RoRa Gold
White Paper

Published by:

Monetaforge

www.monetaforge.ky

March 2024

Executive Summary

RoRa Gold is a Permissioned Security Token offered via Monetaforge. **RoRa Gold Tokens** are collectively entitled to 100% of the value of the RoRa Gold portfolio held and managed within the RoRa Holdings SPC company registered in Cayman Islands. The value of the portfolio will be reported by the company, which is also planning for distributions to be made when cash-flow reserves are deemed sufficient, as determined by management. The RoRa Gold Token portfolio will focus on Gold related investments with some excess cash placed in other trading strategies.

RoRa Gold Token has been published on the Polygon Blockchain with Symbol: **RORAG**
RoRa Gold Token Address: **0xE868D8E0347E791F257f22515718b156F27f20bd**

The Token is a Permissioned Security Token. All Token holders MUST register with Monetaforge at www.monetaforge.ky to enable their OnChain ID and permissioning.

The Token Offering is made available to **US Accredited Investors**, as defined in Rule 501(a) of Regulation D pursuant to an exemption from the registration requirements of the Securities Act available under Rule 506(c) of Regulation D and offered to **Non-US Sophisticated Investors** pursuant to Regulation S promulgated under the Securities Act. Sales of Tokens will be made pursuant to subscription agreements. Definition of Sophisticated Investors and suitability requirements vary by jurisdiction.

RoRa Holdings SPC

RoRa Holdings SPC is a privately held Segregated Portfolio Company registered in Cayman Islands with illiquid mixed assets, Gold, and investment portfolios.

Moneta®

Moneta is a privately held holding company registered in the Cayman Islands. Moneta and the **Moneta Community** are on a mission to enable **Tokenization, Fractionalization, Monetization, and Optimization** of **Real World Assets** (**RWA**) by building an ecosystem to ensure Tokenization success. The subsidiaries of Moneta include Monetaforge and Vi Holdings SPC.

Tokenization of Real World Assets (RWA)

As **Tokenization of Real World Assets (RWA)** takes place across the globe and every industry type, the impact of this transformation is beyond measure.

There are many similarities between what happened with the progression and endless impacts seen with the internet changing all aspects of life, business and society, with what we now see happening as the blockchain and related technologies are progressing towards having endless possibilities. In the early 1990's the internet was used by business and society for blogs, general information, and things like pornography. Institutions were reluctant to use the internet for real world business applications, such as banking, commerce and online transactions. By the mid to late 90s the internet technologies, security, standards, and unprecedented rate of adoption gave rise to almost everything going online. The reluctance was overcome because the technology and acceptance of related standards proved worthy. It's clear, for example, banking online is actually far more efficient, more secure, more reliable and enables greater regulatory compliance than the old traditional banking systems.

The **Blockchain technologies** and several important standards, such as **ERC-3643,** have now proven worthy of enabling Tokenization of Real World Assets in ways that are actually far more secure, way more efficient, truly transparent, reliable, accurate, and simply better than any of the traditional forms of systems that have been in place across most industries, while actually enabling far greater regulatory compliance in industries that such is critical.

Like with the internet and new technologies that changed the world forced convergence of various sectors that were traditionally and totally separate, tokenization is forcing the convergence of groups, disciplines, and in some respects generational mindsets, requiring mutual respect and understanding to accomplish great things together. The internet forced the convergence of media, communications, mobile, merchandising, advertising, etc, etc, and gave way to decentralization and access to endless possibilities.

Financial Securities Market participants and regulators are experienced and have deeply established standards and requirements of ensuring important aspects like Anti-Money Laundering enforcements, Restricted Securities rules, requirements of being an "Accredited Investor" for investments, 12-Month Seasoning periods, "Reg D" filings, etc. etc. But many of these participants lack respect and understanding of how and what the Blockchain can do. They are often aware of "Bitcoin" and negative news such as the FTX mess in 2022 and crypto friendly bank collapses that took place in 2023.

Just as institutions were initially reluctant to embrace the internet for business because the internet was used for blogs and pornography, many in the financial industry are reluctant to embrace the blockchain for use in the securities industry because of the FTX mess and Bitcoin radical price movements.

The fact is that Tokenization is a reality and happening at an unprecedented rate. The participants need to embrace, accept, and learn about how Tokenization can implement traditional financial activities far better than the traditional methods.

The typical Blockchain and crypto familiar participants see the power of tokenization, but don't know about nor necessarily respect the regulations or financial market norms and requirements. The reality is that many jurisdictional regulators, such as the US SEC, have deemed most crypto tokens as "Securities". Therefore the Blockchain world participants need to embrace, accept and learn about these regulations.

These sectors are converging at a rapid pace. Entities, such as Monetaforge, often find themselves needing to educate these different groups of participants about the ways to implement so that all aspects and requirements are addressed to ensure successful implementation.

The crypto technology savvy participants are learning to understand and respect the importance of terms like, Know Your Client (KYC), Anti-Money Laundering (AML) Compliance, Ultimate Beneficial Ownership (UBO), Accredited Investors, Seasoning Periods, transfer restrictions, etc. Understanding the typical methods, models, and regulatory requirements in the financial industry is important.

The financial savvy participants need to learn about blockchain standards like ERC-3643, terms like "minting", "white listing", and "Peer to Peer". They should learn about wallets and best practices to secure 12 word passphrases. It is good for them to know about blockchain networks such as Ethereum, and Polygon. Being aware of what Smart Contracts would be also good. Learning these kinds of things may help them to be less reluctant to embrace tokenization.

Monetaforge®

Monetaforge is a **Virtual Asset Service Provider** (**VASP**) registered with Cayman Islands Monetary Authority (CIMA) that provides **Design, Mint, Issue, and Administration** (**DMIA**) of Permissioned Security Tokens. Tokens issued are compliant with USA SEC and other jurisdiction regulations.

Token holders are registered with Monetaforge to enable their OnChain ID and permission to hold a given token by way of the **ERC-3643 Permissioned Tokens** standard combined with smart contracts designed to enable various jurisdictional regulations. The model enables validation and verification of an investor being a qualified investor (such as being an Accredited Investor in USA or Sophisticated Investor for non-US jurisdictions) to purchase a given token, enforcement of various jurisdictional seasoning periods, transfer restrictions rules enforcement, validation of KYC/AML/UBO, etc. All while enabling decentralization of such via the blockchain. This approach also enables greater transparency and compliance with internationally accepted regulations. An example being tracking of a token chain of custody for the life of the token, which is going far beyond the requirements and goals of the Travel Rule.

The ERC-3643, previously known as T-REX protocol, is an open-source suite of smart contracts that enables issuance, management, and transfer of permissioned tokens.

As cryptocurrencies sparked an initial wave of Initial Coin Offerings (ICOs), leveraging Distributed Ledger Technology (DLT) for issuing diverse digital assets in the form of utility tokens, which demonstrated the potential of blockchain as a shared infrastructure for transferring assets. By directly managing tokens in their wallets, users could transfer the token ownership peer-to-peer without

intermediaries, bringing unprecedented efficiency, accessibility, and liquidity to the cryptocurrency market.

Utility tokens are cryptocurrency tokens that merely grant token holders access or the right to participate on platform(s). When you think of Initial Coin Offerings (ICOs) utility tokens are the tokens offered to investors and grant the investor zero rights to the underlying issuers business.

Security tokens (or digital securities/tokenized securities) are securities represented on a blockchain. In many cases, these securities need to be treated as “Restricted Securities” as defined by regulators in various jurisdictions, such as the US SEC. These tokens grant investors rights akin to those of traditional securities, encompassing equity, debt, and more. They can represent any asset class, including small businesses and real estate. Issuers can conduct Security Token Offerings (STOs) to raise funds, however, security tokens are securities, required to comply with traditional securities laws.

Financial markets, especially private markets, want the same level of efficiency, accessibility, and liquidity the crypto world offers. The challenge is that tokenized securities, or security tokens cannot be permissionless tokens like utility tokens, which can be transferred to anyone. They must be permissioned tokens in order to track ownership and make sure that only eligible investors can hold tokens, in order to comply with securities laws. Global regulatory bodies now increasingly recognize these tokens as securities, requiring enforcement of compliance with existing securities laws, including stringent Know Your Customer (KYC) and Anti-Money Laundering (AML) regulations.

