A Decentralized Web 3.0 Platform for Manufacturing Customized Products

Jiacheng Chen, Bo Qian, Haibo Zhou, and Dongmei Zhao

Abstract

Web 3.0 promises a more decentralized and valuable Internet by allowing users to truly own and control their data. This is in contrast to Web 2.0, where users and their generated data and traffic have been centralized by several major Internet platforms. In this article, we investigate the design of Web 3.0 platforms through a case study. Specifically, we propose a platform targeting the long-tail market comprising customized products. To handle the decentralized user demands, we realize the concept of virtual enterprise (VE) on the platform, such that different users can cooperate through running specific VEs with predefined workflows, thus greatly reducing the transaction costs. Furthermore, we develop the micro-blockchain protocol, allowing users to store consistent copies of transaction data on their own devices. Unlike existing coin-based blockchains, micro-blockchain does not require global consensus, hence it can naturally scale with more concurrent transactions. Moreover, a data-driven trust model is utilized, so that users can evaluate their trust relationship with others solely based on the data of their past activities, instead of requiring the help of a centralized trust authority, which is usually acted by the platform. At last, we point out the key issues on designing general Web 3.0 applications.

Introduction

Through the decades, Internet has incubated a variety of Web applications that greatly enrich our life. The way that users interact with the Web has also changed dramatically. In the so-called Web 1.0 era, a website is more of an online newspaper, since all the contents are put on the website by its owners. In other words, the Web is read-only to users. The most representative websites in this era are yahoo.com and google.com. However, the most charming fact of Internet is that it can easily connect users at different locations. With the development of Internet infrastructure and mobile Internet, high-quality and ubiquitous access to the Internet has become available for more and more people. Then, the Web soon turns into a place where all users can share their life, thoughts, creativity, resources, skills, etc. with others. Given the significant difference with Web 1.0 in that users can actively participate and contribute to the Web, Web 2.0 is so named, and Web applications have achieved unprecedented prosperity.

In Web 2.0, the most successful Web applications are Internet platforms, such as Facebook, Amazon, Airbnb, Uber and YouTube. Platforms are helpful since they greatly reduce information asymmetry and connect service providers and consumers with very low costs. However, platforms lead to several issues [1]. One is the protection of data property right [2]. Since user data is kept by the platform, users do not actually own their data, even with data protection laws such as General Data Protection Regulation (GDPR). Hence, users are under the threat of data leakage and data abuse. What is worse, data silos are created, so the value of data cannot be fully explored, and the owners of data (i.e., users) cannot benefit from the value. Another issue is the over centralization of Web. Due to the inherent network effect of platforms, only very few companies can survive in each field. Consequently, these companies will become ever larger, and will monopolize the market by controlling users' traffic and data, making them more difficult to be replaced.

In recent years, owing to the blockchain hype [3], the idea of decentralization has been prevailing. Then, Web 3.0 is conceived, aiming at decentralizing the Web. Web 3.0 allows users to truly own their data. As illustrated in Fig. 1, a Web 3.0 platform differs from its Web 2.0 counterpart mainly in that it does not keep user data. In other words, user data is decoupled from the application, and can be stored either locally on users' own devices or at a third-party data storage service provider [4]. Platforms still solve the information asymmetry problem, but authorization from users is required to use the data. However, user data generated from other platforms can also be accessed. For users, data becomes visible, controllable, traceable, and profitable. With the ownership of users' data guaranteed by Web 3.0, novel applications can be developed, e.g., for Metaverse.

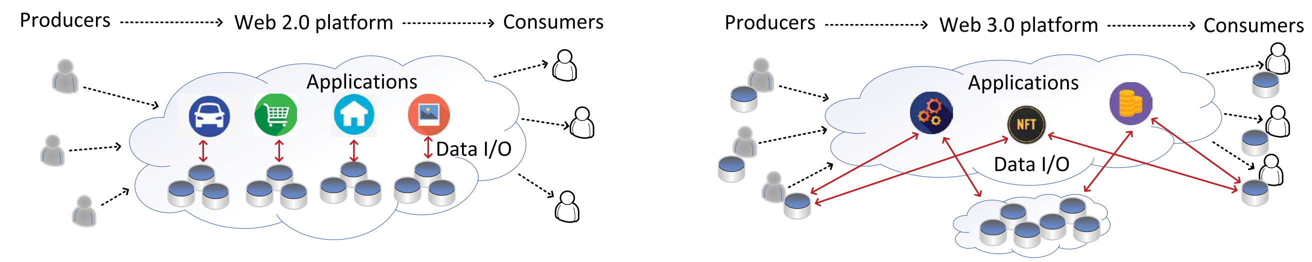
Currently, Web 3.0 applications are far from prepared to seize the market from traditional Web applications. It is also challenging to build a Web 3.0 platform. On the one hand, how to save the huge amount of user data efficiently and securely outside the platform needs to be addressed. On the other hand, without the trust authority acted by the platform, it is difficult for users to establish mutual trust and make transactions on the platform. To this end, we showcase a complete design of a novel Web 3.0 platform. The platform focuses on crowdsourcing users with various

Digital Object Identifier: 10.1109/MNET.2023.3318609
Date of Current Version: 29 March 2024
Date of Publication: 3 October 2023

Jiacheng Chen and Bo Qian are with the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen 518000, China; Haibo Zhou (corresponding author) is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China; Dongmei Zhao is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4L, Canada.

0890-8044/23©2023IEEE

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.



