Rainbow Protocol

Rainbow Protocol

THE ANCHOR OF DECENTRALIZED FINANCE

White Paper Of Rainbow Protocol

RBW Community

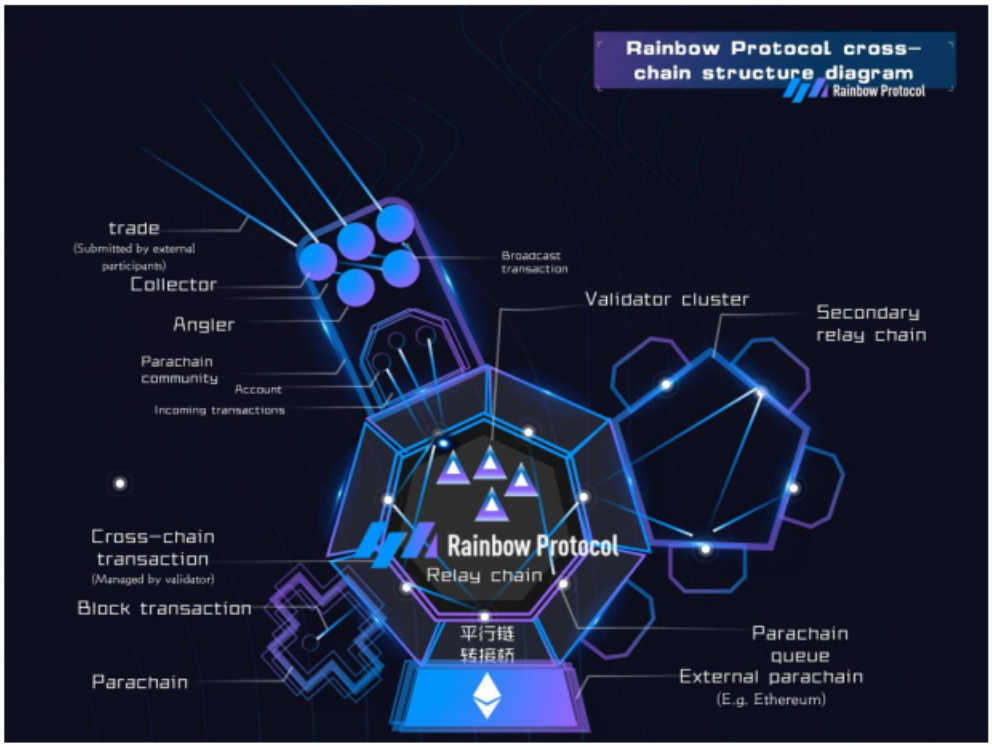
content

Summary: There are many problems in the existing blockchain architecture, not just the extensibilty and scalability issues from the perspective of practicality. We believe that the problem stems from the two very important parts of the consensus architecture: consistency (canonicality) and validity (validity) are tied too tightly. This article introduces a heterogeneous multi-chain architecture that can essentially separate the two.

In order to separate the two and maintain the minimum absolute security (security) and transport (transport) and other basic functions, we will introduce a native support for core extensibilty (core extensibilty) feasible method. For the scalability problem, we solve these two problems by dividing and conquering, and through the incentive mechanism of untrusted nodes, we weaken their endogenous binding relationship.

The heterogeneous nature of this architecture supports the interactive operation of many highly differentiated consensus systems in a trustless and fully decentralized federation, allowing trust-free mutual access to various blockchains.

We propose a way to support backward compatibility with one or more existing networks, such as Ethereum. We believe that this system can provide a useful underlying component that can practically support global business-level scalability and privacy.



1 Preface

The intent of this paper is just a summary of the technical version. It aims to use some principles to describe the blockchain example to be developed and explain the rationality of this possible direction. It lists many specific improvements in blockchain technology, as well as as many details as possible at this stage of development.

It is not meant to be written as a formal proof specification. It is not complete, nor is it the final version. It is not intended to cover non-core modules of the framework, such as APIs, dependencies, language, and usage. This is just a conceptual experiment, and the mentioned parameters are likely to be modified. In response to community opinions and comments, various components will be added, redefined, and deleted. Through experimental evidence and prototypes, give information about what will work and what will not, and it is possible to revise most of the content in this paper.

This paper contains a core description of the protocol and some ideas that may be used to solve multiple problems. It will be a core description that can be used to carry out a series of work during the proof-of-concept phase. A final "1.0 version" will be based on this

agreement, adding some ideas that become provable and decided to be included in the project.

1.1 History

- November 09, 2020: 0.1.0-proof1
- February 20, 2021: 0.1.0-proof2
- May 01, 2021: 0.1.0-proof3
- July 10, 2021: 0.1.0

2 Introduce

Blockchain has promised its great significance and can be applied to many fields including Internet of Things (IOT), finance, governance, identity management, decentralized Internet, and asset tracking. However, despite these technological promises and big talk, we have not yet seen the emergence of major real-world application deployments under the existing technology. We believe this is due to 5 key flaws of the existing technology:

Scalability: How much computing, bandwidth, and storage resources are spent globally to process a single transaction? How many transactions can be processed under peak conditions?

Isolation (Isolatability): Can the differentiated requirements of multiple parties for applications be met close to the optimal degree under the same framework?

Developability: How effective is the tool's work? Do APIs meet the needs of developers? Are the tutorial materials available? Does it integrate power?

Governance: Does the network retain the flexibility to evolve and adapt over time? Whether decision-making can be highly inclusive, reasonable and transparent to provide effective leadership in a decentralized system.

Applicability: Does the technology really solve the rigid needs of users? Do you need other middleware to graft real applications?

In the current work, we mainly focus on the first two issues: scalability and isolation. In other words, we believe that the Rainbow Protocol architecture can provide meaningful improvements in these two areas.

At present, high-performance blockchain implementations such as the Parity Ethereum client can already process more than 3000 transactions per second on consumer-grade high-speed hardware. However, the blockchain network showing the world is limited to 30 transactions per second. This limitation is mainly due to the current synchronous consensus mechanism, which requires sufficient calculation buffer time to process safely, which increases its support for slow hardware. This is attributed to its underlying consensus architecture: the state transition machine, or this way of allowing all participants to proofread and execute transactions, essentially binds its logic to the design of consensus consistency (canonicalisation), or requires All participants agree on all possibilities, validity and history.

This statement applies to proof-of-work (POW) systems like Bitcoin and Ethereum, as well as proof-of-stake (POS) systems like NXT and BitShares. They are essentially subject to the same obstacles, but these The consensus algorithm is a simple strategy that can make the blockchain successful. However, by tightly binding these two structures in one protocol, we also bundle multiple roles and applications with different risk preferences, different scalability requirements, and different privacy requirements. One feature cannot satisfy everyone's needs. Because of this scenario, there have been many widespread appeals, but the network can only tend to be more conservative and serve a small number of people, which ultimately leads to failures in innovation, performance, and adaptability, which is very dramatic.

Some systems, such as Factom, completely remove the state transition machine. However, most application scenarios need to rely on a shared state machine to support the function of state transition. Removing it just hides the problem, but does not give a real alternative solution.

It seems clear now, so a reasonable direction is: Decoupling consensus components and state transition components as routing is for a scalable decentralized computing platform. And if nothing else, this is Rainbow Protocol's strategy to solve the scalability problem.

2.1 Protocol, implementation, network

Like Bitcoin and Ethereum, Rainbow Protocol hopes to be just a network protocol at the beginning, and it is the main public network that runs this protocol (currently hypothetical). Rainbow Protocol tends to be a free and open project. The agreement is made under a Creative Commons license and the code is hosted under the FLOSS license. This project is developed in an open state and receives useful donations from all quarters. A micro-comment submission system (RFCs), but unlike the Python improvement agenda, will provide a way for the public to collaborate and participate in protocol modifications and upgrades.

Our initial implementation of the Rainbow Protocol will be called the Parity Rainbow Protocol Platform, which will include the complete implementation of the protocol and API interfaces. Like other Parity's blockchain implementations, PPP will be designed as a general-purpose blockchain technology stack, not limited to public networks, private networks, or alliance networks. The development so far has been funded by several parties including the British government.

However, this paper is still in the context of the public network. The function we foresee under the public network is a subset of a complete concept (such as a private or alliance network). In addition, in this scenario, all aspects of Rainbow Protocol can be clearly described and discussed. This also means that readers need to know that in non-public (authorized) scenarios, some specific mechanisms (such as interaction with other public networks) are not directly related to Rainbow Protocol.

2.2 Previous work

The decoupling of the underlying consensus from the state transition has been discussed privately for two years, and Max Kaye proposed it in the earliest days of Ethereum.

A more complex scalable solution is called Chain fibers, which dates back to June 2014 and was published at the end of that year. It creates a precedent for a single relay-chain and multiple isomorphic chains that can be executed transparently across chains. Decoherence is achieved through transaction latency, which makes it take longer to process transactions that need to coordinate multiple parts of the system.

Rainbow Protocol draws on a lot of its architecture and subsequent discussions with many people, although it is very different from its design and regulations.

However, there is currently no system running in a production environment that can be compared with Rainbow Protocol. Some only propose some related functions, and there are few essential details. These proposals can be summarized as: a system that discards or reduces the global correlation of state machines, a singleton state machine system that attempts to provide global correlation through homogeneous sharding, and a system whose goal is only heterogeneity.

2.2.1 System without global state

Factom demonstrates an ineffective consistency system that can record data efficiently. Since there is no global state and it brings scalability issues, it can be seen as a scalable solution. However, as mentioned earlier, strictly speaking, it only solves a few problems.

Tangle is a conceptual attempt on the consensus system. Instead of re-packaging transactions into blocks, a global consensus state change order is obtained through a series of consensus. It largely discards the idea of highly structured ordering, and instead introduces a directed non-sequence. The ring graph, the subsequent dependent transactions through a clear direction to help the previous transaction to reach agreement. For arbitrary state changes, this dependency graph will quickly become unmanageable, but for simpler UTXO models, it immediately becomes reasonable. Because the system is always loosely coherent, and transactions are usually independent of each other, large-scale global concurrency becomes very natural. Using the UTXO model can indeed position the Tangle as a currency system for value transfer, without other more general and extensible functions. Because there is no global dependency, and the interaction with other systems needs to know its state deterministically, this method becomes impractical.

