World Yuan

A Currency-Free Electronic Exchange System

Hawk Shea

Changsha,Hunan,China.

Corresponding author(s). E-mail(s): hawkshea@worldyuan.org ;

Abstract

This paper introduces a revolutionary electronic exchange system that enables secure peer-to-peer transactions over network without relying on currencies or centralized authorities. Based on the mathematical principles of commodity exchange, the proposed system establishes a labor-proof mechanism, replacing public ledgers with a peer-to-peer consensus protocol. This innovative approach ensures decentralization and anonymity while addressing the performance and storage limitations of blockchain technology.

For the first time, the study defines a standard for measuring value and demonstrates how "World Yuan" functions as a substitute for traditional currency by fulfilling the roles of a measure of value, a medium of exchange, and a store of value. The system not only overcomes the common problems of cryptocurrencies such as high volatility and the dependence on fiat money, offering a scalable and efficient alternative to the blockchain, but also shakes the status of currency as the core of exchange.

By elucidating the relationships among labor, value and price and exploring novel mechanisms for social exchange and consensus, this work lays the technical groundwork for a future global financial system free from monetary constraints, offering a fresh paradigm for economic and technological innovation.

Keywords: Currency-free economy,Social exchange,Peer-to-peer consensus,Labor proof,Calculation proof,Fair overflow

1 Introduction

Currency[1],as a medium of exchange, has inherent limitations. Due to its physical properties, the division, storage, transportation, and measurement of currency all increase transaction costs. Additionally, the growth rate of commodity production often surpasses that of currency supply, so the development of commodity economy has always been hampered by a limited money supply. Furthermore, in the process of commodity-currency-commodity exchange, the quantity of currency is not only determined by the value of goods but also fluctuates with its own production efficiency. In particular, the inherent inflationary nature of the currency means that the longer it is held, the fewer goods it can exchange for, violating the fundamental principle of equivalent exchange.

Cryptocurrencies have not fundamentally resolved these issues. Their prices are highly volatile and heavily reliant on fiat money valuation, rendering them unreliable as a medium of exchange. Blockchain systems face inherent constraints under the CAP theorem: as of August 2024, the Bitcoin blockchain size has grown to nearly 600GB[2] , while its transaction speed is limited to 7 TPS. The "blockchain trilemma"[3] further emphasizes that blockchain technology is suboptimal for global financial systems.

However, the emergence of cryptocurrencies provides valuable insights, showing that computational activities can generate value. Despite this, current designs remain bound to traditional currency frameworks, hindered by limited supply and ambiguous valuation, which contradict fundamental economic principles. Furthermore, blockchain synchronization methods poorly reflect human behavior patterns. Advances in computer networking technology provide an avenue to surpass these limitations, paving the way for currency-free exchange mechanisms. This calls for a fundamental re-evaluation of exchange principles and mechanisms. World Yuan is built upon a foundation of rigorous economic analysis.

[1]The term "currency" here refers only to commodity money[1, 29.1.2] with an intrinsic value, ignoring fiat money.
[2]The data source:https://ycharts.com/indicators/bitcoin_blockchain_size
[3]https://academy.binance.com/en/articles/what-is-the-blockchain-trilemma

2 Principle of exchange

Exchange is a point-to-point distribution without the participation of central institutions. Both parties occupy each other's products and occupy each other's value. Therefore, we have to start our research from the value.

2.1 Value and price

Labor is the expenditure of physical and mental effort by human beings. The amount of labor produced is called *labor volume*. Value measures the cost of labor invested in achieving a specific labor volume, including both the physical and psychological aspects, as well as the transferred portion of labor that has already been materialized in the means of production, and the condensation of live labor invested in the production process[2].

Value is determined not by individual subjective feelings, but by the volume of labor and the current level of productivity. The physical properties of the outcome determine the labor volume, and the moment at which labor begins is linked to social productivity, so that value is also determined.

The *commodity value* is created during production and originates from its utility[4] . Unless otherwise stated, value refers to commodity value. Since value is embodied in a commodity, it is naturally possessed by the producer and transferred along with the commodity.

Since people are both producer and consumer, exchange connects production and consumption, allowing the value of the producer to make up for the value of the consumer, making reproduction possible.

If people are born equal, then no one has the right to possess more value. Exchanging utilities based on value is the embodiment of political equality. Although value objectively exists,it cannot be measured precisely: people cannot know the situation

[4]The property of a commodity to satisfy a specific human need,what Marx calls "use value", used here to avoid ambiguity

of all producers, nor can they accurately convert various factors of production. They can only make a vague judgment on commodity value. *Price* is a subjective category that reflects people's judgment on the value of labor outcome. Price always fluctuates around the value of the commodities to be exchanged.After a large number of exchanges, the total price approaches the total value, so the exchange is generally fair.

When quantitatively describing the value of a specific thing, value and price can be used interchangeably, as this is necessarily a subjective judgement. However, it should always be remembered that value is an objective existence determined by labor, and whether or not a product is exchanged does not affect its value.It makes no sense to talk about value apart from the moment of production, as discussed in more detail in Section 7.

2.2 Input and output

Input and output are ways of transferring value. Commodity value is outputted when the commodity is given, and value is inputted when the commodity is consumed. Production concentrates labor on products, transforms products into commodities through exchange to achieve value transfer, and consumers obtain utility after consuming the commodity, thereby promoting the development of labor force, as shown in Figure 1.

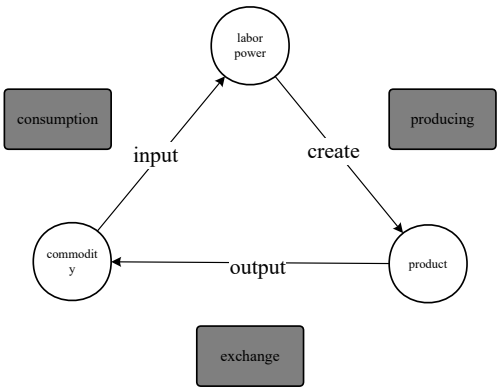

Fig. 1: Value cycle is an advanced form of material cycle in human society

Exchange is the process of outputting value to each other, and the value is equal. In the early times, people bartered and the input of value occurred simultaneously with the transfer (output) of the commodity. However, after the emergence of ownership, exchange can be conceptual, and consumption may not occur for a long time. If the two are synchronized, it is called an "instant output"; otherwise, it is a "promised output".

Promised output complicates the exchange. In order to ensure that the outputted value can be inputted, excessive unproductive labor must be performed before outputting: Suppose that after combining various factors of production, the average value to run 400 meters on a standard track is 1,and is 5 to produce 1 kilogram of flour. If Alice claims to have 1 kilogram of flour, she can be asked to run 6 laps of the standard track before the commodity is given to her. If Alice reneges on the agreement and does not give the flour, the value of the exchange does not cover the running costs, otherwise the running costs can be used for the next exchange. Thus, the incentive to cheat is eliminated.

Such necessary excess labor is called *guarantee*, the outcome is *labor proof*, and *guaranteed value* is created. Both the commodity value and the guaranteed value are meaningful values.

Gold is the labor proof of mining work of people. If gold is exchanged for grain, it becomes the proof of the farmer's cultivation. Possessing a labor proof gives the right to obtain the commodity. Thus, guarantee, like production, becomes a means of creating wealth.

2.3 Reliable value

It is not difficult to conclude from the above analysis that during an exchange, the outputted value by either side cannot exceed the guaranteed value before the input is

made. That is:

$$Outputted\ value \leq \frac{Guaranteed\ value}{\alpha} + Inputted\ value \tag{1}$$

α is called the *guarantee ratio* ($\alpha > 1$), which represents the guaranteed value required for outputting one unit of commodity value.

Value of instant output are inputted at the same time, while value of promised output will be inputted after consumption. A barter exchange always satisfies the above requirements because all outputs are instant outputs. Obviously, the guaranteed value required in circulation is directly proportional to the size of the commodity and inversely proportional to the speed of consumption.

The right-hand side of the inequality is called the reliable value:

$$Reliable\ value = \frac{Guaranteed\ value}{\alpha} + Inputted\ value \tag{2}$$

2.4 Consumption

Consumption is the process by which the utility of a commodity is realized and value is transferred from the producer's loss to the consumer's gain. The producer transfers ownership of the commodity to the consumer, and the consumer *executes* the commodity to obtain utility. *Quantity available* is the quantity of certain commodity that the consumer can execute, and is credited to the output side. After exchange or failed execution, quantity available is increased. Executing the commodity decreases quantity available, and if it is consumed successfully, it is converted into the output side's *quantity inputted*,as shown in Figure 2.

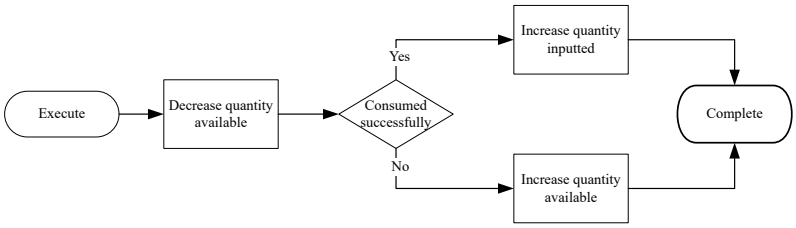

Fig. 2: Input value asynchronously

2.5 Social exchange

Up until this point, the concept of exchange has been limited to a point-to-point basis, which presupposes the existence of excessive labor on both sides. On this basis, establishing an indirect link to the target commodity and achieving an end-to-end exchange is referred to as social exchange.

2.5.1 Purchasing power

Social exchange is mediated by purchasing power, which represents any commodity of a certain value and can be considered the value form of commodity. In order to reduce complexity, offline transactions convert commodities into a certain quantity of purchasing power according to their value, and unify the commodity exchange as purchasing power exchange. When exchanging commodities of unequal value, the purchasing power can be used as change to solve the value matching problem.

Like ordinary commodities, purchasing power can be exchanged and executed, but it is only when it is consumed that it is transformed from its value form to its natural form and utilized. Consequently, the output of purchasing power is generally a promised output.

Although both purchasing power and currency are mediums of exchange, the quantity of purchasing power does not correspond to the quantity of any object[5] . Rather,

[5]In the case of metallic currencies,quantity of metallic currencies is the mass of metal.

it corresponds to the amount of value behind the commodity. The execution of purchasing power is also referred to as "redeeming", and the available purchasing power outputted from A to B is defined as B's "*Equity*" at A.

2.5.2 Exchange chain

Suppose Alice, the owner of a ranch, needs apples, and Carol, an apple farmer, grows apples. The two can be connected through an intermediary, Bob. The prerequisite is that the reliable values between Alice and Bob, and between Bob and Carol, are sufficient to output the value of the apples. Here's how it's done:

1. Alice increases her equity at Bob, for example, by using the milk produced on her ranch to exchange for purchasing power from Bob.
2. Similarly, Bob increases his equity at Carol. Note that at this time, the reliable value between Alice and Carol is 0.
3. Alice asks Bob to redeem the apples produced by Carol, and Bob agrees.
4. Bob asks Carol to redeem the apples for Alice, and Carol agrees.
5. Alice receives the apples from Carol and acknowledges that she has successfully redeemed them with Bob.
6. Bob acknowledges that he has successfully redeemed the apples with Carol.

The above process can expressed as Figure 3(Assuming that everyone has sufficient equity).

