Pallet

Protocol for Abstract-Level Ledger Ecosystem

Beyond Yet Another Chain:
IP Protocol for Future Internet of Values

Contents

Abstract

Today's blockchain encounters the following challenges: scalability and interoperability. To address these challenges, we propose Pallet, Protocol for Abstract-Level Ledger Ecosystem.

Pallet makes inter-chain transactions possible by decoupling the state of contracts from the blockchains and using Juries to execute contracts. Instead of forcing all contracts to run on a single chain, our paradigm is to have one Jury for one contract at all times. The decoupling of DApps from a particular chain boosts scalability. Furthermore, we propose a stake mechanism, which enables Pallet consensus protocol to minimize the perpetration through proper reward and penalty mechanism in the ecosystem.

Description

This document is Pallet white paper draft 0.9.4. Pallet will be continuously updating this white paper to reflect new developments. For the latest information on Pallet white paper and roadmaps, please visit Pallet official web site:
https://pallet.io

Contact

Email: contact@pallet.io
Medium: https://medium.com/pallet
Twitter: https://twitter.com/pallet_io
Forum: forum.pallet.io
Facebook: facebook.com/PalletProtocol

Introduction

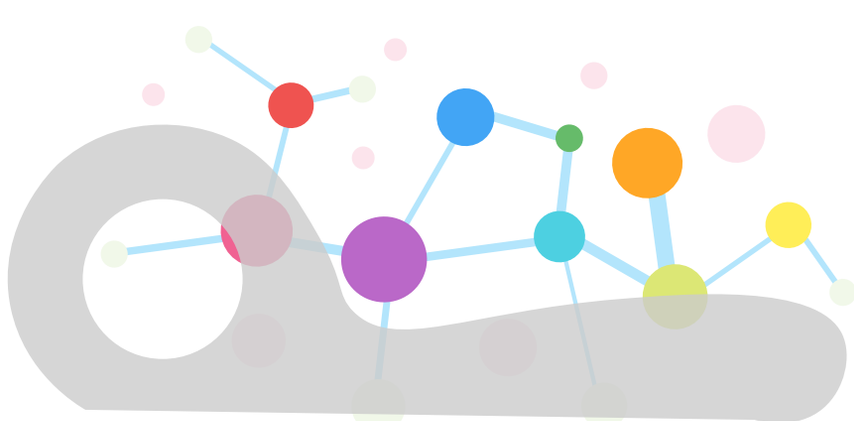
It is high time for today's money system to be revamped. The system today is slow, expensive, exclusive for someone, centralized, and in silos. As a result, the circulation of digital assets globally is not frictionless. Our vision is to enable an ecosystem where values can be transferred in an almost frictionless manner as information delivered on the internet today.

Currently, blockchain has gone beyond the geographical constraints and demonstrated the potential of the Internet of Value (IoV), shedding light on the future of the Internet of Value. However, blockchain today still faces these two challenges. First, the scalability of a blockchain remains elusive today. Second, the interoperability among different blockchains is lacking.

The creation of the genesis block on the Bitcoin blockchain in 2009 marks the beginning of an era of a decentralized and global digital currency. Ethereum extends the utility of blockchain with smart contract. The development of various distributed ledger technologies (DLT) is a precursor to the next-generation, the Internet of Value technologies.

Internet of Value

The Internet of Value is defined as enabling the future of "the exchange of any asset that is of value to someone, including stocks, votes, frequent flyer points, securities, intellectual property, music, scientific discoveries, and more" [Ripple 2017]. The current DLT, however, faces two challenges in building the next Internet-of-Value world -- scalability and interoperability.



Challenges

Scalability

Looking back on the history of DLT, Bitcoin has been a distributed trustless network in which the tokens (values) flow. The consensus of DLT is reached by every node throughout the network to guarantee the correctness. Besides, The smart contract system proposed by Ethereum follows the same principle. More precisely, Ethereum treats the network as a gigantic computer, where all the nodes execute the same procedure in order to reach the consensus on the state. As a result, Bitcoin traffic jam has become a popular search keyword since 2016. In December 2017, Crypto Kitties, an Ethereum based virtual game, severely slowed down the Ethereum transactions. These phenomena reveal the problem of resorting to the consensus of the entire network. Hence, the congestion appears because all smart contracts share the same group of executing nodes, namely, the nodes in the blockchain network. A popular service would greatly affect the performance of other services.

Interoperability

Blockchains today such as Bitcoin or Ethereum are using full nodes as brute-forces trust-machines. These full nodes verify transactions on their respective chain without knowing anything outside their chains. Therefore, such a blockchain becomes a silo to itself, making it more like an intranet today. Building a scalable and inter-operable internet-of-value system is challenging yet crucial today.

