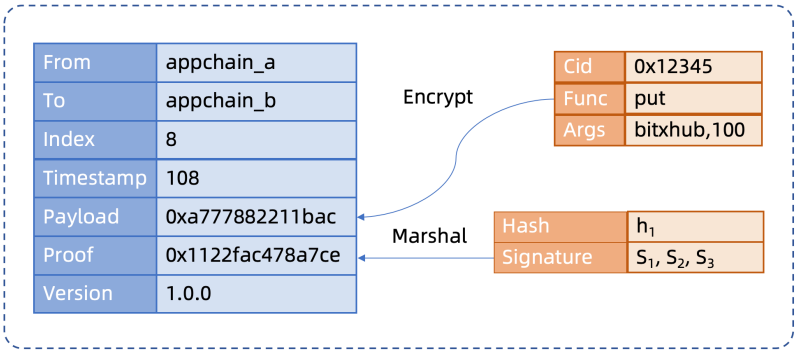

Figure 3-2 IBTP in Heterogeneous App-chain

The block generated by the first kind of the heterogeneous app-chain can be immediately confirmed, and all transactions are endorsed by validators. Thus, Piers can constuct IBTP using the app-chain SDK. Figure 3-2 shows the IBTP structure constructed by the Pier with Fabric v1.4. The "Hash" is the *ProposalResponsePayload* in Fabric. The "Signature" contains the Endorsement array, which is array of signed *ProposalResponsePayload*. The corresponding certifications can be found in register information of app-chain[7].

Take Ethereum as an example of the second type. Ethereum is possibility-based and the interchain transactions it sends are not signed by any nodes. Therefore, in order to prove the existence of interchain transaction, it requires several Piers to sign the transaction and put the signature in the "Proof". To ensure the reliability of Piers, incentives and punishments are adopted.

4 Relay-chain

The relay-chain is used for app-chain management, interchain transaction validation and reliable routing, which is an open permissioned chain based on IBTP. Moreover, the proof of interchain transaction can be transferred among relay-chains to realize trusted inter-layer delivery[8].

4.1 Storage Structure

The relay-chain can be an app-chain in BitXHub. Compared to general blockchain, the relay-chain has one more Merkle Tree generated by interchain transactions. As shown in Figure 4-1, the interchain transactions in each block form a new Merkle Tree. The leaf node is the hash of an interchain transaction. Combined the general root hash with the interchain root hash, a new Merkle Root can be calculated, which will be signed by validation nodes.

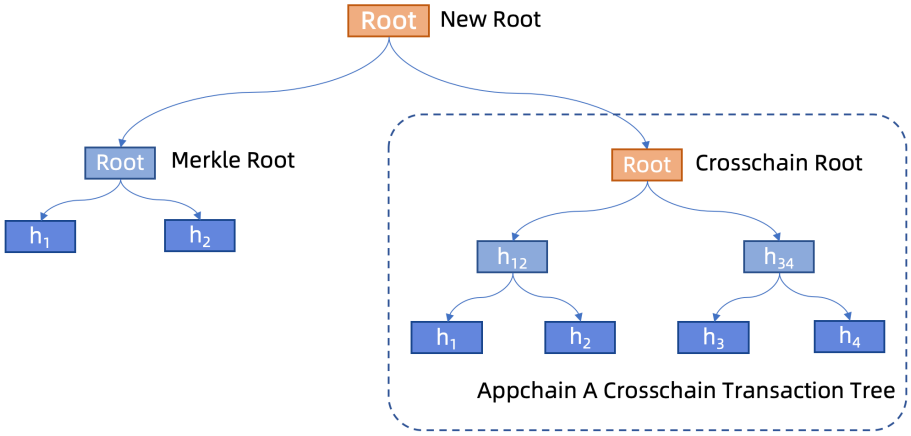

Fig. 4-1 Relay-chain Storage Structure

4.2 Validation Engine

We propose an efficient and pluggable validation engine, which supports interchain transaction validation and reliable routing. That is, interchain transactions sent from different app-chains can be validated using the corresponding rules which can be dynamically registered or updated.

For each interchain transaction, the relay-chain must validate it to avoid conditions that the transaction is created or modified by malicous nodes. The engine can manage multiple complex validation rules using the way like smart contracts, which can support the hot update of validation rules. In this case, the validation engine can handle the rapid iterations of different blockchains.

The validation engine has three main modules:

- **Protocol Parse Module**: This module will parse the interchain transactions. Because all interchain transactions adhere to IBTP, the module can parse the sending chain ID and the "Proof" as the later input for validation engine.
- **Rule Matching Module**: This module will use the sending chain ID parsed by Protocol Parse Module to get the corresponding validation rule. The rule can be found in register information of app-chain. When a new app-chain registers to the relay-chain, the relay-chain will put the scripts of validation rules into the validation engine.
- **Rule Execution Module**: This is a core module of the validation engine. The main part of the module is a WebAssembly Virtual Machine (WASM VM)[9]. The VM can dynamically load different validation rule scripts, which realize the hot update of different validation rules.

