EMIT White Paper (v1)

Edward Yourdon, Isak Andik

EMIT Foundation

Abstract. In this paper, the products, technologies and routes involved in the three systems mentioned in "The EMIT Parchment Paper" are analyzed and refined, and the CROSS, EPOCH, EMIT Core and GDP Market protocols in the systems are discussed and researched in depth. Finally, appropriate implementation methods and plans are given as guidance for actual work on each protocol. This White Paper will also be supplemented and revised in real-time in the process of implementing the whole EMIT Decentralized World based on the actual situation and ongoing research results.

Keywords: DW · DeWe · Blockchain · EMIT · EPOCH · CROSS · EMIT Core · Economy

edward@emit.technolog, isak@emit.technology

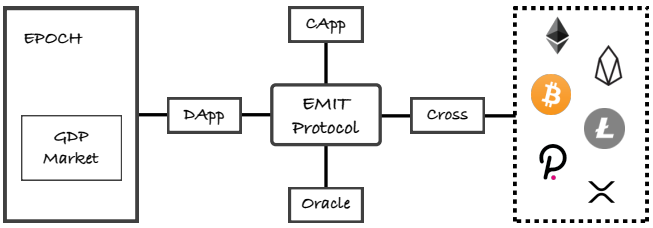
Table

I. Introduction

Guided by the "The EMIT Parchment Paper", EMIT represents three hierarchical concepts:

- First, it is a decentralized world with the real human world as the economic entity, an extension of the real-world economy. Its internal economic structures and relationships beyond the real world can form tremendous Gross Domestic Product (GDP).
- Second, it represents the three major decentralized systems in the virtual world.
    - The EMIT Core Protocol network is a decentralized infrastructure providing flexibly extensible TPS and the cross-chain capabilities of mainstream assets, a framework for easy development of complex smart contracts, and diverse privacy-protected assets.
    - The EMIT Ecosystem cluster will bring diverse economic activities and complex economic structures.
    - The GDP Market will bring convenient trading for diverse economic activities and assets.
- Finally, EMIT represents the foundation for these items, including some of the key decentralized and centralized technologies and social conditions from the web 2.0 era to the present.

According to the version plan in "The EMIT Parchment Paper", during the Prelude Period, we will start community recruitment and realize some cross-chain functions. Throughout the Ancestors Period, we will then gradually implement the EPOCH Ecosystem, EMIT Core Protocol Network, and GDP Market. In the National Period, we will push the EMIT decentralized world to its peak.



In addition to making it easy for developers to develop complex decentralized applications, the EMIT Core Protocol also makes it easy to integrate a variety of decentralized and centralized systems, such as Oracle, cross-chain (Cross), and centralized applications (CApps). Supported by EMIT Core's flexibly extensible TPS and low-latency transactions, the experience of decentralized applications (DApps) will compare favorably with centralized systems.

II. CROSS

In the EMIT decentralized world, all kinds of encrypted blockchain assets will be the cornerstone of the EPOCH ecosystem, so Cross is a basic capability of the EMIT Core Protocol and a driving force of EPOCH. According to "The EMIT Parchment Paper" plan, the EMIT Core Protocol will be officially launched during the Clan Period. Although the EMIT Core protocol itself has the features of Cross and Oracle, the EPOCH ecosystem

will start in the Chaotic Period. The ecosystem at this stage requires support from the cross-chain function. Thus, we will first build a decentralized cross-chain protocol as required for the EPOCH ecosystem, named Cross. In this chapter, we will choose the most reasonable technical solution for Cross, which will migrate to the EMIT Core Protocol after its completion.

2.1. Summary of the Cross Scheme

There have always been two different understandings regarding cross-chain schemes. One is an atomic swap of assets between the two blockchain systems, where the total amount of assets in each system remains unchanged after the transaction. The other is asset transfer between two systems. After the transfer, the liquidity of the transferred assets corresponding to the systems changes, but the sum of their liquidity doesn't change. Cross refers to the latter.

Ever since the InterLedger protocol was proposed by Ripple in 2012, dozens of different technical solutions have appeared for cross-chain transfer between asset types, but in terms of the fundamental principles, these schemes can be divided into two categories, depending on how they are confirmed: notarization and relay.

A. Notarization

The scheme requires a group of trusted organizations in a decentralized context to monitor both blockchains and correctly complete the predefined transfers when asset locking is detected. Notarization has two technical implementation methods, based on the implementation details: multi-signature and distributed signature.

- In the multi-signature mechanism, multiple notary nodes supervise the locked account at the same time. Only after providing a certain number of correct signatures can the account complete the transaction. This method requires both blockchain systems to support the multiple signature mechanism or have a smart contract function. This model is used by projects like ChainBridge, PalletOne, and Ren, etc.
- The distributed signature mechanism makes use of the linear characteristics of an elliptic curve operation to disperse the private key of the asset-locked account to different notaries using signature thresholds. A certain number of notaries must be gathered in order to complete the signature. This approach requires both blockchains to have several fixed Elliptic Curve signature account system modes, such as ECDSA or EDDSA. This is the approach used by Wanchain, Fusion, and tBTC.

The advantage of this scheme is that notary accesses the cross-chain system in the form of a client, and does not intrude into the original chain. This approach is technologically inclusive and flexible, and thus can adapt to the vast majority of current blockchain systems. Its disadvantage is that a third party besides the two chains is required for notarization, which creates an additional possible risk point.

B. Relay Chain

The system chain or the smart contract of two blockchain chains using the relay scheme include each other's lightweight block maps, through which the

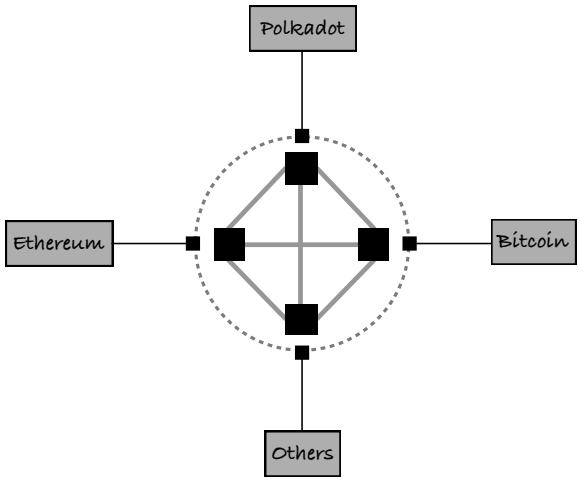
validity and authority of the corresponding other block can be verified. When both the sides produce the blocks, the block's mapping information can be submitted to the other party's smart contract by any node, and the block verification process can be simulated by the other party's smart contract. In this way, other contracts deployed on both blockchains can obtain the status information of the other to verify that the opposite contract has actually received the locked amount, and then complete the subsequent transfer. Hub of BTC-relay and Cosmos, and RelayChain of Polkadot are implemented this way.