Introduction to Pallet

In order to find solutions to the challenges above, we now introduce Pallet, protocol for abstract-level ledger ecosystem. Instead of introducing another blockchain adapting between every chain, we want to connect Pallet to work in a more efficient way to deal with scalability and interoperability issues at the same time.

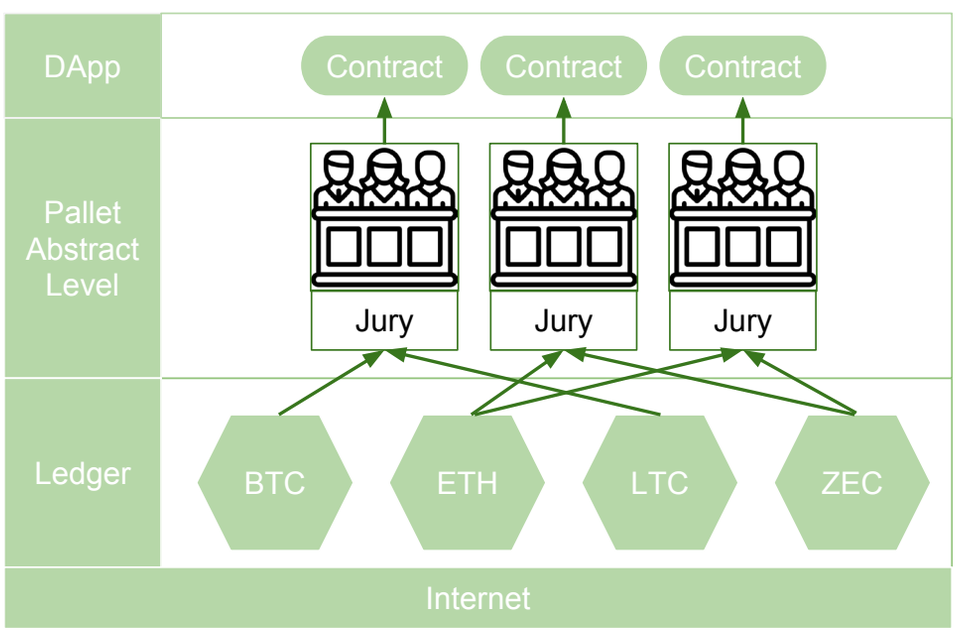


Figure 1 Concept of Pallet system. The contract's execution is on the abstract-level.

In figure 1, we can see that Pallet is an abstract-level smart contract protocol above blockchains, which divides value (or state of contracts) from blockchain. In Pallet's abstract-level smart contract, we only need a group of verifiers to execute one contract. This group of verifiers is called Jury, which consists of individual verifiers called Jurors. In our design, there is no need to reach consensus across the entire network anymore, leading to more efficient and scalable.

Transferring value is no longer bound in a single blockchain as a result of decoupling value and chain. Value can be transferred to any other blockchains easily through our abstract-level protocol.

Here, we will use an example to show how inter-chain exchange in BTC and ETH is achieved through Pallet.

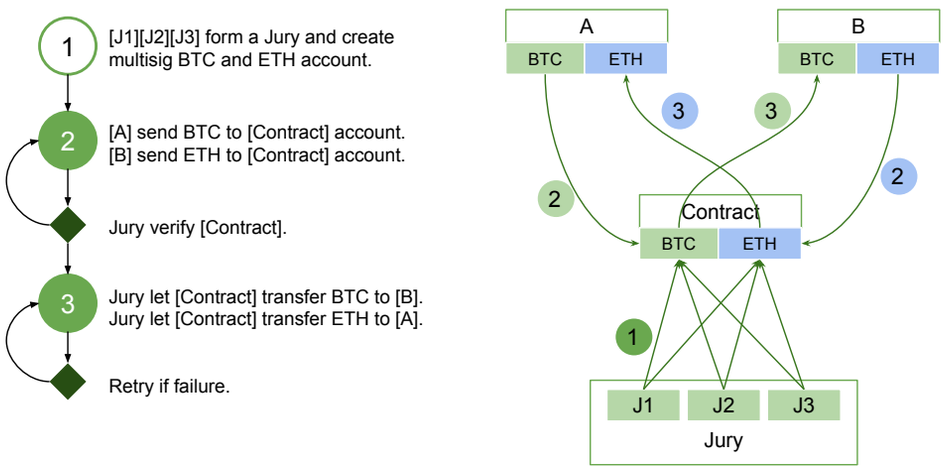


Figure 2: An example of inter-chain transfer between BTC and ETH through Pallet.

Figure 2 shows how we deal with an inter-chain exchange through Pallet. First, an exchange service provider selects Juror J1, J2, and J3, as Jury to execute the exchange contract. Then, Jury creates a multisig account on both Bitcoin and Ethereum network as the contract account. Second, if A and B want to exchange their BTC and ETH, they should send their token into the corresponding contract account. A sends BTC to contract account of BTC, and B sends ETH to contract account of ETH. Then, they may invoke the exchange contract, and the Jury verifies and tries to match the two orders. If correct, Jury will update the state of contract. Last, A and B are allowed to withdraw ETH and BTC from the contract account according to the state of contract.

