UDK 621.09.04 http://doi.org/10.15407/pp2024.01.038

S. V. Strutynskyi, V. A. Yalanetskyi

PROGRAMMING OF ONE-DIMENSIONAL AND TWO-DIMENSIONAL TOKENS FOR TOKENIZATION OF LAND PLOTS

The use of blockchain tools that allows spliting virtual objects into parts is considered. Examples of practical use of the developed algorithms are presented. The concept of one-dimensional and two-dimensional tokens representing one-dimensional and flat objects is proposed. Algorithms for the implementation of one-dimensional tokens are developed, and the peculiarities of their practical application are considered. A designed smart contract allows to conduct a basic list of operations with one-dimensional tokens. Algorithms, providing implementation of two-dimensional tokens, are proposed. Peculiarities of presenting territories of virtual worlds and land plots are suggested. A comparative analysis of the use of NFT and two-dimensional tokens for presenting the Earth surface areas is performed. Methods that ensure ownership of tokens at different levels are proposed.
Keywords: Blockchain, EVM, smart contracts, NFT, fractional tokens.

Introduction

Blockchain technologies proved their effectiveness within creating distributed databases. They found many applications, including cryptocurrencies; tokenized assets presenting fiat currencies, shares, property and other financial instruments; non-fungible tokens representing digital objects; decentralized domain name databases [1].

Some of the tools and solutions based on blockchain technologies have been successfully implemented. Algorithms and software tools that have proven their suitability for practical use and resistance to attacks have been developed. However, current tools have limited capabilities preventing the creation of specialized blockchain-based solutions.

One of the long-term areas of using blockchain technologies is the implementation of database of land plots. Decentralized applications will allow transactions with real estate, land plots and virtual territories. The implementation of solutions, providing an opportunity to perform specialized transactions, requires the deployment of new blockchain tools. An infrastructure that is optimized for working with land plots and virtual territories are to be created.

The tools that are necessary for real estate transactions are at the initial stage of development. There are algorithms and simplified solutions that, if customized, could be widely used.

Analysis of recent research and publications

Cryptocurrencies, tokens and other virtual assets are characterized by different tokennomics. Virtual assets could be generated once within startup or periodically after certain time intervals. NFTs might have more complex emission mechanisms. Most cryptocurrencies and tokens have a variable volume of emission. If the number of coins in circulation fluctuates significantly, this leads to some difficulties, e.g. when determining the current asset capitalization.

Coins in the Bitcoin network are issued to pay miners who generate blocks [2]. The exponential decay of the payment for the creation of the block led to the fact that the volume of funds in circulation ceased to change significantly. The additional emission is insignificant compared to the number of coins in circulation.

Some cryptocurrencies are characterized by constant emission. Payments to miners or validators are conducted from funds paid by customers for the performance of transactions. Such an algorithm reduces inflationary trends significantly.

There are also cryptocurrencies whose emission volume is decreasing. The process is implemented by burning virtual assets. The source of funds could be commission fees, a part of which is withdrawn from circulation. An example of the implementation of such an algorithm is the Ethereum blockchain platform [3]. After the activation of the London hard fork, which includes EIP-1559 [4], a part of the commissions is withdrawn from circulation. The established algorithm is quite flexible, since the volume of ETH emission could decrease or increase according to external factors.

The proposal of stablecoins and tokenized assets is determined by the management company. Most often, their emission of tokens depends on the market demand for them.

Non-fungible tokens are put into circulation through other algorithms. The emission of tokens is implemented via smart contract, which allows you to link NFTs with a digital object. Most often, the token contains a file hash. In this case, the file is not stored on the blockchain, but the hash allows you to identify the data-set and link it to the NFT. In some cases, additional algorithms can be used when performing operations with tokens. In some cases, the author of the NFT can receive royalties from the sale of the token [5].

Financial instruments based on blockchain are characterized by different volume of emission. Most often, the emission of an asset is measured in millions, billions or quadrillion units. A large volume of emission provides customers with certain conveniences when conducting small transactions.

Cryptocurrencies and ERC-20 tokens [6] issued in EVM-compatible networks are characterized by a large volume of emission, for example10^{26}. This is caused by current limitations of Solidity programming language when working with fractions [7]. Usually the *decimals* parameter is used which means the number of decimal places of the fractional part of the token. As a rule, this parameter is chosen equal to 18. In this case, one token contains 10^{18} parts.