2.2.2 Heterogeneous chain system

The side chain is a proposal that supports trustless interaction between the Bitcoin main chain and the affiliate chain. However, there is no specific requirement for rich interaction with side chains: interaction is limited to allowing mutual trusteeship of the other side's assets with the

side chain, which is called two-way peg in the jargon. In the end, it is also to be a framework, by anchoring the Bitcoin chain and other chains, allowing external transactions outside the Bitcoin protocol, and adding subsidiary peripheral functions to Bitcoin. In this respect, the side chain system focuses more on scalability rather than scalability.

Fundamentally, the side chain does not have any terms on validity. From one chain (such as Bitcoin) tokens to another chain, the security is only in the hope that the side chain can incentivize miners to verify transactions consistently. . The security of the Bitcoin network cannot simply work on other chains. Furthermore, a protocol to ensure joint mining of Bitcoin miners (copy their consistent computing power to the side chain) and verify side chain transactions at the same time has also been proposed.

Cosmos is a multi-chain system proposed by continuation of the side chain idea, replacing Satoshi Nakamoto's PoW consensus algorithm with Jae Know's Tendermint consensus algorithm. Essentially, it contains multiple blockchains (running in a space zone) using independent Tendermint instances, and a hub chain using trustless communication. Cross-chain communication is limited to the transfer of digital assets (that is, tokens), not arbitrary information. However, this type of cross-chain communication can return data and paths, such as notifying the sender of the status of the transfer.

Like the side chain, the issue of economic incentives for validators on the space chain has not been solved. The general assumption is that each space chain will hold its own payment tokens issued by inflation. The design is still relatively early, and at this stage, there is also a lack of details of economic means to establish scalable consistency in overall effectiveness. However, compared with those systems that require strong coupling, for the loose coupling between the space chain and the center chain, more flexibility needs to be added to the parameters of the space chain.

2.2.3 Casper

At present, there is no complete discussion and comparison between Casper and Rainbow Protocol, even if it is a fair and thorough (inaccurate) description of the two. Casper is a consensus algorithm that is reshaping PoS. It studies how to get participants to bet on the final

fork. Essentially, even in the case of long-range attacks, it is necessary to ensure the robustness against network forks, as well as the scalability of the basic Ethereum model. Therefore, in essence, the goal of the Casper protocol is much more complicated than the Rainbow Protocol and previous projects, and it also deviates from the basic blockchain model. It has not yet been made, how it will work in the future, and how it will eventually be developed.

However, both Casper and Rainbow Protocol represent an interesting new generation of protocols. The debate on Ethereum is essentially the difference in their ultimate goal and realization path. Casper is a project led by the Ethereum Foundation, which is only designed as a replacement for the PoS protocol, and has no intention of essentially creating a scalable blockchain. The key is a hard fork to upgrade, and it cannot be scalable. Therefore, all Ethereum clients and users need to upgrade, otherwise they will have to stay on the original fork with an uncertain future. Therefore, the deployment of such protocols on decentralized systems will be difficult and requires close coordination.

Rainbow Protocol is different in several aspects; first and foremost, Rainbow Protocol will be designed as a fully scalable and scalable blockchain development, deployment and interactive testing platform. It will be designed as a future-oriented platform that can absorb the latest available blockchain technology without overly complex decentralized coordination and hard forks. We have foreseen several application scenarios, such as highly encrypted consortium chains and low-block time high-frequency chains. They are unlikely to be implemented on Ethereum in the near future. Ultimately, the coupling between them and Ethereum will be very low, and there is no support for untrusted interaction between the two on Ethereum.

In short, although Casper/Ethereum 2.0 and Rainbow Protocol have some similarities, we believe that their ultimate goals are essentially different, not competition. In the foreseeable future, the two protocols will coexist with a high probability.

3 summary

Rainbow Protocol is a scalable heterogeneous multi-chain system. This means that unlike previous single blockchain implementations that

focused on varying degrees of potential application functions, Rainbow Protocol itself was designed to not provide any inherent functional applications. Rainbow Protocol provides a relay-chain, on which a large number of verifiable and globally dependent dynamic data structures can exist. We call these parallel structured blockchains parachains, although they are not required to be a chain.

In other words, Rainbow Protocol will be designed as a collection of independent chains (for example, including Ethereum, Ethereum Classic, Domain Coin, Bitcoin), except for two very important points:

- Security of the merger
- Trustless cross-chain transaction

These two points are also the reason why we call Rainbow Protocol scalable. In principle, a problem is completely solved in Rainbow Protocol: it can be expanded outwards, and there will be a very large number of parachains. Although each parachain is managed in parallel through different network modes in all aspects, this system has the ability to scale.

Rainbow Protocol provides an architecture that is as simple as possible, putting most of the complexity on middleware. This is a deliberate decision, in order to try to reduce the risk of development, so that the necessary software can be developed in a short time, but also have confidence in security and robustness.

3.1 Rainbow Protocol's philosophy

Rainbow Protocol needs to provide an absolutely solid foundation on which to build a next-generation consensus system, covering all risks from the mature design of the production level to the initial idea. By providing strong guarantees for security, isolation, and communication capabilities, Rainbow Protocol can allow parachains to choose their own from a series of characteristics. Indeed, we foresee various experimental and considered blockchain features.

We see that traditional high-market value blockchains (such as Bitcoin and Zcash), low-market value conceptual blockchains, and testnets with close to zero handling fees coexist. We see that the fully encrypted dark alliance chain and the high-functional open blockchain (such as

Ethereum) also exist together, and even provide services for it. We have seen that experimental new virtual machine blockchains, such as the Wasm blockchain for subjective time billing, are changing the difficulty calculation problem from a blockchain method similar to Ethereum to a blockchain method similar to Bitcoin. .

In order to manage the blockchain upgrade, Rainbow Protocol will endogenously support a certain form of governance structure. It is likely that based on the existing stable political system, there will be a two-chamber structure, similar to the Yellow Paper Council. The holders of the underlying equity tokens, as the highest authority, will have referendum control. In order to reflect the needs of users and developers, we hope to establish a reasonable two-house structure, adopting the opinions of users (determined by the bound validators), the opinions of major client developers and ecosystem players. Token holders will retain the highest legal rights and can form a highest court to participate in politics, discuss politics, replace or dissolve this structure, and those final needs that we do not doubt.

To borrow a Mark Twain proverb: "The government and diapers have to be changed frequently for the same reason."

However, it is trivial to organize political participation in a large-scale consensus mechanism. More qualitative changes regarding replacement and additions are hoped not through non-automatic weak laws (such as through block heights and formal proof of new agreements. To achieve consistency, it is not to change all aspects that may need to be changed by including an efficient high-level language in the core consensus algorithm. The latter is an ultimate goal, but in order to implement a reasonable development roadmap, it is more likely to choose the former.

The main principles and rules that Rainbow Protocol values are:

Minimal: Rainbow Protocol needs to have as little functionality as possible.

Simplicity: As long as they can push to middleware, put it in a parachain, or use an optimization method described below, they will not add extra complexity to the basic protocol.

General: There is no need to add any requirements, constraints or restrictions to the parachain; Rainbow Protocol needs to become the

cornerstone of the consensus system development, and try to add the most adaptable expansion and optimization to the model.

Robust: Rainbow Protocol needs to provide a stable base layer. For economic stability, it is necessary to adopt a decentralized method to reduce the possible problems caused by the attack vector of high rewards.

4 Participants of Rainbow Protocol

There are four basic roles in maintaining the Rainbow Protocol network: collator, fisherman, nominator, and validator. In a possible implementation of the Rainbow Protocol, the last role may be split into two: the basic validator and the available guarantor, which will be discussed in Section 6.5.3.

4.1 Verified person

The validator has the highest authority to help package new blocks in the Rainbow Protocol network. The validator needs to pledge enough deposit, because we allow other nominees with funds to recommend one or more validators who can represent them, so part of the deposit of the validator is not owned by them, but belongs to the nominator .

A validator must run a relay chain client on a highly available and high–bandwidth machine. On each block, the node must be prepared to receive a new block on the submitted parachain. This process involves accepting, verifying, and republishing candidate blocks. The appointment of validators is deterministic, but it is actually difficult to predict. Because validators cannot be expected to have fully synchronized data on all parachains, they hope to assign the work of proposing a new block of parachains to a third party, that is, the collector.

Once different validator groups have definitively approved the new blocks of their own parachain, they must begin to approve the blocks of the relay chain itself. This includes updating the state of the transaction queue (that is, transferring from the outgoing queue of one parachain to the incoming queue of another parachain), processing the transaction set of approved relay chains, approving the final block, and absorbing the final parachain. Change.

Under the consensus algorithm we choose, a validator who fails to perform their duties will be punished. In the beginning, if it is not an intentional error, their reward will be withheld, but if it is a repeated error, their deposit will be deducted (by burning), such as double-signing or colluding to provide an illegal block. The malicious behavior of the proof will cause them to lose all their deposits (burn a small part, and most of them will be rewarded to the information provider and the honest verifier).

To some extent, the validator is similar to the current PoW blockchain mining pool.

4.2 Nominee

The nominator is a group with rights and interests, and they delegate the security deposit to the verifier. They have no more roles, except by investing risky capital to show that they trust a particular validator (or group) to maintain the entire network on their behalf. According to their deposit ratio, they will also be rewarded and deducted in the same proportion as the total deposit of the validator.

Like the collectors below, the nominators are similar to the current miners of the PoW network.

4.3 Collector

Transaction collectors are groups that help validators create effective parachain blocks. They will run a full node of a specific parachain, which also means that they have all the necessary information to package new blocks and execute transactions, just like the current PoW blockchain miners. Under normal circumstances, they will collect and execute transactions, create an "unsealed" block, and submit it together with a zero-knowledge proof to one or more currently responsible for proposing the parachain area The validator of the block.

The precise relationship between the collector, nominator, and verifier may be revised. At first, we hope that collectors and validators can work closely together, because there may be only a few (or even one) parachains with very small transactions. The initial client implementation will include an RPC interface to support a parachain collector node to

unconditionally provide a valid parachain block to a (relay chain) validator node. As the cost of maintaining all fully synchronized parachains is getting higher and higher, we have designed additional structures to help separate independent, economically-driven, and other participants.

