International
Capital
Market
Association

ICMA

ICMA contribution to MAS Guardian Fixed Income Framework (GFIF) publication

Summary of third-party proposals for integrating ICMA's Bond Data Taxonomy in token frameworks and DLT platforms complementing section 6, Data Model, of GFIF

4 November 2024

Table of Contents

1. Introduction 2
2. Project Guardian 3
3. ICMA Bond Data Taxonomy 3
4. UBS Proposal to augment CMTAT with ICMA fields and tamper-proof methodology...... 5
5. FeverTokens Proposal: Integrating BDT into so|bond v2.0: A Modular Approach to Enhancing Bond Tokenization...12
6. Tokeny Proposal for Extension of ERC 3643 for ICMA Bond Data Taxonomy 17

1. Introduction

This publication is to be read in conjuntion with the MAS Guardian Fixed Income Framework (GFIF), published on 4 November 2024. GFIF seeks to scale tokenisation of fixed income products by promoting industry standards, notably fixed income protocols, and data specifications building on existing standards such as ICMA's Bond Data Taxonomy (BDT), amongst other objectives. While tokenised assets embed novel features such as smart contracts to automate coupon payments or (peer-to-peer) transfer of ownership, amongst other processes, each debt security has economic terms at its core irrespective of its form i.e. whether "native digital", tokenised or "traditional".

The following complements section 6 of the GFIF, Data Model, and in particular three BDT integration examples set out by UBS, FeverTokens and Tokeny, as referenced in the MAS GFIF publication. It describes how the BDT can be used in a DLT context to define a DLT-based debt security consistently, automate workflows and promote interoperability across different smart contract frameworks and DLT platforms, including public or private DLT or blockchain networks. What data is stored on-chain and off-chain, and the use of oracles (external data sources) if required, will be dependent on the use case and notably the type of DLT or blockchain network and its characteristics i.e. whether private or public, permissioned or permissionless, centralised or decentralised.

The proposal by UBS demonstrates how to augment the Capital Markets and Technology Association (CMTA) smart contract framework for the tokenisation of debt securities under Swiss law, with BDT fields and XML schema. The CMTA contract also supports third-party role based actions that enhances flexibility across issuance workflows. This can make financial transactions faster, more secure and enhance interoperability.

FeverTokens' proposal originates from the the successful SATURN project, which saw the European Investment Bank (EIB) issue €100 million worth of tokenized bonds. FeverTokens' proposal is to integrate the BDT into its open-source smart contract package framework and "so|bond v2.0", which was developed through a collaboration between Crédit Agricole CIB and Skandinaviska Enskilda

Banken (SEB). The framework facilitates the implementation of smart contracts under the Ethereum Virtual Machine (EVM) standard, with openly released source code. This standardization promotes straight-through processing (STP) and interoperability across different platforms and systems, reducing operational friction and enhancing efficiency.

Tokeny's proposal represents the bond onchain by using a compliant-by-design token framework such as ERC 3643 combined with the BDT to describe the asset, allowing the automation of both the issuance process and the post trade lifecycle management.

2. MAS Project Guardian

The Monetary Authority of Singapore (MAS)'s Project Guardian aims to drive and scale asset tokenisation in fixed income, foreign exchange (FX) and asset & wealth management. This included partnerships with global industry associations and financial institutions.

Project Guardian's fixed income workstream (GFIF) collaborated with ICMA to develop protocols and data specifications building on ICMA's Bond Data Taxonomy, and consider the types of risk factors and disclosures required in a tokenised bond offering document. Workstream members also collaborated with GFMA to develop standard clauses for implementing smart contracts.[1]

ICMA's BDT is incorporated to promote industry standards, fixed income protocols and technical data specifications for asset tokenisation (DLT-based). The aim is to provide guidelines that facilitate the use of different protocols and smart contracts based on a consistent data model across the securities lifecycle – both for digital (DLT-based) bonds and traditional debt securities.

The Project Guardian publication in full can be found here.

3. ICMA Bond Data Taxonomy

Whilst the tokenisation of assets is a relatively new phenomenon, the core foundations of a debt security remain, regardless of its form. For example, the economic terms such as an issuance amount denominated in a currency, maturity date and ISIN, remain irrespective of a "tokenised" or "traditional" bond security. ICMA's BDT, launched in early 2023, provides a common language for key bond information typically contained in a bond term sheet in a standardised and machine-readable (XML) format.

The BDT defines unambiguously the data elements of a debt security as well as related information such as party roles, ratings, benchmark, form of note, listing market, governing law, selling restrictions, and DLT platform information that may be communicated between different parties in a transaction (e.g. via APIs, smart contracts, or spreadsheets).

