COTI's
MultiDAG 2.0 Protocol
Light Paper

By:
Dr. Nir Haloani, Eli Hallufgil, Guy Mesika, Alexander Panasenko, Yuval Altun, Tomer Armarnik, Dima Rudenko

Table of Content

Abstract

The purpose of this document is to describe the transition of COTI's Trustchain from the MultiDAG 1.0 protocol (Whitepaper) to the new enhanced MultiDAG 2.0 protocol. In addition, this document will explore the MultiDAG 2.0 enhanced capabilities and provide a high-level technical overview of the new protocol. While MultiDAG 1.0 was a significant milestone in the development of the Trustchain, and acted as a technological proof of concept and infrastructure backbone that enabled the COTI development team to test technical hypotheses and which provided a strong foundation for building a Scalable and Secure protocol, it lacked the fundamental client-side components required to create utilities that will serve COTI's business objectives. The motivation behind MultiDAG 2.0 was to enhance MultiDAG 1.0 capabilities by introducing scalability, performance, and security improvements.

Furthermore, MultiDAG 2.0 will allow a bi-directional wrapped asset swapping between the Trustchain and supported blockchain networks as a result of the COTI Bridge 2.0 upgrades introduced in order to facilitate MultiDAG 2.0.

Glossary

Term	Meaning
DAG	Directed acyclic graph.
MultiDAG	Multi token-directed acyclic graph.
Full Node	A specialized server run by a user for common network tasks.
Financial Server Node	A specialized server operated by COTI for common network financial calculation tasks.
Trust Score	A user metric that is used for effective transaction processing and risk mitigation.
DSP	Double Spend Prevention Node.
Transaction Validation	The process of checking the transaction before attachment to the DAG Cluster.

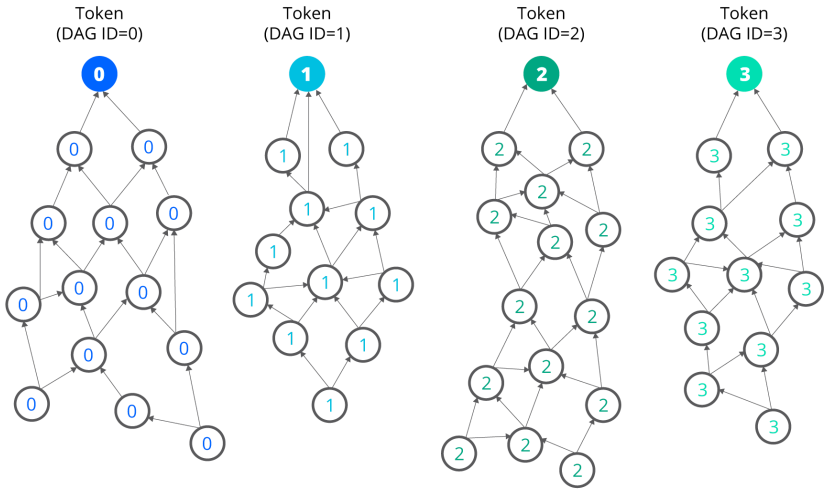
MultiDAG 2.0 Protocol

MultiDAG 2.0 Protocol builds on top of the existing protocol architecture and will enable new types of transactions. These new transaction types will permit Trustchain clients to autonomously Generate new tokens and control token circulation by Minting and Burning them.

After MultiDAG 2.0 goes through its Testnet and Mainnet trial periods, MultiDAG 2.0 will be officially initiated by creating a hard fork transaction that will be attached to DAG and confirmed by network consensus. With this change, also comes a new token standard: CMD (COTI MultiDAG).
From that point, new tokens can be generated, transferred, and burned on the Trustchain as long as the utilized Full Node supports MultiDAG 2.0 protocol.