The advantage of this scheme is that no third party is needed. Any individual can submit block mapping information to complete the relay. The downside is clearly that it's intrusive. Either the relay scheme must be taken into account and the interface set aside when the chain is designed, or it must have smart contracts. Each block must be submitted to each other after generation, taking up space and consuming a lot of fees for the chain that needs to pay the commission charges. Furthermore, if one chain changes its consensus protocol, the opposite relay chain needs to be rewritten. Another problem is that if a longer fork appears in the original chain of the relay, and the chain carrying the relay mapping has been identified and the funds are released, it will split, resulting in unpredictable results.

2.2. Design Principles

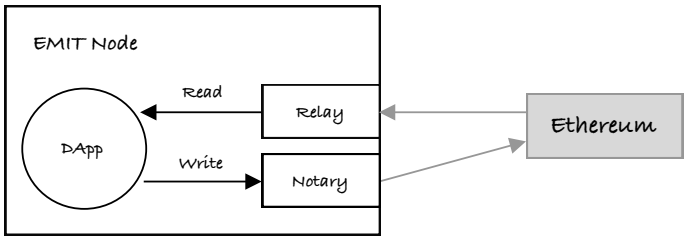
Due to the pre-requirements of the EMIT CROSS project, the following design principles are introduced before specifying the technical solution.

1. The security of the decentralized scheme is not significantly weaker than in the cross-chain system.
2. It is feasible and available in practice.
3. It has a wide range of applicability and can provide cross-chain function for most mainstream blockchains.
4. Cross-chain operation is possible between fungible tokens (FTs) and non-fungible token (NFT) assets.
5. The solution can eventually migrate to the cross-chain framework of the EMIT Core Protocol network.
6. Provides privacy protection to FT and NFT owners and related parties.
7. Another FT asset used as consensus maintenance is not used for the underlying system for now.
8. The design is simple and intuitive, and easy to implement and maintain.



Because Cross will eventually migrate to the EMIT Core Protocol network to accommodate all virtual asset infrastructure, and the EMIT Core Protocol network is independently integrated with other blockchains, Cross should also independently integrate with any third-party blockchain system.

Because the EMIT Core Protocol network will support cross-chain operations, it is natural to put the relays of various other blockchains into its application account. The diversity of other blockchains, as well as the uniqueness of the EMIT Core Protocol, make it difficult to link the protocol to other blockchains through relay, so a notarization mechanism is necessary to initiate instructions to other blockchains. The EMIT Core Protocol must be a hybrid system with both relay and notarization mechanisms. The status of other blockchains is made available to it via relay. Applications on EMIT Core can query the status of other blockchain systems as necessary, while its cross-chain instructions are sent to other blockchains in notarized form. As long as the notarized nodes of the EMIT Core Protocol are sufficiently secure, cost minimization can be used to drive other blockchains.



Since the cross-chain function of EMIT Node will finally adopta hybrid mode, the cross-chain component, as a cross-chain component before the release of the EMIT Core protocol network, it must implement notarization first in order to facilitate the subsequent migration. Cross-chain notarized nodes currently do not have a local state, and they can continue to exist as genesis nodes after the EMIT Core Protocol network is run.

2.3. Implementation Details

A. Decentralized Notarization System

The notarization nodes are network service programs running 24 hours online. Together, they constitute a notarization committee. The committee will be originally composed of several genesis nodes, and other later entrants must be approved by over 2/3 of the committee members. Over 2/3 committee votes can kick a node off.

Each cross-chain order is executed by committee vote. When the committee is small, a vote of more than 2/3 of the nodes is required for the order to take effect. When there are too many nodes, a temporary committee is generated for each cross-chain instruction using the VRF algorithm, and N rounds of committee voting (N>=1) are completed to finalize the results.

Since Cross, like EMIT Core, does not initially issue FT assets, a fee is charged for each cross-chain call to the corresponding asset as a fee to execute the cross-chain transaction. These fees are transferred to the account of the notary node that initiated the vote when the operation is completed.

B. Fairness and Security

Notarization committees have two main security challenges: selfish and malicious behavior. Selfishness means not actively providing notarization service, which reduces the effective resources of the cross-chain service. Malicious behavior means false or invalid votes. Such nodes can reduce the security of locked assets. Cross uses the following measures to prevent this from happening.

KYC for Genesis Nodes
In order to reduce the probability of node collusion in the initial stage, the EMIT Foundation will carry out strict KYC verification for the limited number of initial founding nodes, which can ensure the independence of their social associations to a certain extent, preventing malicious behavior against the initial system and ensuring the benign development of EMIT.

Asset Pledges
For a node to become a committee member and act as a cross-chain verifier of an asset, it needs to stake the asset as a bond, and operation must continue for at least a year before it can exit the position. Under normal circumstances, the pledged asset is returned to the node when it exits. However, in the case of malpractice, the committee may decide to confiscate some of its assets through a 2/3 vote, or to expel the malicious node. The pledged assets of the expelled node will continue to be pledged for the time previously set, until their maturity and return.

VRF
With more committee members, for each cross-chain command, the assets are locked and released by the temporary committee formed by the VRF algorithm.

Under normal conditions, votes by each node are weighted equally. Cross pays commission equally based on the behavior of each notary node, but their voting weight may be reduced in the following two cases.

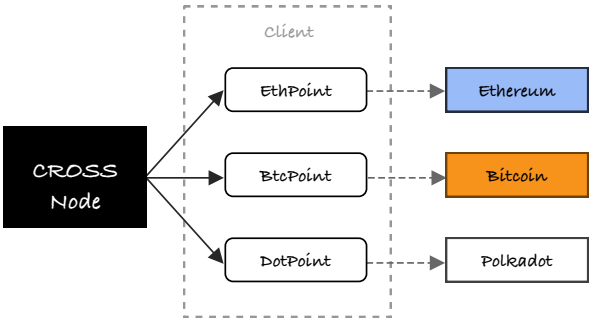
Committee Oversight
The committee periodically reports malicious nodes. If a node is reported by over 1/2 of the committee in a single cycle, then it is considered to be malicious, and its voting weight is reduced.

Negentropy Supervision
After the EPOCH ecosystem start, the Negentropy input by the EPOCH user (refer to the EPOCH chapter) selects which notarized node to exchange for Energy. The amount of Negentropy received directly affects the weight of the node voting in the committee.

