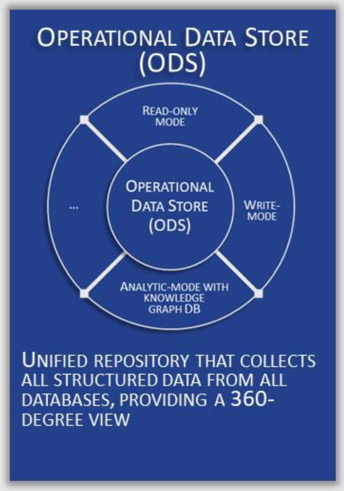
OPERATIONAL DATA STORE

The Operational Data Store (ODS) is a unified repository that collects all structured data from all databases, providing a 360-degree view. In practice, a read-only ODS can cover just one functional or business domain of the enterprise to build a unified view of data within this limited scope.



1. CONDITIONS OF SUCCESS

History

Since the beginning, information systems have gradually structured around multiple data sources. These systems generate information quality issues due to duplications and complex relations between objects stored in these different sources.

In the early 1990s, the need for a unified repository to consolidate these sources into a single point emerged. At that time, it was about preparing data downstream from business intelligence repositories like data warehouses. In this context, the term Operational Data Store (ODS) became widespread. It didn't introduce new storage technologies since the use of relational databases was the norm. It was used as a new data source exclusively for consultation in business intelligence. Although its data model needed to be properly constructed, it was not yet a semantic modeling. It was just necessary to ensure an organized structure of data for their use in decision-making systems, in a context where data warehouses presented significant constraints for the volumes of data managed.

A few decades later, the emergence of massive data storage technologies with big data made the use of ODS less useful: why spend money on this repository when it was possible to dump all data sources into big data? Unfortunately, experience showed that the lack of data structuring in big data harms the quality of analyses.

Today, many companies are dissatisfied with their big data projects partly due to the absence of an ODS upstream of decision-making systems. This results in a lack of semantics in big data that prevents leveraging the deep richness of data.

In parallel with the deployment of big data, the ODS survived outside the needs of decision-making systems, under different names and in a manner limited to certain business or functional domains. The most common are CDI (Customer Data Integration), PIM/PLM (Product Information Management / Product Lifecycle Management), and to some extent MDM (Master Data Management).

The return of the ODS

In this context of losing data meaning in decision-making systems, generative AI seems to offer a miraculous solution to regain meaning in data repositories, whether structured or not. Unfortunately, two new problems arise:

1. The use of AI on decision-making data sources (big data) is not sufficient since the company generally wants to leverage operational data in all its extent to train AIs, with the most accurate freshness level and sometimes in real-time for certain use cases.

2. AI needs information about the meaning of data in their usage contexts to reduce biases, analysis errors, or hallucinations. The more AI systems spread into the company's operational processes, the more these data interpretation flaws become unacceptable and can lead to significant value losses.

To counter these two problems, setting up an operational data repository upstream of AI systems usage becomes a necessity. In other words, AIs draw their data from this repository, whose quality, depth of details, and semantics are sufficient to build systems that make AIs more reliable in all usage contexts. Thus, it marks the strong return of the ODS.

Implementing the ODS

Once the interest in the ODS is understood, choosing its implementation can be delicate due to existing IT systems that already use repositories like CDI, PIM/PLM, or MDM. Some characteristics of the Operational Data Store might appear redundant with these repositories. These architectural decisions depend on each company's context, but you should follow these initial principles:

- The construction of the ODS must aim for coverage of the entire enterprise scope, meaning all business concepts like products, customers, organizations, manufacturing processes, etc. It's often necessary to start a first version of the ODS on a limited scope corresponding to a business or functional domain. However, its future extension should be planned from its foundations to ensure the establishment of a minimum viable architecture to scale. Since the ODS serves to train and feed your AI systems on all your operational processes, it must cover all the company's business concepts.
- Be careful not to create ODSs in silos by favoring short-term agility for a project over a global technical solution associated with enterprise-level modeling work. Indeed, heterogeneous ODSs encourage data duplications and semantic divergences with associated problems.
- The ODS should rely on the metadata catalog of core system data (see the TRAIDA map that covers this area). This ensures that the ODS's semantics are sufficiently solid even if you have a modeling know-how deficit. The ODS's projection on the entire information system scope requires relevant and robust data modeling.
- Depending on whether the ODS usage is read-only or write-mode, information storage technologies can vary between relational databases and knowledge graph-oriented solutions. We will revisit the issue of unstructured data storage later in this document.

With these fairly simple initial principles, you should study the synchronization rules with existing repositories mentioned earlier. It is not relevant to integrate an ODS into your architecture if you haven't clarified the integration rules with your CDI, PIM/PLM, and MDM.