Open-Source ERC3643 Token Standard and T-REX

The open-source ERC3643 token standard and its T-REX implementation were designed to address this need to support compliant issuance and management of permissioned tokens, that are suitable for tokenized securities, either on a peer-to-peer basis or through regulated trading platforms. These tokens are issued in full compliance with the rules specified by the investors (via on-chain identity) and the offerings based on issuers' guidelines. Furthermore, control mechanisms are baked into the tokens themselves. Adopting a “Compliance by Design” approach, T-REX ensures that an investor cannot become a holder of any digital securities without fulfilling all compliance requirements. Furthermore, regulators can affirm the issuer's compliance by auditing the smart contracts that underpin the entire life cycle of the security token. This innovation offers a secure, transparent, and efficient environment for managing security tokens while enforcing on-chain compliance, heralding a new era in the financial securities market.

The management of compliant transactions through T-REX backed permissioned tokens is be based on 4 main pillars creating a decentralized validator:

1. **OnChain ID** is a blockchain based identity management system, allowing for the creation of a globally accessible identity for every stakeholder.
2. **Validation certificates**, or verifiable credentials, (technically speaking, these certificates are the claims, described in the ERC-735 standard used by OnChain ID. Consider these like certificates emitted by trusted third parties and signed on-chain, each of them linked to a single OnChain ID.

3. **Eligibility Verification System (EVS)** whose role is to act as a filter of all the transactions of tokenized securities and will check the validation certificates of the stakeholders. Essentially, the EVS will check that the receiver has the rights to receive the tokens following the specific offering rules and issuer requirements, for investors, applicable for this specific asset. The EVS will block the transaction if the receiver misses a mandatory certificate and will notify them about the reason for the failure. The OnChain validator is implemented on the Identity Registry smart contract through the “isVerified” function.
4. **Compliance rules** (i.e. offering rules) ensuring that the rules of the offering are respected, e.g. the maximum of investors per country of distribution, the maximum of tokens held by a single investor, etc. These rules are not only linked to the identity of the receiver of a transaction but also to the global distribution of tokens at a certain time. The Compliance rules are implemented on the Modular Compliance smart contract through the “canTransfer” function.

These 4 key elements allow issuers to use a decentralized Validator to control transfers and enforce compliance on the holders of the security token. The Validator includes rules for the whole offering (e.g. managing the max number of holders allowed in a specific market, when such rules apply), and rules for each investor (e.g. KYC or issuer-defined eligibility criteria) thanks to the identity management system.

The OnChain ID is an identity system that enables the creation and management of self-sovereign identities on the blockchain. This identification solution allows the enforcement of regulatory legal and other compliance processes and rules regarding digital assets, and enforcement of such processes and rules on any web-based system that requires identity validation.

The main added value of the OnChain ID is that it enables compliance and identity verifications within the pseudonymous framework of public blockchain networks. As the blockchain industry matures, its application across economic activities broadens and accelerates. TheOnChain ID protocol can be used for permissioned “Decentralised Finance” or “DeFi”, bringing the compliance layer currently missing for the institutional players to step in. It can also be used to represent the identification data of virtually anything (real estate, art, financial assets, connected objects, etc).

This White Paper will explain in greater detail the ERC-3643 and T-REX architecture which forms the basis for how Monetaforge designs the tokens they deploy, which is the model of implementation for **RoRa Gold Tokens**.

Disclaimer and Risk Warnings:

Cryptocurrency and security tokens are speculative in nature and involve substantial risk, **including the risk of complete loss**. Past performance has no bearing on future performance and there can be no assurance that any cryptocurrency, security token, coin, or any other crypto asset will be viable, liquid, or solvent. Nothing in this document or any **Monetaforge** communication is intended to imply that any of the digital assets published by Monetaforge are low-risk or risk-free. Anyone considering investment in tokens should conduct their own due diligence before investing. **Monetaforge** is NOT endorsing any given token investment.

Tokeny is also not responsible for or providing any investment advice. Information in this document for informational purposes ONLY to describe the technical architecture of the Tokens.

Token Architecture Model

The implementation of a Permissioned Security Token requires a Token Architecture Model that considers, manages and enforces a complex and diverse set of rules and regulations that can vary across different jurisdictions. The model must take into account that different restrictions may apply depending on the token holder resident, whether or not they qualify as an Accredited Investor, different seasoning periods, and exemptions that may apply, to name a few.

Constraints for Tokenized Securities

The rules governing utility tokens and their issuance remain relatively undefined or vague in most jurisdictions, Security Token Offerings (STOs) represent a different paradigm. STOs utilize blockchain technology as a registry, proof of ownership, and transfer infrastructure for securities, which are regulated instruments in every country. Consequently, STOs must comply with the relevant regulations in the countries where the security tokens are sold to investors, issued, and distributed.

	Utility Token	**Security Token**
Purpose	Usage	Investment
Regulation	Non-existing or vague in most cases	Stringent as existing securities laws should be taken as reference
Life cycle	Simple	As complex as a security
Secondary Market	Nearly no constraints	As complex as a security

Figure 1 : comparison between utility and security token

A key distinction between Initial Coin Offerings (ICOs) and STOs resides in the token lifecycle. ICOs, dealing with utility tokens, yield tokens with a relatively straightforward lifecycle: once distributed across a decentralized network, their governance primarily stems from their token economics. However, for security tokens, the landscape is distinct. The issuer, or its appointed agent, generally remains liable for enforcing controls post-issuance and throughout the entire lifespan of the security token. Additionally, the issuer may need to execute corporate actions (like dividend/interest payments) or corporate events (such as calling for an AGM/EGM), necessitating ongoing interaction with (and some control over) their investors.

Security Tokens usually require various forms of administration over the life of the token. This enables things like mint, burn, recovery, communications and facility of managing legal exceptions to various regulatory restrictions. In many cases this model can actually implement the regulations far better than the traditional systems of brokers and exchanges. For example, many brokers enforce the 12 month seasoning period allowing for the legal exemptions that are allowed by the regulations because they don’t have a

system to manage it properly. Within the 12 months seasoning, a security can be sold and transferred to another “Accredited Investor” as long as the remaining period of seasoning transfers with the security. This is difficult and often not possible to manage with existing broker and exchange systems, so they just block the transfers for 12 months with no exceptions. A well designed Token Architecture Model can manage this situation.

Two main types of controls are associated with the issuance, holding, and transfer of security tokens:

1. **General Regulatory Controls:** These are regulations applicable to the security in question, independent of the token itself. For instance, obligations related to Anti-Money Laundering (AML) and Know Your Customer (KYC) protocols, such as investor identification, proof of identity verification, and blacklist checks, fall under this category.

2. **Specific Security Controls:** Certain controls might be tied specifically to the security being issued. These could include restrictions related to the investor's type, location, or investment limit within a specific period. Such controls may arise from the regulatory environment the issuer operates within or be linked to eligibility criteria defined by the issuer for commercial reasons (e.g., limiting access to a specific share class with distinct fee characteristics to investors from a particular country).

To meet these diverse control requirements, a high level of reusability and flexibility is crucial in token design. This inspired us to develop the ERC-3643 standard. This standard provides a suite of generic tools that aid token issuers in implementing and managing necessary controls and permissions for security tokens. This is achieved through a flexible decentralized validation system (EVS + Compliance contract), enabling relevant parties to establish necessary rules for approving the holding and transfer of tokens.

In order to reach critical mass of users and partners as quickly as possible, Tokeny, the company that initiated the OnChain ID system, has decided to open up the governance of the protocol by issuing OID, a governance token to incentivise the various players in the value chain and accelerate the decentralization of the ecosystem. OID tokens allow holders to participate in the governance of OnChain ID and interact more broadly with the protocol, its applications, and its participants. Numerous entities such as securities issuers, investors, auditors, governmental entities, large financial institutions, custody solutions and law firms have already accepted to join the OnChain ID ecosystem.