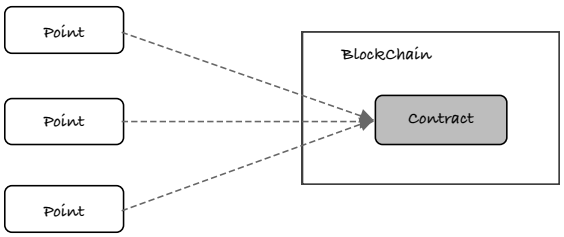
C. Point Components

The CROSS node is connected to different blockchains through components with the suffix "Point," such as EthPoint to Ethereum, BtcPoint to Bitcoin, and Polkadot via DotPoint. Each Point has a standard interface called by the Cross node. The Point component is equivalent to the client of a blockchain. Cross reads information off the chain and initiates cross-chain instructions through Point.

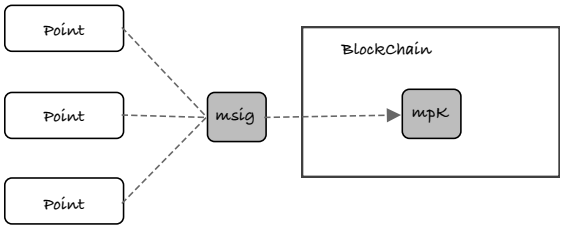
D. Multi-Signature Function



For blockchain systems with smart contracts, funds are locked in the smart contracts, and the Point components of different Cross nodes in the committee are registered in a particular smart contract and called according to decentralized rules, achieving a function similar to multiple signatures. Cross uses an aggregate signature mechanism to solve the problem that



multiple nodes vote too much on Gas. When the authentication node detects a new cross-chain transaction, its signature of the transaction is immediately broadcast. Any node can collect the signatures of others in the committee, and then aggregate them into a call parameter and submit it to the smart contract.



For blockchains without smart contracts, if the account and signature system supports a multi-signature function, the Point components of the different Cross nodes in the committee use the multi-signature function to access the account with locked funds.

However, for the notarization mechanism with multiple signatures, if the number of signatures changes, the public key of the locked account also changes. In this case, it's necessary for the notarized nodes of the committee to coordinate via consensus to generate another locked account to receive the assets of the original locked account in order to complete the membership or withdrawal.

2.4. Plan

During the Ancestors Period, EMIT will implement a decentralized cross-chain wallet, which will gradually connect to Ethereum, Bitcoin, Polkdot, Tron and other blockchains to drive the start of EPOCH with more assets on it. In the Clan and Ark stages of the Prelude Period, they will be migrated into the EMIT Core Protocol network. At the same time, we will look for and add a public chain that is secure enough to support both FT and NFT tokens as the Cross privacy plan, as well as a public chain that is safe enough to support smart contracts, with low fees, to host Cross advance contracts. After the EMIT Core Protocol network is launched, these will be migrated into it.

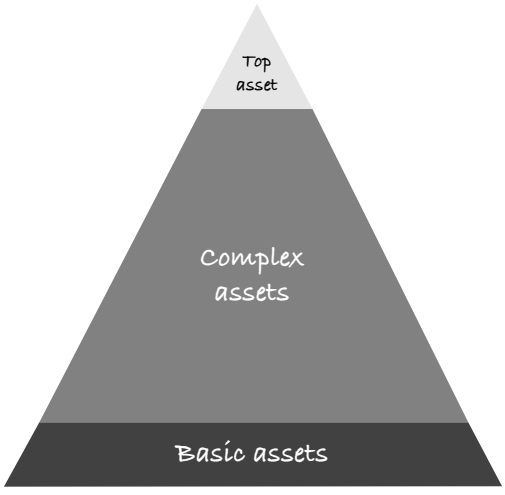
III. EPOCH Ecosystem Cluster

EPOCH is a decentralized ecosystem cluster, composed of components of various scales and dimensions. The initial ecosystem is provided by EMIT Foundation, which will also cultivate and popularize the community to initiate ecosystem projects. Regardless of the approach, the ecosystem will follow for a unified and open decentralized standard - the EPOCH Protocol. Driven by the protocol, these ecosystems will associate with each other with their own operations, constructing a more complex, larger, and higher-dimensional ecosystem to finally build a complex decentralized world economy.

3.1. Asset Hierarchy

In view of the complexity of the decentralized world economy, although we can't precisely predict its specific future situation, because its foundation and original motivation are clear, we can roughly speculate on its macrostructure.

Its economic base is not completely consistent with the real world. In the decentralized world, the evolved original economy is mainly driven by investment, arbitrage, and risk aversion. Therefore, future assets should be roughly divided into three tiers by liquidity: top, complex, and basic assets.



A. Top Assets

These assets form the upper structure of the decentralized world economy and provide the original force to drive its development. Their task is to allow

the natural behavioral motivation of speculation to flow into the decentralized world. They consist of assets with higher investment, collection and speculative value from the decentralized world. Their type, quantity and value are unstable, but they have the ability of value amplification corresponding to the lower assets, and can activate and leverage the circulation of the whole decentralized economy.

B. Basic Assets

As the foundation of decentralized world economy, basic assets constitute the footstone of decentralized world technology and economy. They open up channels to the real world, economically and physically guarantee security of the decentralized world, and ensure that it is not a castle in the air. Basic assets exist in the decentralized world in the form of interface and genesis assets (refer to "The EMIT Parchment Paper"), both existing historically and newly emerging in the future, and are infused with a large amount of early capital. These assets are equivalent to real-world natural resources such as land, oil, coal and metal mines. In the decentralized economy, their types and values fluctuate somewhat less than other assets.

C. Complex Assets

Complex Assets are the physical underpinnings of economic value in the decentralized world. This category has the largest volume, and triggers the most complex layer of user behavior. Based on the speculative nature of top assets, as well as the strong support of the basic assets, it evolves on the basis of complying with the EPOCH protocol, resulting in complex and numerous types of assets, as well as various related behaviors, which causes resource dependence and an economic closed loop in the decentralized world, thus forming a large decentralized world of GDP. This, in turn, supports the value of both the top assets and the underlying assets.

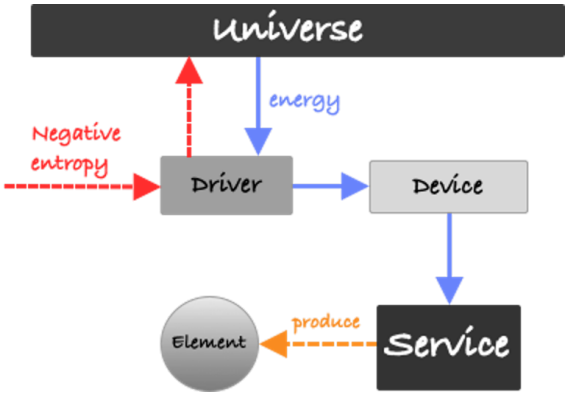
3.2 EPOCH Protocol

The EPOCH Protocol defines six basic interacting components, through which we can construct an ecosystem with an economic closed loop. The components then connect and interact with each other to form a larger and more complex ecosystem. The artifacts defined in EPOCH exist throughout the EMIT network, and some link to other decentralized systems through Cross.