2. IMPORTANCE OF THIS CARD FOR YOUR TRANSFORMATIVE AI

If you do not have a good understanding of your data, it is difficult to train AI systems correctly and connect them to your real-time information sources (RAG integration). Therefore, you need to have a data catalog with up-to-date and reliable contents. Two data repositories need to be made available to the AI:

1. Structured Data: Mostly from your operational applications, primarily your backends, which TRAIDA refers to as core system data.
2. Unstructured Data: This includes multimedia data, encompassing various formats such as images, sounds, videos, documentation, emails, etc.

The first repository is handled using a metadata catalog on core system data (see the TRAIDA core system data card) coupled with the implementation of an ODS (the focus of this document). The second repository requires the use of a knowledge management repository, addressed in the form of an Enterprise Knowledge Graph (see the relevant TRAIDA card).

The use of the ODS can be considered in three modes, which we will describe in the following sections.

READ-ONLY MODE

The read-only ODS feeds AI systems and does not accept update flows in return.

More specifically, ODS data is exclusively updated by injection flows from source systems. Applications and AI do not directly update it. The feedback from the ODS to the source systems is limited to data cleaning and quality improvement decisions. ID deduplication and data normalization mechanisms should rely on specific technologies outside the ODS, typically within the scope of MDM (Master Data Management).

Since this ODS operates in read-only mode, using knowledge graph-oriented database technology is pertinent. This approach allows for reusing the metadata catalog related to core system data (see the corresponding TRAIDA card). This "schema-free" system offers great flexibility in implementing the ODS data model. Additionally, as it is not intended to support updates, the level of semantic modeling can remain quite basic. For example, there should be no complex integrity control rules to model.

Given that the repository is read-only, it is feasible to develop multiple ODSs for isolated business or functional domains without the risk of introducing update silos with associated data duplications and quality issues. However, if the future goal is to implement a write-mode ODS, careful consideration is needed regarding the use of this flexibility; isolated ODS repositories would need to be deconstructed and integrated into the write-mode ODS.

Unstructured data can be incorporated into this knowledge graph-oriented repository. Alternatively, integration with a specialized knowledge management repository, itself based on graph database technology, can be considered (see the TRAIDA Enterprise Knowledge Graph card).

Governance processes for data injection flows are necessary to manage versions and data landing zones.

Finally, since this ODS is limited to providing data without direct updates, a simple user interface for consulting data sets by business concepts is generally sufficient. For data analysis needs, see the next topic, "Analytic-mode with knowledge graph DB."

WRITE-MODE

The write-mode ODS follows the recommendations already described for the read-only ODS, but introduces differences that we will now list.

First, the use of knowledge graph technology is no longer as obvious as for the read-only ODS. This ODS accepts data updates directly from application systems and AIs, giving it greater responsibility within the overall information system but also increasing the requirements for data quality control, especially regarding integrity constraints. Additionally, the frequency of transactional update flows may necessitate specific database technologies. The best solution between relational database technology and knowledge graphs must be decided. Although relational databases are often the best choice for repositories with intense updates on structured data (OLTP), two disadvantages should be considered:

1. Since this is not a "schema-free" approach, creating relatively rigid data schemas is required, which comes with stronger modeling quality demands. This point turns into an advantage in the long term by ensuring the durability of data structures and avoiding the trap of poor agile iterations in a "schema-free" environment, which sometimes leads to chaotic solutions.
2. The difficulty of handling unstructured data. The solution here is to integrate directly with a dedicated knowledge management technology for multimedia data (see the TRAIDA Enterprise Knowledge Graph card).

The modeling effort for a write-mode ODS is more significant than for read-mode alone. It requires modeling complete ontologies based on a glossary and taxonomy of business concepts shared across the enterprise. Although these semantic mechanisms are also necessary for Master Data Management (MDM), the ODS does not replace MDM for at least two reasons:

1. Data governance features remain simple with the ODS, as its goal is to provide data to AIs rather than managing the data itself.
2. Unlike MDM, which offers rich governance features for business but whose scope is limited to the most shared data in the enterprise (reference and master data, see TRAIDA card on MDM), the ODS specializes in managing operational data.

This point about MDM encourages studying the integration with write-mode ODS: the ODS specializes in managing operational data, and the MDM governs the most widely shared data in the enterprise. Both repositories then share responsibility for certain integrity rules, which should be adjusted in your context.

It should also be noted that deploying a write-mode ODS across multiple isolated functional or business domains is not possible. This would result in data duplication risks and a reinforced silo effect, detrimental to overall information quality. Therefore, a systemic approach to modeling with a global enterprise scope is required, referring to our previous point on the need for complete ontologies. Remember, the read-mode ODS does not introduce such risks to data quality and can be deployed by contexts.

