CET

Contributors

Name	Organization
Marley Gray	Microsoft
Jason Pancis	Envision Blockchain

Taxonomy Formula: tF’{d,t,m,r}+phEGO

Token Specification Summary

Token Classification

Template Type:	**SingleToken**	**This token has no sub or child tokens.**
Token Type:	Fungible	Tokens have interchangeable value with one another, where any quantity of them has the same value as another equal quantity if they are in the same class or series.
Token Unit:	Fractional	This token can be sub-divided or split into smaller units or parts based on a certain number of decimal places.
Value Type:	Intrinsic	This token is purely a digital token represents value directly, it represents no external physical form and cannot be a receipt or title for a material item or property.
Representation Type:	Unique	Token instances are unique having their own identities and can be individually traced. Each unique token can carry unique properties that cannot be changed in one place and their balances must be summed. These are like bank notes, paper bills and metal coins, they are interchangeable but have unique properties like a serial number.
Supply:	Infinite	Infinite supply indicates that tokens in the class can be created and removed with no cap and also potentially reflect negative supply for certain business cases.

This a reference token which represents a specified volume of metric tons of green house gas (GHG) emissions. This token is fractional or divisible up to 8 decimal places. It is a unique token so that it may be issued in quantities of a fraction or numbers larger than 1. Meaning, if in the production of an item that a supplier creates generates 1.242 tons of carbon a token can be created with the same amount. This token also has the Carbon Emission Generator Object property that allows the creator of the token to specify what item the carbon is related to, which is helpful for track and trace supply chain requirements.

Example

CET tokens are useful for tracking carbon emissions within a supply chain as it allow for CETs to follow items through the supply chain between parties. This allows for carbon tracing for commodities and an enables accounting for total carbon emissions for finished goods to be calculated and traced to sources.

Analogies

Name	Description
Carbon Emissions Token	A token created by a supplier of goods that generates carbon emissions.

CET is:

- Divisible
- Transferable
- Mintable
- Roles

CET Details

Unique Fractional Fungible

Type:	**Base**
Name:	Unique Fractional Fungible
Id:	3e05130c-969a-4dfc-abe6-c83fad98a4ec
Visual:	&tau;<sub>F'</sub>{<i>d</i>}
Tooling:	tF'{d}

Version:	1.0

Definition

Unique, fractional fungible tokens 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. Similar to physical cash money, a crypto-currency is an example of a fungible token that is divisible. Because this token is unique, it will have its own identity and can have unique properties like a serial number. Implementations should support a GetBalance or List for owners to see their balances or tokens they own.

Example

Fiat currency is the most widely understood example of a fractional fungible item. A fractional fungible is divisible, so you can 'make change'.

Analogies

Name	Description
Physical Money or Cash	Cash, or fiat money, is freely accepted between parties and can have varying denominations. Money has a face value, on a coin or bill, and can be summed together to represent higher value. It can be divided, making change, and consolidated from many smaller denominations to larger ones and still have the same value.
General Admission Movie Ticket	Purchasing a general admission ticket to a movie only allows for you to have a seat, but the seat that you actually get depends on factors like when you arrive.

Dependencies

Artifact Type	Symbol	Description
Base	t	Base Token Definition

Incompatible With

Artifact Type	Symbol	Id
Behavior	~d	d5807a8e-879b-4885-95fa-f09ba2a22172

CET - 56368dff935dfdc5bd365f6984c802674c394eed6d99b3c0a668321e9cb8f1ca3

Influenced By

Description	Symbol	Applies To

Artifact Files

Content Type	File Name	File Content
Control	fractional-fungible.proto	
Uml	fractional-fungible.md	

Code Map

Map Type	Name	Platform	Location

Implementation Map

Map Type	Name	Platform	Location

Resource Map

Map Type	Name	Location	Description

Base Details

Token Name:	CET
Token Type:	Fungible
Representation Type:	Unique
Value Type:	Intrinsic
Token Unit:	Fractional
Symbol:	CET

Owner:	
Quantity:	0
Decimals:	8
Constructor Name:	Constructor

Behaviors

Specification Behavior

Divisible

Taxonomy Symbol: d

> *An ability for the token to be divided from a single whole token into fractions, which are represented as decimal places. Any value greater than 0 will indicate how many fractions are possible where the smallest fraction is also the smallest ownable unit of the token.*

Example

Divisible is common for crypto-currencies or tokens of fiat currency. For example, the US Dollar is divisible to 2 decimal places, where a value like .42 is possible. Bitcoin, is divisible up to 8 decimal places.