The six basic building components of EPOCH are **Service**, **Driver**, **Negentropy**, **Energy**, **Device**, and **Element**. Both Device and Element exist as assets; the difference is that the Device is a NFT asset, while Element is a FT asset.

Let's take a look at how these six components complete the simplest economic process.

During operation of the EPOCH framework, consumption of resources in real-world computing Devices, or destruction of assets in the decentralized world, generates Negentropy, which is collected by the Driver objects and converted into Energy by a Service named Universe. After this, the Energy is fed into a Device, converted by the Device, and then becomes a Service. The Energy is then fed into a Device, transformed by the Device, and then becomes a Service, which can generate Elements or a Device according to the Energy input. The Elements can also be converted to other Elements, Devices, or Negentropy at other services.



A. Negentropy

The term Negentropy is a concept borrowed from thermodynamics, and is mainly used to represent and measure the flow of consensus-valued information from the real world to the decentralized world.

There are two main sources of Negentropy in the decentralized world: one is based on Proof Of Work (POW), and the other based on Proof of Asset Destruction (POAD). Of course, the properties of this Negentropy differ by the source, and they cannot be mixed.

For POW, the Negentropy (NE) magnitude is mainly calculated using the following formula:

$$NE_{pow}=\frac{2^{256}}{H} \quad (H \le 2^{256})$$

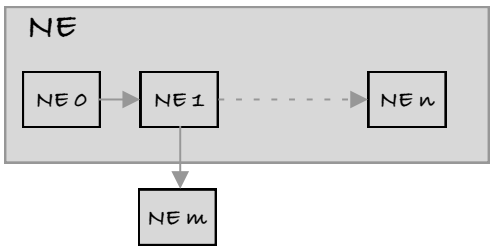
H represents the data obtained after hashing.

For the POAD proof, NE is equal to the number of tokens destroyed.

$$NE_{poad} = D_{token}$$

D represents the number of tokens destroyed.

The same type of Negentropy flow can be superimposed in a chain relationship:

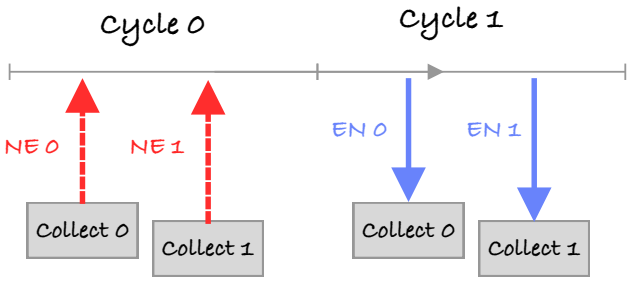


$$NE = NE_1 + NE_2 + ... + NE_n$$

The Driver is the only object used to collect Negentropy in the decentralized world. Only through the conversion by the Driver can the Negentropy be exchanged with the Service for Energy (EN). Negentropy represents value or wealth conversion, and cannot be traded.

B. Energy

Energy is a value stream uniformly allocated by the Services in the EPOCH framework, the "Universe" of services. It exists in allocation cycles, which are measured in fixed real-world time intervals. The amount of Energy produced by the Service is fixed in the same cycle. The Energy is then allocated according to the Negentropy added by each account during the cycle.



The relationship between Negentropy and Energy satisfies the following formula:

EN is the fixed Energy output during a Cycle, and NE is the sum of the

$$EN_{cycle,i} = EN_{cycle} \cdot \frac{NE_{cycle,i}}{NE_{cycle}}$$

Negentropy collected during the Cycle.

Properties of Energy

To describe the diversity of different energies, Energy has a value of length 32 bits as its property. After the Energy has been collected by the Driver, some properties associated with this Driver are added.

Energy Acquisition and Storage

Energy cannot be traded, and can only be exchanged by the Driver to the Service with Negentropy, and stored in that Driver.
Energy represents the metric data of the EPOCH operations. Its value and type represent the settlement result of the operation, and it cannot be traded.

C. Services

Only one instance of each Service exists globally; it has no owner and any other object can interact with it. This global object provides a public service for other objects. For example, the Universe object provides Negentropy exchange services for the Drive object, and the Chaos object acts as a receiver of Energy for the Device. The Chaos object, as a receiver of Energy, provides the Light element synthesis service to the Device object. Global

objects are the main providers of EPOCH ecosystem security, and can refuse to interact with illegal objects.

D. Device

The Device is the most complex building block inside the EPOCH protocol, and is responsible for asset diversity. Essentially, Devices are NFT assets that have an interface to interact with Energy, acting as a conduit for its transformation from the Driver to the final recipient Service. As NFT assets, Devices can change owners, and can even be combined and leased to provide diverse trading behavior.

Internal Structure of Devices

There are two standard subcomponents inside Devices: properties, and input/ output interfaces.

1. Properties

The properties of the Device are all data representing its current state, the most important being the following four properties.

- **Mass:** The highest Energy level that the Device can reach, which is 256 bits.
- **Energy Level:** The output performance of the Device, which is 256 bits.
- **Update Time:** The time when status change occurred.
- **Genes:** Data that cannot be changed after the Device is created, such as creation time, type, and coefficients of various curves.

2. Input and Output interfaces

The Energy leaves the Driver and enters the input, and then flows out to the Service connected to the output according to certain rules.

Energy Charging

When the Energy level of the Device is less than its mass, the Device can restore itself to maximum Energy through charging. The efficiency of charging is related to its current Energy level.

$$EN_{charge}=C_0 \cdot \left(\frac{Level+C_0}{C_0}\right)^{C_1}-C_0 \quad (1 < C_0 , 0 < C_1 <1)$$

The amount of Energy consumed to fully recharge from level 0 to Level

Production

When the Energy level of the Device matches its mass, the Device can start operations. Production also requires Energy consumption, and the production efficiency is related to current mass.

$$EN_{make}=M_0 \cdot \left(\frac{Quality+M_0}{M_0}\right)^{M_1}-M_0 \quad (1 < M_0 , 1 < M_1)$$

The amount of Energy required to raise the mass from 0 to Quality.

Response Output.

