DDK Asset Issue
www.ddkoin.com

Be The Future By Our Innovation

DDK asset issuing (interchain) development

In the world of cryptocurrencies there are no easy mechanisms for managing the funds in various life circumstances or according to your business needs. DDK platform is dedicated to solving this issue.

For example, currently, there are over 15 millions of Bitcoin wallets from which more than a quarter of all existing bitcoins were lost forever. Running customer's own side chains with their own blockchain and cryptocurrencies is not a solution.

To solve this issue, DDK decided to use smart contracts: making crypto-assets quick, safe, and convenient.

Smart Contracts are the special computer software on the blockchain, which will be executed by a network of computers. They are the key to unlock the world, where any computer-oriented task can be performed completely autonomously and correctly, without fear of external manipulation or imitation.

Using smart contract technology, our customers' financial tasks become available to all who can use it in a fully automated, independent way, which records these tasks directly to the blockchain.

Although smart contracts are unambiguous in their meaning, only those who are familiar with the programming language can understand them, so, it makes it difficult to reach an agreement with any party that does not have technical knowledge.

This technology excludes the possibility of closing banks or financial institutions and changing, deleting transactions, and also ensures that each payment will be made exactly as indicated, without any risk of misuse or malicious interference.

Companies strive to use smart contracts in their work, correcting their various faults. Without easy-to-use, proven and reliable methods to

effectively create accurate smart contracts, their widespread adoption will never happen. Simply put, smart contracts will allow you to use advanced locking options for the widest audience.

Logically, the DDK generating platform may be represented by the following parts:

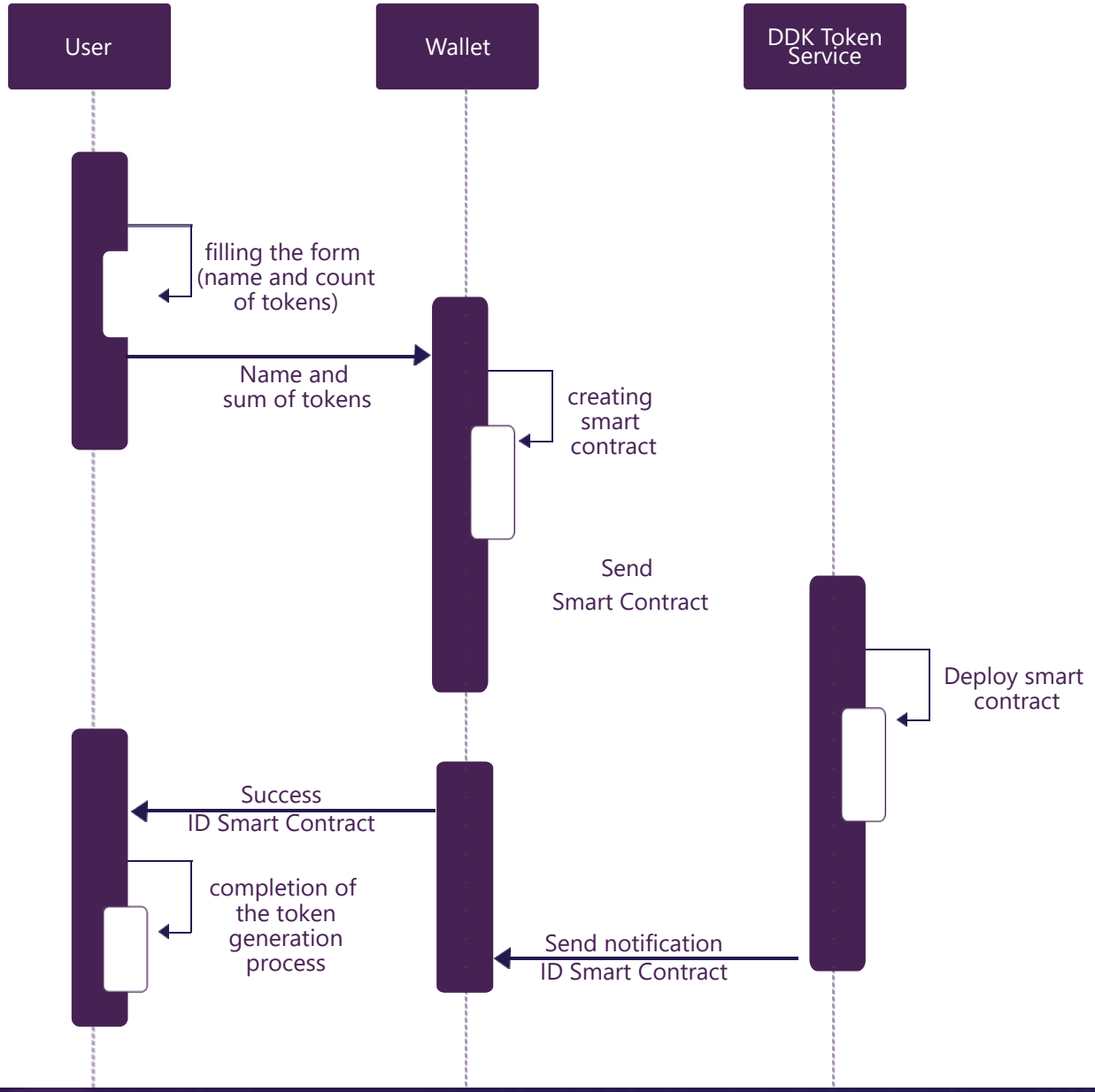
Front-end: start of token generation process and filling the forms (names and count of tokens)