The approach involves the use of auxiliary units. For example, users can specify the transfer amount in thousandths or billionths of the token. Ethereum uses Finney, Gwei and other units to represent fractional parts.

In the generalized case, the entire volume of emission can be represented through one token, and its parts can be used for calculations among customers. A similar replacement can be used for comparative analysis of various blockchain-based financial instruments.

The concept will also make it possible to implement some applied algorithms. At the same time, you can use the same approaches when conducting operations with cryptocurrencies, tokens and NFTs issued in a single instance. In certain cases, the approach can simplify network interaction and transaction processing. However, difficulties might arise within the implementation of the concept due to the need for additional emission. Operations with land plots require more complex algorithms for the emission of tokens and other approaches in determining the volume of the emission.

Current algorithms do not consider the specifics of properties. A special approach for land plots is used. It considers that each of the objects is presented as a non-fungible token [8]. There is no possibility of simultaneous ownership of the plot on several levels (by the owner and sub-owners). However, such decisions are necessary to ensure ownership of the plot by a private person while this territory will belong to the state. New algorithms will make it possible to implement more complex forms of real estate ownership. There is no possibility of spliting land plots into fragments of arbitrary area and configuration or combining them in current solutions.

Non-fungible tokens can be linked to cadastral numbers [9], which include information about them. It is placed in an external database. The cadastral number determines the boundaries of the land plot and allows you to get all the information about it. All data on the territory are placed in the register. However, the database is constantly changing. Additional entries are made to it during registration of new plots. The spliting or unification of territories will be accompanied by a change in cadastral numbers. Thus, storing data on properties in a separate database creates certain difficulties when synchronizing information in the land cadaster and blockchain.

One of the features of NFTs is their indivisibility. The owner can only sell the token as a whole, as it cannot be split into parts. The widespread distribution of NFTs has led to an increase in demand for them, as well as an increase in their value. The price of some tokens reaches millions of US dollars, which significantly reduces the liquidity of these assets, making them similar to precious stones, which also have a large value and cannot be divided into parts. However, the implementation of non-fungible tokens on smart contract platforms provides an opportunity to create more flexible mechanisms for issuiance and interaction with tokens.

The high prices on NFTs have led to the emergence of new tools that allow you to split tokens into parts and conduct transactions with them. The simplest solution that enables NFT splitting involves locking the token in a smart contract. At the same time, the contract issues ERC-20 tokens. When an NFT is locked, a certain number of fragments are created, for example 100. These assets can be sold for a corresponding part from the NFT's market value. The number of fragments is small, and it reduces the risk of losing one of them.

The technical implementation of fractional NFTs requires the use of additional algorithms. There are several projects that allow you to increase liquidity and split non-fungible tokens into parts, and each of the solutions has certain features.

The NFTX trading platform allows you to create repositories that hold multiple NFTs [10]. An asset lock allows contract to release a certain number of ERC-20 vTokens that represent parts of the repository. The algorithm allows the user to withdraw a random NFT from the repository by burning a single vToken.

NIFTEX and Fractional.Art platforms use more complex algorithms to split non-fungible tokens. After locking the NFT, the smart contract issues ERC-20 tokens. To trade these parts, the developers suggest creating cryptocurrency pairs on decentralized trading platforms, such as UNISWAP.

To regain ownership on a locked NFT, you must collect all the fragments and use a smart contract to return the NFT. This splitting method can lead to the fact that some fragments can be lost, burned or frozen (reserved) by their owners. If you do not know special algorithms, NFT may remain locked in a smart contract forever.

The Fractional.Art platform involves setting a certain price value for a non-fungible token. This value is determined by voting, in which only fragments' owners participate. Anyone can buy a locked NFT at an auction if they bid higher. However, the minimum value of a non-fungible token cannot be lower than the value determined during the voting. When redeeming, the fragments are burned, and their owners receive a corresponding share of the value of the non-fungible token.

The NIFTEX platform also provides the possibility of redemption of fragments. The algorithm allows any fragment owner to redeem NFT. The initiator of the transaction sets the redemption price, which requires him to block the relevant funds. If any of the other owners believe that the buyout price is low, they have the option to buy out the ownership share of the iniciator of the transaction at the specified value. At the same time, the NFT remains locked. Otherwise, the initiator receives a non-fungible token, and other co-owners get blocked funds.