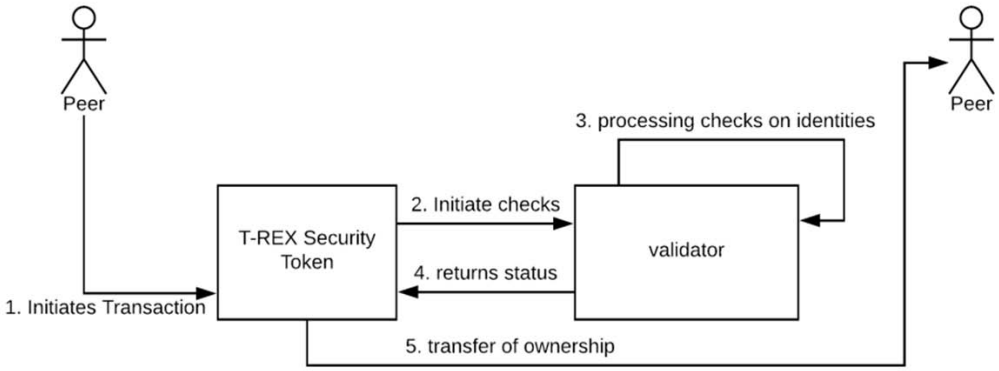
Monetaforge is a participant of this ecosystem.

Decentralized Validation System

Transfers of ERC-20 tokens typically proceed in a specific manner on any Ethereum Virtual Machine (EVM) blockchain, adhering to the standard ERC-20 implementation:

Transactions occur directly between two peers without restrictions or controls. Transactional freedom is comprehensive and pseudonymous, with AML/KYC checks primarily conducted when cryptocurrencies convert into fiat currencies or vice versa. However, various methods allow circumventing these checks, such as unregulated exchanges or direct peer-to-peer exchanges.

An ERC-3643 compliant ERC-20 permissioned tokens transaction follows a more controlled process:



1. **Transaction Initiation:** The token holder initiates the transaction via the security token smart contract. This action differs from a standard ERC-20 token, as the smart contract's transaction function has been modified
2. **Validator Engagement:** The smart contract's transfer function calls upon the validator (comprising the compliance contract, identity registry, identity registry storage, trusted claim issuer registry, and trusted claim topics registry) to initiate checks on the receiver's OnChain ID.
3. **Compliance and Eligibility Checks:** The validator ensures that the receiver's OnChain ID holds the necessary claims, issued by a trusted third party listed in the trusted issuers registry. It also verifies that the proposed transfer complies with rules set forth in the compliance smart contract.
4. **Evaluation of Transfer:** If the receiver's OnChain ID possesses the requisite claims (personal data validated by trusted third parties such as KYC, AML, sovereign identity, etc.), and the transfer does not violate any compliance rules, the transfer of tokens is allowed to proceed. However, if the OnChain ID lacks necessary claims, or the transfer breaches any compliance rule, the transfer is rejected.
5. **Transfer Execution or Rejection:** If the checks are successful, the transfer of tokens is executed. In case of a rejection, an error message is generated, explaining the necessary steps to acquire the missing claims or the reason for the transfer's non-compliance.

This process embodies the ethos of T-REX: ensuring compliant and secure transactions for all parties involved in the token exchange.

ERC-3643 Permissioned Tokens

Using the ERC-3643 Permissioned Tokens standard is the most suitable for issuing security tokens. The reason for this lies in the inherent need for controlled transactional freedom when dealing with such financial instruments, as investors must meet certain eligibility criteria. These criteria can be regulatory in nature or stipulated by the issuer themselves.

The primary technical distinction between standard ERC-20 tokens and ERC-3643 permissioned tokens lies in the conditional nature of the transfer function in ERC-3643 tokens. Transaction can only be executed if a decentralized validator approves it, based on specific governance criteria defined for the token.

Despite this modification to the transfer function, it's important to note that ERC-3643 tokens maintain full compatibility with all ERC-20 based exchanges and tools due to their underlying structure. Integration into an existing ERC-20 compatible platform requires a minor modification: processing pre-checks before any transfer to verify the transaction's compliance status with the decentralized validator. If a transaction fails to meet compliance, the platform should provide clear feedback on why it's non-compliant and, where possible, guidance on achieving compliance. However, in certain cases, compliance may not be achievable, for example, when attempting a transfer to a resident of a country listed on the T-REX compliance blacklist.

In the realm of "security token protocols", most solutions promoted so far are indeed permissioned tokens. They feature a modified transfer function that requires approval from an external validator service to control token transfers. T-REX goes a step further with its implementation, incorporating a fully on-chain identity management system. This allows issuers to have direct control over the transfer of ownership, enhancing the security and compliance of the system.

Onchain Identities Management

With security tokens subject to rigorous governance, adhering to all relevant regulations, particularly Know Your Customer (KYC) rules, is paramount. As such, we posit that effective identity management is crucial to ensuring this compliance on the blockchain.

Given that security token ownership is recorded on the blockchain, it's essential to track token ownership and inhibit illicit transactions directly on-chain. To this end, we propose linking wallet addresses to identities via an OnChain ID contract on the blockchain itself. However, to maintain privacy and comply with personal data regulations, we suggest storing only validation certificates (claims) issued by trusted third parties (KYC providers, government entities, lawyers, etc.) on-chain, rather than personal data. These trusted parties validate the data off-chain, and their certificates are utilized by the decentralized validator to determine whether parties can hold or transfer a specific security token.

Connecting an investor's wallet to an OnChain ID can bring substantial benefits to stakeholders in the burgeoning security tokens market. For instance, should an investor lose access to their wallet, a token

issuer can use the linked OnChain ID to recover the investor's tokens. This process involves verifying the investor's personal data provided during recovery against the off-chain data associated with the OnChain ID contract tied to the lost wallet. Once the investor's identity is confirmed, the issuer's agent can initiate the recovery function. This action forces a transfer of tokens from the lost wallet to a newly provided wallet, while also updating the identity registry and the OnChain ID contract.

Furthermore, OnChain IDs and the stored certificates can be reused for passing KYC for other security tokens or even in scenarios beyond investment (e.g., account opening at an exchange, identification with compatible web services). In the realm of the Internet of Value, OnChain IDs could become as ubiquitous as Google and Facebook accounts in the Internet of Information, with the added benefit of being genuinely owned and controlled by their users.

T-REX Infrastructure

Overview

T-REX aims to deliver a comprehensive toolset for issuing, managing, and transferring security tokens on EVM blockchains, leveraging the power of the ERC3643 permissioned token standard. The subsequent sections will delve into T-REX's technical intricacies, explaining the functions and capabilities of the various smart contracts integral to its operation.

Furthermore, T-REX is designed with extensibility in mind, allowing for easy integration of additional smart contracts to manage corporate actions, taxes, and other related aspects.

Based on Standards

An important aspect of the token model architecture and the T-REX technical implementation is compatibility with industry Token Standards

ERC-20

ERC-20 tokens are recognized as an industry standard, embraced by all players in the blockchain sector. 3 These tokens are fungible, often non-permissioned, and facilitate effortless peer-to-peer transfers on EVM blockchains.

https://github.com/OpenZeppelin/openzeppelin-solidity/tree/master/contracts/token/ERC20

The ERC-20 smart contract outlines and implements the fundamental attributes of the token, including its name, symbol, total supply, and the number of decimals for display purposes. Essentially, the ERC-20 smart contract equips the token with all the requisite functionalities to meet the standards of a conventional token.

Identity standards on the Blockchain

Incorporating robust KYC and AML procedures is non-negotiable for any project seeking to adhere to stringent governance standards and regulatory compliance. This is particularly relevant for Security Token Offerings (STOs), which are mandated by regulations to enforce strict KYC and AML protocols. We believe that the most effective way to implement these checks within a blockchain infrastructure involves linking them to a universally accepted identity model.

To this end, we have adopted OnChain ID, which is built on the ERC-734 and ERC-735 standards. These standards provide a widely accepted model for creating, maintaining, and populating identities on the blockchain. By leveraging these standards, we've built functionalities into our smart contracts that permit interactions only from well-identified, reputable entities, be they individuals or organizations. The claims attached to an identity contract foster a web of trust among token issuers, third-party claim issuers, and the identity contract holder, thereby streamlining identity management.