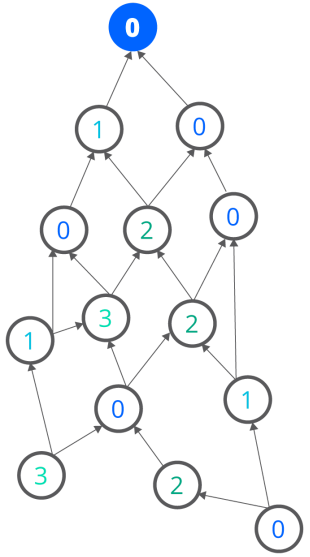
The main difference between the legacy protocol and the updated one is its (cluster: token), MultiDAG 2.0 uses a single cluster for multiple tokens (1:n); this new approach increases confirmation times for low utilized tokens and reduces the need for starvation transactions during low traffic network conditions.

MultiDAG 1.0



In MultiDAG 1.0, each transaction helps to reach consensus for other transactions of the same token type. For example, a transaction in DAG ID=1, will only be able to attach to other transactions in DAG ID=1 to reach consensus. It can not attach to DAG ID=2 or DAG ID=3 transactions. As a result, it takes more time to reach consensus.

MultiDAG 2.0



In MultiDAG 2.0, transactions help each other to reach consensus regardless of the token type. For example, a transaction from Token ID=1 can attach to a transaction of Token ID=2 and Token ID=3 to help reach consensus. The more transactions on the DAG, the faster they will be approved, regardless of the token type.

Token Generation

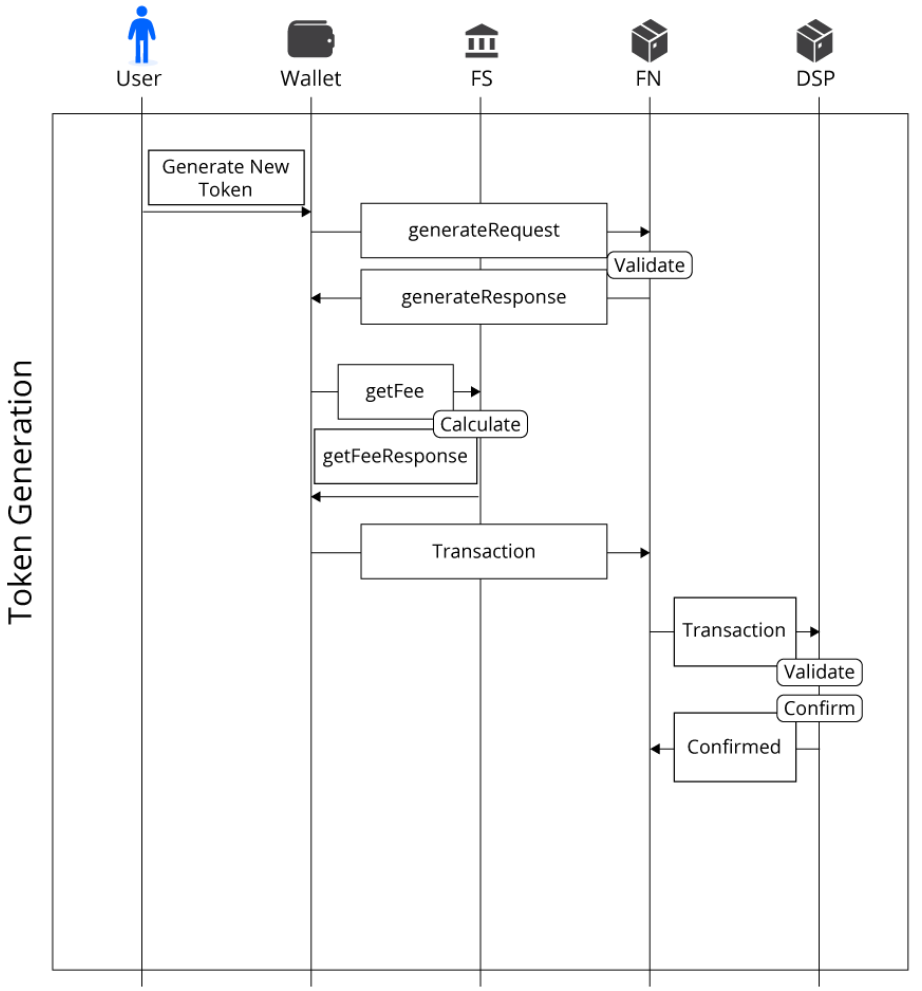
In order to issue a new Token, the client application will input the following token parameters **Name**, **Symbol**, **Total Supply**, and **Scale**. After the client submits all required parameters, the system will validate the input; if the request is valid, the system will provide a fee estimation.