When the Device performs an output operation, its Energy level decreases in a certain ratio over time while moving the corresponding amount of Energy in the Driver to the output port.

$$Level_{oleft}=Level_0 \cdot O^t \quad (0 < O < 1)$$

The change in Energy level after a period of output.

Dissipation

When the Device does not have any Energy input, its Energy level also slowly decreases with time, i.e. dissipation. Dissipation is very similar to a drop in Energy level through output, except that no Energy is output whatsoever.

$$Level_{dleft}=Level_0 \cdot D^t \quad (0 < D < 1)$$

The change in Energy level after a period t.

When the Energy level of the Device drops to a certain level, the Device degrades and the quality becomes half of what it was before.

E. Driver

The Driver is tied to the user's account once it is created, and cannot be traded to anyone else. The Driver acts as the source of Energy output. Acter the account converts the Negentropy to Energy, it receives and stores the Energy generated by the Service, and then outputs it to the Device. Although the Driver cannot be traded, it can provide services to other accounts. It is also the proof of identity of the user account in the decentralized world.

The Driver has three properties: mass, amount of stored Energy, and Energy properties.

The Mass is influenced by the maximum total Energy output; the more output, the higher the mass.

$$\mathrm{E}N_{dout} = R_0 \cdot \left(\frac{Quality_{driver}+R_0}{R_0} \right)^{R_1} - R_0 \quad (1 < R_0 , 1 < R_1)$$

The Energy output required to reach a certain mass.

The mass affects the maximum Energy output per unit time.

$$\mathrm{E}N_{dleft}=\mathrm{E}N_{dori} \cdot Q^t \quad (0 < Q < 1)$$

The Energy left after a period of time if the Energy reserve is consistent with the mass.

The same Driver also has dissipative characteristics, which are also influenced by the mass.

$$D_{driver}=\frac{L}{L+Quality_{driver}} \quad (1 < L)$$

The dissipation factor is determined by the mass of the Driver.

$$EN_{dleft}=EN_{dori}\cdot P^{t}$$

The Energy remaining due to dissipation after a period with no output.

Similar to the dissipation of the Device when the Energy of the Driver drops to a certain percentage of its mass, a collapse occurs and the mass is halved.

F. Elements

In the EPOCH framework, all FT assets are Elements, and there is no difference between Elements of the same number and type – just as with normal tokens. Elements can be traded and partitioned very finely. At certain times, they can be converted to Negentropy or other elements by Services.

Elements have two attributes.

Category

256 bit data is used to distinguish different Element types, such as BTC Elements and light Elements, which are not interchangeable.

Decimal precision

An 8 bit value represents the number of the decimal digits when counting Elements.

Elements have two sources.

- The first is cryptographic tokens transferred from other chains via CROSS.
- The other is generated by Services in EPOCH, driven by Energy.

G. Some notes

As shown by the characteristics of the individual objects in the EPOCH protocol, EPOCH has the following characteristics.

1. A market to buy and sell Elements and Devices is possible
2. A labor market is possible
3. A market to rent Elements and Devices is possible
4. With the help of degradation or evolution of the Driver, credit-like relationships are possible
5. Users must make trade-offs between hiring, leasing, producing, and partnering to get the right value growth model for them
6. Users must allocate their generated Negentropy rationally and adjust it in time to maximize their benefit.
7. Devices are characterized by depreciation
8. Energy dissipates, producing a similar effect to inventory cost

9. High-quality equipment has minimal Energy waste, but also higher collection value
10. Due to Drive attribute classification and quality attributes, some participants will develop productivity advantages that form the basis for non-zero-sum market exchanges

3.3. Plan

At various times during EMIT's Prelude Period, the EMIT Foundation will release decentralized products related to EPOCH to guide and drive the development of the community and the establishment of an economic closed loop. Alternatively, it may issue initiatives for community-related activities.

IV. EMIT Core Protocol

The EMIT Core Protocol is a decentralized blockchain protocol, standards and rules for nodes in the EMIT Core decentralized network to maintain ledger data and interact with each other. These nodes operate independently and exchange information in a network environment, forming a scalable, efficient, and tamper-proof decentralized infrastructure.

The protocol is the cornerstone of the decentralized world of EMIT, and as such it requires the following characteristics in addition to the basic features of a public blockchain, and faces the following challenges.

- Consensus of all kinds without FT tokens, and natural cross-chain support. This will enable better use of existing blockchain assets, and will allow users to place their purpose on the ecosystem and the ecosystem assets on top of it, rather than on the assets defined by EMIT Core itself.
- Elastic scaling to support perpetual development of the entire EMIT. This requires scalability in terms of compute, storage, bandwidth, etc. to handle the future needs of growing business.
- Ease of complex DApp development. The future EPOCH Ecosystem may be highly complex. It is very difficult to design such complex DApps in the traditional application framework in the form of virtual machines, so EMIT Core needs to support and simplify their development.
- Supporting complex assets. In addition to FT assets, the EPOCH ecosystem will have a wide variety of productive activities involving NFT assets, and asset types will become diverse. That makes it important to natively support complex asset types in a unified form.
- User privacy protection. Privacy is everyone's right, and a symbol of people's independence and freedom, and EMIT Core needs to support privacy-enabled assets and transactions.

Cross-chain operations have been explored and studied in the previous chapters, and an implementation of EMIT cross-chaining has been proposed. The remaining items will be discussed next.

4.1. Summary of the Schemes

A. Flexible Expansion

TPS scalability has been a critical and difficult challenge since the advent of blockchain systems. With the development of blockchain, storage and bandwidth will also face scalability challenges. A variety of technical

solutions to the scalability problem have been proposed, challenging the scalability trilemma noted by Vitalik (the inevitable trade-off between decentralization, scalability, and security). These solutions fall into two main categories: layer 1 and layer 2 solutions. Layer 1 solutions focus on the blockchain system's own consensus mechanism, while layer 2 shifts transaction processing from a heavier on-chain system to a lightweight off-chain system. Our current focus is on layer 1 solutions.

Existing layer 1 solutions can be divided into two categories: slice and directed acyclic graph (DAG). They each have their own characteristics.

Slice

The main approach of the slice scheme is to allocate computing, storage, and bandwidth resources by splitting the single-blockchain structure into a multi-chain structure, with each chain processing different transactions in parallel. The split can be divided into two ways according to its granularity: grouping and lattice.

Grouping is implemented by dividing the accounts into groups, each corresponding to a set of servers and a chain of independent extensions, and maintaining a number of accounts. Transactions between sending accounts within a group is consistent with a separate blockchain, but transactions between sending accounts in different groups requires more work. NEAR and Etheruem 2.0 use this approach, and Polkadot and Cosmos are similar, except that they slice based on applications rather than accounts. The problem with the grouped approach is that it often requires a separate master chain to handle cross-slice transactions and maintain consistency across the slices. This independent master chain therefore becomes the bottleneck for cross-slice transactions.

