

NAGA: multi-blockchain based decentralized platform architecture for cryptocurrency payment

Dendej Sawarnkatat, Sucha Smanchat

Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

Article Info

Article history:

Received Jan 30, 2021

Revised Dec 17, 2021

Accepted Jan 5, 2022

Keywords:

Blockchain

Cross-cryptocurrency

Cryptocurrency

Decentralized finance

Multi-blockchain

Smart contract

ABSTRACT

Paying for electronic commerce products with cryptocurrencies is an increasingly popular method. However, the situations where a seller expects one specific cryptocurrency as a payment while a buyer only possesses another, inevitably create inconvenience to the buyer, which may lead to cancellation of purchase. In this paper, we propose a light-weighted software architecture of a payment system called NAGA platform that works with a number of crypto blockchain networks to support cross-cryptocurrency payments where buyers can pay in one currency, and the seller automatically receives another of his/her choice. This minimizes the complexity and inconvenience of the buyer, leading to an increase in sales and revenues of the electronic commerce system. With the built-in crypto exchange, cross-cryptocurrency payment can be processed with real-time exchange rates that enables both buyers and sellers to receive cryptocurrency of their preferred choices.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Dendej Sawarnkatat

Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok

1518 Pracharat 1 Rd, Wongsawang, Bangsue, Bangkok, 11000, Thailand

Email: s5807011910038@email.kmutnb.ac.th

1. INTRODUCTION

Nowadays, e-commerce activities such as buying products and services through online stores are becoming increasingly popular with people of all genders and ages around the world. This has resulted in an increasing number of e-commerce providers, leading to various forms of competition to attract new customers or to retain their old customer base as long as possible. The same key strategy that most service providers emphasize is to provide their customers with the most satisfactory experience possible. One of the key factors in the shopping experience is the convenience of using the system, particularly in the payment options. Although there is a wide selection of good quality products with a reasonable price that are available on a particular platform, if the customer is somehow unable to pay for it, then the sale would not occur. Besides, the downside is that the customer may stop shopping with that service provider and switch to another provider that supports the type of payment that they want.

One of the most popular new payment methods offered by online merchants is cryptocurrency payments. These currencies are considered as alternative investment assets with high yields and are also convenient to spend because they are highly secured and focus on privacy. While there are many advantages to using cryptocurrency, there are still obstacles in the way of using it as a means of payment. For example, there are many of cryptocurrencies today. In the event that some customers hold some cryptocurrencies that are not accepted, it will result in more complex transactions. It may result in the need to sign up for a new

account with another crypto exchange that supports the required ones. In the near future, the problem is likely to escalate, so any online shopping service that can solve such a problem, will increase the chances of survival in the competition by gaining an advantage over other competitors.

The online payment methods of e-commerce have evolved steadily since the advent of the internet, initially being provided by financial service providers with a centralized architecture such as credit card and fund transfer. Those services are subject to relatively high fees and regulations. As a result, efforts have been made to develop a non-intermediate and privacy-focused payment system using cryptography technology but unfortunately, these services were unsuccessful due to technological inadequacy and failing to show clear benefits to users. Eventually, the emergence of the cryptocurrency, led by Bitcoin [1], sparked an unprecedented widespread popularity of this kind of asset, causing a huge impact on the financial industry. As a result, many cryptocurrencies such as Litecoin [2] and Dash [3] have emerged to support the demand for these new digital assets.

In addition, the major reason behind the popularity of cryptocurrencies is the enabling technology known as blockchain, which has been researched and documented on academic journals that target various industries such as supply chain management [4], [5], logistics [6], [7], finance [8], [9], energy [10], [11], entertainment [12], and medicine [13]. But due to the limitations of the early generation of blockchain and the lack of powerful development tools, its adoption is difficult and complex, leading to the invention of the next generation of blockchain, such as Ethereum [14], which, in addition to being able to operate as a cryptocurrency, also supported the execution of arbitrary code embedded on the blockchain called “smart contract”. As the development of cryptocurrency and blockchain has grown exponentially, there are thousands of cryptocurrencies and blockchain platforms out there today. This leads to the current situation, where the problem is how to enable those platforms to seamlessly communicate with each other, and to make cross-cryptocurrency crypto payments easier.

As a result, various platforms have been invented to support communication and interoperability among heterogeneous networks of different blockchains. Some interesting projects that have high potential and are gaining attention at the time of writing are cosmos network [15] and polkadot [16]. Firstly, cosmos network is powered by tendermint [17] consensus engine, which is implemented on byzantine fault tolerant (BFT) algorithm [18], [19]. It consists of hubs and zones. Each zone, which refers to a joining blockchain network, connects to only a hub through inter-blockchain communication (IBC) protocol while a hub-to-hub connection also happens in similar fashion. The platform is highly scalable since there is no limit on the number of hubs or zones. Any communication among connected zones, such as messaging passing and valuable asset transferring, can pass through a single or multiple hub as needed. Each zone can be developed with cosmos software development kit (SDK) and also has its own sovereignty with specific security model. Besides, a specialized zone called “Peg zone” can be created specifically for connecting with non-compliance or legacy blockchain networks such as Bitcoin or Ethereum.

Next, “polkadot” employs consensus techniques, which consists of two protocols [20]: i) block producing protocol called “BABE” and ii) chain finality protocol called “GRANDPA” that allows high throughput of transaction processing. Platform-wise, it embraces a sharded model on interconnection of many blockchain networks where each network is called “shard”. By default, a shard connects to “relay chain” which is the main singleton blockchain network that provides interoperability among shards such as communication and a shared security model. Any joining network must comply with some specific requirement regarding consensus method, and subjects to governing by the platform security policy. Since the connection to the relay chain is controlled by some auction mechanism, it is not as highly scalable as the cosmos network. Note that a custom bridge shard can be built to connect with non-compliance blockchain networks such as bitcoin and ethereum as well. To summarize, developing applications on those platforms require sophisticated tools and knowledgeable personnel. In addition, the implementation of any of those systems is complex in the event that it is used privately and, most importantly, does not support automatic exchange of cryptocurrencies. In this paper, we propose an architecture that focuses on solving the inconvenient problem of cross-cryptocurrency payments with a direct automatic exchange capability under an uncomplicated and easy to develop solution for small and medium e-commerce services that do not require super-fast massive processing speed similar to those well-known stock and foreign exchanges, and ultimately do not want to incur huge investment in resources to get started.

