

The Regulated-METM Token Standard

Thomas Miller, Daniel Lew, Louis de Beer & Santiago Del Valle

Version 0.1
27th of May, 2019

Abstract

The Metamorph R-METM Token Standard is a protocol that provides a cutting-edge RegTech framework for the creation of, and compliance with, regulation of both utility and security tokens. The R-METM Token Creation Platform provides ownership of the R-METM Token Contract, facilitates the deployment of the R-METM token, where the participant is able to specify the name, the number of tokens, the number of decimal points, which is connected to an already deployed Regulator Contract, where the issuer/compliance officer becomes the admin of the Regulator Contract managing that jurisdiction, with the whole process being subject to the relevant regulation of the traditional instrument.

During the creation process, the participant who is the creator of the token, and its ultimate owners are checked and whitelisted as being compliant with the relevant regulation, including Anti-Money Laundering (AML), Know your Client/Customer (KYC), Know your Business (KYB), and applicable regulation in the relevant jurisdiction. All transfers to new owners of R-METM tokens are checked and whitelisted, which provides seamless trading opportunities. The deployment of the R-METM Token Standard on the Ethereum blockchain platform provides transparency to regulators and the community and ensures that the R-METM Protocol is trusted.

Creators of R-METM can furthermore use the services of the Metamorph FinTech Partners to issue securities, such as security tokens, in a compliant manner, which can be traded via the Metamorph Platform via the 0x Protocol or any other centralised or decentralised orderbook which is compliant with ERC20.

Table of contents

Utility tokens, Security tokens and the Regulated-METM Token Standard (R-METM)	4
Creating a utility token	4
Creating a security token	5
Legal	6
Issuing a security token:	6
Trading a security token:	7
Regulatory Passporting:	7
Legal and Consultants' Panel	7
Australia	8
USA	9
EU	10
Technical	12
Permissioned Tokens	12
Regulated-METM Token Standard	12
1. The Regulated-METM Token Contract (ERC20)	13
Transfer Sequence (Including Regulator Capacity)	13
Issuance Sequence (Including Regulator Capacity)	14
2. The Metamorph Token Regulator	15
Contract Features and States	15
Contract Locked	15
Fractional Trading	15
Blocking sender from sending	16
Blocking receiver	16
Allowing a participant to send tokens	16
Allowing a participant to receive tokens	16
The Check Function	16
Pausing All Trades	16
Pseudo Anonymous on chain record	16
Off-Chain KYC etc.	17
The mint function	17

Adding Regulators and setting permissions	17
3. The MetaMorph Regulator Registry	18
Future Proofing: Updating Regulator Contract	18
0x Protocol	19
Example how Regulated-METM Contract (ERC20) interacts with 0x protocol	19
Conclusion	20

Utility tokens, Security tokens and the Regulated-METM Token Standard (R-METM)

It is important to understand the difference. A regulated token can be either a utility Token or a security token, depending on the use case.

Regulated-METM Token Standard (R-METM)	
<p>A token that is regulated in terms of certain types of legislation, such as anti-money laundering, know your client (KYC) and other legislation designed to regulate the type of token, the identity of the owner, and the reason for trading the token.</p> <p>While utility tokens <u>may be</u> subject to regulation, security tokens are <u>always</u> subject to regulation.</p>	
<p>Utility Tokens A token with a utility value, such as a reward token or a token that can be used within an enclosed ecosystem as a method of exchange.</p>	<p>Security Tokens A token that derives its value, or is connected to, an external underlying tradeable asset. If a token is used to raise funds, it will be deemed to be a security.</p> <p>Tokens can even be deemed securities if they develop their own intrinsic value, such as a value due to scarcity. If a token is a security token, it has to be issued properly in terms of securities legislation, which varies from country to country.</p> <p>Issuing a security token in one country does not ensure that it can be traded in another.</p>

Creating a utility token

The MetaMorph Token Creation Platform makes it possible for users to issue and deploy a ERC-20 utility token onto the Ethereum blockchain. Users simply have to

select the name, ticker symbol, number of decimals and the total supply and they are ready to launch their very own token.