The block lattice structure is the extreme form of slicing. Each account has a separate chain, and transfers between different accounts are cross-chain transfers. The approach simplifies the process by separating the sending and receiving into two transactions. Each account operates independently, thus providing very high throughput. This is the approach taken by Nano and Vite, but Nano uses an eventual consistency algorithm, does not support smart contracts and DApps, and is prone to forks and state rollback problems. And while Vite uses smart contracts to solve the rollback problem, just like the grouping solution, it also uses a master chain called the snapshot chain. Vite actually uses snapshot chaining for validation, a scheme that in a sense undermines the advantages of a sharded structure such as a block lattice; throughput ultimately depends on the performance of the snapshot chain.

Directed Acyclic Graph (DAG)

A DAG is essentially not a chain but a graph. In general, each block of a blockchain has only one parent reference, which results in multiple blocks forming a chain-like structure. The DAG, on the other hand, extends the number of references of the parent block so that it can refer to multiple parents. Because the block structure of a DAG is no longer a chain, blocks can be created and linked completely concurrently, enabling huge throughput. Spectre and IOTA both use this approach. DAG is essentially a more chaotic biased sequential structure, and the confirmation of blocks usually needs to be achieved by some statistical rules that can cause the structure to converge. Thus, although DAG has huge throughput, the confirmation is slow, and it is ultimately consistent and incompatible with smart contracts.

B. Complex DApps

There are two forms of current DApps, one in the form of smart contracts on the blockchain, and the other in the form of subchains.

Smart Contracts

Normally the blockchain consensus system has to be modified after completion by means of a hard fork – that is, updating all network nodes of the protocol, a complex process accompanied by certain risks. Smart contracts can enable users to customize the consensus protocol without a hard fork. Smart contracts date back to 1995 and were proposed by cryptographer Nick Szabo. This is a computer program that executes the terms of a contract. In the blockchain world, a smart contract typically represents a special account that can take incoming transactions as input and run a pre-programmed piece of code to perform operations on the account, which include modifications to account data and transfers of assets. A smart contract is a special type of DApp driven by a virtual machine running on a blockchain, which expresses certain meanings to the outside world by modifying the data associated with the account.

Ethereum introduced the concept of smart contracts to blockchain for the first time, an attempt to turn the blockchain system into a decentralized operating system. Various DApps and ecosystems continue to emerge on Ethereum; for example, the vast majority of DeFi applications are built on Ethereum, which has led to a boom in the decentralized trading ecosystem. Subsequently, many public chain systems started providing smart contract functions, and decentralized ecosystems has been developed on these chains, which include EOS, TRON, NEO, and SERO.

However, this form of DApp has some limitations. Since smart contracts rely on the state of the account to express meaning externally, each modification to the state requires consistency confirmation, and the methods and data structures used to implement the contracts have limitations that restrict the ways available to optimize system performance. It's also easy to create vulnerabilities when implementing complex applications using a more convoluted approach. Furthermore, because the virtual machines running smart contracts are part of the system consensus, the application is very cumbersome to maintain and upgrade. Finally, smart contracts usually require users to send transactions to drive them and do not run tasks spontaneously.

Subchains

With the development of cross-chain and flexible scaling technologies, solutions using sub-chains as DApps are starting to emerge. This approach implements DApps as independent public chains, and connects them to each other through a cross-chain mechanism, through which more complex and efficient application forms can be easily supported. Polkadot aggregates public chains with different consensus mechanisms to form a large ecosystem through Relay Chain, whose sub-chains are called parachains. Cosmos can link subchains using Tendermint's consensus through Cosmos Hub; those subchains are called Zones.

There are two problems with this approach to DApps. First, although these projects provide SDKs to ease the burden on developers, the development of a complete public chain is difficult, unlike development of smart contracts, which only requires attention on how to implement the task.

In addition, a master chain still communicates between different sub-chains, which limits the performance of DApps to a certain extent.

C. Privacy Protection

Identity, transactions and state are the main objects of privacy for decentralized systems. Although blockchain system accounts achieve a certain degree of identity concealment by means of pseudonyms and broadcasting, since all data in public chains are unconditionally open to the public, and most public chains are unable to hide transaction details, an attacker can always aggregate different addresses to the identity of a small number of real objects using statistics. In some sense, most public chains lack privacy protection, and such blockchains are not applicable for many scenarios. Since the privacy of a transaction, in a sense, represents privacy of identity and status, we classify transactions according to their means of privacy. Various blockchains already implement some degree of privacy protection, and these methods do so by obfuscating or hiding some or all of the information about the transaction. Next, four privacy algorithms are described, based on the comprehensiveness of the hidden information.

Obfuscate information on both sides of a transaction – Coin Shuffle

The way to shuffle coins is to disrupt the input and output of multiple transactions in a centralized or decentralized way, but without changing the final global state. Dash, for example, uses a decentralized CoinJoin approach to implement privacy protection features. Mimblewimble takes a similar approach to identity obfuscation. This scheme has some problems – for example, it relies too much on specialized nodes for coin shuffling, and these nodes know the transaction information of all parties, so there is a possibility of information leakage. There is a need to wait for other transactions, and the input and output after coin shuffling may be correlated.

Ring signatures – obfuscating initiator information

With ring signatures, the transaction initiator puts unrelated accounts into the same transaction as the transaction initiator. It is difficult for transaction validators to distinguish the real originator is from the multiple fake ones. A ring signature is an encryption algorithm. The CryptoNote protocol in Monero uses this approach to hide transaction initiator. This scheme is not resistant to dusting attacks, and if the other accounts involved in the ring signature are disclosed, the real originator will also be exposed.

Zero-knowledge proofs – hiding all information about the transaction

Zero-knowledge proofs enable a third party to verify that the information meets certain assumptions without disclosing the original information, and are implemented using algorithms such as homomorphic encryption and secure multi-party computation. Blockchain systems use zero-knowledge proofs to hide the details of a transaction, including the addresses of the sender and receiver, and the content of the transaction, without preventing the ledger generator from verifying the validity of the transaction. Currently, ZCash uses zk-SNARKs to completely hide transaction information, Monero checks the range of transaction input and output asset amounts via Bulletproofs, and SERO hides transaction amounts and addresses for FT and NFT assets via SuperZK, and uses smart contracts to hold and distribute encrypted assets. In a sense, zero-knowledge proofs are the most private of

the currently available and practical privacy schemes, enabling obfuscation of almost all information about the entire transaction and leading to privacy of the account assets themselves. However, their most important limitation is that their inefficiency. The more knowledge needs to be proved, the more computation is required.