Current ways of splitting NFT into fractions provide necessary functionality, but have certain limitations. Many algorithms provide only splitting into fixed parts. The technical implementation imposes certain obligations and restrictions on each of the fragment owners. In order to unlock an non-fungible token, additional social and market mechanisms are needed.

Current solutions do not consider the type of non-fungible token when splitting it into parts. If the NFT is a part of the virtual world or a flat image, the token fractions will not be associated with a specific area of that object. Prospective solutions should provide the ability to split a non-fungible token into arbitrary-sized pieces that are associated with specific parts of a digital object represented by an NFT. There is also a requirement to ensure asset ownership at multiple levels. Progressive algorithms should provide more complex forms of ownership on a virtual object, which will expand the scope of their practical application significantly.

The purpose and tasks of research

The purpose of this article is the research of specialized tools of EVM-compatible networks that implement the splitting of tokens into parts, tokenization of properties and establishing the features of the practical application of the proposed algorithms for solving applied problems.

The research objectives include substantiation of the need for one-dimensional and two-dimensional tokens to represent linear and flat objects and conduct operations with them.

The tasks include the research of algorithms that allow entering data on the parameters of land plots directly the blockchain and the specifics of their application. The work includes the development and analysis of algorithms that ensure the splitting of objects, in particular, virtual territories into parts of an arbitrary configuration and the unification of these parts.

Tasks include the analysis of algorithms that allow ownership of territory at different levels and control over transactions by the issuer establishing the features of the practical application of the proposed solutions.

1. The concept of one-dimensional tokens

Let's consider the geometric interpretation of a one-dimensional token, the emission volume of which is one unit. It can be represented as a segment of unit length in one-dimensional space.

One of the ends of the segment is selected as a reference point. According to this interpretation, the shares belonging to the owners will be represented by parts of the segment, which are determined by the coordinates of the beginning and end. For example, Alice owns 0.45; Bob - 0.15; Carol - 0.25; Dave - 0.05; and Ed – 0.1 token (Fig. 1). Unlike ERC-20 tokens, customers own plots defined by two coordinates (for example, Bob will own plot [0.45; 0.6]). At the same time, plots equal in size will not be identical and fungible.

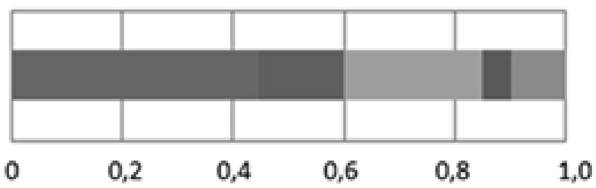

Fig. 1. A geometric interpretation of the one-dimensional token with a segment

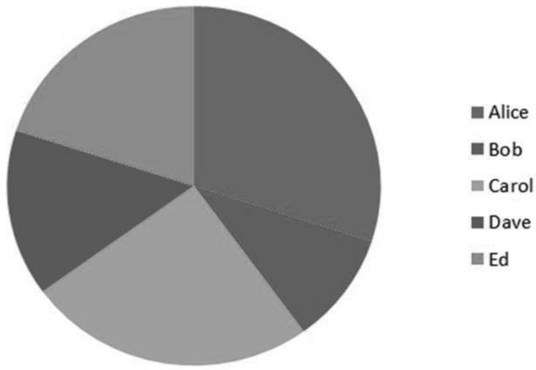

Fig. 2. Geometric interpretation of the one-dimensional token with a circle

A circle of unit length might be used for the geometric interpretation of the token instead of a line segment. It is divided into sectors of different sizes with corresponding angles (Fig. 2). It is necessary to choose a reference point on the circle, in relation to which the boundaries of the plots belonging to customers are determined. According to this interpretation, the units of measurement can correspond to radians or degrees.

The initial emission involves the transfer of token fragments to one or more customers. Fragmentation of the segment and token will be observed after performing a number of operations. After the redistribution of ownership shares, some customers may own several parts (Fig. 3). For example, Alice owns the parts [0; 0.3] and [0.65; 0.8], Bob – [0.3; 0.4] and [0.8; 1.0], and Carol – [0.4; 0.65].