The Figure 4-2 shows how the validation engine works:

1. When a new app-chain need to the relay-chain, the scripts of corresponding validation rules need to be registered to relay-chain.
2. When an interchain transaction is sent to relay-chain, the relay-chain will deliver the transaction infomation to the validation engine. The engine will parse the information and get the ID of the sending chain. Using the ID, the engine can find out the scripts which can execute the corresponding validation rules.
3. After finding out the scripts, the engine will call the WASM VM to execute the scripts. The WASM VM will validate the "Proof" in IBTP and confirm whether the

transaction is valid.

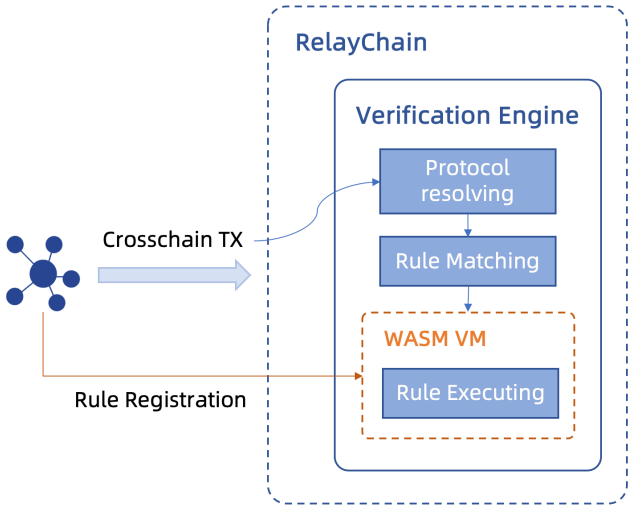

Fig. 4-2 Validation Engine Processes a Transaction

Based on the structure described above, the engine has several advantages:

- Efficient: The WASM VM can make the process of validation efficient.
- Flexible: The update of validation rules is fast and low-cost.
- Comprehensive: The structure of validation engine can satisfy the needs of varied blockchains.
- Convenient: The app-chain staff and developers can manage and update the validation rules directly.
- Secure: The WASM used in the engine is restricted. The developer can only use several functions and libraries permitted by the engine.

4.3 Transaction Routing

Besides validation, relay-chain is also responsible for routing. There are two things need to be routed: one is the interchain transaction, the other is the callback of the interchain transaction.

All transactions which need to be routed in a block will form a Merkle Tree as shown in Figure 4-3. Then, the Merkle Root will be signed by validation nodes. (AC1 indicates the No. 1 interchain transaction from chain A to chain C)

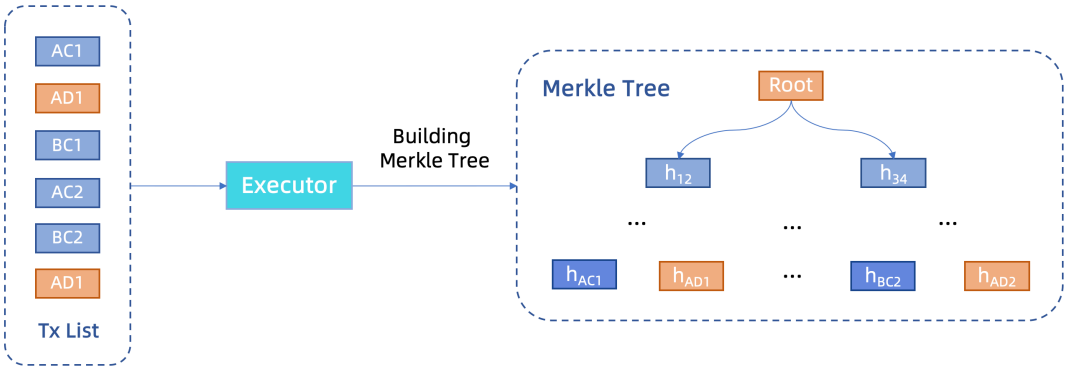

Fig. 4-3 Building Merkle Tree

The routing module will classify all transactions according to the ID of receiving chains after the relay-chain finishes processing a block. Figure 4-4 shows the routing module classify the transaction into two parts: C and D. Pier C and Pier D will synchronize the transactions and proofs from the corresponding parts. If the transaction is an inter-layer transaction, the relay-chain information need to be added into "Proof" of IBTP, which can be validated by the receiving relay-chain.

