CMTAT

Engineering a Security Token on EVM, with a Path to Privacy

Ryan
(Taurus/CMTA)

CMTAT

- Security Token Standard framework
- Multi-blockchain support:
  - Public: Ethereum/EVM, Tezos, Solana (Specification only)
  - Privacy: Aztec (2025), Zama FHE (March)
- Used by Taurus, UBS (money market fund), Obligate (Debt), Daura (equities), 21.co (past), Zand (Emirati Dirham (AED))

EVM Version

- Written in Solidity
- Built on top of ERC-20 and OpenZeppelin libraries

Security

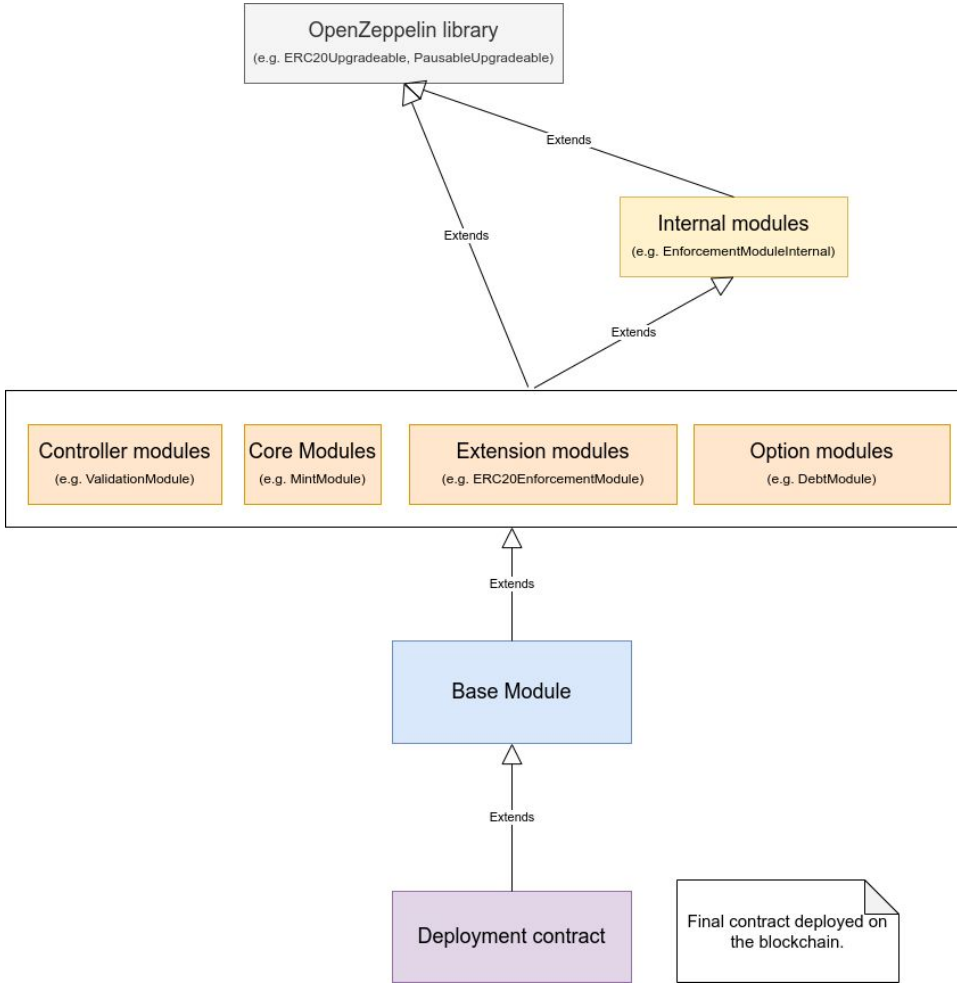
- Major releases are audited (Halborn, ABDK)
- Minor releases: Static analyzer (Aderyn/Slither) & AI Audit tools (Nethermind Audit Agent and Wake Arena)

Challenge

- One codebase for:
  - Several use cases (equities, debt, stablecoin)
  - Several jurisdiction and region
- Codebase must be flexible, modular and easily adaptable
- Must also be integrated inside Ethereum ecosystem (ERC)