2. RESEARCH METHOD

Paying with cryptocurrencies through the proposed platforms relies on mechanisms that support both single and cross-cryptocurrency payments. The single currency payment, which relies on only a few components of the NAGA platform, has a 3-step sequence of operations: i) payment initialization, ii) payment execution, and iii) payment finalization. In contrast, the cross-cryptocurrency payment flow is more complex because the platform has to work with two crypto networks of the respective currencies, so

there are 5 steps: i) payment initialization, ii) request messaging, iii) payment execution, iv) respond messaging, and v) payment finalization.

- Payment initialization: a buyer (user) makes a checkout request on an e-commerce provider system. Upon receiving such a request, the provider sends the quotation to a corresponding component of the NAGA platform that connects to the provider's cryptocurrency blockchain network. The NAGA platform records the quotation on the smart contract and then generates the inter-blockchain transactional message destined to the buyer's preferred cryptocurrency network. The inter-blockchain message gets serialized and gets routed to the destination network.
- Request messaging: the serialized message passes through the inner messaging router and both inward and outward messaging bridges until it reaches the final destination. Upon arrival at the destination, it gets deserialized back into quotation so that it can be processed.
- Payment execution: the subcomponent that is responsible for handling the payment creates and records a new invoice on the corresponding smart contract and then sends it as a part of the notification to the buyer (user) for payment collecting. The notified buyer makes a payment in his currency to the platform where, in turn, the whole payment gets traded (or swapped) into the provider's preferred currency on the crypto exchange operated by the NAGA platform itself. Note that the NAGA platform may deduct some small fee from each payment prior to the currency exchange. After the currency exchange processing, the crypto exchange also deposits the converted net amount into the provider account on the designated escrow on the native currency blockchain network. Finally, the platform issues a proof of purchase that is sent back to the buyer and is combined with other essential information to compose a new acknowledgement, which is a serialized responding or resulting message that is sent back to the provider (seller).
- Respond messaging: similar to the request messaging stage, the responding message passes through the same yet reversed path back to the originating cryptocurrency network. As the message arrives, it is deserialized and gets delivered for finalization.
- Payment finalization: the arriving message triggers the payment finalization through an updating of the relevant quotation record on the smart contract and sending a new alert, which contains information regarding a deposited fund and a copy of proof of purchase, back to notify the e-commerce provider of payment completion. At this point, the provider may proceed with its fund withdrawal. Be advised, since the single currency payment does not require both outgoing and incoming messaging and currency exchange operations, therefore, the request messaging and response messaging steps are irrelevant.

2.1. Design abstraction and concepts

2.1.1. NAGA specialized client software

This novel software performs extra functionalities to allow the communication between two different blockchain networks. Such a system can act as an additional specialized software (client) application that runs on top of existing blockchain peer nodes of each particular cryptocurrency network. The NAGA client application must work in conjunction with native blockchain client software to make a cross-cryptocurrency payment possible.

2.1.2. Blockchain overlay network (BON)

On each blockchain network of supported cryptocurrency, there must exist a group of special peers that runs not only normal client software but also NAGA specialized client software. Those peers also connect to each other forming an overlay network of special peers. This network supports all extra functionalities provided by the NAGA platform. Since such a group represents a "circle" of tightly connected peers, for simplicity, this group or circle is called a "ring" to represent the existence of a special relationship among those peers. From the concepts and abstracts that are used in the architectural design of the NAGA platform, hereinafter is a description of the key components and internal structure of each. The main components consist of 2 parts: NAGA Core and NAGA Extension as shown in Figure 1 and more details in Figure 2.

2.2. NAGA core system

This is the main central system that administers all aspects of core functionalities of the whole platform. Some notable functionalities are: i) handling cross-blockchain (currency) transaction routing via a specialized BON-based transaction router, ii) providing decentralized crypto exchange that allows buying and selling of all supported currencies across platforms, and iii) issuing proof of purchase as the evidence of successful transactions. The core system comprises of various vital sub-systems as described in the sub-section 2.2.1 to 2.2.3.