4.4 Fishing lers

Unlike the other two participants, anglers are not directly related to the block packaging process.They are independent "bounty hunters" and are motivated by a one-time large reward.

To be precise, due to the existence of anglers, we can reduce the occurrence of malicious behavior, even if hope happens only because the private key was accidentally leaked, rather than an intentional malicious attempt.The starting point of this name is to consider the frequency they expect earnings and the size of the final reward.

Fishlers are rewarded if they timely report and prove at least one mortgaged participant.Illegal behavior involves signing two different blocks with the same parent block, or approving an invalid block on a parallel chain.To prevent transition rewards due to the disclosure of the private key to anglers, the underlying reward for anglers reporting illegal message signatures about individual verifiers starts from a minimum, which gradually increases as other anglers report more illegal signatures.Based on our basic security assumption: at least two-thirds of the verifiers are honest, and the asymptotics will be set at 66%.

Phlers are somewhat similar to the full nodes of current blockchain systems, requiring relatively few resources and no need to promise stable online time and large bandwidth.Fishlers are so much different that they only have to submit very few deposits.This deposit is used to prevent waste of veriators computing time and resources for witch attacks.It is immediately withdrawn and is likely not more than an equivalent of several dollars, but a monitor of misconduct may be heavily rewarded.

5 Design review

This chapter attempts to give a global and complete description of a system.A deeper explanation of the system is given in the following chapter.

Figure 2: General schematic diagram of Rainbow Protocol.It shows collectors collect and broadcast transactions from users, and also broadcasts candidate blocks to anglers and verification.It is shown that the user submits a transaction, transferred first to outside the parallel chain and then through a relay chain to another parallel chain as a transaction that can be executed by the account there.

5.1 common view

On the relay chain, Rainbow Protocol achieves mutual consensus on effective blocks via a modern asynchronous (asynchronous) Byzantine fault tolerance (BFT) algorithm.The algorithm is inspired by the simple Tendermint and HoneyBadgerBFT.The latter provides an efficient fault tolerant algorithm in an architecture with arbitrary network defects as long as most of the validation are honest.

Perhaps a network of permission proof (PoA) mode is sufficient, however Rainbow Protocol is a network that can be deployed in fully open and open scenarios, without having to trust any special organization and authority to maintain it, so we need a way to manage validation groups and motivate them to abide by the law.We choose to use a PoS-based consensus algorithm.

5.2 Certificate of equity

We assume that the network can measure how many interests (stake) per account has.To more easily compare with existing systems, we call units of measure "token" (tokens).Unfortunately because it can only serve as a simple value measure to the account and without any personalization, multiple reasons make the term less idealized.

Through a nominated Proof of Interest (Nominated Proof-of-Stake NPos) structure, we conjecture that verifier elections will not be very frequent (most likely once a quarter, at most once a day).Incentive through a proportional distribution of additional tokens (probably approximately

10%, up to 100% per year) and collected transaction fees.While currency issuance generally cause inflation, because all token holders have fair participation, token holders do not suffer assets over time and they are happy to participate in the consensus mechanism.The mortgage required for the development of the whole network equity certificate must reach a specific minimum proportion.According to the market mechanism, it will achieve the goal of effective tokens for additional issuance.

The veriators rely heavily on their interest they mortgage in.The existing verifier deposit will start when they leave and remain for longer (perhaps about 3 months).Such a long deposit freeze is designed to also punish future misconduct until periodic blockchain checkpoints arrive.Misconduct is punished, such as reduced rewards, and if it intentionally destroys the integrity of the network, the verifier will lose part or all of his equity to other verifiers, information providers or all equity holders (through burning).For example, a verifier trying to approve two branches on different forks simultaneously (sometimes known as short-range attacks) can be identified and punished by later methods.

Checkpoint locks (checkpoint latch) circumvent long-range "equity-free mortgage" (nothing-at-stake) attacks, preventing highly dangerous chain reconstruction (chain-reorganistation) longer than the general length.To ensure that the latest clients that begin synchronization are not misled into the wrong chain, the network has regular "hard forks" (longest to the reviewer's deposit freeze period) and hardcoding the hash of the recent checkpoint block (hard-code) into the client.This will work well in the future by gradually decreasing the length of the finite chain (finite chainlength), or periodically reset the creation block (genesis-block).

5.3 Parallel chain and the collector

Each parallel chain will provide the same security guarantee to the relay chain: the block head of the parallel chain is included in the block of the relay chain, followed by some confirmation information to ensure that no chain reconstruction or double cost (double-spending) occurs.Similar to the security guarantee of Bitcoin side chains and joint mining, Rainbow Protocol also strongly guarantees the effectiveness of parallel chain state transactions.The verifier are randomly divided into many groups based on a cryptographic algorithm.A parallel chain corresponds to a group and even each block may be different.This

setting means that the relay chain is at least as short as the exit time of the parallel chain.This paper does not discuss the particular decision methods of grouping, perhaps either around the RanDAO-like submission-disclosure (commit-reveal) framework or combining the cryptographic hash values of the previous block of the parallel chain.

Such groups of validation need to provide candidates for parallel chains and ensure that they are valid (otherwise lose deposit).Effectiveness revolves around two important points: First, it is endogenous and all state transformations are executed fairly executed, including the quoted external data also (such as transactions).Second, the participants need to easily access any external data of the candidate blocks, such as external transactions, and then they can download the data and execute the candidate blocks manually.Valators can submit empty blocks (null) that do not contain any external transaction data and if they do, take the risk of reward reduction.They work with the Collector on a gossip (gossip) agreement in the parallel chain, the Collector collects the transaction into the block and provides a non-interactive proof of zero knowledge (noninteractive zero-knowledge) that the parent block of the subblock is valid (charging any fees for the work).

The prevention of garbage (spam) data is left to the parallel chain protocol itself: the relay chain essentially excludes "calculated resource measurement" and "transaction-fee."Nor does the parallel chain essentially mandatory to stipulate the relevant agreement (although equity holders are unlikely to be willing to accept a parallel chain that does not provide a reasonable mechanism).It is clear that it is not all like Ethereum's fee rules, or like Bitcoin's blockchain fee model, or any other garbage prevention model that has not yet been proposed.

The relay chain of the Rainbow Protocol itself would most likely have existed in an Ethereum-similar account and state model, possibly a derivative version of the EVM.Because relay chain nodes will need to do a lot of other calculations, they will minimize transaction throughput by increasing fees, and our model also includes block size limits.

5.4 Cross-chain communication

The most critical part of the Rainbow Protocol is cross-chain communication.Because there can be some kind of information channel between parallel chains, we say that Rainbow Protocol is a scalable

multi-chain system.In Rainbow Protocol, communication can be simple: when executing a transaction in a parallel chain (based on the logic of that chain), a transaction can be forwarded to the second parallel or relay chain.External transactions of the blockchain in the current production environment can only be completely asynchronous, and they do not have the original ability to return any information to its source party.

Figure 3: A basic schematic diagram showing the main logic of routing submitted transactions ("commit")

To ensure minimal implementation complexity, minimal risk, and minimal parallel chain architecture constraints, these cross-chain transactions are no different from the current standard external transactions.These transactions have a source field to identify a parallel chain and an address that can be any length.The fees paid for cross-chain transactions is not like the current Bitcoin or Ethereum system, but must be managed through the negotiated logic of parallel source and parallel purpose chains.A proposed improvement in the Serenity version of Ethereum would be a simple way to manage this cross-chain resource payment, although we assume that others would propose more state-of-the-art approaches.

The problem of cross-chain transactions can be solved with a simple queue mechanism that uses the Meckel tree (Merkle tree) to guarantee the data authenticity.The task of the relay chain is to transfer transactions from the exit queue of the source parallel chain to the incoming queue of the destination parallel chain.The forwarded transactions are referenced on the relay chain, not the relay chain itself.To prevent one parallel chain from sending junk transactions to another, it is stipulated that the incoming queue of the target parallel chain cannot be too large when each transaction is sent after the end of the previous block.If the block is too large and the entry queue is processed, the destination parallel chain will be seen as saturated, and the transaction will not be routed again in the next few blocks until the entry queue drops below the critical value.These queues are managed on relay chains, allowing each parallel chains to determine their saturation size with each other.If the transaction is sent to the stagnant target chain, the failure can be reported synchronously (because there is no return path, and if the second transaction fails for the same reason, it

may not send a reply to the source caller, which requires some other recovery methods).

5.5 Rainbow Protocol and Ethereum

Thanks to the Turing-complete nature of Ethereum, at least within simple demonstrable security boundaries, we expect rich interaction possibilities between Rainbow Protocol and Ethereum.In short, we envisioned that transactions coming out of Rainbow Protocol allowed the verifier to sign first and then fed to Ethereum, where it could be interpreted and executed through a deal forwarding (transaction-forwarding) contract.Counterdirection, we also envisioned that from a special format log in an "outgoing contract" (break-out contract) on Ethereum, it could quickly prove whether a message is really going to be forwarded.

5.5.1 From Rainbow Protocol to Ethereum

By selecting a Byzantine fault-tolerant algorithm where the verifier consists of a series of equity holders generated by authorized voting, we are able to obtain a secure consensus mechanism with the appropriate number of verifiers that do not frequently change.Within a system with a total of 144 verifiers, 4s outgoing time and finality of 900 blocks (allowing reporting, punishment, and repairing malicious behavior similar to two-way voting), the effectiveness of a block can reasonably be considered for proof with at least 97 signatures (plus two-thirds of 144), followed by 60 minutes of risk-free-injection validation time.

Ethereum can include an inward contract (break-in contract) that controls and maintains 144 signatures, since elliptic curve digital signatures cost EVM 3000 gas computation and because we only want validation operations to occur in most reviewers (not all), Ethereum confirms that the basis of an instruction from Rainbow Protocol does not exceed 300,000 gas—— is only 6% of the block 5.5 million gas limit.Increasing the number of validation (only necessary when processing dozens of blockchains) inevitably increases cost, however it is clear that transaction throughput increases over time as Ethereum technology matures and architecture improves.Another fact that all verifiers are involved (for example only the verification with the highest deposit) the limitations on this structure would be more reasonable.