To add a claim to their OnChain ID, an identity holder must first request it from a relevant trusted third party (a trusted claim issuer for the specific claim topic, as listed in the trusted issuers registry smart contract). Upon verifying the requestor's eligibility for the claim, the claim issuer signs a message containing: a) the requestor's OnChain ID address, b) the claim topic, c) optional data (e.g., hashes to off-chain data references stored by trusted claim issuers), and d) a valid signature of the claim information by a key listed on the Claim Issuer smart contract. The identity owner (the claim holder) then stores this claim in their identity contract. Alternatively, the claim issuer can add the claim themselves, provided the identity owner has given approval.

In the T-REX framework, it's this synergy of identities and claims that enables on-chain validation and verifies an investor's eligibility to hold a specific security token. Each time an investor is slated to receive a position in a specific security token, T-REX verifies whether their OnChain ID contains the necessary claim(s) to hold that token.

https://github.com/ethereum/EIPs/issues/735
https://github.com/ethereum/EIPs/issues/734

Proxy Standards (ERC-1822 and Beacon Proxy)

ERC-1822

ERC-1822 establishes a universal standard for proxy contracts, guaranteeing compatibility with all other 6 contracts. It achieves this compatibility by storing the Logic Contract’s address in a distinct storage location within the proxy contract. A compatibility check also allows for successful upgrades, which can be performed indefinitely or as defined by custom logic. Furthermore, the standard provides a method to choose from multiple constructors without inhibiting bytecode verification.

In the context of the T-REX protocol, we adopt an adapted version of the Universal Upgradeable Proxy Standard (UUPS) proposed by ERC-1822. However, we do not directly store the implementation contract's address in the proxy storage as per the ERC-1822 standard. Instead, the T-REX proxy contract stores the address of an external contract, known as the Implementation Authority.

Beacon Proxy

The Beacon Proxy Standard introduces another approach to managing contract upgrades. In its structure, a beacon proxy points to a beacon contract which stores the address of the current implementation. This beacon contract can update its implementation address, thereby upgrading all beacon proxies that reference it.

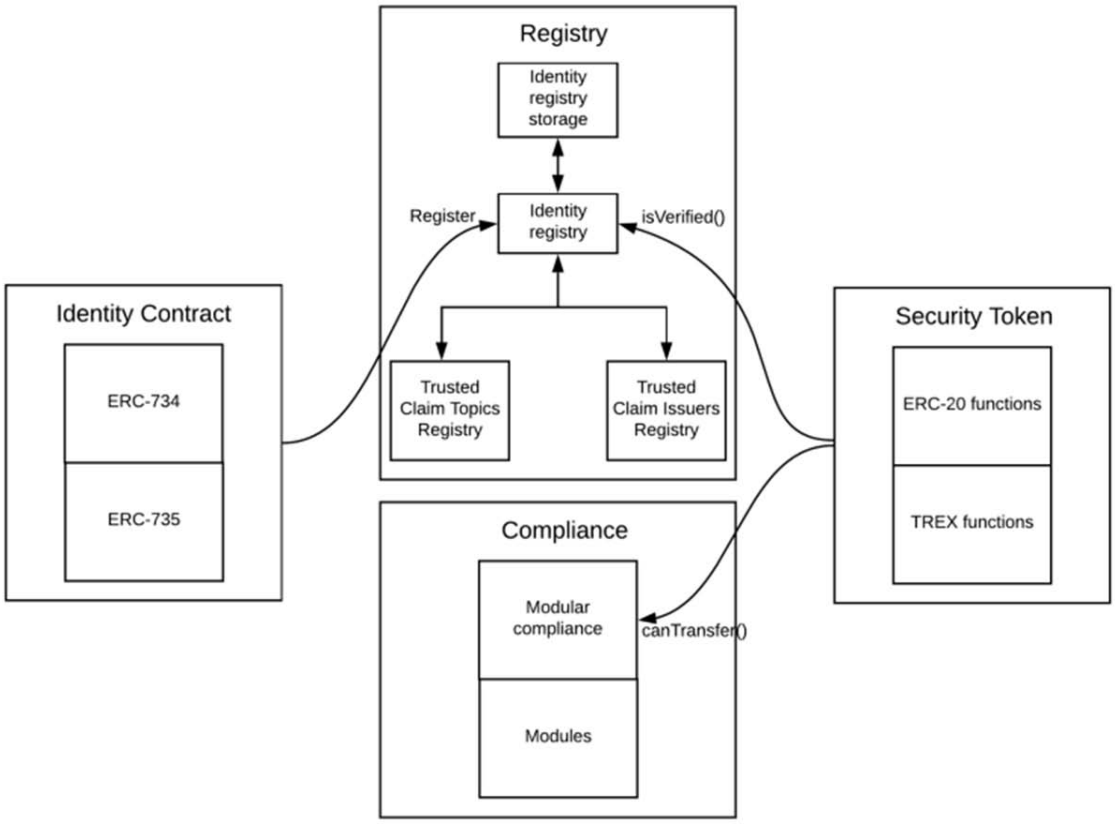
In the T-REX protocol, we harness the structure of the Beacon proxy, but implement it like an ERC-1822 contract. Our proxy contract references an external contract, the Implementation Authority, which contains the addresses of all T-REX contract implementations. This Implementation Authority address fits into the storage slot that the ERC-1822 standard reserves for implementation.

The benefits of the T-REX protocol's method include centralized versioning management for all T-REX tokens deployed using the same Implementation Authority address, making it an ideal choice for simultaneous token upgrades. Token issuers can change the address of their tokens' Implementation Authority contract, giving them the power to manage versioning independently or delegate it to any trusted third party. The T-REX factory contract sets the Implementation Authority address during token deployment by default.

https://docs.openzeppelin.com/contracts/3.x/api/proxy#beacon
https://eips.ethereum.org/EIPS/eip-1822

T-REX Components (Smart contracts library)

Below is the illustration of T-REX components with global interactions:



OnChain ID

An OnChain ID contract is a smart contract deployed by a user to interact with the security token, or for any other application where an onchain identity may be relevant. Based on the ERC-734 and ERC-735 standards, this contract stores keys and claims related to a specific identity, and encompasses all essential functions for managing these elements.

The OnChain ID contract is not tied to a specific token, meaning each user needs to deploy it only once. Afterward, it can be used in any context where an onchain identity may be beneficial. A comprehensive description of the functions is available in the OnChain ID documentation . 8 Moreover, an OnChain ID contract is also deployed and associated with the token smart contract to represent the identity of the financial asset itself. The claims provide information about the asset and all pertinent corporate actions and details regarding the security. Essentially, it serves as an onchain "golden copy" of the asset, and can be augmented with any relevant claim throughout the token's life cycle.

Identity Registry

The Identity Registry smart contract serves as the execution hub for the Eligibility Verification System (EVS). It establishes connections with the Trusted Issuers Registry and Claim Topics Registry to stipulate identity requirements and executes the "isVerified"function. This function compares the EVS requirements with the identities of investors to ascertain their eligibility status.

The Identity Registry contract is linked to the Identity Registry Storage contract, which houses a dynamic "whitelist" of identities. Utilizing the Identity Registry Storage, the Identity Registry can retrieve the association between a wallet address, an OnChain ID, and a country code that denotes the investor's country of residence. This country code is set in compliance with the ISO-3166 standard.

The management of the Identity Registry lies with the issuer's agent(s), meaning only the issuer's agent(s) can execute functions to add or remove identities from the Identity Registry Storage. It's worth noting that the agent role on the Identity Registry is assigned by the issuer, who can designate themselves as the agent if they prefer to maintain complete control.

Each security token has a specific Identity Registry, as each one is associated with a Claim Topics Registry and a Trusted Issuers Registry smart contracts. These contracts dictate the eligibility rules of the token and are unique to a token. A comprehensive description of the functions is available in the T-REX contracts documentation.

https://docs.tokeny.com/docs/identity-registry
https://www.iso.org/iso-3166-country-codes.html
https://docs.onchainid.com/

Identity Registry Storage

The Identity Registry Storage contract functions as a repository for a mapping table, associating wallet addresses with the corresponding OnChain ID addresses of investors. This correlation enables the Identity Registry to extract the OnChain ID linked to a specific wallet, thereby retrieving the claims associated with that particular OnChain ID address. The roster of wallets and identities contained in this mapping encompasses all individuals whose data the issuer possesses and might need to utilize.