Pallet Protocol

Characters in Pallet

Jury

Jury is the fundamental unit to maintain the security and integrity of Pallet. More specifically, it will be assigned to run contracts and manage multi-signature accounts. To achieve a secure and decentralized design, Jury is designed to compose of many participants, called Jurors. Every Juror pays a deposit to guarantee the security.

Pallet Distribution Contract (PDC)

Pallet Distribution Contract (PDC) takes the responsibility of the safety of Pallet network. The following is what PDC does:

· Maintaining Pal tokens, the native token of Pallet, which is used for transaction fee and maintenance fee
· Maintaining the deposit of Jurors
· Randomly choosing the Jurors in a Jury
· Arbitrating when Jurors cannot reach consensus

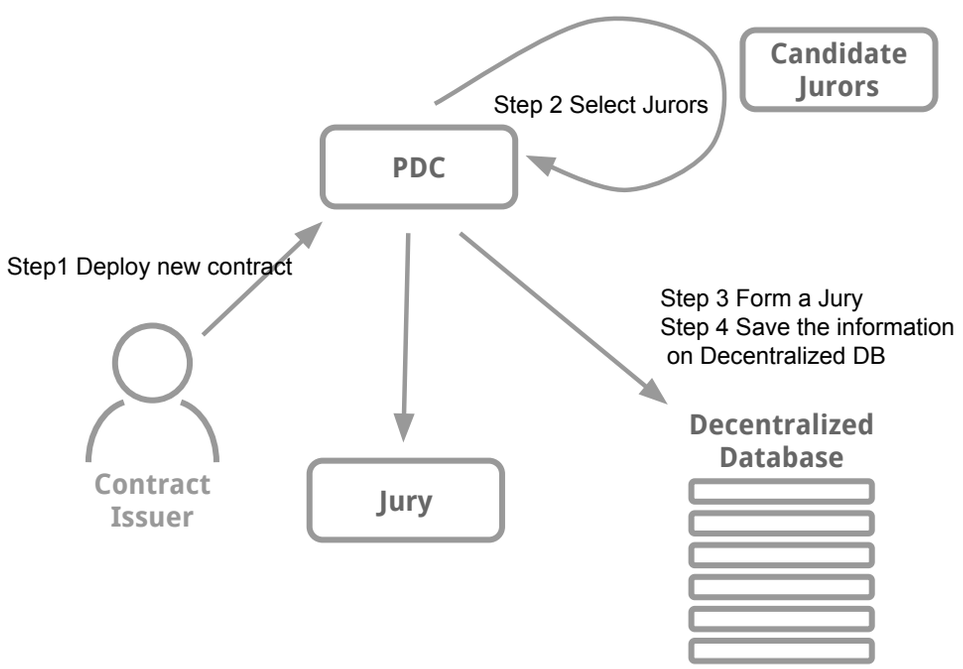
Contract Issuer / Invoker

Contract issuer and invoker use a tool called Attorney to issue a new contract or invoke an existed one. The tool, Attorney, will communicate with the Jurors in Jury.

Interacting with Contracts

Contract Deployment

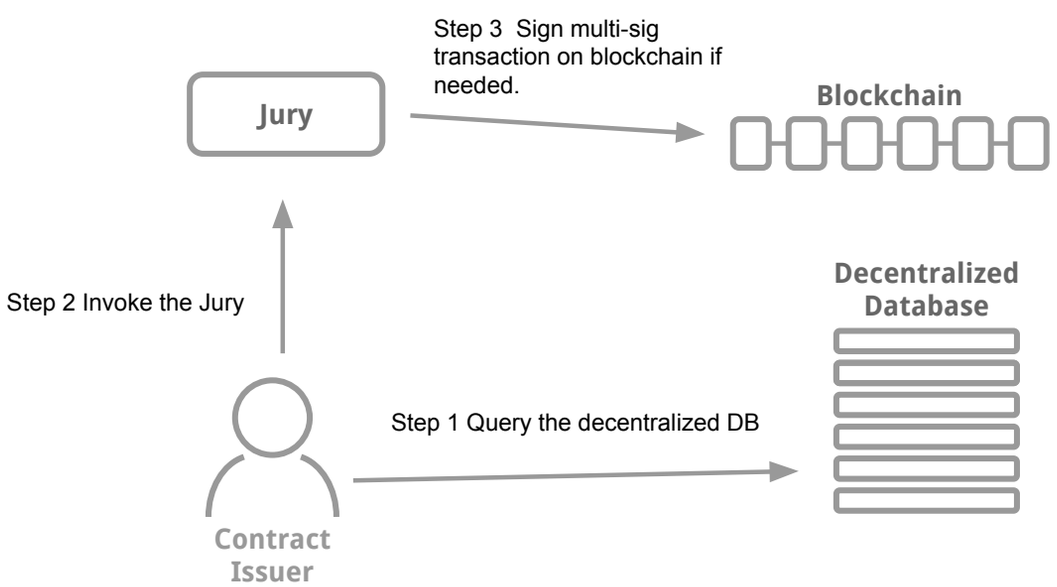
In Pallet, all kinds of services are created through contracts. Once a contract issuer attempts to deploy the contract, a list of Jurors will be designated by PDC to form the Jury. Subsequently, the issuer sends the contract executable with the initial parameters to the Jury. The Jury will then verify and process the contract. After finishing processing, the contract code and the designated Jurors will be saved in a decentralized database.



