
The CMTA Token



JP Aumasson

@veorq

CSO @ taurushq.com

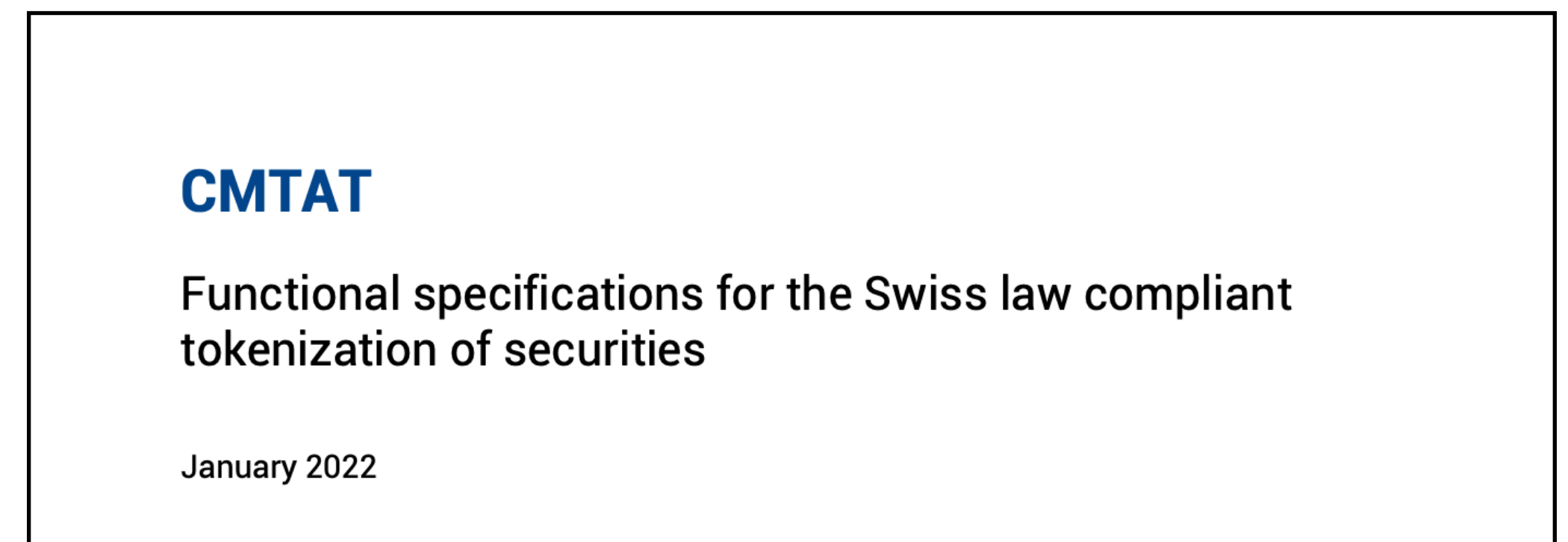
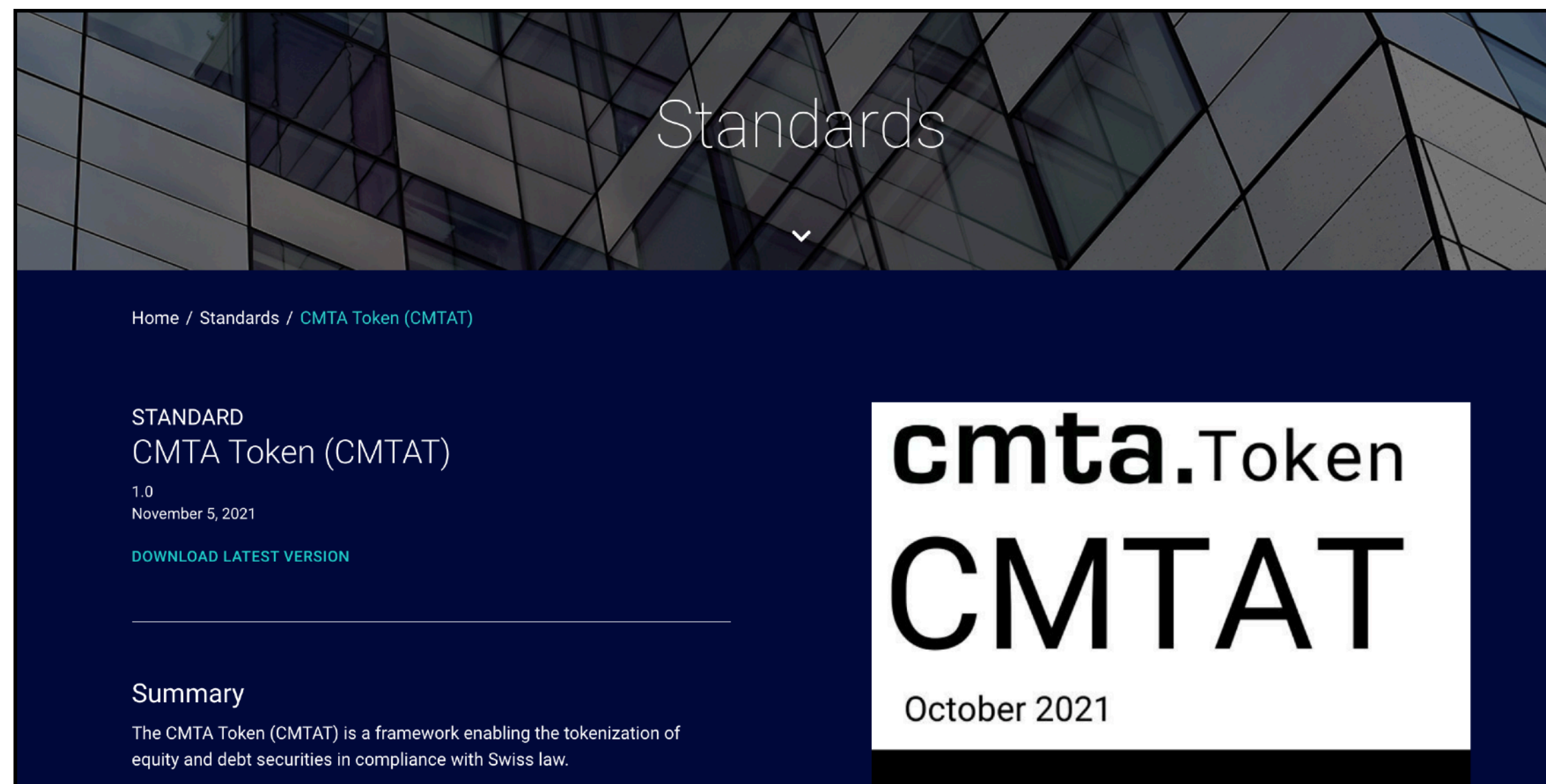
SCHWEIZERISCHE NATIONALBANK
BANQUE NATIONALE SUISSE
BANCA NAZIONALE SVIZZERA
BANCA NAZIUNALA SVIZRA
SWISS NATIONAL BANK