At its most basic level, this list should incorporate investors who have undergone the necessary Know Your Customer (KYC) and eligibility evaluations, and are consequently authorized to hold the issuer's security tokens. Moreover, it can also include potential future investors.

The Identity Registry Storage can be tied to one or multiple Identity Registry contracts. Its primary objective is to segregate the Identity Registry's functions and specifications (which determine eligibility rules specific to a token) from its storage. This arrangement permits the maintenance of a separate Eligibility Verification System (EVS) for each token while sharing the list of investors vetted by the isVerified() function, implemented in the Identity Registries. This function checks the eligibility of the

receiver in a transfer transaction. A thorough description of the functions is available in the T-REX contracts documentation.

Trusted Issuers Registry

The Trusted Issuers Registry smart contract serves as a storage system for contract addresses (OnChain IDs) of all trusted claim issuers specific to a given security token. Trusted claim issuers, along with the claim topics they are authorized for, reside within the Trusted Issuers Registry, with each issuer possessing their unique set of responsibilities concerning the claims they are permitted to issue.

To qualify for holding the token, the OnChain ID of token owners (the investors) must possess claims endorsed by the claim issuers housed in this smart contract. The token issuer retains the ownership of this contract, providing them with the ability to manage this registry in accordance with their requirements. A comprehensive description of the functions is available in the T-REX contracts documentation.

Claim Topics Registry

The Claim Topics Registry smart contract houses all the claim topics necessary for holding the security token. The OnChain ID of token owners (the investors) must possess claims of the topics stored within this smart contract, which must be issued by the corresponding trusted issuers as cataloged in the Trusted Issuers Registry. Ownership of this contract is vested in the token issuer, providing them with the autonomy to manage this registry in line with their specific needs. An exhaustive explanation of the functions is made available in the T-REX contracts documentation.

https://docs.tokeny.com/docs/claim-topics-registry
https://docs.tokeny.com/docs/trusted-issuers-registry
https://docs.tokeny.com/docs/identity-registry-storag

Permissioned Token

T-REX security tokens are built on the fundamental structure of the established ERC-20 standard, but are enriched with additional functions to ensure full regulatory compliance in security token transactions. The transfer and transferFrom functions are designed conditionally, executing a transfer solely upon the approval of the decentralized validator (EVS + Compliance). It means that transfers can only be triggered when both investor rules (via EVS) and offering or additional rules (via Compliance) are met.

This ensures that the permissioned tokens can only be transferred to verified parties, thereby preventing the tokens from landing in the wallets or OnChain IDs of ineligible or unauthorized investors. The T-REX standard also features a robust mechanism for the recovery of security tokens, should an investor lose access to their wallet's private key. A historical record of all recovered tokens is meticulously maintained on the blockchain for the sake of transparency.

Beyond these features, T-REX tokens implement a plethora of additional functions. These empower the token owner or their designated agent(s) with the capability to manage aspects such as supply, transfer rules, lockups, and any other facets integral to managing a security at their discretion. For a detailed explanation of these functions, please refer to the T-REX contracts documentation.

Modular Compliance

The Compliance smart contract is a dynamic tool used to establish the rules of the token offering. These rules, once set, are meticulously adhered to throughout the entire lifecycle of the token. For instance, the compliance contract can define the maximum number of investors per country, the maximum quantity of tokens per investor, and the countries where the token can circulate. The latter is achieved by using the country code corresponding to each investor in the Identity Registry.

Designed with a modular approach, the compliance smart contract grants Token Issuers the flexibility to add or remove compliance modules. Each of these modules corresponds to a distinct and independent compliance rule. This flexibility enables the issuer to shape compliance rules based on the specific requirements of the token transfers.

The compliance contract is activated at every transaction by the Token. It then evaluates whether the transaction is in compliance with the rules of the offering. It returns a TRUE if the transaction adheres to the rules and a FALSE if it does not. For an in-depth description of the functions, please refer to the T-REX contracts documentation on restrictions..

https://docs.tokeny.com/docs/compliance
https://docs.tokeny.com/docs/token-smart-contract

Monetaforge has developed a custom compliance module smart contract to handle the regulatory requirements for transaction restrictions. This includes configuring the mandatory “seasoning period” or holding/escrow time that is required by different country jurisdictions, as well as managing the transfer of these holds when a token transfer occurs. This is often described as the “travel rule”.

These compliance holds are created whenever a mint transaction is executed and they are stored in a separate smart contract on a per OnChain ID/identity basis. This allows for an investor who needs to change their wallet address to do so without losing their compliance holds on their tokens.

The custom compliance module also performs rule checks when a token transfer occurs to determine if the transfer is allowed based on the compliance hold, accreditation status or other country specific requirements. A configurable component allows for quick addition/modification of these rules.

Implementation Authority

The Implementation Authority contract, as previously discussed in the proxy section, is a pivotal contract that oversees the versioning of T-REX contracts utilized by proxies. It operates in two distinct capacities:

1. **Main Implementation Authority Contract:** This contract is utilized by the T-REX factory to deploy all T-REX tokens. The owner of this contract can add new versions of the T-REX contracts and decide to upgrade all proxies linked to the implementation authority to a different version. Furthermore, it enables token owners to deploy auxiliary Implementation Authority contracts and modify the contract their proxies use for versioning.
2. **Auxiliary Implementation Authority Contracts:** These are deployed when a token issuer wishes to change the reference Implementation Authority contract for their proxies. Unlike the main contract, auxiliary contracts cannot add new versions; they can only fetch versions from the main contract, which serves as a constant reference. However, they can opt to run a different version at any given time compared to the main contract. The owners of auxiliary contracts have the flexibility to revert to the main contract whenever necessary.

For a comprehensive understanding of the functions, please refer to the T-REX contracts documentation. https://docs.tokeny.com/docs/implementation-authority

Factory

The T-REX Factory Smart Contract embodies a robust utility, enabling the simultaneous deployment and configuration of all contracts within the T-REX suite in a single blockchain transaction. This transaction is characterized by its flexibility, accommodating numerous parameters that align with the unique requirements of the token issuer. The T-REX Factory references the main Implementation Authority contract. The owner of the T-REX factory retains the privilege to update this reference point to a different contract as needed.

All contracts deployed by the factory are in the form of proxies, pointing to the main Implementation Authority. These proxies are deployed utilizing the CREATE2 opcode. This allows for the deployment of a token at the same smart contract address across multiple EVM blockchains, provided the factory itself is deployed at an identical smart contract address on the different blockchains.

To prevent misuse, the T-REX deployment function is safeguarded by an “onlyOwner” modifier at the contract level. This is crucial to avoid scenarios where someone could deploy a token with the same address on a different blockchain without being the same owner in control across the blockchains, potentially leading to scams and confusion.

The owner, who has the ability to call the T-REX deployment function, can either be a wallet (though not recommended) or an external smart contract. The latter manages deployment roles and ensures the same token address cannot be deployed by different users on different blockchains. This can be achieved by utilizing oracles or simply incorporating the T-REX owner’s wallet as part of the salt used by the CREATE2 opcode to generate deterministic addresses.