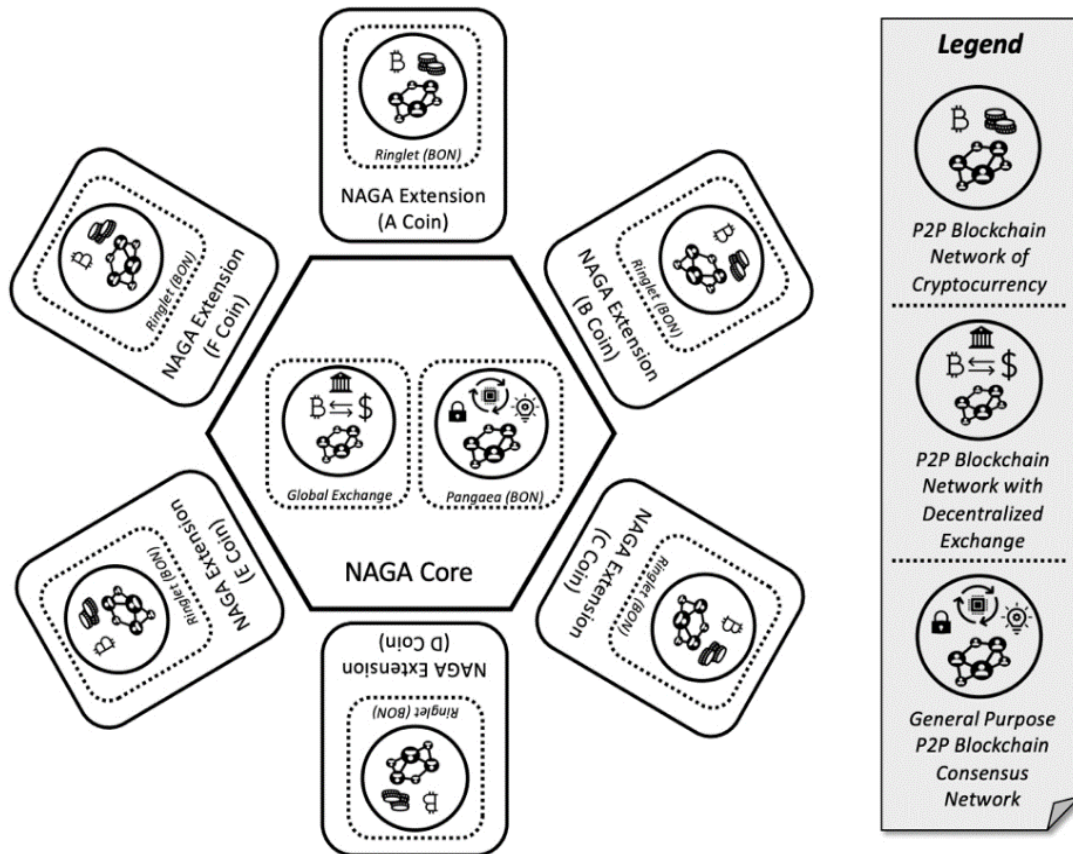


Figure 1. Multiple NAGA extension instances connecting to NAGA core system

2.2.1. Global crypto exchange (GEX)

This component simulates inter-ring (blockchain) rebalancing of all supported currencies on its market trading boards. By design, it consists of two main parts which are firstly, decentralized crypto exchange and, secondly, BON that simulates a publicly accessible blockchain network. In other words, it is a semi-decentralized crypto exchange that provides a non-native cryptocurrency trading market facility on a (providing) blockchain network with its own native currency. The implementation of this exchange is inspired from a simplified version of bristol stock exchange (BSE) [21] where only two general trading strategies are selectively supported which are: limit order and market order trading. Any enthusiast can trade any supported foreign cryptocurrencies on this network through their corresponding proxies. In fact, those proxies are actually ERC-20 standardized tokens [22] built on top of the smart contracts deployed on the providing network. There exchange related functionalities are divided into four vital groups as followed: i) order processing is responsible for handling orders by matching a pair or a group of buying and selling orders that can be settled which results in exchange of intended assets between all stakeholders; ii) liquidity pool management controls and monitors all available token supplies and calculates the exchange rate between any pair of assets upon and exchange or swapping request; iii) token management includes the token supply management and necessary information regarding token holders such wallet address and balance record keeping. Lastly, iv) account management refers to the administrative aspects regarding trading account such as account creation and maintenance as well as recording all currencies and tokens balance on the exchange.

The main purpose of this component is to provide liquidity support during token swapping for cross-cryptocurrency payment according to liquidity strategy selection. The default strategy is an atomic sequence of subtransactions consists of selling the base currency or token for NAGA native currency at the best price and then buying as much as target currency token at the lowest using that native amount. However, if there is not enough liquidity from atomic orderbook trading to support exchange requests, then the fallback strategy of liquidity pool is activated to provide the needed fund in order to fulfill such requests. The role and related operations of liquidity pool is governed by simplified model of constant product market making (CPMM) [23] found on most popular automated market making (AMM) [24] used by most decentralized finance services on various cryptocurrency networks.