User data storage at **app servers**	
Pros	**Cons**
Data mining & analysis	User privacy
Personalized recommendation & Ads	Data leakage & abuse
Data monopoly	Data silos

User data storage at **third-party servers/local devices**				
Pros				
Visible	Traceable	Controllable	Profitable	Sharable

FIGURE 1. Difference between Web 3.0 and Web 2.0 platforms.

abilities to manufacture customized products. Unlike mass-produced products that have very large demands but only a few stock keeping units (SKUs), customized products have a large number of SKUs, but the demand for each SKU is small. However, the aggregate demands are large, forming the so-called long-tail market, which is essentially decentralized. It is difficult for a traditional company to make profits from the long-tail market, since without economies of scale, the costs will be extremely high. Instead, with a platform, users with different skills can easily form virtual enterprises (VEs) [5] and cooperate on making customized products. To build the Web 3.0 platform, user activity organization, user data storage and user mutual trust establishment are all designed in a decentralized manner. Therefore, the studied case is representative and our design can be generalized to other Web 3.0 platforms. The main contributions of this article are listed below:

- We realize the concept of VE for decentralized user organization and cooperation. We present modeling of a general VE and design the workflow for running a VE on the platform.
- We develop micro-blockchain, a decentralized and scalable user data storage scheme. We design deterministic consensus protocol and immutable data structure for micro-blockchain.
- We utilize a data-driven method for evaluating mutual trust between users in the decentralized environment. We introduce the transaction graph and calculate trust under the Bayesian inference framework.

The remainder of this article is organized as follows. In Section II, we introduce the modeling of VE and the workflow of manufacturing a customized product on the platform. The design of micro-blockchain is described in Section III. Then, decentralized trust calculation is presented in Section IV. Last, Section V concludes the whole article and discusses future directions on designing general Web 3.0 applications.

Decentralized Organization: Virtual Enterprise

Manufacturing products usually require multiple steps carried out by different parties. Therefore, the users crowdsourced by the platform must be organized for realizing efficient cooperation. As companies are such organizations in the real world, we resort to the concept of VE. A VE is a temporarily established, one-time organization for accomplishing a specific task. It can also be regarded as a "Co., Ltd." — company with limited lifetime. Since the costs of establishing a VE on the platform are very low, VE can quickly seize the emerging opportunities. In order to let users establish and operate VEs for different tasks easily, we need to model a general VE first. Then, the established VEs can operate in a decentralized way with all participants complying with the predefined workflows. Furthermore, the platform can generate the enterprise resource planning (ERP) systems for the VEs automatically.

> With the development of Internet infrastructure and mobile Internet, high-quality and ubiquitous access to the Internet has become available for more and more people.

A VE can be modeled as a multi-party, multistep transaction. Specifically, it is modeled by a set of participants, called Roles, a set of subtasks, called Nodes, and a set of procedures. A procedure is basically a state transition from one Node to the next, yet an execution result (success or failure) is associated with the transition as the condition. The procedure also specifies the Role assigned to undertake the Node. By connecting all the Nodes with procedures, the workflow of VE is determined. Note that a VE is similar to a state machine, but it can be in more than one states at the same time, so it allows parallel execution of Nodes.

To further illustrate the model of a certain VE, other powerful model description languages can be used, such as behaviour tree (BT). A simple example of a VE described by BT is given in Fig. 2. The task of VE is to manufacture a customized product related with an intellectual property (IP). The first Node is Product Design and is assigned to the Designer Role. The second and third Nodes are Design Reviews, assigned to Factory Role and IP Authorizer Role, respectively. These two Nodes can be executed in parallel. If the result of either Node is failure, then execution process returns to the Product Design Node again. Otherwise, the execution proceeds to the next Sample Manufacture Node, and will be completed by the Factory Role. Note that exceptions are not handled in this example for clarity.

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.

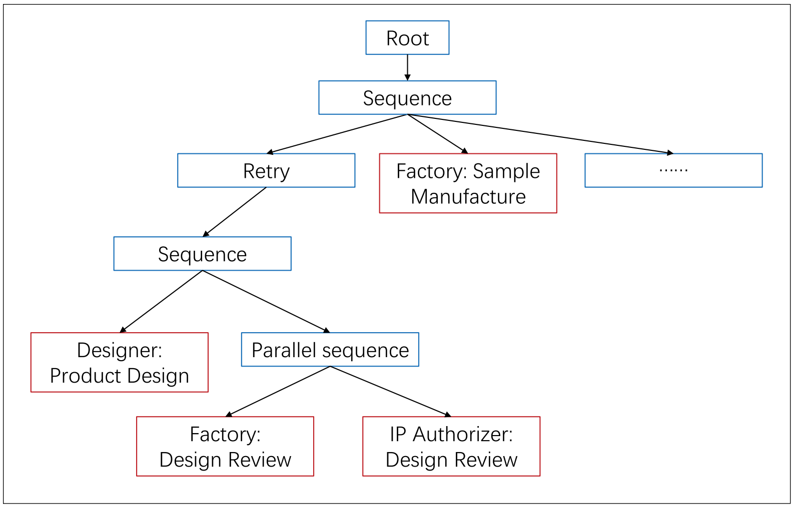


FIGURE 2. Behavior tree of a VE example.

