



ENTERPRISE
ETHEREUM
ALLIANCE

Digital Asset Tokenization: Technical framework and Considerations

Presented by: Daniel Norkin

Co-chair EEA Tokenization IG | Cofounder of Envision Blockchain

Agenda

- Introduction to the EEA Tokenization Interest Group
- Types of Tokens
- Technical Meets Business Rules

About Me

Daniel Norkin

EEA Tokenization Co-Chair / Hedera Partner / ConsenSys Partner / Cofounder of Envision Blockchain

A thought leader and pioneer in innovative technology, Daniel Norkin has built his reputation as someone who keeps his projects cutting edge. He co-founded Envision Blockchain Solutions, a professional services company that builds applications in the Web3 space. By adopting transformative technologies Envision Blockchain has helped organizations set themselves up for long term success.

Introduction to the EEA Tokenization IG

Our Focus

The EEA Tokenization Interest Group discusses issues and industry needs around tokenization using Ethereum technology.

The Group is analyzing gaps in available technology, and areas where it seems unnecessarily difficult today to implement use cases successfully, whether the root cause is related to ease-of-use, regulation complexity, a technology gap or some other issue.

The group is also collecting information on EEA members who offer commercial services around Tokenization, in order to make the information available to other EEA members and the general public.

In addition, the EEA's Tokenization Interest Group has started its Tokenization Webinar Series for members and non-members alike. The Webinar series features five presentations; (1) Legal/Regulatory Framework; (2) Technical Framework; (3) Token Economics Framework; (4) Ethical Framework; and (5) Case Studies for Digital Asset Tokenization.

Introduction to the EEA Tokenization IG - Webinar Series

- Webinar 1: U.S. Securities Laws, Rules, and Regulations (Benjamin Bukari, Co-Chair, EEA Tokenization Interest Group finished)
- [Webinar 2: Technical framework and Considerations](#)
- Webinar 3: Tokenomics Framework
- Webinar 4: Ethical Framework
- Webinar 5: Industry Use Case (putting all of the webinar's learning together)

Types of Tokens - Overview

01

Fungible Tokens

- Fungible tokens (fractional) have interchangeable value with each other, where any owned sum of them from a class has the same value as another owned sum from the same class.
- Example: Fiat or Cryptocurrency
- Can be whole, fractional, etc...but the definition changes accordingly