If the client decides to proceed with the token generation process, a token generation transaction will be created. The network will prevent duplicate token generation by performing validation on both Full Node and DSP Node levels.

Once the token generation process is complete, the token is, in effect, issued by the issuing party and can now be minted and transferred freely on the Trustchain.

Example:

Name: *Universal Payment Token*
Symbol: *UNI*
Max Supply: *1,000,000*
Scale: *8 (0.00000001)*



Token Minting

Once a token is generated and reaches consensus, the token issuer may wish to increase token circulation by performing a mint action.

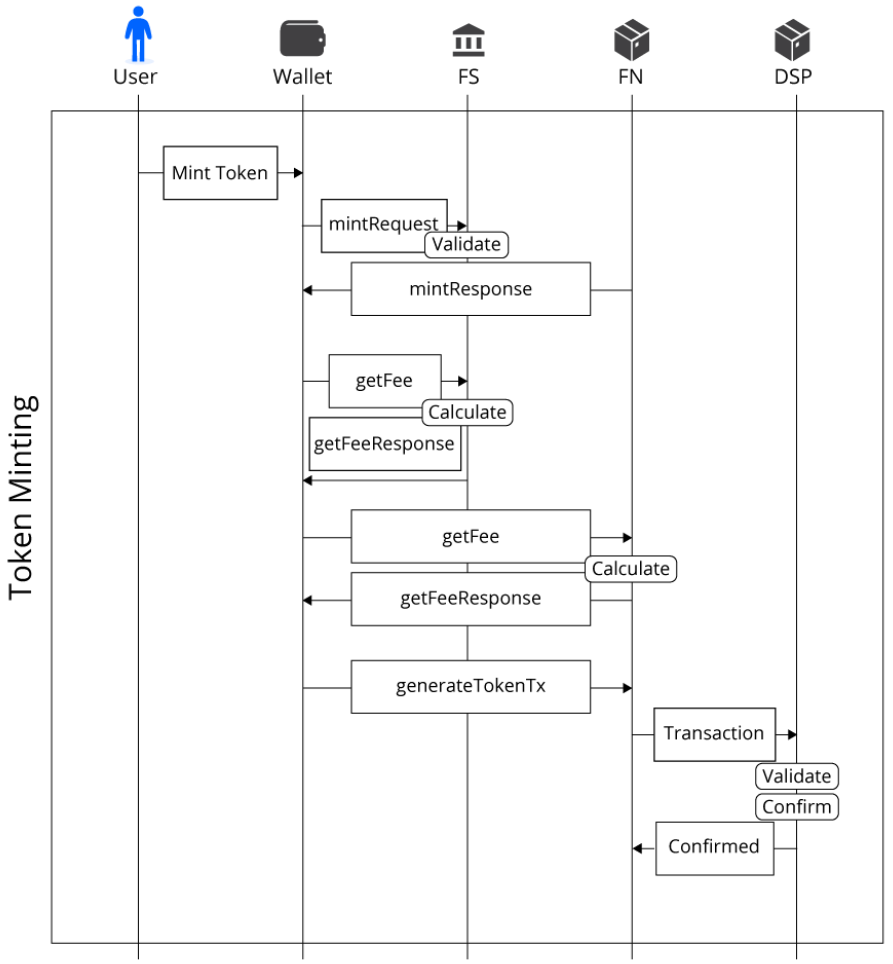
The mint action is completed by submitting a mint request to the Full Node and obtaining fee estimates from the latter and a Financial Server Node fee.