The token is also eligible to be listed on the MetaMorph Pro decentralised exchange against the METM, wETH and wBTC pairs inclusive of utility token creation cost. It is also possible to have a legal opinion drafted regarding the nature of the token from NXT Legal & Compliance (NXT:Legal), a leading FinTech and RegTech legal firm based in Australia.

Use cases for utility tokens include access tokens which gives the holder the right to enter and use the services in a specific ecosystem, verifiable voting rights, transparent fundraising and many more.

Creating a security token

This is the ideal solution for businesses looking to create a regulated cryptocurrency security token on the Ethereum blockchain. The new MetaMorph R-METM token standard (Regulated-METM) will more than suffice for any need, regardless of the scope of the project or business.

Business owners simply have to choose the name and ticker symbol during the process and the token will then be reserved and registered on the Ethereum blockchain while the legal team takes over and contacts the business regarding its specific needs, before the token is issued.

Use cases for security tokens include representation of traditional assets such as equity, derivatives, debt and real estate as well as other assets that classify as a security and which are subject to regulation under security laws.

Legal

A token can be a utility token or a security token.

The R-METM token standard allows for the creation of a utility token instantly in that it takes care of the KYC/AML requirements.

In the same way, the R-METM token standard also allows for the creation of a security token instantly, but this is a lot more complex in a RegTech sense and needs legal advice.

A security token is a security, or a security contract, and needs to be issued compliantly under the financial services regulations of a particular jurisdiction and when traded, also needs to be compliant with the financial services regulations in the jurisdictions where traded. In other words, it needs to be validly “passported” as well as issued, in addition to KYC/AML requirements.

We have done an overview of three jurisdictions being the Australia, the U.S. and the EU.

We are not providing legal advice, rather, we are using the examples below to explain the above.

It is very important that you get legal advice in respect of issuing and trading securities in your jurisdiction.

Issuing a security token:

A security token is a security and as such has to be **issued** in terms of the regulatory framework of the jurisdiction in which it is issued. For instance, if it is issued in Australia, it must be done under the auspice of an Australian Financial Services Licensee (AFSL) and in terms of the *Corporations Act 2001* (Corporations Act) and the Regulations thereto, under the governance and enforcement of the Australian Securities and Investments Commission (ASIC), who provides regulatory guides and class orders.

In the USA, the security would be issued in terms of at least four acts (and some legal precedents), with the U.S. Securities and Exchange Commission (SEC) applying the RegA+ regulatory framework to such issuance.

Trading a security token:

Once a security is issued, it must be traded on a regulated market also, which is subject to regulations also.

Regulatory Passporting:

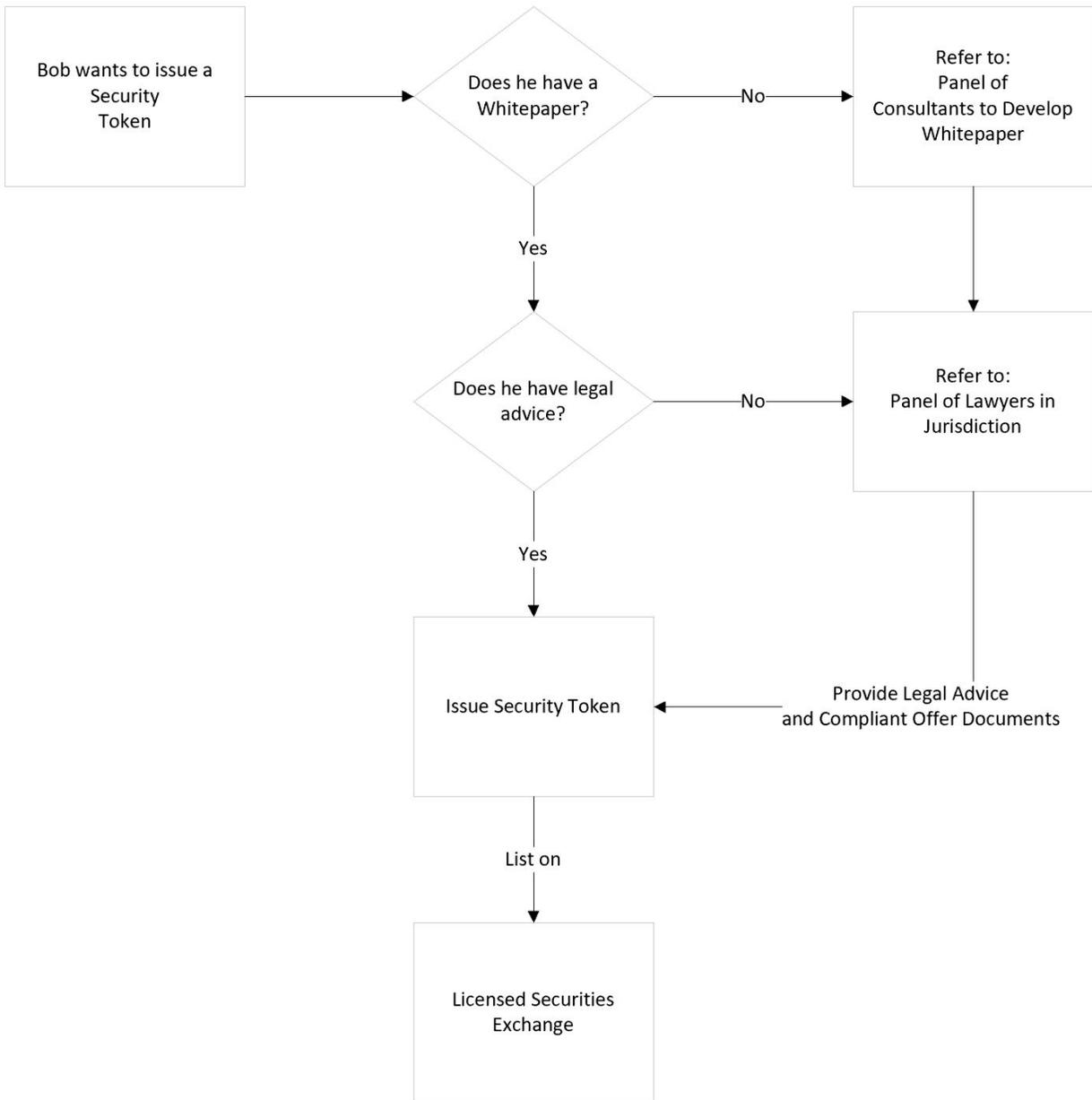
If a security is issued in a compliant manner in Australia, as an example, it cannot be necessarily traded in the U.S., or the EU. For this reason it also needs to be “passporting” in a compliant manner between jurisdictions.

Legal and Consultants’ Panel

We work closely with NXT.Online, a firm of consultants, and NXT.Legal, a law firm specialising in SmartContract, RegTech and Fintech, who facilitate the off-chain legal processes related to issuing a fully securities compliant and regulated cryptocurrency token to investors.

The laws regulating securities in various jurisdictions vary and NXT.Legal will be able to provide an initial opinion to you regarding the securities laws in your jurisdiction and be able to engage with their legal partners to provide you with a securities solution, trading solution and Regulatory Passporting in the appropriate jurisdiction.

In the event that you wish to issue a security token, the following process applies:



Australia

It is quite possible to issue a security under Australian securities law, the Corporations Act, using the services of an Australian Financial Services Licensee (AFSL) as a responsible entity such as:

- a licensee authorised to issue securities such as interests in a managed investment scheme. It is possible to link the interest, such as a unit in a unit trust to security token, which can then be traded; or
- a licensee authorised to issue debentures in a debenture mortgage scheme, where a special purpose vehicle holds an asset, such as a building against a loan and a mortgage, and the mortgage interest is 'fractionalised' into debentures held by the investors. The terms and conditions of the debenture can be written into a token and issued as a debenture mortgage token.

When a security token is traded, it would generally need to be traded under the 'making of a market' type license in Australia.

It can be traded via the metamorph.pro platform as MetaMorph does not operate a financial market as is contemplated by s767A of the Corporations Act for the reasons as set out in our disclaimer contained in the Terms and Conditions on the Token Creation Platform.