02

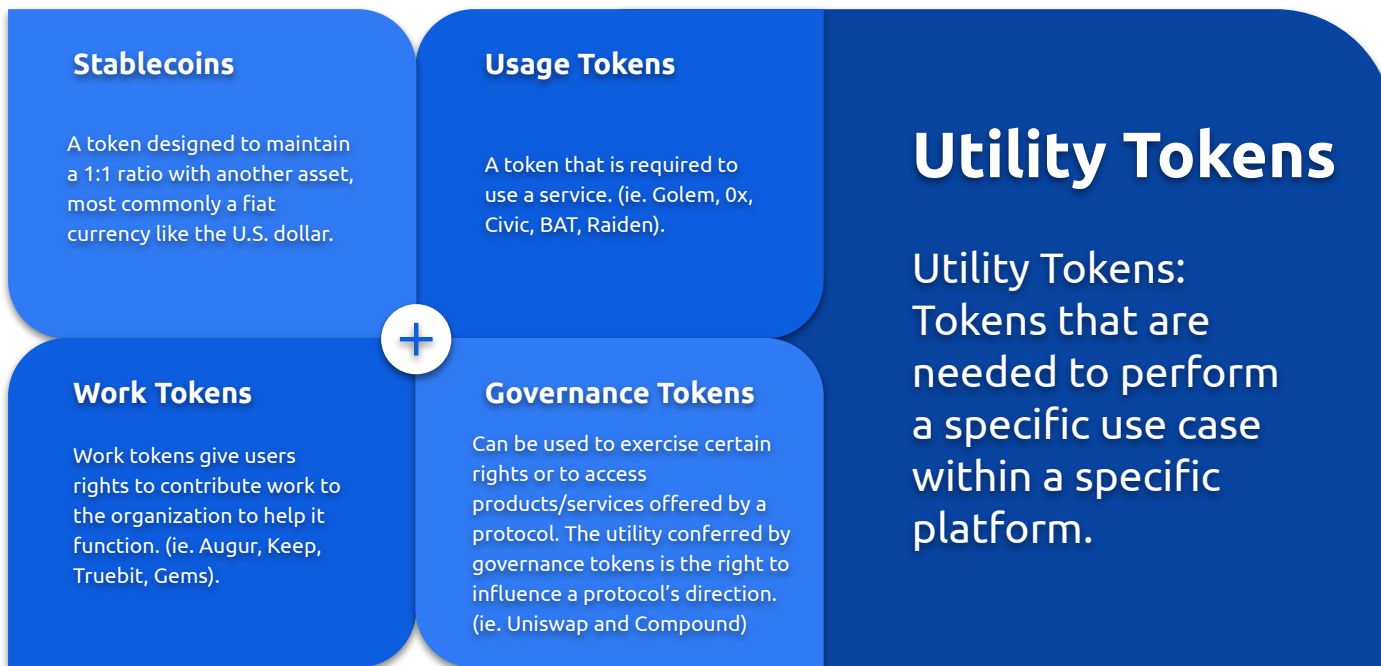
Non Fungible Tokens

- A restriction on the token in that there can only be 1 whole token (singleton) in the class and is not divisible. This behavior is only available to non-fungible base types.
- Example: Original Art
- Can be whole, singleton, fractional, etc...but the definition changes accordingly

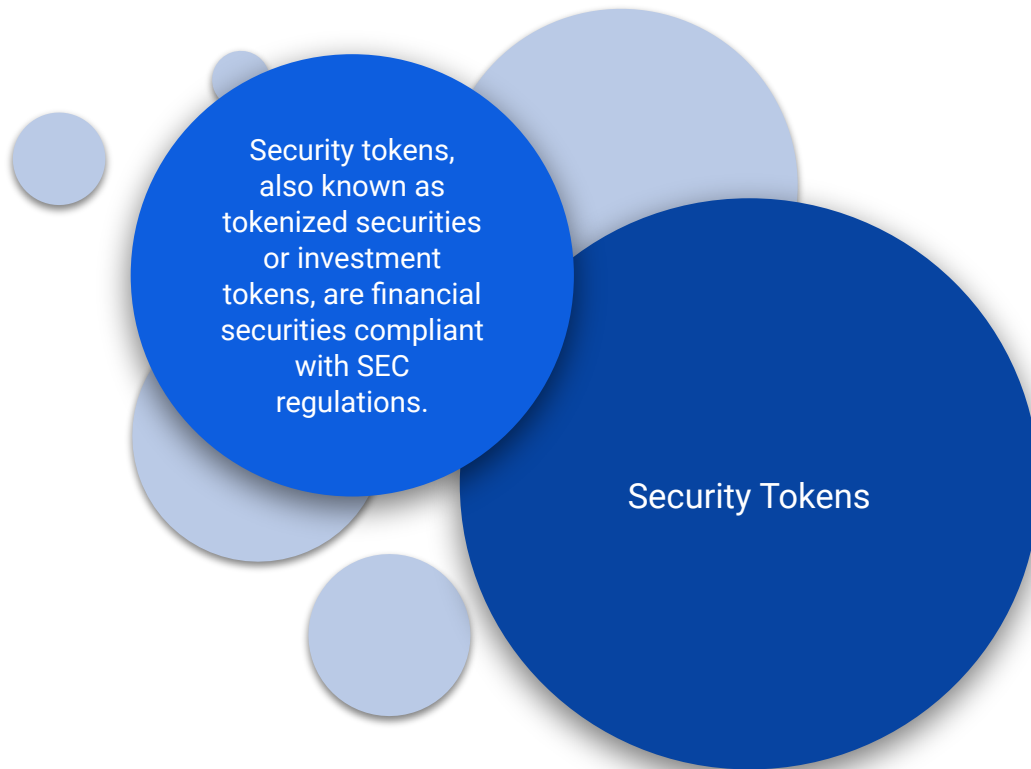
Types of Tokens - Fungible Tokens



Types of Tokens - Utility Tokens



Types of Tokens - Security Tokens



Technical Meets Business Rules - Standards

Ethereum Improvement Proposals (EIPs) describe standards for the Ethereum platform, including core protocol specifications, client APIs, and contract standards.