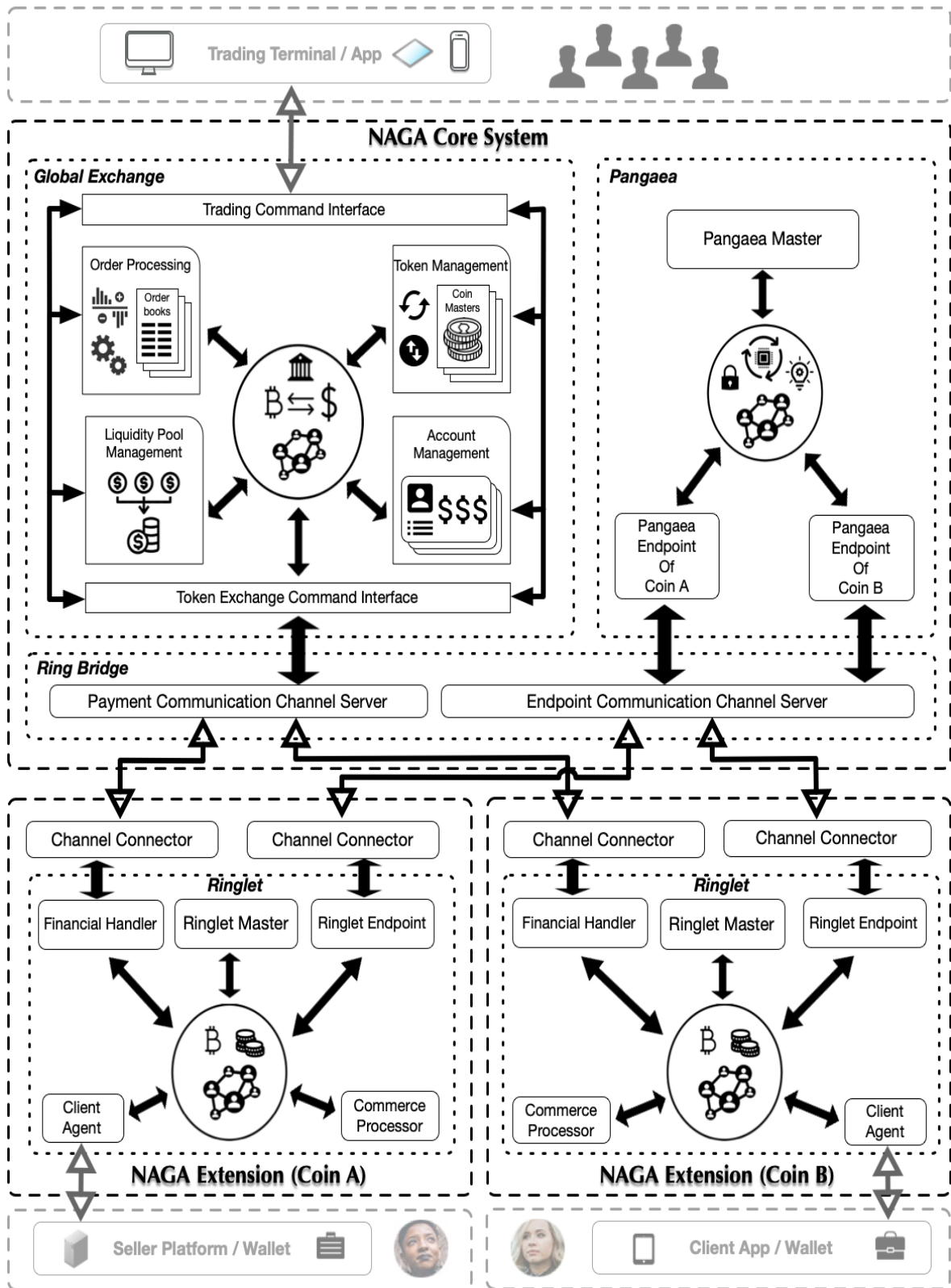


Figure 2. Detailed component layout and communication path

2.2.2. Cross-blockchain transaction router (pangaea)

This is a unique BON based connection hub that connects to each BON that has its own currency. It acts as the main central ring that provides the interoperation among various supported cryptocurrencies and

handles inter-BON traffic by performing cross-cryptocurrency payment message routing for all connected blockchain networks. In addition, it also manages all information regarding participating blockchain networks by maintaining a repository of registered currency (ringlet) that connects to the NAGA platform.

2.2.3. Ring bridge

By default, each ringlet connects to pangaea through this component. It provides bi-directional transactional communication between a connected ringlet and pangaea. Internally, ring bridge contains a message queuing system that powers two communication channels namely *payment communication channel server* and *endpoint communication channel server*, which facilitates connection between financial handler and global exchange and is used to connect pangaea endpoint with its corresponding ringlet endpoint respectively. The communication between financial handler and GEX enables the latest exchange quotes and making a cross-cryptocurrency automatic exchange transaction request between any two currencies while the connection between pangaea endpoint and ringlet endpoint enables cross-blockchain transaction transmitting and receiving.

2.3. NAGA extension

NAGA extension is considered an integral part of NAGA platform. It serves as an interface to the supported cryptocurrencies that are overlaid on the cryptocurrency's blockchain network. The NAGA Extension is made up of two key components: ringlet and channel connector.

2.3.1. Ringlet

This component is a BON that gets created specifically on top of the target cryptocurrency network. Each of the supported cryptocurrencies has its Ringlet connected to Pangaea. By default, it must contain all necessary specialized member nodes namely ringlet master, payment handler, commerce processor, and client agent.

- Ringlet master: it is an administrative agent of a ringlet designated to manage the life cycle operations of all member peer nodes.
- Commerce processor: it provides quotation request handling service for both single-currency and cross-cryptocurrency payment by storing all quotation record history on the blockchain, and by coordinating with the local financial handler regarding payment activities of all requests.
- Financial handler: there is only one for each ringlet. It is responsible for processing payment by providing three vital services, which are: i) storing all historical payment records on the smart contract, ii) enabling deposit and withdrawal services to all clients, and iii) notifying ringlet-wide regarding all monetary related events. In other words, it acts as a bank that manages the cryptocurrency account of all users on that ringlet including activity tracking.
- Client agent: each instance is a platform delegate of the owner client. It provides important functions such as making a deposit to and withdrawing a payment from financial handler and creating a quotation request for payment collection.
- Pangaea endpoint: it resides on the pangaea and is responsible for registering its corresponding ringlet. Basically, what it does are: i) to receive requesting messages from the originating ringlet through pangaea and then to forward them to the destination ringlet and ii) to receive replying messages from the destination ringlet and to route them via pangaea back to the originating ringlets.
- Ringlet endpoint: it resides on each ringlet and acts as a messaging gateway for that ringlet. It serializes all outgoing messages to pangaea and deserializes all incoming messages from pangaea.

2.3.2. Channel connector

It acts as connection client to communication server, which allows the bi-directional messaging capability between NAGA Extension and NAGA Core. During operation, two instances of channel connector are created where one connects to the endpoint communication channel server (ECCS) and the other connects to the payment communication channel server (PCCS). The former allows the ringlet endpoint to send and receive message to and from the corresponding Pangaea Endpoint. While, the latter enables the financial handler of the NAGA Extension to communicate with the global exchange during the crosscryptocurrency payment.

2.4. General usage

This section describes the usual scenario where the NAGA platform and an e-commerce platform are cooperating together in order to facilitate the exchange of merchandises and payments between buyers and sellers during business engagement. The complete flow of operations is depicted on Figure 3. The flow begins at step 1 when a buyer wants to buy from a merchant via an e-commerce platform that accepts crypto

payments but does not accept the currency that the buyer has, but the payment is still possible through the NAGA platform. In step 2 the merchant created a request to NAGA platform to use its cross-cryptocurrency payment service in the form of a blockchain-based quotation that charges the buyer in the currency of their choice. Next step 3, the quotation is then transmitted through the NAGA Core system for further routing.