In practice, an order can be used instead of apples, and the two adjacent sides can even carry out a pure exchange of purchasing power. Each person both outputs purchasing power to the previous person and executes purchasing power from the next person. The starting point is the consumer, the end point is the producer, and the second hop (Bob) is the *access point*. There may be more intermediaries actually involved, forming an exchange chain. The key is to find all the intermediaries from the consumer

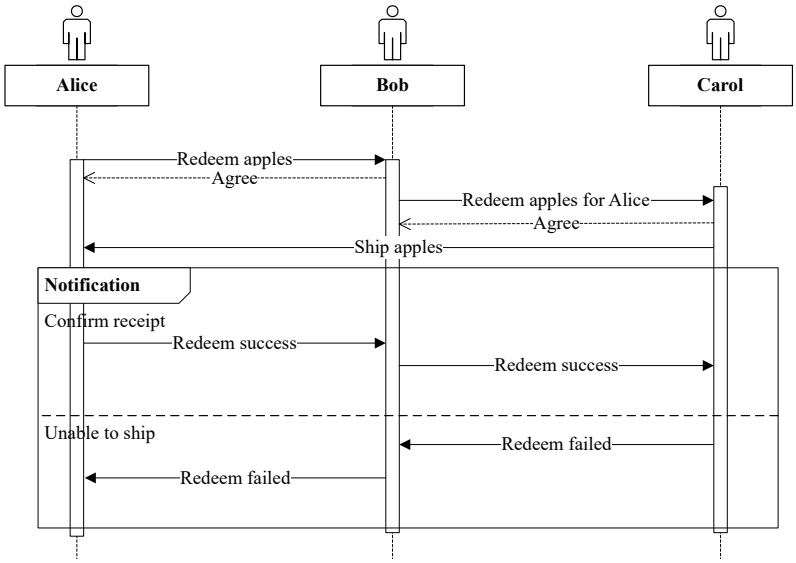

Fig. 3: Exchange milk for apples

to the producer, which is called *consumption routing*, and the path formed by the intermediaries is called *consumption path*.

Offline traders can join the system through instant outputting. Before the first exchange, they may need to wait for the access point to guarantee and increase the reliable value. Once the guarantee is complete, subsequent exchanges can continue until the value to be exchanged exceeds the reliable value of the other side.

2.5.3 Profit

Since value is determined at the time of production, it is important to explain how profit is generated.Profit is part of value, and it has three sources.

Firstly, the exchange itself has a cost, which is ultimately included in the value of the exchange. These costs are independent of the production of the commodity and include the space, rent, labor, and physical or mental toll of waiting for the exchange to take place. For example, milk from a farm has additional value when it is transported over long distances and then put on the shelves of a city.

The second source of profit is the contradiction between individual productivity and social productivity. Value is the result of the compromise between individual and collective labor. To illustrate, consider a society comprising two fishermen. One can catch a fish in 30 minutes, while the other requires an hour. The average labor time for fishing is thus 0.75 hours. Upon exchanging their respective catches, one gains 0.25 hours while the other loses 0.25 hours. It is important to note that in reality,exchange of commodities with the same utility will never occur. However, when both fishermen exchange with an intermediate commodity, such as currency, profit is generated through price transmission. This results in an inefficient producer losing while an efficient producer gains. The gap between rich and poor may widen under the Matthew effect.

Finally, it is important to note that exchange as a form of distribution does not necessarily result in equal value being outputted by the two sides involved. Depending on the dynamics of supply and demand, or if one side has a say, the commodity values of the exchange may differ greatly.Profits from unfair transactions and exchange costs are collectively referred to as "explicit profits". What is usually referred to as profit is explicit profit.

2.6 Store of value

Exchange faces the contradiction between production and demand, present and the future. So producers have a natural need to store value: they convert commodities into purchasing power and then exchange them for commodities when need. Besides, the value of commodities is determined by the era of production, but exchange price refers to the current level of productivity. In order to solve this contradiction, it is necessary to change the value from the changeable commodity form to the unchangeable fixed form.

In the past, people used to exchange commodities for universal equivalents and deposit them in the bank. Equivalents are also commodities whose prices are constantly changing and do not in themselves have the effect of "fixing" value. Moreover, the system of transfer settlements made equivalents symbolic from the moment they were deposited in the bank, replacing them with symbols both for withdrawals and for loans.

Transfer settlements were unfair. Commodities and debts were forced to be tied to symbols of value, whether or not they were backed by real currency. The fractional reserve system allowed a single sum of money to appear in two places, and one person's deposits could be used to make loans to others at the same time, so that the nominal currency supply expanded several times compared to the original deposits [6] . Lenders buy commodities and services from the society before inflation, and then pay back the same amount of value symbols when prices rise, but the interest compensated to depositors is far less than the loss from inflation. In essence, the debit exploits the depositor, and the purpose of value storage cannot be achieved.

We can define storage of value as exchanging commodities for someone's purchasing power which can be consumed in future. The commodity value corresponding to the purchasing power is fixed, both now and in the future, and the limitation of output prevents the creation of value symbols.

Storage of value can be achieved by exchanging purchasing power directly in kind, or, if necessary, by converting purchasing power at the access point to purchasing power at the target through social exchange. Social exchange of purchasing power does not differ from exchange of commodities in general, except its execution. The producer does not transfer commodities to the consumer but outputs purchasing power, which, unlike exchange, is unidirectional. Social exchange of purchasing power, which must be delivered through output, does not allow for bookkeeping.

In a currency-free economy, there are no restrictions on the supply of purchasing power, except for reliable value.The fall in the price level is a natural consequence of

[6]This is called the money multiplier[1, 29.3.3], and it is the reciprocalof the reserve ratio

the increase of social productivity. The same quantity of purchasing power can be used to buy more commodities if demand remains unchanged. Although the purchasing power gained from sales is less, the purchasing power used for labor and raw materials is also less, so that the effort and the reward are still in balance.Producers who are the first to reduce prices will have an advantage in the competitive marketplace, and once society as a whole accepts this concept, everyone will enjoy the benefits of advances in productivity.

3 Anonymity

In communication, if an unrelated side cannot identify, trace or block a specific host[7] ,the communication is considered anonymous. Capital freedom requires anonymous exchange.

Communication over insecure channel can be simplified to the model shown in Figure 4:

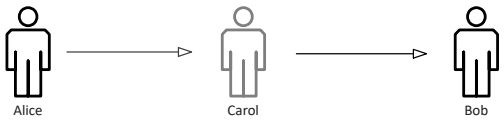

Fig. 4: Unsecure communication via middleman

Alice and Bob have public keys K_a and K_b,private keys K_A and K_B, and Carol has K_a and K_b and is eavesdropping the communication.

If the communication is anonymous, Carol cannot analyze whether it is related to Alice or Bob. If the information transmitted on the network meets this requirement, it is called *traceless information*. If all transmissions are traceless information, it has the highest anonymity.

[7] To avoid confusion with the "node" in data structures, use "host" to refer to the node concept in network communication.

We will now introduce the key concepts introduced by World Yuan to achieve anonymity.

3.1 HGL

HGL(Host Global Locator) is used to uniquely identify a host across the entire network. It has minimal anonymity and is generated by double hashing the public key:

$$\begin{aligned} HGL_a &= \text{Hash}(\text{Hash}(K_a)) \\ HGL_b &= \text{Hash}(\text{Hash}(K_b)) \end{aligned} \tag{3}$$

Hash is a hash function.

3.2 Pair entropy

The host needs to manage local private keys and external public keys, and only exchanges public keys with the outside world once. *Pair entropy* is a kind of traceless information: *Which private key is used to communicate with which peer who holds which public key.* The pair entropy of the same pair of hosts remains the same and is meaningful only to each other. The system formed by these two hosts is called a host pair. The hosts in the host pair are the positive pole and the negative pole, represented by 1 and 0 which is called the "polarity". The positive pole is the side with larger public key.

To generate pair entropy, the public key, the other side's HGL, and the signature of the HGL are sent to the remote host. A secret value is calculated using the Diffie-Hellman key exchange protocol as the pair entropy(represented by Pe). This process is called registration,as shown in Figure 5.The key pair is also used for digital signature[8]. After registration, each host adds a new registration record to the database.For example, contents in Table 1 will be added to Alice's database.

[8]Diffie-Hellman key exchange can be made over Edwards25519, the same group as the one used for signatures.

Table 1: Registration record of Alice with Bob

Own Polarity	Pair Entropy	Own Private Key	Opponent Public Key	Registration Time
0	P_e	K_A	K_b	0 for incomplete registration

Pair entropy associates local and remote private keys. The pair entropy can be used to derive shared information required for the host pair. The constituent elements of information are transmitted over the network, but only the two ends can calculate a consistent result.

3.3 HPI

Using pair entropy and own polarity, you can derive *HPI (Host Pair Identifier)*. HPI is used to retrieve registration records quickly:

$$HPI = \text{Hmac}(Polarity, Pe) \tag{4}$$

Hmac is a key-related hash message authentication code function. HPI actually corresponds to a directed edge formed by two points in the network topology. A third side cannot track the endpoints of the edge simply by HPI.

3.4 HHL

HHL (Host Hidden Locator) adds randomness to HPI to achieve complete anonymity, so that it is impossible to determine whether two messages point to the same object.

Assuming Nonce is a random number of no less than 32 bits, and the operator $||$ combines two bit sequences, then HHL is a function of HPI:

$$HHL(HPI, Nonce) = \text{Hash}(HPI||Nonce) \tag{5}$$

In a host pair, someone's self is the local host, and the other is the remote host. Each message contains the HHL of the recipient and a related random number. After receiving the message, the random number is calculated with each HPI in the database, and if the HHL is in the database, the message is received, otherwise the message is discarded.

Returning the other side's HHL is used as a sign of successful registration.Figure 5 shows the entire registration process.

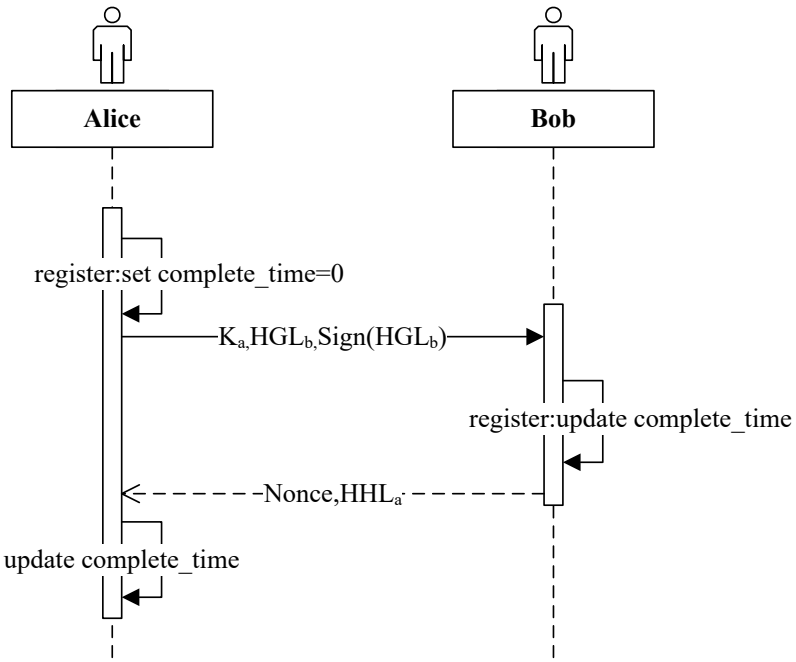

Fig. 5: Full registration sequence chart

After registration, the mapping from HGL to HPI should be saved. One HGL may be included in multiple host pairs, and one physical host may host multiple logical hosts. For Alice, at least the following information should be saved after registering

with Bob:

$$\text{registration information} \begin{cases} HPI_a & \rightarrow \text{Registration record} \\ HGL_a & \rightarrow [HPI_a, \dots] \\ HGL_b & \rightarrow [HPI_a, \dots] \end{cases} \tag{6}$$

Only HHL and HGL are transmitted over the network, but the latter is not traceless information and should be used as little as possible.

3.5 Authentication

Suppose Alice sends a message M to Bob and the signature is I.We hope that:

1. Alice and Bob can prove that the message was generated by Alice and sent to Bob.
2. Given K_a, K_b, M and I,Carol cannot tell whether the message is related to K_a or K_b. In other words, $\{K_a, K_b, M, I\}$ is traceless information.