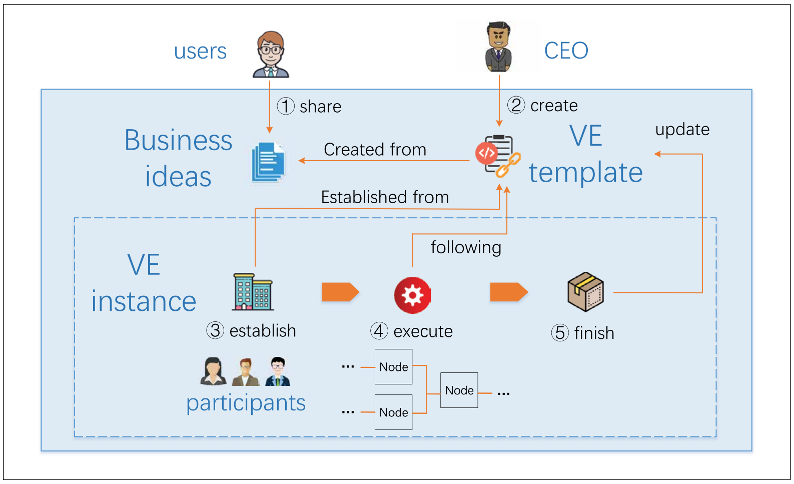


FIGURE 3. Lifecycle of a VE.

Based on the above model of VE, a user can easily create a new VE template on the platform through a proper user interface (UI), such as filling in a form. The user is regarded as the chief executive officer (CEO) Role of the VE. To create a VE template, first, CEO creates other necessary Roles of the VE. Then, CEO decomposes the whole task of VE into subtasks and creates Node for each subtask. Each Node is assigned to a previously created Role. Also, the predecessor Node and fallback Node should be selected when necessary. A Node's predecessor is where it comes from a successful execution, and its fallback Node is the next Node if it returns a failure result. There are also two special Nodes: one is the Start Node, which has no predecessor; and the other is Finish Node, which indicates the end of execution process. With specific translation rules, the VE template created from the above process can be translated into the BT model, and can also be used to generate the ERP system automatically.

The whole lifecycle of a VE can be finished on the platform through five steps, as shown in Fig. 3. In step 1, users share their creative business ideas or demands, e.g., describing a customized product that does not exist in the current market. In step 2, based on a business idea, a user (i.e., CEO) can create a VE template as described above. In step 3, based on the VE template, the CEO can establish a VE instance through an open recruiting process. Specifically, CEO will list the requirement and price (payment) for each Role, and other users can apply for the Roles. CEO will then pre-select a user for each Role, and starts a confirmation process. All pre-selected users can see each other during confirmation, and a user can freely quit at this time. If all pre-selected users confirm, then the VE instance is established, and the users become participants of the VE instance. In step 4, the VE instance begins execution automatically, following the workflow defined by the VE template. At each Node, after finishing the subtask, the user undertaking the corresponding Role can submit the Node together with some proof data. If the Node cannot be done, for example, the subtask is Review and the work from previous Node cannot pass the review, then the user can also reject the Node, leading to a failure result. Finally, in step 5, if Finish Node is submitted, then the VE instance terminates, and the participants get paid. However, the VE template still exists, and CEO can update the VE template or establish a new instance.

Decentralized Data Storage: Micro-Blockchain

The key feature of a Web 3.0 platform is that the platform assures users' control and ownership of their data. Therefore, we adopt a decentralized user data storage scheme so as to let users actually own their data by storing data on their own devices, such as laptops and mobile phones. For our platform, user data mainly refers to the data generated from running a VE, and the data property right belongs to the participants of the VE. The decentralized data storage system should meet the requirements of consistency and scalability. Consistency guarantees that there are no conflicts between different copies of data stored at each user, while scalability ensures that the performance (e.g., throughput and delay) of data storage will not degrade dramatically when the number of concurrent VEs on the platform is large.

Blockchain is a well known decentralized data storage technology. Each node stores a copy of the whole transaction data, and the consistency of data is guaranteed by the consensus protocol. Blockchains are often categorized into two types, namely public (permissionless) blockchain and consortium (permissioned) blockchain.

In a public blockchain, all the nodes can propose a block, which can be accepted by other nodes. Public blockchains usually face the scalability problem. For example, it is well known that the throughput of Bitcoin is about 7 transactions per second, and the delay (confirmation time) of a transaction is about 60 minutes, i.e., time for generating 6 blocks. One reason is the requirement of global consensus, which is resource-consuming and inefficient. Actually, it is not necessary for every single transaction to be processed by all the nodes. This idea leads to layer-2 technologies, including lightning network, side chain, sharding, etc. Some other interesting ideas also emerged from research. For example, Monoxide [6] partitions the whole network into several parallel

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.

chains (zones) so as to reduce the burden of each chain. Transaction atomicity across different zones is achieved, and Chu-ko-nu mining is designed such that the mining power of each zone is at the same level of the whole network, thus the security of each zone is guaranteed. In Flow [7], the whole workload is separated into the non-deterministic consensus part and deterministic computation part, which are assigned to two different node roles, respectively. By slicing the workload with different properties, the overall throughput can be improved.

Another reason is the randomness of consensus protocol, i.e., the proposer of a new block is randomly determined. For example, in the proof-of-work consensus protocol, the valid proposer is the node that solves a math puzzle first, and the corresponding probability is proportional to the computing power of the node. Such randomness will lead to forking, a temporary inconsistency of the global state. Although forking can be resolved later by the consensus protocol, it is a waste of resources and will degrade the system performance. Therefore, some research works attempted to improve the consensus efficiency by handling forking properly. In [8], the Conflux blockchain utilizes a Tree-Graph structure without discarding any forks, such that more blocks can be used. Algorand [9] exploits a cryptographic primitive called verifiable random function (VRF), which can deterministically conduct a sortition process. A block proposer as well as a set of committee nodes are selected from the sortition, thus the probability of forks is eliminated.