USA

The Securities Act of 1933, s 2 (a) (1) defines a security as:

“The term “security” means any note, stock, treasury stock, security future, security-based swap, bond, debenture, evidence of indebtedness, certificate of interest or participation in any profit-sharing agreement, collateral-trust certificate, preorganization certificate or subscription, transferable share, investment contract, voting-trust certificate, certificate of deposit for a security, fractional undivided interest in oil, gas, or other mineral rights, any put, call, straddle, option, or privilege on any security, certificate of deposit, or group or index of securities (including any interest therein or based on the value thereof), or any put, call, straddle, option, or privilege entered into on a national securities exchange relating to foreign currency, or, in general, any interest or instrument commonly known as a “security”, or any certificate of interest or participation in, temporary or interim certificate for, receipt for, guarantee of, or warrant or right to subscribe to or purchase, any of the foregoing.”

The Investment Company Act of 1940 has virtually the same definition, and deals with the requirements for being an 'issuer' of securities and the Securities Exchange Act of 1934 deals with the trading of securities.

In SEC V Howey CO., 328 U.S. 293 (1946), the U.S. Supreme Court found that:

For purposes of the Securities Act, an investment contract (undefined by the Act) means a contract, transaction, or scheme whereby a person invests his money in a common enterprise and is led to expect profits solely from the efforts of the promoter or a third party, it being immaterial whether the shares in the enterprise are evidenced by formal certificates or by nominal interests in the physical assets employed in the enterprise.

In summary, a security is defined by both the U.S securities legislation as above and the Howey test, where a group of persons pool their funds under the control of a third party, with the expectation of profit.

The treatment of the issuance of cryptocurrency has not been finally defined by the SEC and it is under discussion, but it is suggested that any security token offering be done in compliance with SEC regulations as they stand.

EU

The general Rules in respect of issuing of securities is contained in the DIRECTIVE 2003/71/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 4 November 2003 on the prospectus to be published when securities are offered to the public or admitted to trading and amending Directive 2001/34/EC.

Security tokens are regulated as securities under MIFID II — Financial Instruments Directive (2014/65/EU). This is the most important piece of relevant European legislation defining what a security is. Security tokens will be probably treated as “transferable securities”. These are the most common form of securities, but they could also come in other forms, e.g. derivatives. The law regarding issuing and trading securities in the EU is currently being harmonized and in the process of gradual unification. This means it is becoming easier to offer and trade securities across the whole EU .

If one wants to issue transferable securities and offer them publicly, they need a prospectus approved by the national financial regulator of the respective member country. This is regulated by the Prospectus Directive (2003/71/ES) as of now. If the national regulator approves the prospectus in one EU country and the security needs to

be offered in another EU country, it is only necessary to notify the regulator in the other country and the process is quite straightforward. In short, it is possible leverage blockchain-friendly EU countries to approve the prospectus, which can then be used in other EU countries by notifying the national regulatory bodies. Furthermore, a new Prospectus Regulation is planned for July 2019, so the process will most likely be even easier and more straightforward.

Technical

MetaMorph is built on the 0x protocol. 0x protocol is a set of contracts that facilitate an orderbook system sitting on the ethereum smart contracting platform. The ERC20 token standard describes the functions and events that an Ethereum token contract has to implement.¹ The exchange is not related to the token creation process but gives some context, explains how permissioned tokens evolved out of the 0x protocol community¹ and one way ERC20 tokens can be traded in a decentralised manner yet still satisfy the needs of regulators.

Permissioned Tokens

Permissioned tokens place restrictions on transfers directly within a token's contract code, limiting ownership to Ethereum addresses that meet certain requirements. Aside from transfer restrictions, these tokens can behave like any other ERC20 token that one may be familiar with. The most common implementation of a permissioned token is straight forward: every time the token's transfer function is invoked, the token contract checks a Registry contract to see if the recipient is registered to a whitelist. The transfer will only complete successfully if the recipient is registered to that whitelist.²

Regulated-METM Token Standard

The Regulated-METM token standard³ falls into the emerging category of permissioned tokens. The Regulated-METM standard adheres to the ERC201 interface allowing our tokens to be listed on any platform that supports ERC-20 tokens. In the same contract it also provides the need for compliance on deployment and transfers of these tokens representing traditional assets.

The Regulated-METM Standard consists of 3 contract addresses.