Assuming that these verifiers rotate daily (more conservative and more likely to receive weekly or even monthly), the network cost to maintain this Ethereum transfer bridge is about 540,000 gas per day, or at current gas prices, $45 dollars a year.A basic forwarding transaction through a transfer bridge will cost about $0.11; of course additional contract calculations will cost more.By caching and bundling multiple transactions, inward transaction costs can be simply shared, reducing the cost per transaction.If a forwarding requires enough 20 transactions, the cost of forwarding a basic transaction is reduced to about $0.01.

In this model the Rainbow Protocol verifier node needs to do little more than signed messages.To be able to route transactions to the Ethereum network, we assume that any verifier needs to belong to the Ethereum network, more likely to provide few rewards to the first to forward the message on the network (rewards are paid to the transaction sponsor).

5.5.2 From Ethereum to Rainbow Protocol

Using a concept called logging, forward the transaction from Ethereum to the Rainbow Protocol.When an Ethereum contract wishes to derive a deal to a certain parallel chain above the Rainbow Protocol, it simply calls a special "outgoing contract out."That outgoing contract takes any necessary fee and then generates a log print instruction to prove its existence through the Meckel tree and the block hash.

In the following two cases, the validity can be proved very simple.In principle, the only requirement is that each Rainbow Protocol node run a standard fully synced Ethereum node.But this itself is very heavy dependent.A more lightweight approach is to provide a simple proof that only contains the part of the status tree that the exchange must know correctly before checking the validity of the log.Such a proof similar to simple payment verification (SPV-like) does not need to provide large amounts of information.More conveniently, the verifier may not need to run the node at all, the deposit system in Rainbow Protocol supports third-party participants to submit blocks because other third parties (aka anglers) may also provide a proof that their block is invalid (specifically state and receipt roots are wrong), so these people also risk losing their deposit.

There is no final eventable agreement on a PoW network without final certainty (non-finalising) like Ethereum.To adapt to this, the program

needs to rely on a certain number of block confirmation, or until that dependent transaction is already at a particular depth within the chain.On Ethereum, this depth extends from the most vulnerable block (completely unknown to the network) to 1,200 blocks (from Frontier online to Ethernet trading).On the stable version of Homestead, most exchanges choose the number of 120 blocks, and we may also choose similar parameters.

So we can imagine that the Ethereum interface on Rainbow Protocol has some simple functions: accepting the new block header of the Ethereum network, and verifying that its PoW, can combine a block header with sufficient depth (and the corresponding information forwarded within Rainbow Protocol) to verify the proof of specific logs printed from the Ethereum's export contract, and about the invalid receipt root in the previously received but not determined header.

A forwarding incentive is needed to really get the Ethereum block header data in the Rainbow Protocol network (and any SPV proof of effectiveness and consistency).This may be designed to be a simple payment act (funded by fees collected over Ethereum) to anyone who can provide a effective block.To be able to cope with bifurcation, the verifier needs to retain the recent thousands of information or native supported by the protocol

5.5.3 Rainbow Protocol and Bitcoin

The interaction between Rainbow Protocol and Bitcoin is very challenging: a so-called "two-way anchor" architecture is useful from the network perspective on both sides.However, due to the limitations of Bitcoin, how to provide a security anchor is a very difficult task.You can use Ethereum like processes to forward a transaction from Bitcoin to Rainbow Protocol: by "an outgoing address" (break-out address) controlled by the Rainbow Protocol verifier to host the tokens (and attached data) transferred by the transfer.Encouraging the Prophet (oracles) can provide a SPV proof by combining a confirmation period, and the prophets demonstrate the possibility of double flowers in a transaction by identifying a non-consistent block.Any Bitcoin hosted in an outward address is in principle controlled by the same group of validation.

The question is how to guarantee that these Bitcoins are controlled by rotating sets of validation.Bitcoin has more limited than Ethereum to combine any signature rules in a contract, and most bitcoin clients can only accept multiple signatures of up to 3 parties.The ultimate expansion to 36 or the highest one you hoped for is impossible in the existing Bitcoin protocol.One option is to modify the bitcoin protocol to support this feature, however the hard fork is very difficult to arrange and discuss in the world of Bitcoin.Another possibility is to use a method of threshold (threshold) signature, to construct a public key address jointly controlled with a cryptographic structure by multiple private key fragments, and a participant requires most or all of these people to make a valid signature.Unfortunately, threshold signatures are computed very resource consuming and at the polynomial level of complexity (polynomial complexity) compared to Bitcoin ECDSA.

Since the fundamental nature of entry security is determined by mortgaged verifiers, another option is to reduce the number of private key holders with multiple signatures to only heavily pledged verification, so that threshold signatures become feasible (or worst, possibly directly with native multiple signatures of Bitcoin).This method reduces the total amount of managed bitcoin due to preventing the illegal behavior of validation.However, this is an elegant compromise that can simply set the total fund cap that can be safely transferred in both networks (the deposit loss that the verification attack fails, and the potential bitcoin gains that the attack success may receive)

Therefore, we argue that under the existing bitcoin framework, it is unrealistic to develop a parallel chain that can safely transfer bitcoin between two networks, although the holders of bitcoin can also coordinate these works in an uncertain future.

6 Agreement details

This Agreement can be roughly divided into three parts: consensus mechanism, parallel link port, cross-link transaction routing system.

6.1 Relay chain operation

The relay chain is similar to Ethereum and is also state-based, containing a mapping of account information to state storage where the

information mainly includes the balance and transaction counter (prevent replay).The goal of placing the account system here is to record how many interests each identity controls in the system.But there are still some noteworthy differences:

- Contract cannot be deployed through transactions; this is to make the relay chain as unfunctional as possible and does not support open deployment contracts.
- There is no resource counter (gas); because some of the functions the public can call is fixed, the gas recording system principle is not applicable.So in all features, a more common fee standard is used so that those dynamic code is executed more efficiently and the transaction format is simpler.
- There are default contracts with special features that manage the automated execution of transactions and the output of network messages.

The relay chain will have a EVM based virtual machine but a lot of modifications to maximum simplification.It will have specific features of the built-in contracts (similar to those located between 1-4) running the platform, including consensus, verifier, and parallel chain contracts.

It is likely to choose Web-Assembly (Wasm) without EVM,; all structures are similar, but these built-in Wasm-based contracts use a generic language, not the immature language with many restrictions on EVM.

It may also be borrowed from other aspects currently derived on Ethereum, such as changes proposed in the Serenity version, such as simplifying the receipt format of transactions in order to execute parallel transactions without state conflict in a block, etc.

It is possible that Rainbow Protocol will deploy a pure (pure) blockchain system similar to Serenity that does not contain any underlying protocol of the chain.But we think it brings more complexity and development uncertainty, so it is not quite worth implementing such a more efficient and brief great protocol at this stage.

In order to manage the consensus mechanism, many functions of small pieces are needed: verification person collection, verification person mechanism, parallel chain, etc.These can all be placed in an overall

protocol.However, to achieve modularity, we describe these as contracts with relay chains.This means that they are all objects managed by the relay chain consensus mechanism (like object-oriented), but not necessarily EVM-like bytes and may not be addressed through the account system.

6.2 Equity contract

This contract manages the set of validators:

- Which accounts are validators;
- Which ones can become validators in the short term;
- Which accounts have pledged rights and interests in order to nominate validators
- Each person's attributes, including balance, acceptable deposit ratio, address list, session identity

It allows an account to register when it wants to become a validator (some requirements must be met), to nominate a user, and to log out when it wants to withdraw from the validator role. It also contains some functions for verification and agreement.

6.2.1 Liquidity of the equity tokens

Usually we want to mortgage as many equity tokens as possible from the network, because it related to the total market value of the mortgage equity and the security of the network.This can simply motivate verifiers through currency issuance and earnings distribution.However, this poses a problem: if the tokens are mortgaged in equity contracts to prevent evil, then how to ensure the basic liquidity of the token, and then support price discovery?

One approach is to provide a forward derivative contract to manage secondary tokens derived from the mortgage token.But this is hard to achieve in untrusted situations.These derivative tokens cannot be traded equivalent for the same as bonds issued by different governments in the euro zone: mortgaged tokens may be deducted for reduced value.As for the European governments, they may also default.For tokens pledged by the verifier, consider where his malicious behavior may be punished.

Based on our principles, we chose a simpler solution: not to pledge all the tokens.This means that some (possibly 20%) tokens will be forced to remain negotiable.Although the scheme is not perfect from a security perspective, it does not fundamentally affect the security of the network.Only an 80% interest will also be confiscated as indemnity compared to the 100% pledge.

We will also use a reverse auction mechanism to fairly determine the proportion of pledged and circulating tokens.Token holders interested in becoming verifiers can submit a request to the equity contract stating the minimum percentage they wish to pay.At the beginning of each session (which may be counted once an hour), the system fills up the examiner's slot based on the deposit and expense ratio of each intention examiner.A possible algorithm is to select those who submit the deposit for no higher than the "Total Deposit Target / Number of Slots" and not less than half of the "sublow deposit."If we don't fill these slots, we quickly reduce the sub-low deposit to meet the conditions.

6.2.2 Nominate

Users can untrust the equity token to an activated verifier to perform their duties.Nominations are completed through a "" approval-voting " system.Each prospective nominee may submit a statement to the equity contract indicating the identity of one or more verifiers whom they trust and may perform their duties.

During each session, the nominee's deposit is distributed to one or more witnesses representing them.The deposits of these verifiers are equally distributed.The nominee's deposit is used to verify the person for their liability and will be able to receive interest or bear the corresponding deduction.

6.2.3 Deposit forfeiture / burning

Certain actions of validators will result in punitive forfeiture of their deposits. If the deposit is reduced to the minimum allowed, the session will end prematurely and another session will begin. An incomplete list of actions that will lead to punishment:

• Belongs to a group of validators of a parachain, but does not provide legality verification for the blocks of the parachain;

• Signed an illegal block of the parachain;

• Do not process messages that have been voted into effect in the exit queue;

• Not participating in the consensus process;

• Sign simultaneously on two competing forks of the relay chain.

