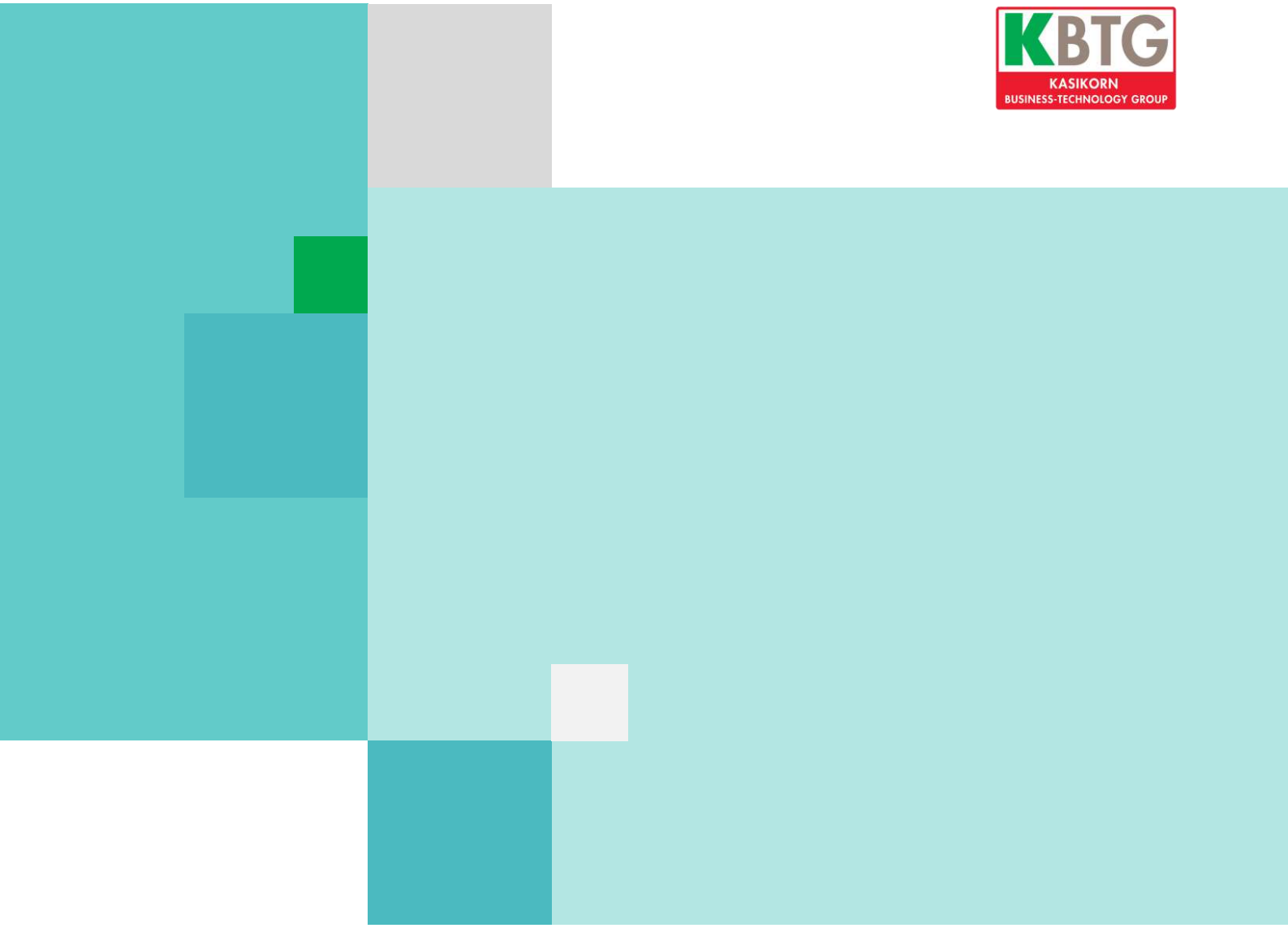

Cross-chain Integration for The Future of Finance

(Hyperledger Fabric & Ethereum Integration)

CO-RESEARCH PAPER

21.05.2021

INTRODUCTION

Blockchain adoption in financial industries grows exponentially in recent years. Several financial services transform from inefficient and centralized to consortium-driven that reduces redundancy operations and delivers trustworthy services to customers.

From the other end of the spectrum, Decentralized Finance (DeFi) proves itself as a potential future of financial services that require no centralized authority to control the operation. DeFi also grants more transparency to financial services, since everyone can access and prove the smart contract.

With this information, we see an opportunity in intertwining traditional financial services to DeFi to bring the benefit from the best of both worlds. It innovates new kinds of financial services that are backed by the trustworthiness of traditional financial instruments and of open architecture to the table.

In this report, we are focusing on the technical implementation of interconnecting a consortium-based use case to a public blockchain.
Tokenization of a financial product is one of the gateways to create new use cases.

Likewise, the selected use case, invoice financing, utilizes blockchain to interconnect altogether: banks, sponsors, and suppliers. The traditional invoice financing is moved to DeFi by making tokenization over invoices. The tokens allow

investors to purchase and trade invoices in the exchange market. Every step from invoice financing to trading is operated and verified automatically by smart contracts. Moreover, the return might be higher than the traditional way because the operating cost of the centralized authority is removed.

BUSINESS CASE

The technical perspective cannot measure solely how suitable the prototype is. A proper business use case is also mandatory for measuring.

The chosen use case offers distinct benefits. First, it helps small business owners to conserve adequate cash flow. Then, it also helps to offer retail investors an alternative way of investment. Last but not least, to ensure the data integrity for companies holding customer sensitive data. The use case is technically feasible based on the fully integrated digital foundation and blockchain for financial stability.

On the aspect of supplier business, liquidity is very crucial. An uncomplicated way to enable liquidity is to transform an invoice into cash. Differently, the classic invoice financing way is to let the suppliers request a loan from banks by using an invoice as collateral. If the invoice will be paid by a reliable sponsor, banks will offer a loan. The use case is taking a step further. The invoice will be tokenized and the tokens can be exchanged like a stock. In result, the invoice amount will be divided into small fractions with the affordable amount per unit and can distribute to more retail investors. In the meantime, investors can invest in this token with the

guaranteed return because the collateral has a reliable sponsor with an agreed payment period. However, the ownership of the tokenized invoiced must be depended on the related law.

The loan service focuses on suppliers that have transactions with creditable sponsors. Suppliers might get loans from banks by using holding invoices. By making a loan, suppliers receive the money earlier than the agreed payment period from sponsors. After banks ensure the authenticity of the invoice with sponsors, banks will approve the loan to suppliers.

A process to turn any invoices into asset tokens is called tokenization. Under Thailand's regulations, the asset tokens must be issued and distributed during initial coin offerings (ICO) by an ICO Portal. The ICO Portal has multiple responsibilities like due diligence, converting physical assets to be digital assets, offering the ICO, and paying return after invoice collection. Every task is done automatically by smart contracts. The combination of invoice financing and digital asset tokenization helps small and large businesses. It allows many parties to construct a new ecosystem for lending and investing altogether. This business becomes faster lending using an invoice and allows investment on invoices by a digital token. Moreover, it also offers an opportunity for retail investors to alternatively invest in the new product.

Furthermore, retail investors are finding a new opportunity to invest in a low-risk business and gain better profits than bank savings or inflation. If so, an invoice is used to make a loan with a bank and be tokenized. The tokens are sold out to

retail investors. They receive an amount of return periodically. All involved parties have a win-win situation.

OBJECTIVE

Currently, KBTG is running in production a blockchain as a foundation called Hyperledger Fabric which contains document-based data. The elected use case also uses the invoice data, loan data and trades it through the Ethereum network on ERC-20 protocol. The tokenization and exchanging part utilize Hyperledger Besu and will be implemented by Atato. These are used in a different blockchain. The white paper's objective is to archive interconnection between two blockchain networks via decentralized oracle which is an exchange channel. Furthermore, a virtual ecosystem is built to support digital currency exchange for investment units as digital tokens.