Analogies

Name	Description
Analogy 1	divisible analogy 1 description

Type:	**Behavior**
Name:	Divisible
Id:	6e3501dc-5800-4c71-b59e-ad11418a998c
Visual:	<i>d</i>
Tooling:	d
Version:	1.0

Dependencies

Artifact Type	Symbol	Description

Incompatible With

Artifact Type	Symbol	Id
Behavior	~d	d5807a8e-879b-4885-95fa-f09ba2a22172

Influenced By

Description	Symbol	Applies To

Artifact Files

Content Type	File Name	File Content
Control	divisible.proto	
Uml	divisible.md	

Code Map

Map Type	Name	Platform	Location

Implementation Map

Map Type	Name	Platform	Location

Resource Map

Map Type	Name	Location	Description

Is External:	True

CET - 56368dff935dfdc5bd365f6984c802674c394eed6d99b3c0a668321e9cb8f1ca6

Constructor:

Divisible responds to these Invocations

Properties

Name: Decimals

Value Description: Set to a number greater than Zero, allowing subdivision

Template Value: 8

Invocations

GetDecimals

Id: 01f7ef04-1215-45f1-b118-12b4a76db9ad

Description: Return the value

Request

Control Message: GetDecimalsRequest

Description:

Parameters

Name	Value

Response

Control Message: GetDecimalsResponse

Description: Return number of decimal places

Parameters

Name	Value
Decimals	integer

GetDecimals

Id: 01f7ef04-1215-45f1-b118-12b4a76db9ad

Description: Return the value

CET - 56368dff935dfdc5bd365f6984c802674c394eed6d99b3c0a668321e9cb8f1ca7

Request

Control Message: GetDecimalsRequest

Description:

Parameters

Name	Value

Response

Control Message: GetDecimalsResponse

Description: Return number of decimal places

Parameters

Name	Value
Decimals	integer

Specification Behavior

Transferable

Taxonomy Symbol: t

> *Every token instance has an owner. The Transferable behavior provides the owner the ability to transfer the ownership to another party or account. This behavior is often inferred by other behaviors that might exist like Redeem, Sell, etc. This behavior is Delegable. If the token definition is Delegable, TransferFrom will be available.*

Example

Analogies

Name	Description
Analogy 1	transferable analogy 1 description

Type:	**Behavior**
Name:	Transferable
Id:	af119e58-6d84-4ca6-9656-75e8d312f038
Visual:	<i>t</i>
Tooling:	t
Version:	1.0

Dependencies

Artifact Type	Symbol	Description

Incompatible With

Artifact Type	Symbol	Id
Behavior	~t	a4fa4ca8-6afd-452b-91f5-7103b6fee5e5

Influenced By

Description	Symbol	Applies To
If the token is Delegable, TransferFrom should be enabled.	g	[ ]
If Compliance is present, a CheckTransferAllowed request has to be made and verified before a Transfer request or a TransferFrom request.	c	[ ]
If issuable is present, an AcceptTokenRequest from the token issuer, in response to a RequestTokens, has to be made and verified before a Transfer request.	i	[ ]

Artifact Files

Content Type	File Name	File Content
Control	transferable.proto	

CET - 56368dff935dfdc5bd365f6984c802674c394eed6d99b3c0a668321e9cb8f1ca9

Uml	transferable.md	

Code Map

Map Type	Name	Platform	Location

Implementation Map

Map Type	Name	Platform	Location

Resource Map

Map Type	Name	Location	Description

Is External:	**True**
Constructor:	

Transferable responds to these Invocations

Transfer

Id: 5d4b8f10-7857-4a2f-9b8c-d61e367a6bcc

Description: >A transfer request will invoke a transfer from the owner of the token to the party or account provided in the To field of the request. For fungible or dividable non-fungible tokens, this request may also include value in the Amount field of the request to transfer more than one token of the class in a single request.