Some actions threaten the integrity of the network (such as signing illegal parachain blocks, or signing multiple forks). In order to expel these verifiers, their deposits will be confiscated. In addition, there are some less serious behaviors (such as not participating in the consensus process) or those that cannot be clearly identified (such as being in an inefficient group), which will only result in a small part of the deposit being penalized. In the latter case, the stirring function of a second-level group can be used to make malicious nodes suffer more punishment than normal nodes.

Because synchronizing the block of each parallel chain is a very big job, in some cases (multiple and illegal signatures), the verifier cannot easily detect their own misconduct.It is necessary here to point out here that some participants outside the verifier can also report these illegal acts and receive rewards from them, but they are not quite like the anglers.

Because some of the cases are very serious, we hope to simply pay bonuses from the forfeited deposit.We generally prefer to redistribute using burn tokens instead of bulk transfer.Burning burning can increase the value of the token as a whole and compensate the whole network, not only the specific parties involved.This is mainly as a security mechanism, only very bad behavior will be a very large amount of punishment.

It is important that the bonuses must be high enough for the web to feel that the verification work is worth doing and certainly cannot be much higher than the cost or invite rich enough, orchestrated international-level criminal hackers attacking unfortunate verito force illegal behavior.

The prescribed bonus cannot be much higher than the malicious examiner's deposit, otherwise it will improperly encourage illegal behavior: the examiner reports himself for the bonus.The solution is to

either directly limit the minimum amount of deposit to become a verifier or indirectly educate the nominees: if the verifier deposit is too small, they may not have enough incentive to follow the rules.

6.3 Registration of the parallel chain

This module is used to record each parallel chain in the system.It is a relatively simple database-like structure that manages the static and dynamic information for each chain.

Static information includes the index of the chain (an integer) and the identification of the validation protocol.The protocol identification is used to distinguish between different parallel chains, only so that validation can run the correct validation algorithm and then submit legitimate candidate blocks.An original proof-of-concept version focuses on how a new validation algorithm is placed in the client, so that for every new type of blockchain, a hard fork is added.However, with strict and high efficiency, it is possible to let the veriknow the new verification algorithm without hard fork.A possible implementation is to describe verification algorithms for parallel chains, such as WebAssembly, in a determined, locally compiled, platform-independent language, etc.To verify the feasibility of this method, we have to do more research, after all, there will be a great advantage if we can avoid hard forks.

Dynamic information involves transaction routing systems, such as a global consensus on the entry queue of a parallel chain (discussed in the next section).

A referendum must be passed to register the new parallel chain.This could have been managed directly internally, but would be better through an external referendum contract that could also be used for the governance of many more other scenarios.Specific parameters about the parallel chain voting registration system (e. g. quorum, majority ratio) will be formalized proof into an irregularly updated "main constitution" system, and of course the initial stage may only be in a traditional approach.The specific formula is not within the scope of this paper, such as 2 / 3 of the majority passed, and the system 1 / 3 of the tokens are involved in the vote.

There are also some operations to pause and delete parallel chains.We hope to never pause a parallel chain, but this is designed to handle

some emergencies in the parallel chain.The most obvious case is that the verifier runs multiple client implementations of a parallel chain that may be unable to reach consensus on a block.We also encourage validation to use multiple client implementations so that this can be detected early to prevent deposits from being deducted.

Because the suspension is an emergency measure, a ators vote rather than a referendum.For the restart operation, it may be done directly by a verifier vote or by a referendum.

Delete operational parallel chains can only be done by a referendum, and provide a loose smooth exit transition that making them a separate blockchain or part of other consensus systems.This period may be months and is best to be made by parallel chains based on their own needs.

6.4 Package the relay chain blocks

The process of block packaging is essentially a process of consensus and of making basic data meaningful.In a PoW chain, the packaging has a synonym called mining.In this scheme, it involves collecting the verifier's signatures for the block validity, availability, and consistency of blocks including relay blocks and blocks of all the parallel chains it contains.

The underlying BFT consensus algorithm is also not the current scope of work.Instead of describing it, we use a primitives to describe a state machine driven by consensus.Ultimately we hope to be inspired by some existing consensus algorithms: Tangaora (BFT variants of Raft), Tendermint, and HoneyBadgerBFT.Consensus algorithms require concurrent consensus on multiple parallel chains.Assuming that once the consensus is reached, we can irrefutably record who is involved.We can also narrow the wrongdoer into a group with only which malicious participants to include, so that the collateral injury can be reduced in punishment.

These proofs, the state roots of the relay chain, and the transaction roots are stored together in the block header of the relay chain.

The packaging process for relay and parallel blocks is in the same consensus generation mechanism, and the two types of blocks jointly form the content of the relay chain: parallel chains are not collected

after "submitted" by their groups.This leads to the relay chain process, but also allows us to complete the consensus across the system at one stage, minimize delays and support more complex data availability, which will be useful in the routing process.

A simple table (2 D) can be modeled for each participant.Each participant has a series of information from other participants existing in the form of a signature, describing the candidate blocks of each parallel chain and those of the relay chain.These information has two parts of the data:

Availability: For submitted transactions for this block in the export queue, do you have sufficient information to correctly validate the candidate block for the parallel chain in the next block?They can cast 1 (know) or 0 (not sure).When they cast 1, they promised to vote the same in subsequent votes.The subsequent vote and this does not correspond will lead to punishment.

Effectiveness: Is the block of the parallel chain valid and contains all the quoted external data (such as transactions)?This is related to the examiner's vote on the parallel chain.They can cast 1 (valid), −1 (invalid), or 0 (uncertain).As long as they cast non−0, they promise to vote the same in subsequent votes.The subsequent vote and this does not correspond will lead to punishment.

All veriators must vote; under the rules above.Consensus processes can be modeled like many standard BFT consensus algorithms, and each parallel chain is parallel.In addition to the small probability that a small number of malicious participants are assigned to the same parallel chain group, consensus algorithms can support the network as a whole, and the worst case is just one or more parallel chains locked by empty blocks

Availability: For submitted transactions for this block in the export queue, do you have sufficient information to correctly validate the candidate block for the parallel chain in the next block?They can cast 1 (know) or 0 (not sure).When they cast 1, they promised to vote the same in subsequent votes.The subsequent vote and this does not correspond will lead to punishment.

Effectiveness: Is the block of the parallel chain valid and contains all the quoted external data (such as transactions)?This is related to the examiner's vote on the parallel chain.They can cast 1 (valid), −1 (invalid), or 0 (uncertain).As long as they cast non−0, they promise to vote the same in subsequent votes.The subsequent vote and this does not correspond will lead to punishment.

All veriators must vote; under the above.Consensus processes can be modeled like many standard BFT consensus algorithms, and each parallel chain is parallel.In addition to a small probability of assigning a few malicious participants to the same parallel chain group, the consensus algorithm still supports the network overall, and at worst it is only the case where one or more parallel chains lock by empty blocks (and one turn makes a penalty)

The basic rule for determining whether a independent block is valid (allowing all verifiers to reach consensus as a whole, and then these parallel chain blocks become consistent data references on the relay chain):

• At least two−thirds of validators need to vote "yes" and no one to vote "no".

• More than one−third of validators are required to vote "yes" on the availability of egress queue messages.

In terms of effectiveness, if there is at least one "yes" and at least one "no" vote, a special condition is activated, and the entire validator must vote to determine whether there are malicious participants or whether unexpected points have been generated. cross. In addition to valid and invalid, the third type of vote is also supported, which is equivalent to voting "yes" and "no" at the same time, indicating that this node has conflicting opinions. This may be due to the divergence of multiple client implementations run by the node owner, and it indicates that there may be unclear points in the parachain protocol.

When all validators' votes have been recorded, and it is found that the winning opinion is less than a certain number of votes (the detailed parameters may be up to half, or less), then it can be assumed that the parachain has an unexpected hard fork, this The consensus of the parachain will be automatically suspended. Otherwise, we assume that

malicious behavior has occurred and punish those validators who voted "yes" for the losing opinion.

The conclusion is that only enough signature votes can reach consensus, and then the blocks of the relay chain are packaged and the next block is packaged.

6.5 Improvement in the relay chain block packaging

The method of packing blocks ensures that the system works properly because the key information of each parallel chain is available by more than a third of the verifiers, so it doesn't scale well.This means that as more parallel chains increase, the work for each validation increases.

In an open consensus network, how to ensure data availability is an open problem, but there are still ways to alleviate the performance bottleneck of verifier nodes.One simple solution is to verify that people are just responsible for verifying the availability of the data, and then they don't need to really store, communicate, and copy the data themselves.The second scheme is data isolation, which is likely related to how the collector organizes the data, and the network can have some interest or income incentive for the collector to guarantee that the data provided to the verifier is available.

However, this scheme may bring a bit scalable, but still does not solve the fundamental problem.Because adding more parallel chains usually requires increased verifiers, the consumption of network resources (mainly bandwidth) grows at the square of the total chain number, which is unsustainable in the long term.

Ultimately, we may consider the fundamental constraints on ensuring the consensus network security where the demand for bandwidth grows by the number of verification multiplied by the total number of message entries.We cannot trust consensus networks that separate data stored at different nodes because it separates data from operations.

6.5.1 Delency introduction

The way to simplify this rule is to first understand the concept of immediacy. 33%+1 of validators eventually need to vote on the validity of the data, rather than immediately (immediately), we can better use

the characteristics of data exponential propagation to help cope with the peak of data communication. A reasonable equation (though not proven):

(1) Delay = number of validators * number of blockchains

Under the current model, the scale of the system can only be scaled with the number of chains to ensure distributed computing of data; because each chain requires at least one validator, we have reduced the complexity of availability voting to only It has a linear relationship with the number of validators. Now that the number of validators can grow as much as the number of chains, we are over:

(2) Delay = Quantity 2

This means that as the system grows, the increase in bandwidth and latency within the network is known, but the number of blocks required to achieve final certainty is still increasing by the square. This problem will continue to haunt us, and it may also force us to create a "non-flat" architecture, that is, there will be many Rainbow Protocol chains arranged in a hierarchical structure, and messages are routed through a tree structure. .