A signature that meets the above requirements is known as a *traceless signature*.Suppose that syntax for performing a digital signature is "Sign($message, privatekey$)".The valid signature is:

$$I(M, K_A) = \text{Sign}(\text{Hmac}(M, Pe), K_A) \tag{7}$$

Traceless signature is often sent with HHL of remote host. It should be noted that anonymity is not confidentiality, which should be implemented with the help of underlying protocols such as TLS or DTLS.

4 Peer-to-peer consensus

In traditional banking system, the records of assets are centrally stored in the bank's system, which leads to an issue of power imbalance between the bank and the depositors. World Yuan proposes a new method to enable two distrusting parties to reach a consensus on current system state through a proposal and signature mechanism.

A complex system can be viewed as an interconnected network of countless host pairs. Instead of relying on a global ledger, World Yuan's consensus mechanism is based on a peer-to-peer design, where each host pair only needs to store data related to each other. This approach avoids the brain-fracture problem in distributed system and the centralization of computing power in blockchain. This design can still maximize utility by reasonably dispersing risks in the event of partial loss of information.

4.1 Overview

A system's initial state is defined by a hard code and is the earliest consensus. Later, system state updates are achieved through mutual agreement on proposals. A proposal is the smallest unit of business operation and defines a specific state change. When both sides agree on one or more proposals, the system state is updated and the new state becomes the consensus.

The order of the proposals is crucial for their validity. A different ordering of the proposals would lead the system along a different evolutionary path. The sequence of proposals that have been agreed upon is called the "world line", while the sequence of proposals awaiting confirmation forms the "forecast line".As shown in Figure 6, the consensus points to the end of the world line.

The host derives future states based on the forecast line, where each node on the forecast line corresponds to a future state, which is called a "preview". There may be more than one forecast line, and the core task of consensus is to select a path from the many forecast lines to expand the world line.

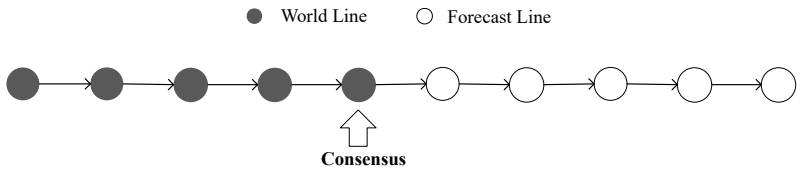

Fig. 6: World line and forecast line

4.2 Forecast line

The forecast line starts at the next position of the consensus. A node on the forecast line consists of the proposal and other information. The height of a node is the distance from the node to the consensus. Table 2 lists common fields included in all proposals.

Table 2: Common fields of proposal

Field	Description
Author	Author's polarity
Creation sequence	Used to count local proposals by author
Signature	Author's signature.See below

The *creation sequence* is the number assigned by the author to his or her proposal for the current consensus, starting from 1. For the same author, the height of the proposal for an earlier creation is also smaller.The author can compute a hash value for the proposal as an index, which is calculated from the content and the consensus progress(see Section 4.3).

Overlaying an older state with a proposal results in a new state, which is called the proposal's *addition*. Addition is defined for every proposal, and the preview is derived by adding the proposals.

When sorting, local proposals are linked directly to the end of the forecast line and sent to the remote host, while remote proposals need to be stored in the queue for later processing.See Figure 7 for the specific process.

When processing a proposal, the following checks must be completed:

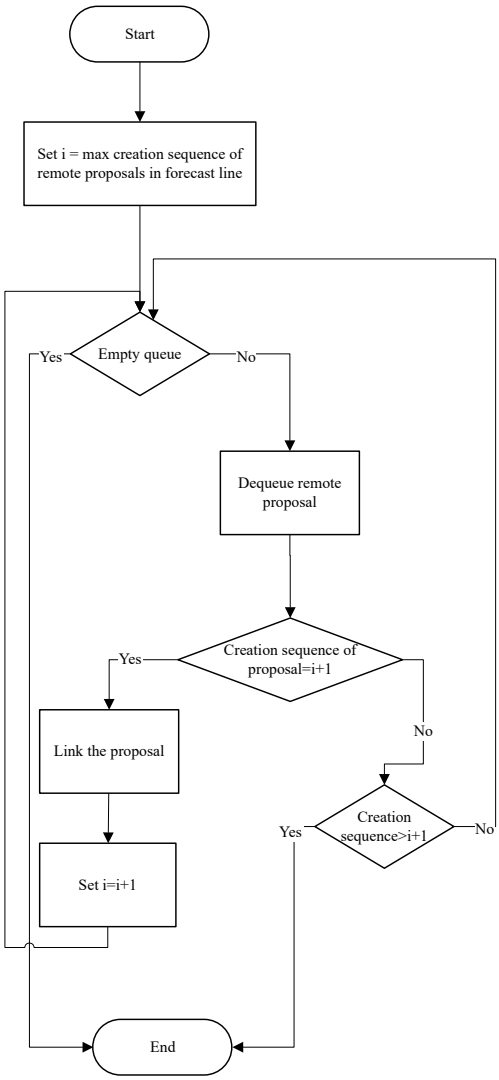

Fig. 7: Remote proposal processing

1. *Formal Check*:The structure and value of the proposal must be correct,otherwise they will be discarded before being linked or added to the queue.
2. *Integrity Check* :For specific system state, there are always some constraints that must be met, which are called *state constraint*. The *precondition* of the proposal must be met in the previous state, and the new state cannot violate the state constraint. Proposals that fail the integrity check will be *rejected*.

Rejected proposals will still be retained on the forecast line for reference. In addition to the integrity check, the system can additionally define a review procedure to determine whether to accept the proposal,which is implemented after integrity check. After consensus is reached, all accepted proposals will be added together to derive a new system state.

4.3 Consensus

Assuming that the positive pole creates four proposals $A_1 - A_4$ and the negative pole creates three proposals $B_1 - B_3$, the following three possible evolutionary paths of the system are listed:

$$A_1 \to B_1 \to A_2 \to B_2 \to A_3 \to A_4 \to B_3$$

$$A_1 \to A_2 \to B_1 \to B_2 \to B_3 \to A_3 \to A_4$$

$$B_1 \to A_1 \to A_2 \to A_3 \to B_2 \to B_3 \to A_4$$

The goal of consensus is to select the unique path from all the eligible paths. A path can be uniquely determined by its hash value: each node has a hash value, which is calculated from the hash value of the previous node, the proposal, and the acceptance flag (whether to accept or not).So the nodes are connected as in Figure 8. The hash value of the path is the hash value of the end point.

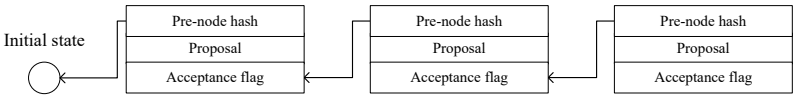

Fig. 8: Hash of path

A consensus contains information about:

Depth The number of times the world line has been extended, with an initial state depth of 0;

World line hash Hash value of the end of the world line;

Final state Formed after superimposing all proposals on the previous consensus state.

Signatures from both sides To ensure that both sides approve of the content of the consensus.

Depth and world line hash are collectively referred to as *consensus progress*. The significance of a consensus is that it determines the path for extending the world line and cannot be denied by either side.

4.3.1 Consistency signature

Reaching a consensus depends on a signature process on both sides. Since the forecast line is growing dynamically, an *anchor point* needs to be selected as the location for the next consensus. The path from the consensus to the anchor point is the *forecast path*. The *coordination sequence* describes the order of the nodes on the forecast path and is denoted by the letter S. Both sides jointly sign to decide which path is appended to the end of the world line. Consistency signatures study the question of how many signatures at least are required so that neither side can deny or modify a state. The concept of *signature pattern* is introduced for this purpose. A signature pattern is a pattern formed by the signatures alternately signed by both sides. For example, AB indicates that A signs first and then B signs.

For any side A, since consensus requires at least one signature from each side, only paths with signatures AB or BA need to be considered. Any combination of these two signature patterns with the forecast path may be like:

$$S_1BA, S_2BA, S_3AB, S_4AB, S_5BA, S_6BA, \ldots$$

In order to reach a consensus, A keeps signing after B has signed, for example continuing to sign S_4 to form the signature pattern ABA. From A's point of view, the path

with the most signatures is therefore unique, which is S_4. However, from B's point of view, there are two problems:

- B can also uniquely determine a path with the signature pattern BAB, such as S_5 or S_6, and signatures of S_4 are not always the longest.
- A may also have added a signature to S_3. That is, S_3ABA and S_4ABA do not mutually exclude each other.

For problem 1, the signatures can only start with A. A is called the leader and B is called the follower. For problem 2, B only needs to continue signing on the basis of the three signatures to form the signature pattern ABAB. If both sides are dishonest, no one will be harmed; otherwise, as long as one side is honest, it is impossible for both S_3ABAB and S_4ABAB to appear at the same time.

At this point, we can give an accurate criterion for consensus: **With alternating signatures and the leader signing first, consensus is the forecast path with no fewer than 3 signatures and the most signatures**.The entire process involves at least 3 communications,as shown in Figure 9.

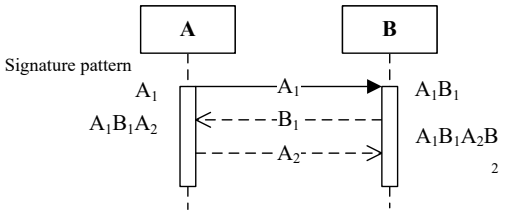

Fig. 9: Signing process to establish consensus

To be fair, the two sides take turns being the leader. At the current depth, the leader is the side whose polarity is equal to the depth modulo 2. After consensus is reached, the space of the world line is reclaimed, and only the consensus is retained by the host to save space.

4.3.2 Consensus preparation

The process of fixing the anchor point at some position on the forecast line, and the two sides forming a consensus on the next state through interactive messages is called *consensus preparation*. Consensus preparation synchronizes the follower's coordination sequence with the leader's.

Messages

Consensus preparation defines the following messages, all of which carry consensus progress:

Proposal Send one or more nodes of the forecast path to the other side, either actively or triggered by a download. Individual proposal can be part of specific business message.

Reset request Sent when local consensus lags behind remote consensus, expecting a reset command from the other side. The consensus progress corresponds to the consensus to be synchronized to.

Reset command Contains the consensus information to which the consensus progress in the reset request points. After receiving the reset command, the signature of the consensus is verified first, and then compared with the local consensus. If it is later than the local one, the local data is unconditionally replaced and the forecast line is also cleared.

Coordination Sends the coordination sequence to the other side. the local one, ignore the message. If the con

Download The follower downloads one or more nodes from the leader's forecast path.

Sign The host signs the new consensus in the order of the signature pattern during the consensus building process.

When a message is received, check the consensus progress first. If the consensus progress does not match the local one, ignore the message. If the consensus depth is

greater than the local one, send a reset request. If response is required for a message, there should be a timeout and resend mechanism.

Synchronization

The synchronization process may involve sending a number of coordination messages. A coordination message is indicated by the letter M and is distinguished by a subscript. Before sending the coordination message, set the local anchor point to the end position of the forecast line.

The coordination message serializes the coordination sequence into a series of bits. Each node on the forecast path is represented by two bits, with the odd bit being the author and the even bit being whether it is accepted. The maximum creation sequence of both sides can be calculated based on the number of odd bits 0 or 1. Finally, a byte is added to record the number of valid bits in the previous byte. This encoding method, in which the width is fixed and the storage order corresponds to the logical order, is called *ordered bit encoding*. Figure 10 implements a forecast path with 9 nodes encoded in 4 bytes.