The Full Node will accept token minting requests only from the token issuer before the mint process initiation Full node will validate mint request parameters and prevent minting token amounts that would breach the token total supply cap defined during the token generation process the network will prevent double-spend transactions.

When the minting transaction reaches network consensus, minted tokens will be transferred to the receiving address specified during the token mint request.

Example 1: *a token was generated with Total Supply set to 1M tokens, the client submitted a request to mint 500K tokens, and the mint request was successful. The client submitted a mint request for 600K, mint request will be rejected as the requested mint amount (600K) > Mintable Amount (500K).*

Example 2: *a token was generated with Total Supply set to 1M tokens. The client submits two parallel requests to mint 1M tokens to two separate Full Nodes, resulting in the first transaction to reach consensus will be considered successful and the second request will be considered rejected.*



Token Transfer

Once a new token is minted, it will be transferred to the receiving address stated during the mint request. Once the minted token is received, it can be transferred to any Trustchain address, similar to transferring $COTI. As tokens that were generated on the Trustchian do not always have value (price), transfer fees will be charged from the client's $COTI balance.

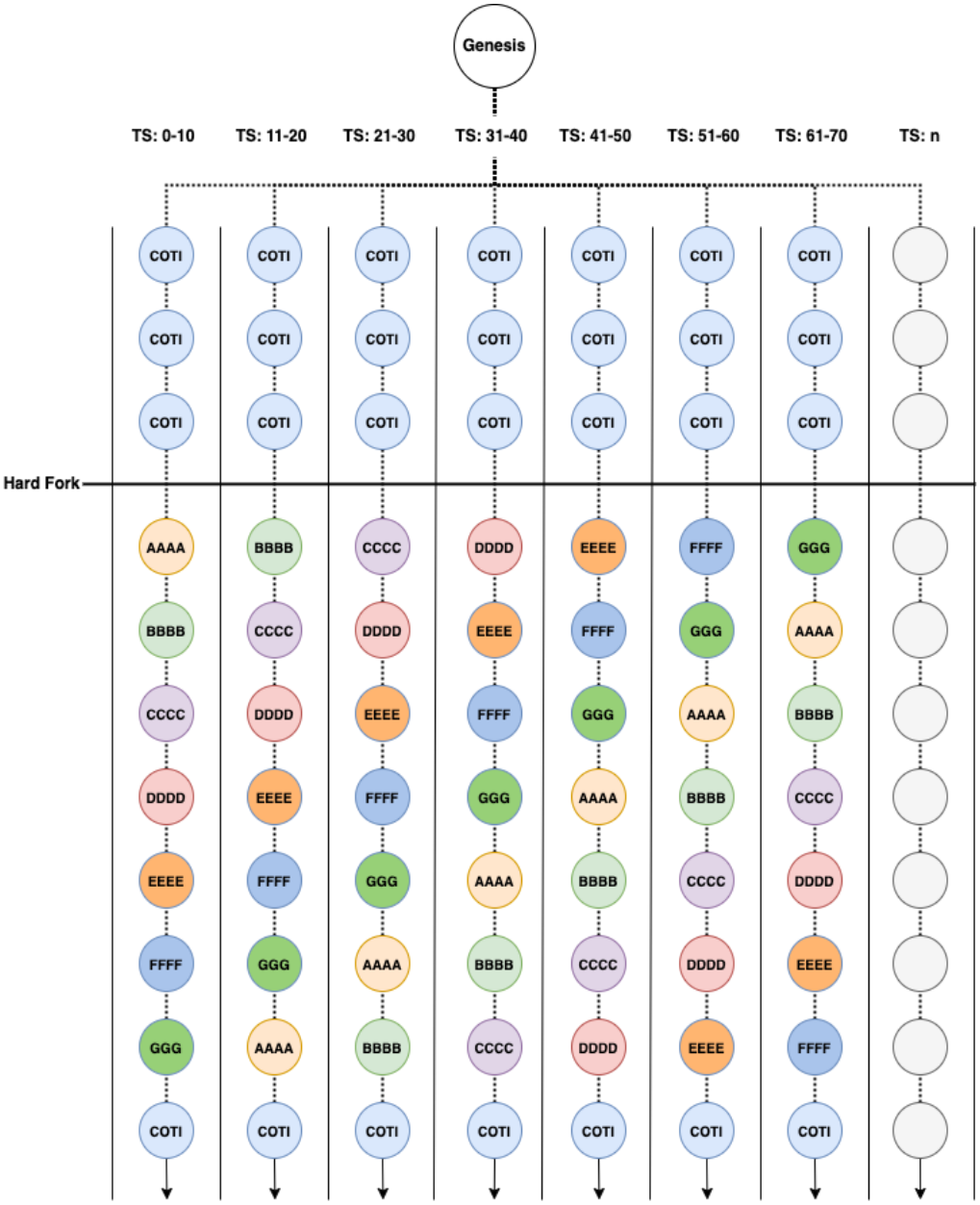
Consensus

Network consensus remains unchanged; nevertheless, adding additional interconnected transactions to the DAG will significantly improve transaction confirmation times for low utilized tokens thanks to highly utilized tokens. Any transaction, regardless of its type or utilized token, will help drive previously attached transactions towards confirmation as long as they are in the same Trustscore range, as can be observed in the diagram.

Burn Tokens

To control token circulation, tokens can be minted to increase circulation and burned to reduce circulation. Clients can reduce token circulation by transferring tokens to a "zero" address, after the transfer transaction has been confirmed, transferred tokens are burned and cannot be retrieved or transferred to any other address. Burning tokens in the following manner: assuming token max supply set to 1M, and issuer minted 900K tokens (circulating supply), remaining mintable supply equals 100K. For example: If 200K tokens are transferred to "zero address" then New Circulating Supply equals 700K (Circulating Supply = Current Circulating Supply - Burned Supply)

NOTE: *This feature is not yet fully implemented and will be introduced later on based on protocol roadmap*