Compared with public blockchains, being a node of a consortium blockchain requires authorization. Hence, consortium blockchains usually have much fewer nodes, and can easily achieve a higher transaction throughput. However, the privilege for authorizing a node is usually controlled by an association, leading to high node management costs. Thus, consortium blockchains are mainly used for long-term cooperation among authoritative institutions or large organizations.

Inspired by the blockchain technology and its recent advance, we propose micro-blockchain, a lightweight decentralized mechanism tailored for storing data owned by multiple users. The main idea of micro-blockchain is to create an independent micro-chain for each VE, and the nodes of the micro-chain are the devices of users participating the VE. Besides, there will be no global chain used for synchronizing all the micro-chains, since there is no global state (e.g., users' accounts) to be maintained. Micro-blockchain is scalable since different micro-chains can work in parallel without conflicting with each other. Furthermore, although the platform is public and every user can participate in a VE freely, a micro-chain is only authorized to its nodes, i.e., participants of the VE. Therefore, users' data property right is protected, and the perimeter of each micro-chain is secured.

Same as modeling of a VE, we still treat a VE as a multi-party, multi-step transaction. Like other blockchains, a transaction represents a state change of the VE. To record the change, a block will be generated for the transaction and then stored on the micro-chain after consensus protocol. In the VE model, a transaction corresponds to a Node, and the initiator of the transaction is the Role assigned to the Node. Transaction data includes the proof data and execution result of the Node, which are submitted by the Role of Node. With each transaction recorded on micro-chain, the whole execution process of a VE can be easily validated.

Inspired by the blockchain technology and its recent advance, we propose micro-blockchain, a lightweight decentralized mechanism tailored for storing data owned by multiple users.

In the following, we elaborate on the consensus protocol of micro-blockchain. For each happened transaction, consensus should be reached on deciding how it should be stored by each user. There are two main tasks for the consensus protocol, namely to store the correct data, and to store the data on each user with consistency. Hence, consensus protocol contains two layers, namely user-level consensus, and data-level consensus. Note that most of the existing blockchains do not validate the correctness of stored data. However, we believe it is necessary for our platform to make sure that the stored data is correct, i.e., all the users consent to write the data, before it is immutably written on the blockchain. In order to avoid the troubles caused by forking, we adopt a deterministic consensus strategy. Specifically, the block proposer for each transaction can be pre-determined from the VE template. For example, the initiator of the transaction, can be assigned as the block proposer by default. All the other users complying with the consensus protocol will approve the assignment. Although single point failure may occur, failure of the user will also influence execution of the VE, which is a larger problem. In other words, since the users/Roles are non-fungible, assigning a deterministic block proposer will not be less secure than randomly selecting one.

The workflow of consensus protocol is illustrated in Fig. 4. At the current Node, the user undertaking the Role initiates a transaction by broadcasting the Node execution result and proof data to other users. The transaction is attached with a counter, and is cryptographically signed by the user. During the user-level consensus, other users receive the transaction and validate the contained proof data and signature. After a user successfully validating the transaction, a cryptographically signed response message with the same data as received will be sent back. The response message is regarded as a vote for the transaction. Each user will collect and verify the votes from others, until all valid votes are collected. At this time, user-level consensus is reached, otherwise, the consensus rolls back to the very beginning, and no data will be saved. The data-level consensus begins after user-level consensus. With the transaction data and all the received votes, the pre-determined block proposer generates the block and propagates it to other users, together with a counter and signature. When receiving the block, a user successively checks the identity of the block proposer, the validity of the block, and the validity of user-level consensus. If all the examinations pass, then the user agrees with the block through a signed vote and propagates it to other users. Similarly, data-level consensus is reached when all the users

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.

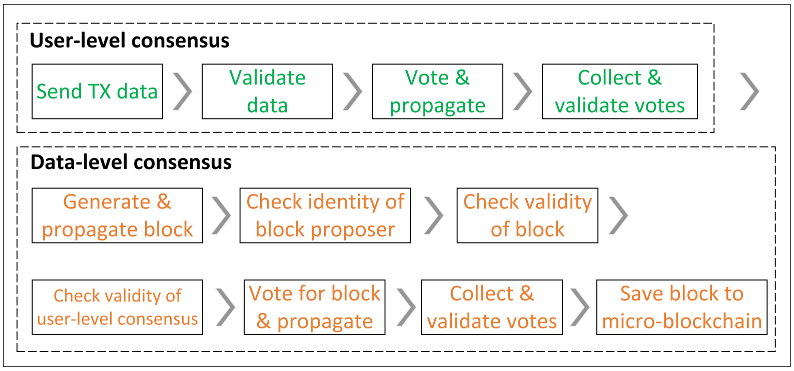


FIGURE 4. Workflow of consensus for micro-blockchain.