From a technical perspective, each field is defined by its data type, multiplicity and restrictions in the BDT's XML Schema Definitions (XSDs). Many of the fields are defined by ISO standards such as date, currency and legal identifiers. The XSDs' embedded logic includes syntax patterns and predefined enumerations to facilitate accurate and consistent data exchange and straight-through processing

[1] See MAS press release of 27 June 2024: https://www.mas.gov.sg/news/media-releases/2024/mas-expands-industry-collaboration-to-scale-asset-tokenisation-for-financial-services

(STP), for example, for the announcement of a new issue, allocation and pricing, clearing and settlement, distribution or asset servicing.

The BDT has been designed and developed by a broad range of bond market constituents to promote automation and reduce the risk of fragmentation across issuance, trading, settlement and distribution of debt securities. The BDT is technology agnostic and designed to be used both for traditional debt securities as well as DLT-based debt securities.

In February 2024, it was announced that the Government of the Hong Kong Special Administrative Region (SAR) of the People's Republic of China had successfully completed a multi-currency digital green bond issuance. This marked the first adoption of ICMA's Bond Data Taxonomy (BDT) by a government issuer and was a first for a green bond. In June 2024, ICMA and the Hong Kong Monetary Authority (HKMA) ran a webinar to discuss the key features of the issuance and how ICMA's BDT was implemented. To view the webinar in full, please see the video here.

Disclaimer

Nothing set out herein shall constitute or be construed as legal advice or ICMA guidance or recommendations. While the information contained herein is taken from sources believed to be reliable, ICMA does not represent or warrant that it is accurate or complete and neither ICMA nor its employees shall have any liability arising from or relating to the use of this publication or its contents, including any information on any third-party website which may be referred to in this document.

ICMA does not recommend or endorse any third-party solution or BDT implementation proposal.

4. UBS Proposal to augment CMTAT with ICMA's Bond Data Taxonomy and tamper-proof methodology

Overview

The Capital Market Technology Association Token (CMTAT) is a digital token standard developed to streamline the issuance, trading, and management of securities on blockchain platforms. It aims to improve efficiency, security, and compliance in capital markets by providing a robust framework for tokenizing financial instruments like bonds and equities. Integrating fields proposed by the International Capital Market Association (ICMA) into CMTAT will ensure industry-standard adherence and enhance the token's utility for bond issuance.

Integrating ICMA fields into CMTAT is crucial for standardizing the capital markets industry. Using a common taxonomy like the Bond Data Taxonomy (BDT) and XML schema for transaction details enables faster, safer exchanges of financial instruments. This standardization enhances interoperability, streamlines collaboration, reduces errors, and ensures regulatory compliance, leading to a more efficient, transparent, and secure capital markets ecosystem.

Tokenization in Bond Markets: Challenges, Solutions & Impact

Current Challenges

- Fragmented Processes Leading to Delayed Issuance: New tokenized products are taking too long to issue due to fragmented systems and lack of automation.
- Regulatory Compliance Risks: Without a standardized bond taxonomy, inconsistencies arise, creating potential regulatory and compliance risks.

UBS's Role:

- UBS is committed to revolutionizing the bond market by integrating advanced technology for a more efficient and compliant tokenization process.
- Our focus is on bridging the gap between traditional and digital finance through blockchain technology and implementing innovative blockchain solutions to streamline bond issuance.
- UBS, in collaboration with MAS, developed a standardized framework integrating ICMA Bond Data Taxonomy (BDT) for tokenized fixed income securities, bridging traditional finance (CeFi) and decentralized finance (DeFi).

Business Need:

- Standardization: Introducing an industry-wide standard to enable smoother processes and automation in bond tokenization.
- Regulatory Adherence:

- Ensuring that bond issuance meets global regulatory requirements through integrated data structures like the ICMA Bond Data Taxonomy (BDT).
- Mitigating operational risks through a unified bond data taxonomy.

Solution

- Key Standards Adopted:
  - ICMA Bond Data Taxonomy (BDT): Provides a unified language for bond issuance.
  - ERC20 & CMTAT: Current tokenization standards that can work alongside BDT to ensure cross-system compatibility.
- Bridging CeFi and DeFi: Addressing the gap between centralized (CeFi) and decentralized finance (DeFi) by enabling seamless integration of bond data on blockchain using Solidity smart contracts.
- Automation Benefits: Straight-Through Processing (STP): Automated workflows ensure faster issuance, reduce errors, and minimize human intervention i.e. automated compliance checks and secure on-chain settlement.
- ICMA Bond Data Taxonomy (BDT): Provides a unified standard for bond issuance, as demonstrated in the Project Guardian white paper co-authored by UBS and MAS, ensuring compliance and efficiency.