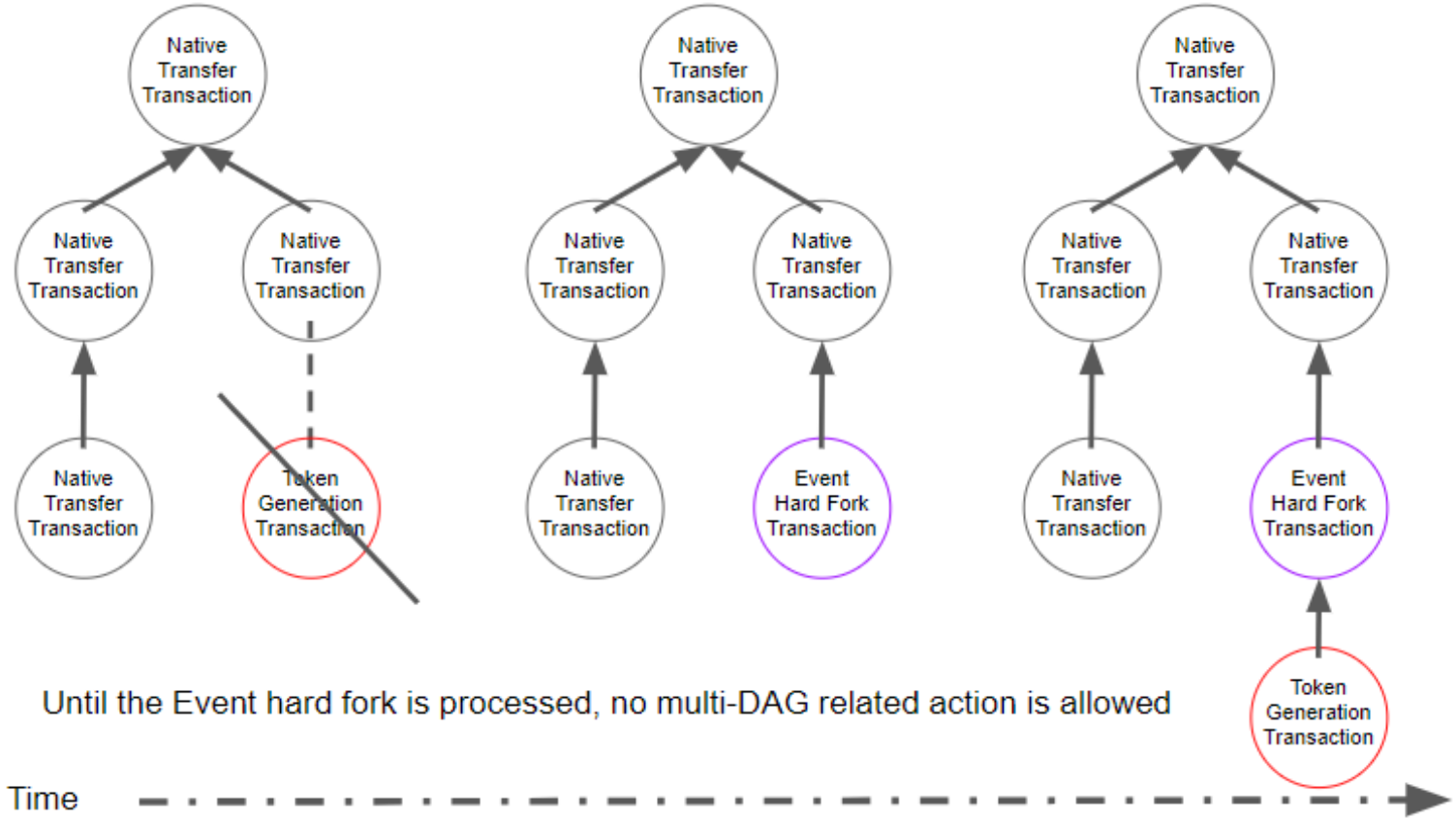


Hard-Fork

A hard-forking event is required to migrate the Full Nodes from MultiDAG 1.0 to MultiDAG 2.0. Once MultiDAG 2.0 completes its trial period, a hard-fork transaction will be transmitted on the network. Full Nodes that have updated to the latest version will be able to initiate the new generate & mint transaction and process transfer transactions containing assets other than $COTI.

Fees

Both the generate and mint transactions require fees to be paid to participating nodes. Fees will be paid using $COTI originated from the Token issuer wallet balance.



Protocol Roadmap

This section describes MultiDAG protocol features based on the version.

Feature	MultiDAG 1.0	MultiDAG 2.0	MultiDAG 2.X
Trustchain Integration	☑	☑	☑
Single DAG Cluster Per Token	☑	☐	☐
Single DAG Cluster For Multiple Tokens	☐	☑	☑
Token Generation	☑	☑	☑
Token Minting	☐	☑	☑
Token Burning	☐	Partial Implementation*	Finalize Implementation
Utility Tokens	☐	☑	☑
Wrapped Tokens	☐	☑	☑
Ledger Support	☐	☑	☑

* Token burning is partially implemented in MultiDAG 2.0 the full set of functionality will be released in future protocol versions.

Summary

MutiDAG 2.0 is a protocol enhancement built on top of the technical knowledge accumulated by building and testing MultiDAG 1.0. based on the acquired knowledge, the main change introduced was the transition from a Single DAG Cluster per Token (1:1) to a Single DAG Cluster for Multiple Tokens (1:n). Based on our research, this transition will result in an overall performance improvement for all transactions performed in the network, regardless of the token ID and of the transactions volume. In addition, MultiDAG 2.0 brings the required capabilities to Generate & Mint Enterprise and Utility tokens that will make a significant impact on business growth.