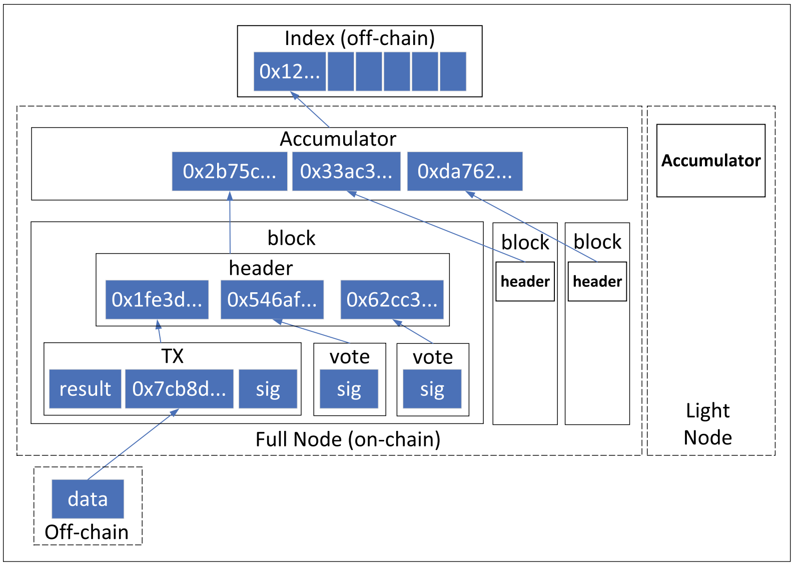


FIGURE 5. Data structure for micro-blockchain.

vote for the block. At last, the block is stored on chain and each user has a consistent copy. Since a block is generated for each transaction, the confirmation of transaction is fast.

At last, we describe the data structure of micro-blockchain, as shown in Fig. 5. We adopt an immutable data structure, which mainly exploit the cryptographic primitive called accumulator. A cryptographic accumulator is efficient for set membership tests [10], and it has been used by other well-designed blockchains such as Libra. In Fig. 5, an arrow indicates a hash operation from the original data to a fixed-length result, represented by "0x...." On the bottom, we can see that a transaction (TX) is at least composed of the hash of submitted proof data, result of the current Node and a cryptographic signature. The data itself can be stored off-chain, since the data may contain large files such as videos, which will consume a lot of storage space of the user's device. Besides TX, votes from the user-level consensus are also included in the block, with each vote containing a signature signed on the transaction data. The TX and votes are then hashed separately and the hash results constitute the header of the block. The header of each block is hashed again and is put into the accumulator. When the VE terminates, the accumulator is hashed and can be saved off-chain for indexing. In order to further save the storage space, a user can run a light node by only keeping the accumulators. With micro-blockchain, each user's data are essentially backed up by other users, and can be recovered easily from other users' devices.

Decentralized Trust: Transaction Graph Based Trust Calculation

Besides information asymmetry, another main obstacle for doing business is the lack of trust. Unfortunately, it is much harder to establish trust between users on a Web 3.0 platform. On the one hand, unlike a centralized Web 2.0 platform, the decentralization characteristic of a Web 3.0 platform prohibits it from acting as a trustworthy third-party to fill the trust gap between users. On the other hand, the user credit score system of Web 2.0 platforms are not applicable, since credit scores are mainly used for quantizing the trust that users earn from the platform. For example, a high credit score does not necessarily mean that another user will also trust the user for making a transaction. In addition, for a multi-party transaction, it is required that each pair of users should have a certain level of mutual trust, thus the total trust required is proportional to the square of the number of users.

In a decentralized environment without a common trust authority, trust can only be established by either direct contact or transitivity on social networks [11]. To this end, we introduce a data-driven decentralized trust model for our platform, based on the work by Jøsang et al. [12]. With the model, a trust value can be calculated for two arbitrary users, no matter whether they have made any transactions before. The calculation is solely based on the past activities of users, i.e., the transaction data stored on users' devices. Then, a user can quantitatively evaluate the trust with other potential participants of the VE, and thereby decide whether to join the VE.

In order to find all the potential connections between two users, an undirected transaction graph (TG) is generated based on all the finished VEs on the platform. A user is a vertex in TG, and an edge indicates that the pair of users have cooperated in at least one VE before. We adopt Bayesian inference to calculate the trust value of two neighbor users. In general, Bayesian inference can update the posterior probability of an event based on the historical observations on the event. We use a binomial event to model a VE, which can either be successfully ended or abnormally terminated. After the ending of each VE incorporating the two users, an observation will be obtained, and it is represented by a 2D vector taking the following values: [1,0] if the VE successfully ends, [0,1] if the VE abnormally terminates due to either of the two users, and [0,0] if the VE abnormally terminates due to other users in the VE. Further, as time elapses, the vector is discounted by multiplying a decay factor. Then, at the current time, the discounted vectors of all observations are added, and the two elements in the vector are used as the two parameters of beta distribution, whose expectation (a.k.a. the Bayesian reputation score) can be used to characterize the success probability of a new VE incorporating

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.

```
Initialization: P ← ∅, G ← ∅
Input: TG, source vertex S, destination vertex D
Output: An SPG G with two terminals S and D
Step 1: Find all the paths from S to D with breadth first searching (BFS)
for current path p during BFS do
    Calculate confidence for p by (s1) and (s2)
    if confidence < threshold then
        Stop searching from p
    if p reaches D then
        Save p to path set P
Step 2: Sort P by confidence (from high to low)
Step 3: Build G from empty graph
for p ∈ P do
    Add p to G
    if G is not an SPG then
        Remove p from G
        return G
```

ALGORITHM 1: Trust Graph Simplification

the two users. We interpret this success probability as the trust value of the two users.