Business Impact:

- Faster Time-to-Market: Standardized processes accelerate bond issuance, improving efficiency.
- Real-Time Transparency and Auditability: On-chain records provide auditability and transparency for stakeholders.
- Regulatory Compliance: Alignment with global standards reduces compliance risks.
- Cost Efficiency: Reducing manual processes and associated costs through automation.

Tokenization in Bond Markets: Process and Deployment Strategy

Outlining UBS's end-to-end tokenization process for bonds, integrating ICMA BDT into UBS's tokenization flow, dealing key steps from data input to compliance and validation to streamline operations and enhance compliance.

Deployment Strategy

1. Deployment Steps:

    - Deploy Rule Engine, Document Engine, and Debt Engine modules.
    - Integrate these modules with the main smart contract for seamless operation.

2. Maintenance Schedule:

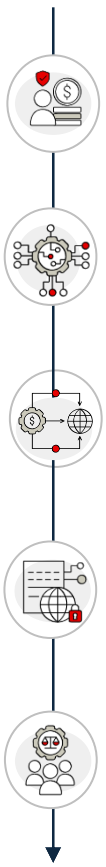
    - Monthly Updates & Quarterly Audits: Routine updates, optimizations, and security assessments.

- Annual Review: Comprehensive evaluation and improvement of the tokenization modules.

Compliance Framework

- Swiss DLT Law: Ensure that smart contracts are aligned with Swiss digital ledger technology regulations.
- KYC/AML Compliance: Integrate KYC and AML procedures to ensure full compliance in token issuance.
- Integration with ICMA standards, as established in the Project Guardian white paper, ensures that digital bond issuance processes align with global regulatory requirements.

Process Flow



Input Bond Data: Utilize ICMA BDT to capture bond characteristics (e.g., interest rates, issuer, maturity dates).

Mapping to Smart Contracts: Fields are mapped into Solidity contracts following standards such as CMTA and ERC20, ensuring compatibility.

Tokenization: The bond is tokenized on-chain, securing data integrity and ensuring compliance through automated processes.

Tamper-Proof Mechanism: Implement Merkle Trees and hashed data for enhanced security, ensuring the authenticity and validity of records.

Industry Adoption: The Project Guardian framework has become a reference point, standardizing digital bond issuance for global markets.

CMTAT Token Features

- Base Token Specification
- Transfer Restriction
- Document Management
- White List Management
- Token Contract Pause
- Snapshot/Checkpoints
- Support for Debt Instruments
- Thirdparty Security Audit
- Role Based Access Control
- Security Identifiers
- Mint and Burn to Any Address
- Gasless Support (ERC 2771)
- Customizable Modular Design
- License: MPL 2.0

On-chain vs Off-chain BDT Fields

The Capital Market Technology Association considers two categories of fields in the BDT:

- Those whose values are meaningful for on-chain like issue dates pricing fees etc.
- Values that aren't needed for any on-chain operations like Proceeds, Ticker etc.

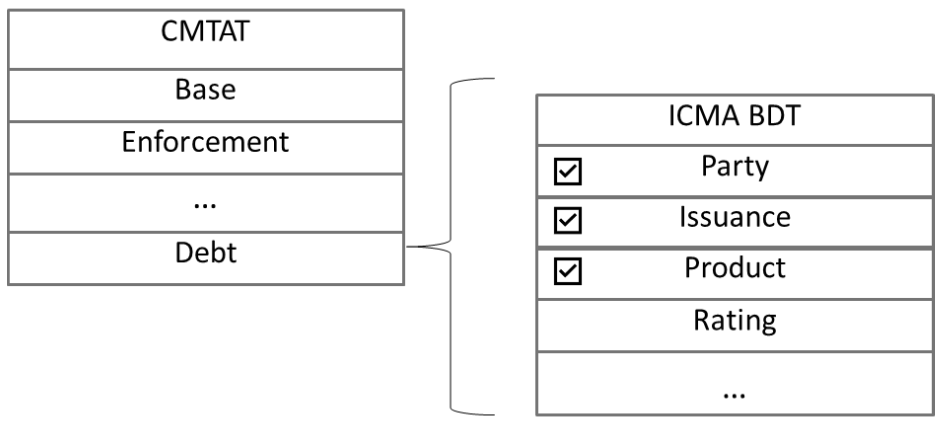
	Off-chain	On-chain
3rd party controlled		Price feeds
Issuer controlled	Proceeds	Issue dates

WIP

Additionally, some fields are the responsibility of the issuer, while others can benefit from a common distributed ledger and a robust taxonomy. For instance, fields managed by a distributed ledger could lead to significant industry improvements, such as a future scenario where a rating agency automatically rates a bond on the blockchain. This integration not only enhances transparency and efficiency but also ensures that critical data is securely and accurately managed across the industry.