Request Message:

TransferRequest

Description: The request

Request Parameters

Name	Value
To	AccountId to transfer ownership to.
Quantity	Number of tokens to transfer.

Response Message

TransferResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the owner based on the outcome of the transfer request.

TransferFrom

Id: 516b4e2f-4a14-4c4f-a6f2-1419d4af35c6

Description: >A transfer request will invoke a transfer from the owner of the token to the party or account provided in the To field of the request. For fungible or dividable non-fungible tokens, this request may also include value in the Amount field of the request to transfer more than one token of the class in a single request.

Request Message:

TransferFromRequest

Description: The request

Request Parameters

Name	Value
From	AccountId to transfer ownership from.
To	AccountId to transfer ownership to.
Quantity	Number of tokens to transfer.

Response Message

TransferFromResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the owner based on the outcome of the transfer from request.

Specification Behavior

Mintable

Taxonomy Symbol: m

A token class that implements this behavior will support the minting or issuing of new token instances in the class. These new tokens can be minted and belong to the owner or minted to another account. This behavior may be invalidated by a restrictive behavior like Singleton, where only a single instance of the token can exist. Mintable is technically delegable, but it's delegation should be controlled by a behavior like Roles.

Example

A consortium of oil producers needs to create tokens for each barrel of oil they are putting on the market to trade. There are separate classes of tokens for each grade of oil. Producers of barrels will need be have the ability to mint new tokens in order to facilitate the trading of them in the supply chain.

Analogies

Name	Description
SKU	A token class can represent a particular item SKU, where the manufacturer of the item has the ability to mint or issue new inventory of the SKU into the supply chain.

Type:	**Behavior**
Name:	Mintable

Id:	f9224e90-3cab-45bf-b5dc-0175121e2ead
Visual:	<i>m</i>
Tooling:	m
Version:	1.0

Dependencies

Artifact Type	Symbol	Description

Incompatible With

Artifact Type	Symbol	Id

Influenced By

Description	Symbol	Applies To
Roles is common to implement to provide authorization checks for invoking the behavior. Highly Recommended that Role restrictions be applied to MintTo invocations.	r	[ ]
If Compliance is present, a CheckMintAllowed request has to be made and verified before a Mint request or a MintTo request.	c	[ ]

Artifact Files

Content Type	File Name	File Content
Control	mintable.proto	
Uml	mintable.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	Open Zeppeli	EthereumSolidity	https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20Mintable.s

	n		ol

Implementation Map

Map Type	Name	Platform	Location
Implementation	Implementation 1	ChaincodeGo	

Resource Map

Map Type	Name	Location	Description
Resource	Regulation Reference 1		

Is External:	**True**
Constructor:	

Mintable responds to these Invocations

Mint

Id: 3ddf15db-c919-4f72-a57b-d089931bc901

Description: A request to create new token instances in the class by the owner or a party or account in a role that is granted this permission. Minted tokens using this invocation will be owned by the owner or token pool account. Requires a Quantity field in the request.

Request Message:

MintRequest

Description: The request

Request Parameters

Name	Value
Quantity	Number of new tokens to create.

Response Message

MintResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the invoker based on the outcome of the mint request.

MintTo

Id: 70499b23-a1dd-4c87-90d6-6e45400f28b5

Description: A request to create new token instances in the class by the owner or a party or account in a role that is granted this permission to another party or account. Requires a To and Quantity fields in the request.