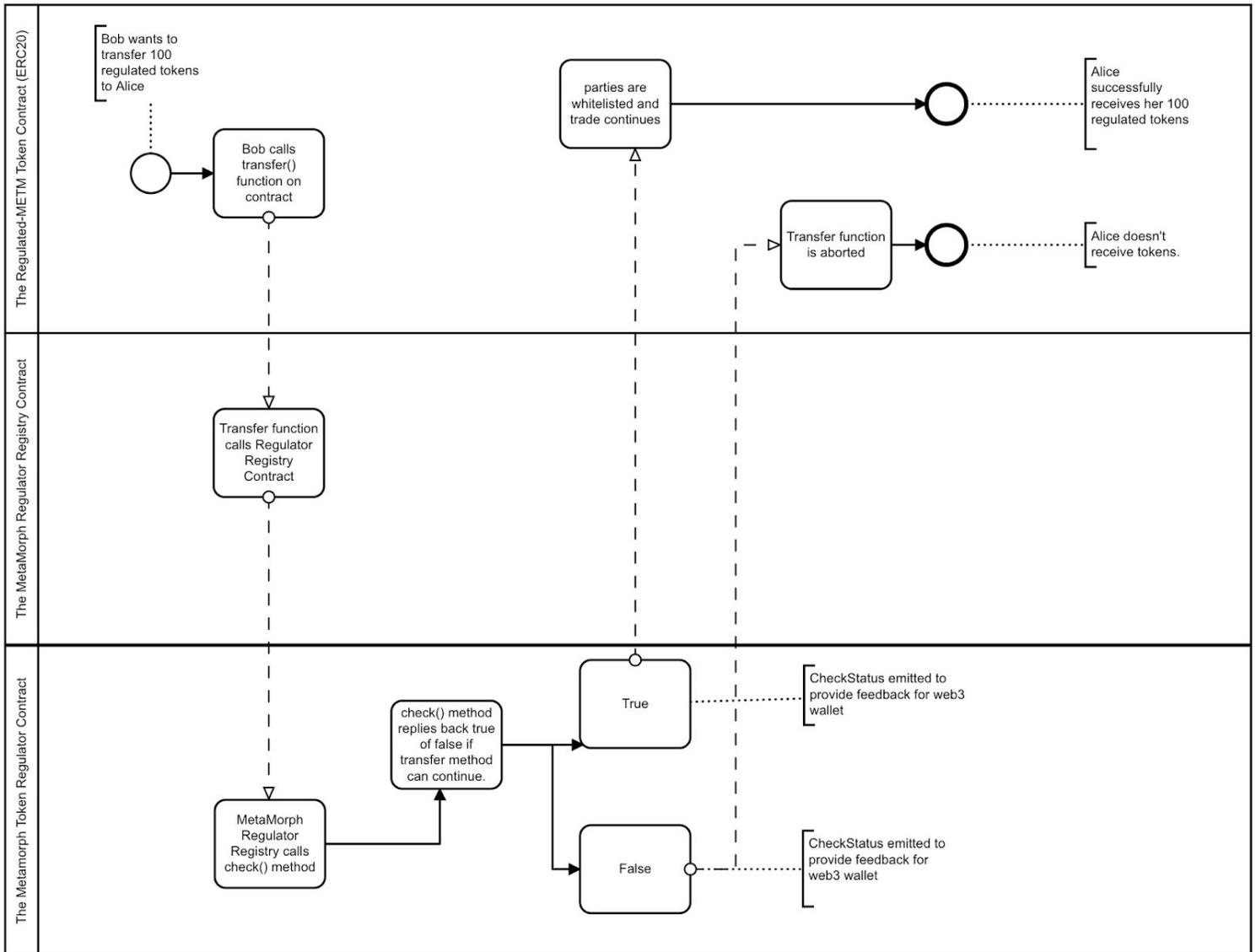
1. The Regulated-METM **Token Contract** (ERC20),
2. The Metamorph Token **Regulator**, and
3. The MetaMorph Regulator **Registry** (ability to upgrade Regulator).

1. The Regulated-METM Token Contract (ERC20)

This contract replaces the standard `transfer()`⁴ method with one that includes a compulsory `check()`⁵ method, if the check method returns back with the appropriate response (i.e. true) the transfer function continues to execute.