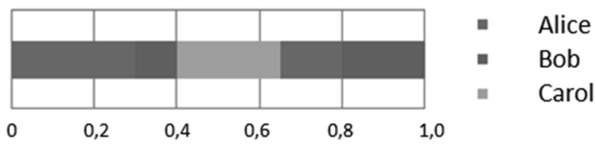

Fig. 3. Fragmentation of the one-dimensional token

The implementation of a one-dimensional token does not depend on a geometric interpretation (segment, circle or another curve). It is necessary to set its initial coordinate and its increment for each plot, which determine the share of ownership of the customer. In the case the customer has several plots, it is necessary to specify the parameters of each of them.

If only the share of ownership are to be determined, a one-dimentional token will function similarly to ERC-20 token. Each customer will own a certain part of the total emission volume. In this case, the order of location of plots, as well as their number, is not important. To find the property of the subject, it is more appropriate to use the initial coordinate of the plot and the increment of the coordinate. An alternative option is to specify the coordinates of the beginning and end of the segment directly.

Since a customer can own a large number of plots, several sources of assets can be used to perform transactions. A similar algorithm has been successfully implemented in Bitcoin [2] and other cryptocurrencies. The technical implementation of Bitcoin requires the use of one or more unspent transaction outputs (UTXO) to perform the transfer. At the same time, data on the sources of assets is included in the transaction. One-dimensional tokens can be implemented in EVM-compatible networks, and customers can send several plots in a single transaction (Fig. 4).

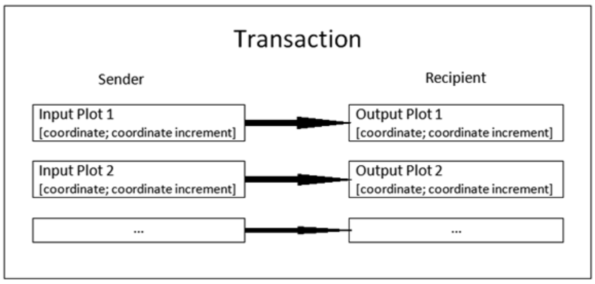

Fig. 4. Transfer of several fragments within one transaction

When transferring, the customer sends previously received plots (Input Plot 1, Input Plot 2, ...), accordingly, it is necessary to check the ownership rights to them. The transaction requires the address of the receiver and the parameters of the source and transferred plots (Output Plot 1, Output Plot 2, ...).

According to the technical implementation, sending a part of a one-dimensional token will be similar to the transfer of a ERC-20 virtual asset, but it will require two parameters that determine the beginning and end of the fragment.

2. Peculiarities of technical implementation and practical application of one-dimensional tokens

A one-dimensional token (1DT) does not conform to ERC-20 , so its implementation on an EVM-compatible network requires specialized algorithms and functions. It is recommended to choose a token emission equal to 10^{18}. If you use the value of the *decimals* parameter equal to 18, then the volume of emission will correspond to one token. In this case, the minimum plot size is limited. If we consider fragments of unit length, they can be placed sequentially from zero to the maximum value. In fact, each plot will be unique.

To implement the token in an EVM-compatible network, it is necessary to develop a smart contract and declare two arrays *Beginning_of_the_Fragment* and *Fragment_Length*, which are created with the *mapping* tool. The first array determines the coordinates of the starting point of the plot, and the second is used to specify its size (Fig. 5a).

```
mapping(address => uint[]) Beginning_of_the_Fragment;
mapping(address => uint[]) Fragment_Length;
```

a)

b)

Fig. 5. Implementation of a one-dimensional token in an EVM-compatible network

Arrays are created for each address with a non-zero balance. If the customer owns only one plot, the arrays contain one record each. In other cases, data structures include information about a large number of fragments. Elements of arrays can contain zeros that appear after complete transferring a plot to another customer.

Implementation of a one-dimensional token in an EVM-compatible network requires the use of specialized functions. The *GetSummaryBalance()* function (Fig. 6) allows you to find the total length of all fragments belonging to the customer. The algorithm adds the lengths of the fragments and finds the share of ownership corresponding to a certain address.

