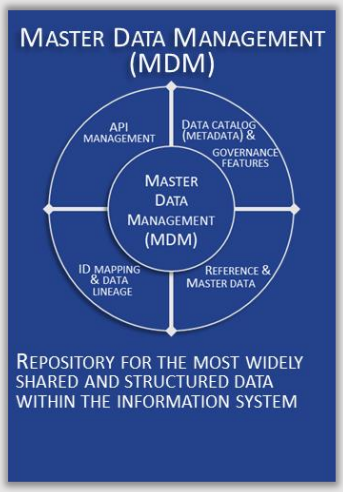
MASTER DATA MANAGEMENT

Master Data Management (MDM) serves as a repository for the most widely shared and structured data across the information system. It is particularly important for AI at scale, as it plays a crucial role in creating ontologies in conjunction with the Operational Data Store (ODS).



1. CONDITIONS OF SUCCESS

Master Data Management (MDM) offers advanced data governance features such as version and variant management, temporal management (historical), version comparison and merging, data deduplication, data cleaning, data authoring UI, etc. The richer this governance is, the less feasible it is to apply it to data that is frequently and massively (OLTP) modified. Therefore, master and reference data are primarily concerned with MDM.

For instance, the stock of a product in a company's offer catalog evolves in real-time with the flow of orders. However, the physical locations of these stocks in warehouses remain stable over a predetermined period, such as a day, week, or longer. MDM does not manage stock values for each order but handles data concerning their warehouse locations. This is a meta-knowledge applied to the concept of stock. Specifically, MDM manages the metadata of the business concept of "stock" (name, format, nature, application linkage, etc.) without knowing the successive stock values of products. Conversely, for product storage locations, MDM manages both the metadata of associated business concepts (warehouses, geographic location) and the values with warehouse instances and their physical addresses.

The previous example highlights two principles essential for establishing a minimum architecture to scale Artificial Intelligence:

- Metadata is indispensable for describing business concepts used by the company in a unified manner without semantic ambiguities, regardless of their formats, nature, and life cycles: Format: integer, character string, video, sound, multimedia; Nature: operational, decision-making, governance; Life cycle: update frequency.
- The richer the data governance features, the more their usage is limited to long-life cycle data. This mainly concerns the most shared data in the company, namely reference, master, and metadata. This limitation results from technical constraints and the commitment of data management teams (data stewards) whose role is to work on the most shared data within the company. Most of the time, it is the MDM that provides these rich governance features.

In other words, MDM enhances the quality of the most shared data in the information system, which: Carries the core business referential integrity rules; Is used for data consolidation at the reporting level; Is deeply integrated into operational processes.

These data, and thus the underlying business concepts they embody, cannot be managed in silos without risking semantic discrepancies that compromise quality.

The goal is not just to consolidate the most shared data to create a single access point, like an ODS. The objective is also to manage their updates in compliance with globally applicable governance rules within the company. These updates are then reflected in the consuming applications.

To better understand the importance of MDM, here is a list of metadata that passes through its governance:

1. Identifiers of business concepts (sometimes called business objects or strategic data elements) and the relationships between them, describing the mesh between business concepts in the form of taxonomies and a semantic model.
2. The nomenclature (or identity card) of each business concept: name, description, creation date, modification date, and other widely shared data between applications.
3. The life cycle of each business concept. For example, the business concept Customer can follow this life cycle: prospect, new customer, active customer, passive customer, former customer, closed customer. Depending on the state of the business concept, integrity rules are declared to frame updates (governance).
4. The company's business glossary.

This set of metadata embodies a unified data model independent of the specific data structures of existing applications. Therefore, it is in the MDM that the necessary ontologies for anchoring with AI, as recommended in the TRAIDA framework, are found.

This MDM should serve as a launchpad for creating your ontologies, loading them from your applications, updating them by users in charge of governance (data stewards), and resynchronizing them with your application systems.

Here, you need to develop your roadmap to clarify the collaboration between MDM, ODS, and the metadata catalog on core system data (see respective TRAIDA sheets).

In the scenario where MDM and ODS share the same technological solution, they can then be integrated or merged into the same tool. Here are two possible cases:

- ODS in a knowledge graph database with a no-schema MDM in relational database: integration needs to be planned between the two, and a pivot repository for ontology management should be chosen.
- ODS and MDM in a same knowledge graph database: fusion is possible.

The main obstacle to fusion is the lack of governance functions in the ODS to fulfill the role of an MDM. This governance must be exercised not only at the metadata and data levels but also at the underlying data model level, i.e., ontologies. Deploying ontologies without properly managing their versions over time is not recommended. As applications evolve independently of ODS and MDM repositories, change management is required to ensure proper synchronization of the systems in place.