Contract Invocation

After the contract has been deployed, other participants are able to invoke it. To invoke the contract, the contract invoker queries the decentralized database by the contract ID first. Then, the database will return the contract executable and the list of Jurors who are responsible for the execution of the contract. After gathering necessary data, the contract will be packed with parameters into request object and sent to Jury.

When the Jurors receive the request, they execute the contract independently along with the latest contract state and invocation parameters. If everything runs as expected, the results of those Jurors will be the same, and the contract state will be shifted to the next one. If an interchain transaction is triggered, they will sign a multi-signature transaction on that blockchain as well.



Protocol Details

Deposit

o ensure the safety of Pallet system, Jurors must pay a deposit to prevent them from conducting fraud. To become a Juror to earn transaction fee, participants need to follow this process: First, pay a deposit to become a candidate Juror. When there's a vacancy, one of the candidate Jurors may be selected into a Jury. The Juror can earn transaction fee when executing contracts. The deposit can be withdrawn when the smart contract ends. The Juror can also withdraw its deposit after invoking the PDC to select a new Juror for replacement.

The proper amount of deposit can be evaluated through a model that contains different properties, including the value of the contract, the size of Jury, the credibility of Juror and the design of contract. We are designing a mechanism called Prosecutor to reduce the required deposit. The effect of Prosecutor depends on the consensus protocol in Jury. In case of Ripple consensus protocol (80% of signatures are needed), only 25% of deposit is needed to secure the contract.

Transaction Fee

To provide an incentive for Jury's execution, Jurors gain fees from contract participants by executing contracts. Contract participants need to pay some Pal tokens as the transaction fees to those Jurors. The transaction fee would be much lower than the transaction fee of other blockchains because only the Jurors in the corresponding Jury will run it. The Jury will only execute the contract after they verify that the Pal tokens have been successfully transferred.

Maintenance Fee

Try to imagine the case below: There is a contract which is invoked barely. In other words, the Jury of this contract does nothing most of the time. However, the contract owner or some participants still wants to keep this contract alive. This situation causes problems. If we just keep this contract alive, all Jurors will be idle all the time, but their deposits will still be locked. In this case, the liquidity of Pal token decreases badly.

Therefore, besides transaction fee, we proposed the mechanism of maintenance fee as well. Every contract should have enough maintenance fee to keep itself alive. Contract owner or others who want to keep the contract alive should always be aware of having enough maintenance fee. The maintenance fee will be charged by the Jury at a fixed interval. Once maintenance fee becomes 0, the contract will be halted, and all the Jurors can get their deposit back and just leave this contract. Contract owner should take maintenance fee as an inevitable cost of buying the service (both computing and storage) of each Juror.

This design solves two problems. First, it will prevent Pal token from being locked endlessly and improve the liquidity of Pal token in the whole Pallet network. Second, it will also prevent each Juror from storing a huge amount of useless contract data permanently.

PDC consensus

The character of PDC looks like a traditional blockchain, which is a trustworthy machine. So PDC should guarantee to make all decisions correctly. PDC uses a Proof of Stake(POS) mechanism to reach a consensus. To prevent PDC from being the bottleneck of Pallet, most of the work is only done by the Jury without invoking PDC.

Recap

Based on the aforementioned Pallet architecture, the Jury can execute the contracts and interact with the underlying blockchains. Jurors in a Jury reach the consensus to perform the reliable contract execution. Such design makes the execution efficient and scalable, since the consensus is generated by the Jury of this individual contract instead of all Jurors on the network. To reduce the cost of transaction fees and settlement latency, only contract states are stored in the underlying blockchains at the request of contract participants.

Pallet Attributes

Scalability

The key of scalability is “one contract, one Jury”. A participant-based design is used by Pallet to achieve the consensus.

A variety of consensus mechanisms have been adopted by current blockchain providers. However, most of the algorithms they use are still based on the concept of consensus throughout the whole network. Every transaction consumes the computing power of the whole network for reaching the consensus.