By proposed design, banks can manage requests and documents for the invoice financing process easier than before. After the application, an invoice is bound to a loan document. It means the invoice is ready for the tokenization process. The tokenization process is converting the invoice to be ERC-20 tokens and allow everybody to exchange over the platform.

The invoice and loan processes aim to operate over the document-based platform. All data is stored securely on Hyperledger Fabric blockchain. One of the key features of Blockchain technology is immutability. Therefore, the integrity of data is guaranteed. Also, it maintains the logics using document-rule and smart contracts to administer the flow.

The system needs data privacy. So, the private blockchain is the answer.

Tokenization is one of the DeFi use cases. The tokenization will operate over both public and private blockchain. The blockchain which fits these tasks is Hyperledger Besu. There is no way to transform it into tokens. The integration is needed to convert these data into tokens. The decentralized oracle will do that to reach the goal. The invoice and loan, all of the processes and calculations need to program somewhere. Formulas and logics do not store in the application. The blockchain is the place to deploy them as a smart contract and ensure that all dApps have the same formulas and logics.

The purpose of the research presents a possible use case that can run multiple blockchain networks altogether. Existing assets might be an application based on blockchain. A use case might need to expand to be a more exciting business to improve competition in a market.

LITERATURE REVIEW

Regarding the current state of the art in blockchain integration, there are various kinds of frameworks and techniques currently used in this field. As discussed in (Belchior et al.), the solutions for blockchain interoperability can be divided into three categories: Cryptocurrency-directed approaches, Blockchain engines, and Blockchain connectors.

Cryptocurrency-directed approaches

Cryptocurrency-directed approaches are focused around cryptocurrency trading, for example the authors (Belchior et al.) presented Sidechains which allow for the transfer of assets between a main chain and a sidechain where a user can lock X number of tokens on the main chain, and get an equivalent amount of tokens created on the sidechain. This technique is a so-called two-way-peg. The underlying submitted blockchain transaction must be verified by a smart contract, and then it will be relayed to another smart contract in the side chain via Relayers.

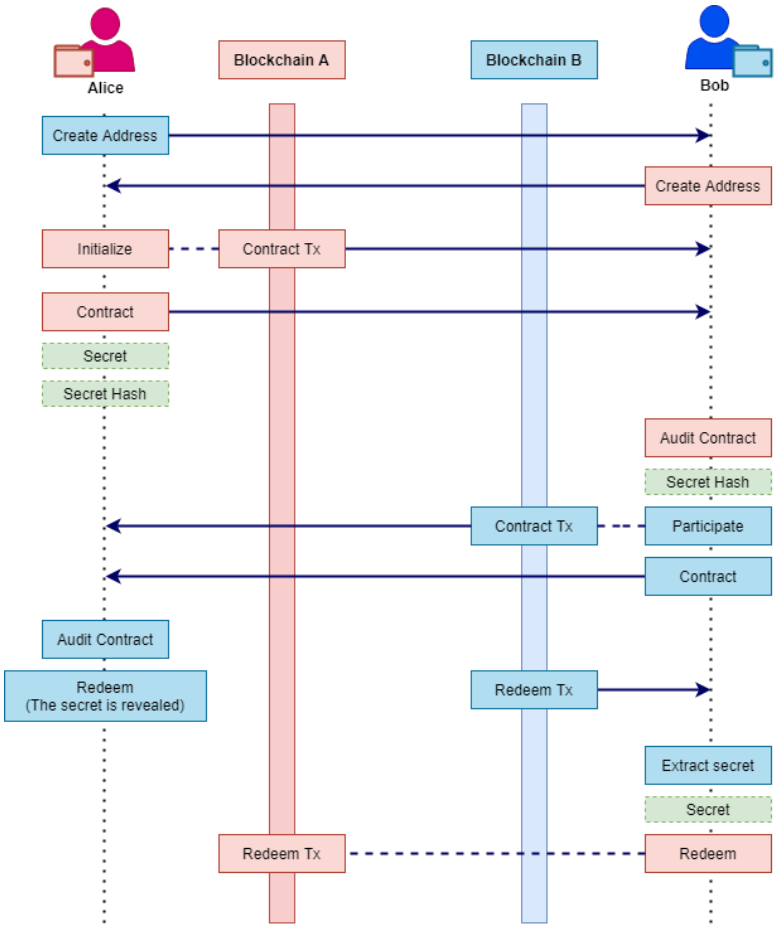

Figure 1: A diagram illustrates the interaction between Alice and Bob on Blockchain A and B in order to exchange their tokens held on different chains using Hashed Time-locks technique. It is based on the diagram in (Koens).

Another approach is considered as a cryptocurrency-directed way is Notary schemes; a notary which keeps monitoring events occurring in multiple chains and triggers transactions in another chain. This solution has been widely used in both centralized and decentralized cryptocurrency Exchanges. Another technique is called Hashed Time-Locks which is used to enforce atomicity of operations between two parties. A subset of this kind of techniques is Atomic swaps allowing two parties to exchange cryptocurrencies held on different blockchain networks. As illustrated in Figure 1, Alice creates a smart contract on Blockchain A with the locked tokens which will be transferred to Bob ledger if it gets unlocked. The lock depends on both time and a secret only known to Alice at that point in time. Afterwards, Alice will send the hash of the secret to Bob. Bob will then create a smart contract on Blockchain B with the locked tokens which will be transferred to Alice ledger if it gets unlocked. Similarly, the lock depends on both time and the hash given by Alice. Subsequently, Alice will redeem the tokens on Blockchain B; this will reveal the actual secret to Bob. Eventually, Bob will be able to redeem the tokens on Blockchain A using the revealed secret. Please see (Koens) for more details of Atomic swaps. Overall, these mentioned solutions are focusing around cross-blockchain token transfers.

Blockchain Engines

The second category described in (Belchior et al.) is Blockchain engines which aim to provide platforms to share infrastructure that supports different layers in blockchain

architecture (e.g., data, network, consensus, incentive, contract, and application layer); the engines will offer interoperability among the blockchains within the platforms. Some of the well-known solutions regarding this category are Polkadot and Cosmos.

Polkadot provides developers with Substrate framework to build customized blockchains to become part of its parachain network. Parachains can be viewed as parallelized chains participating in Polkadot network. Independent chains such as Bitcoin or Ethereum can be linked to the network via bridges. The core of Polkadot interoperability is based on state transition validation, handled by the chain-relay validators. On the other hand, each independent parallel blockchain in Cosmos is called zone. Zones are connected via Hubs. The inter blockchain communication protocol is used to achieve cross-zone transactions. Both of these solutions are mainly discussed in the context of public blockchains. In the case of Hyperledger Fabric and Ethereum integration, these solutions might not yet be applicable to the case. Additionally, as discussed in (Polites), a main drawback of Polkadot is the absence of atomic composability between parachains, which means that dApps on different parachains aren't able to interoperate seamlessly. The author also emphasizes that atomic composability is the key feature required for DeFi applications.

Blockchain Connectors

The last category which is discussed in (Belchior et al.) is Blockchain Connectors which are the cross-chain

communication solutions that do not belong to the first two categories. It can be divided into multiple sub-categories, but the most applicable one would be Trusted Relays which requires trusted parties to redirect transactions from source to destination blockchains. One example of Trusted Relays is Hyperledger Cactus which enables interoperability through a set of validators collecting cross-chain transactions requests, signing, and delivering them. This solution also supports Hyperledger technologies such as Fabric and Besu (Ethereum). Another sub-category worth mentioning is Blockchain Agnostic Protocols such as Interledger Protocol (ILP). One of its implementations is Hyperledger Quilt; the Interledger Protocol can be considered as a decentralized, peer-to-peer payment network relying on a generalized hash locking scheme.