Conference on Cryptoassets and Financial Innovation

Open standard of a Swiss security token

- By a **CMTA Working Group** started in March 2021 orchestrated by yours truly
- Thanks to all contributors: Atpar, Bitcoin Suisse, Blockchain Innovation Group, Hypothekarbank Lenzburg, Lenz & Staehelin, Metaco, Mt Pelerin, SEBA, Swissquote, Sygnum, Taurus, Tezos Foundation



Motivation: Switzerland' “Lex DLT”

- Officially allows tokenisation of Swiss companies since **August 2021**
- CMTAT saves companies from having to design and audit a custom contract
- Becoming the *de facto* Swiss security token standard

- H. Registerwertrechte

- I. Errichtung

- Art. 973d⁶⁴³

¹ Ein Registerwertrecht ist ein Recht, das gemäss einer Vereinbarung der Parteien:

1. in einem Wertrechtregister gemäss Absatz 2 eingetragen ist; und
2. nur über dieses Wertrechtregister geltend gemacht und auf andere übertragen werden kann.

² Das Wertrechtregister muss die folgenden Anforderungen erfüllen:

- H. Droits-valeurs inscrits

- I. Constitution

- Art. 973d⁶⁵¹

¹ Est droit-valeur inscrit tout droit présentant les caractéristiques suivantes par convention entre les parties:

1. il est inscrit dans un registre de droits-valeurs au sens de l'al. 2, et
2. il n'est possible de le faire valoir et de le transférer que par ce registre.

² Le registre de droits-valeurs doit satisfaire aux exigences suivantes:

CMTAT is already used

- High-profile tokenisation projects used CMTAT
- Integrated in the Taurus-CAPITAL token issuance solution

FINANCEMENT

QoQa et MagicTomato succombent à la mode des tokens

Des PME continuent à ouvrir leur capital à de petits investisseurs via des jetons numériques. Reste ensuite à entretenir l'intérêt autour de leurs titres. Deux entreprises expliquent comment elles comptent s'y prendre

<https://www.letemps.ch/>

Tokenisierung: Eine kurze Erklärung

5. Mai 2022

 Von QoQaBrew

24 KOMMENTARE FAVORITEN

Im Herzen des QG, dem neuen QoQa-Hauptquartier in Bussigny (VD), wird Anfang 2023 die **Brauerei QoQa Brew** seine Tore eröffnen. Die Idee dazu ist im Otterbau entstanden und da wir die Dinge gerne anders angehen als alle anderen und, wann immer möglich, unsere Community miteinbeziehen, haben wir und das QoQa Brew Team entschieden, gemeinsam mit **unserem Partner Taurus**, die Brauerei mithilfe der Tokenisierung, einer innovativen partizipativen Finanzierung, auf die Beine zu stellen.

<https://qblog.qoqa.ch/>

Why not reusing an existing token?

- Often limited to adding **transfer restrictions**
- Tied to a **specific** blockchain or ecosystem
- Not designed to conform to the **Swiss law**

ERC-1404

Simple Restricted Token Standard

| an open standard for security tokens

```
contract ERC1404 is ERC20 {  
    function detectTransferRestriction (  
        address from,  
        address to,  
        uint256 value  
    ) public view returns (uint8);  
  
    function messageForTransferRestriction (  
        uint8 restrictionCode  
    ) public view returns (string);  
}
```

ST-20

SECURITY TOKEN STANDARD

Functionalities

- **Modular** design, with mandatory and optional modules
 - **Base** (mandatory): basic transferable token functionalities

1. **TotalSupply**: For a particular CMTAT token, any person may know the total number of tokens in circulation at any point in time. ●
2. **BalanceOf**: For a particular CMTAT token and a particular user, any person may know the number of tokens currently recorded on the user's ledger address. ●
3. **Transfer**: Users may transfer some or all of their tokens to some other ledger address (that the transferor does not necessarily control). ●
4. **Mint**: Issue a given number of tokens to a given ledger address. ■
5. **Burn**: Burn (destroy) a given number of tokens from a given ledger address. ■
6. **Pause**: Prevent all transfers of tokens on the ledger until "**Unpause**" is called. ■
7. **Unpause**: Restore the possibility to transfer tokens on the ledger, in principle after "**Pause**" is called. ■
8. **Kill**: Self-destruction of the contract and effectively of the tokens, thereby preventing any transfer or other operation. ■

Mandatory attributes, applicable to all CMTAT tokens:

- Name
- Ticker symbol (optional)
- Token ID (ISIN or other identifier) (optional)
- Reference to the terms of tokenization, the terms of the instrument, and other relevant documents (e.g. prospectus or key information document). The reference can take the form of an URL, a combination of an URL and of specific directions allowing the user to retrieve the relevant documents (e.g. "[domain].com/shares > Tokens") or a fingerprint.