In Pallet protocol, on the contrary, the consensus is reached among the Jurors in one Jury, which is formed by a group of Jurors. Once it is reached, Jury will store the contract state internally. With this approach, we can effectively reduce the congestion of the whole network.

Interoperability

Contracts in Pallet are not executed by the miners of blockchains but by Juries that are decoupled from blockchain. So, the execution model of Pallet can be applied to various blockchains, including popular existing blockchains such as Bitcoin and Ethereum, as long as the blockchains allow users to put arbitrary data into the block if needed.

Take Bitcoin as an example. The arbitrary data that can be stored in the OP_RETURN field of a transaction sized up to 80 bytes. Having such a small size is a problem for the entire contract executable or contract state. To deal with it, we put the hashes of the packaged contract executable and states instead of the entire data into the blocks. This technique allows the Attorney to publish the contract progress to all blockchains in the same way.

Pallet contract supports multiple blockchains. A contract developed for one blockchain (e.g., Bitcoin) can be easily reused on another blockchain (e.g., Litecoin) as long as there are well-defined system functions and libraries in relevant blockchains. Attorney and Jury can easily interact with those blockchains and generate new transactions correspond to original contract.

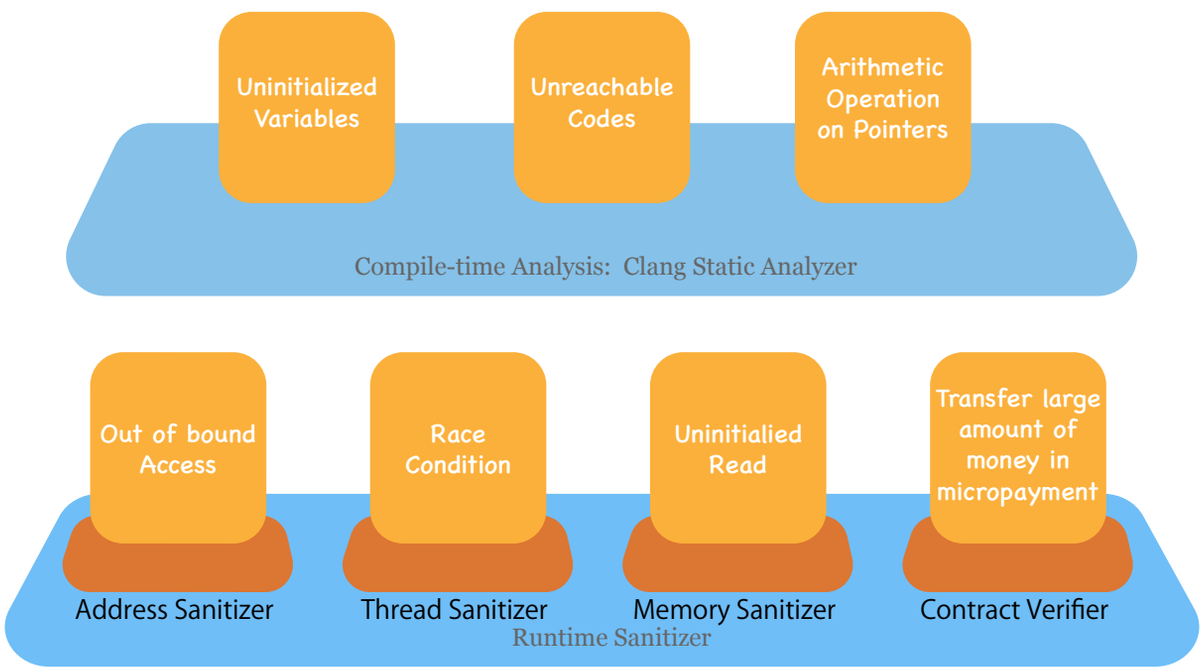
This will significantly reduce the cost of contract development and make inter-communication more easily. To benefit the developers, we will support some of the basic libraries and provide a detailed specification.

It is also possible to build a contract that can interact with different blockchains simultaneously, so users can trade tokens from different blockchains in one Pallet contract invocation to make the inter-chain token exchange distributed, atomic, and immutable. Unlike EtherDelta that is slow and specific to the Ethereum blockchain, the Pallet contracts of inter-chain token exchange can be executed in a multi-tasking way by different groups of selected Jury, and the execution of the Pallet contract is fast because the contract progress confirmation in the blockchain is not necessary.

Other attributes

Security

The LLVM-based technology makes the contract execution more secure. Instead of trying to adopt new contract-oriented programming languages which limit the contract developers to write the contracts, Pallet allows the developers to use the programming languages they are familiar with and utilize compile-time analysis tools, runtime sanitizers[1], and customized rules to check if there are any errors or threats in the contracts.



