**ERC-3643**

**About ERC-3643**

The open-source ERC3643 token standard and its T-REX implementation were designed to address the need of 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.

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

* ONCHAINID , a blockchain **based identity management system**, allowing for the creation of a globally accessible identity for every stakeholder.
* A **set of validation certificates,** or verifiable credentials, (technically speaking, these certificates are the claims, described in the ERC-735 standard used by ONCHAINID, that will be described further in the document.
* An **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.
* A set of **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.

These 4 key elements allow issuers to use a **decentralized Validator to control transfers and enforce compliance on the holders of the security token**.

**Transactions involving T-REX compliant ERC-20 permissioned tokens**

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 ONCHAINID.
3. Compliance and Eligibility Checks: The validator ensures that the receiver's ONCHAINID 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 the rules set forth in the compliance smart contract.
4. Evaluation of Transfer: If the receiver's ONCHAINID 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 ONCHAINID 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

**Main T-REX components**

* **ONCHAINID**
* **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 stipulateidentity requirements and executes the "isVerified" function. The Identity Registry contract is linked to the Identity Registry Storage contract, which houses a dynamic "whitelist" of identities. 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.
* **Identity Registry Storage:** The Identity Registry Storage contract functions as a repository for a mapping table, associating wallet addresses with the corresponding ONCHAINID addresses of investors.
* **Trusted Issuers Registry:** The Trusted Issuers Registry smart contract serves as a storage system for contract addresses (ONCHAINIDs) 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.
* **Claim Topics Registry:** The Claim Topics Registry smart contract houses all the claim topics necessary for holding the security token. The ONCHAINID 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.
* **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.
* **Modular Compliance:** The Compliance smart contract is a dynamic tool used to establish the rules of the token offering. 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
* **Implementation Authority** operates in two distinct capacities:
  + **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.
  + **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
* **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 issue. All contracts deployed by the factory are in the form of proxies, pointing to the main Implementation Authority

**Key Functions of ERC-3643 Tokens**

* **Asset Representation**: ERC-3643 tokens represent assets like real estate, art, and commodities, containing important data on asset ownership details, legal conditions, and compliance.
* **Interoperability**: A significant feature of ERC-3643 tokens is their ability to integrate with various popular DeFi protocols. Interoperability means these tokens will effectively be incorporated into multiple DeFi applications, enhancing the value offered by the broader blockchain environment.
* **Programmable Protocols**: ERC-3643 tokens utilize Ethereum’s smart contracts through programmable protocols, enabling self-executing and configurable protocols related to the transfer, distribution, and governance of tokenized assets.
* **Legal Compliance**: Regulatory compliance is integrated into ERC-3643, addressing fundamental issues raised by the tokenization of real-world assets. This standard creates channels to adhere to jurisdictional rules and provides clear guidelines on asset tokenization, avoiding legal infringements.
* **Immutable Ownership Records**: Ownership of ERC-3643 tokens is stored on the blockchain, giving them immutable and highly transparent characteristics. This immutable ledger provides a verifiable and reliable custodial blockchain, making tokenized assets more secure.
* **Upgradability**: The design of ERC-3643 is upgradable, eliminating the need to migrate to another smart contract. It allows the system enough flexibility to evolve with the dynamic environment of technology and policies. To qualify for holding the token, the ONCHAINID 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.

The management of compliant transactions through T-REX backed permission tokens is based on four main pillars creating a decentralized Validator:

1- ONCHAINID, a blockchain-based identity management system, allows for creating a globally accessible identity for every stakeholder.

2- A set of validation certificates; technically speaking, these certificates are the claims described in the ERC-734 and ERC-735 standards used by ONCHAINID, which will be described further in the document.

3- A transfer manager whose role is to act as a filter for all the transactions of tokenized securities and will check the validation certificates of the stakeholders. Essentially, the transfer manager will check that the receiver has the right to receive the tokens following the specific compliance rules and issuer requirements applicable for this specific asset. The transfer manager will block the transaction if the receiver misses a mandatory certificate and will notify him about the reason for the failure.

4- A set of Compliance rules ensuring that the rules of the offering are respected, e.g., the maximum number of investors per country of distribution, the maximum number of tokens held by a single investor, and more. These rules are linked to the identity of the receiver of a transaction and the global distribution of tokens at a certain time.

These four key elements allow issuers to use a decentralized validator to control transfers and enforce compliance on the security token holders. The Validator includes rules for the whole offering, such as managing the maximum number of holders allowed in a specific market.

**Integration of ERC-3643 standard (T-REX) and buildchain use cases**

**Usecases of T-REX implementation**

The T-REX implementation offers a comprehensive suite for managing security tokens on blockchain platforms, focusing on compliance and identity management. It includes various operations and token transactions such as creation of identities on the blockchain, compliant transfer of ownership, stakeholder management, and integration with exchanges (both centralized and decentralized). The system ensures that token transfers meet all necessary criteria, including KYC and AML compliance, through an on-chain validation process. The tokens are based on the ERC20 standard, but we could extend the contract to include safe minting and transfers of NFTs

**Beginning the project**

When project manager or owner is choosing contractors, the use of the decentralized identity can be used, checking credentials of contractor applicants. The chosen contractor will recieve a generated token/NFT that would grant him access to managing the building during the build process. Or contractor could recieve a valid credential, showing he is working on this project.

**Design phase**

When designing the building making changes to the model would reqire a certain verifiable credential.

**Sustainability compliance**

Sustainability compliance officer can be assigned the role of trusted issuer, where he could issue a credential that the building plan is compliant. The credential would be valid for a certain amount of time. Another special use case here would be that if the credential isnt renewed for a building that is creating revenue (museum, hotel, …) they cant except any more tokens.

**Tokenization of building assets**

By extending these contracts to include ERC721 tokens, we can tokenize certain assets of the buildings and asign them to certain contractors, to do work on them. Eg windows can be tokenized and minted/transfered to a contractor, that has a certain credential attached to his identity. Transfer of this token is then limited to the defined rules.

A diagram of a diagram

Description automatically generated with medium confidence