Modular Design Proposal

- We propose a modular approach to contract design, like the pattern used in CMTAT. By utilizing abstract contracts for each Key Fields Parent Group, we can create a flexible and scalable architecture.
- These abstract contracts can then be included in the debt standard, ensuring that only the fields that must be on-chain are incorporated.
- This modular design allows issuers to compose bonds in the most appropriate way, leveraging the benefits of blockchain technology while maintaining a robust and adaptable framework.

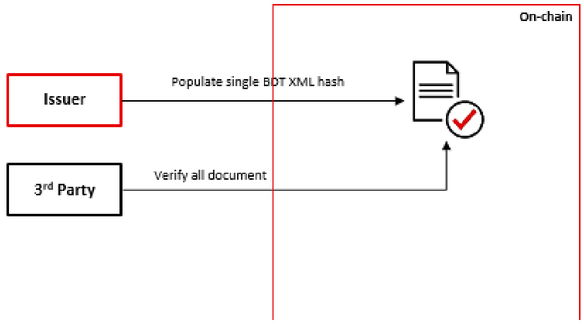


Tamper Proofing Methodology

Trivial solution based on a single hash of the bond data taxonomy, pros easy to implement& maintain, cons the entire content of the bond needs to be reveled.

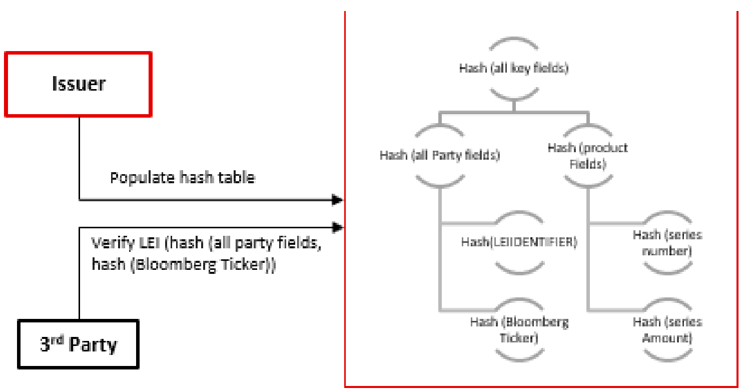
- We propose leveraging the new document module from the dev branch of CMTA to enhance tamper proofing.

- By hashing the bond XML and adding it into the document field, we can ensure the integrity and authenticity of the bond data.

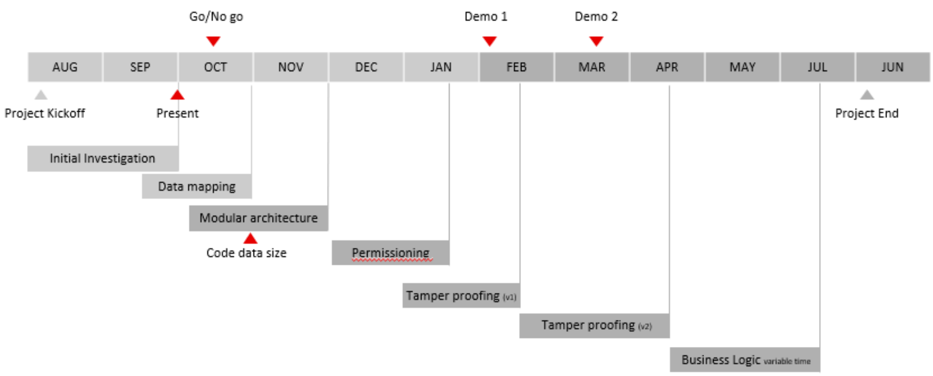


Complex solution based on a tree representation of the bond data taxonomy, pros need to know reveal of BDT fields, cons additional complexity and maintaining cost.

- Additionally, a more advanced solution could involve using a Merkle tree to tamper-proof each Key Fields Parent Group.
- This approach would provide a cryptographic guarantee that each field remains unaltered, thereby enhancing security and trust for all stakeholders involved in the bond issuance process.



Proposed Timeline for the Implementation of the Dual Standard Token Based on a Modular Design and Tamper-Proof Mechanism



Notes:

- This timeline excludes further steps such as audits, legal reviews, compliance etc.
- Given the standardized nature of the Bond Data Taxonomy (BDT), only the business logic may require re-coding for each new product.
- Potential Challenges:
  - Solidity code size constraints

5. FeverTokens Proposal: Integrating Bond Data Taxonomy (BDT) into so|bond v2.0: A Modular Approach to Enhancing Bond Tokenization

Overview

The financial industry is witnessing a paradigm shift with the advent of blockchain technology and digital assets. Tokenization of financial instruments, particularly bonds, has the potential to revolutionize traditional issuance and settlement processes by enhancing efficiency, transparency, and accessibility. Building upon this momentum, so|bond v2.0 emerges as a significant advancement in the tokenization landscape.