Security Guarantee

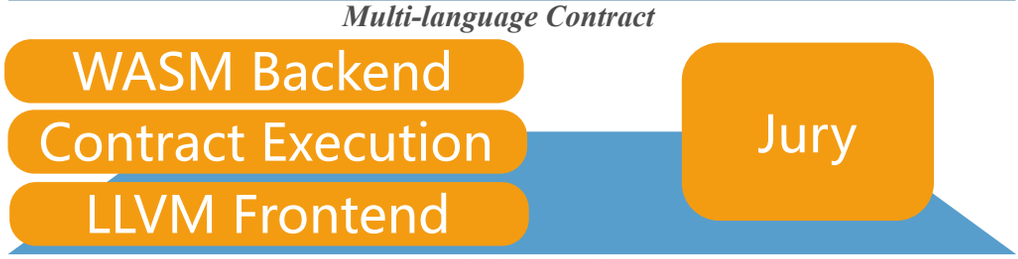
1 https://clang.llvm.org/docs/AddressSanitizer.html, https://clang.llvm.org/docs/ThreadSanitizer.html, and https://clang.llvm.org/docs/MemorySanitizer.html

Security is also achieved through Pallet consensus protocol. Jury should deposit asset to arbiter contract as stakes for later execution. If Jurors of a Jury collude, PDC (Pal Distribution Contract) will re-examine the contract execution. If they are incorrect, the arbiter contract will confiscate their stakes as compensation.

Multi-platform and Multi-language

Because the execution model of Pallet contract is flexible, Attorney and Jury can be implemented in many different ways. In our first step, we plan to use LLVM and WASM as the core technology to build the contract executable and execute the contracts. Through the LLVM toolchain, Attorney can easily compile the contracts with simplification and restriction written in several programming languages such as C++ and Python into LLVM bitcode, and the LLVM bitcode can be further converted into WASM that allows efficient execution on multiple platforms. So, Pallet contract is not only decoupled from the underlying blockchains, but also decoupled from contract languages and execution platforms.As a result, developers can choose the language they are familiar with and the platform they need.

Multi-language Contract



Decoupled Execution

Multi-Platform

Underlying Blockchain

On-Chain/Off-Chain Interaction

Pallet allows the contract to act according to the real world events by retrieving data from a function we provide. This function will gather data and the information will be verified by Jury to achieve the consensus.

Privacy

For the sensitive contracts, the contract issuers can encrypt the contracts and only allow the contract participants' Attorneys and the selected Jurors to have the public key for decryption and execution. For the highly sensitive contracts, the contracts can be encrypted, and the computation on Jurors is on ciphertext, so that even Jurors cannot know the information stored in the contracts.

Application Scenarios

Decentralized Exchange

The decentralized exchange(DEX) is defined as the direct exchange without trusting the third party. The requirement of achieving a DEX procedure is the ability of atomic swapping with a point-to-point order. Popular DEX services built on Ethereum aim to exchange between Ether and the tokens following ERC-20 or ERC-721 standard. These DEX services are implemented by smart contracts in Ethereum, such as EtherDelta.

The problem for those DEX is that it can only run on one blockchain. Therefore, cross-chain token exchange is not available. Besides, due to the fact that execution of smart contract needs consensus from all nodes in Ethereum network, placing order and confirmation is very slow in the DEX on Ethereum. These are the reasons why traditional centralized exchanges still operate regardless of the increasing hacking news.

Luckily, Pallet's fundamental structure solves those problems being in DEX for a very long time. Based on one contract, one Jury's structure, the performance of exchange contract on Pallet would be much better than current DEX. Placing orders and matching orders will be done in an instant. In addition, decoupling gives Pallet a universal inter-chain exchange solution. Rather than building relay chains between different blockchains and then bridging them together, Pallet will use different relay bridges to connect different blockchains and thus achieving interoperability.

Financial Instruments

ETF

Financial instruments are monetary contracts between parties. They can be created, traded, modified and settled. There are very few cryptocurrency ETF over the world, and all controlled by big financial institutions. You can use Pallet to create your own ETF, holding assets such as cryptocurrencies, commodities, or bonds. Create more opportunities for investors all over the world.

Forwards / Futures / Swap / Options

Standing on the shoulder of Pallet decentralized exchange, people can use Pallet contract to create their portfolio to diversify their risks, or create futures and options according to their investment strategies. Ethereum smart contract was introduced to give cryptocurrency more flexibility, and Pallet makes it better. Using inter-chain operability Pallet provides, you can compose your derivative by a Pallet contract that derives its value from the performance of the underlying entities.