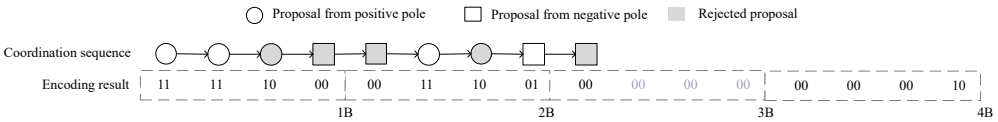

Fig. 10: Encoding of coordination sequence

Either side can propose M_1 to enter the consensus preparation stage:

- If sent by the leader, when the follower receives M_1,it first checks the data and downloads any missing nodes locally from the other party. The download message also uses Ordered Bit Encoding, with each bit representing whether or not to download a node.

- If sent by the follower, when the leader receives M_1,it refers to the local forecast path and sends the proposals that are missing from the other party and the local coordination sequence M_2

After this is complete, the leader has transmitted all the information about the local forecast path to the follower and sends the first signature of the consensus as Figure 9. After verification by the follower, the local forecast path is rearranged according to the leader's coordination sequence and the anchor point position is modified. Nodes that are not on the forecast path are arranged after the anchor point.

Synchronization does not affect the continuing submission of proposals, but the signatures of nodes after the anchor point will be invalidated in the future due to changes in consensus progress. In order to reduce unnecessary communication, local proposals with a height greater than the leader's anchor point are not sent, and remote proposals with a height greater than the leader's anchor point (including those in the queue) are not received and are deleted.

Conflict resolution

The attitudes of the leader and followers towards the proposal may conflict. The follower processes each node on the forecast path starting from height 1:

1. If the integrity check fails, the proposal is rejected. Otherwise, proceed to step 2.
2. Compare the local and remote acceptance flags. If they are inconsistent, follow the system's divergence strategy. The divergence strategy determines whether a certain type of proposal is accepted and both sides have the same behavior.

If the conflict is resolved (the local and remote coordination sequences are consistent), the follower sends a second signature, the leader verifies it and sends a third signature, and then the follower appends a final signature.

If the conflict still exists, the follower sends M_3 without adding new proposals, and the leader resolves the conflict in the same way as the follower and restarts synchronization. If both sides follow the same rules, they can reach agreement after a maximum of two synchronizations and conflict resolutions. The complete process is described using time sequence chart in Figure 11.

Cleanup

The forecast path becomes the world line after consensus is reached. However, before the consensus progress can be updated, some cleanup work needs to be done:

1. New nodes on the world line are sequentially input into the *closing function* and then deleted. The closing function uses the node and the previous state as arguments. The closing function can make new proposals, or call an external procedure, or both, but idempotency must be guaranteed, especially when calling an external procedure.
2. For the proposals on the forecast line, the creation sequence is rearranged starting from 1 and an integrity check is performed. Those that fail are discarded and the remaining ones have their signatures modified.If a proposal is discarded, its creation sequence will be used by the next node.

Once the above-mentioned clean-up work is complete, the consensus progress is updated. The world line can be deleted to save space, and the proposals on the forecast line are sent to the remote host. At the new consensus depth, the roles of leader and follower are swapped, and proposals that were not included in previous consensus have the opportunity to be applied this time.

Reset

Peer-to-peer consensus supports the reset request and reset command. If local data is corrupted, it is possible to restore it from another host to the latest consensus. However, peer-to-peer consensus only guarantees that an honest party will not suffer

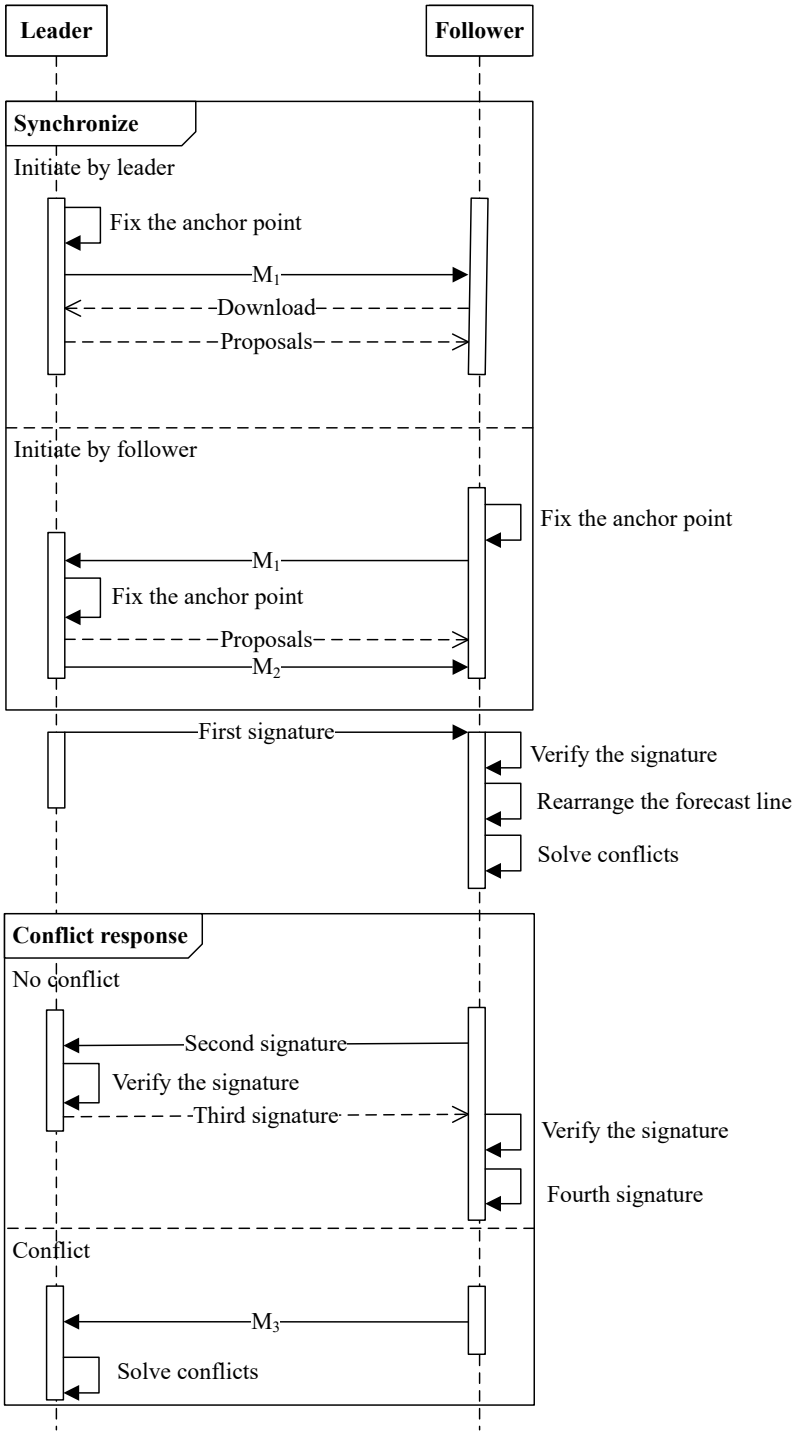

Fig. 11: Consensus preparation

losses if the data is intact, and does not guarantee the reliability of the reset command. If no consensus with deeper depth and longer signatures is found in local,the content of the reset command must be accepted. Use of peer-to-peer consensus means that you are willing to independently assume responsibility for storage and the risk of data loss.

4.4 Recommendations for implementation

When implementing peer-to-peer consensus, the system designer should first analyze the system state, define the state structure, initial values, the state constraint.The proposal is then defined according to business needs, including the structure of each proposal, preconditions, additions, review procedure, divergence strategy, and closing function. The above are called *consensus model* of the system.

Also, the trigger condition for consensus preparation needs to be determined to balance the system performance and resource consumption. The simplest way is if the forecast line is not empty, wait for a period of time and then the leader sends a coordination message. If the follower is late in receiving the coordination message, it can send it by itself, and all proposals added during this period will have a chance to be confirmed. World Yuan adopts this approach, where the waiting time is called the *window period*, and the follower waits for more than two times the window period to send a coordination message.

5 Network

World Yuan provides stable network facilities and routing service for the execution and feedback of purchasing power.

5.1 Consumption path

All participants in social exchange can be regarded as nodes in a directed graph with weights. In Figure 12,the right holder is upstream, and the duty bearer is downstream. The weight of an edge indicates the upstream's equity at the downstream.

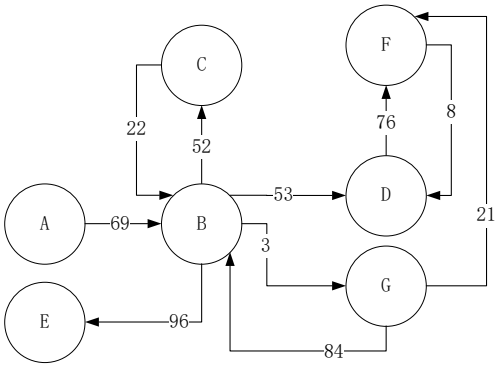

Fig. 12: Model of social exchange relationships

If two nodes are connected, the maximum commodity value that one side can obtain from the other will not exceed the bottleneck cost of path. This is called the *path flux* of consumption. For example, in the case of Figure 12, C can obtain F's commodities through the consumption path $C \to B \to G \to F$ or $C \to B \to D \to F$, with path fluxes of 3 and 22. However, if the purchasing power to be executed exceeds 3, only the second one can be used.

It can be proved that for any directed graph, if the following two conditions are satisfied:

1. Both the out-degree and in-degree of all nodes are non-zero;
2. There exists a reachable fixed node (sink);

then the graph must be strongly connected.

It can be derived that: when path flux is not considered, if each host both initiates and accepts promised outputs, and there exists a consumption path to some sink, then there must exist mutually reachable consumption paths between any two hosts.

To guarantee the universal transferability of purchasing power,**for non-sink hosts, the prerequisite for accepting promised outputs is that there must exist a valid consumption path from the output initiator to the sink.** When there are multiple sinks in the network, the system will naturally partition into several strongly connected components. As long as a host in one component designates a host in another component as an access point, these two components will achieve topological fusion. This mechanism causes the network topology to always exhibit a trend of evolving towards a strongly connected graph.

To ensure the stability of intermediate nodes, the host that requests the inclusion of a route must retain a certain equity at the upstream, which is referred to as the host's "*viscosity*". The guaranteed value is the sunk cost of exchange, while viscosity is the opportunity cost of withdrawing from exchange. Both viscosity and path flux describe the equity of nodes in consumption routing, but in opposite directions. Guarantee converts machines and electricity into guaranteed value, which is then converted into the utility required by the person doing calculations. Profit, viscosity and guarantee are incentives for the host to operate continuously.

Consumers can place minimum requirements on the viscosity of nodes through which consumption must pass. There may be multiple paths that meet the requirements for path flux and minimum viscosity, which requires an evaluation of path quality. The greater the path flux, the more value that can be redeemed; the longer the path, the more delay and uncertainty. It can be defined as:

$$Path\ quality = \frac{Path\ flux}{Distance} \tag{8}$$

5.2 Consumption routing

Flooding is a method for finding all paths to a destination. A node forwards a routing message to all nodes that satisfy a certain condition except the source, until it reaches the destination. Appendix A provides an example and validation code for finding a target node with flooding method.

World Yuan also needs to consider issues such as the viscosity of next hop and the path flux.A routing message contains the following fixed fields:

Token The producer determines whether the request is legitimate or not by token which is described later.

Minimum flux The lower limit of the path flux.

Minimum viscosity The minimum viscosity required to join a consumption path.

Producer's HGL It's used to judge whether the current node is connected to the producer.

Minimum flux and minimum viscosity restrict which hosts can act as intermediate nodes. Before initiating a route request, the consumer and producer exchange a pair of random numbers X_1, X_2 to compute the token as shown in Figure 13.