4.2. Implementation Details

By studying and summarizing current blockchain protocols, through some assumptions and new decentralized algorithms, the EMIT project team finally proposed a decentralized public ledger protocol, EMIT Core, which can scale performance, throughput and storage capacity while ensuring security.

A. Network Environment Assumptions

1. The ratio of the number of correct nodes to the total number of nodes at any given time must be greater than S.
2. When a correct node receives a certain message, all correct nodes will receive this message within time T.

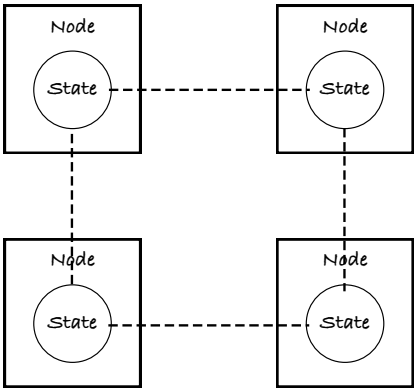
B. Block-lattice Ledger

EMIT Core uses a block-lattice ledger, which has the following features.

1. Each account is a separate blockchain.
2. Blocks of each chain can only be generated by the holder of the account private key.
3. A transfer between two accounts requires an outgoing transaction from the originating account and an incoming transaction from the receiving account.
4. Each account has a Merkle tree to record its current status.
5. When the outgoing transaction corresponding to an incoming transaction is recorded, the outgoing transaction is marked as settled, at which point the originating account can discard the settled blocks and save only the account status and the unsettled blocks.
6. In addition to transactions, the account provides key-value storage space.
7. Assets are described by (Field, ID, Count), Field represents the application serial number, ID represents the asset identification number, and Count represents its quantity. Such a generic asset description can represent both FT and NFT assets.

Unlike grouped slicing, this ledger generalizes behavior across chains. To achieve high throughput and low latency, the ledger decouples transactions and splits them into two parts, originating and receiving, created by separate accounts. Since the creation of different account blocks doesn't affect each other, this model allows for huge storage and throughput resilience after the introduction of the blocks' settlement status. Unlike Nano's account model, EMIT Core supports diverse assets and stores data in key-value form through the account's Merkle tree.

C. DApps

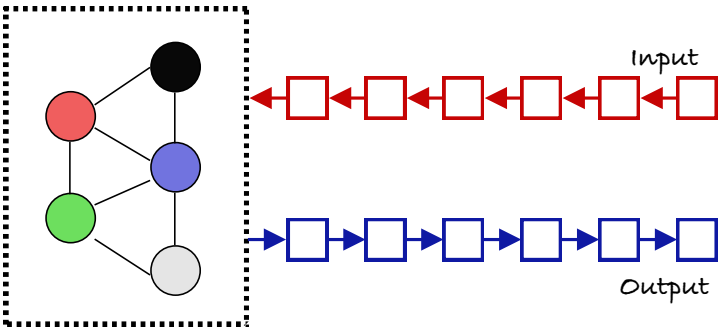


EMIT Core DApps contain the business requirements of executable smart contract applications, but are also defined more flexibly at a higher level. This definition makes it possible to implement complex DApps on EMIT Core in addition to Oracles, authority nodes, Cross, and other application methods. With EMIT Core's large throughput and low-latency system architecture, DApps approximate the experience of centralized applications.

Smart Contract Applications

The traditional definition of a smart contract application is based on a Turing-complete virtual machine. The final state of the storage is the ultimate meaning represented by the smart contract. This then means that a consistent state needs to be maintained within all nodes, and ultimately the state within each node's smart contract account needs to be consistent regardless of how the smart contract is manipulated. Comparison is often implemented using the root of the Merkle Tree of the state.

EMIT Core DApps



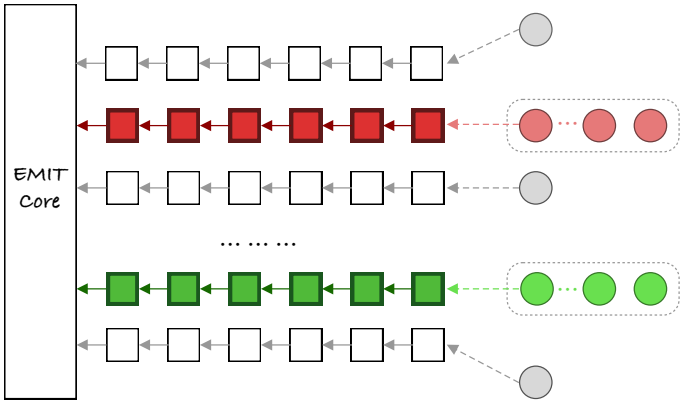
In EMIT Core, a DApp is defined as follows.

It consists of a set of P2P network nodes which can agree on a sequence of inputs and outputs as a whole. In extreme cases, the application can have only output or input sequences.

In EMIT Core's implementation, a DApp is a special account whose inputs and outputs are blocks on the same chain. Unlike normal accounts that require an account private key signature to get out of the block, the account signature of a DApp is an aggregated signature of a group of nodes elected by VRF. There are two roles for nodes of DApps, production and audit nodes, which are selected through VRF. Production nodes are responsible for producing blocks, and audit nodes verify and sign them.

Application Accounts

Like regular accounts, DApps have an account in the EMIT Core protocol that looks similar to a regular account in that it is a blockchain. Unlike normal accounts, which are managed by private key holders, however, application accounts are managed by a set of decentralized nodes that perform signature operations on blocks through a group of nodes filtered by VRF (similar to n-m multiple signatures).



There is no limit to the number of application accounts that can be registered in EMIT Core. There is also no limit to the number of nodes a single application account can manage. The behavior of the application refers to the EMIT Core definition of a DApp, and its input-output sequence blocks are managed by the EMIT Core protocol. Because of the huge throughput of the protocol, the interaction between applications and between applications and general accounts will become very efficient.

Application 0

Application 0 is a decentralized system consisting of decentralized nodes, which maintains the blockchain corresponding to its own account. Application 0's nodes initially consist of genesis nodes, and other nodes may join according to predefined rules. The node corresponding to Application 0 actually represents the consensus of the EMIT Core protocol; its main role is to asynchronously moderate the EMIT Core network. When other accounts need to create new applications, they apply through Application 0.