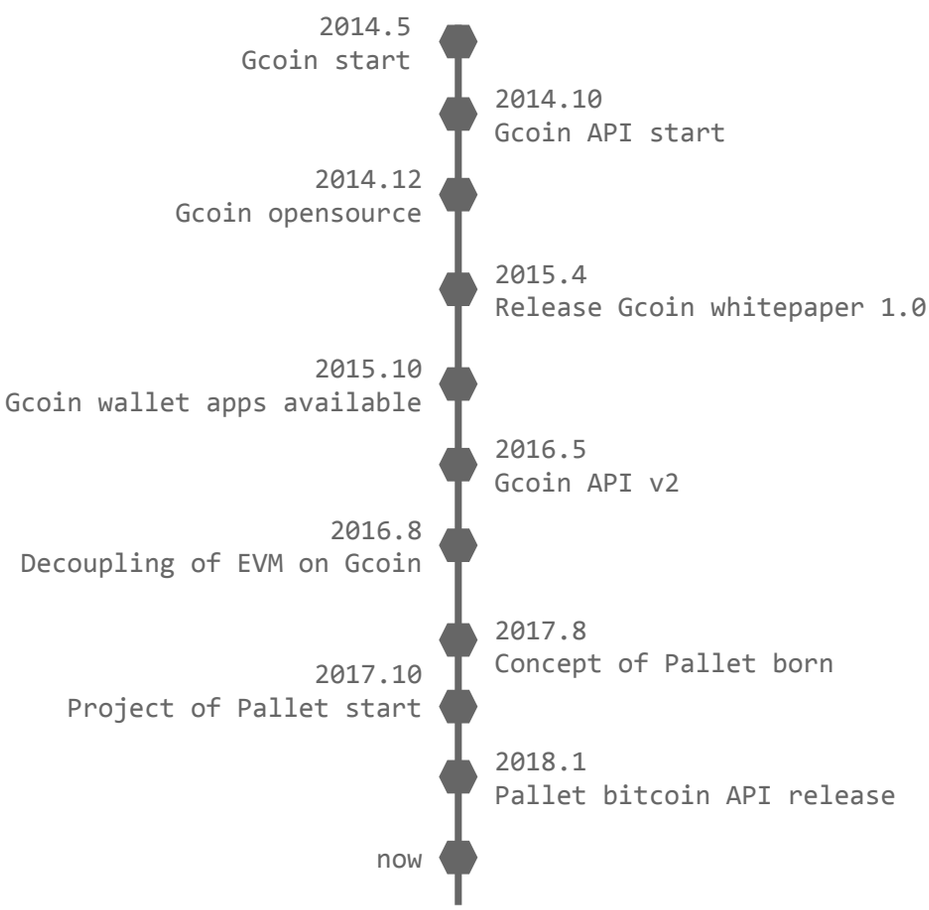
Mutual Funds

Mutual fund is a professionally managed investment fund that pools assets from many investors to purchase securities. Pallet is the best platform to share their investment strategy. In other words, they can create their mutual funds,and define the reward in the contract. Everyone will have the opportunities to create their own mutual funds like everyone has the right to use pallet and create their own color. They can distribute the funds into different cryptocurrencies as they want. With Pallet, there is no bounds for people’s creation.

About Our Team

We are a group of passionate people who are fond of technology, and believe in the vision of blockchain: the true Internet of Value. We have developed Gcoin, an open source project of blockchain protocol adopted by several institutions, since 2014. In the meantime, we have developed a series of tools[2] for Gcoin and tried to promote blockchain technology for practical usage. Several members are globally from prestigious companies such as Google, Intel and universities like Stanford.