Apart from the frameworks reviewed in (Belchior et al.), it's worth mentioning the Ion framework (Clearmatics). It is primarily focused on interoperability between permissioned or private blockchains. The framework aims to be use-case agnostic. It allows developers to build contracts that interacts cross-chain using a concept called continuous execution. Basically, continuous execution is the ability to execute a function on a blockchain if and only if a specific state transition is proven to have occurred on another blockchain; this means it requires blocks from the other blockchain to be stored and validated before the function execution. Similarly, on Hyperledger Cactus, there are validator nodes which run a consensus algorithm to agree upon the state of the underlying blockchain and a proof of state of the blockchain is produced and signed by several validator nodes (Hyperledger

Cactus). On the contrary, Hyperledger Quilt does not maintain block hashes or merkle proofs of the destination blockchain. For more details on the overall design of Ion and implementation of Ethereum-Fabric interoperability using Ion, please see (Clearmatics).

Blockchain Oracles

Due to the fact that this paper focuses on Hyperledger Fabric and Ethereum integration, it is worth taking a look at common solutions that have been used in Ethereum blockchain. The problem we are trying to solve in this case is about getting data into a decentralized application (dApp), which can be done via Oracle Services as described in (MatZand). Basically, oracle refers to the third-party or decentralized data feed services that provide external data to dApps. The author divides oracles into two types which are software oracles and hardware oracles. Software oracles provide smart contracts with easily accessible online information such as stock index prices, weather forecasts, etc. On the other hand, hardware oracles provide scanned information such as registered mail scanning, supplier goods delivery scanning and so on.

In our use case, the data required by our private Ethereum blockchain (Hyperledger Besu) will be sourced by Hyperledger Fabric. To put it another way, we can wrap Hyperledger Fabric with RESTful APIs which will provide the required information to our Ethereum blockchain. The communication between the RESTful APIs and Hyperledger Fabric can be done via gRPC and Hyperledger Fabric SDKs (Agarwal). This way, the data can be fed into the Ethereum

smart contracts with the help of oracles. Note that this is only one-way communication which passes data from Hyperledger Fabric to Ethereum. If the data on Ethereum must be sent back to Hyperledger Fabric, we definitely need a different solution that supports two-way communication. Even though the use case in this paper is limited to private blockchains on both sides, one might extend the use case to consortium or public blockchains to make it become more useful in the future. Therefore, to make the solution future-proof, it is worth taking a look into a decentralized oracle protocol.

Chainlink

In recent years, Chainlink has gained a lot of attention in the blockchain community. According to its whitepaper (Ellis et al.), existing oracles are centralized services which introduce a single point of failure. Chainlink team aims to make a decentralized oracle network including reputation and security monitoring services. It is composed of both on-chain and off-chain components. Chainlink's on-chain component is first responsible for selecting oracles via off-chain listing services. In most of the use cases, there will be more than one source of data coming into the on-chain component which is also responsible for result aggregation. A different aggregation strategy can be used to derive the final result. If it is numeric data, averaging across all the data sources might be considered with removal of outliers. If the value is not a number, the majority of sources must return the identical value; otherwise it returns an error. The last part of the on-chain component is data reporting which plays an important

role in the reputation system providing a means for users to evaluate oracle performance holistically.

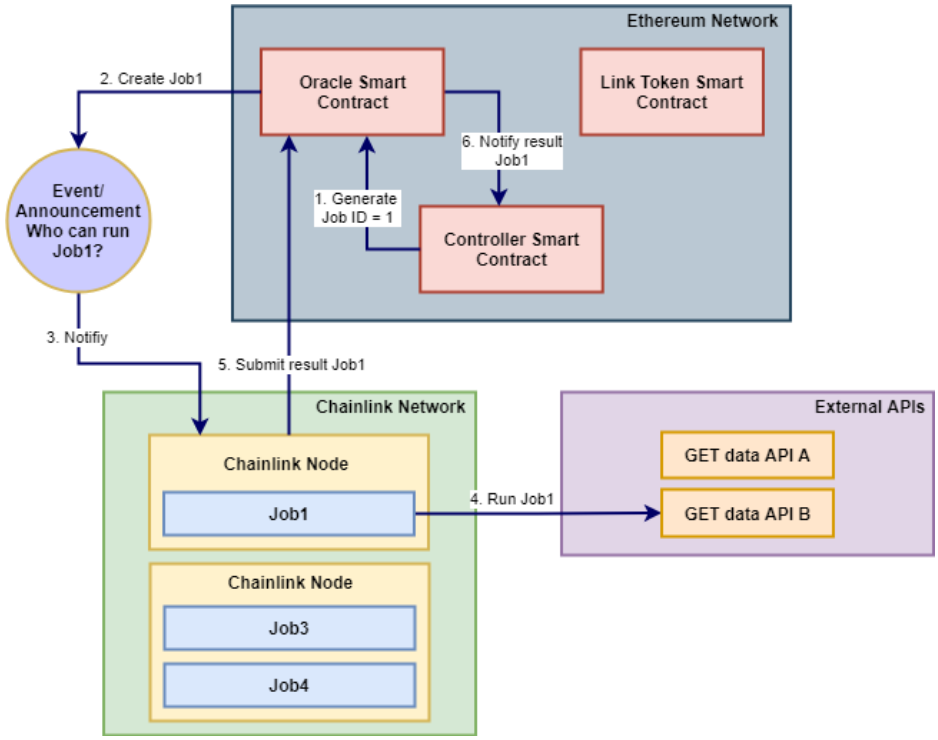

Figure 2: Simplified Chainlink components together with Ethereum blockchain and External APIs (Invoice Financing Platform API). It is based on the diagram in (Zhang).

As illustrated in Figure 2, Chainlink off-chain components consist of a network of oracle nodes connected to the Ethereum network. Basically, these oracle nodes (Chainlink nodes in the diagram) are responsible for working on the assignments requested by the dApp smart contracts. Each assignment, also known as a job, is a set of smaller subtasks. There are two types of these subtasks: core and external adapters. A basic example of a core adapter is a task to make a HTTP request, parse JSON response, or convert a value to a blockchain format. However, beyond these built-in subtasks, one can define custom subtasks so-called external adapters. This allows interacting with complicated multi-step

APIs to be simplified. No matter if it is a core or external adapter, the job specification will be defined using JSON Schema.

One key aspect of Chainlink is how it makes oracles secure. Getting false data feed might lead to payout to the incorrect party. In other words, there is no perfectly trustworthy data source, since the data maybe maliciously corrupted due to faulty web sites, cheating service providers, or honest mistakes. One proposed solution is to distribute the oracle requests across both oracle nodes and data sources; this involves both on-chain and off-chain result aggregation. In addition, Chainlink validation system should record oracle failures as well as apparent erroneous responses by oracles. Developers can consider both availability and correctness statistics for oracles in order to select the best performing oracles for their jobs. Last but not least, Chainlink's reward and penalty model which relies on ERC-20 LINK token to pay the node operators for the retrieval of data from off-chain data feeds, and during oracle selection the operators need to deposit the penalty amount with an oracle service provider; this amount would be lost due to their misbehavior. An example of the misbehavior would be erroneous responses by the oracle as measured by deviations from the responses provided by its peers in case of result aggregation across multiple oracles.

Comparison

	Decentralized?	Requires trusted 3rd party?	Public chain support	Private chain support	EVM support	Comments
Hyperledger Cactus	Yes	No	Yes	Support both Hyperledger Fabric and Hyperledger Besu.	Yes	It allows developers to build APIs handling transactions that span across multiple blockchains. It is not production ready at the time of writing.
Hyperledger Quilt	Yes	No	Yes	Yes	No	It is focused on building payment networks to transfer value across distributed ledgers using a hash locking scheme.
Clearmatics Ion	Yes	No	Yes	Focused on permissioned or private blockchains.	Yes	It allows transactions on one blockchain to be executed conditionally based on the state from another blockchain.
Polkadot	Yes	No	Yes	Yes	Yes (e.g. Moonbeam)	It allows cross-chain communication via an asynchronous message-passing protocol which guarantees the delivery of messages. It is not production ready at the time of writing.
Cosmos	Yes	No	Yes	Yes	Yes (e.g. Ethermint)	It is similar to Polkadot, but with its own scheme and messaging protocol.
Atomic Swaps	Yes	No	Yes	Yes	Yes	It allows transactions on different chains to be executed atomically using techniques like Hash Time Locked Contract.
Chainlink	Yes	No	Yes	Yes (supported by BaaS platform)	Yes	It provides smart contracts with data from the outside world in a decentralized way.
APIs	No	Yes	Yes	Yes	Yes	It provides smart contracts with data from the outside world. It does not support decentralization out of the box.