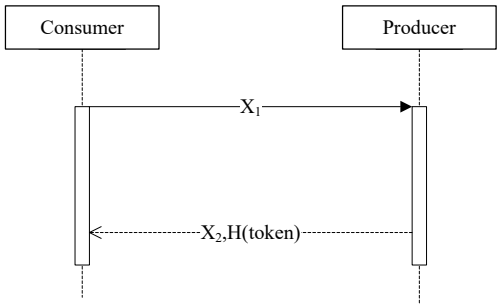

Fig. 13: Get (Update) the token

And

$$token = \text{Hmac}(X_1||X_2||\ Producer's\ HGL, \text{Hash}(Pe))$$

This is called *pre-routing*. Each token has an expiration date. After obtaining it, it is filled in the routing message and broadcast to the network. If the token expires, it will be rejected by the producer. This mechanism ensures that routing requests can only be initiated by the consumer.

In addition, the message needs to maintain three dynamic fields, which are updated as it is forwarded:

1. Bloom filter. Duplicate hosts through which the message has passed are not allowed, which is checked by the Bloom filter. Each host can be calculated based on the token to get a node ID:

$$NodeID = \text{Hash}(Token||HPI)$$

   Add the node ID to the Bloom filter before sending the routing message, and check whether current node's ID is in the Bloom filter after receiving the message. If it is, the message will not be forwarded. When using a Bloom filter with 7 hash functions and a capacity of 36 bytes, the false positive rate for paths with no more than 30 nodes is less than 1%.Re-initiating the routing can reduce the probability of the same host being misjudged.
2. Host list.The host list keeps track of all the edges the route has passed through over network. It is initially empty and will be filled in with the HHL of the next hop as the message being forwarded. The HHLs in the host list are also calculated using the token as random number, so it is impossible to analyze the host list to find out the nodes the message actually passed through.
3. Path flux.The path flux is initially set to infinity. Each time a message arrives at a node, it is compared with the equity at the downstream and updated.

When the message arrives at the destination host, the producer caches the host list, minimum viscosity, and path flux for retrieval by the consumer. During the gathering

process, if the cache is full, there are two policies: randomly discard the paths in the cache, or stop receiving subsequent routing messages with the same token. The complete flow for processing routing messages can be found in Figure 14 and Figure 15.

In practice, to reduce the scope of forwarding, hosts that can be used as intermediate nodes are often managed separately. These hosts provide stable service and have high credibility and are gradually separated from ordinary hosts. They are fixed as production or transmission facilities for purchasing power, known as value net. The consumers and producers situated at the periphery of network constitute the resource net. Commodities are transported into network by producers and out at consumers,as in Figure 16.It can be named "network differentiation".

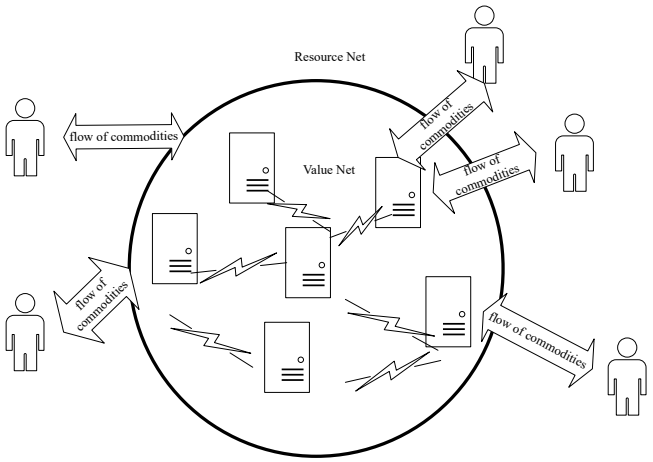

Fig. 16: Network differentiation

Network differentiation improves system service level. A host can join or leave the resource net at any time. If the network cannot contact the producer, the redemption will be canceled due to unreachable.

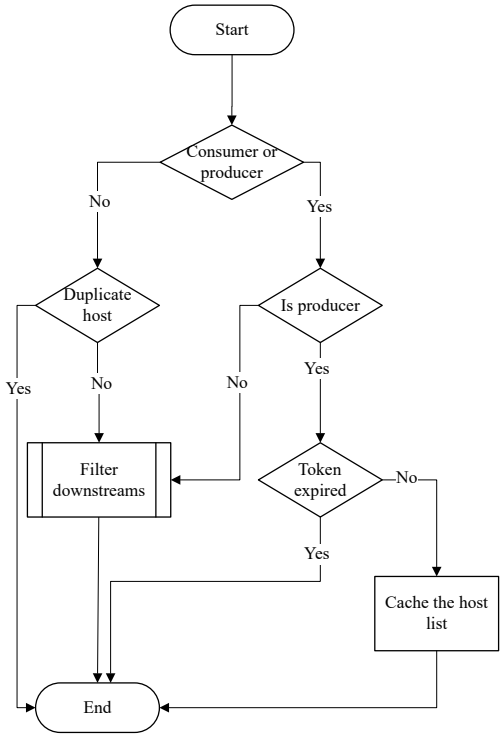

Fig. 14: Routing message processing

Filtering downstreams is an attempt to try the producer or each connected host as the next hop:

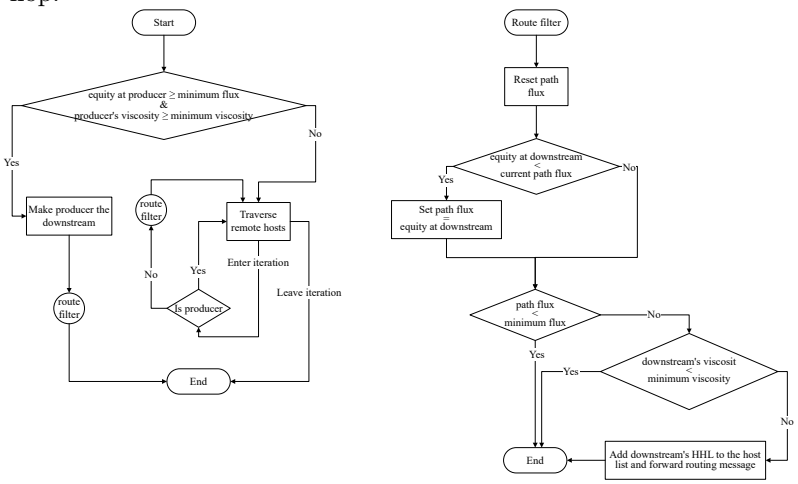

Fig. 15: Route downstream filter sub-flowchart

5.3 Routing fetch

The consumer should send routing message and fetch the collected consumption paths from the producer before the routing token expires. The producer will clear the data after the token expires or the result is fetched.

The routing result contains token, host list, minimum viscosity and path flux, which are pulled by the fetch message. The fetch message can specify the maximum number of paths to be returned. If it is set to 0, all results will be fetched; otherwise, the producer will sort the paths in the cache by quality and fetch every $\frac{Total\ number}{Maximum\ number}$ path. For example, if a total of 20 paths are gathered and a limit of 4 is set, path 1, 6, 11 and 16 will be fetched. This is to reduce the impact of invalid paths.

The consumer stores the results in a local routing table and marks the paths as unknown. When redeeming, the paths are taken out according to the principle of unknown priority and quality priority. If a path is unreachable during execution, it is deleted from the routing table. After the redemption is end, the path is marked as available.

5.4 Fair overflow

The network is constantly transmitting routing and redemption messages, and the capacity of the host is limited. To prevent denial-of-service attacks, World Yuan implements a *fair overflow* strategy for messages on the network. Incoming messages are stored in the fair overflow queue and their sources are recorded. A high water mark is used to limit the maximum number of queued messages. If the number of queued messages exceeds the high water mark, earlier messages from the link that has occupied the most queue space are discarded until the queue falls below the low water mark.

Suppose a host has established four links A, B, C, and D. The low water mark is 18 and the high water mark is 25. The messages in the queue, represented by link and

entry order, are:

$$A_1, C_2, A_3, B_4, D_5, A_6, B_7, D_8, C_9, C_{10}, B_{11}, A_{12}, D_{13}, D_{14}, A_{15}, C_{16}, B_{17}, A_{18}, D_{19}, C_{20}$$

Suppose there are 12 more messages waiting to enter the queue. They are:

$$D_{21}, A_{22}, B_{23}, C_{24}, A_{25}, D_{26}, C_{27}, B_{28}, A_{29}, D_{30}, C_{31}, B_{32}$$

According to the rule, $D_{21}, A_{22}, B_{23}, C_{24}, A_{25}$ can enter the queue directly and fill it up to the high water mark. Table 3 shows the occupancy of the queue.

Table 3: Fair overflow queue usage

Link	Messages	Count
A	A_1 A_3 A_6 A_{12} A_{15} A_{18} A_{22} A_{25}	8
B	B_4 B_7 B_{11} B_{17} B_{23}	5
C	C_2 C_9 C_{10} C_{16} C_{20} C_{24}	6
D	D_5 D_8 D_{13} D_{14} D_{19} D_{21}	6

Before inserting D_{26}, messages that must be discarded to fall below the low water mark are listed in Table 4.

Table 4: Table of messages to be discarded

Discarded message	Water mark	Count
A_1	24	A:7 B:5 C:6 D:6
A_3	23	A:6 B:5 C:6 D:6
C_2	22	A:6 B:5 C:5 D:5
D_5	21	A:6 B:5 C:5 D:5
A_6	20	A:5 B:5 C:5 D:5
B_4	19	A:5 B:4 C:5 D:4
D_8	18	A:5 B:4 C:5 D:4
C_9	17	A:5 B:4 C:4 D:4

At this point, the queue becomes:

$$B_7, C_{10}, B_{11}, A_{12}, D_{13}, D_{14}, A_{15}, C_{16}, B_{17}, A_{18}, D_{19}, C_{20}, D_{21}, A_{22}, B_{23}, C_{24}, A_{25}$$

The remaining messages ($D_{26} - B_{32}$) can be added to the queue smoothly.

When an attacker sends a large number of messages, they are blocked at the first hop. This allows each link to make full use of the network while equally occupying resources during peak usage. If the network is to provide differentiated services, multiple fair overflow queues can be configured. For example, if redemption has a higher priority than consumption routing, a separate queue can be set up for redemption-related traffic.

6 Labor proof

For a substance to serve as labor proof, it must meet certain conditions:

Standardization Labor proof is essentially the material carrier of human labor, and its existence represents the corresponding human labor consumption.

Supply elasticity When the economy grows rapidly, commodity production speeds up, the demand for purchasing power increases, and the guaranteed value required also increases. Coupled with increased market competition, labor power will shift to the guarantee industry. When economic growth slows down and the demand for purchasing power weakens, the human and financial resources invested in guarantee will flow back to the production sector. Guarantee have the function of regulating the social division of labor, and labor proof must be produced on demand without being restricted by raw materials.

Portability Labor proof must be easy to divide, store and transport. Portability leads to scarcity: if a substance is easy to produce and obtain, its value density is too low.The more labor proof must be transferred to exchange a certain value of the commodity.

Uselessness Guarantee is a type of useless labor. Labor proof has no utility and cannot compensate for the negative returns of fraud. This is why the premium of gold far exceeds its industrial and aesthetic uses.

Currency is a poor form of labor proof, and information can be transmitted quickly through networks, making it easy to coordinate social exchange and the ideal form of labor proof. For the purpose of electronic exchange, a method of digitizing labor proof must be found.

6.1 Calculation volume

The calculation volume is the number of times a certain calculation is performed. Let f be a uniformly distributed hash function with an output space of $[0, T)$. The average calculation volume by constructing different messages m until $f(m)$ is less than a constant φ is:

$$E(\varphi) = \frac{T}{\varphi} \quad \varphi \in (0, T] \tag{9}$$

where T is the upper limit of the output space of the hash function. And φ is called the *calculation scale* and the smaller it is, the more attempts are needed to find a matching input.