6.5.2 Public engagement

The MicroOpinion (micro-complaints) system is a way that can promote public participation.There can be some external participants similar to anglers to regulate the veriators.Their task is to find the verifier who provide non-available data.They can submit a micro opinion to the other verifiers.This scheme requires a PoW or deposit mechanism to prevent witch attacks, otherwise it fails the entire system.

6.5.3 Availability Guarantee Person

One final scenario was to nominate a second group from the verias usability guarantor (Availability Guarantors).They also need to make a deposit as the regular verifier and may come from the same group (choosing them over a long period, at least a session).Unlike ordinary verifiers, they do not need to switch between parallel chains, but only to form a single group that regulates the availability of all important cross-chain data.

This scheme alleviates the equality relationship between the number of verification and the number of chains.The number of chains can eventually grow (along with the original group of validation), however participants can still maintain sublinear growth or constant growth, especially those involved in data availability verification.

6.5.4 Collector settings

An important aspect of the system to guarantee is the reasonable selection of collectors who make parallel chain blocks.If a parallel chain is controlled by a certain collector, it becomes less obvious whether the external data becomes available, and the person can attack more simply.

In order to distribute collectors as widely as possible, we can manually measure the weights of parallel chain blocks using a pseudo–stochastic approach.In the first example, we want to verify that people tend to select candidate blocks with more weighting, which is an important part of the consensus mechanism.We must also motivate veriators to find the maximum weight candidates who can allocate their rewards to proportional.

In the consensus system, to ensure that the collector's block is chosen with equal opportunities, we use a random number generator to weight each candidate block connecting all the collectors.For example, use the collector address and some cryptographically secure pseudo–random numbers for different or (XOR) operations to determine the optimal block (winning ticket).This gives each collector (more accurately each collector's address) the chance to beat others randomly and fairly.

The validation generates an address to the winning ticket closest through a witch attack, and to prevent this, we add some inertia to the collector's address.A simple way is to need their address to have a basic balance, and another more elegant way is to consider the balance of the address together to calculate the probability of winning.The modeling is not done yet, and we will likely allow few balances to be collectors.

6.5.5 The block is overweight

If a collection of veriators is attacked they may generate a block that is valid but takes a lot of time to execute.This question comes from specific difficult data questions such as large mass factorization

problems etc, verifier groups may take a very long time to solve the answer, and if someone knows some shortcuts, their candidate blocks have a huge winning advantage.If a collector knows that information and everyone else is busy counting old blocks then he has a big advantage for his candidate blocks win.We call this type of called overweight (overweight) blocks.

To prevent verifiers from submitting these overweight blocks that substantially exceed normal blocks, We need to add some warnings: because the time taken to execute a block is relative (depending on how much it is overweight), So there will be three possible votes: the first is that this block is definitely not overweight, More than 2 / 3 of validation declare that they can finish within a certain period of time (e. g. 50% of the block time); The other is that this block is absolutely overweight, More than 2 / 3 of validation declare that they cannot execute the block within a limited time; Another thing is that the differences of opinion are basically flat, We will do some punishment in this case.

To ensure that verifiers can predict whether the block they submit is overweight, they may need to publish their execution performance on each block.After a while, they can evaluate the performance of their processor by comparison with other nodes.

6.5.6 Collector Insurance

Another question is left to the reviewer: in order to check the effectiveness of the collector block, they can not calculate the transactions themselves, but have to do like the PoW network.Malicious collectors can fill illegal or overweight blocks to veriators to obtain substantial potential opportunity costs by victimizing them (wasting their resources).

To prevent this, we provide a simple strategy for the veriator.First: The parallel chain candidate block sent to the verifier must be signed with a rich relay chain account, and if not, the verifier will discard the block immediately.Second: These candidate blocks are sorted by a combinatorial algorithm (or multiplication), with factors including account balances above a certain limit, the number of blocks that collectors have successfully submitted in the past (excluding those with penalties), and

the proximity of the winning ticket.The limit here should be equal to the penalty for filing the illegal block.

To warn the collector against sending illegal or overweight transactions to the verifier, any verifier can pack a transaction in the next block, indicate the illegal block, and transfer the collector part or all of the balance to the injured verifier.Such transactions have higher priority than other transactions, preventing the collector from transferring his balance prior to punishment.The amount of punishment may be dynamically determined and most likely part of the authenticator block reward.To prevent the examiner from arbitrarily forfeiting the collector's money, the collector may appeal the examiner's decision, form a random jury of the verifier, and make some deposit.If the jury found the verifier reasonable, the deposit was given to the jury.If unreasonable, the deposit is returned to the collector, and the verifier is punished (because the verifier is the core role and the punishment is heavier).

6.6 Cross-link transaction routing

Cross-link transaction routing is the core function of the relay chain and its validation.The main logic is managed here: how a submitted transaction (in short, a "commit") is forcibly routed from the exit of one source (source) parallel chain to another target (destination) parallel chain without any trust person.

We carefully selected the above words; in the source parallel chain, we do not need to explicitly bind this submitted transaction.The only constraint in our model is that parallel chains must try to package to their full export capabilities, and these submissions are the result of their block execution.

We organize these submissions with a first-in-first (FIFO) queue.The number of queues serving as a routing baseline (routing base) may be about 16.This number represents the parallel chain performance that we can support directly, without using multi-phase (multi-phase) routing.Rainbow Protocol will support this direct routing at first, however we may also adopt a multi-phase routing operation (hyperrouting hyper-routing) as a way of future system expansion.

We assume that all parties are aware of the validation grouping case for the next n,n + 1 in two blocks.In summary, the routing system has the following stages:

• Collector s: the validator V[n][S] among the contract members.

• Collector s: FOR EACH Team s: Ensure that there is at least one validator V[n][S] in the contract.

• Collector s: FOR EACH Team s: Assuming that the exit [n−1][s][S] is available (all the data submitted to S in the previous block)

• Collector s: Construct candidate block b for S: (b.header, b.ext, b.proof, b.receipt, b.egress).

• Collector s: Send proof information proof[S] = (b.header, b.ext, b.proof, b.receipt, b.egress).

• Collector s: Ensure that the external transaction data b.ext is already available to other collectors and validators.

• Collector s: FOR EACH group s: Send export information egress[n][S][s]= (b.header, b.ext, b.receipt, b.egress) to the receiver group of the next block ' S validator V[n+1][s].

• Validator v: members of the same group pre−connected to the next block: let N = Chain[n+1][V]; connect all validators so that Chain[n+1][v] = N.

•Verifier v: Collect all entry data of this block: FOR EACH Group s: Retrieve the exit egress[n−1][s][Chain[n][V]], get Chain[n] from other verifier v [v] = Chain[n][V]. It may be by randomly selecting the proof data of other verifiers.

•Verifier v: Receive the export data of the candidate block for the next block: FOR EACH group s, receive egress[n][s][N]. Vote on the validity of the block exit; re−publish among the intention validators so that Chain[n+1][v] = Chain[n+1][V].

• Verifier v: Waiting for consensus.

Representative: In block n, current export queue information from the source from parallel to the target to parallel.The collector s belongs to

the parallel chain S.verifier V[n][s] is the verifier group of parallel chain s in block n.Instead, Chain[n][s] is the parallel chain to which the verifier v belongs in the block n.block.egress[to] is an export queue that is sent from the parallel chain block block to the target parallel chain to.

The Collector wants to adopt the block they produce, so the collection (transaction) fee serves as an incentive and ensures that all the target group members of the next block are aware of the export queue of the current block.The validation person incentive is to reach a consensus on the relay chain block, so they do not care about which collector block to eventually adopt.In principle, a verifier can collaborate with a collector to reduce the probability of adopting other collectors. However, because the verifier of the parallel chain is randomly assigned, this is also difficult to succeed, and it may be reduced by fees, which ultimately affects the consensus process.

6.6.1 External data availability

If you want to complete all distributed processes in a decentralized system, a legacy for years is: how to ensure that the external data of a parallel chain is available.The core reason of this problem is that it is impossible to generate a non-interactive proof of availability or not.Within a Byzantine fault-tolerant system, we need to rely on external data to verify the validity of arbitrary transactions.Assuming that the maximum number of Byzantine nodes we can tolerate is n,, we need at least 1 node of n + to prove the availability of the data.

Rainbow Protocol is a system hoping to be scalable, which raises the question: if a fixed percentage of verifiers has to prove the effectiveness of the data and assume they really store the data for judgment, how can we avoid the increasing demand for bandwidth / storage space such as the system grows.One possible answer is to establish a group of witnesses (guarantors) whose numbers grow linearly as Rainbow Protocol grows as a whole.This is mentioned in 6.5.3.

We also have a second trick.Collectors have an inherent incentive to ensure the availability of all data, otherwise they can no longer produce subsequent blocks and get a handling fee.Colleccollector can also form a group of complex and diverse (because the parallel chain verifies the randomness of members) and is difficult to access.Allow recent

collectors (possibly recent thousands of blocks) to challenge external data from a parallel chain block to get a reward for a bit of verifier.

The verifier must contact these groups with obvious offensive actions that will prove, obtain, and return the data to the collector, or upgrade the situation directly by proving its nonavailability (as the plaintiff directly refuses to provide the data record, the verifier of the misconduct will directly disconnect) and contact more verifiers to test together.In the latter case, the collector's deposit would be returned.

Once more than the legal number of verifiers prove the nonavailability of the transaction, the verification team can be dissolved, the collector team of illegal acts is punished and the block retracted.

6.6.2 Routing to "Submit."

The head of each parallel chain contains an outlet tree root (egress-trie-root).This root contains a lattice list of routing information submitted exit each with a serial (concatenated) structure.A Meckel tree proof can be provided between the verifiers of a parallel chain, thus proving that the block of a certain parallel chain corresponds to the exit queue of another parallel chain.

Before starting processing a parallel chain block, the exit queue of each parallel chain specified block is incorporated into the entry queue of our block.It is assumed that pseudo-random numbers (CSPR) with cryptographic security can be used to guarantee a fair pairing of parallel chain blocks.The collector calculates the new queue and extracts the exit queue according to the logic of the parallel chain.

The contents of the entry queue are explicitly written to the parallel chain block.This has two purposes: First, parallel chains can perform non-trust synchronization independently without relying on other chains.Second, if the entire entry queue cannot be processed within a block, this approach can simplify data logic; verifiers and collectors can continue to process the following blocks without making data references.