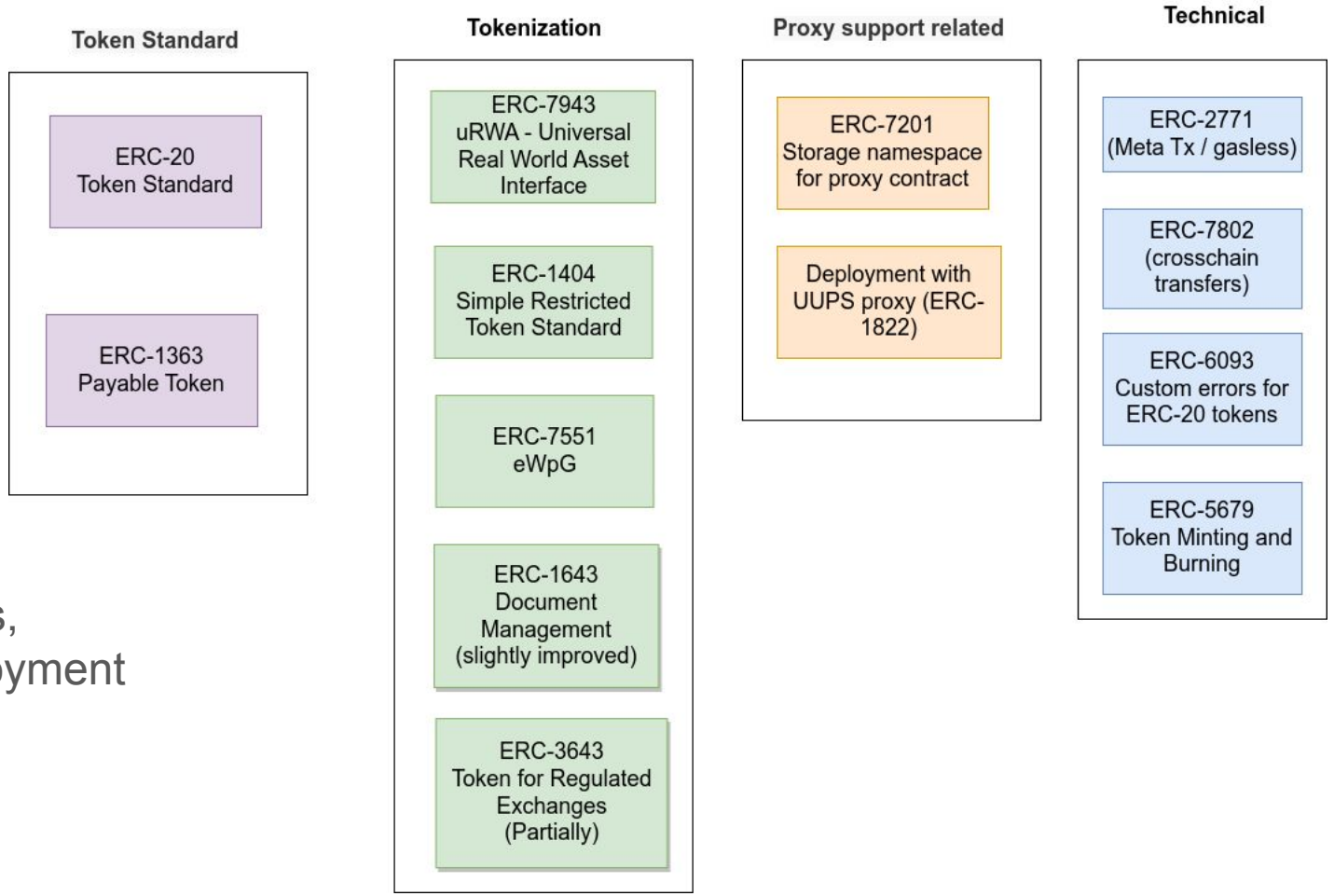
Architecture

- Functionalities are separated into modules (abstract contracts)
  a. CMTAT version for ERC-721 with only non-ERC-20 modules
  b. Adapt the contract using only the specific modules you need
- Several versions available (Standard, Debt, Light (stablecoin))
- Can be deployed through a proxy (upgradeable) or as a standalone contract
- Several utilities contracts such as RuleEngine & Rules for compliance rules



Ethereum Standard

Use: modules, specific deployment version



Making CMTAT cross-chain

Goal

- Make it easier to enable CMTAT with a cross-chain bridge

Challenges

- Maintain the same rights and properties on each blockchain
- Ensure a consistent whitelist/blacklist across all blockchains
- Enable transferability between different blockchains (primary focus)

Key Considerations

- Bridge selection and integration
- Token–bridge compatibility
- Bridge configuration and support

Making CMTAT cross-chain

Specific modules

- ERC-7802 (Optimism and Unichain)
- Chainlink CCIP (set CCIP admin)

External contracts

Two adapter contracts for LayerZero:

- ERC-7802 cross-chain minting and burning

- ERC-3643 standard minting and burning

Privacy - Design

- Select the technology (Aztec, Zama FHE, Midnight, etc.)
- CMTAT version: Aztec (Noir), Zama FHE (Solidity)
- Define which information is public vs. private
  - Address: public or private?
  - Balances and transfer amounts: public or private?
  - Total supply: public or private?
    - If public, this may potentially leak some balances (e.g., via mint and burn operations)
- Add compliance features: viewing keys, address freezing, and burn functionality

Privacy - Integration

- More complex for institutions to integrate into existing systems
- Additional components to manage
  - **Zama**: KMS and gateway
  - **Aztec**: encrypted private notes, nullifiers, and viewing keys
- Since the on-chain state is private, a private key is required to decrypt balances and state

CMTAT GitHub

X
(personal account)