Back-end: creating a smart contract, deploy smart contract ID, and name and sum of tokens to Node DDK, Deploy Smart contract onto the Node DDK, sending ID Smart contract notification to Back-end, and completion of the token generation process

DDK platform is a complete solution from the viewpoint of the contract life cycle, starting from contract creation till its completion, either due to its execution or expiry.

Also this technology excludes Web Wallet allowing customers to store and create their own coin. They can create, issue, send, and receive these coins across DDK network. Users can also use Web Wallet as a normal wallet to store, send, and receive cryptocurrencies.

Flow of token generation



Process Overview

1. Fill the form to generate a token.

User (the token holder) fills the page with the parameters of the token generation.
Selects a token, enters the amount (number) of tokens.
To Easy Coin Creator transfers the token ID, the amount (number) of tokens.

2. Data transfer for the token generation.

User transmits data to Easy Coin Creator to generate a token: Token ID, amount (number) of tokens.

3. Creating a Smart Contract.

Easy Coin Creator generates a Smart Contract, which records data about generating of token.This will be used for creating smart contract.
For this will used tool for Smart Contract creating.
Data will be transferred to the smart contracts: For this will be transfered data for smart contract : the ID of the token, the amount (number) of tokens, the ID (Address) of the wallet to which the tokens will be credited.

4. Deploy a Smart Contract.

The DDK Node confirms the operation of generating a token and fixes the parameters of the Smart Contract (token ID, amount (number) of tokens, ID (Address) of the wallet to which tokens are credited).

5. Confirmation of token generation.

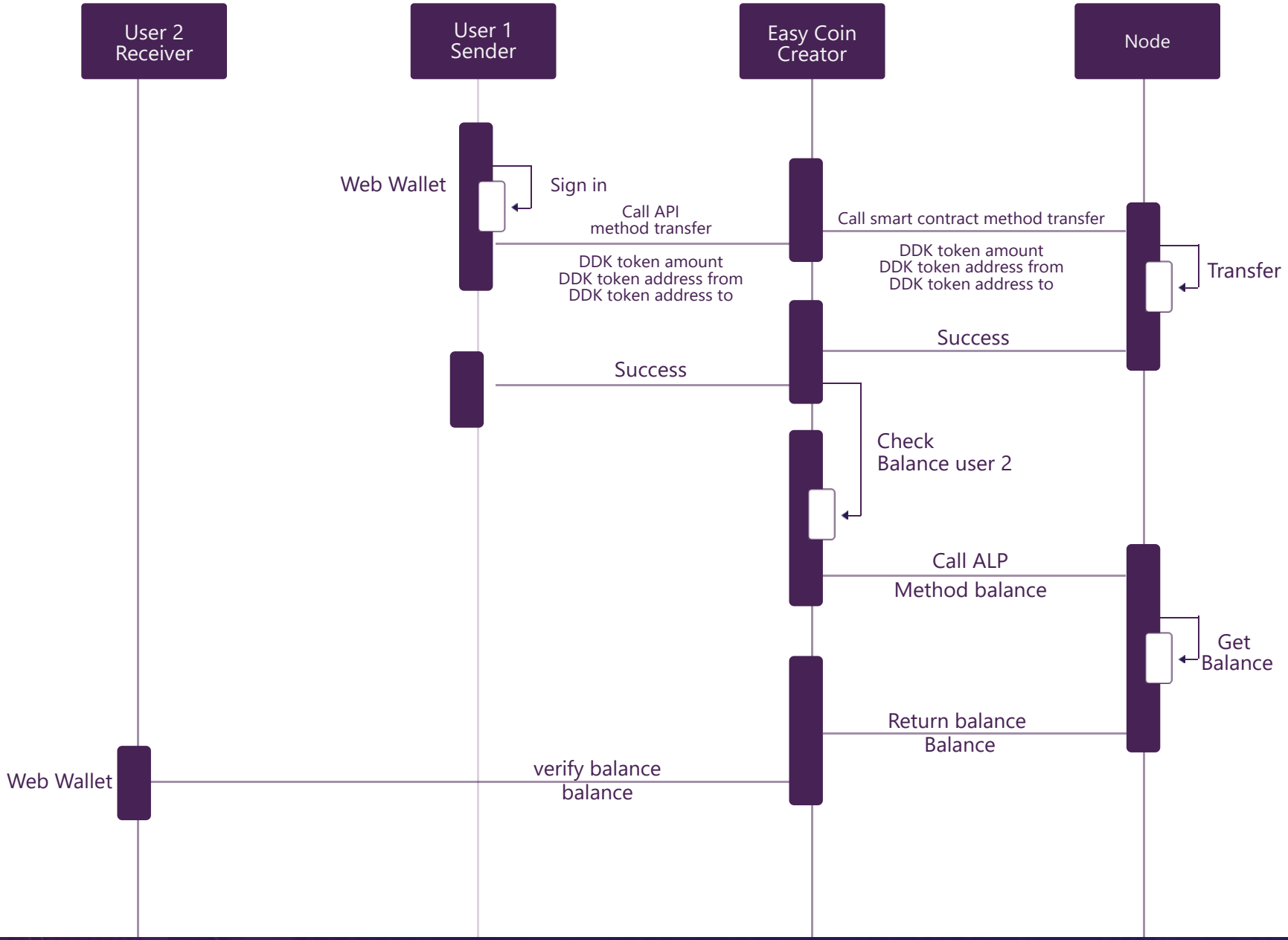
DDK Node sends a confirmation about successful transaction to Easy Coin Creator.
Data of the Smart Contract is fixed. The Smart Contracts ID is transmitted.