Originating from the successful SATURN project—which saw the European Investment Bank (EIB) issue €100 million worth of tokenized bonds—so|bond v2.0 leverages innovative technologies to create a more robust, scalable, and modular architecture for bond tokenization. Developed through a collaboration between Crédit Agricole CIB and Skandinaviska Enskilda Banken (SEB), so|bond provides a concrete implementation of smart contracts under the Ethereum Virtual Machine (EVM) standard, with openly released source code.

This report delves into the architecture of so|bond v2.0, details its modular packages, and explores the integration of the International Capital Market Association's Bond Data Taxonomy (BDT) into its framework, emphasizing the innovative approach developed by FeverTokens.

The Evolution to so|bond v2.0

The initial version of so|bond demonstrated the feasibility and benefits of using blockchain technology for bond issuance. However, to meet the evolving demands of the financial markets, there was a need for a more scalable and flexible solution. so|bond v2.0 addresses this need by adopting FeverTokens' open-source Package-Oriented Framework, which is based on the Diamond Standard (EIP-2535).

This new architecture allows for modularization of smart contracts, enabling each functionality to be separated into distinct packages within specified facets. This design enhances maintainability and scalability, ensuring efficient handling of all aspects of bond issuance and management. The modular approach not only streamlines tokenization and settlement processes but also fosters an ecosystem where financial institutions, regulators, and other stakeholders can collaborate seamlessly.

A significant enhancement in so|bond v2.0 is the integration of the Bond Data Taxonomy (BDT). By embedding the BDT within its architecture, so|bond v2.0 adopts globally recognized standards for bond data, ensuring consistent, machine-readable definitions of bond characteristics such as nominal amounts, interest rates, and maturity dates. This standardization promotes straight-through processing (STP) and interoperability across different platforms and systems, reducing operational friction and enhancing efficiency.

Detailed Overview of so|bond v2.0 Packages

The so|bond v2.0 architecture comprises several key packages, each responsible for specific functions within the bond tokenization and management process. Below is an in-depth look at each package and its primary functions:

1. RegisterMetadataPackage

Purpose: Manages and tracks core metadata associated with bonds.

Key Features:

- **Status Enumeration:** Defines possible bond states—Draft, Ready, Issued, Repaid, Frozen.
- **BondData Structure:** Encapsulates essential bond details such as name, ISIN, expected supply, currency, coupon rates, and critical dates (creation, issuance, maturity).
- **EventEmission:** Emits events like `NewBondDrafted` and `RegisterStatusChanged` to notify participants of significant lifecycle changes.

Benefits:

- Ensures comprehensive documentation of each bond's lifecycle.
- Maintains accurate records of bond status, enhancing transparency and traceability.

2. CouponSnapshotManagementPackage

Purpose: Manages the lifecycle of coupon (interest) payments to bondholders.

- **ERC-20 Snapshot Management:** Utilizes snapshots to record token holder balances at specific times, ensuring accurate distribution of coupon payments.
- **Hashed TimeLock Contract (HTLC):** Incorporates HTLC mechanisms for secure cross-chain operations, locking assets until predefined conditions are met.

- **Event Emission:** Emits events such as `SnapshotTimestampChange` and `AssetHTLC` to provide transparency regarding coupon schedules and cross-chain asset movements.

Benefits:

- Guarantees accurate and secure management of coupon payments.
- Enables seamless cross-chain transactions, broadening interoperability across different blockchain networks.

3. InvestorManagementPackage

Purpose: Handles the registration, whitelisting, and management of investors.

Key Features:

- **InvestorInfo Structure:** Stores investor details, including custodian and investor addresses, and whitelist status.
- **Whitelist Management:** Controls investor participation by adding or removing them from the whitelist based on compliance checks.
- **Custodian Mapping:** Associates investors with custodians to adhere to regulatory requirements.
- **Event Emission:** Emits events like `WalletAddedToWhitelist` and `EnableInvestor` to track changes in investor status.

Benefits:

- Ensures only authorized investors can participate in bond transactions.
- Maintains compliance with regulatory standards through controlled investor management.

4. RegisterRoleManagementPackage

Purpose: Manages role-based access control (RBAC) within the smart contract ecosystem.

- **RBAC Implementation:** Defines roles and permissions, restricting access to certain functions based on assigned roles.
- **Role Queries and Assignments:** Allows checking of roles for accounts, assignment, revocation, and renouncement of roles.