Functionalities

- **Modular** design, with mandatory and optional modules
 - **Base** (mandatory): basic transferable token functionalities

1. **TotalSupply**: For a particular CMTAT token, any person may know the total number of tokens in circulation at any point in time. ●
2. **BalanceOf**: For a particular CMTAT token and a particular user, any person may know the number of tokens currently recorded on the user's ledger address. ●
3. **Transfer**: Users may transfer some or all of their tokens to some other ledger address (that the transferor does not necessarily control). ●
4. **Mint**: Issue a given number of tokens to a given ledger address. ■
5. **Burn**: Burn (destroy) a given number of tokens from a given ledger address. ■
6. **Pause**: Prevent all transfers of tokens on the ledger until "**Unpause**" is called. ■
7. **Unpause**: Restore the possibility to transfer tokens on the ledger, in principle after "**Pause**" is called. ■
8. **Kill**: Self-destruction of the contract and effectively of the tokens, thereby preventing any transfer or other operation. ■

Optional attributes, applicable to tokens used for debt securities:

- Guarantor identifier (if any)
- Bondholder representative identifier (if any)
- Maturity date
- Interest rate
- Par value (principal amount)
- Interest schedule format (if any). The purpose of the interest schedule is to set, in the parameters of the smart contract, the dates on which the interest payments accrue.
 - Format A: start date/end date/period
 - Format B: start date/end date/day of period (e.g. quarter or year)
 - Format C: date 1/date 2/date 3/....
- Interest payment date (if different from the date on which the interest payment accrues):
 - Format A: period (indicating the period between the accrual date for the interest payment and the date on which the payment is scheduled to be made)
 - Format B: specific date
- Day count convention
- Business day convention
- Public holidays calendar

Functionalities

- **Modular** design, with mandatory and optional modules
 - **Snapshot** (mandatory): record the distribution of tokens at a given time

1. **ScheduleSnapshot**: For a particular CMTAT token, the issuer may schedule the creation of a snapshot at a certain time. The time of the newly scheduled snapshot cannot be before the time of the latest scheduled, but not yet created, snapshot. ■
2. **RescheduleSnapshot**: The issuer can change the time of a scheduled snapshot. The new scheduled time cannot be before the time of the previously scheduled snapshot or after the time of the next scheduled snapshot (*i.e.* scheduled snapshots cannot be reordered). ■
3. **UnscheduleSnapshot**: For a particular scheduled snapshot, the issuer can cancel a previously scheduled snapshot. The unscheduled snapshot must be the last scheduled snapshot, and its time must be in the future. ■

4. **SnapshotTime**: For a particular scheduled, but not yet created, snapshot, anyone may know the snapshot time.
5. **SnapshotTotalSupply**: For a particular created snapshot, anyone may know the total number of tokens that were in circulation at the snapshot creation time.
6. **SnapshotBalanceOf**: For a particular created snapshot and a particular ledger address, anyone may know the number of tokens recorded on the relevant ledger address at the snapshot creation time.

Functionalities

- **Modular** design, with mandatory and optional modules
 - **Validation, Authorization** (optional)

Functionalities:

1. **ValidateTransfer**: Send a request for validation of a particular transfer, given the sender and recipient addresses, and the amount to the issuer. ●
2. **SetRuleEngine**: Assign a set of rules to be enforced by the **ValidateTransfer** function. Said rules are to be defined in a separate contract. ■

d. Authorization module

Rationale: Issuers may wish to implement a role-based access control to the token functionalities, rather than distinguish only between issuer and user. This may help reflect the issuer organization's governance model. The Authorization module thus allows the issuer to assign responsibilities and authorizations to various persons (accounts).

Functionalities:

1. **GrantRole**: Grant a role to a given account. ■
2. **RevokeRole**: Revoke a role from the given account. ■
3. **HasRole**: Tell whether a given account has a given role. ■