Request Message:

MintToRequest

Description: The request

Request Parameters

Name	Value
ToAccount	Account Id to mint the tokens to.
Quantity	Number of new tokens to create.

Response Message

MintToResponse

Description: The response

Response Parameters

Name	Value
Confirmation	A confirmation receipt or error may be returned to the invoker based on the outcome of the MintTo request.

Specification Behavior

Roles

Taxonomy Symbol: r

> *A token can have behaviors that the class will restrict invocations to a select set of parties or accounts that are members of a role or group. This is a generic behavior that can apply to a token many times to represent many role definitions within the template. This behavior will allow you to define what role(s) to create and what behavior(s) to apply the role to in the TemplateDefinition.*

Example

Analogies

Name	Description
Minters	A role called 'Minters' for a token can have accounts in the role. The MintTo behavior invocation will be bound to the role check to ensure only account in the 'Minters' role are allowed to mint new instances in the class.

Comments

Roles has a constructor control that creates roles and applies them to certain behaviors of the token at creation of the class from the template.

Type:	**Behavior**
Name:	Roles
Id:	c32726da-9787-4dd8-8de3-d07d1733d0f6
Visual:	<i>r</i>
Tooling:	r
Version:	1.0

Dependencies

Artifact Type	Symbol	Description

Incompatible With

Artifact Type	Symbol	Id

Influenced By

Description	Symbol	Applies To

Artifact Files

Content Type	File Name	File Content
Control	roles.proto	
Uml	roles.md	

Code Map

Map Type	Name	Platform	Location
SourceCode	Open Zeppelin - Roles	EthereumSolidity	https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/Roles.sol

Implementation Map

Map Type	Name	Platform	Location

Resource Map

Map Type	Name	Location	Description

Is External:	**True**
Constructor:	

Roles responds to these Invocations

RoleCheck

Id: 00a665e3-1dda-441e-8262-5750435c153c

Description: Internal invocation when the applied behavior is called to check if the requestor is a member of the role.

Request Message:

IsInRole

Description: The request

Request Parameters

Name	Value
AccountId	AccountId of the requestor.

Response Message

True/False

Description: The response

Response Parameters

Name	Value
IsInRole	True/False

Properties

Name: Role

Value Description: A group or list an account can be a member or be in.

Template Value:

Invocations

GetRoleMembers

Id:

Description: Request the the list of member accounts in the role.

Request

Control Message: GetRoleMembersRequest

Description: The request

Parameters

Name	Value

Response

Control Message: GetRoleMembersResponse

Description: The response

Parameters

Name	Value
Members	Returning the list of accounts in the role.

AddRoleMember

Id: 600357f8-0499-47f8-87a5-eedf4ad034af

Description: Add a member to the group or role property.

Request

Control Message: AddRoleMemberRequest

Description: The request

Parameters

Name	Value
RoleName	Name of the role you are adding a member to. Optional parameter if there is only one role.

AccountAddress	Address, name or identifier of the account to be added to the role.

Response

Control Message: AddRoleMemberResponse

Description: The response

Parameters

Name	Value
Added	True or False.

RemoveRoleMember

Id: 97e160bb-6c60-4f1d-923b-813b07b89638

Description: Remove a member to the group or role property.

Request

Control Message: RemoveRoleMemberRequest

Description: The request

Parameters

Name	Value
RoleName	Name of the role you are adding a member to. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be removed from the role.

Response

Control Message: RemoveRoleMemberResponse

Description: The response

Parameters

Name	Value
Added	True or False.

IsInRole

Id: e42b1b16-074a-4d7d-b9f9-f69a2397a21b

Description: Check to see if an account is in the role.

Request

Control Message: IsInRoleRequest

Description: The request may be internal only and not exposed externally.

Parameters

Name	Value
RoleName	Name of the role you are checking membership of. Optional parameter if there is only one role.
AccountAddress	Address, name or identifier of the account to be checked.

Response

Control Message: IsInRoleRequestResponse

Description: The response

Parameters

Name	Value
InRole	True or False.