6. Confirmation of the token generation.

Easy Coin Creator sends a notification to the holder of the token about the successful operation. Token ID, the amount (number) of tokens, Smart Contract ID are transmitted.

7. Token generation process on the DDK platform is Complete .

Flow of transfer transaction



Process Overview

1. Preparing information for transferring tokens.

The token holder (source) logins into the wallet,
Selects a token,
Enters the amount that they want transfer,
Enter target wallet for transfer.

2. Transfer operation.

The token holder (source) transfer tokens to each other's wallet (target).
The API used (the method of transferring the money on the DDK platform).
Transmitted data about: token ID, transfer amount, Token ID (source), Token ID (target).

3. Smart Contract Creating.

Back-end of the System, generates a Smart Contract and traces data about operation of transferring tokens from source wallet to target wallet.
Smart contract creating tool and the transferring method is used to. transmit data about Smart Contract (token ID, amount of transfer, wallet (source) ID, wallet (target) ID).

4. Transfer funds.

DDK Node confirms the tokens transfer operation and traces the parameters of the Smart Contract (token ID, transfer amount, wallet ID (source), wallet ID (target)).

5. Tokens transfer Confirmation.

DDK Node sends a confirmation to the back-end of the successful transaction. Smart Contract data is traced. Transmitted the wallet ID (source), the wallet ID (target).

6. Tokens transfer Confirmation.

Back-end sends a notification to the token holder (source) about successful transfer operation.Transmitted token ID, the transfer amount, wallet ID (source), wallet ID (target).

7. Preparing a request to verify the balance of target wallet.

Back-end from smart contract parameters (token ID, transfer amount, wallet ID (target) prepares a request for verification of the fact of crediting tokens to the target wallet.

8. Checking the balance of the target wallet.

Back-End sends to DDK Node request to verify the fact of crediting of tokens to the target wallet.

API is used for obtaining wallet balance on the DDK platform. The ID of the Transmitted data about token ID, the transfer amount, target wallet ID.

9. Confirmation of the target wallet balance.

The DDK Node sends the back-end result of the request to verify the crediting of tokens.