Transfer Sequence (Including Regulator Capacity)

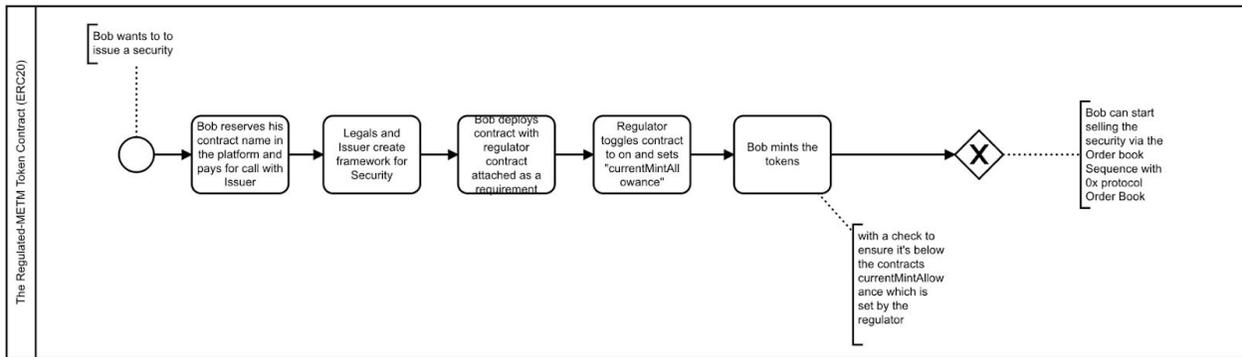
The regulator controls which addresses have full trading rights.



It is important to note that as the participant deploys a regulated token via the token creation platform the ERC20 contract is locked until the allocated admin governing that jurisdiction via the regulator contract, toggles the contract 'on'.

Issuance Sequence (Including Regulator Capacity)

The R-METM standard inherits all the functions from the standard ERC20 token. After a consultation with legals, the participant will deploy the contract and become the contract owner, where it will be governed by a regulator contract governing that particular jurisdiction. The compliance officer will become an admin of the Regulator Contract.



2. The Metamorph Token Regulator

It is likely one Metamorph Token Regulator contract will be implemented per jurisdiction.

This contract gets called by the Regulated-METM Token Contract (ERC20) everytime the `transfer()`⁴, `transferFrom()`⁶ and `mint()`⁷ functions are called.

Contract Features and States

These states have been created should the web3 wallet implementing these features wish to provide an error code on why the transfer didn't complete ultimately informing the end user on what they need to do next to become compliant.

The Metamorph Token Regulator has 3 main features with their associated states:

1. Toggle lock or unlock transfers `setUnlocked()`⁸ on a token contract, with the following error codes:
 - a. 0 = full trading
 - b. 1 = Token contract is locked
2. Toggle ability to trade a fraction of a token. `setPartialTransfers()`⁹
3. Set the following permissions `setPermission()`¹⁰ via uint8 value:
 - a. 2 = Sender is not allowed to send the token.
 - b. 3 = Receiver is not allowed to receive the token.
 - c. 0x1 = Allowing a participant to send tokens.
 - d. 0x2 = Allowing participant to receive tokens.

Contract Locked

The contract is deployed as locked, by default until an admin of the regulator contract changes the contract state to 0 = full trading. This can be done using the `setUnlocked()`⁸ function.

Fractional Trading

There is the ability to enable or disable fractions of tokens being traded. This can be done using the `setPartialTransfers()`⁹ function.

Blocking sender from sending

In certain circumstances there will be a need to block a sender from sending tokens. Setting the state of an end user's ethereum address to 2 will enable this ability. This can be done using the `setPermission()`¹⁰ function.

Blocking receiver

There may be times where it is required for an end-user to be able to send their tokens however not allowed to receive anymore. Setting the state of an end user's ethereum address to 3 will enable this ability. This can be done using the `setPermission()`¹⁰ function.

Allowing a participant to send tokens

Setting the state of an end user's ethereum address to 0x1 will enable this ability. This can be done using the `setPermission()`¹⁰ function.

Allowing a participant to receive tokens

Setting the state of an end user's ethereum address to 0x2 will enable this ability. This can be done using the `setPermission()`¹⁰ function.

The Check Function

The `check()`⁵ function runs through above states and if it returns true, trading is permissioned and the transfer function calling the `check()`⁵ function will be allowed to continue the requested transfer.

Pausing All Trades

If the regulator wishes the pause all trades of the the instrument, they can do so simply by toggling the contract to off. This can be done using the `setUnlocked()`⁸ function.