One unit of calculation volume can be defined as $E(\varphi)$ hash calculations.

6.2 Calculation proof

In order to defend against replay attacks, "m" must contain at least the calculation object, the accumulated guaranteed value, and a random number. The calculation object declares for whom the labor is being performed. A valid m value is a proof of 1 unit of calculation.

The human resources consumed in the production, depreciation, and maintenance of computers can be converted into equivalent human labor.Multiple calculation proofs constitute a digital labor proof whose labor volume is the calculation volume.

Use N to represent the random number and HPI of Bob to replace the calculation object. If Alice calculated 300 units for Bob and then calculates 100 more units, and the price of the calculation volume is 1, then a valid labor proof would be in the format:

$$\{HPI_b, N_1, N_2, N_3, \ldots, N_{100}\}$$

100-unit calculation proof by Alice

Each calculation proof N satisfies:

$$f(HPI_b||300||N) < \varphi$$

6.3 Space optimization

When computing power grows rapidly, labor proofs can become very large.
For example in order to represent a labor volume of ten thousand, ten thousand calculation proofs must be contained.To overcome the problem, the calculation scale φ is replaced by the difficulty target ω:

$$\omega = 2^{-k} \cdot \varphi \quad (k \geq 0) \tag{10}$$

Add k to the hashing process of the calculation proof as well, and then compare the result with the difficulty target, which must satisfy the following inequality:

$$f(HPI_b||k||300||N) < \omega$$

300 is the accumulated guaranteed value assumed above. Accordingly, the calculation times that need to be attempted are:

$$E(\omega) = 2^k \cdot E(\varphi) \tag{11}$$

Each calculation proof represents 2^k of calculation volume, and k is called the difficulty coefficient. Halving the difficulty target doubles the calculation volume represented by the calculation proof. The labor volume corresponding to n calculation proofs is:

$$2^k \cdot n \quad (n \leq \omega) \tag{12}$$

7 Metrology of value

Value is objective in nature. Misperceptions can arise if commodity value is confused with the daily concept of value, or with price. Some common fallacies are analyzed in the following.

Certain commodities seem to increase in value after they are produced. Take wine as an example, aged wine and ordinary wine are two different commodities.The moment when such commodities are produced is not determined, the survival is production until it is withdrawn from circulation. Buying a ten-year-old wine is the same as buying a currently produced wine that has been fermented for ten years, even if it has not been kept in the cellar. Conversely, keeping a car that was produced twenty years ago for a long period of time is not part of the production and will not increase its value. Unless it is a vintage car from the last century with collector's value, which has undergone more maintenance to satisfy people's vintage aesthetic needs.

If you confuse value with price, some commodities seem to decrease in value with the development of productivity. Before the end of the 19th century, people had not found an effective way to extract aluminum from nature, so the metal aluminum was

very scarce. It is said that when the French Emperor Napoleon III entertained his guests, everyone else used gold or silver cups, but he used aluminum cups alone, and the value of aluminum cups today seems to be far lower than it was at the beginning. Unlike vintage cars, it is difficult to tell whether an aluminum cup is a contemporary handicraft or a Napoleonic luxury item. But whether or not one can tell the difference between two identical cups, the value of an aluminum cup from 100 years ago and a modern one is determined at the moment it was produced: once one finds a way to differentiate between the two, the price moves back towards the value, which is why antiques are valued more highly than objects of the same utility. **The value of a commodity that has not been produced changes over time, but the value of an existing commodity is certain**.

The price of some obsolete electronics, such as cell phones, does not return to value, but this has nothing to do with value. Price is dependent on exchange and is distorted by supply and demand.It can be approximated that:

$$\text{Price} = \text{Value} \times \frac{\text{Demand}}{\text{Supply}}$$

Even if there is no exchange, the labor spent on old cell phones is a fait accompli.

Market economy has a certain degree of blindness, value is an important factor affecting price but not the only one, and it is impossible to achieve absolute fairness in exchange as a form of spontaneous distribution.Under private ownership, a minority who control scarce resources (e.g., land, minerals) can exchange them for goods embodying greater value from others with minimal labor input for transformation or utilization, thereby profiting through unequal exchanges.

Value is also independent. Labor volume is absolute, but the value to complete the labor volume is relative. Labor volume cannot stand in the place of value. Currency is a confusion of value and labor volume, since the value of a certain mass of currency is

constantly changing. As an independent quantity, value must be metered again. Using value rather than labor volume as the measurement of value is the most essential difference between World Yuan and the currency.

7.1 Units

When talking about exchanges, the *value unit* and the *exchange unit* are always implicit in .

Dime is the basic value unit, 1 World Yuan equals 10 dimes. Just as there is no time shorter than Planck's time, matter can be continuously divided, but at a certain level of division, the value becomes so tiny that it approaches transaction costs. Exchange value less than 1 dime is meaningless, and any exchange value is an integer multiple of 1 dime.

Exchange unit is an indivisible unit of commodity in exchange. It can be a physical unit, such as hour, meter, kilogram, liter,or a packaging unit, bottle, barrel, dozen, box and so on. Multiple exchange units can exist at the same time, as cola can be sold by can or by dozen.

The smallest exchange unit is determined by the social productivity of the commodity. Commodities such as gold, which have a low productivity, represent a significant value even when measured in grams. But for commodities such as sand, a unit that is too small corresponds to very little value. In manual sand digging days, the value of 1 kilogram of sand might be higher than 1 dime, and the use of kilogram is possible, but when the sand mining ships, conveyor belts and other equipment popularized, the productivity of sand increased, and 1 kilogram of sand became insignificant, and then people generally use the ton as exchange unit.

Generally speaking, with the development of productivity, when the value of a smallest exchange unit commodity can not be expressed in integer [9],it is necessary to

[9] When rounding by dime will cause a large error in commodity value.

consider upgrading the exchange unit. "Dime" is both a value unit and the smallest exchange unit of purchasing power.

7.2 Pricing

World Yuan can be used as a tool for studying the price of any commodity over time, regardless of the exchange unit or volume. Assuming that the calculation volume is Q and the value is V, price of per calculation volume:

$$P = \frac{V}{Q}$$

As a measure of value, any price can be expressed as a positive integer ($P \geq 1$), then:

$$V \geq Q$$

In terms of calculation proof, ignoring collisions, $Q \leq \varphi$. Thus, it is sufficient to define the minimum of V as below:

$$1\ World\ Dime = \varphi\ Tiny \tag{13}$$

"*Tiny*" is the smallest value unit, used exclusively for high-precision pricing. Since φ is extremely large(refer to Section 8.1) , it can be used even in the smallest physical units to satisfy the needs of general commodity pricing.

It is a gradual process that the value of all commodities including currencies be measured in World Yuan, not the other way around.

7.3 Datum

According to the two factors[10] that determine value, the dime can be expressed in terms of the labor volume of a standardized commodity at a given moment in time. Let's stipulate that:

1 dime is the labor necessary in June 2024 to produce $\frac{5}{11}$ kWh electricity.

In other words, 1 unit of electricity is worth 11 dimes in June 2024 with 5 kWh as the exchange unit. Even if the value of electricity decreases gradually in the future, it does not affect the dime itself. The values are designed according to the following considerations:

1. Lower than value of most exchanges.
2. Much higher than the cost of a single host to store and process an exchange.
3. The minimum value of a labor proof is 60 dimes. Can't generate too many trivial labor proofs, but the reliable value needed for small transactions can't be too large.

8 Guarantee

The host creates guaranteed value by presenting labor proofs to the exchange object. World Yuan uses BLAKE2b algorithm to output 256 bits for guarantee. A labor proof is a tuple of

{HPI, Accumulated guaranteed value,Set of calculation proofs,Difficulty coefficient}

8.1 Constants

1. Calculation scale

$$\varphi = \texttt{0x0000000000000046F2F2B0D9F0941C1770D82025656F4AFFD4214357BD851546}$$

One calculation volume corresponds to 260 PetaHash operations.

[10] The physical characteristics and production time of labor products.

2. The guarantee ratio is 120%.
3. Number of calculation proofs must not exceed 85 and total price must not be less than 60 dimes.

8.2 Price limitation

A quote must be submitted with the labor proof. The host keeps a rolling record of all quotes sent or received during a given period of time, which is called calculation quotes.In calculation quotes,calculation volume is denoted by $C_1 \dots C_n$ and the price is denoted by $V_1 \dots V_n$.The average price is denoted by P:

$$P = \frac{\sum_{i=1}^{n} V_i}{\sum_{i=1}^{n} C_i} \tag{14}$$

The host sets the calculation cost at initialization—the cost of completing 1 calculation volume locally. When someone receives a labor proof, he or she first trys to add the offer to the calculation quotes and the unit price of the calculation volume must not be higher than the *limited price*, while the quote that exceeds it is an overflow offer that is directly nullified. The limited price is different in different cases:

- the total sample number of calculation quotes $n < 2$: the limited price is 500% of the calculation cost.
- $n \geq 2$: the limited price is equal to the average price plus 2σ if the calculator did not make an overflow offer last time, otherwise it is equal to the average price minus 2σ. σ is the standard deviation of the calculation quotes:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (\frac{V_i}{C_i} - P)^2}{n}}$$

Every time you receive a calculation proof offer from an external host, you have to log whether it is an overflow offer or not, and use it to determine the limited price for

the next time. If the other side sends overflow offers twice in a row, it means that its computing efficiency is far below the network average, and it is forbidden to guarantee again for 24 hours, which is a reasonable time left for the other side to upgrade its equipment.

8.3 Quotation strategy

By default, when the size of calculation quotes data is less than 2, quote at calculation cost, otherwise, try to quote at the larger of calculation cost and average price. If an overflow offer is made to a target, you can choose to reduce the subsequent price by an appropriate amount while not dropping below the calculation cost. The submitter does not know the average price at remote and tends to follow the local average price to avoid making overflow offers.

A professional calculator can either guarantee for himself to sell purchasing power to unspecified clients or act as a third side to provide calculation service to traders.

9 Output

Outputting value is to output purchasing power equal to commidity value[11] , which increases the outputted value of the output side and equity (of the other) at this side. In the case of instant output, equity is reduced and the inputted value is increased.

Exchange is the combination of two outputs with opposite directions and equal quantities into one operation.

[11] Profit is included in commodity value indicated here.

10 Execution

The outputted value must be inputted. A consumer can execute equity at any access point to make a specific demand for commodity, as long as he/she knows the consumption path to the target commodity. Execution will reduce the consumer's equity at access point.

10.1 Session

The host pair is the minimal commodity circulation structure. Execution is transparent to the host in host pair. Upstream asks downstream for an execution of purchasing power, and downstream either rejects it, cancels the execution, or accepts it. Eventually, an execution either succeeds and the value is inputted (determined by the upstream), or fails and be cancelled (determined by the downstream).This process can be represented by Figure 17.

The context of a given redeem is called a *session*, and there may be multiple sessions between hosts in both directions at the same time, and a session is always adjacent to either the upper or lower session,as shown in Figure 18.

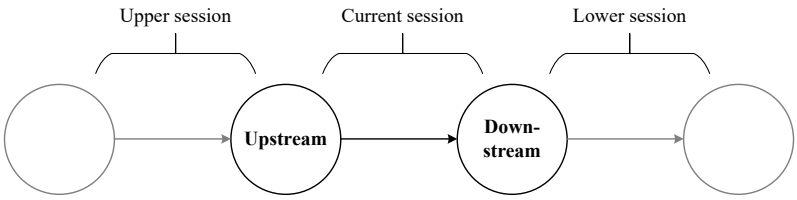

Fig. 18: Associated sessions

Except for the consumer and producer, a host is both the downstream of the upper session and the upstream of the lower session. When redeeming, the upstream initiates a new session to the downstream and assigns a unique session ID; after redeeming, the session is destroyed.