Table 1: Comparison of blockchain integration solutions

All in all, each solution discussed earlier is aimed at solving different kinds of problems. In order to pick the most suitable solution, we need to analyze each aspect of our problem first. As you may already know by now, all of the techniques and integration frameworks discussed so far are focused on either asset transfer or arbitrary data transfer between different blockchains. For the proof of concept which will be discussed later on, the overall architecture is divided into two main components; first component is the invoice financing platform which is composed of Hyperledger Fabric and the web services providing access to the data stored on the blockchain. Second component is a consortium Ethereum blockchain using Hyperledger Besu. In our use case, Hyperledger Fabric will be the primary source of data which needs to be fed into the Ethereum blockchain. Furthermore, the invoice financing data required can be divided into multiple fields. Hence we need a tool that helps us to take off-chain arbitrary data into the Ethereum blockchain.

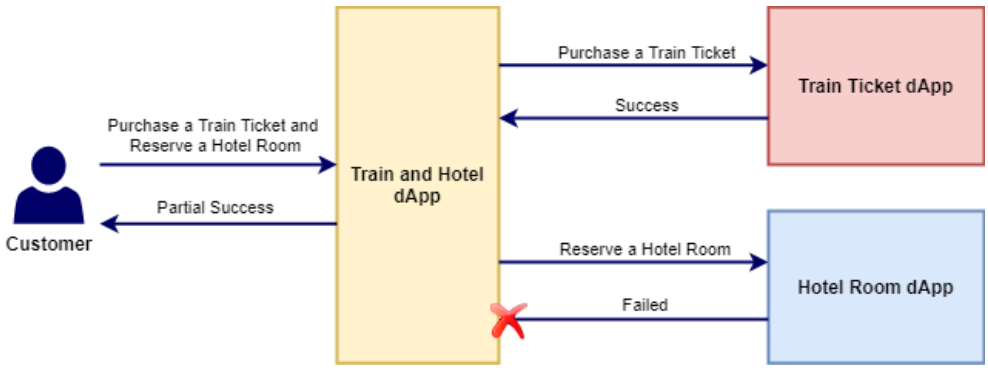

Figure 3: A sequence diagram illustrates partial success on a Train-and-Hotel transaction across different shards

Another key aspect of the solutions is to enable cross-chain atomic operations. A technique like Hashed Time-Locks has been used to facilitate the exchange of assets across chains.

On the other hand, when it comes to atomic composability, a more sophisticated protocol must be implemented. This is because one way or another we will run into the Hotel-And-Train problem where a consumer wants to both purchase a train ticket and reserve a hotel room in an atomic way. This is possible when the transaction happens on a single blockchain (shard), but hard to achieve when it spans across multiple blockchain networks in an asynchronous manner. As shown in Figure 3, a consumer purchases a train ticket on one shard, which in turn reserves a hotel room on another shard. If the hotel reservation fails, the prior transaction must be rolled back as well. As discussed in (Zhou), one way to solve the problem is to embrace eventual consistency and introduce a way to handle asynchronous messages with callback. However, it's only possible if the operation is revertible, but it is not the case in most Decentralized Finance (DeFi) Applications. This kind of a distributed transaction can also be solved by the use of the classic two-phase commit protocol involving locking assets for extended periods of time, similar to the proposed solution discussed in (Das et al.). If your dApp requires atomic composability, it is definitely worth to apply such kind of solution. In addition, it would be very interesting to explore the possibility of using Hash Time Locked Contract to solve such a Hotel-And-Train problem, probably by locking transaction receipts instead of tokens.

In the proof of concept (POC), it is safe to assume that there is no need for atomic composability, but it might be one of the enhancements to make when the product is going live in a production environment. After we have evaluated all the solutions, it seems that Chainlink is the one that suits our

requirements at least for POC. Firstly, this is because the flow of data between Hyperledger Fabric and Ethereum is unidirectional from Hyperledger Fabric to Ethereum. This exactly matches the use case of oracles. Additionally, as discussed earlier, atomic composability is not required for the POC; Chainlink seems to qualify all of our requirements, with the extension to make the oracles be decentralized in the future.

PROOF OF CONCEPT

Technology and Capabilities - ConsenSys

A PoC platform was built with an extension of ConsenSys products, in particular our digital assets issuance and lifecycle management platform called Codefi Assets. Codefi Assets enable the configuration of new asset templates with a specific set of metadata attached to them along with a workflow enabling the execution of their lifecycle.

ConsenSys Codefi Assets Platform can be used to issue various asset tokens (ERC-20, ERC-1400, ERC-721, Zk enabled ERC-20, etc.), and manage their lifecycle. Codefi Assets API would be used for integration with different user applications, and Codefi Orchestrate middleware would be used to facilitate orchestration of blockchain transactions, key management, transaction signing, smart contract event management, identity management and messaging.
The following Consensys Products are selected for this PoC:

- Consensys Codefi Assets API
- Consensys Codefi Orchestrate
- Consensys Quorum

Detailed capabilities of each of the products are highlighted as below:

Consensys Codefi Assets API

- Digital Assets Management (Issue, Manage, and Exchange)
- Transaction management
- Account management
- (future scope) Private channels: send private messages on-chain from node to node, according to any predefined schema.

This feature leverages the "privacy group" feature of Consensys Quorum.

- (future scope) L2 scaling

Consensys Codefi Orchestrate

- Transaction management
- Account management
- Private Key Vault Integration, transaction signing
- Smart Contract Management
- Encrypted Network Messaging

ConsenSys Quorum

- Go open-source Ethereum client
- Java open-source Ethereum client
- Private transaction managers
- Private key management
- Additional tolling and upgrades for production.

Figure 4: ConsenSys enterprise stack

Business Case - Invoice Financing

The research does not refer to any existing business. It is an experimental business that is potentially a real business soon. Nowadays, there are some blockchain applications in the financial industry that help manage and store essential data. But, it is accepted only for document-based data. One day, banks want to expand their use case to use digital tokens, it is difficult to modify the current flow of financial processes, and migration to an unproven application is too risky for financial institutions. There are numerous possible ways to apply it for expanding the existing blockchain application and taking much time. The one possible solution is merging the existing with a new application altogether. With this solution, the current application can work as it should and interoperate with a new application without changing much. This research aims to be an example of an existing blockchain application requiring interoperating with any blockchain application.

Secure Document Exchange - Hyperledger Fabric

The secure document workflow is based on Hyperledger Fabric foundation and developed by the KBTG team. The Hyperledger Fabric is an enterprise-grade permissioned distributed ledger framework. By design, the system supports any document schema and creates any document rules upon it. The infrastructure can be either an on premise server or a cloud service. Furthermore, privacy, security, and correctness are the priority. The Hyperledger Fabric has plentiful features and concepts. Some of them are chosen to apply to this

platform. For example, the channel concept is for isolation and confidentiality to any data over the platform. As well as, the endorsement-policy is used for verification before submitting transactions to a blockchain, and the transactions must satisfy the endorsement policy before marking valid. There are two verifications on this platform. Preventing incorrect data and guaranteeing that all data is correct and acceptable. Firstly, an application layer checks documents initially. Secondly, it passes to a smart contract layer that will check document rules per request. This platform ensures the document rules are stored securely on the blockchain and will be fetched when needed.

Secure Value Transfer - Hyperledger Besu

Since the goal of the POC is to showcase the integration between Hyperledger Fabric and Ethereum blockchain, we decide to run the Ethereum blockchain in a private permissioned network on a blockchain-as-a-service provider. The infrastructure is powered by Hyperledger Besu which is an open-source Ethereum client able to run on the Ethereum public network, private networks, or test networks. Besu is widely used to develop enterprise applications requiring secure, high-performance transaction processing in a private network. Our demonstration is focused on a private network, but not limited to it.