Now we consider how to operate two trust values from two adjacent edges on the TG. Suppose we have three vertices A, B, C and two edges AB and BC. The first operation is trust discounting (denoted by ×), and it describes how trust is propagated from A to C through a common "acquaintance" B. The second operation is trust consensus (denoted by +), and it describes how trust is superposed at B from two different sources A and C. Then, subjective logic theory (SLT) is used to formulate the above two operations [12]. The theory uses four parameters δ = {*belief, disbelief, uncertainty, base rate*} to describe the judgement on an event. If the event is binomial, then beta distribution can be used, with base rate equal to 0.5. Then, we can calculate the trust value through following steps: (s1) parameters of beta distribution are transformed to SLT parameters following (9) in [12], (s2) trust discounting and trust consensus can be done by following the rules described in Section 6.3 of [12], (s3) the SLT parameters are transformed back to beta distribution parameters to calculate the Bayesian reputation score.

Before we can finally calculate end-to-end trust value between two arbitrary users, we still need to find all the paths first. Since different paths can have overlapping edges, only some of them will be used so as to avoid repetitive calculation. However, we should also take care of information loss when discarding edges. More formally, the objective of graph simplification is to generate a two-terminal series-parallel graph (SPG) [12]. For example, for the sixth topology shown in Fig. 6, the trust between user *A* and *E* can be expressed by $((AB \times BD) + (AC \times CD)) \times DE$. Note that each edge should only appear once in the simplified SPG, and this is called the canonical expression of SPG. SPG has the property that it can be transformed into canonical expression.

It is known that recognition of SPG can be done in linear-time. One way is to check whether there exists a vertex (except the two terminal vertices) in the graph that only has one predecessor vertex and one successor vertex. If such a vertex does not exist, then the graph is not an

For users joining the platform independently without invitation, they can start with participating in VEs with smaller economic scales and lower risk levels, so as to gradually connect themselves to TG.

SPG. Then, we can find the optimal SPG from TG for two users. As in [12], optimality is defined according to the confidence of SPG, which can be calculated from 1 – *uncertainty*. Since it is difficult to find optimal SPG from TG directly, a heuristic greedy algorithm is adopted so as to find a suboptimal SPG for trust calculation. The algorithm first finds all the available paths between two vertices with breadth first graph searching. During the searching, the confidence of current path is calculated, and the path will be discarded if its confidence drops below a threshold. Then, the suboptimal SPG is constructed from zero by iteratively adding a path into it in a greedy way, until it does not pass the check of SPG. We show the algorithm in Algorithm 1.

For a new user to join the platform, the most efficient way is through an invitation from an old user. It is supposed that the two users have a certain level of mutual trust in the real world, so a high observation value can be assigned to the users, e.g., [3,0]. Thus, the new user has an initial connection to the TG via the old user, and can have higher trust values with other users through the old user's connections. Besides, the old user will also benefit from the new user's successful transactions on the platform. For users joining the platform independently without invitation, they can start with participating in VEs with smaller economic scales and lower risk levels, so as to gradually connect themselves to TG.

At last, we conduct a series of experiments to illustrate how mutual trust between two users are affected by the happened transactions. Specifically, we evaluate totally six cases, and their topologies are shown on the right of Fig. 6. After a step, some transactions happen, and the trust values between terminal nodes will be updated. A red (green) edge shown in Fig. 6 indicates one more successful (failed) transaction happened during each step. For example, in case 3, user A and B, as well as user B and C, will make a successful transaction after each step, and the values of edge AB and BC will change. There are totally three groups of tests. The initial values of all edges are set to [1,0] by default. In the first group of tests consisted of case 1 and 2, we observe the trust value of two neighbor users A and B. After 9 steps, we can see that the trust value reaches about 0.9 and 0.1, respectively. Therefore, successful transactions can increase the trust value, while failed transactions affect the trust value negatively. In the second group, we test the effect of trust discounting from the comparison between case 3 and 4. After 9 steps, the former's trust value reaches about 0.8, while the latter is only about 0.6. The reason is that the enhanced trust between A and B cannot be transited to C via a weak trust relationship between B and C. Last, we evaluate the effect of trust consensus. We consider a basic trust chain in case 5, and compare it with case 6. In both cases, we also set the value of DE to [3,0]. We can see that with the existence of a parallel path, the end-to-end trust value in case 6 is higher, since trust relationship between

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.