If the entry queue of the parallel chain exceeds the threshold for block processing, it is marked as full on the relay chain, and no new message is received until the queue is emptied.The Meckel tree is used to prove that the collector's operation in parallel chain blocks is credible.

6.6.3 Abuse

The minor flaw in this architecture is the possibility of post-bomb attacks (post-bomb attach).All the parallel chains send a maximum number of submissions to another parallel chain, which instantly fills the entry queue of the target chain and makes a Dos attack without any damage.

Normally, assuming that there are non-malicious collectors and verifiers for N parallel chains and a series of normal synchronization, then a total of N x M authenticators are required, L collectors for each parallel chain, and the possible data path (data path) for each block have:

Validation By: M-1 + L + L:M-1 represents other verifiers in the parallel chain collection, the first L provides a parallel chain candidate for each collector, and the second L represents the front block data that the whole collector of the next block needs to put in the export queue.(The latter case may be worse because the data is shared between collectors).

Collector: M + kN:M represents the number of validation connections related to each parallel chain block, and kN represents the load of the next block seeding (probably some favorite collectors) to the exit queue of each parallel chain validation group (seeding).

Thus, the possibility of each node data path grows linearly with the complexity of the system.This is also reasonable that when the system scales to hundreds of parallel chains, the delay of communication becomes larger, thereby reducing the growth of complexity.In this case, a multi-level routing algorithm is used to reduce the data path for the peak period, but caching and transaction delays are introduced.Hypersquare routing (Hyper-cube Routing) is a new mechanism that can be built on the underlying routing methods described above.For nodes, their number of connections grows from growing with parallel chains and node groups to logarithms with only the number of parallel chains.This may require queues through multiple parallel chains to eventually deliver submission.

Routing itself is simple and certain.We start with limiting the number of lattices of the entry / exit queue; the total number of parallel chains is routing-base (b), the number fixed with the parallel chain changing to routing-exponent (e).Under this model, our total number of messages

grows by O (be), while the data path remains constant, and the delay (or the number of blocks needed to pass) grows by O (e).

Our routing model is a e dimensional hypercube with b possible positions.For each block we route the message around an axis.To ensure the delivery delay of the worst-case e blocks, we use round-robin fashion to rotate each axis.

As part of the parallel chain processing, once given the current block height (routing dimension), a message outside the entry queue is immediately routing to the appropriate exit queue grid.This process requires sending more data on the transfer route however this can be a problem and may be solved by some alternative data load sending methods such as containing only one reference and not in the commit tree (post-

- •Sub0: If Mdest∈ {2,3}, then sendTo(2), otherwise keep
- •Sub1: If Mdest∈ {2,3}, then sendTo(3), otherwise keep
- •Sub2: If Mdest∈ {0,1}, then sendTo(0), otherwise keep
- •Sub3: If Mdest∈ {0,1}, then sendTo(1), otherwise keep
- Phase 1, for each message M:
- •Sub0: If Mdest∈ {1,3}, then sendTo(1), otherwise keep
- •Sub1: If Mdest∈ {0, 2}, then sendTo(0), otherwise keep
- •Sub2: If Mdest∈ {1,3}, then sendTo(3), otherwise keep
- •Sub3: If Mdest∈ {0, 2}, then sendTo(2), otherwise keep

The two dimensions here are easily seen as the first two bits (bits) of the target index.The second block handles the low-order bits.Once all occurs (in any order), the submission is routed.Maximize the randomness (Serendipity).A modification to the basic proposal is to fix the number of validation to c2-c and c-1 per group.Excluding the original scheme of loosely allocating the verifier between the parallel chain in each block, for each parallel chain group, in the next block, each verifier is assigned to the unique different parallel chain group.This results in unvariability between two blocks, and for arbitrarily paired parallel chains, two veriators swap their duties.However, we cannot use this to ensure

absolute availability (individual validation may often offline, even non–malicious), but can optimize this scheme.

This scheme will also have sequelae.The parallel chain requires to restructure the verifier set.In turn, the number of people is bound to the square level of the number of parallel chains, eventually growing rapidly from few beginning, and becoming unaffordable at 50 parallel chains.None of these are any essential questions, and for the first question, it was also needed to frequently restructure the sets of verifiers, regardless of the number of verifier sets.When the number of sets is few, multiple verimay be assigned to the same parallel chain, then the factors affecting for the full parallel chains are constant.For the problem requiring many verifiers in many parallel chains, this can be mitigated by this multi–stage hypersquare routing mechanism discussed in 6.6.3.

6.7 Verification of the parallel chains

Valators mortgaged a large amount of deposits, and their primary goal is to verify whether the parallel chain block is valid, including but not limited to: state conversion, including external transactions, execution waiting for submission in the entry queue, and executing the final status of the exit queue.The process itself is relatively simple.Once the validation completes the packaging of the previous block, they are free to prepare candidate blocks for the parallel chain for subsequent rounds of consensus.

The reviewer initially finds a parallel chain block through the parallel chain collector (described below) or one of his deputy reviewers.The data of the parallel chain candidate blocks contains block heads, front block heads, external data input (called transactions for Ethereum and Bitcoin, however they may also be arbitrary structure, arbitrary purpose), export queue data, internal proof data of state conversion effectiveness (for Ethereum this may be many state / storage tree nodes used to perform each transaction).Experimental evidence shows that this dataset has up to several hundred K bytes (KiB) for the current Ethereum block.

If the verification is not complete, the verifier tries to obtain the relevant information from the transformation of the previous block, starting with the verifier of the previous block, and then going to all those who have signed the data.

Once a verifier receives such a candidate block, they validate it locally.The validation process is included in the validation module of the large class of parallel chain, the consensus-required software module must be written in all Rainbow Protocol implementations (in principle share a C ABI library in multiple implementations, which reduces security as they are just references to a single implementation).

This process extracts the front header and then tested by the hash values recorded in the relay block just reached consensus.Once the validity of the parent block header is validated, a specific validation function in the parallel chain class is called.This is a function that receives many data items (probably several given right now), and the return value is a simple judgment of whether the block is valid.

Most of these verification functions will first examine the data items of the head that can be derived directly from the parent block (such as the parent block hash, height).Then, in order to process transactions or submissions, they will try to populate the internal data structure.For a blockchain like Ethereum, all transactions are needed to fill such a lot of data into the Meckel tree.Other types of blockchains may have other treatments.

Once verification is completed, both the entry submission and external transactions (or otherwise represented) are fixed according to the rules of the chain.(A possible default is for all entry submissions to be handled before the service external transaction, however this should be determined by the logic of the parallel chain).With this provision, a series of export submissions are created and do meet the collector's candidate block requirements.Eventually the reasonably filled block heads and candidate block heads are checked together.

After the verification completes the candidate block, he votes the block hash and sends the necessary verification information to the other deputy veriin the group.

6.7.1 Parallel chain collector

Parallel chain collectors do not need to pay a deposit, they complete tasks similar to the miners in the current blockchain network.They belong to specific parallel chains.To work, they must have fully synchronized relay and parallel chains.

The exact meaning of full synchronization depends on the type of parallel chains, although they all contain the current state of the parallel chain entry queue.In the case of Ethereum, it should also have at least some recent blocks of Meckel tree databases, but may also contain very many other data structures, such as Bloom filters proving the existence of the account, genetic (familial) information, log output, and bifurcated retreat forms for corresponding height blocks.

To keep the two blockchains synchronized, they must maintain a trading pool to "catch" (fish) transactions and receive transactions properly verified on the public net.With the chain and trading pool, the collector can pack the new candidate blocks for the selected verifier of each block (knowing they are identified due to the relay chain), affiliated with some necessary information (e. g. proof of validity from the node network, etc.), and submit it to the verifier.

They collect handling fees for all transactions in return.There are lots of economic incentives here.In a highly competitive market, if collectors have surplus, share fees with parallel chain validation to motivate them to pack a specific collector block.Likewise, some collectors may raise the fees required to pay, making the block more attractive to the reviewer.In this case, the normal market mechanism allows the transaction with higher fees to skip the queue and pack it into the chain faster.

6.8 Network design

The network design requirements in the traditional blockchain such as Ethereum and Bitcoin are generally relatively simple.All transactions and blocks are unbooted and broadcast in gossip.There will be more involved in the synchronization module and Ethereum can respond differently depending on the category, but in reality this is more of the node strategy than the content of the protocol itself.

Ethereum improves the current network protocol based on the devp2p protocol, supports multiplexing of multiple subprotocols in a single node connection, and thus supports multiple p2p protocols simultaneously, but Ethereum's protocol remains relatively elementary, and it has not yet completed important functions such as supporting QoS.The desire to create an omnipresent "web3" agreement basically failed, with only a few projects crowdfunded from Ethereum.

The Rainbow Protocol demand is even more fundamental.Compared to a complete unified network, Rainbow Protocol has a variety of participants, each with different needs, and participants need to have a lot of different network channels to exchange data.Essentially, this means the need for a protocol that can support a more hierarchical network structure.In order to promote more new types of blockchain to expand the network, the new hierarchy.

A deeper discussion of the network protocol is not within the scope of this paper, and we need more demand analysis.We can divide the network participants into two classes (relay chains, parallel chains), each with three small classes.Participants in each parallel chain communicate with each other without the other chains:

•Relay chain participants

•Verifier: P, divide each parachain into multiple subsets P[s]

•Availability Guarantor: A (replaced by a validator in the basic agreement)

• Relay chain client: M (member of each parachain)

• Parachain participants:

• Parachain collectors: C[0], C[1],...

• Parachain fisherman: F[0], F[1],...

• Parachain client: S[0], S[1],...

• Parachain light client: L[0], L[1],...

Often we believe that several communication occurs between network members and their settings:

- • P | A<–> P | A: In order to reach a consensus, the validator/guarantor must be connected.
- •P[s]<–> C[s] | P[s]: Each validator who is a member of the parachain will connect with other members to discover and share blocks, such as collectors.
- •A <–>P[s] | C | A: Each availability guarantor will need to collect signed consensus-related cross-chain data from the validator; the collector can broadcast to the guarantor to optimize the consensus

on their block . Once completed, the data will be broadcast to other guarantors to promote consensus.

