TiesDB: The Short guide

Abstract

The development of blockchain and crypto-economics inevitably pushes forward the market of decentralized applications. In fact, we are at the threshold of a whole new world that 100% repeats the world of the classical ("centralized") Internet: decentralized social networks, instant messengers, stock exchanges, business platforms, recruiting agencies, cloud storages, search engines, information aggregators, and the media. Thus, the market in the years to come will face a global infrastructure problem - the need to store large amounts of data in a structured way. This problem is not solved at this time.

What is the point of structured data storage?

At first glance it seems that everything is simple. In fact, decentralization creates a lot of difficulties and new problems that have to be solved practically on the go.

The first problem is WHERE to store the data? If there is no single server, then there is no guaranteed site where you can add information in a structured way. The decentralized cloud storage market offers a solution in the form of renting users' computers and storing files on them, but this solution is extremely limited in its capabilities, because the content of information is not taken into account - as a rule, these are simply files. No one can get access to their content or perform complex operations with them.

The second problem: HOW to store data? Simple files do not represent a data structure. They can not be linked to each other, neither one can search by the contents of the file. To search for data, you need to organize information in a special way, to provide a quick search on the contents of a file or a fragment of it, by title and by document content, with the ability to add a new document and delete the old one, modify the document, etc.

The third (and the most important problem): the issue of TRUST between the organizers of content storage in a decentralized network. The fact is that a public decentralized network assumes that everyone can become a member of the network. A free license for the product code assumes that any user who bought the server and installed the appropriate software can be an organizer of the storage (that is, a node). Members of a single network do not know each other and are not required to. They can be located in different parts of the world and even be in open conflict with each other, while effectively supporting the work of the database as a whole, performing their part of the work. The problem of trust between participants and the ways of interaction between those who do not trust each other by default, is a key problem of public, distributed, decentralized systems and the solution of this problem is a fundamental step in the development of the decentralized applications market.

And why not use a blockchain for such a database?

Some projects are now doing so. They simply put large amounts of content directly into the Ethereum blockchain. In fact, it seems logical. After all, the blockchain is: A) Database. B) Decentralized. C) It solves the problem of trust between nodes. It seems that this is the simplest solution, but the blockchain has several significant drawbacks that cross out its advantages for storing large amounts of content:

- Blockchain was conceived as a system for conducting financial transactions and ensuring their correctness. And it is completely unclear why to put large documents, photos and video there.
- At the heart of the blockchain is the formation of the block, the inclusion of blocks in the chain, the acceptance of the created chain of blocks by the decentralized participants. This process, due to its nature, is rather slow.
- Blockchain, by definition - is an immutable system. It was created with the goal of permanently fixing important actions or entities and preventing them from changing or forging (deleting) in the future. So, you can not put photos, videos and other documents into the block, which, due to their specificity, can be constantly added, deleted, changed.
- Blockchain is an immutable system, which means that when you publish a photo, then make changes to it and publish another photo, then publish the third, etc. - all the previous versions will also remain forever in the blockchain and will be available for viewing, which may not coincide with the wish of the photo owner.

What solutions are on the market right now? And what are their drawbacks?

There is a number of solutions on the market for creating distributed databases. The essence of those solutions is that a lot of computers are united in a single network and interact with each other as a single base. These solutions include Cassandra, MongoDB, etc. The main drawback of those solutions is that they are not intended for use in a public network. That is, a node containing a part of the database can not be an absolutely extraneous entity that does not pass the verification in a single trust center. Thus, the solutions described above for creating a distributed database are virtually centralized, they are subject to a single source that distributes access rights to content and allows you to join the network only to previously trusted sites that it trusts.

What should the new database be like?

The new type of database should be:

1. Public: anyone can join the network and become a node.
2. Decentralized: it should not have a single trust center and a control center.
3. It should provide all the capabilities of a distributed network: the content should be stored on different servers, yet used as a single piece of information.