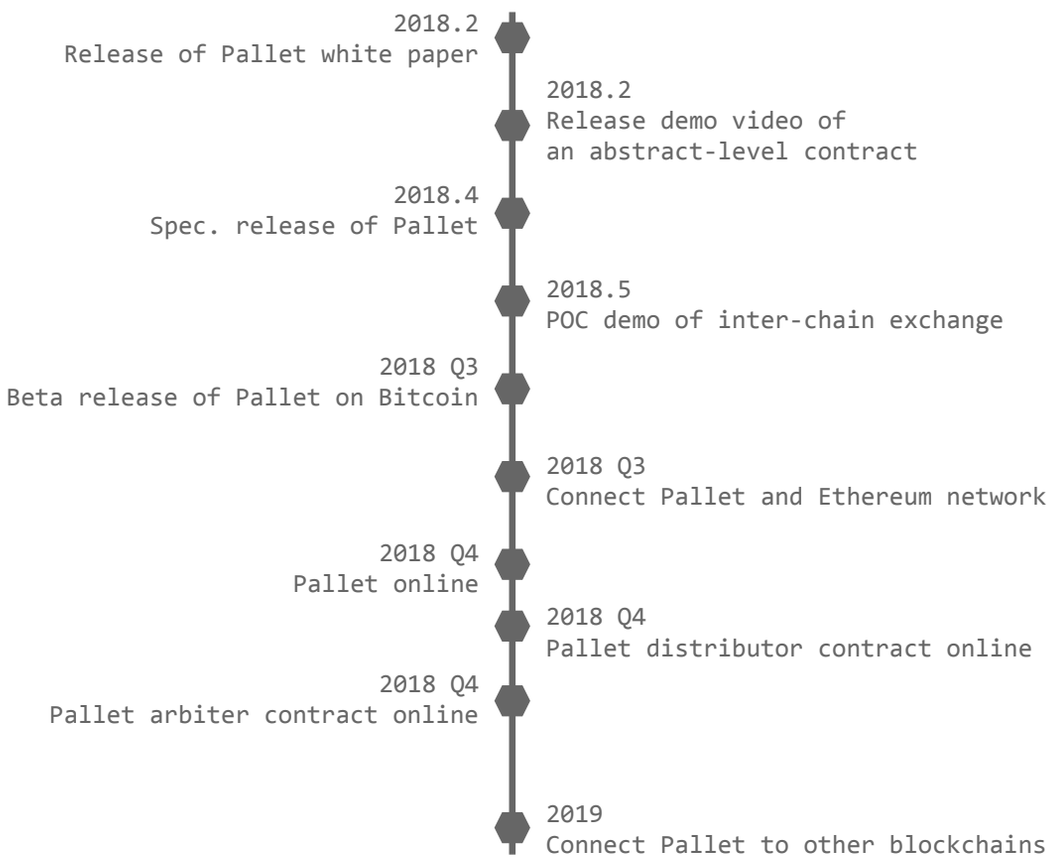
Starting in 2016, when the team tried to decouple[3] Ethereum's smart contract system, making it adaptable to Gcoin, we realized the real potential of decoupling. As it turns out,it can do more than we think before because it can overcome the challenges nowadays. From then on, we have started to research and design a new protocol. Thanks to the efforts so far, we now introduce Pallet to initiate a new generation of the Internet of Value.



History timeline of our team

2: https://github.com/OpenNetworking/OSS-Federation https://github.com/OpenNetworking/OSS/
3: https://github.com/OpenNetworking/cotract_server https://github.com/OpenNetworking/oracle_server https://github.com/OpenNetworking/go-ethereum

Road Map



Conclusion

Pallet is an abstract-level smart contract protocol which decouples execution from underlying blockchains. As a result, execution of contracts can be more scalable and able to interact with different blockchains. Benefiting from leveraging LLVM and WSAM, contract in Pallet can not only be programmed in multiple languages but also be reused by existing tools to provide a secure and high performance execution. In addition, Pallet allows users to trade on-chain and off-chain properties.

Appendix

Token issuance demo code

Code 1: Pseudo code of token issuance

```
/* This is a pseudo code of a contract running on Pallet.
This contract will demonstrate how to issue a token in a
contract.
Some methods are defined in this contract, mint(), transfer()
and get_balance().
There are some predefined variables and methods provided by
Pallet contract APIs. */

init(args):
    // init(args) will be called only once when deploying.
    state = new_contract_state()
    state.set_issuer(current_user)
    state.set_empty_user_balance()
    set_contract_state(state)

run(args):
    // All invocations will start here.
    current_user = get_current_user()
    state = get_contract_state()
    param = get_parameters()

    if (args == “Mint N”) {
        return mint(N)
    } else if (args == “transfer N tokens to user U”){
        return transfer(N, U)
    } else if (args == “get_balance of user U”) {
        return get_balance(U)
    } else {
        return invalid_invocation(“Wrong arguments”)
    }
```

```
mint(n):
    issuer = state.get_isser()
    user_balance = state.get_user_balance()
    if (current_user == issuer) {
        user_balance[issuer] += n
        state.set_user_balance(user_balance)
        set_contract_state(state)
        return OK
    } else {
        return invalid_invocation(“Permission denied.”)
    }

transfer(n, receiver):
    user_balance = state.get_user_balance()
    if (user_balance[current_user] >= N) {
        user_balance[current_user] -= N
        user_balance[receiver] += N
        state.set_user_balance(user_balance)
        set_contract_state(state)
        return OK
    } else {
        return invalid_invocation(“Insufficient token.”)
    }

get_balacne(user):
     // Assume all balance infos are public.
     user_balance = state.get_user_balance()
     return user_balacne[user]
```

Glossary

The Internet of Value: The Internet of value will enable the exchange of any asset that is of value to someone.

DLT: Distributed Ledger Technology

Abstract level: Pallet is a light-weight protocol running on a higher level on the blockchain, where we call abstract level.

Attorney: A user-side client which is responsible for contract preprocessing and supports contract deployment or contract invocation command.

Jury: A group of chosen workers who are responsible for executing and verifying contract running on Pallet.

Juror: Contract verifier who is responsible for contract execution in the Jury group.

Pal Token: Pal token is used as the transaction fees and maintenance fee of contract executions for Jury and is the native token of Pallet.

Gas: The money pay for Jurors as an incentive to run contracts**.**