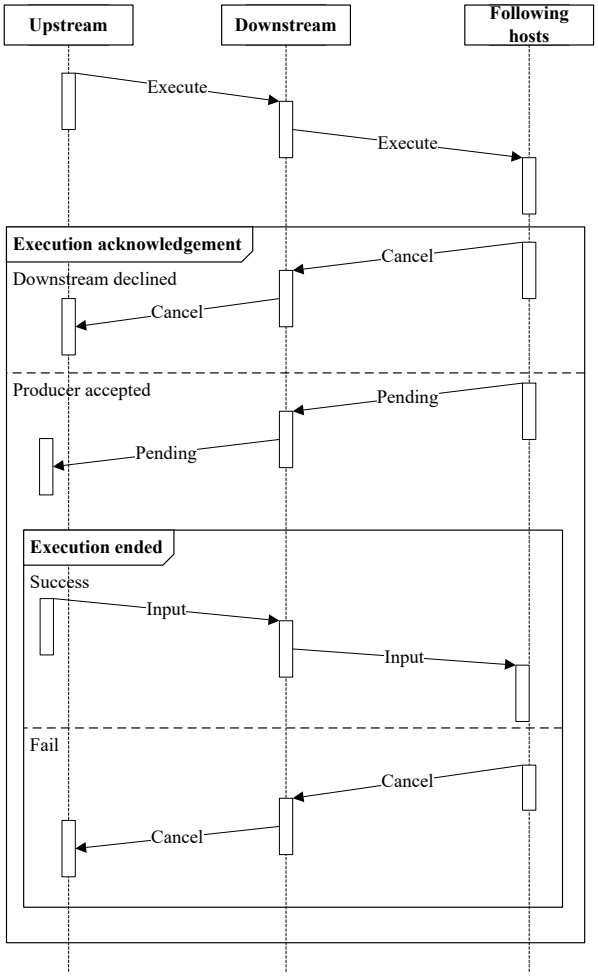

Fig. 17: Redemption sequence chart

10.2 Sharding

Sharding refers to the use of multiple consumption paths by a consumer to execute the same order. The value of the order is divided into multiple pieces and executed in different sessions. Sharding can spread equity across different access points, reduce the flux requirement of a single path, and reduce the impact of individual abnormal session.

Each piece has a value of $V_1 - V_n$. To distinguish between the objects to be executed in different sessions, the consumer and the producer exchange a pair of random numbers X_1 and X_2 during order negotiation.

The consumer sends the number of pieces, order information, and random number X_1 to the producer, and the producer generates a random number X_2 and sends it back to the consumer. The consumer then calculates the hash of the order based on this:

$$order\ hash = \text{Hmac}(X_1||X_2||Order\ information||V_1||\cdots||V_n, \text{Hash}(\text{Hash}(Pe))) \tag{15}$$

The *trace code* is obtained by hashing the order hash with the specific piece number:

$$Trace\ code = \text{Hash}(Order\ hash||Piece\ number) \tag{16}$$

The *execution subject* is the hash value of the trace code, which will be passed all the way down the consumption path together with the "execute" directive.When the producer detects a directive matching its own order based on the execution subject, it can update the status of the piece. If the execution subject matches the local order, it is called a local subject and is only valid at the consumer and producer.

The consumer divides an order into pieces according to a certain algorithm, and sequentially retrieves multiple[12] paths whose flux is not less than the value of piece from routing table. The minimum viscosity of each path to handle a piece is limited according to the risk appetite. Host list, minimum viscosity, execution subject and so on are written into the execute directive for each piece, and sent through access point of the consumption path. Each time the directive passes through a host, the HHL in the host list is taken out to determine the next hop and a session is established. An error at any point along the way will cause the current session to be cancelled,

[12]The minimum number of paths is 1, and it is less than or equal to the number of pieces.

affecting the upstream session until all previous sessions have been canceled. If the error is generated by the producer, the error message and signature are also pushed upstream.

Figure 19 depicts the possible consumption paths of entrusting an order to 5 sessions for execution.

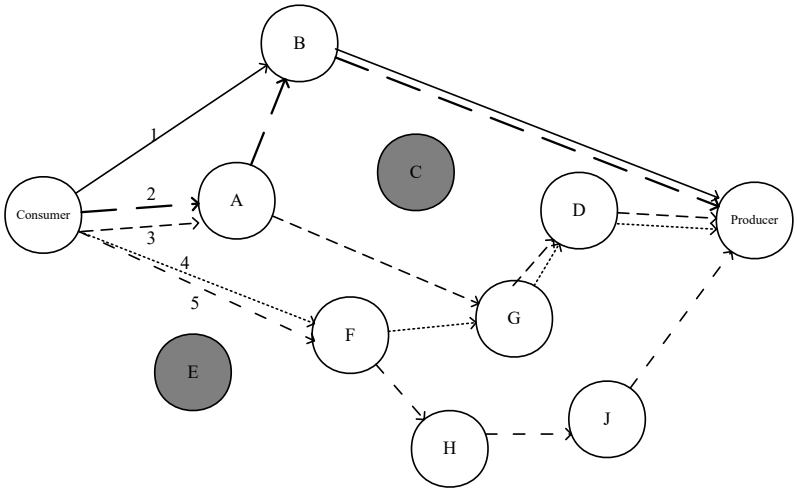

Fig. 19: Paths traveled by pieces

Different types of lines represent different paths. For example, piece 3 reaches the producer via A-G-D. If the execution of a piece fails, simply retry by replacing another path, without affecting other sessions. Notice that the nodes with dark backgrounds in the figure are not in any path. This may be due to their viscosity or insufficient equity.

10.3 Convergence

The execution of all pieces of one order is finally combined for evaluation, which is referred to as convergence. Convergence includes execution convergence and input convergence.

Execution convergence is reached when the execute directives of all pieces have been accepted by the producer, and the commodity is only delivered after that. A session

will be canceled in the following cases, which may result in the failure of execution convergence:

1. The piece cannot be found in the order based on the execution subject and value
2. The order is canceled
3. Duplicate execution from different session on the same piece

After the commodity has been completely consumed, the consumer issues an input directive to each session where the piece is located, which is called input convergence. The input of current session will also trigger input of the lower session.

11 Consensus model

All unclosed sessions are stored in the state,which is defined via Table 5.All numeric values in Word Yuan are variable-length non-negative integers.

Table 5: The state definition and initial values

Field	Description	Polarity Direction[a]	Initial Value
Accumulated guaranteed value	Total value of labor proofs submitted	Calculator	0
Accumulated outputted value	Total quantity of purchasing power outputted	Output side	0
Accumulated inputted value	Total quantity of purchasing power consumed by the opponent	Output side	0
Equity	Purchasing power exercisable by the consumer at the output side	Output side	0
Session map[b]	Mapping of ID[c] to session	Upstream	Empty

All fields consist of two parts from both sides of the exchange, with polarity as the subscript.

[a]The "polarity direction" indicates which side the subscript corresponds to.

[b]The session map contains all pending sessions from both sides.See Table 6.

[c]Refer to Equation 19 for generation method of session ID.

11.1 Constraint

Both sides involved in the exchange must abide by the following constraint:

$$\begin{cases} Accumulated\ outputted\ value \leq Reliable\ value \\ Equity \geq 0 \end{cases} \tag{17}$$

Where

$$Reliable\ value = \frac{Accumulated\ guaranteed\ value}{\alpha} + Accumulated\ inputted\ value \tag{18}$$

11.2 Session

Table 6 shows the structure of a session. The session ID is calculated using the pair

Table 6: Session structure

Field	Optional	Description
Token	No	The token of the selected consumption path which is used with HHL to match remote host
Upper session HHL[a]	No	Remote host corresponding to upper session, filled in by upstream
Lower session HHL	Yes	Remote host corresponding to lower session, filled in by upstream
Execution subject	No	Hash value of the trace code, filled in by upstream
Quantity	No	Quantity of purchasing power to execute, filled in by upstream
Status	No	Status of the session
Routing error	Yes	Error code returned by a non-producer
Peer error	Yes	Error code returned by the producer, as well as the signature

[a]Fields about the upper and lower session enable the ability to push messages forward and backward.

entropy and execution subject as follows:

$$Session\ ID = \text{Hmac}(token||Execution\ subject, \text{Hash}(Pe)) \tag{19}$$

Session status

Possible session statuses are:

Sent The upstream has sent an execute directive to the downstream.

Pending The producer has accepted the execute directive for the piece.

Canceled The execution was rejected somewhere.

Inputted The consumer has confirmed that the piece's value was inputted.

If a session stays in the first two status for a long time, it becomes a zombie session.

Error enumeration

When canceling the current session due to failure of execution at the lower session or other reasons,the routing error or peer error must be set,which both support following values:

- Precondition not satisfied.
- Violation of state constraints.
- Active rejection—A host rejected the execution for other reason.
- Unreachable—Unable to connect consumption path based on host list.

Peer error additionally supports the following two values:

- Invalid subject—The producer cannot find a matching order.
- Duplicate execution—The producer has already accepted another session's execute directive for the same piece.

11.3 Proposals

Proposals in World Yuan are part of the directives, and are classified into three categories: guarantee, exchange and consumption,as shown in Table 7.

Since peer-to-peer consensus does not record business processes, applications should back up their own proposals.

Table 7: Proposals of World Yuan

Directive		Purpose
Guarantee		To increase guaranteed value. Sent with a labor proof
Output		To increase both sides' outputted value and equity
Consumption	*Execute*	When the upstream requires redemption
	Acknowledge	When the producer accepts redemption
	Cancel	For downstream to decline an execution
	Input	For upstream to confirm inputted value
	Destruction	To destroy a session with specific ID

All directives required by the system are described below, and they are all authenticated with traceless signature.

11.3.1 Guarantee

The laborer submits a labor proof to the labor object. Labor proof consists of a set of calculation proofs and the difficulty coefficient.See Table 8 for the directive definition.

Table 8: Directive guarantee

Content	Price,the labor proof
Formal Check	Price greater than or equal to 60
Precondition	None
Review Procedure	1.The number of calculation proofs is between 1 and 85 2.Each calculation proof has been verified 3.It's not an overflow offer
Addition	The author's accumulated guaranteed value+=[a]price
Divergence Strategy	Reject

[a]operator '+-' or '-=' indicates an increment or decrement operation on the corresponding value in the state.

11.3.2 Output

The exchange value can be expressed in terms of purchasing power quantity. World Yuan regards exchange as a special output directive, where both sides output equal quantities.

The definition of output directive is shown in Table 9. Instant output requires the *proof of delivery*, which is a proof that the output side has fulfilled its obligation to deliver. It can be a receipt, matching the recipient's warehouse entry information, or

Table 9: Directive output

Content	• Promised output quantity(positive pole) • Instant output quantity(positive pole) • Proof of delivery (positive pole),optional • Promised output quantity(negative pole) • Instant output quantity(negative pole) • Proof of delivery (negative pole),optional
Formal Check	The output quantity of both sides cannot be 0 The output quantity of the author cannot be 0; If the output quantity of both sides is not 0, then they must be equal If the instant output quantity is not 0, a proof of delivery is required
Precondition	None
Review Procedure	If the local host depends on a sink node and the other side's promised output quantity is not 0, then there must exist a consumption path through the counterparty to the sink If local host is not the author and its output quantity is not 0, user consent must be obtained If the other side's instant output quantity is not 0, try binding its delivery proof to the proposal
Addition	To output side: Accumulated outputted value += promised output quantity + instant output quantity Equity += promised output quantity Accumulated inputted value += instant output quantity
Divergence Strategy	Reject

a receipt provided by the postal system. It can be validated by the input side and can only be used once.

11.3.3 Execute

The upstream requests for the execution subject to the downstream, and the downstream does not conduct additional review, but may cancel it afterwards.See Table 10 for the directive definition.