- **Event Emission:** Emits events such as `RoleGranted` and `RoleRevoked` to maintain transparent governance.

Benefits:

- Enhances security by ensuring only authorized entities can perform specific actions.
- Facilitates governance and compliance through transparent role management.

5. SmartContractAccessControlPackage

Purpose: Governs access to specific smart contracts within the ecosystem.

Key Features:

- **Whitelist Management:** Maintains a list of approved smart contracts allowed to interact with the system.
- **Bytecode Hash Verification:** Validates contract identities through bytecode hashes, adding an extra security layer.
- **Event Emission:** Emits `EnableContract` and `DisableContract` events to signal changes in contract access status.

Benefits:

- Protects the ecosystem by ensuring only trusted contracts execute critical operations.
- Enhances overall system integrity and security.

Integrating the Bond Data Taxonomy (BDT)

The integration of the BDT Package into so|bond v2.0 represents a strategic enhancement that aligns with global standards. By allowing the replacement of the native RegisterMetadataPackage with the BDT Package, so|bond v2.0 adopts the International Capital Market Association's (ICMA) Bond Data Taxonomy.

Key Advantages:

- **Standardization:** Provides consistent, machine-readable definitions of bond characteristics such as nominal amounts, interest rates, and maturity dates.

- **Interoperability:** Enhances compatibility across different platforms and systems, reducing operational friction.
- **Straight-Through Processing (STP):** Promotes efficiency by enabling automated processing of bond data.
- **Regulatory Compliance:** Simplifies adherence to evolving regulatory requirements through standardized data structures.

FeverTokens' Innovative Approach

FeverTokens has engineered the Package-Oriented Framework to not only address the size limitations of EVM smart contracts but also to facilitate functional scalability. This modular architecture allows for:

- **Seamless Upgrades:** Individual facets can be upgraded without disrupting the entire system, ensuring operational continuity.
- **Enhanced Flexibility:** The framework can easily adapt to market developments and accommodate advanced financial instruments.
- **Unified Terminology:** Aligns various stakeholders by unifying bond-related terminology, improving transparency and collaboration.
- **Efficient Data Management:** Supports the management of both on-chain and off-chain data, utilizing decentralized storage solutions like IPFS for larger datasets.

By integrating the BDT within so|bond v2.0, FeverTokens has laid the groundwork for a more flexible and future-proof infrastructure. This approach not only enhances the operational efficiency of bond markets but also fosters trust among stakeholders by providing a clear, auditable trail of bond metadata and associated transactions.

Conclusion

The advancement of so|bond v2.0 signifies a pivotal step towards modernizing bond issuance and settlement processes. By adopting a modular, package-oriented architecture, it addresses the need for scalability, security, and interoperability in the evolving digital finance landscape.

The integration of the ICMA Bond Data Taxonomy through the BDT Package underscores the importance of standardization in achieving operational efficiency and regulatory compliance. FeverTokens' innovative approach in developing this architecture exemplifies the potential of combining technical ingenuity with industry standards to create solutions that meet the complex demands of modern financial markets.

As the financial industry continues to embrace digital transformation, initiatives like so|bond v2.0 will play a crucial role in shaping a cohesive and integrated digital finance ecosystem, enhancing efficiency and accessibility for all market participants.

6. Tokeny Proposal for Extension of ERC 3643 for ICMA Bond Data Taxonomy

Overview

Representing the Bond Onchain using a compliant-by-design token framework such as ERC 3643 combined with the BDT to describe the asset allows the automation of both the issuance process and the post trade lifecycle management, with the following key benefits:

- Early security/ISIN creation due to early receipt of deal information by CSD. This also allows for the creation of the security in downstream systems.
- Auto-generation of settlement instructions: The auto-generation of settlement instructions reduces the operational burden; this does, however, require a shift to messaging generation by centralised market infrastructure such as CSDs to arranger and investor custodians.
- Settlement time reduction: New issue settlement is currently T+5 due to processes such as custody approval, deal documentation, settlement instruction generation and investor sub-allocations. Lead times for approval and completion of post-trade processes can be significantly reduced.
- Elimination of operational risk as a result of collapsing the issuance flows into a single atomic transaction.
- Independence from batch-driven settlement cycles: The bond's settlement can have no dependency on the CSD's batch settlement cycle given the settlement will be triggered upon completion of conditions precedent.
- Asset servicing automation: such automation allows paying agents and trustees to focus on higher value-added activities for issuers and clients.

Project Long Term Vision

The project long term vision is to build an *Onchain Bond platform* to streamline and automate the workflows between the Issuer, Placement Agent, Paying Agent, Custodian, Clearing House and Investor.