To help prevent mistakes and defragmentation, one of the many Ethereum development standards focuses on token interfaces. These standards help ensure smart contracts remain composable, so for instance when a new project issues a token, that it remains compatible with existing decentralized exchanges.

Standard
Track

Core

Networking

Interface

ERC

Meta

Informational

Technical Meets Business Rules - Most Common Tokens

01

ERC 20

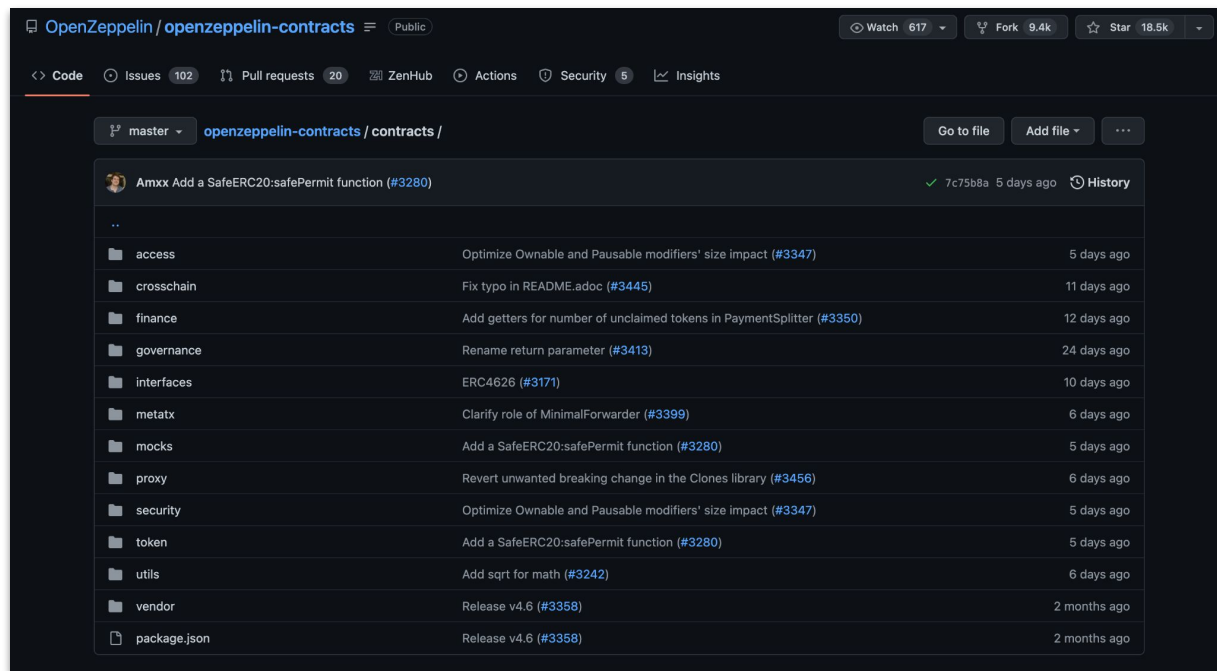
- Fungible

02

ERC 721

- Non Fungible

Technical Meets Business Rules - Where are the smart contracts?



The screenshot shows the GitHub repository for OpenZeppelin/openzeppelin-contracts. The repository is public and has 617 watches, 9.4k forks, and 18.5k stars. The current view is the 'contracts' directory on the master branch. A table lists the contents of the directory, including subdirectories and files, along with their commit hashes and the time since the last update.