Corporate actions support

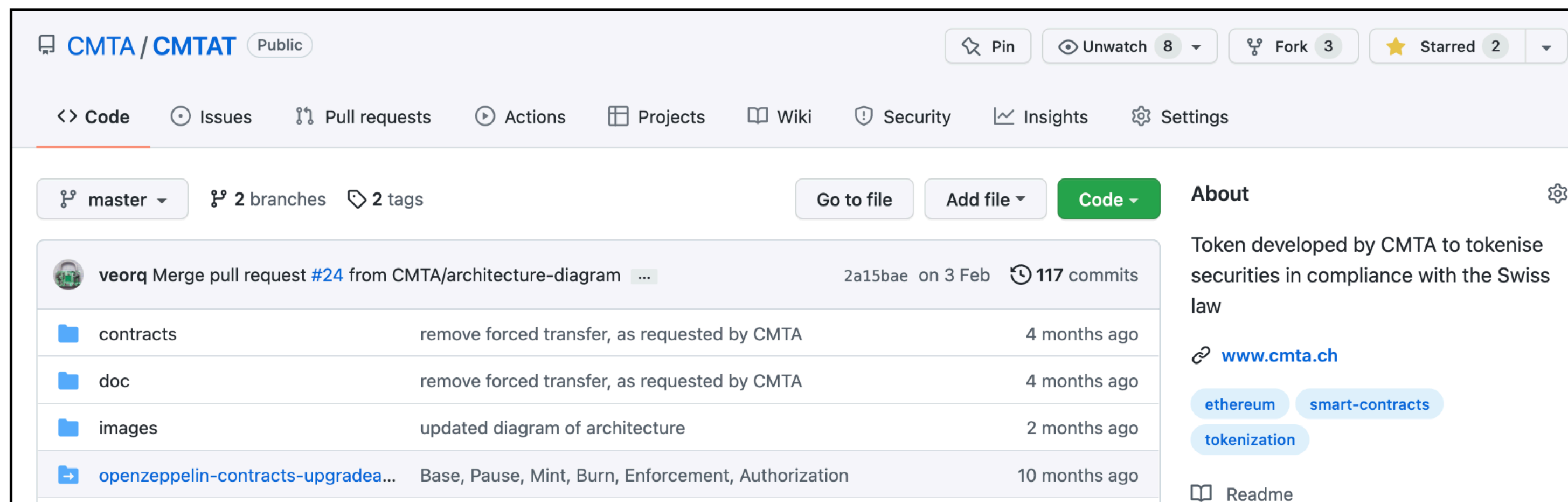
- **Guidelines** on how to deal (off-chain for the moment) with:
 - (Reverse) splits: new token contract deployed
 - Dividend and interest: triggered from a snapshot
 - Credit events: API defined to set flag/rating of an account

A module to manage on-chain credit events may support the following functionalities:

1. **FlagDefault**: The tokens are flagged as representing debt instruments in respect of which the issuer has defaulted (bond agent function). ■
2. **FlagDefaultRemove**: The tokens are no longer flagged as representing debt instruments in respect of which the issuer has defaulted (bond agent function). ■
3. **FlagRedeemed**: The tokens are flagged as representing debt instruments that have been redeemed (bond agent function). ■
4. **SetRating**: The tokens are flagged as representing debt instruments that have been given a particular rating (bond agent). ■

Reference implementation

- Open-source **Solidity** version at <https://github.com/CMTA/CMTAT/>
 - Suitable for Ethereum, Avalanche, Binance Smart Chain
- Supports **proxying and gasless** transactions with standard APIs
- Security audit by ABDK



Certification process

- Process to assert that a token is conform with the law (e.g. CMTAT extensions, variants)
- Recognized experts for **legal** and **tech** aspects appointed by CMTA

CMTA.Tokenized Shares
Certification

▼

CERTIFICATION
CMTA.Tokenized Shares Certification

Certified CMTA.Tokenized Shares are shares that have been tokenized in accordance with Swiss law and industry standards.

CMTA.Tokenized Shares certification is a guarantee to shareholders, operators of trading systems, and other business partners that a company's shares have been validly tokenized in accordance with Swiss law and highest industry standards.

Downloads

- [Certification Mark Regulations](#) EN FR
- [Standard for the Tokenization of equity securities](#) EN FR
- [Logo usage guidelines](#) EN FR
- [Fee Schedule](#) EN FR
- [Application Form](#) EN FR

In the works

- Full support for **debt** products and **structured** products
- **On-chain** drag-along and delivery-versus-payment
- More **implementations** (Tezos and others)
- Share tokenisation **Q&A**

Thank you!

- <https://cmta.ch/standards/cmta-token-cmtat>
- <https://cmta.ch/certification/cmta-tokenized-shares-certification>
- <https://github.com/CMTA/CMTAT/>



JP Aumasson
@veorq

CSO @ taurushq.com