For an exhaustive understanding of the functions, please refer to the T-REX contracts documentation.

https://docs.tokeny.com/docs/factory

Additional Smart Contracts

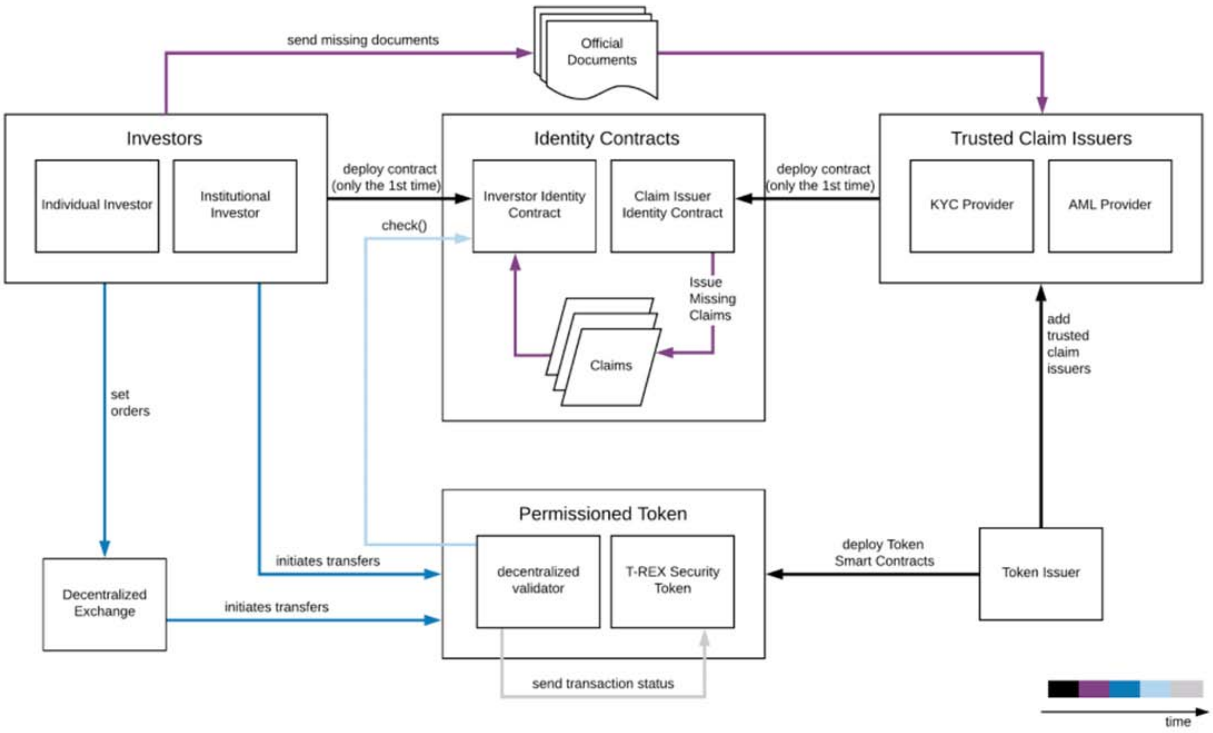
In addition to the core T-REX suite, numerous supplementary smart contracts can be incorporated to enhance the functionality of security tokens and cater to the specific requirements of issuers and investors. By incorporating these features, the T-REX ecosystem is able to seamlessly integrate traditional securities market functions with the inherent benefits of blockchain technology.

- **Investor Rights and Opportunities:** Investor rights, such as voting, dividends, and announcements, can be easily implemented by the issuer or tokenization platform. This allows for the creation of a more comprehensive and adaptable security token ecosystem. Other potential investor rights that could be integrated into the protocol include:
    - Participation in corporate governance decisions
    - Real-time access to company financial information
    - Transparent communication channels between issuers and investors
- **Harnessing the Power of Blockchain:** Tokenized securities can leverage the unique capabilities of blockchain technology to improve market efficiency and user experience. By implementing additional smart contracts, the T-REX protocol can offer:
    - Enhanced transparency for all market participants
    - Elimination of multi-stage reconciliations
    - 24/7 processing and global reach
    - Improved security and immutability of records
- **Tax Compliance and Automation:** The T-REX protocol can also incorporate smart contracts that address tax compliance, automating the calculation and distribution of taxes on token transactions. This feature would greatly streamline the process and ensure regulatory compliance for both issuers and investors.

In summary, the T-REX protocol can be expanded to include a wide array of additional smart contracts, enriching investor rights, and harnessing the full potential of blockchain technology. By doing so, it sets the stage for a new era in the securities market, characterized by unparalleled efficiency, transparency, and global accessibility.

Stakeholders

Overview



Issuer

As the primary stakeholder in the T-REX protocol, the issuer – the entity deploying the security token – assumes a vital and multifaceted role. Tasked with overseeing and managing the comprehensive suite of smart contracts, the issuer's responsibilities extend far beyond the initial token deployment.

- **Determining Claim Topics and Trusted Issuers:** The issuer's first responsibility is to establish the claim topics required by their investors and identify the trusted claim issuers who can verify those claims. These determinations are influenced by several factors including the jurisdiction of issuance, the intended distribution countries, and the unique attributes of the security token itself.
- **Engaging with Issuance Platforms:** Issuance platforms can provide invaluable assistance to issuers, helping them deploy and manage the necessary smart contracts that underpin their token. Additionally, these platforms can facilitate investor engagement, assisting in creating their OnChain ID or connecting with an existing OnChain ID, as required.
- **Administering the Suite of Smart Contracts:** Upon the deployment of a token, the issuer assumes ownership of the suite of smart contracts at the blockchain level. This places them in a pivotal role, responsible for administering the token and adding agents. While the issuer can execute these tasks independently, they may also utilize interfaces provided by a tokenization platform for streamlined management.

In summary, the issuer's role within the T-REX framework is comprehensive and multifaceted, extending from the initial determination of claim topics and trusted issuers to the ongoing administration of the suite of smart contracts. Their role is central to the successful deployment and management of security tokens within the T-REX ecosystem.

Claim Issuers

Claim issuers occupy a pivotal role within the T-REX ecosystem, authorized by token issuers (e.g., a designated third-party KYC platform tasked by the issuer to conduct KYC checks) and registered in the Trusted Issuers Registry. As a Trusted Claim Issuer, they have the authority to augment a specific investor's OnChain ID.

Claim issuers can maintain multiple signer keys (at least one) in their own OnChain ID. This model fosters true interoperability among multiple KYC/AML providers. By employing a standard protocol to verify claims, it insulates Issuers from the specificities of individual AML/KYC providers. The Claim Issuers, therefore, serve as an essential link in the T-REX ecosystem, facilitating the addition and verification of claims while promoting a unified and interoperable environment.

Distributors, Exchanges and DeFi

In the T-REX ecosystem, exchanges also need to manage identities and hold claims to qualify as eligible token holders and accept T-REX token deposits. The integration of identity management within exchanges is a critical aspect that hinges significantly on the exchange's internal transfer mechanism.

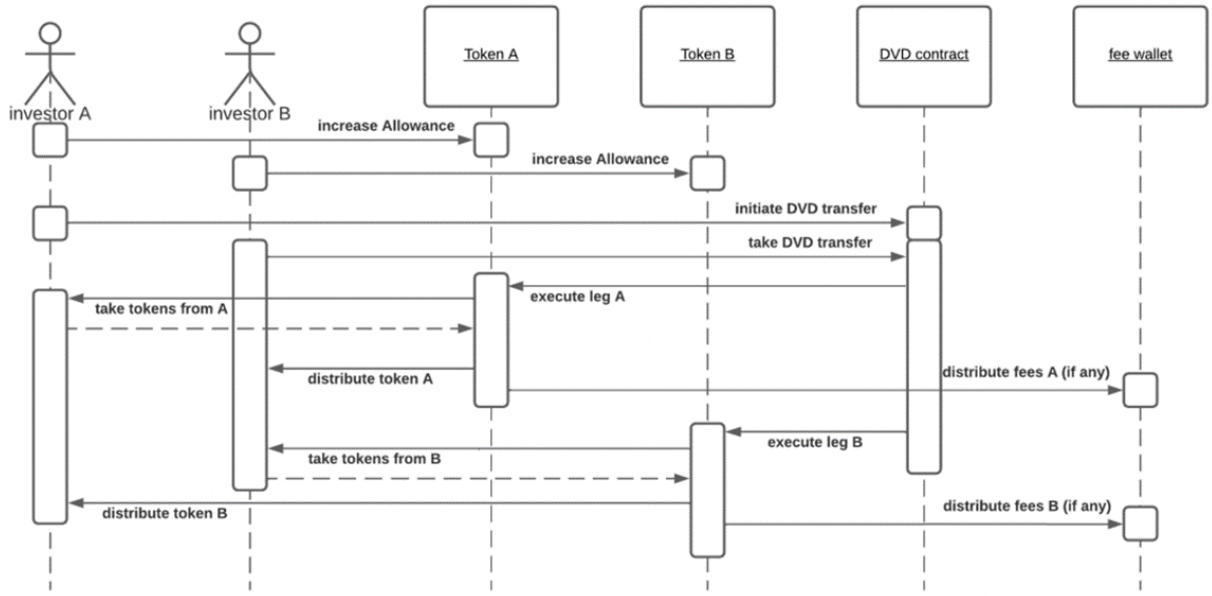
Depending on the specific transfer mechanism in place, the integration method of the identity protocol will vary. This foundational understanding will be further detailed and explored in the following subsections:

Direct P2P Trades

In the T-REX ecosystem, individual investors have the capability to directly trade tokens among themselves. However, the compliant service of the token restricts any transfer of tokens to unauthorized addresses. For a successful transfer, several conditions must be met:

1. The buyer must have a valid identity in accordance with the security token standards, and this identity must be registered in the identity registry.
2. The buyer must also satisfy the claim requirements of the token.
3. The token should not be within its lockup period.
4. Lastly, the token transfers must adhere to the compliance rules set forth in the compliance contract.