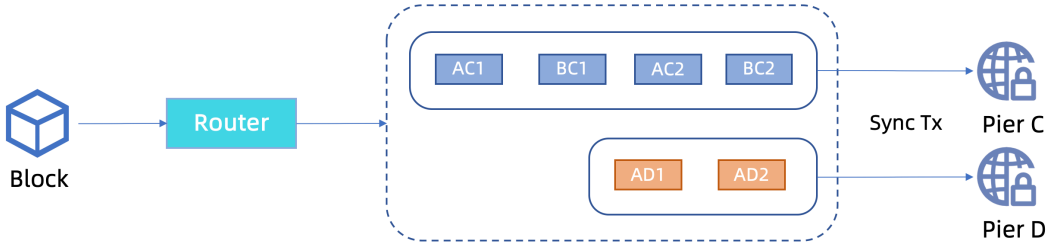

Fig. 4-4 Interchain Transaction Classification and Execution

Piers can parallelly execute the transactions from different sending chains. There are two situations when Piers finish the execution. If there is no callback for the interchain transaction, the transaction finishes when the receipt is sent back to the relay-chain. Otherwise, the transaction finishes when the receipt is sent back to the sending chain after packed in the relay-chain.

5 Interchain Gateway

Due to the sophistication of interchain scenarios, the key to stimulate the ecology of interchain system is the convenience of access to it, which leads to the extensibility and scalability.

So interchain gateway Pier is designed to satisfy the convenience for heterogeneous chains to access to interchain system and the extensibility of multi-layer system.

In essence, Pier is an intermediate module that takes part in filtering and connecting in our system. It is an interactive component that connects different blockchains. Pier is designed to serve two different roles：

1. Connection between app-chain and relay-chain
2. Connection between different relay-chains

On the one hand, Pier acts as a middleware, bridging app-chain and relay-chain on the single relay-chain layer. Suffered from the complexity of connecting heterogeneous blockchains, interchain system is desperate to have an intermediate module to simplify the process and enhance the convenience of access to interchain system.

On the other, it can be a "router" in multi-relay-chain system layer. Piers form a P2P routing network and each Pier contains a part of the interchain routing table. The entire Pier network realizes the efficient broadcast of inter-layer transactions through DHT.

Despite of the two roles in different layers, Pier is still a intermediate module interacting with two blockchains in essence.

For the convenience and scalability described above, the design of how app-chains are plugined into a interchain system and the implementation of the core functions of Piers are well considered.

5.1 Core Functions

5.1.1 Transaction Collection

The fundamental function of Pier is monitoring transactions, and more importantly, ensuring the orderliness and validity of interchain transactions. To achieve this, Pier needs higher permission on the app-chain. For example, when validating a fabric interchain transaction, it needs to have permission to obtain endorsements and validate them.

Due to different mechanisms in heterogeneous blockchains, the strategy of collecting interchain transactions should also be determined respectively. Here we introduce two different ways to collect transactions both in isomorphic app-chains and heterogeneous app-chains.

As for isomorphic app-chains, IBTP struct can be constructed with the SPV validation from Merkle Tree after one interchain transaction has been caught.

In heterogeneous app-chains, the main obstacles to construct IBTP struct are weakness of probabilistic consensus and lack of ability of signing interchain transactions like Ethereum. The consensus of Ethereum is based on PoW[10], which means even if when one interchain transaction is received by Pier, it cannot confirm the transaction. Thus the strategy here is to wait for transaction confirmation (conventionally 20-block time). Afterwards, the interchain transaction can be sent to the relay-chain.

In addition, considering that single Pier may send unconfirmed transactions to relay-chain, it is necessary to enhance the security with a Pier cluster (the specific figure is shown below), where each Pier is endorsed by an authority. And relay-chains adopt a penalty mechanism to prevent several Piers from cheating together. Moreover, Ethereum itself does not provide signatures for interchain transactions, which can be solved by using multiple Piers to confirm the existence of interchain transactions and to sign for these interchain transactions.

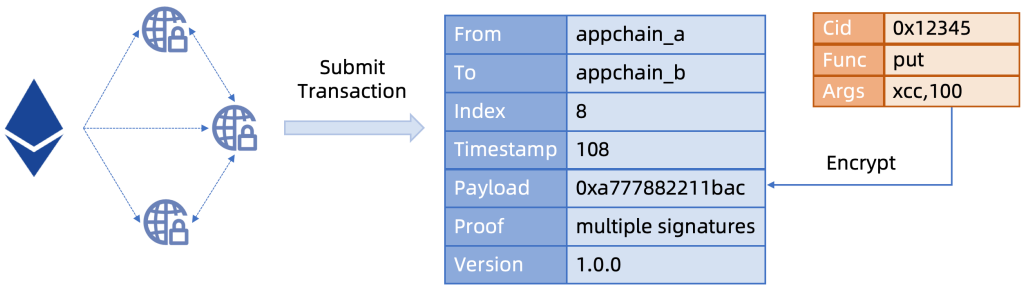

Fig. 5-1 Construction of Interchain Transaction in Heterogeneous App-chain

5.1.2 Synchronization and Execution

After receiving several interchain transactions, the relay-chain will pack them into a block. Each Pier connected with the relay-chain will automatically synchronize related interchain transactions and make the SPV validation.