Integration - Chainlink

Our setup is considered as consortium blockchains; the first part is Invoice Financing Platform containing Hyperledger Fabric. The second part is the consortium Ethereum

blockchain deployed on blockchain-as-a-service platform which provides the underlying blockchain infrastructure powered by Hyperledger Besu. The missing part here is the integration between Hyperledger Fabric and Ethereum. As discussed earlier, Chainlink as decentralized oracles can be used to retrieve data from Invoice Financing Platform and feed it into the Ethereum smart contracts. To be more specific, we have a Chainlink external adapter which works as a proxy to the APIs provided by Invoice Financing Platform to allow us to query Invoice-related data from Hyperledger Fabric. The adapter handles the authentication with the APIs, as well as transforming the response of the APIs into the right format required by the smart contracts in Ethereum. The adapter is defined as part of the Chainlink job specification.

In order to understand the usage of Chainlink, it is better to have a clear understanding of the workflow of the POC. The use case we chose here is about invoice financing where there are a bunch of stakeholders involved:

- Bank: A financing company which accepts the transaction, buys the invoice, and gives the seller a cash advance.
- Seller: A business which sells a good or service to a customer (buyer). The business then issues an invoice and gives the buyer a time window to complete the payment.
- Buyer: A customer who purchases a good or service from the business (seller).
- ICO Portal: An entity which aims to protect investors. It controls the initial coin offerings (ICO) and auditing their smart contracts. It also has the objective of avoiding financial crime by enhancing know-your-customer (KYC) processes.

- Retail Investors: Interested investors who buy into the offering and receive a new cryptocurrency token approved by the ICO Portal.

The core workflow can be divided into 6 steps; the first 3 steps occur on Invoice Financing Platform as illustrated in Figure 5. During step 4, the data will be fed into the Ethereum blockchain via Chainlink. Please find the following details for each step:

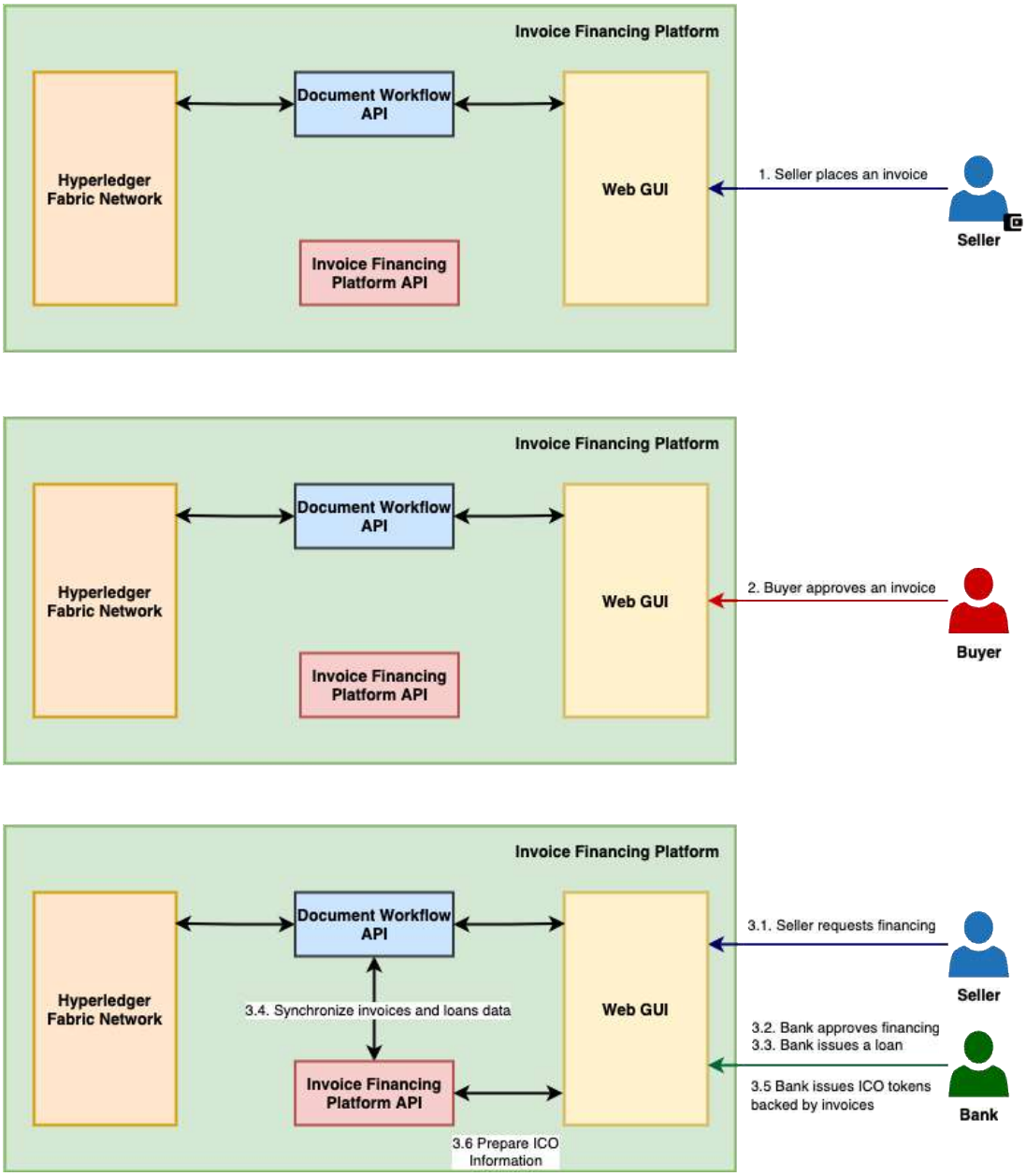

Figure 5: A simplified diagram illustrates how a seller, a buyer and a bank interact on the invoice finance platform.

- Issuing invoice (Step 1): A seller and a buyer must finish a transaction between them (A business between the seller and the buyer is excluded from the process). Afterwards, the seller issues an invoice to the invoice financing platform with a bound relationship with the buyer.

- Accepting invoice (Step 2): The buyer can approve or reject the invoice that is issued by the seller. In case of rejection, the process ends. On the other hand, the buyer guarantees the invoice, which means the buyer purchases the invoice. The buyer updates the invoice status to approved.
- Invoice financing (Step 3): The seller wants money immediately, but he does not know how to get them. Fortunately, he finds that banks can help with this. Banks require the invoice for making a loan. So, the seller submits the invoice and a request for a loan altogether to banks for performing invoice financing. Also, the loan is created by banks after the request is approved. Banks consider distributing an invoice to retail investors by preparing synthetic digital token information by using loans and invoice data for performing ICO issuance later on. Before finishing the ICO issuance, all of the loans must be issued completely.

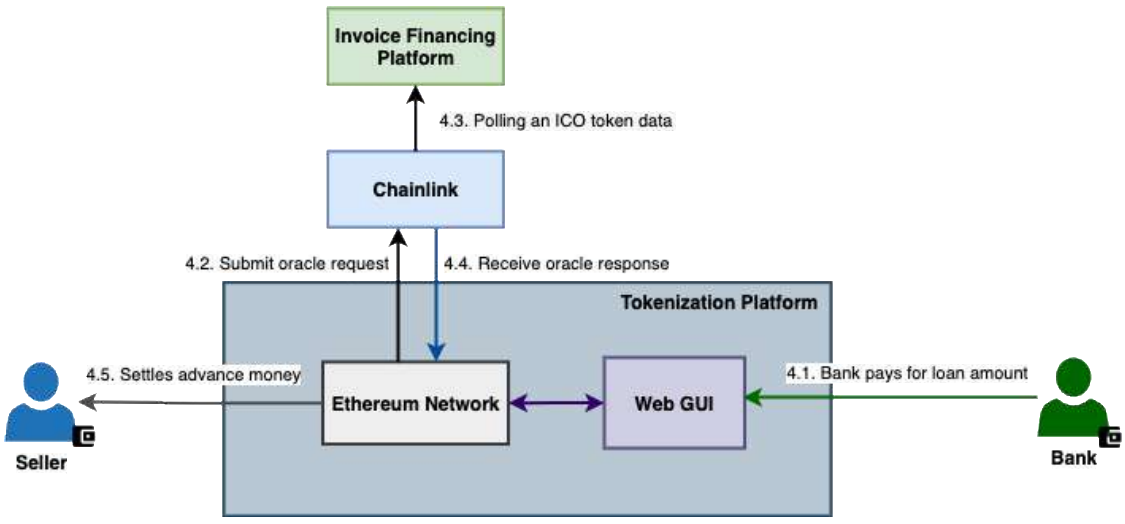

Figure 6: A sequence diagram illustrates how Chainlink participates in Step 4 (Lending) of the POC.