The T-REX repository provides an implementation of a Direct P2P Bilateral Transaction Manager, known as the Delivery-versus-Delivery (DVD) Transfer Manager. This contract ensures secure management of transfers, where a transfer can only proceed if the counter-transfer is successful between eligible investors, following an off-chain agreement about the transaction details. The sequence diagram below illustrates the operation of a DVD contract, including potential fees collection and distribution.

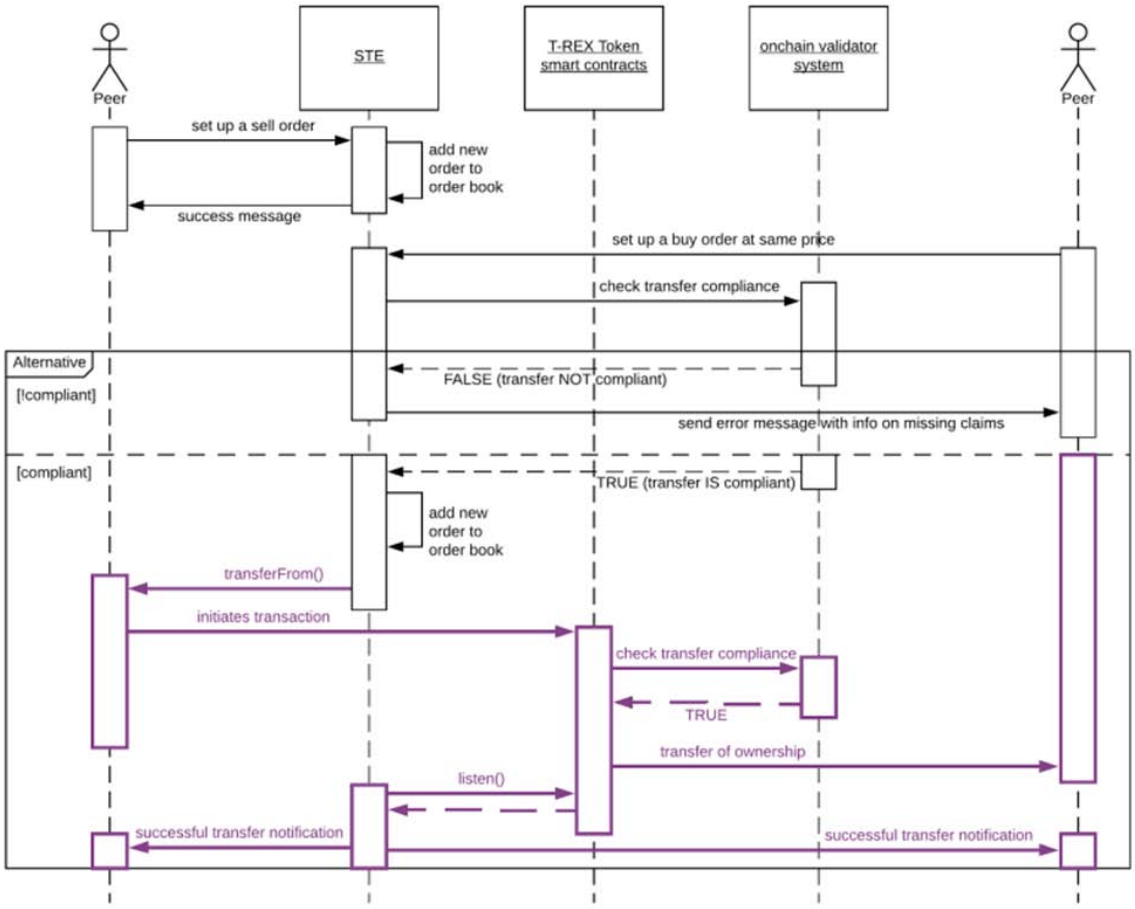


The steps for executing a DVD transfer are as follows:

1. Investors A and B agree on a transfer, with Investor A needing to know Investor B's wallet address.
2. Investor A authorizes the DVD contract for Token A, the token they will send to Investor B in exchange for Token B. The granted allowance must be at least equal to the amount of Token A involved in the DVD transfer.
3. Similarly, Investor B authorizes the DVD contract for Token B, the token they will send to Investor A in exchange for Token A. Again, the granted allowance must be at least equal to the amount of Token B involved in the DVD transfer.
4. Investor A initiates the DVD transfer by calling the relevant function with the previously agreed parameters.
5. Investor B verifies the parameters set by Investor A against their preliminary agreement. If everything aligns, Investor B triggers the DVD transfer on the transfer ID created by Investor A by calling the relevant function.
6. The DVD smart contract completes the token distribution as per the DVD contract parameters in a single transaction. This ensures that if either investor fails to provide the promised tokens, the entire transaction is reversed, preventing any loss of tokens.

Decentralized Exchanges with Off-Chain Order Book

Certain exchanges, such as those based on protocols like IDEX or dYdX , allow direct address management to facilitate distributed transfers. These Decentralized Security Token Exchanges (STEs) can interact with the T-REX identity management and transfer protocol by adhering to the following process:



1. A token holder logs into the STE and places a sell order.
2. The STE verifies the seller's balance and allowance to ensure they possess enough tokens to fulfill the order and confirms the exchange has permission to access the tokens for settlement.
3. If the seller's balance and allowance meet the necessary criteria, the order is added to the STE's order book, and a success notification is sent to the order issuer. If the balance or allowance is insufficient, the order is rejected, and an error message is returned to the order issuer.
4. A prospective buyer places a buy order on the same token at a price equal to or higher than the existing sell order. The buyer must have sufficient balance and allowance on the token used for the purchase to add the order to the order book.
5. The STE checks the buyer's identity contract to confirm if they hold the necessary claims to receive the token being traded. It also verifies if the potential transfer aligns with the compliance rules stipulated in the compliance contract. This compliance check is performed using the same method and code used to verify compliance before executing a standard transfer.
6. The identity registry and the compliance module return the buyer's status. If the necessary claims are lacking, or if the potential transfer violates the compliance rules, the order is canceled, and an error notification detailing the requirements for compliance is sent to the buyer. Depending on the reason for the failure, the buyer might be able to rectify the situation.

7. If the buyer has the necessary claims and the compliance rules are not violated, the buy order is added to the order book. The STE then initiates the transactions using the transferFrom() function. This function is invoked on the Token contract and checks the receiver's claims, similar to a standard P2P T-REX transaction. If all conditions are met (which should be the case, as the STE pre-checked the claims before adding the orders to the order book), the transfer is executed. While the process may vary slightly depending on the STE protocol used, this description represents the most typical scenario.
8. The STE monitors the blockchain, and once the transaction is successfully included in a block, it sends a success message to the transaction counterparties.