File/Directory	Commit Hash	Time Ago
access	Optimize Ownable and Pausable modifiers' size impact (#3347)	5 days ago
crosschain	Fix typo in README.adoc (#3445)	11 days ago
finance	Add getters for number of unclaimed tokens in PaymentSplitter (#3350)	12 days ago
governance	Rename return parameter (#3413)	24 days ago
interfaces	ERC4626 (#3171)	10 days ago
metatx	Clarify role of MinimalForwarder (#3399)	6 days ago
mocks	Add a SafeERC20:safePermit function (#3280)	5 days ago
proxy	Revert unwanted breaking change in the Clones library (#3456)	6 days ago
security	Optimize Ownable and Pausable modifiers' size impact (#3347)	5 days ago
token	Add a SafeERC20:safePermit function (#3280)	5 days ago
utils	Add sqrt for math (#3242)	6 days ago
vendor	Release v4.6 (#3358)	2 months ago
package.json	Release v4.6 (#3358)	2 months ago

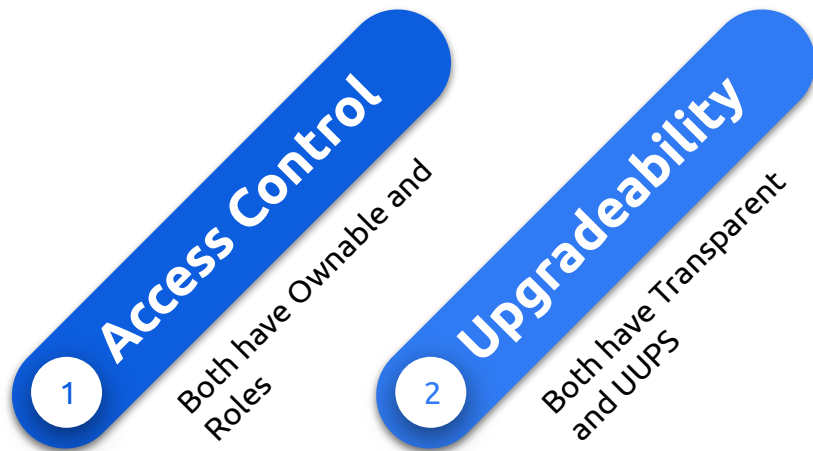
A library for secure smart contract development. Build on a solid foundation of community-vetted code.

- Implementations of standards like [ERC20](#) and [ERC721](#).
- Flexible [role-based permissioning](#) scheme.
- Reusable [Solidity components](#) to build custom contracts and complex decentralized systems.

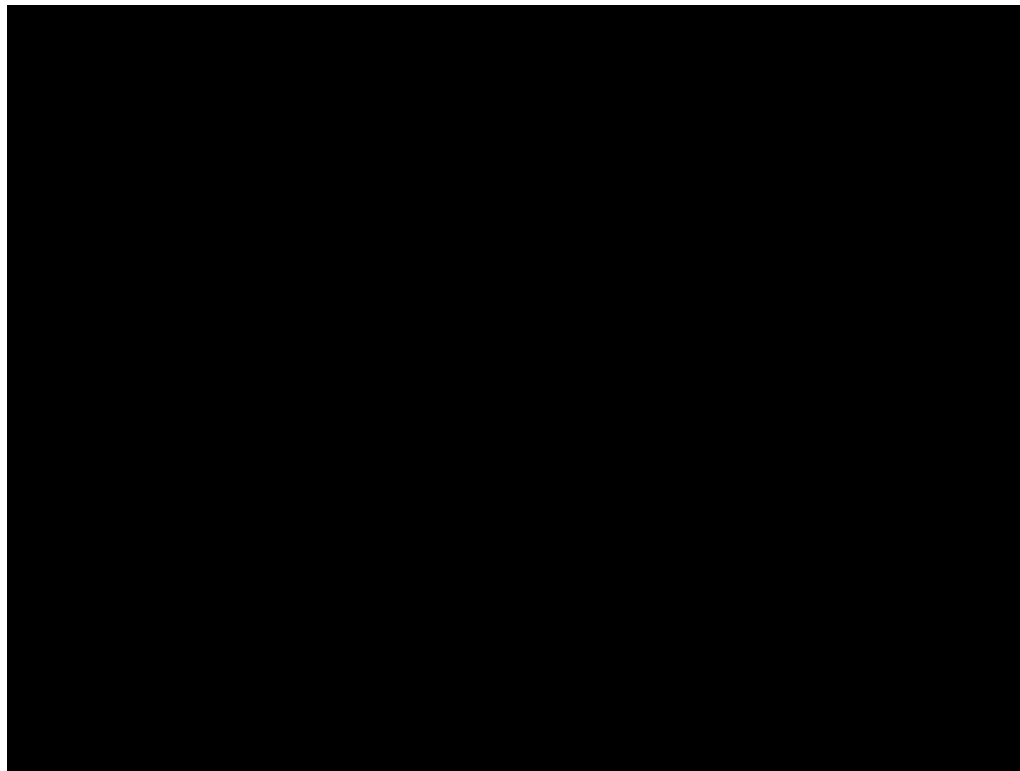
<https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/contracts>