4. The content should be stored in a structured form, providing the ability to quickly find information within it.

The most important problem that arises when you create this type of database is trust between nodes.

What is the "node's trust" problem?

The problem is that in a public network with open source, anyone can become a server (a network node). Once one of the network users has installed the necessary software, one automatically joins the network, accesses the network content and can make changes to it, add new content, and delete the old.

Since the new network user is never checked by anyone, he can be a spammer or may have the intention to destroy the entire network by making changes to the database that will block its operation.

To prevent it, the nodes must work on pre-verified mathematical algorithms and a methodology that will filter out malicious or malicious changes to the database from malicious users. All this should happen automatically, without the intervention of third parties and analysis of conflicts that can block the operating of the nodes. The network itself must be built in such a way that the "trust" between nodes is ensured by encryption algorithms, as well as algorithms for automatic content validation.

How does TiesDB solves the "nodes trust" problem?

As soon as the server becomes a network node, it automatically receives a portion of content that was previously added by other users from other sites.

Next, the site server users start adding new content or making changes to the old one.

Each content modification from the user's side is signed with a private key and paid for.

Signing with a private key certifies the source of the changes (that is, that the data has been changed by the user, and not by someone on his behalf), and payment is the motivation of the owner of the node to maintain the server.

Business platforms based on TiesDB should lay down in their business models of the project economy such conditions, under which payment for the storage of content is compensated to the user from the side of the project, as it is realized in Ties.Network. Thus, paid modification motivates the owners of the database hosting, and for the user those changes are actually free, because they are compensated by a business project created on the basis of the database.

The user-added content is stored simultaneously on different nodes, ensuring the reliability of storage in the decentralized network. If something happens to the server of the node (should it break, be closed, or should a server owner start to display incorrect or aggressive

behavior in relation to other network members), the user content will not be lost and the network will continue to operate as if the node that had dropped out of work had not previously existed at all.

A Node owner makes a security deposit on the smart contract that protects the network from node malicious behavior node. If the node is convicted of not storing content, but takes money from users for it, it will be fined for the amount of the security deposit and will be automatically suspended from working with other network members.

Algorithms for blocking malicious behaving node:

Potential problem	**Prevention algorithm**
The node does not store user content.	When a client (user program) accesses a node, content is taken from other nodes if the node does not have the content.
Delete or replace user content on a private server.	Since the content is signed with a user private key, the next time the content is accessed, the database will provide valid content from other nodes, and this node will be asked to change its content to a valid one. The same applies to valid, but old data. Nodes that issue old data will be asked to update them to newer data received from other nodes.
Making changes on behalf of others to other nodes (SPAM).	Each change is paid for
Deleting other users content on the server of another node. Delete the entire database.	Each content is signed with a private key and each modification is paid. Without a private key, you can not make changes to the content of others or your own. neither can you delete it.

How does TiesDB interact with the blockchain?

TiesDB is an independent entity and is designed for structured storage of large data sets in a public, decentralized, distributed network. Formally, it is not tied to the blockchain. Nevertheless, since TiesDB uses financial tools to motivate participants, it must coordinate its actions with the blockchain, store information on deposits, rates, and mutual settlements of participants. Such kind of information can be reliably stored only in the blockchain. In addition, in order to reconcile conflicting information about making changes to the database and related financial transactions, it is also necessary to use blockchain, for example, if information about the goods is stored in the database, information about their quantity should be kept in the blockchain. Storage of information on the quantity of goods only in the

database can lead to successful purchase of goods, which are no longer in stock. For preventing this kind of situation, you can also use blockchain.
In the first release, TiesDB will be compatible with Ethereum blockchain, which is a highly developed market for decentralized applications (https://dapps.ethercasts.com) now. In future TiesBD will be compatible with all popular blockchains and can be used for any projects in any blockchain community.

What business cases might need TiesDB?

The Market of decentralized applications in general might need it, and so any business cases where it is necessary to build a decentralized community, without a central server, where there will be many independent centers that need to be linked together into a single storage unit for data storage and processing.