- • P[s]<–> A | P[s']: Parachain verifiers will need to collect additional input data from the previous verifier or availability guarantor set.
- •P[s]<–> A: When it is necessary to report, the phisher will announce to any participating party.
- •M <–> M| P | A: The relay chain client outputs data to the verifier and guarantor.
- •S[s]<–> S[s] | P[s] | A: Parachain client outputs data to validators and guarantors.
- •L[s]<–> L[s] | S[s]: The parachain light client obtains data from the full client.

That different flat–layer network (like–Ethereum devp2p) per node is no longer adapted to ensure efficient transmission.It is likely to extend in the protocol to introduce a reasonable node selection and discovery mechanism, and it may also plan some forward–looking algorithms that guarantee that the order of the nodes is "accidentally" connected in due course.

Specific strategies for different types of party nodes will vary: for a scalable multi–chain system, the collector either needs to continuously reconnect the selected reviewers or connect a group of reviewers to ensure that they never break their lines, even if most of the time they are useless for themselves.Collectors also maintain one or more stable connections to a collection of availability guarantees to ensure rapid dissemination of consensus data.

Availability guarantees will remain interconnected and maintain stable connections to the verifier (parallel chain data for consensus and consensus), some collectors (for parallel chain data), some anglers and some full nodes (for missing information).Examators tend to find other reviewers, especially those in the same group, and collectors who can provide parallel chain blocks.

Phlers and general relay or parallel chain clients tend to maintain a connection with the verifier or guarantor, but many nodes similar to them do.Parallel chain light client in addition to connecting to other light client, also connects to a parallel chain full client.

6.8.1 Problem with node rotation

In the plan of the basic protocol, the verifier group of each block is randomly transformed, and the verifier is randomly assigned to verify the transaction of a certain parallel chain.How to transfer data between unrelated nodes will be a problem, which must rely on a fully distributed and well-connected node network to ensure that the required jump distance (the worst delay) grows only at the log level of the network size (a similar Kademlia protocol will help), or you must extend the block time to support the necessary connection negotiation and establish a node collection connection that can meet the current communication needs of the node.

None of these are good solutions: forcing longer outs prevents the network from supporting some specific programs or blockchain.Even a perfectly fair network connection leads to bandwidth waste, because pushing a lot of data to unrelated nodes affects the scaling function of the network.

However, these directions facilitate the problem, an optimization scheme that can reduce the susceptibility of the set of parallel chain verifiers, reredistribution after a block (such as 15 block, if 4s block time, only reconnect), or rotate only one verifier for a period of time (for example, if a parallel chain allocates 15 verifiers, then the average situation is all rotation in one minute).By increasing the local predictability of parallel chains, to reduce the number of node rotation, and still guarantee the advantages of connections, we can guarantee the randomness of connections between nodes.

6.8.2 Path to an efficient network protocol

The most efficient and reasonable direction of development is to focus on transforming an existing protocol rather than developing one from scratch.Several peer-to-point protocols we will explore include: GNUnet. of libp2p, GNU of devp2p, IPFS of EthereumThis paper does not do much about a comprehensive introduction of these protocols and how to

build a modular node network, dynamic node transformation, scalable subprotocols, but this will be an important step towards implementing Rainbow Protocol.

7 Practiirability of protocol

7.1 Cross-chain transaction payments

In addition to the gas mechanism of computing resource statistics like Ethereum, which brings more freedom and simplicity, it also raises the important question: how does no gas, parallel chain prevent other parallel chains from forcing them to do operations?However we can rely on the "transaction commit" entry queue cache to block one chain from filling another with transaction data and we have not yet provided other anti-garbage mechanisms for the same effect.

This is a higher level of the problem to be solved.Since the chain can attach any data to the "transaction submission," we need to guarantee the calculated fee before starting.Similar to a proposal in the Ethereum Serenity version, we can imagine the parallel chain having an "introverted" (break-in) contract to provide verifiers with guarantees paid in exchange for a specific amount of preallocation of computing resources.These resources may be measured by gas-like mechanisms, but may also be using completely different conceptual models such as subjective computational time models or general Bitcoin-like billing models.

Whatever value model the "introverted" contract defines, we cannot simply assume that under-chain invocation are available to them, so the scheme is not useful.However, we can imagine a second "outgoing" (break-out) contract in the source chain.The two contracts form a bridge to recognize and provide equivalent exchange (equity tokens can be used to pay for settlement).Calling other chains means using this bridge as a proxy to negotiate how to pay for the computational resource consumption of the target parallel chain.

7.2 Add a chain

Adding a parallel chain is relatively inexpensive operation but not free.The more parallel chains means the fewer verifiers for each parallel

chain, and the more verimeans less deposit per person.The problem of forcing a parallel chain is alleviated through the fishermen.Due to the problems of the consensus mechanism itself, the growing set of veriators essentially leads to higher delays.Each parallel chain may have a great burden on verification algorithms for unfortunate veri.

Therefore, the verifier and / or other equity holders will add a new chain to pricing.The markets for this chain are either two types:

• Chains that have no net contribution to the network (by locking coins or burning equity tokens, such as alliance chain, Doge chain, specific application chain);

• By providing some functions that are not available elsewhere, a chain that can bring more value to the network (such as privacy, internal scalability, built-in services)

We will eventually incentivize stakeholders in the community to add sub-chains-through economic means or according to the degree of willingness to add functions to the relay chain.

It can be expected that the new chain just added will have a short removal period, which allows the new chain to be tested first without any compromise and long-term value risk.

8 Distribution and management

8.1 Operation and management

Rainbow Agreement [Rainbow Protocol]'s team, as a believer of Sir Tim Bernard Lee (Sir Tim Berners Lee), we are convinced that from the first day of the agreement it belonged to all humans, not the tool a small percentage used to make profits.Rainbow Protocol(, Rainbow Agreement) The main mission of establishing the foundation in London is open, impartial and transparent and does to operate the Rainbow Protocol network for profit and to support Rainbow Protocol's development team.Rainbow Protocol Foundation is approved by the London Accounting and Business Authority (ACRA) and regulated by the London Companies Act, which is operated independently managed by a board of trustees or management board of trustees and independent of the government.London is known for its stable and sound legal and financial

environment, and Rainbow Protocol Foundation is a nonprofit organisation (Non-Profit Entity) founded in London, a legally formed organisation to support or participate in the public or private interest without any commercial interest.The "profits" earned by the foundation, known as a surplus, will be continuously retained as funding for other activities without distributing the profits among its members.

8.2 Governance structure and voting

In order to make Rainbow Protocol Foundation make reasonable use of the foundation's funds and resources under the premise of openness, justice and transparency, constantly promote the rapid development of the rainbow protocol Rainbow Protocol protocol, expand the application scenarios of the Rainbow Protocol protocol, and absorb more institutions, companies and organizations to enter the open source rainbow protocol Rainbow Protocol ecology, the foundation has set up a three-layer organizational structure

8.3 Technology R & D Department

The technology R & D department is responsible for the development and audit of the underlying technology, and is the basic department of the foundation.For true

The guarantee team maintains information exchange and consistent step, and the technology research and development department should work with other departments (in particular

Product design and production department) to exchange information, timely adjust and communicate the project details, and determine the next stage of research

Hair direction.

8.3 Product Design and Production Department

The product design and production department is responsible for the enrichment and improvement of the product framework provided by the technical department, establishing sustainable specific development strategies, including conducting market research, coordinating the product functions, and undertaking the UI design and image design of Rainbow Protocol.Members need to always understand the dynamics, hot spots and feedback of the community, actively communicate with the token

holders, and hold technical exchange meetings and other activities from time to time.

8.4 Ecological operation Department

Based on the technology and product departments, the ecological operation department is responsible for "one outside and one inside" — — first, extend the work deep, actively develop partners, closely connect Rainbow Protocol, end users, partners, so as to create an open, distributed, privacy protected global entertainment ecosystem chain; and secondly, build the community ecosystem, form a benign interaction, free flow of information and fully symmetrical user community.

8.5 Marketing and Promotion Department

The Marketing Department is responsible for promoting the core or derivative products and services of Rainbow Protocol, including but not limited to contacting media cooperation, advertising, design user interaction, etc.The department will work closely with the ecological operation department to develop the most appropriate publicity plan according to the requirements of the partners and end users.

8.6 Token issuance and management

Token name: RPW (Rainbow Protocol)

Total distribution: 99,999,999 RPW

Distribution and distribution method: no pre-excavation, no lock warehouse, no private placement, all tokens are issued through DeFi mining, and the initial issuance limit is 5,000,000 pieces. According to the consensus of RPW and the development of the community, RPW will reasonably release a more flexible and perfect decentralized financial mechanism

9 Conclusion

We propose a possible direction for a heterogeneous multichain protocol that is scalable and capable of being backward compatible with currently existing blockchain networks.Under this agreement, the parties jointly create a complete system for their own benefit that can be extended in a very free way and without the current inherent cost of those ordinary

blockchain to the user.We give a general outline of this architecture, including the required actors' roles, their economic incentive models, and the actions they need to do.We have identified a basic design and discussed its strengths and limitations; our future direction is to remove these limitations and move towards a fully scalable blockchain scheme.

9.1 Missed materials and open questions

Network design is generally isolated from protocol implementations.It is not fully discussed how the network is recovered from such experimental conditions.Networks require a deterministic non-zero time and they recover from relay chain bifurcations should not be a big problem, however they need to be carefully integrated into the consensus protocol.

This paper also does not specifically discuss the deposit forfeiture and the corresponding reward rules.Now we assume the winner-takes-all: but probably the best incentive.A short-cycle submission-disclosure process supports many fishermen to claim bounties and then reach a fairer reward distribution system, but this process of reporting malicious behavior will also introduce greater delay.

9.2 Express gratitude

Thanks to the proofreaders who helped publish this framework article.Specifically mention Peter Czaban, Ken Kappler, Robert Habermeier, Reto Trinkler and Jack Petersson.Thanks to those who contributed the idea, specifically mention MarekKotewicz and Aeron Buchanan.Thanks to all the others who have helped.All the errors are still mine.

Part of this work, including an initial survey of the consensus algorithms, is funded by Point72 and the Quantum Fund.