Pseudo Anonymous on chain record

The whitelist is kept on-chain via the regulated contract. This is referred to as pseudo anonymous because names are not linked with the ethereum addresses on-chain.

Off-Chain KYC etc.

The regulator will keep an off-chain record of which parties own which ethereum address.

The mint function

There will be times where they end user will be permissioned to mint more tokens, this will be done with permission from the regulator contract. The Regulator contract will set the `currentMintAllowance()`¹ allowing the end user to mint tokens up to the current mint allowance.

Adding Regulators and setting permissions

As regulators need to come onboard they will be added to this contract as an admin with their permissions set via the `setAdmin()`¹¹ function allowing them to regulate the ERC20 contracts that point to this regulator contract. The regulator/compliance officer will mark end-user ethereum addresses with their appropriate whitelist rights (described below) adhering to the particular jurisdiction if they pass all the regulatory checks associated with their traditional instrument.

3. The MetaMorph Regulator Registry

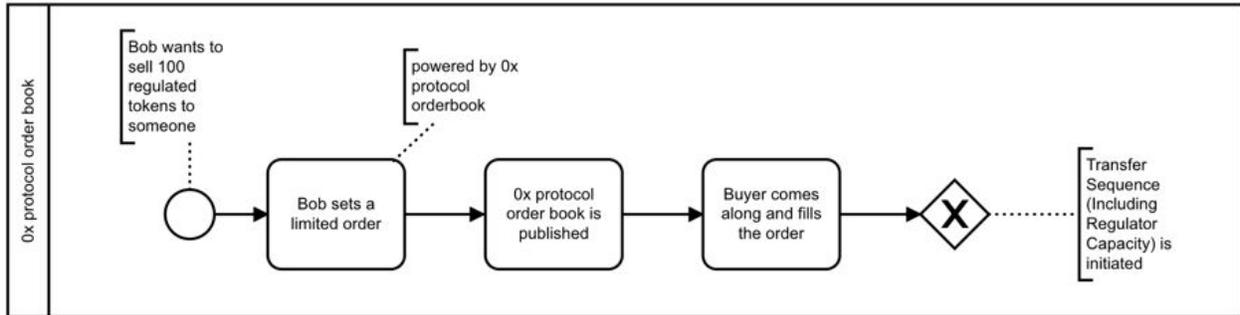
Future Proofing: Updating Regulator Contract

Although it is highly unlikely as a future proofing fail safe, the MetaMorph Regulator Registry¹² is introduced to simply be a proxy contract, that allows The Metamorph Token Regulator to be upgraded with a different contract should there be a need for a new capacities in the Metamorph Token Regulator. This essentially allows the regulator contract to be upgradable.

0x Protocol

Example how Regulated-METM Contract (ERC20) interacts with 0x protocol

Below is the order book Sequence with 0x protocol order book. The orderbook can be centralised or decentralised.



Conclusion

In conclusion, from our interpretation of the risks and the requirements of traditional financial instruments, we believe that we are on a path to pave a safe way for blockchain technology to be safely implemented to underpin traditional securities and allow all stakeholders to begin testing the benefits for themselves.

The current contract setup allows for a regulator contract to govern all ERC20 contracts in that particular jurisdiction which mirrors and assists the regulator with a pretty cool tool that can be added to the regulators 'digital toolbox'.

References:

1. https://theethereum.wiki/w/index.php/ERC20_Token_Standard
2. <https://blog.0xproject.com/compliant-peer-to-peer-trading-4dab8e5c3162>
3. <https://github.com/MetaMorphPro/R-METM-Standard>
4. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphRegulatedToken.sol#L48>
5. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphTokenRegulator.sol#L150>
6. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphRegulatedToken.sol#L65>
7. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphRegulatedToken.sol#L81>
8. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphTokenRegulator.sol#L92>
9. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphTokenRegulator.sol#L119>
10. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphTokenRegulator.sol#L107>
11. <https://github.com/MetaMorphPro/R-METM-Standard/blob/9f3541d4baee807b6a3024e079d9e02786ce1bc6/rToken/MetamorphTokenRegulator.sol#L130>
12. <https://www.sec.gov/corpfin/framework-investment-contract-analysis-digital-assets>