At the NAGA Core in step 4, the routing process takes place to determine the route to the destination NAGA Extension overlaying the specified crypto currency blockchain network. Once the destination is known, the quotation is sent for processing at the NAGA Extension. Later step 5, after the quotation has been processed, the bill with the necessary details is sent to the buyer's app or browser connected to the crypto wallet. In step 6, once the buyer is acknowledged and accepts the bill, he or she can make payment in the selected crypto currency to the NAGA Extension. Next step 7A, after receiving the payment, the NAGA Extension verifies the validity and issues a proof of purchase with an electronic signature to the buyer. A certain amount may be deducted as a processing fee. Simultaneously step 7B, the NAGA Extension sends the successful payment results back to the source via a reverse route through the NAGA Core system. The results contain three pieces of information: i) the response with the result code, ii) a request to the NAGA Core to perform currency swapping from the received currency to the merchant's preference via an internal crypto exchange, and iii) the purchase proof copy for merchants to compare during the process of receiving the product or accessing that service.

At step 8, after the currency exchange is complete, payment will be sent to the NAGA extension that has issued that quotation with the first and second part of the results. Later step 9, the NAGA extension permanently stores the first part of the results on the blockchain together with that quotation and delivers the proof of purchase to the merchant who made the request. Once the merchant receives a proof of purchase at step 10, the merchant system stores the proof in a database to prepare it for the buyer to access the system or deliver the product when the buyer shows the same purchase proof. Finally, at step 11, the purchaser accesses the system by submitting a proof of purchase as usage license or acceptance of the merchandise.

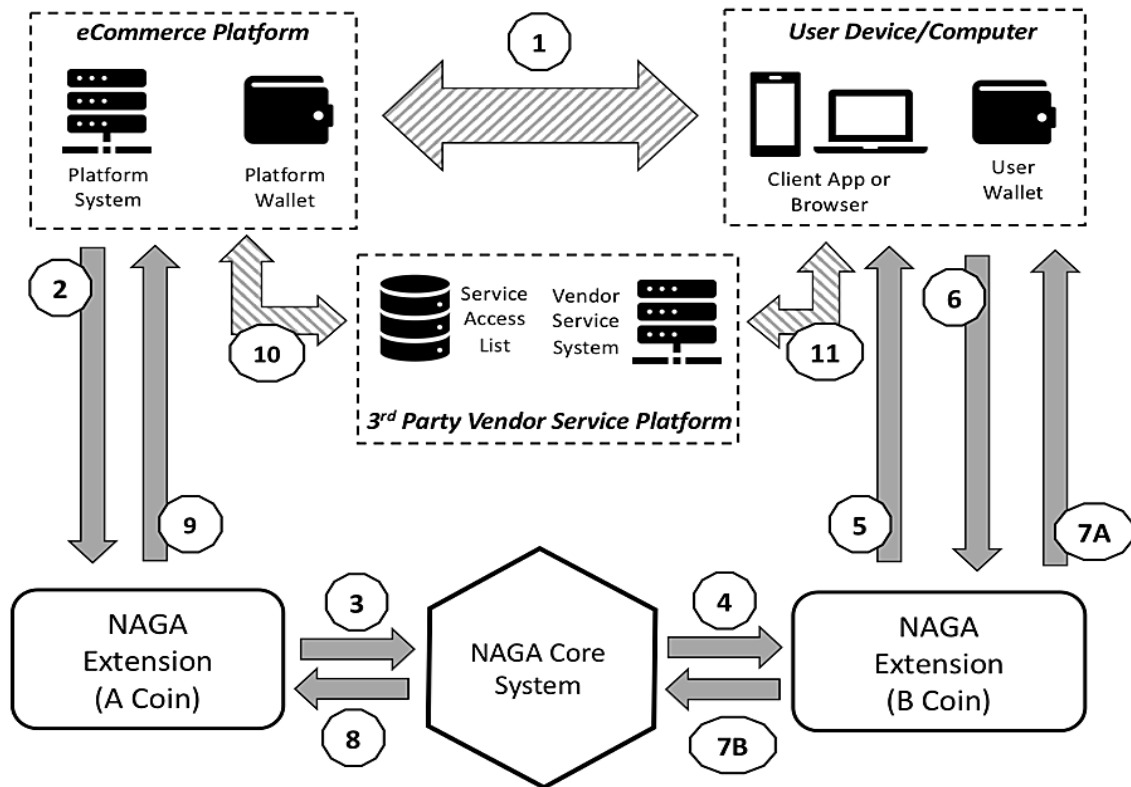


Figure 3. Payment flow diagram

2.5. Development planning

Developing the NAGA platform prototype is a complex endeavor that consists of two sub-projects: crypto application system development and smart contract delegate development for blockchain. The

mechanic of the whole NAGA Platform depends on related components working together, where the logic of the high-level operation is carried out by the crypto application system that controls the operation of the smart contracts which act as delegates that manage and execute the related low-level transactional operations within the blockchain. The following are brief description of crypto application system and smart contract delegate respectively.

2.5.1. Crypto application system

It is an application server that is implemented with a multi-layered, object-oriented software library suite that consists of four major components layering vertically one by one. Each upper layer relies on its adjacent lower layer to provide simple and fine-grain services while the upper layer composes those low-level services and abstracts them into high-level user-oriented functionalities.

- Base layer: this is the lowest layer which is the native blockchain client interfacing layer that enables low-level interaction with blockchain in most essential aspects. All of these layer's artefacts provide interaction with Ethereum blockchain and smart contract development. The functionalities that provided are: i) contract lifecycle management, ii) crypto wallet management, and iii) blockchain transaction management.
- Node layer: this layer is built on top of the base layer. Internally, it contains two vertical stacked sublayers of application programming interface (API) groups where the higher sublayer called "type-specific" lays on top of the lower sublayer of shared and supportive functional libraries called "Ringnode". The lower sublayer provides three functionalities as followed: i) node lifecycle management, ii) smart contract delegate management, and iii) supportive utilities. On the other hand, type-specific API library is a group of libraries that are used to create specific ringnodes namely: pangaea master, ringlet master, pangaea endpoint, ringlet endpoint, client agent, commerce processor, financial handler, coin master, and exchange system.
- Ring layer: this layer provides high-level ring-related administration API access to platform users in simple and cohesive fashion. It consists of two sublayers where the lower one provides basic, shared operations and supportive functionalities such as: i) ring lifecycle management, ii) member node management, iii) messaging system, and iv) blockchain network simulator while the upper provides type-specific functional for each ring components which are: pangaea, ringlet, global exchange, and ring bridge.
- Runtime layer: this layer consists of three different common components that are used as building blocks to create a set of NAGA platform specific application such as NAGA Core and NAGA extension. The API of this layers provides the administrative interface of the major components such as pangaea and ringlet with NAGA control and monitoring application. Using this application, the administrator of the platform can monitor and perform lifecycle management tasks of those major components as well as specialized operations such as Ringlet connection establishment and Ringlet disconnection operation.