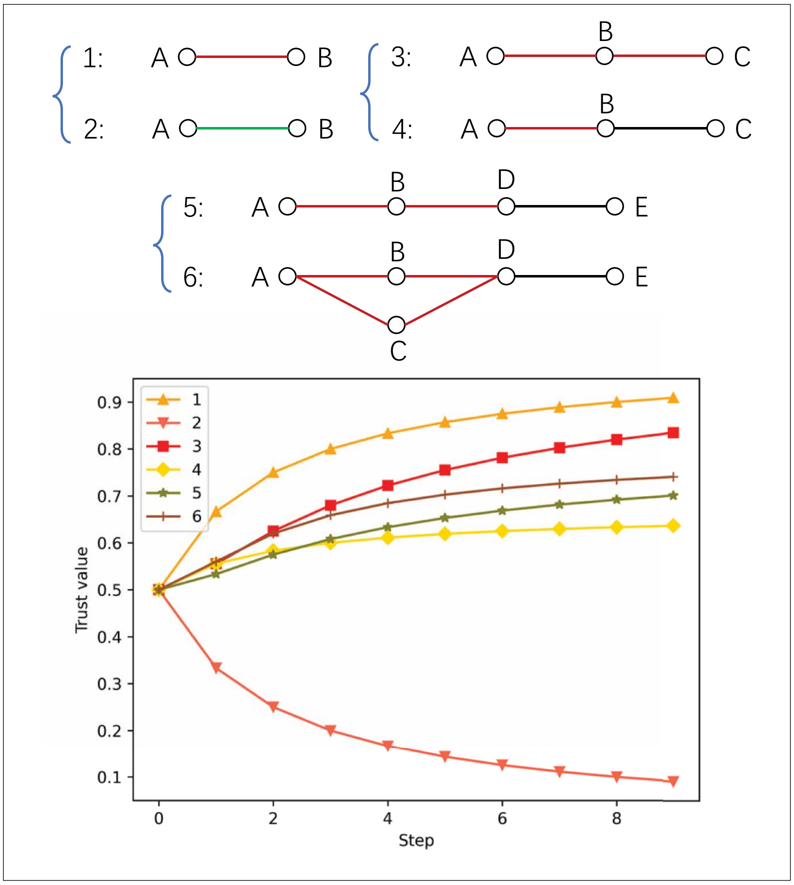


FIGURE 6. Trust value comparison for different cases.

A and D is strengthened. Also, E can be treated as a new user invited by the old user D, so it can utilize D's connections to establish a higher trust relationship with A.

Conclusion and Future Directions

In this article, we have showcased the design of a decentralized Web 3.0 platform that aims at manufacturing customized products and meets the decentralized long-tail demands in the market. With the developed VE model, users can cooperate by running different VEs in a decentralized way with low costs. User data is also stored in a decentralized way following the micro-blockchain protocol, with data correctness and consistency guaranteed. Furthermore, with the help of transaction graph and trust operations, mutual trust can be quantitatively evaluated between two arbitrary users based on their own transaction history.

Future Directions

1) Order Matching on the Platform: A VE is basically an order with a very small scale, so the average cost of producing a single unit of good is high. To reduce cost, the key is to reduce the marginal cost of a user when participating in a VE. An effective way in economics is to increase the scale of production. However, the mass production of standardized goods cannot be applied to manufacturing customized products. To this end, we utilize the theory of mass customization (MC) [13]. MC aims to provide customized products to users with the same level of cost as mass production. The idea is to transform customized production of a good into many batch productions of every part of the good. Through decomposition and reorganization of production workflow, the same part of the product can be reused by many different products, so the total demand on this part becomes larger, and the cost is reduced.

Based on the theory of MC, an order matching system can be developed on the platform, such that orders/VEs containing similar subtasks can be aggregated for a user, so as to reduce the average cost for finishing the subtasks. Order matching is a more efficient way to establish multiple VEs, since the platform can utilize global information to match the requirements between orders and users. It can also realize economies of scale for users. Suppose there exist a number of VEs on the platform awaiting for establishment. For each VE, the CEO sets a highest acceptable price for each Role on doing the subtasks. Each user is capable of undertaking some Roles of different VEs, and also has a lowest acceptable price for doing the subtasks. Besides, the total production scale of VEs that the user participates should be larger than a minimum amount, so as to reduce cost. The objective of order matching is to maximize the number of matched orders.

2) Research Directions in Web 3.0: Web 3.0 is inevitably a trend for the next decades, and will revolutionize the Internet towards a more decentralized paradigm. In order to let users truly own their data, we believe that the way to store user data must be changed. Besides blockchain-based solutions, a third-party data storage service can also be considered [4]. To access user data, each user should have a decentralized identity in Web 3.0, which can be realized through the Cybertwin [14]. As the user's entry point to the Internet, Cybertwin is naturally trustworthy to the user, so it can be used as the delegate to access the personal data. Authorization of data access can further exploit the zero-trust architecture (ZTA). With user data separated from the application, data communication and computing in such an application-data decoupled architecture also deserve further studying. An efficient and secure way for fetching data from users is required. In order to save network bandwidth and reduce the risk of data leakage, it is better requesting the result calculated from data than requesting the data directly. This can be achieved through function-as-a-service (FaaS) deployed at the data storage service provider. Another problem is how to standardize the format of various kinds of data, such that data can become interoperable among different applications. With the above prerequisites, study on data economy is indispensable. Without data silos and monopoly, data can be treated as assets and can be traded in the market [15]. Users can gain profits from data, and different companies can all acquire the complete set of user data to provide better services to users. Last, it is also interesting to integrate AI-generated contents (AIGC) with Web 3.0. As an example, product design in our case can be replaced by AIGC. Also, the creation of VE template can be assisted by fine-tuned large language models.

Acknowledgment

This work is supported in part by the Natural Science Foundation of China (NSFC) under Grant 62001259 and Grant 62271244, in part by

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.

the National Natural Science Foundation Original Exploration Project of China under Grant 62250004, in part by the Natural Science Fund for Distinguished Young Scholars of Jiangsu Province under Grant BK20220067, and in part by the High-level Innovation and Entrepreneurship Talent Introduction Program Team of Jiangsu Province under Grant JSSCTD202202.

References

[1] S. Listokin, "Does industry self-regulation of consumer data privacy work?" *IEEE Secur. Privacy*, vol. 15, no. 2, pp. 92–95, Mar./Apr. 2017.

[2] C. I. Jones and C. Tonetti, "Nonrivalry and the economics of data," *Amer. Econ. Rev.*, vol. 110, no. 9, pp. 2819–2858, 2020.

[3] Y. Lin et al., "A unified blockchain-semantic framework for wireless edge intelligence enabled Web 3.0," *IEEE Wireless Commun.*, early access, Mar. 7, 2023, doi: 10.1109/MWC.018.2200568.