Algoritm 1. Finding of the total balance.

```
function Get Summary Balance
get Number of User Fragments
set SummaryBalance := 0,  i := 0
    while  i < Number of User Fragments do
        increase Summary Balance by Fragment Length[i]
        increase i by 1
    end loop
return Summary Balance
```

Fig. 6. Implementation of the function *Get-SummaryBalance()*

The GetPlotsParameters() function (Fig. 7) displays two arrays of data that show the parameters of the fragments belonging to the customer. The first array represents the coordinate of the beginning of the plot, and the second denotes its length. The function does not display null array elements. The result of GetPlotsParameters() can have the following form (Fig. 5b).

Algoritm 2. Finding the parameters of the plots belonging to the user

```
function Get Plots Parameters
get Number of User Fragments
set i := 0
    while  i < Number of User Fragments do
        if Fragment Length[i] != 0 then
            return Beginning of the fragment[i]
            return Fragment Length[i]
        increase i by 1
    end loop
```

Fig. 7. Implementation of the function *Get-PlotsParameters()*

To transfer a plot to another customer, you can use the function *Transfer (Sender, Recipient, Beginning of the Fragment, Fragment Length)* (Fig. 9). The procedure involves the creation of additional elements in the arrays of the recipient represented by the *Recipient* address. The resulting ownership share of the sender varies depending on which part of the fragment is being sent. At the same time, three transfer options are possible:

- The plot is transferred in full. Elements of the *Beginning of the Fragment and Fragment Length* arrays corresponding to the specified area are reset to zero.
- The sender transfer a part of the plot that is at the beginning or end of the source fragment. In this case, the algorithm change the values of array elements that determine the beginning of the fragment or its length.
- The sender transfer a fragment that divides the source plot into three parts. The first and third parts remain with the owner, and the second (middle) is transferred to the recipient. In this case, the parameters of the source fragment are changed on the sender's side. An additional element of the *Beginning of the Fragment* and *Fragment Length* arrays is also created. It includes the parameters of the third part.

The increase in the number of customers and transactions will lead to significant fragmentation of the one-dimensional token. In some cases, one customer will own several adjacent plots. If the fragments are placed in such a way that the end of one of them coincides with the beginning of the next, then it will be more appropriate to combine them. To perform this operation, the *Defragmentation*() function is used (Fig. 8). It allows you to find adjacent areas and combine them into one. Since the operation requires gas consumption and also changes the parameters of the arrays that determine the ownership of the customer, it can be carried out only by the owner.

Algoritm 4. Merging of the fragments

```
function Defragmentation
get Number of User Fragments
set i := 0
  while  i < Number of User Fragments do
    set j := 0
      while  j < Number of User Fragments do
        if (Beginning of the Fragment [i]+ Fragment Length[i] ) ==
        ==Fragment Length[j]  && Fragment Length[j] != 0 then
          Fragment Length[i] += Fragment Length[j]
        increase j  by 1
      end loop
    increase i  by 1
  end loop
```

Fig. 8. Implementation of the function *Defragmentation()*

Algoritm 3. Transfer of the fragment

```
function Transfer (Sender, Recipient, Beginning of the Fragment, Fragment Length)
set r := Number of Recipient Fragment
set s := Number of Sender Fragments
create Beginning of the Fragment [Recipient] [r] = Beginning of the Fragment
create Fragment Length [Recipient] [r] = Fragment Length
set i := 0
 while i < s do
     if (Beginning of the Fragment [Sender] [i] == Beginning of the Fragment then
        Beginning of the Fragment [Sender] [i] =Beginning of the Fragment + Fragment Length
     else
        Fragment Length[Sender] [i] = Beginning of the Fragment –
        –Beginning of the Fragment [Sender] [i]
           if (Beginning of the Fragment [Sender] [i] +Fragment Length[Sender] [i] !=
           != Beginning of the Fragment+Fragment Length) then
              create Beginning of the Fragment [Sender] [s]=Beginning of the Fragment +
              + Fragment Length
              create Fragment Length [Sender] [s]=Beginning of the Fragment [Sender] [i]+
              +Fragment Length[Sender] [i] - Beginning of the Fragment - Fragment Length
     increase i by 1
 end loop
```