2.5.2. Smart contract delegates

When designing and creating smart contracts, each contract is named after the ringnode that owns the contract. During execution, if interacting with the blockchain, ringnode is invoked through that contract method. The available smart contract delegates are pangaea master, ringlet master, pangaea endpoint, ringlet endpoint, client agent, commerce processor, financial handler, coin master, and exchange system.

3. RESULTS AND DISCUSSION

In researching the proposed architecture, we developed a prototype that uses an open source Ethereum blockchain network simulation called ganache system [25] instead of real cryptocurrency networks. It simulates the operation of the NAGA platform that supports a number of cryptocurrencies, each with its own blockchain network. The whole prototype development consists of two separated subprojects where the first one focuses on developing the main platform and its supplementary libraries and the other is about implementing embedded artifacts such as smart contract delegate and blockchain related functionalities. The main platform is implemented with JavaScript language in conjunction with Truffle development environment [26] to implement all smart contract programs. During execution of the prototype, all of the embedded smart contracts are deployed inside all running instances of Ganache Ethereum blockchain simulator while the main platform is executed on a server-side JavaScript engine.

To prove the operationality of the proposed architecture, a set of test scenarios is devised to address the most common of both failed and successful cases of the architecture. The test scenarios can be divided into 5 possible major cases which are: i) intra-ring successful payment, ii) intra-ring unsuccessful payment,

iii) inter-ring successful payment, iv) inter-ring unsuccessful payment due to originating ringlet error or communication error, and v) inter-ring unsuccessful payment due to destination ringlet error.

3.1. Setup and assumption

Prior to running any test cases, there requires the setup and configuration process, which in real world production would only need to be performed once. Nonetheless, for this experimental running of the prototype, the Mocha test suite is required to perform the following tasks such as simulating e-commerce Platform and client connection establishing, simulating decentralized exchange instantiation, and finally simulating all e-commerce transaction payment. Note that one important assumption on all test cases is that there are NAGA platform owned accounts called “escrow accounts” with sizable balance of local cryptocurrency on every Financial Handler on all Ringlets to support every currency swapping that might happen during execution of all test cases. Also, all payments are subjected to processing fee deduction by the NAGA Platform.

3.2. Case study results

The following paragraphs describes the experimental results obtained from the implementation of the prototype system to study the performance in the various case studies mentioned above. The results were divided into two groups based on their mode of operations which are intra-ringlet and inter-ringlet respectively. Both the successful and failure scenarios of each mode are elaborated and discussed.

3.2.1. Intra-ringlet tests

The first case (1) is an Intra-Ringlet (or single currency) payment (similar to traditional platform) in which the prototype has performed according to its design without any glitches. In the second case (2), the flow does not end, without payment settlement to test the ability to handle such behavior. The transaction processing results and error codes describing the reasons or obstacles preventing the payment processing from continuing to complete are notified to the originating party. In this case study, there are a wide variety of sub-cases but only a few scenarios are of particular interest because they are the most likely to occur. To begin with, considering testing a normal operation if the payment is canceled by the user, the system does not have any error and shows the result of the cancellation by the user for whatever reason, which happens frequently. Subsequently, it is a study by mimicking the situations in which some components of the NAGA platform that are either malfunctioning or failing to function, but the platform's error handling mechanism manages to allow the system to continue. These cases need to be studied thoroughly because of their widespread impact are events in which the Financial Handler, the main component responsible for payment management, fails to function. This is a very critical case as it leads to a situation where the entire system fails. As a result, payment services cannot be provided to all users. Note that, the difference between this case and the first one is that it consists of two alternative steps. The first one occurs during payment execution, as there are certain situations where payments cannot be processed. In the second one, during the payment finalization flow, the system informs the requester of the payment failure and the related causes that occur during payment execution. After running all related cases, the prototype is also able to manage all tests as expected.

3.2.2. Inter-ringlet tests

Here, the inter-ringlet payment testing is described, starting with the third case (3) of a normal operation with successful payment, with all steps shown in the Figure 4, in which the prototype system has successfully performed as designed without error. Subsequently, the following two cases are relevant to the payment failure. The testing also mimics situations where specific errors cause such failure to occur. In the fourth case (4) testing, the failure occurs during the requesting flow with two main reasons why the payment was unsuccessful, one of which is an error occurring in the originating ringlet and the other is due to the failure of communication channel between the originating ringlet and the destination (or remote) ringlet. The first reason is a failure caused by some critical components from the originating ringlet being unable to function due to a number of errors. However, the critical one that affects system-wide performance the most is caused by a failure of a component called Ringlet Endpoint, which is responsible for the communication from the inside of a Ringlet to the external or other parts of the NAGA platform and vice versa. Inevitably, the absence of Ringlet Endpoint causes the ringlet to be unable to communicate with the others. The second reason is not from inside the Originating Ringlet but can be caused by a number of factors. However, the situations that are most likely to occur are: the failure of the channel connector connected to the source Ringlet, and failure of the channel connector connected the destination ringlet.