Technical Meets Business Rules - ERC 20 vs ERC 721

Business Rules (Features)		ERC 20	ERC 721
1	Mintable	✓	✓
2	Pasuable	✓	✓
3	Burnable	✓	✓
4	Permit	✓	✓
5	Vote	✓	✓
6	Auto Increment ID	✓	✗
7	Enumerable	✓	✗
8	URI Storage	✓	✗



Technical Meets Business Rules - Creating Smart Contracts



Technical Meets Business Rules - Extending Smart Contracts

What happens if you want to use an existing token standard...but need more sauce?

Example: EIP 4910 Draft implemented in treetrunk.io

Extension of the **ERC-721** standard for NFTs to correctly define, process, and pay (hierarchical) onchain royalties from NFT sales, going beyond EIP-2981.

Authored by Dr. Andreas Freund, "The proposal builds upon the **OpenZeppelin Smart Contract** Toolbox architecture, and extends it to include royalty account management (CRUD), royalty balance and payments management, simple trading capabilities -- Listing/De-Listing/Buying -- and capabilities to trace trading on exchanges. The royalty management capabilities allow for hierarchical royalty structures, referred to herein as royalty trees, to be established by logically connecting a "parent" NFT to its "children", and recursively enabling NFT "children" to have more children."

<https://github.com/Therecanbeonlyone1969/EIPs/blob/patch-1/EIPS/eip-4910.md>

Technical Meets Business Rules - EIP 4910 Draft (New Concepts)

- **Royalty Account (RA)**
 - A Royalty Account is attached to each NFT through its tokenId and consists of several sub-accounts which can be accounts of individuals or other RAs. A Royalty Account is identified by an account identifier.
- **Account Type**
 - This specifies if an RA Sub Account belongs to an individual (user) or is another RA. If there is another RA as an RA Sub Account, the allocated balance needs to be reallocated to the Sub Accounts making up the referenced RA.
- **Royalty Split**
 - The percentage each Sub Account receives based on a sale of an NFT that is associated with an RA
- **Royalty Balance**
 - The royalty balance associated with an RA
- **Sub Account Royalty Balance**
 - The royalty balance associated to each RA Sub Account. Note that only individual accounts can carry a balance that can be paid out. That means that if an RA Sub Account is an RA, its final Sub Account balance must be zero, since all RA balances must be allocated to individual accounts.
- **Token Type**
 - Token Type is given as either ETH or the symbol of the supported ERC 20/223/777 tokens such as DAI
- **Asset ID**
 - This is the tokenId the RA belongs to.
- **Parent**
 - This indicates which tokenId is the immediate parent of the tokenId to which an RA belongs.

Thank You!

- Feel free to submit questions to daniel.norkin@envisionblockchain.com
- Stay tuned for the EEA's Tokenization IG Webinar Series
 - Webinar 1: U.S. Securities Laws, Rules, and Regulations (finished)
 - [Webinar 2: Technical framework and Considerations \(finished\)](#)
 - Webinar 3: Tokenomics Framework
 - Webinar 4: Ethical Framework
 - Webinar 5: Industry Use Case (putting all of the webinar's learning together)

Get Involved!

The EEA welcomes additional input on this important work and participation in the Tokenization Interest Group. Here's more about who should join and how to get involved!

Non-EEA Members:

If you are interested in the work of the EEA Tokenization IG and would like to contribute, please contact james.harsh@entethalliance.org and visit the EEA Tokenization IG page: <https://entethalliance.github.io/tokenization>

EEA Members:

Please contact the EEA Secretariat or your member council representative. You can also log into the Member Collaboration site and join the working group to receive future meeting invitations.

Learn more about becoming an EEA Member and be sure to follow us on Twitter, LinkedIn, and Facebook for all the latest.