Fig. 9. Implementation of the function *Transfer()*

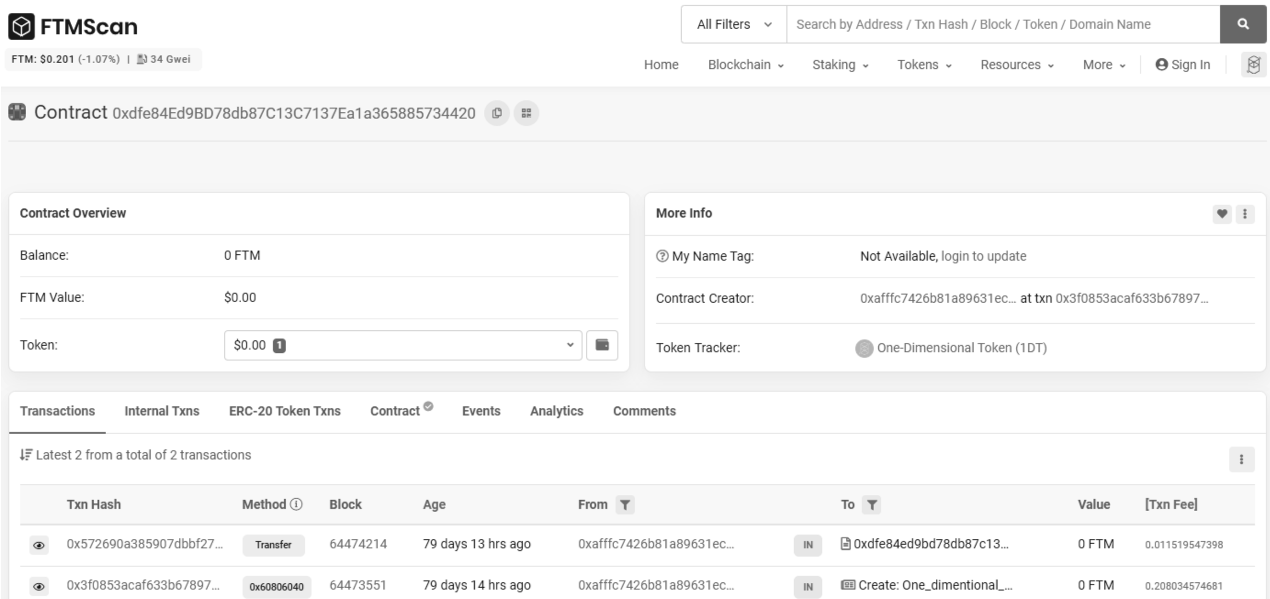

Fig. 10. One-Dimensional Token (1DT) smart contract

The One-Dimensional Token (1DT) smart contract corresponding to the proposed algorithms is deployed on the Fantom network [11] at the address: 0xdfe84Ed9BD78db87C13C7137Ea1a365885734420 (Fig. 10).

The functions of reading the smart contract allow you to get general information about the token, determine the share of ownership and set the parameters of the plots owned by the customer. Record functions enable owners to transact and unify adjacent plots. There are also tools that provide the possibility of transferring plots using a third-party smart contract.

The plot transfer is implemented through the *transfer()* function and involves entering three parameters: the address of the recipient, the initial coordinate of the plot and its size. When transferring a plot, it is recommended to set the gas limit to more than 274000 units. Since the 1DT does not comply with the ERC-20 standard, it is possible to find out the current balance of the customer only using the corresponding reading function.

One-dimensional tokens have a number of differences compared to ERC-20 virtual assets and NFTs. These include the possibility of splitting and unifying parts of the token. At the same time, each of the fragments is unique and is defined by two parameters. The one-dimensional token allows us to demonstrate the functionality of the solution and its capabilities.

3. Algorithms for the implementation of two-dimensional tokens and their application for solving applied problems

A two-dimensional token is a virtual asset defined in a two-dimensional coordinate system. Graphical interpretation of an object – a section of a spatial surface. In some cases, it is a part of a plane, sphere or cylinder.

A token can represent an entire closed surface, allowing it to be used for a sphere. In other cases, the virtual asset corresponds to only a part of the surface. For example, a token can represent an area on a plane, bounded by a polygon. When speaking about practical application of the token, it is most appropriate to use a rectangle. This plot is splited into customer-owned fragments.