The last case study (5) is quite similar to the second case since it takes place during the payment execution flow of which there are two sub-cases: i) the cancellation of the payment by the user, which is the most common occurrence and ii) the critical component error within the destination Ringlet. Elaborately,

there are two possible fatal errors in the second sub-case that greatly affect the overall system operation, one of which is the unavailability of Commerce Processor causing it to fail to process any transaction, and the second is the failure or absence of Financial Handler, which causes the inability to complete any financial related operations. Through testing a series of failed case studies, the NAGA system was able to address the problems as it was designed by responding back to those involved and recording the payment details and associated error codes on the blockchain which is transparent for further investigation.

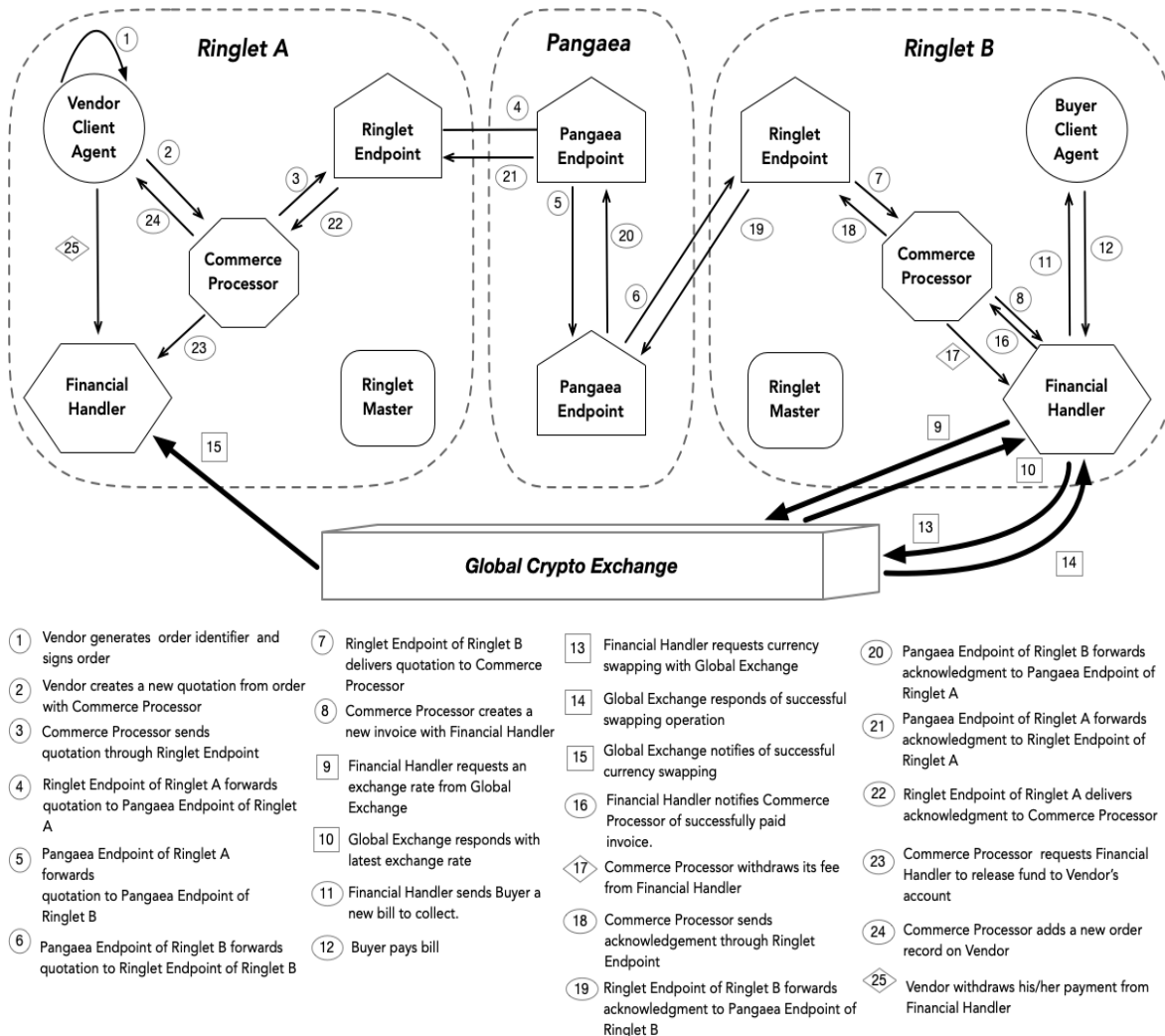


Figure 4. Inter-ringlet (cross-cryptocurrency) payment

4. CONCLUSION

This paper proposes a new architecture inspired from the effort to untangle the complexity and inconvenience caused by the adoption of traditional cryptocurrency payment especially in the situations where multiple currencies are involved. Development of any technologies that would simplify cross-cryptocurrency payment may, in turn, lead to more adoption of cryptocurrency as a mean of payment and could give any embracing e-commerce platforms a non-trivial edge or competitive advantage that may result in converting more sales and increasing revenues on such platforms.

Using liquidity for atomic order book trading would yield the best available exchange rate in real-time but there might be a situation where there is not enough supply on either buying or selling sides of order books to fulfill the swapping. When such a situation takes places, the liquidity pool swapping would be the best backup mechanism to facilitate such an operation since there have been study that most of decentralized exchanges with automated market making provide swapping service with exchange rate as closed as of those offered on any popular centralized exchanges.




The various scenarios on case studies show the feasibility and robustness of this architecture. However, this research still focuses solely on both facilitating the simpler payment method and the error handling mechanisms in high-probability situations, which did not cover other subcases that are less likely or difficult to predict. Since this research is not designed in cases where multiple errors occur simultaneously, the mechanisms involved may need to be further investigated to avoid the failure of the holistic operation.