- Lending (Step 4): This step is actually a part from the prior step (Step 3), but it occurs in the Ethereum blockchain. Please refer to Figure 6 for more details of

the interaction described here. This step starts with a mechanism of polling the data from Invoice Financing Platform for newly created tokens.

For the sake of simplicity, this polling mechanism has been replaced with manual interaction on the web-based graphical user interface (Web GUI) connected to the smart contracts. To be more exact, there are two sub steps involved; the smart contract will submit oracles requests to Chainlink during the first sub step. Due to the fact that there are a number of data fields for each token and Chainlink only supports one field of data per request. Hence there needs to be multiple oracle requests for a single token. This step is meant to import the data required to complete the whole process up until distribute a return to investors. Each oracle request will be processed by a Chainlink node operator by calling an API endpoint provided by Invoice Financing Platform to retrieve newly created token information. The callback function specified on each oracle request will be triggered to return the corresponding data field back to the smart contract. After all the oracle requests for a loan have been fulfilled, the second sub step will be triggered to complete the lending process of the loan by transferring the advance amount in ERC-20 Thai Baht Tokens from the bank's wallet to the seller's wallet.

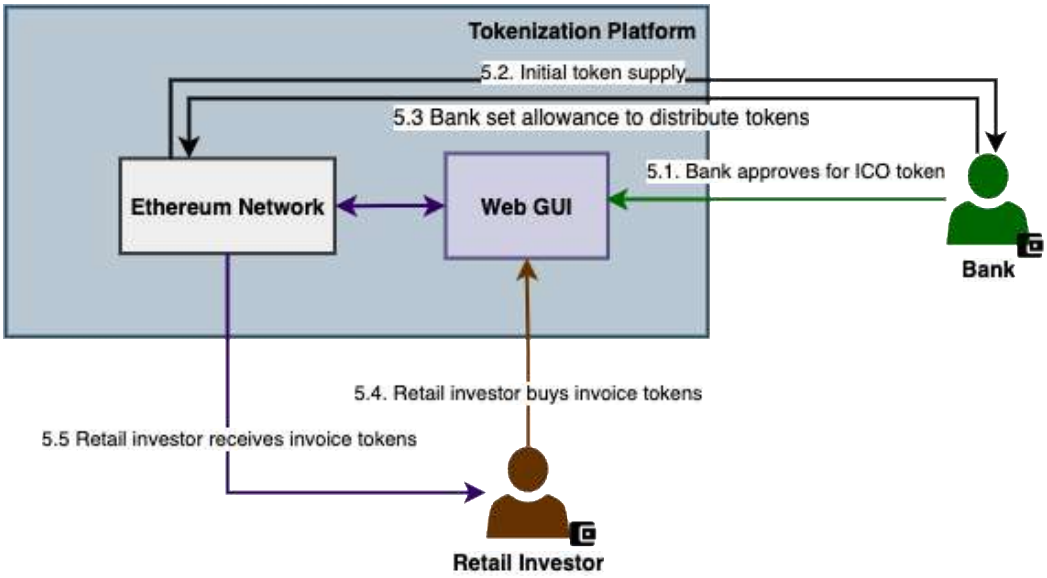

Figure 7: A sequence diagram illustrates in Step 5 (Tokenization and Sales) of the POC.

- Tokenization and Sales (Step 5) (Figure 7): After the lending process is completed, an ICO Portal will issue an ERC-20 token which represents one or more invoices. Please note that each invoice is always associated with a single loan offered to a particular seller. When the smart contract is triggered by ICO Portal to approve a token imported in the prior step, a new ERC-20 token will be created on a new Ethereum address. Initially the total supply of the newly created ERC-20 token will be distributed to the bank wallet associated with each loan of the token. Afterwards, retail investors can purchase the tokens using their THB token. The exchange rate between the token and THB token is retrieved from Invoice Financing Platform API during step 4. According to the fact that the token is ERC-20, retail investors can send and receive the tokens using a software Ethereum wallet such as Metamask.
- Distribute a return to investors (Step 6) (Figure 8): On the due date of the invoice, the status of the invoice will be changed to ready for collection. The exact same polling mechanism used in step 4 will keep track of such invoices in

order to trigger the invoice collection causing THB tokens to be transferred from the buyer's wallet to the bank's wallet. However, as in step 4, the polling mechanism is skipped in this POC for the sake of simplicity. Subsequently, ICO Portal can trigger the smart contract to distribute a return to all the investors holding the tokens. Please note that this distribution is triggered once per a pair of token and invoice.

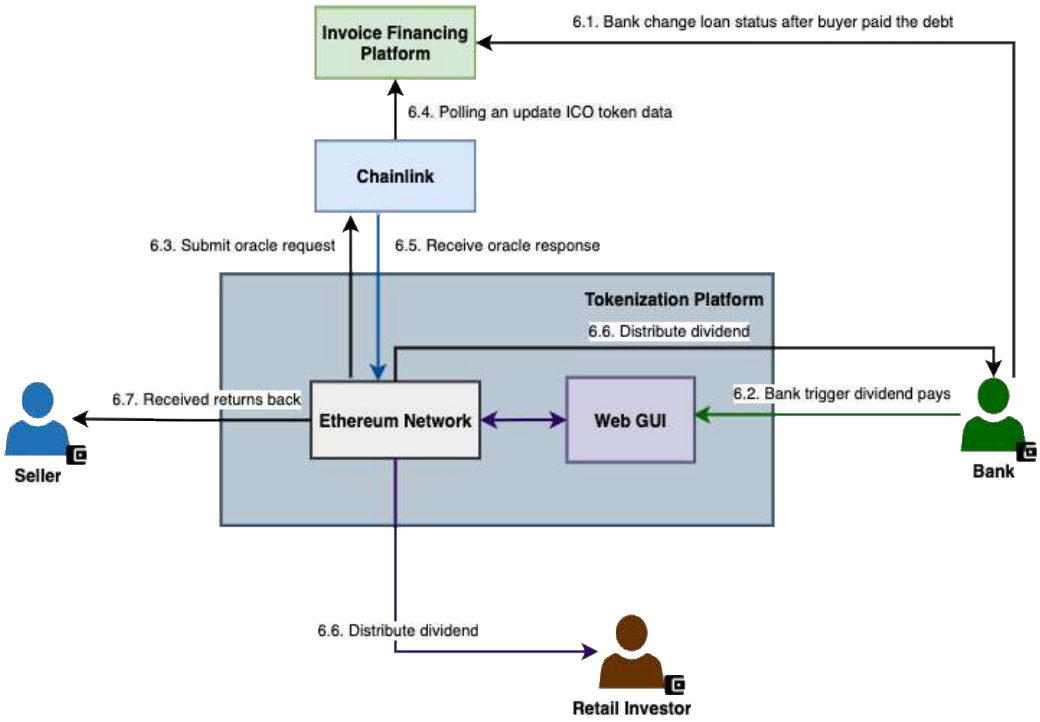

Figure 8: A sequence diagram illustrates in Step 6 (Distribute returns) of the POC.