Through this process, the protocol ensures only trades validated by the decentralized validator are permitted on such exchanges. It also demonstrates how these exchanges can expand the system's reach by interacting with non-compliant peers. These peers receive customized error messages with step-by-step instructions on how to achieve compliance. The purple section in the diagram represents the on-chain processing, while the rest of the process is handled off-chain.

https://dydx.exchange/
https://idex.io/

Decentralized Exchange - Automated Market Makers

Automated Market Makers (AMMs) represent the most prevalent type of Decentralized Exchanges at present. A prominent example of this standard is the Uniswap protocol. These exchanges facilitate instant liquidity, enabling token holders to "swap" their tokens for other tokens. The transaction quantity and value are dictated by the swap volume, the swap ratio versus the desired tokens in the available Liquidity Pools (LPs), and the token availability within these LPs. As the quantity of tokens to be swapped grows relative to the size of the Liquidity Pool, the swap rate becomes less favorable.

For T-REX tokens to be swappable on an AMM, the Liquidity Pool contracts, through which the T-REX tokens transit during the swap process, need to be eligible (associated with an eligible OnChain ID holding the required claims) and compliant with the rules defined in the Compliance contract. Liquidity Pool addresses need to be added to the Identity Registry Storage, giving the Token Issuer and their agent(s) the authority to approve or reject an exchange pair. Once a pair receives approval, Liquidity providers can begin depositing tokens into the LP to start earning transaction fees on each swap.

The process for an investor aiming to buy Security Tokens would unfold as follows:

1. The investor navigates to the DEX platform and connects their wallet.
2. The investor chooses the two tokens to be swapped.
3. If one of these tokens is an ERC-3643 token, the DEX initiates the necessary functions to verify if the swap can occur on-chain (verifying eligibility with an isVerified() function call, ensuring compliance with a canTransfer() function call, checking the freezing status, and confirming the token is not paused).
4. If any of these checks fail, the DEX should display the reason and guide the user on how to rectify the issue (if feasible).
5. If all checks are successful, the DEX should permit the user to sign the swap transaction and broadcast it to the blockchain.
6. Once confirmed on-chain, the swap proceeds and tokens are distributed accordingly.

Centralized Exchange (CEX) - Investor Owned Wallet

In the context of centralized exchanges that utilize one wallet per investor, with the private keys held by the CEX, the deposit process requires a special approach. Wallets must be linked to the OnChain ID of the exchange, enabling the tagging of wallets as exchange wallets on the compliance contract. This arrangement facilitates user-specific tracking of deposits by verifying the OnChain ID of the depositor during a transfer from a wallet tied to an investor OnChain ID to a CEX OnChain ID.

For example, this procedure can set monthly limits on deposits and withdrawals on CEX per investor. Prior to the deposit taking place, the investor's CEX wallet must be registered in the Token's Identity Registry and linked to the OnChain ID of the CEX to facilitate transfers.

Here's the sequence of events for depositing tokens to a CEX:

1. An investor links their CEX wallet to the OnChain ID of the exchange. This step is crucial for ensuring that the exchange can track the wallet as part of its operations and be able to enforce any necessary compliance rules.
2. The linked wallet is then registered in the Identity Registry of the Token by a token agent. This registration ensures that the wallet is recognized by the token's system and can interact with it.
3. Once the registration is complete, the investor can initiate the deposit. The system checks the OnChain ID of the depositor to validate the transfer from the investor's wallet to the CEX OnChain ID.
4. Depending on the compliance rules set on the contract, certain restrictions may be enforced. For instance, monthly limits on deposits and withdrawals per investor can be implemented.
5. The transfer of tokens is then completed, and the investor's tokens are deposited into the CEX.

However, it's important to note that the activities within the CEX cannot be tracked and validated by the Eligibility Verification System (EVS) or by compliance measures. Therefore, the CEX has the responsibility to ensure the compliance and eligibility of investors.

In order to provide a complete cap table, the CEX should also supply data about token holders to the token issuer. This information exchange is crucial for maintaining transparency and ensuring that all parties involved have a comprehensive understanding of the token distribution. In the context of centralized exchanges that utilize one wallet per investor, with the private keys held by the CEX, the deposit process requires a special approach. Wallets must be linked to the OnChain ID of the exchange, enabling the tagging of wallets as exchange wallets on the compliance contract. This arrangement facilitates user-specific tracking of deposits by verifying the OnChain ID of the depositor during a transfer from a wallet tied to an investor OnChain ID to a CEX OnChain ID.

For example, this procedure can set monthly limits on deposits and withdrawals on CEX per investor. Prior to the deposit taking place, the investor's CEX wallet must be registered in the Token's Identity Registry and linked to the OnChain ID of the CEX to facilitate transfers. Here's the sequence of events for depositing tokens to a CEX:

6. An investor links their CEX wallet to the OnChain ID of the exchange. This step is crucial for ensuring that the exchange can track the wallet as part of its operations and be able to enforce any necessary compliance rules.
7. The linked wallet is then registered in the Identity Registry of the Token by a token agent. This registration ensures that the wallet is recognized by the token's system and can interact with it.
8. Once the registration is complete, the investor can initiate the deposit. The system checks the OnChain ID of the depositor to validate the transfer from the investor's wallet to the CEX OnChain ID.

9. Depending on the compliance rules set on the contract, certain restrictions may be enforced. For instance, monthly limits on deposits and withdrawals per investor can be implemented.
10. The transfer of tokens is then completed, and the investor's tokens are deposited into the CEX.

However, it's important to note that the activities within the CEX cannot be tracked and validated by the Eligibility Verification System (EVS) or by compliance measures. Therefore, the CEX has the responsibility to ensure the compliance and eligibility of investors.

In order to provide a complete cap table, the CEX should also supply data about token holders to the token issuer. This information exchange is crucial for maintaining transparency and ensuring that all parties involved have a comprehensive understanding of the token distribution.

Centralized Exchange - Pooled Wallets

Unlike the case with investor-owned wallets, some centralized exchanges operate on a model where investors deposit their tokens into a few pooled wallets managed by the exchange. In this model, the CEX must have a mechanism to reconcile deposits with the corresponding user accounts, as the sender address is key to associating deposits with the correct users.

Here is the sequence of events for depositing tokens to a CEX using pooled wallets:

1. An investor initiates a deposit to one of the pooled wallets managed by the CEX. As these wallets are already registered in the Token's Identity Registry and linked to the OnChain ID of the CEX, no additional linkage is necessary on the part of the investor.
2. The system checks the sender address of the deposit. This step is critical, as the sender address is used to reconcile the deposit with the correct user account on the CEX.
3. As with the investor-owned wallet model, compliance rules set on the contract can enforce certain restrictions. For instance, monthly deposit and withdrawal limits per investor can be implemented.
4. Once the sender address is verified and any compliance rules are satisfied, the transfer of tokens is completed, and the investor's tokens are deposited into the pooled CEX wallet.

Despite the difference in wallet management, the same challenge exists with regard to the cap table management for the token issuer. Since the on-chain cap table only recognizes the pooled wallets of the CEX, the internal distribution of tokens among individual investors within the CEX remains invisible on-chain. Consequently, it's the CEX's responsibility to ensure the compliance and eligibility of its users.

To maintain a complete and accurate cap table, the CEX must supply data about token ownership within the exchange to the token issuer. This exchange of information is vital for maintaining transparency and ensuring all parties involved have a comprehensive understanding of the token distribution.

Conclusion

The advent of blockchain technology has opened up new possibilities for the financial markets, particularly in the realm of asset tokenization. The T-REX protocol, as outlined in this whitepaper, represents a significant step forward in this direction. By enabling compliant issuance and management of permissioned tokens, T-REX brings unprecedented efficiency, accessibility, and liquidity to the market.

However, the journey towards widespread adoption of tokenized securities is not without challenges. Regulatory compliance, identity management, and ensuring the rights and security of all stakeholders are critical issues that need to be addressed. The T-REX protocol, with its open-source ERC3643 token standard and decentralized validation system, offers robust solutions to these challenges.

As we move forward, it is essential to continue refining and expanding these solutions, keeping pace with the evolving regulatory landscape and the needs of the market. The future of financial markets lies in harnessing the power of blockchain technology, and the T-REX protocol is poised to play a pivotal role in this transformation.

Monetaforge implements the ERC-3643 token Standard and leverages the Tokeny T-REX protocol and Tokeny platform in the token architectural model and long term token administration.

Additional Information Resources

https://www.erc3643.org/

https://tokeny.com/