Two-dimensional tokens can be used to represent areas of a flat or spherical surface, assuming the use of identical algorithms. An arbitrary fragment belonging to the customer can be represented by a closed loop. The most optimal is the use of a polygon, the parameters of which are set by the coordinates of the vertices and the order of their sequence.

Let's consider the implementation of a two-dimensional token on a spherical surface. It is advisable to determine plot parameters in the spherical coordinate system. In this case, the position of a point in space is determined by two angles and a radius. When the radius is constant, it is enough to specify two angular coordinates.

To determine the location of the object on the surface of the Earth, we use a geographic coordinate system, and the parameters of the point are determined by latitude and longitude. For example, object coordinates can be specified in the following form: 50.449214; 30.357773. The number of characters after the separator determines the accuracy of object positioning.

A territory of arbitrary shape on a spherical surface can be represented in the form of a polygon [12]. The order of passing the vertices of the polygon determines its configuration (Fig. 11).

Fig. 11. Representation of a land plot in the form of a polygon

In certain cases, it is appropriate to represent a polygon using the coordinate of one vertex and increments that determine the position of others. The algorithm allows to reduce the amount of data recorded in the blockchain. The parameters of a plot of arbitrary shape (*Area*) can be set by an array (Fig. 12). One of the vertices is selected as the initial coordinate (*Initial coordinate*). The coordinates of other points are set in the form of increments in relation to the initial coordinate (*coordinate increment 1, coordinate increment 2, ... coordinate increment n*). The

Area

	Latitude	Longitude
Initial coordinate	**50,449214**	**30,357773**
coordinate increment 1	0,004703	0,113006
coordinate increment 2	-0,123239	0,04289
coordinate increment 3	-0,120425	-0,131187
coordinate increment 4	-0,075794	-0,213764
coordinate increment 5	-0,145759	-0,375646
coordinate increment 6	-0,090647	-0,464117
coordinate increment 7	-0,048376	-0,380012
coordinate increment 8	0,043858	-0,466657
coordinate increment 9	0,089249	-0,234732
coordinate increment 10	0,145127	-0,288953
coordinate increment 11	0,163612	-0,175346

Fig. 12. Representation of plot parameters using an array

order of the vertices when passing a closed circuit corresponds to clockwise movement.

The algorithm allows you to perform operations with fragments of an arbitrary configuration, to split and unify them. A special configuration area may contain a void if its interior belongs to another customer. In this case, the territory can be represented by two polygons that do not contain cavities, but have common edges and vertices.

Two-dimensional tokens can represent areas of any flat or spherical surfaces. They can be used in virtual worlds containing territories owned by customers. The developed algorithms can be used to represent areas of the Earth or another celestial body.

Two-dimensional tokens can be used to distribute NFTs or other virtual assets among multiple owners. At the same time, each customer can own a flat fragment of any configuration.

If it is necessary to have sub-owners of the first and second levels, then the process of emission and transfer of the land plot will be different. The first level sub-owner must issue a token. In order to carry out operations with a land plot, a second-level sub-owner must initiate a transaction. However, to implement it confirmations from sub-owners of the first and second levels are required. When using such an algorithm, all transactions with virtual assets will be controlled by all their owners.

When transferring, information about the parties will not be disclosed, since only their addresses are placed in the distributed database. Blockchain technologies will provide protection against unauthorized access, eliminate the possibility of canceling a confirmed transaction, and also prevent double spending operations [14].

When creating and conducting operations with large territories that include a significant number of vertices of the bounding polygon, a significant amount of information must be entered into the blockchain. We can reduce the amount of data by using a base coordinate and coordinate increments for the remaining vertices. It is advisable to use abbreviated coordinates within operations with small areas.

The use of two-dimensional tokens for the tokenization of land plots will require interaction with databases containing cadastral numbers and other information about properties. It is more appropriate to place the specified information in decentralized databases, e.g. by using IPFS.

Conclusions

One-dimensional tokens are a powerful tool for testing, creating arbitrary-sized plots and conducting transactions. The developed smart contract allows customers to own fragments of any one-dimensional object and perform operations with them.

2D tokens can be used to fragment maps, flat images, and other 2D objects. They have a number of advantages over NFTs when used to represent land plots and virtual territories. Two-dimensional tokens allow you to create fragments of arbitrary configuration, split and unify them.

