White Paper: A Unified Metadata Standard for the TDM Framework

Version: 1.1

Author: Gemini Agent (as a collaborative soundboard)

Abstract

This document proposes a standardized, hierarchical tag ontology designed to bring consistency, searchability, and scalability to the Token Decoder Maps (TDM) framework. The current ad-hoc creation of tags and the use of separate `Type` and `Category` fields leads to organizational debt and ambiguity. The proposed solution is to deprecate these fields and unify all metadata under a single, powerful `Tags` field that utilizes a prefix-based, controlled vocabulary. This directly aligns with the TDM philosophy of "Precision over Ambiguity."

---
1. The Problem: The Inefficiency of Uncontrolled Metadata

An uncontrolled approach to metadata, using ad-hoc tags alongside separate `Type` and `Category` fields, inevitably leads to inconsistency. This results in:
* **Redundancy:** The same information is often captured in multiple fields.
* **Poor Searchability:** It becomes impossible to reliably find all related tokens with a single, simple query.
* **Cognitive Overhead:** The user must constantly decide which field to use for which piece of information.
* **Inconsistent AI Output:** An AI agent generating tokens will produce varied and unpredictable metadata without a clear standard.

---
2. The Solution: A Unified, Prefixed Tag Ontology

The proposed solution is to implement a **controlled vocabulary** and unify all metadata under a single `Tags` field. This system organizes tags into distinct, logical categories using a prefix and provides a single source of truth for all token metadata.

The official recommendation is to **deprecate the `Category` and `Type` fields entirely**. The new, prefixed `Tags` field is more powerful and explicit, capable of handling both primary classification (e.g., `#type/project-task`) and secondary context.

Example of Refactoring:

Before:

::MX-PROJECT-TASK-ID::
- **Title:** Refactor the Parser
- **Type:** Refactor
- **Category:** #TDM
- **Tags:** [#API, #Core]

After:

::MX-PROJECT-TASK-ID::
- **Title:** Refactor the Parser
- **Tags:** [#type/refactor, #project/tdm, #topic/api, #topic/core]

---
3. Proposed Tag Categories

The ontology is built on several primary prefixes:

* **`#status/`**: Describes the current state of a note or task in a workflow (e.g., `#status/todo`, `#status/in-progress`).
* **`#type/`**: Defines the fundamental nature of the content (e.g., `#type/project-task`, `#type/whitepaper`, `#type/en-token`).
* **`#project/`**: Associates a note with a specific, high-level project (e.g., `#project/tdm`, `#project/rsis`).
* **`#tech/`**: Denotes a specific technology or tool (e.g., `#tech/python`, `#tech/obsidian`).
* **`#topic/`**: Describes the general subject matter (e.g., `#topic/ai`, `#topic/project-management`).

---
4. Implementation Guide

1. **Create a Canonical Document:** Establish a `tag_ontology.md` file that lists all approved tags and their categories.
2. **Update Token Templates:** Remove the `Category` and `Type` fields from all official TDM token templates.
3. **Integrate with AI Instructions:** Update the AI's core directives with a rule to only use tags from the canonical ontology within the single `Tags` field.
4. **Refactor Existing Tokens:** Gradually refactor the metadata in existing tokens and notes to align with the new, unified system.

---
5. Conclusion

By implementing a hierarchical tag ontology and unifying all metadata under a single `Tags`

field, the TDM framework becomes significantly more powerful, searchable, and coherent. This structured approach applies the principle of "Precision over Ambiguity" to the very foundation of the token-based system.