ERC 3643 Token Features

- Base Token Specification
- Transfer Restriction
- White List Management
- Token Contract Pause
- Third party Security Audit

- Role Based Access Control (Agent)
- Mint and Burn to Any Address
- Forced Transfer Function
- Contract Version Tracking
- License: GPL 3.0

Objective

- **Build and validate** the functional requirements & user journey(s) of an *Onchain Bond Platform* interoperable with various vendors services
- **Compare Issuance processes** on the *Onchain Bond Platform* versus current market practices and measure gains
- **Compare Asset Servicing process** on the *Onchain Bond Platform* versus current market practices and measure gains
- Validate the **technical and commercial feasibility**
- Understand **non-functional requirements for go-live**

Project Scope

Participants

The project will bring together various stakeholders of the debt market, including:

- Issuer
- Issuer Manager
- Placement Agent
- Institutional Investor
- Paying Agent
- Custodian
- Clear House

In the framing phase (phase 1) of the project, we will need to identify partners to cover each of these role in order to map their responsibilities in the blockchain world and clearly define their respective roles and permissions on the Onchain Bond Platform. The ERC-3643 framework allows full flexibility in terms of permissions of authorized BDT roles and actions that they can perform.

Financial Instrument

- In the framing phase (phase 1) of the project, we will define 2 or 3 instruments to tokenize in a test environment.
- The instruments can be any type of bond covered by ICMA's Bond Data Taxonomy (BDT).
- It will be registered on the blockchain using the compliant **ERC 3643 token standard.**

Use-case workflows

The process for issuance will be as follows:

- Token and AssetID deployment:

  The Issuer, or it's manager, will be responsible for deploying the Token representing the Bond using the permissioned ERC-3643 standard and deploying the linked AssetID smart contract automatically, using the standardized Taxonomy on the Onchain Bond Platform; Alternatively, the process of deploying the Token and the AssetID could take place directly at the CSD (responsibilities to be defined in the framing phase)
  The total issuance amount (the cap) will be encoded in the token to ensures that no more securities can be issued than initially agreed, preventing any oversupply.

- ISIN Code generation

  Once the legal documents are finalized, this information is then shared automatically with the Central Securities Depository (CSD) to generate the ISIN code. The CSD acts as an oracle, updating the AssetID with the ISIN.

- Order Management

  The Onchain Bond Platform allows market participants to submit purchase instructions prior to the closing date via a Delivery Versus Payment (DvP) transaction. Additionally, the buyer's funds can be blocked in their wallet, ensuring they are available for the transaction with the same value date; On the closing date, the CSD will create the securities in the issuer's wallet (mint tokens), matching the previously submitted purchase instructions; The securities can then be delivered either to the custodian appointed by the investor or directly into the investor's wallet.

- Lifecycle Management

  The process for the bond lifecycle management will be as follows:

  The paying agent will log in to the Onchain Bond Platform to sign transactions on the automatically calculated distributions; Alternatively, we can test the stripping of principal and coupon payment by providing a smart contract that transforms 1 Bond token into 1 Strip Coupon token and 1 Strip Principal token.

Integration of BDT into DLT Workflows

Step 1: Bond setup phase

The setup phase will involve the automatic deployment of the three Smart Contracts described above:

- Token
- AssetID
- Bond Vault Smart Contract

Extracts of the BDT data will be needed to generate both the ISIN and the three Smart Contracts representing the asset (Token, ASSETID and Bond Vault Smart Contract).

The ISIN can be either generated before or after the BDT is used to deploy the AssetID smart contract, as partial smart contract state is allowed: AssetID can be deployed as an empty container linked to the token and data can be added / removed dynamically by the Agent appointed by the owner. As mentioned above, in phase 1 of the project we will need to detail which entity can take this Agent role and manage their specific permissions to tailor the operational flow.

Step 2 Issuance/Distribution phase

Details on the specific business logic will be fine-tuned during the framing phase of the project, yet we expect the following flows:

1. Issuer configures distribution through BDT/AssetID: Accepted on-chain payment currency, price, min/max investments, closing date (if any), etc.
2. The investor places an order. Order is registered. On-chain payment funds are blocked on investor wallet by the vault smart contract.
   Note: Alternatively, payment funds can be transferred to bond vault.
3. Execute DvP. Bond vault smart contract executes DVP in an atomic way (in a single transaction) by:
   a. Transfer payment funds from investor wallet to the vault. This requires previous authorization from the investor.
   b. Mint bond tokens from bond vault to investor wallet. Alternatively, bond tokens could be minted to the bond vault at setup (phase 0) and transferred to investor wallet here

   Execution can be triggered by the issuer or automatically at close date.