After confirming that received transactions are credible, Pier will unpack the payload part of the interchain transaction to get the IBTP struct. Next step is to construct and commit a transaction to the app-chain.

5.2 Plugin

Given that the architectures of existing blockchains are quite different from each other, it is a challenge to design an interchain system which can access these heterogeneous blockchains conveniently. To break down the complexity of the interchain system, a novel plugin mechanism is adopted.

We believed that complexity is mainly caused by the tight coupling between the module interoperating with app-chains and others. Therefore, we turn the former to a plugin, which is relatively independent from Pier. Plugins are responsible for interacting with their related app-chains and are dynamically loaded at runtime. This mechanism requires following interfaces implemented by each plugin:

- Monitoring interchain events on app-chain
- Executing interchain transactions from other app-chains
- Querying the transactions status from receiving chain

In general, plugin has superorities below:

- Hot update
- Easy adaption to various app-chains, with no need to change Pier code

5.3 Interchain Routing Network

When talking about the network of multiple relay-chains, interchain transactions between relay-chains require a routing system, which is a P2P network consisting of a Pier cluster.

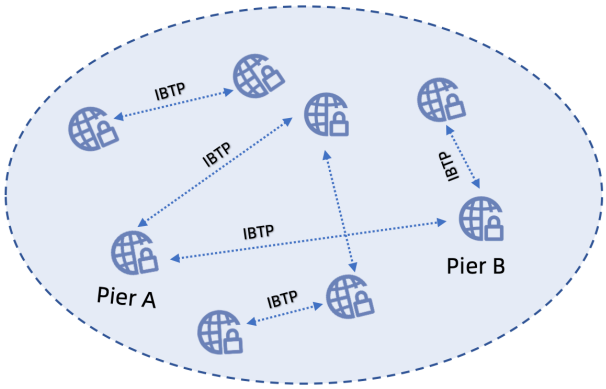

Fig. 5-2 Network Topology of Routing System

Each Pier maintains a routing table of the entire network, the record of which is a mapping from receiving chain to its related relay-chain.Whenever added into the network, a new Pier need to broadcast information about its relay-chain and app-chain. The Information is critical for other Piers to update and maintain their routing tables. When one Pier is trying to send an interchain transaction, the first thing is to find another Pier which is connected to the target relay-chain on the routing table. Then the target Pier will validate the interchain transaction with the signed information from source relay-chain. If passed, it will be sent to its target relay-chain. And the receipt of the transactions will be routed the same way to where it is from[11].

6 Conclusion and Future Work

BitXHub provides an innovational and trustable interchain strategy. It sets a common interchain transfer protocol (IBTP) as the fundamention to create an inter-blockchain platform. Holding on the ideas of decentralization, scalability, high performance and availability, our platform not only realizes the value interoperability of on-chain assets, data and services but also changes blockchain status from "island" to "network".

The vitality of interchain technology comes from all developers and users. BitXHub wishes to propose a universal protocol to enhance the public confidence of interchain platform. It also wishes to build a free, active and advanced open source community to enrich the standards. Depending on these, it can connect various blockchain platforms to exploit the ecological system and excavate the further value of interchain.

References

[1] Buterin, Vitalik. "Chain interoperability." R3 Research Paper (2016).

[2] Cachin, Christian. "Architecture of the hyperledger blockchain fabric." Workshop on distributed cryptocurrencies and consensus ledgers. Vol. 310. 2016.

[3] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.

[4] Castro, Miguel, and Barbara Liskov. "Practical Byzantine fault tolerance." OSDI. Vol. 99. No. 1999. 1999.

[5] Szydlo, Michael. "Merkle tree traversal in log space and time." International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2004.

[6] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).

[7] Fabric docs: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

[8] Kan, Luo, et al. "A multiple blockchains architecture on inter-blockchain communication." 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2018.

[9] Haas, Andreas, et al. "Bringing the web up to speed with WebAssembly." ACM SIGPLAN Notices. Vol. 52. No. 6. ACM, 2017.

[10] Chen, Zhi-dong, et al. "Inter-blockchain communication." DEStech Transactions on Computer Science and Engineering cst (2017).

[11] Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, 2016.