In conclusion, the usage of Chainlink illustrated in the workflow is focused on importing the data from Hyperledger Fabric into Ethereum blockchain in an asynchronous manner. This is due to the fact that there is no need for such atomic operations across different blockchains in this particular use case. It is acceptable for the data on Hyperledger Fabric to be eventually synchronized with the Ethereum blockchain.

Findings

Based on our experience so far with Chainlink, there are a couple of points worth discussing. First, the current limitation of Chainlink is that it supports only a single-word response to be fed through Chainlink Oracle. This leads to multiple Chainlink requests per token in order to synchronize all the required data fields. Second, in this case, there is a single source of data which is provided by Invoice Financing Platform APIs, the value of LINK token was set to zero in this POC. In other words, the decentralization aspect of Chainlink has not been utilized in this POC, but one might find it beneficial to a consortium environment where several parties provide the same set of data.

BUSINESS USAGE

Value Added

The technologies discussed in this paper allow the integration of different blockchain technologies combined to deliver an application that spans across different networks. Businesses can therefore extend, rather than replace the investments they previously made on blockchain technologies that fit a particular use-case. This is advantageous not only to leverage existing networks and blockchains, but also as a guarantee that future investments will be able to connect to newer technologies as they become available. Furthermore, the key business value of this POC is to illustrate that the existing implementation on Hyperledger Fabric can be integrated well with Ethereum blockchain. However, atomic composability of the smart contracts on different chains needs to be sacrificed. Depending on the use case, it might be acceptable or not. Considering the cost of the implementation of the Chainlink integration, it might be worth considering rewriting the existing chaincode in Ethereum smart contracts; nevertheless, some specific features which are exclusive on Hyperledger Fabric must be sacrificed. Rewriting everything is not always the option; the cost of redevelopment of the existing code might be far greater than the cost of integrating the blockchains.

Production Ready Solutions and Areas for Improvement

The POC implemented as part of this research is meant to illustrate a possible solution for the integration between Hyperledger Fabric and Ethereum. Nevertheless, a number of trade-offs have been made between simplicity and completeness. The followings are suggestions for improvement:

- The polling mechanism discussed earlier has been replaced with manual interaction on the Web GUI with Metamask connected to the Ethereum blockchain instead. This is definitely required to complete the production-ready integration in order to seamlessly import the data from Hyperledger Fabric into Ethereum.
- Security is another key aspect which is a must to be implemented carefully. A great example would be to implement role-based access control in Solidity; this allows us to control who can participate in each step of the workflow.
- There is only one source of external data in this POC. When there is more than one party that provides the data. Thus, some sort of aggregation strategy must be implemented. This will further utilize the decentralization aspect of Chainlink, especially in a consortium environment.
- To overcome the limitations of single-word response on Chainlink oracles, the multi-word response feature scheduled to be released in the upcoming Chainlink version must be incorporated into the POC. This will significantly reduce the number of Chainlink requests required to import each token from Hyperledger Fabric to Ethereum.

- For this POC, each approved invoice financing token is turned into an ERC-20 token. In other words, a new ERC-20 token is deployed every time a new token is approved by ICO Portal. This is not just a trivial technical detail, but it will impact user experience where retail investors need to trade each token representing one or more invoices separately. In addition to this, regulatory aspects should be considered. As under Thailand regulations, an investment token must apply for and receive an approval from the Securities and Exchange Commission. Therefore, tokenizing a token for a group of invoice might be more practical than tokenize a single invoice. Some other vehicles also should be considered.
- In addition to the above, if there is a need to create a secondary market to trade tokenized invoices after their issuance, the amount of liquidity available should be considered as well. In low-liquidity scenarios, automated market makers such as liquidity pools could be used to enable buying and selling without an order book. In these scenarios, the liquidity token of the underlying pool could become tokenized as well, in particular when using pools that allow trading of more than one token pair.
- Exploring the space of technical and legal solutions to the problems outlined above could become the topic of subsequent work to advance Kasikornbank's objectives.

Last but not least, to be able to maintain blockchain-based applications on production, it requires the providers such as ConsenSys which offer the production-ready 24x7 support for these technologies and the companies like Atato are able to develop the necessary integration and infrastructure. This overall gives satisfactory guarantees to businesses in the

ability to build these solutions, which are now production ready.

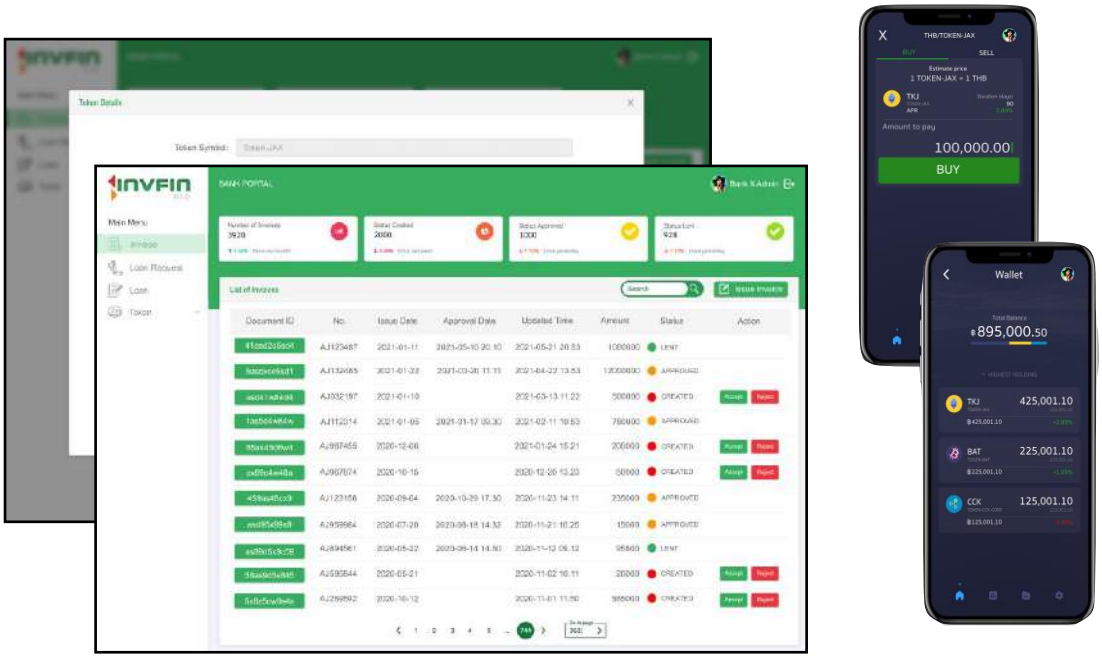

Figure 9: GUI for the invoice financing platform and the investor mobile wallet

CONCLUSION

This research at KBTG and Atato is aimed to demonstrate that existing implementations of Hyperledger Fabric can be integrated with Enterprise Ethereum blockchain without too much effort required. As discussed in this paper, some trade-offs must be made to achieve such cross-chain communication. However, we believe that the benefits are far greater than the cost of such implementation as illustrated in the POC. The ultimate goal here is to minimize the cost of rewriting the same logic of underlying code in a different blockchain platform and focus on achieving interoperability between them. Eventually, the technologies used to implement this POC, such as Solidity, Metamask, and

Chainlink have been widely used in the DeFi space. The work we have done in this research makes us foresee the potential of integrating traditional financial products through tokenization which is one of the steps to enhance the traditional finance world to DeFi.

OUR PARTNERSHIPS

About KBTG

At KBTG, we never cease to develop and innovate financial technologies on top of our “Customer First” mindset. We are the driving force behind KBank’s success as well as their navigator exploring the digital world beyond Thailand. Our wide-ranging online banking services are equipped with concrete infrastructure and strong barriers capable of guarding customers’ valuable assets and data. Utilizing our expertise in FinTech business, combined with new generation’s outside-the-box thinking, KBTG aims to transform into the best tech organization of Southeast Asia.

Apart from developing financial solutions for KBank’s customers, KBTG is ready to help other companies conceive new innovations and bridge the gaps through a process called Co-Innovation Runway. With KBTG as a partner, together we will work on the conundrum from start to finish, until we achieve the final product. We will answer the business' need with our expertise in technology and banking, creating innovations and services that benefit us and most importantly, our customers in every step of their lives.