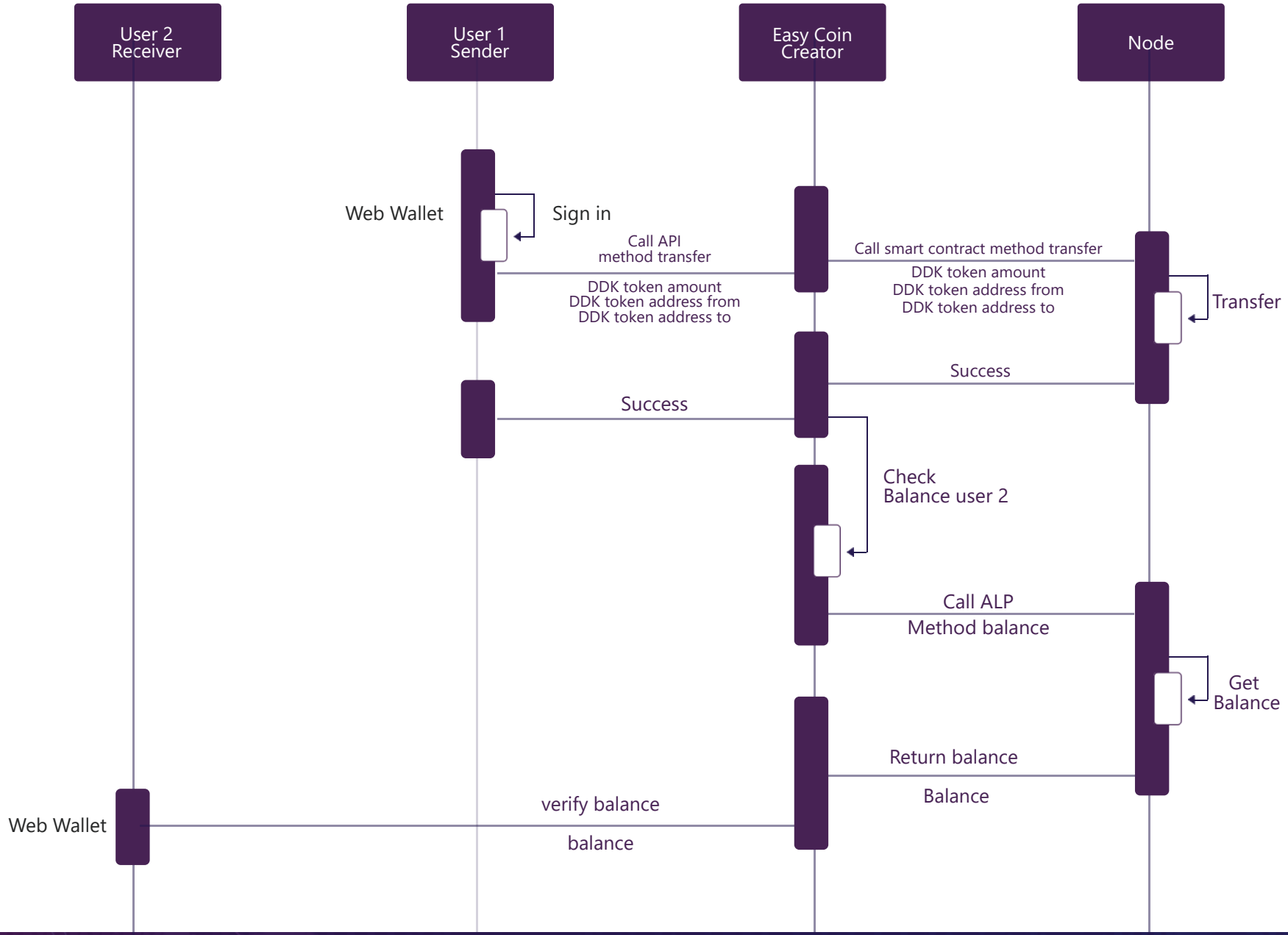
Transmitted token ID, target wallet ID to which the balance (remnant) of the target wallet.

10. Sending confirmation of the executed operation for transfer of tokens.

Back-End generates and sends to the (target) token holder, a confirmation of the successful completion of the transfer operation and verification of the balance of the target wallet.

11. Completion of the transfer of tokens from source wallet to target on the DDK platform.

Flow of token migration



Process Overview

1. Choosing the type of smart contract.

Before migration process you need to choose (old) smart contract, with exact parameters.

Token migration process is starting and DDK Token Service is transmitted to (old) smart contract Type ID and (old) smart contract ID.

2. Preparing data to create a new smart contract.

DDK Token Service requests a list of (old) tokens holders that must be migrated, from old smart contract. The list of token holders is nec essary for the correct distribution of new tokens by wallet in the new smart contract.

In DDK Node transmits a request to provide a list of holders of old tokens of old smart contract and old smart contract ID.

3. Data transfer for creating a new smart contract.

DDK Node prepare a list of token holders of the old smart contract, and transmitted it to DDK Token Service for the creation of a new smart contract.

Transmitted a list of old token holders from old smart contract to new smart contract ID.

4. Creating a new smart contract.

DDK Token Service creates a new smart contract.

Parameters of the new smart contract, which is also transferred from the old smart contract list of token holders.