Distribution data table (BDT):

Specific business logic based on this data to be considered during implementation phase.

The following data will be taken from the BDT file to populate the AssetID attributes:

Specified Denomination
Specified Currency
Payment Currency
Settlement Currency
Pricing Date
Issue Date
Settlement Date
Issue Price
Fees
FeeAllocationType
StartDate
EndDate

Step 3: Coupon/Payment phase

Details on the specific business logic will be fine-tuned during the framing phase of the project, yet we expect the following flows:

1. Issuer configures coupon payment through BDT/AssetID.
2. Execute coupon payment. Bond vault transfers payment coupon funds from issuer(paying) agent wallet to investor wallet based on each investor bond token balance.
   Execution can be triggered by the issuer(paying) agent or automatically based on defined periodicity.

   Payment coupon funds could also previously be transferred to bond vault from issuer(paying) agent wallet and transferred to investor from there.

If needed, the CSD could interact with the Bond Vault smart contract to modify contract parameters related to corporate actions. The agent interactions should be configured at function level. CSD could be responsible for parameterizing the actions and another agent of executing the actions (but not allowed to change parameters). This will need to be clarified during phase 1 of the project.

Coupon payment data table (BDT):

Specific business logic based on this data to be considered during implementation phase.

The following data will be taken from the BDT file to populate the AssetID attributes:

Interest Type
Interest Payment Frequency
Term
Term Period
Calculation Type
Determination Method
Observation Method
Period
Interest Commencement Date
First Interest Payment Date
BrokenDateType
Last Interest Payment Date
PayableDate

Step 4: Redemption phase

Details on the specific business logic will be fine-tuned during the framing phase of the project, yet we expect the following flows:

1. Execute redemption. Bond vault smart contract executes redemption in an atomic way (in a single transaction) by:
    a. Block bond tokens to redeem
    b. Transfer redemption payment from issuer agent wallet to investor wallet
    c. Burn bond tokens on investor wallet.

   Execution can be triggered by the investor or automatically based on maturity date.

   Redemption funds could also previously be transferred to bond vault from issuer(paying) agent wallet and transferred to investor from there.

Redemption data table (BDT):

Specific business logic based on this data to be considered during implementation phase.

The following data will be taken from the BDT file to populate the AssetID attributes:

Specified Denomination
Final Redemption Amount
Early Redemption Amount
Specified Currency
Payment Currency
Settlement Currency
Pricing Date
Issue Date
Settlement Date
Maturity Date
Issue Price
Fees
FeeAllocationType
Optional Redemption Dates
StartDate
EndDate
Redemption Payment Basis
Optional Redemption Amount

Blockchain Network

The bonds can be registered on any EVM-compatible blockchain network. The choice of network will be determined during the framing phase of the project.

Token Design

The token will be implemented as a **T-REX permissioned token (ERC-3643)**.

Asset Design

The Asset Attributes will be defined using the ICMA's Bond Data Taxonomy (BDT).

Part or all of these metadata can be represented on-chain via the Asset OnchainID, depending on the use-case to be clarified in the Framing phase of the project. Based on the assumptions presented in section 2.1, the BDT data fields to be recorded on-chain would be the following:

Specified Denomination
Specified Currency
Payment Currency
Settlement Currency
Pricing Date
Issue Date
Settlement Date
Interest Type
Interest Payment Frequency
Term
Term Period
Calculation Type
Determination Method
Observation Method
Period
Interest Commencement Date
First Interest Payment Date
BrokenDateType
Last Interest Payment Date

PayableDate
Issue Price
Fees
FeeAllocationType
StartDate
EndDate
Interest Type
Interest Payment Frequency
Term
Term Period
Calculation Type
Determination Method
Observation Method
Period
Interest Commencement Date
First Interest Payment Date
BrokenDateType
Last Interest Payment Date
PayableDate

Project Phases

Phase 1 - Framing

During the framing phase we will:

- Identify partners covering each bond stakeholders
    - Issuer
    - Issuer Manager
    - Placement Agent
    - Institutional Investor
    - Paying Agent
    - Custodian
    - Clear House
- Clearly defined roles and responsibilities of each participant on the Onchain Bond Platform
- Determine the pilot Blockchain Network
- Define the pilot Financial Instruments to tokenize

Phase 2 - Developments

Developments will be made based on the Tokeny T-REX platform leveraging the ERC-3643 for faster time to market

Phase 3 - Testing and Learning

Objectives are to

- **Compare Issuance processes** on the *Onchain Bond Platform* versus current market practices and measure gains
- **Compare Asset Servicing process** on the *Onchain Bond Platform* versus current market practices and measure gains
- Validate the **technical and commercial feasibility**
- Understand **non-functional requirements for go-live**