In a further study from this research, a possible approach would be to increase the scope and capacity of error handling mechanisms to cover potential new scenarios, which required a more comprehensive design of the case studies to be thorough. In addition to the aforementioned error handling mechanism, currently the currency swapping process on the global exchange in the prototype is simulated in the way that there are enough buy and sell orders for the two associated currency pairs to facilitate currency swapping, which may in fact not be enough. Hence, other new methods or techniques may be invented to allow the process to continue in such situations, such as using a global exchange as a market maker or as a trader in the event that there are not enough trading orders to make the currency swapping continue without interruption.




REFERENCES

- [1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008.
- [2] C. Lee, "Litecoin whitepaper." <https://github.com/litecoin-project/litecoin> (accessed Sep. 06, 2019).
- [3] E. Duffield and D. Diaz, "Dash: a payments-focused cryptocurrency white paper," *GitHub*. <https://github.com/dashpay/dash/wiki/Whitepaper> (accessed Sep. 06, 2019).
- [4] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2017, pp. 4182–4191, doi: 10.24251/hicss.2017.506.
- [5] M. M. Queiroz, R. Telles, and S. H. Bonilla, "Blockchain and supply chain management integration: a systematic review of the literature," *Supply Chain Management: An International Journal*, vol. 25, no. 2, pp. 241–254, Aug. 2019, doi: 10.1108/SCM-03-2018-0143.
- [6] Z. Merkaš, D. Perkov, and V. Bonin, "The significance of blockchain technology in digital transformation of logistics and transportation," *International Journal of E-Services and Mobile Applications*, vol. 12, no. 1, pp. 1–20, Jan. 2020, doi: 10.4018/IJESMA.2020010101.
- [7] S. S. Arumugam *et al.*, "IOT enabled smart logistics using smart contracts," in *2018 8th International Conference on Logistics, Informatics and Service Sciences (LISS)*, Aug. 2018, pp. 1–6, doi: 10.1109/LISS.2018.8593220.
- [8] T. M. Hughes, "The global financial services industry and the blockchain," *The Journal of Structured Finance*, vol. 23, no. 4, pp. 36–40, Jan. 2018, doi: 10.3905/jsf.2018.23.4.036.
- [9] J. R. Varma, "Blockchain in finance," *Vikalpa: The Journal for Decision Makers*, vol. 44, no. 1, pp. 1–11, Mar. 2019, doi: 10.1177/0256090919839897.
- [10] B. Teufel, A. Sentic, and M. Barmet, "Blockchain energy: blockchain in future energy systems," *Journal of Electronic Science and Technology*, vol. 17, no. 4, p. 100011, Dec. 2019, doi: 10.1016/j.jnlest.2020.100011.
- [11] S. J. Pee, E. S. Kang, J. G. Song, and J. W. Jang, "Blockchain based smart energy trading platform using smart contract," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, Feb. 2019, pp. 322–325, doi: 10.1109/ICAIIIC.2019.8668978.
- [12] S. C. Bilow, "Introduction: blockchain in media and entertainment," *SMPTE Motion Imaging Journal*, vol. 129, no. 1, pp. 20–21, Jan. 2020, doi: 10.5594/JMI.2019.2948002.
- [13] I. Radanović and R. Likić, "Opportunities for use of blockchain technology in medicine," *Applied Health Economics and Health Policy*, vol. 16, no. 5, pp. 583–590, Oct. 2018, doi: 10.1007/s40258-018-0412-8.
- [14] V. Buterin, "Ethereum: a next-generation smart contract and decentralized application platform." 2014, Accessed: Sep. 04, 2019. [Online]. Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_White_Paper_-_Buterin_2014.pdf.
- [15] J. Kwon and E. Buchman, "Cosmos: a network of distributed ledgers," *Cosmos Whitepaper*, 2016. .
- [16] G. Wood, "Polkadot: vision for a heterogeneous multi-chain framework," 2016. <https://polkadot.network/PolkaDotPaper.pdf> (accessed Sep. 06, 2019).
- [17] E. Buchman, "Tendermint: byzantine fault tolerance in the age of blockchains," 2016.
- [18] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, Jul. 1982.
- [19] M. Castro and B. Liskov, "Practical byzantine fault tolerance," 1999.
- [20] J. Burdges *et al.*, "Overview of polkadot and its design considerations," *arXiv preprint arXiv:2005.13456*, 2020.
- [21] D. Cliff, "BSE: a minimal simulation of a limit-order-book stock exchange," *CoRR*, Sep. 2018.
- [22] K. H. Ansari and U. Kulkarni, "Implementation of Ethereum Request for Comment (ERC20) Token," *Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020*, Apr. 2020, doi: 10.2139/ssrn.3561395.
- [23] Y. Zhang, X. Chen, and D. Park, "Formal specification of constant product (xy=k) market maker model and implementation," Oct. 2018.
- [24] V. Mohan, "Automated market makers and decentralized exchanges: a DeFi primer," *SSRN Electronic Journal*, Dec. 2020, doi: 10.2139/ssrn.3722714.
- [25] D. Murdoch, "Ganache: personal blockchain for ethereum development." .
- [26] "Truffle: a development environment, testing framework and asset pipeline for ethereum, aiming to make life as an ethereum developer easier," *GitHub*. <https://github.com/trufflesuite/truffle> (accessed Sep. 06, 2019).

BIOGRAPHIES OF AUTHORS

Dendej Sawarnkatat    is an independent scholar and a professional instructor in the fields of blockchain, data science and artificial intelligence. He obtained his master's degree at University of Louisiana at Lafayette, USA. He is currently pursuing Ph.D. degree in Information Technology at King Mongkut's University of Technology North Bangkok. His current research interests are blockchain, cryptocurrency, artificial intelligence, data science and machine learning. He can be contacted at email: s5807011910038@email.kmutnb.ac.th.



Sucha Smanchat    is currently a lecturer at the Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Thailand. He obtained his Ph.D. degree at Monash University in Melbourne, Australia. During his study, he was involved with the development of a prototype scheduler for Nimrod/K system. His current research interests are in cloud workflow scheduling techniques, and machine learning and algorithms for bioinformatics. He can be contacted at email: sucha.smanchat@itd.kmutnb.ac.th.