The developed algorithms can be used to represent the territories of virtual worlds. Current projects use fixed-sized plots, while two-dimensional tokens provide greater flexibility by allowing fragments of arbitrary configuration to be operated on.

The technology can be applied to spherical surfaces, including the Earth surface, but it requires setting a reference point. The use of two-dimensional tokens will require interaction with external databases containing data on cadastral numbers. At the same time, the algorithm must be compatible with blockchain-oracles that provide access to external state databases that personalize the owners of tokenized land plots.

Two-dimensional tokens can provide ownership of a surface area at different levels. A certain fragment can have one owner who is the issuer and controls the operations with the virtual asset and a sub-owner who conducts transactions with the plot or its parts directly. Such an algorithm provides an opportunity to implement a model of ownership that is characteristic of the land market of an individual country.

References

1. Bauer, D.P. (2022). Ethereum Name Service. In: Getting Started with Ethereum. Apress, Berkeley, CA. doi: 10.1007/978-1-4842-8045-4_9.

2. Lisdorf, A. (2023). Bitcoin. In: Still Searching for Satoshi. Apress, Berkeley, CA. doi: 10.1007/978-1-4842-9639-4_4.
3. Arslanian, H. (2022). Ethereum. In: The Book of Crypto. Palgrave Macmillan, Cham. doi: 10.1007/978-3-030-97951-5_3.
4. Vitalik Buterin, Eric Conner, Rick Dudley, Matthew Slipper, Ian Norden, Abdelhamid Bakhta, "EIP-1559: Fee market change for ETH 1.0 chain," Ethereum Improvement Proposals, no. 1559. (2019). [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-1559.
5. Jethmalani, Mehak, & Yaskil, Rivka, Aspler. (2023). Non-fungible token (NFT) purchase and transfer system. US Patent US20230298001, filed March 21, 2022, and issued September 21, 2023.
6. Fabian Vogelsteller, Vitalik Buterin, "ERC-20: Token Standard," Ethereum Improvement Proposals, no. 20. (2015). [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-20.
7. Ritesh, Modi. Solidity Programming Essentials. (2018). Birmingham, UK: Packt Publishing, 2018. ISBN 9781788831383.
8. Shang, Qiuyun & Price, Allison. (2019). A Blockchain-Based Land Titling Project in the Republic of Georgia: Rebuilding Public Trust and Lessons for Future Pilot Projects. Innovations: Technology, Governance, Globalization. 12. 72-78. doi: 10.1162/inov_a_00276.
9. Dale, Peter, & John McLaughlin. (2020). Land Administration (Oxford, 2000; online edn, Oxford Academic, 12 Nov. 2020). doi: 10.1093/oso/9780198233909.001.0001.
10. Introducing Our Decentralized NFT Marketplace. https://medium.com/nftx/introducing-our-decentralized-nft-marketplace-1c1c4d394724
11. Fantom Foundation: https://fantom.foundation/
12. Cadle, Farris W. (1991). Georgia Land Surveying History and Law. University of Georgia Press. ISBN 978-0-8203-1257-6.
13. Bellare, Mihir, & Neven, Gregory. (2006). Identity-Based Multi-signatures from RSA. Topics in Cryptology – CT-RSA 2007. pp. 145–162.
14. Chohan, Usman W. (2021). The Double Spending Problem and Cryptocurrencies. doi: 10.2139/ssrn.3090174.

Received: 01.12.2023

Про авторів:

Струтинський Сергій Васильович,
доктор технічних наук, професор кафедри прикладної гідроаеромеханіки і механотроніки Національного технічного університету України «КПІ імені Ігоря Сікорського».
Кількість наукових публікацій в українських виданнях – понад 200.
Кількість наукових публікацій в зарубіжних виданнях – понад 10.
Індекс Хірша – 5. https://orcid.org/0000-0001-9739-0399

Яланецький Валерій Анатолійович,
старший викладач кафедри інформаційних систем та технологій Національного технічного університету України «КПІ імені Ігоря Сікорського».
Кількість наукових публікацій в українських виданнях – понад 30.
https://orcid.org/0000-0001-6163-0258

Місце роботи авторів:

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,
проспект Берестейський 37,
Тел.: (044) 204-82-54,
E-mail: strutynskyi@gmail.com,
v.yalanetskyi@gmail.com