Joining and Exiting by Application Accounts

Decentralized systems, including Application 0, have nodes that require conditional joining and exiting, with the specific joining conditions determined by consensus of the application's current validation node. When a candidate node initiates an application to join an application node, the decentralized verification nodes of that application agree by consensus, and the candidate node can become one of the verification nodes to participate in the consensus of the application. In the case of unanimous consensus of the verifying nodes, the verifying nodes can then exit the application account.

D. Confirmation Mechanism

Ordinary accounts and application accounts form the account structure of EMIT Core. Each account in this structure is independent, so it's difficult for the commonly used blockchain protocols to attacks if the private key holder of an individual account, or the management node of an application account, commits malicious behavior. Therefore, EMIT Core uses two steps to ensure that double-spending attacks by the administrators of both accounts cannot succeed in the case of high throughput and low latency: random checks of the validity propagation network, and Application 0's application confirmation mechanism. In the case of high throughput and low latency, double spending attacks by the administrators of both accounts cannot succeed.

Validity Propagation Network

EMIT Core's Application 0 node, while responsible for the management of its own accounts, constitutes an efficient P2P propagation network with two responsibilities: broadcasting messages to the whole network to the extent possible, and verifying the validity of each transaction. The nodes use DHT and Gossip protocols to form a P2P network, ensuring that each node is connected to at least D or more neighboring nodes; the larger D is, the faster the broadcast speed will be. Messages are then broadcast between each node using the Gossip protocol, ensuring that any valid message is quickly broadcast to the entire network. When a node receives a transaction, it first confirms the validity of the transaction, such as whether the incoming transaction has a corresponding outgoing transaction, the signature of the transaction is correct, and the account balance is sufficient. Effective networks ensure that the messages being disseminated are effective.

Random verification

Random verification means that the confirmation verification requester randomly selects some nodes among the global nodes and gets the account information on these nodes. If these account states are consistent, then the state can be considered the correct state of the current account. Security can be improved by increasing the number of checks or the number of nodes per check.

Random checks are required in three situations.

- The node handles ledger forks. Since only the owner of an account can change its ledger, usually the ledger will not be forked. But in the event of an error in the transaction sending process, or if a malicious account manager launches an attack on the network, the ledgers of their own accounts will diverge. At this point, the node uses a random checking mechanism to determine the correct branch.
- Client-side verification of outgoing transactions. EMIT Core's transactions consist of incoming and outgoing transactions. The receiver of the transaction can check the sender's account through random verification to conclude whether the outgoing transaction is confirmed or not.
- Ledger consolidation: EMIT Core nodes batch consolidate ledgers of a certain height based on changes on the global Merkle Tree. A random check is initiated when a branch of a Merkle Tree does not change for a period of time. If the results are consistent, the account is consolidated. If there are consolidated heights prior to the arrival of the transaction, they are discarded directly.

Random verification is a final consistency check that allows node conflicts to be handled and client confirmations verified separately and independently. Once consistency is observed in a random check on some occasion, then this state is irrevocable.

Application 0's Application Confirmation Mechanism

In some extreme cases, such as a bifurcation where each of the two branches occupies half of the nodes and cannot converge within a short period, the random confirmation mechanism is unable to obtain a conclusion. In this case, the client can apply to Application 0 for arbitration and reimbursement of a certain amount of money. At this point, Application 0 outputs an arbitrated block and broadcasts it to the whole network. All nodes go through this arbitrated block to select the correct branch for that account. Since this process is asynchronous, it will only impact this exceptional account.

4.3. Security Analysis

As long as the number of correct nodes in EMIT Core is more than 50%, then the EMIT Core protocol is secure and convergent. Next, we analyze some common attacks in the blockchain space.

A. Double-Spending Attacks

A double-spending attack is when the attacker first initiates a trade for A. After A confirms the success of the trade, the attacker retracts this trade and uses the same funds again through a certain means. In POW or POS blockchain projects, this is often achieved by a long-range attack, in which the attacker first packages a transaction on the normal chain A, while secretly generating a longer chain B with stronger arithmetic or shares. Chain B does not reflect the transaction yet. After the transaction is confirmed, chain B is broadcast to the public. Since both POW and POS are longest chain consensus, chain A is replaced by chain B, at which point the transactions that A had previously confirmed are no longer valid. EMIT Core is not the longest chain consensus, and this consistency no longer changes when A observes a transaction being confirmed with a random verification. It is overwhelmingly likely that the other node observations are consistent with A.

B. Invalid-Block Attacks

An invalid-block attack is when an attacker takes control of a slice A during a cross-slice transaction by some means, and establishes an invalid transaction on A that points to slice B, such as transferring a large amount of money. Since the verifier in slice B does not have the verification information for slice A, slice B cannot verify the validity of this transaction on slice A across the slice. All nodes in EMIT Core have unsettled data for all accounts, so it is very easy to verify the existence of rollover transactions. Since EMIT Core's slicing is based on individual accounts and uses final consistency, it does not affect the throughput and latency profile of the entire network.

C. Sybil Attacks

In a Sybil attack, an attacker generates a large number of authenticated nodes with different identities at low cost in order to affect the security of the decentralized system. In EMIT Core, normal nodes need to follow certain rules to become verified, including requirements to pledge a certain amount of digital assets, have a high level of drive in EPOCH itself, prove that they already have the correct ledger data, and pass a vote by 2/3 or more verified nodes. Therefore, it is difficult for an attacker to forge a large number of identities at low cost.

D. Other Attacks

Since the EMIT Core protocol is not the longest chain consensus, it also does not use voting for block validation. Therefore, long-term attacks and nothing-at-stake attacks, which are often encountered in POS and POW, do not exist in the EMIT Core protocol.

4.4. Conclusion

EMIT Core has sufficient security and availability. The randomness of a random verification is ensured by the true random number provided by the verifier itself, and the security level can be determined by the number of random verifications. Cases where the verification mechanism fails to converge can be solved by using Application 0's Application Confirmation acknowledgement mechanism. Since Application 0 chains rarely need to coordinate (except for random verifications that fail to converge), all other mechanisms are independent and do not interfere with each other, so EMIT Core can achieve huge throughput and resilience.

V. Economic Model

EMIT Core is designed as a protocol with no native FT assets. All assets on it are transferred from other blockchain systems or are generated on the EMIT ecosystem. The application 0 node profits from the entire EMIT ecosystem and economy, acting as one of the links in EPOCH negentropic transformation, and it deducting a portion of the Energy from the transformation. Of course, this Energy cannot be traded directly to others, but can only be used as gas to participate in various EPOCH ecosystems. When the GDP Market is released, the verification nodes, as the cornerstone of the EMIT Core protocol, will draw commissions from the marketplace.

The economic model and the technical details of EMIT will be elaborated upon in a further series of documents.