Finally, attention should be drawn to the opportunity of using knowledge graph technology for implementing MDM. The advantage of this technology is the possibility of automatically obtaining an initial ontology version from data sources, with AI support for better interpreting the flows. This capability to avoid modeling work is an attractive aspect of the "free-schema" brought by graph databases (see also TRAIDA card on core system data). However, the semantic accuracy and long-term solidity of automatically generated ontologies are not the best. Indeed, semantic accuracy does not come from existing data flows since they are altered by accumulated poor designs over time (technical debt) and accompanying semantic ambiguities. Instead of starting too quickly with a "free-schema" approach, it is better to go through specific semantic modeling work and consider automatically created ontologies as drafts of a target version to be built. Therefore, it is uncertain that knowledge graph technology is the best option for MDM, especially since this repository requires flawless OLTP management. The alternative is to use a "no-schema" database technology backed by a relational DBMS that ensures better referential integrity and OLTP management than graph technology.

Why is MDM important for AI?

Ontologies form the semantic backbone from which your AI training and enrichment (RAG) processes must be built. You should not integrate your AI directly with heterogeneous data sources from applications. There would be a significant risk of letting quality defects in the data flows, which would cause errors in AI systems without even being able to trace the source of these defects (biases, hallucinations, bugs). The use of ontologies imposes an effort to clarify and clean your data and promotes decoupling between your data and AI.

For example, consider an AI system that needs access to all knowledge about your customers. Without MDM/ODS systems and ontologies, the AI would have to query a series of heterogeneous applications and databases to find customer data, with risks of semantic ambiguities and poor quality due to technical debts accumulated over time in these systems. Conversely, with MDM/ODS, the AI system directly accesses a single, reliable point that provides, through ontologies, a high-quality data source.

Integrating a data mesh strategy

Data mesh is a data architecture that aims to break down silos to organize databases by business domain (see TRAIDA core system data card). Therefore, there should no longer be data duplication and poor data quality. In practice, the roadmap to transition the entire information system to a data mesh can take several years, providing valuable time for MDM. Moreover, unless the different data mesh databases can offer cross-functional and shared governance functions, a central mechanism for managing this governance will still be necessary. The longevity of MDM is even more evident when different database technologies are used to deploy the data mesh. In this case, it is unlikely that these databases can share governance without an MDM acting as a pivot.

Considering multimedia data

MDM handles structured data and integrates multimedia data through links to specific storage areas, ideally a graph database repository (see TRAIDA card about Enterprise Knowledge Graph), big data, or a cloud storage server like Google Drive.

Complement on ODS and MDM integration

When using MDM as a pivot repository for managing metadata, reference, and master data, it is possible to use complementary data access virtualization technology to retrieve operational data located in the ODS for reading purposes.

2. IMPORTANCE OF THIS CARD FOR YOUR TRANSFORMATIVE AI

AI systems are initially trained on large volumes of data, where semantic structuring is not fundamental. However, the subsequent fine-tuning processes involve smaller volumes of data that require increased semantic mastery to achieve relevant results. At the most detailed level of this training, it may involve real-time access to specific business concept information located in a database (RAG: Retrieval Augmented Generation). At this access level, it is crucial to have metadata describing this business concept.

Thus, without effective metadata management across the entire information system, it will be impossible to fine-tune AI systems, leading to disappointing results. This is a fundamental reason for the shift from big data, which lacks semantic management, to ODS and MDM systems that rely on powerful semantic management.

If this observation seems relevant to your context, you should establish a roadmap for systemic metadata (ontology) management. It is not just about creating a metadata catalog to understand your existing data (see TRAIDA core system data card) but about building your minimum semantic model to gradually scale your AI strategy. You have two possible approaches:

1. Operational Data Store (see TRAIDA ODS card): This goes beyond metadata management by handling all operational data. However, its lack of governance functions may prevent its use as a pivotal metadata repository. It is crucial to manage the lifecycle of ontologies to avoid failures in scaling due to misalignments between the ODS and applications.
2. Master Data Management: Specialized in managing metadata, reference, and master data, MDM's governance functions are more powerful than those of the ODS, giving it an advantage as the company's central metadata catalog. It must be capable of managing the lifecycle of ontologies to ensure synchronization with applications.

Therefore, MDM is a pivotal element in building the semantic management platform necessary for scaling your AI systems. Depending on your context, you will need to determine your own roadmap to synchronize the ODS, MDM, and Enterprise Knowledge Graph (EKG).

REFERENCE AND MASTER DATA

Reference data includes the codifications used in your applications. Some are standardized, such as country codes, others are industry standards, and some are specific to your context. They often follow a simple structure like code and label.

Master data refers to the identity cards of your business concepts. First, you need to list these concepts to build a catalog and then a glossary. Each identity card consists of the most stable and shared data among applications. For example, for the business concept Customer, the master data would include: identifier, name, surname, address, email, date of first purchase, status (payment in progress, payment OK, pending validation, no longer active, archived...), and relations to other business concepts (Product, Sales, Billing, etc.).