If the upstream is the consumer, it is executing the local subject. If the downstream detects that the "downstream session HHL" is null on receiving the message, it means that he/she is the producer and is also executing the local subject. As an upstream, you should construct an "execute instruction" as follows:

Table 10: Directive execute

Content	token, upper session HHL,lower session HHL, execution subject, quantity,host list
Formal Check	1.All fields cannot be null except for the lower session HHL 2.The quantity is greater than 0
Precondition	The session ID does not exist in the session map
Review Procedure	None
Addition	Equity at output side -= quantity Add new session to session map with status "sent"
Divergence Strategy	Accept

1. Leave "lower session HHL" blank.
2. If it refers to a local subject, fill in random values for "upper session HHL", and fill in "execution subject", "quantity", "token" and "host list" as appropriate. If it is not the execution of a local subject, these fields should be consistent with the directive from the upper session.
3. If it refers to a local subject,the first HHL is popped up from the host list, which is the HHL of the downstream. Otherwise, the "lower session HHL" in the upper session directive is the downstream of current session. For non-local subject, the "upper session HHL" is calculated by the HPI and token of remote host of upper session.
4. If the host list is not empty, take the next HHL and fill in the "lower session HHL". A consumption path $A \to B \to C \to D \to E$ is used to demonstrate the entire execution process. A is the consumer and E is the producer. The HHL of host A pointing to B is denoted as $\overrightarrow{AB}$.

   Table 11 shows some of the fields of the execute directives issued by each upstream.

Table 11: Partial fields of execute directive by different upstreams

Session ID	Direction	Host List	Upper Session HHL	Lower Session HHL
0FF0C1	$A \to B$	$\overrightarrow{CD}$ $\overrightarrow{DE}$	Random value	$\overrightarrow{BC}$
7D23F2	$B \to C$	$\overrightarrow{DE}$	$\overrightarrow{BA}$	$\overrightarrow{CD}$
813889	$C \to D$		$\overrightarrow{CB}$	$\overrightarrow{DE}$
2D0632	$D \to E$		$\overrightarrow{DC}$	

11.3.4 Acknowledge

If the downstream sends an acknowledge directive to the upstream, it means that the producer has agreed to execute a certain order piece. The acknowledge realizes an end-to-end response from the producer to the consumer.See Table 12 for the directive definition.

Table 12: Directive acknowledge

Content	Session ID,trace code
Formal Check	All fields cannot be null
Precondition	The session exists and its status is "sent";the hash of trace code equals to the execution subject
Review Procedure	None
Addition	Change the session's status to "pending"
Divergence Strategy	Accept

11.3.5 Cancel

The downstream sends a cancel directive to the upstream indicating that the execution of the session is canceled.See Table 13 for the directive definition.

Table 13: Directive cancel

Content	Session ID,routing error,peer error
Formal Check	Session ID is not null; routing error and peer error cannot both be null
Precondition	The session exists and its status is "sent" or "pending"
Review Procedure	None
Addition	Upstream's equity at the downstream += quantity to be executed of the session Session status changed to "canceled" Set routing error, peer error
Divergence Strategy	Accept

If the cancel directive is forwarded from the lower session, the error information will be filled in exactly as that of the lower session; otherwise, fill the error information as appropriate.

11.3.6 Input

The upstream sends an input directive to the downstream, indicating that the consumer has confirmed the input of corresponding order for the session.See Table 14 for the directive definition.

Table 14: Directive input

Content	Session ID
Formal Check	Session ID is not null
Precondition	The session exists and its status is “pending”
Review Procedure	None
Addition	Change the session’s status to “inputted” The accumulated inputted value of the downstream is incremented by the quantity of session
Divergence Strategy	Accept

11.3.7 Destruction

When a redemption ends, the associated sessions can be destroyed to avoid accumulating useless data.See Table 15 for the directive definition.

Table 15: Directive destruction

Content	Upstream of session[a],session ID
Formal Check	All fields cannot be null
Precondition	The session exists and the status is “canceled” or “ inputted”. If it is “canceled”, the author can only be the upstream, otherwise it can only be the downstream
Review Procedure	None
Addition	Delete the session with this ID from the session map
Divergence Strategy	Accept

[a]Represented by polarity.

11.4 Closing function

World Yuan provides pre-defined closing functions to handle the exception with insufficient reliable value through *guarantee planning*, and to properly maintain sessions. These functions are processed before user-defined closing functions.

11.4.1 Guarantee planning

In the case of a rejected output directives, with reference to the state after addition:

1. Check that the other side meets the constraint in Equation (1) and, if the other side is issuing an instant output, make sure that the proof of delivery is valid and unbound.Exit if failed.
2. Then check that the restraint is met locally,and exit if successful.Otherwise calculate the reliable value still required, also known as the *value gap*:

$$Value\ gap = Accumulated\ outputted\ value - Reliable\ value$$

In order to facilitate the calculation,convert variables uniformly into a form based on the state before addition:

$$\begin{aligned} Value\ gap =& Accumulated\ outputted\ value \\ &+ Promised\ output\ quantity \\ &- Reliable\ value \end{aligned} \tag{20}$$

Equation (1) is equivalent to the trader's value gap not being greater than 0. The value gap may decrease due to the input of value,and the value in execution is the value in transit:

$$\begin{aligned} Value\ in\ transit =& Accumulated\ outputted\ value \\ &- Accumulated\ inputted\ value \\ &- Equity \end{aligned} \tag{21}$$

The user receives an alert of a reliable value shortfall, which contains both the value gap and the value in transit, and decides whether or not to guarantee. When the value

gap is reduced by guarantee, the required guaranteed value is α times the amount of the reduction.

11.4.2 Session management

Session-related directives drive changes in execution state, and the host must respond appropriately as shown in Table 16.

Table 16: Closing functions for session-related directives

Directive	Role Of Host	Acceptance Flag	Local Subject	Closing Function
Execute	Upstream	Rejected	No	Send the cancel directive to the upper session
Execute	Downstream	Accepted	Yes	1. Try the execution convergence and sends an cancel directive if an “invalid subject” or “duplicate execution” error occurs 2. Sends the acknowledge directive to the upstream
Execute	Downstream	Accepted	No	Attempts to connect to the lower session and sends the execute directive.If the connection fails, sends a cancel directive with unreachable error
Acknowledge	Upstream	Accepted	No	Send the acknowledge directive to the upper session
Cancel	Downstream	Accepted	No	Send the destruction directive to the lower session
Cancel	Upstream	Accepted	Yes	1.Send the destruction directive 2.If the routing error or peer error is “violation of state constraint”, remove the consumption path from local routing table
Cancel	Upstream	Accepted	No	Send the cancel directive to the upper session
Input	Upstream	Accepted	No	Send the destruction directive to the upper session
Input	Downstream	Accepted	Yes	Send the destruction directive
Input	Downstream	Accepted	No	Send the input directive to the lower session

12 Compliance

World Yuan supports anonymity, but this anonymity applies only to unrelated sides. For financial institutions, meeting anti-money laundering (AML) requirements may necessitate verifying user identities and implementing regulatory measures on accounts, in accordance with relevant laws, policies, and regulations. These compliance measures can be easily implemented at the application layer, and include, but are not limited to the following:

Enforcing KYC policies

Remote host controller can access full service only after providing complete information.

Verify computational power source

Financial institutions may require clients to disclose the origin of computational power used for labor proofs or submit a signature from legitimate computation providers. For high-value labor proofs, their value can be temporarily classified as unverified guarantee,and excluded from reliable value:

$$\begin{aligned} \textit{Reliable value} = & \frac{\textit{Accumulated guaranteed value} - \textit{Unverified guarantee}}{\alpha} \\ & + \\ & \textit{Accumulated inputted value} \end{aligned} \tag{22}$$

Legitimate computation providers may have already undergone tax registration and implemented KYC measures for their clients.They can issue signed attestations for their computations,leveraging all fields of labor proof outlined in Section 8 .Such a signature can be submitted as part of the guarantee directive's business data. Once financial institutions confirm the legitimacy of the guarantee, the corresponding value

is deducted from the unverified guarantee. This mechanism effectively prevents computational power from sanctioned countries or regions from entering the international financial system.

Restrict the proof of delivery

For instant output, the proof of delivery must originates from a responsible side who has conducted due diligence on the delivered items or possesses sufficient client information.

End verification

During order negotiations or prior to product delivery, producers may require consumers to provide identification, purpose statements, or relevant permits as supporting documentation.

13 Conclusion

World Yuan is both a system of exchange and a concept of exchange centered on value. World Yuan will have a profound impact on traditional economic structures and may become the infrastructure for the next generation of the digital economy. It is anonymous, naturally deflationary, and close to a centralized payment system in its operational efficiency and interaction,which is easy for most people to accept. World Yuan is designed as a universal exchange system rather than another speculative product,with purchasing power tied to value.

There will be an industry specializing in guarantee, providing equipment or computing service to all those who want to join World Yuan. So whether online or offline, with or without professional computing equipment, there will be opportunities to use the system, and the computing facilities currently available for blockchain can also be fully utilized.

Declarations

The authors have no competing interests to declare that are relevant to the content of this article.

Funding

No funding was received to assist with the preparation of this manuscript.

Appendix A Flooding for pathfinding

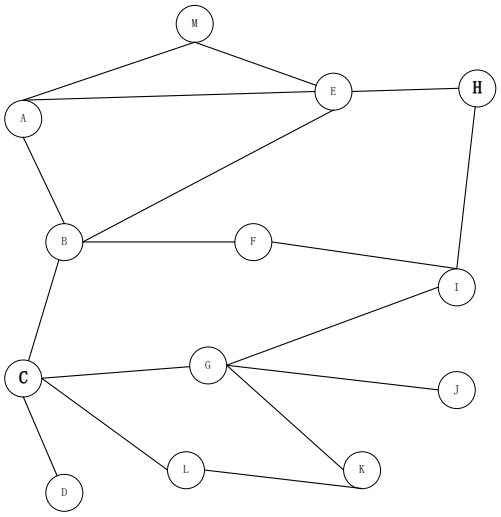

Fig. 20: Example of the network in which consumer C and producer H are located

In Figure 20,to find H using the flooding method from C, 27 paths must be tried, of which 12 are connected.This can be verified using the following Python code:

```
def dfs_find_all_paths(graph, current,path, all_paths):
    #Add the current node to the path
    path.append(current)
```

```
    can_extend = False
    if current != end_node:
        # Traverse all adjacent nodes of the current node

        for neighbor in graph.get(current, []):
            if neighbor not in path:  # Ensure that the node has not been visit
                can_extend = True
                dfs_find_all_paths(graph, neighbor, path, all_paths)

    # If the path cannot be extended further, record the current path
    if not can_extend:
        all_paths.append(list(path))

    # Backtrack, remove current node
    path.pop()

# Initialize the graph
graph = {
    'A': ['M', 'E', 'B'],
    'B': ['A', 'E', 'F', 'C'],
    'C': ['B', 'G', 'L', 'D'],
    'D': ['C'],
    'E': ['A', 'M', 'H', 'B'],
    'F': ['B', 'I'],
    'G': ['C', 'I', 'K', 'J'],
        'H': ['E','I'],
    'I': ['H','F', 'G'],
```

```
        'J': ['G'],
    'K': ['L', 'G'],
        'L': ['C', 'K'],
    'M': ['A', 'E']
}

#Set the start and end points of the path
start_node = 'C'
end_node = 'H'
all_paths = []

# Start searching
dfs_find_all_paths(graph, start_node, [], all_paths)

# Output all paths
print("A total of %i paths were tried:" % len(all_paths))
for path in all_paths:
    print("->".join(path))
```

References

[1] Mankiw, N. G. *Principles of economics* Eighth edition edn (Cengage Learning, Boston, MA, 2018).

[2] Marx, K. *Capital: a critique of political economy* Penguin Classics (Penguin books, London, 1992).