KBTG has been pioneering blockchain infrastructure for large-scaled cross enterprise usages. We are also aiming for developing digital assets ecosystem that will bring a new type of asset class for Thai people, visit https://www.kbtg.tech/

About Atato

Atato is a digital assets technology company that specializes in developing blockchain platforms and infrastructure.
Since 2017, atato has worked with major financial institutions and digital assets businesses to deliver innovative blockchain solutions.
All our blockchain development is based on Ethereum, and as a result, we developed a strong partnership with ConsenSys.
With this research paper partnership, atato wishes to showcase Ethereum's extensive tools and products ecosystem for the next generation of financial platforms. (Learn more on www.atato.com)

About ConsenSys

ConsenSys is the leading Ethereum software company. ConsenSys enables developers, enterprises, and people worldwide to build next-generation applications, launch modern financial infrastructure, and access the decentralized web. ConsenSys' product suite composed of Infura, Quorum, Codefi, MetaMask, Truffle and Diligence, serves millions of users, supports billions of blockchain-based queries for our clients, and has handled billions of dollars in digital assets. Ethereum is the largest programmable blockchain in the world, leading in business adoption, developer community, and DeFi activity. On this trusted, open source foundation, ConsenSys is building the digital economy of tomorrow. To explore ConsenSys products and solutions, visit http://consensys.net/.

The tools and solutions such as ConsenSys Quorum, MetaMask and Truffle have been used to build the solution that is open, efficient and secure. A software suite that is worth mentioning and is provided by ConsenSys is Codefi. Codefi is utilizing blockchain technology to optimize business processes and payments, digitize financial instruments, and build customized decentralized applications. It provides an Operating System composed of APIs, user experiences and private label platforms that help people take advantage of new blockchain innovation. Moreover, it specifically provides wide-ranging and flexible APIs to solve the pain points of most Enterprise use cases and accelerate the automation of business processes across industries on private Ethereum networks or Ethereum mainnet. Since 2014, ConsenSys has tokenized billions of dollars in digital assets, including a wide range of consumer products, stablecoins, real estate, and financial instruments, and ConsenSys software and infrastructure has powered tens of billions of dollars in blockchain-based transactions.

This paper and the research behind it would not have been possible without the exceptional support of ConsenSys . They have provided invaluable guidance throughout this research. We would like to thank them for their great support and expertise.

GLOSSARY

Term	Definition
API	An application programming interface (API) is a computing interface that defines interactions between multiple software intermediaries.
DeFi	Decentralized finance is an experimental form of finance that does not rely on central financial intermediaries such as brokerages, exchanges, or banks to offer traditional financial instruments, and instead utilizes smart contracts on blockchains, the most common being Ethereum.
ERC-20	The ERC-20 introduces a standard for Fungible Tokens, in other words, they have a property that makes each Token be exactly the same (in type and value) of another Token.
FinTech	Financial technology (abbreviated fintech or FinTech) is the technology and innovation that aims to compete with traditional financial methods in the delivery of financial services.
GUI	The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.
ICO	An initial coin offering or initial currency offering is a type of funding using cryptocurrencies. It is often a form of crowdfunding; however a private ICO which does not seek public investment is also possible.
KYC	The know your customer or know your client (KYC) guidelines in financial services require that professionals make an effort to verify the identity, suitability, and risks involved with maintaining a business relationship.
LINK	LINK is an ERC-20 token used in Chainlink network.
Metamask	MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain.
POC	Proof of concept is a realization of a certain method or idea in order to demonstrate its feasibility
Shard	A database shard, or simply a shard, is a horizontal partition of data in a database or search engine. Each shard is held on a separate database server instance, to spread load.
Solidity	Solidity is an object-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various blockchain platforms, most notably, Ethereum.

Source: https://en.wikipedia.org/

BIBLIOGRAPHY

- Agarwal, Akarsh. “Understanding the Node-SDK of Hyperledger and Running the First Example.” *Medium*, 7 October 2017, https://medium.com/mlg-blockchain-consulting/understanding-the-node-sdk-of-hyperledger-and-running-the-first-example-5f6753393ec8. Accessed 9 February 2021.
- Belchior, Rafael, et al. “A Survey on Blockchain Interoperability: Past, Present, and Future Trends.” *arXiv*, 29 August 2020, https://arxiv.org/abs/2005.14282. Accessed 9 February 2021.
- Clearmatics. “Ion Stage 2 Ethereum-Fabric Interoperability PoC.” *Github*, 30 January 2019, https://github.com/clearmatics/ion/wiki/Fabric-Architecture. Accessed 9 February 2021.
- Clearmatics. “Ion Stage 2: Toward a General Interoperability Protocol (Part 1).” *Medium*, 30 August 2018, https://medium.com/clearmatics/ion-stage-2-toward-a-general-interoperability-protocol-part-1-d12b9d7316d3. Accessed 10 February 2021.
- Das, Sourav, et al. “Efficient Cross-Shard Transaction Execution in Sharded Blockchains.” *arXiv*, 29 January 2021, https://arxiv.org/abs/2007.14521. Accessed 10 February 2021.
- Ellis, Steve, et al. “ChainLink: A Decentralized Oracle Network.” *Chainlink*, 2017, https://link.smartcontract.com/whitepaper. Accessed 8 February 2021.
- Frankenfield, Jake. “Initial Coin Offering (ICO).” *Investopedia*, 3 November 2020, https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp. Accessed 11 February 2021.
- Hyperledger Besu. “Besu Enterprise Ethereum Client.” Hyperledger Besu, 24 December 2020, https://besu.hyperledger.org/en/stable/. Accessed 16 February 2021.
- Hyperledger Cactus. “Hyperledger Cactus Whitepaper.” Github, 2021, https://github.com/hyperledger/cactus/blob/main/whitepaper/whitepaper.md. Accessed 17 February 2021.
- Koens, Tommy. “Atomic Swaps for Distributed Ledgers.” Medium, 29 October 2019, https://medium.com/ing-blog/atomic-swaps-for-distributed-ledgers-cacfb7e1d90. Accessed 19 February 2021.
- MatZand. “Overview of Advance Ethereum concepts of Oracle, off-the-chain data, PoS and TPS.” Hackernoon, 13 August 2019, https://hackernoon.com/overview-of-advance-ethereum-concepts-of-oracle-off-the-chain-data-pos-and-tps-e0sk2dc4. Accessed 9 February 2021.
- Polites, Jared. “Protocol Alternatives That Challenge Ethereum.” Hackernoon, 29 January 2021, https://hackernoon.com/protocol-alternatives-that-challenge-ethereum-wl8731g1. Accessed 9 February 2021.
- Sharon, Alita. “Thailand SEC greenlights first ICO portal.” OpenGov Asia, 27 March 2019, https://opengovasia.com/thailand-sec-greenlights-first-ico-portal/. Accessed 11 February 2021.
- Velotrade. “What is Invoice Financing?” Velotrade, 2021, https://www.velotrade.com/guides/what-is-invoice-financing/. Accessed 11 February 2021.
- Zhang, Jim. “How Chainlink Works Under the Covers.” Kaleido, 12 February 2020, https://www.kaleido.io/blockchain-blog/how-chainlink-works-under-the-covers. Accessed 19 February 2021.
- Zhou, Qi. “Solving Train-And-Hotel Problem with Asynchronous Message (and await/async).” Ethereum Research, 1 July 2019, https://ethresear.ch/t/solving-train-and-hotel-problem-with-asynchronous-message-and-await-async/5836. Accessed 10 February 2021.
- Hyperledger Fabric. “Hyperledger Fabric - Hyperledger Confluence” Hyperledger Confluence, 2021, https://wiki.hyperledger.org/display/fabric. Accessed 4 Mar 2021.
- Hyperledger Fabric. “Glossary” Hyperledger Fabric Read the Docs, 2021, https://hyperledger-fabric.readthedocs.io/en/latest/glossary.html. Accessed 3 Mar 2021.