ID MAPPING AND DATA LINEAGE

Each business concept referenced in the MDM is fed by source applications and synchronized with consumer applications, which can be the same or different. Typically, a primary (or master) source application is designated as responsible for the main ID in the ID mapping that needs to be constructed to reference all source and consumer applications. The materialization of this data mapping in the MDM (ID mapping) can take various forms, often involving metadata.

Accumulating the identifier mappings of different business concepts allows us to identify chains in their usage across applications. For example, a Product Management application sends an update of product descriptions to the Sales and Marketing applications. In this case, there is a chaining for the Product business concept that starts with the Product Management application and links to the other two applications, Sales and Marketing. This formalization of knowledge enhances the semantic scope in the MDM and improves governance. For instance, it could be decided that the MDM is responsible for directly propagating product description changes to the target applications.

DATA CATALOG (METADATA) & GOVERNANCE FEATURES

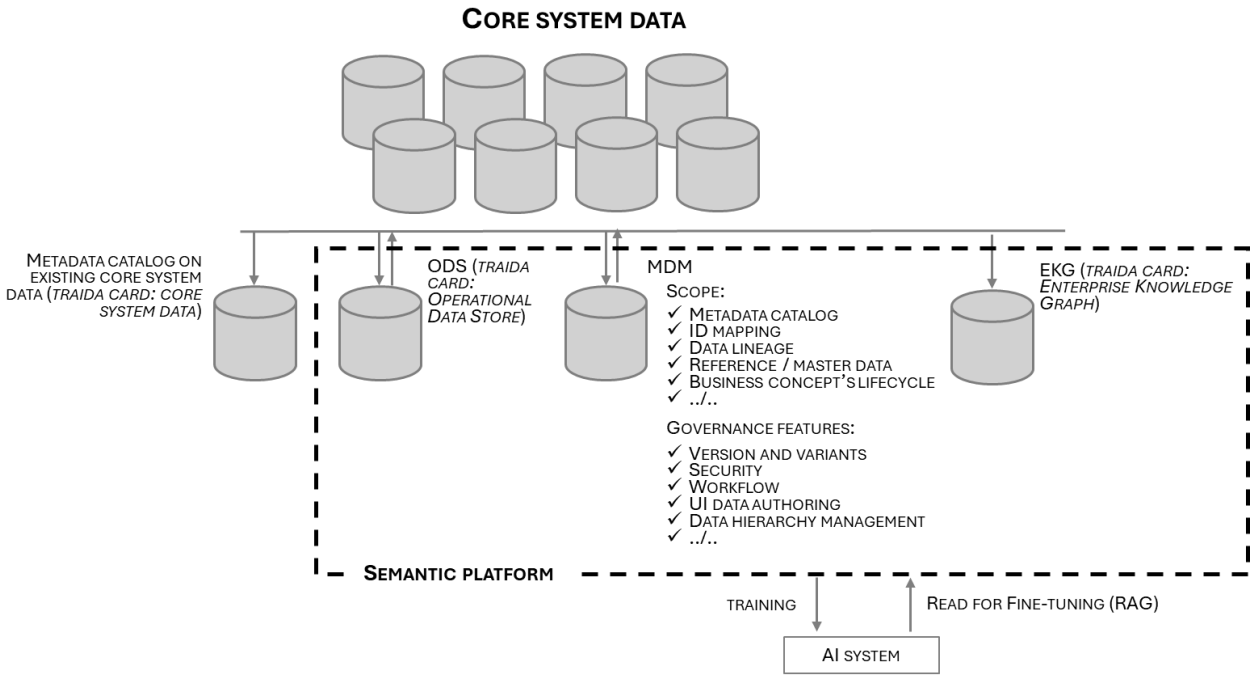
As mentioned earlier and reiterated several times, metadata management is crucial for gradually scaling your AI systems. This is an essential part of establishing a semantic management platform as recommended by TRAIDA. The data catalog, and more specifically the metadata catalog, is better managed in the MDM due to its powerful governance features. Here is a non-exhaustive list of these functions: Data model versions with data migration functions between models; Data spaces and data sets by variants and versions; Data update screens automatically available from data models and customizable as needed; Data hierarchy manipulation; Data update and validation workflows; Security; History; Archiving; And more.

API MANAGEMENT

API management involves documenting and configuring service contracts used for interactions between applications. A service contract is a business concept whose identity card consists of the description of the service's purpose, its input and response parameters, configuration possibilities, etc. Instead of leaving this information solely in technical documentation (e.g., Javadoc), it is beneficial to elevate it as metadata within the MDM to manage their life cycles.

For example, consider an API for retrieving a product sheet described in a Javadoc. This description is brought into the MDM to reference the consumers of this API: the Sales application uses it with a filter that only retrieves pricing data, while the R&D application uses it with a filter that only provides technical data.

3. BLUEPRINT



4. YOUR SITUATION & OBJECTIVES