[4] A. V. Sambra et al., "Solid: A platform for decentralized social applications based on linked data," MIT CSAIL & Qatar Comput. Res. Inst., Ad Dawhah, Qatar, Tech. Rep., 2016. [Online]. Available: http://emansour.com/research/lusail/solid_protocols.pdf

[5] B. Katzy and G. Schuh, "The virtual enterprise," in *Handbook of Life Cycle Engineering: Concepts, Methods and Tools*. U.S.: Springer, 1998, pp. 59–92.

[6] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2019, pp. 95–112.

[7] Y. Hassanzadeh-Nazarabadi et al., "Flow: Separating consensus and compute," 2020, *arXiv:2002.07403*.

[8] C. Li et al., "A decentralized blockchain with high throughput and fast confirmation," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 515–528.

[9] Y. Gilad et al., "Algorand: Scaling Byzantine agreements for cryptocurrencies," in *Proc. 26th Symp. Operating Syst. Princ.*, 2017, pp. 51–68.

[10] I. Ozcelik et al., "An overview of cryptographic accumulators," in *Proc. 7th Int. Conf. Inf. Syst. Secur. Privacy*. Setúbal, Portugal: SciTePress, 2021, pp. 661–669.

[11] O. Richters and T. P. Peixoto, "Trust transitivity in social networks," *PloS One*, vol. 6, no. 4, 2011, Art. no. e18384.

[12] A. Josang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proc. 29th Australas. Comput. Sci. Conf. (ACSW)*. Adelaide, SA, Australia: Australian Computer Society, 2006, pp. 85–94.

[13] S. Guo et al., "Inventory management in mass customization operations: A review," *IEEE Trans. Eng. Manag.*, vol. 66, no. 3, pp. 412–428, Aug. 2019.

[14] Q. Yu et al., "A cybertwin based network architecture for 6G," in *Proc. 2nd 6G Wireless Summit (6G SUMMIT)*, Mar. 2020, pp. 1–5.

[15] L. Xue et al., "Blockchain-based fair and fine-grained data trading with privacy preservation," *IEEE Trans. Comput.*, vol. 72, no. 9, pp. 2440–2453, Sep. 2023.

Biographies

Jiacheng Chen (Member, IEEE) (chenjch02@pcl.ac.cn) received the Ph.D. degree in information and communications engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018. From December 2015 to December 2016, he was a Visiting Scholar at BBCR Group, University of Waterloo, Canada. He is currently an Assistant Researcher with the Peng Cheng Laboratory, Shenzhen, China. His research interests include future network design, B5G/6G networks, and resource management. He has won the *Journal of Communications and Information Networks* (JCIN) Best Paper Award in 2016, the Chinese Institute of Electronics (CIE) Best Paper Award in Electronic and Information in 2020, and the Best Paper Award in PIMRC'23.

Bo Qian (Member, IEEE) (boqian@pcl.ac.cn) received the B.S. and M.S. degrees in statistics from Sichuan University, Chengdu, China, in 2015 and 2018, respectively, and the Ph.D. degree in information and communication engineering from Nanjing University, Nanjing, China, in 2022. He is currently a Post-Doctoral Fellow with the Peng Cheng Laboratory, Shenzhen, China. His research interests include resource management in B5G/6G networks, VANET, network economics, blockchain, and game theory. He was a recipient of the Best Paper Award from IEEE VTC2020-Fall.

Haibo Zhou (Senior Member, IEEE) (haibozhou@nju.edu.cn) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. From 2014 to 2017, he was a Post-Doctoral Fellow with the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. He is currently a Full Professor with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. His research interests include resource management and protocol design in B5G/6G networks, vehicular ad hoc networks, and space–air–ground integrated networks. He was a recipient of the 2019 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award, 2023–2024 IEEE ComSoc Distinguished Lecturer, and 2023–2025 IEEE VTS Distinguished Lecturer. He served as the Track/Symposium Co-Chair for IEEE/CIC ICCC 2019, IEEE VTC-Fall 2020, IEEE VTC-Fall 2021, WCSP 2022, IEEE GLOBECOM 2022, and IEEE ICC 2024. He is currently an Associate Editor of the IEEE Transactions on Wireless Communications, IEEE Internet of Things Journal, *IEEE Network Magazine*, and *Journal of Communications and Information Networks*.

Dongmei Zhao (Member, IEEE) (dzhao@mcmaster.ca) received the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in June 2002. In July 2002, she joined the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where she is a Full Professor. From April 2004 to March 2009, she was an Adjunct Assistant Professor with the Department of Electrical and Computer Engineering, University of Waterloo. Her current research areas are mainly in mobile computation offloading, energy-efficient wireless networking, and vehicular communication networks. Dr. Zhao is the Co-Chair of the Mobile and Wireless Networks Symposium of IEEE GLOBECOM Conference in 2020, the Wireless Networking Symposium in IEEE GLOBECOM Conference in 2007, and the Green Computing, Networking, and Communications Symposium in International Conference on Computing, Networking and Communications in 2020, and the Co-Chair of the Technical Program Committee for IEEE International Workshop on Computer Aided Modelling and Design of Communication Links and Networks in 2016. She has been in Technical Program Committee of many international conferences in her fields. She is a Professional Engineer of Ontario. She is an Associate Editor of the IEEE Internet of Things Journal. She served as an Associate Editor for the IEEE Transactions on Vehicular Technology from 2007 to 2017. She also served as an Editor for *EURASIP Journal on Wireless Communications and Networking* and *Journal of Communications and Networks*.

Authorized licensed use limited to: Nanjing University. Downloaded on March 28,2025 at 02:48:42 UTC from IEEE Xplore. Restrictions apply.