The list of holders of the tokens to be sent to the Node DDK, which must be migrated, according to the valid smart contract, the ID (ad dress) of the new smart contract. This action is the initialization of the Deploy operation of the new smart contract.

5. Deploy a new smart contract.

For new tokens, it is necessary to create a new smart contract with the following characteristics: list of new token holders, the amount (num ber) of new tokens.

DDK Node started the procedure for deploying a new smart contract. The parameters of the new smart contract (token ID, list of token holders for the new smart contract, the ID of the new smart contract, the amount (number) of new tokens) are fixed.

6. Start the migration procedure.

DDK Node starting the process of token migration from old smart contract in a new smart contract (transfers the new smart contract to the migration mode). Being used the (old) token ID, the new token ID, the old smart contract ID (address), the new smart contract ID.

7. Send notification of the beginning of migration.

DDK Node sends to the old token holders (holders from list of holders of tokens under the current smart contract), from which the migration occurs, notification of the beginning migration and the need to implement it.

The ID of the transmitted token, the ID of the new token to which the migration occurs, the ID (address) the current smart contract and the ID (address) of the new smart contract are transmitted.

8. Voting of token holders

To start the token migration procedure, the token holders must perform the action in their electronic wallet.

It is inadmissible to anyone, even the owner of the smart contract, to "burn" the tokens of other token holders. This kind of operations on their tokens can only be carried out by their holder.

Therefore, the token holder of the current smart contract must also migrate the tokens. For example, to migrate to a token holder, it is sufficient to transfer any amount (number) of tokens to the ID (address) of the new smart contract in the migration mode. The token holders confirm the fact of the migration of their tokens using the voting method and fix their decision.

The ID of the electronic purse, the ID of the token from which the migration occurs, the ID of the token to which the migration occurs, the amount (number) of tokens, the ID (address) of the valid smart contract, the ID (address) of the new smart contract, the voice (confirmation) of the holder are transmitted.

9. Fixing the decision of the holders of the tokens.

Based on the information received (see clause 8), the Node DDK captures the confirmation holders of the token for conducting a token migration operation and launching directly functionality of token migration.

10. "Burning" of tokens, creating new tokens.

Node DDK using the list of holders of the token from which the migration occurs, and confirmation of their holders "burns" the tokens from which migration occurs and creates new tokens on the new smart contract.

The amount of new tokens created should correspond to the amount of "burned" tokens in the holder's section (ID of Web wallets).

11. Preparation of data to verify the completeness of the migration.

To verify the correctness of generation of new tokens and the correspondence of their amount (number) of old tokens,

To verify the compliance of the distribution of new tokens by electronic purses of tokens holders Node DDK uses the parameters: ID (address) the current smart contract, the ID (address) of the new Smart Contract, the amount (number) of tokens under the smart contract,

the amount (number) of tokens under the new smart contract.

12. Checking the correctness and completeness of the token migration.

Node DDK checks the correctness of the distribution of new tokens by electronic purses of holders, equality of the total amount (number) of new tokens to the sum (amount) tokens from which the migration was made (the amount (number) of tokens and new tokens is not should change). The data which is used: the ID (address) of the current smart contract, ID (address) of the new smart contract, amount (number) of tokens for the Smart Contract, amount (number) of tokens under the new Smart Contract, ID of electronic purses.

13. Fixing the parameters of the migration operation

The Node DDK sends a confirmation to DDK Token Service that the operation has been completed migration of the token and transmits the ID (address) of the new smart contract, the list of the holders of the token for new smart contract.

14. Completion of the token migration operation

In the DDK Token Service, a confirmation from the Node DDK is fixed and a characteristic is set completion of the token migration opera tion. The data is used: the ID (address) of the new smart contract, list of holders of the token under the new smart contract. All holders of the new token and smart contract Owner will be notified.

15. Sending a confirmation to the holder of a new token of the performed migration operation token

DDK Token Service forms and sends a new token to the holder, a notification with confirmation of successful execution of the operation of the token migration with indication of the amount (number) of new tokens and the ID (address) of the new smart contract. The data is used: the ID (address) of the new smart contract, the list of holders of a new security key for the smart contract, the amount of new tokens in context of electronic wallets

16. Sending confirmation to Owner about completed token migration operation

DDK Token Service generates and sends to smart contract Owner a response with confirmation about successful execution of the opera tion of the token migration with indication of the amount (amount) of new tokens and the ID (address) of the new smart contract. The data is used: the ID (address) of the new smart contract and list of holders of the token under the new smart contract.

17. Complete the process of performing the operation of token migration on the DDK platform


KOIN
www.ddkoin.com