Governance processes around the write-mode ODS are more complex than those necessary for the read-only ODS. It is no longer just about governing data injection flows into the ODS from source systems, but also the upward flows from the ODS to them and to the AIs for cascading updates. It is essential to precisely address the needs for managing histories, versions, security, archives, traceability of flows, and rollback if necessary. With such needs, considering a data fabric solution might be relevant. A particular point to carefully study is the governance of ontologies. You must consider variants and versions of your data model over time and its synchronization with all integration points regarding data sources and update targets. This is a complex subject requiring appropriate expertise and technologies, not just marketing intentions.

Finally, the user interfaces of the write-mode ODS are richer than those of the read-only ODS. It is no longer just about displaying data sets by business concepts but also allowing their update, including the links that express their relationships in more or less complex data hierarchies according to operational processes, and with the required level of security.

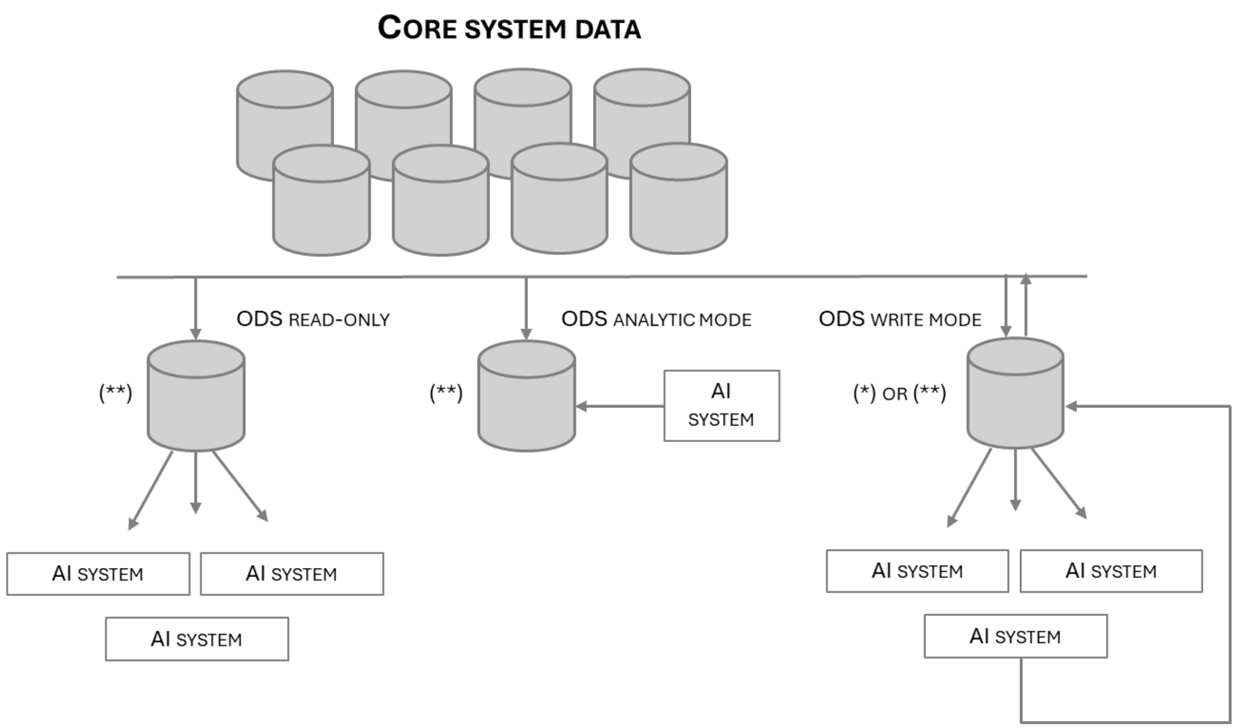
The management of unstructured data follows the same principles already indicated for the read-only ODS. To reiterate, the recommendation is to use a dedicated knowledge management repository (see the TRAIDA Enterprise Knowledge Graph card).

ANALYTIC-MODE WITH KNOWLEDGE GRAPH DB

An analytical ODS aims to analyze operational data, particularly through AI. It is no longer about providing ODS data to AI systems but about applying AI directly on the ODS. Since knowledge graph technologies enable powerful visualization and the use of inference rules, their use can be favored. Additionally, the "schema-free" approach facilitates the implementation of the analytical ODS by reducing the modeling effort required. Knowledge graph technology works well with the read-only ODS that reuses it, but less so with the write-mode ODS when it relies on relational database technology.

Ultimately, the technological choice for implementing the ODS involves considering the combined needs of read-only, write-mode, and analytical ODS. The universal choice of a graph-oriented database is not always possible, especially when the write-mode ODS is highly transactional with intense and complex injection and restitution flows (OLTP). It is then conceivable to implement two ODSs simultaneously and non-competitively: an analytical ODS using a "schema-free" approach for local AI use, and a write-mode ODS on relational database technology for feeding AI systems and synchronizing data updates to application systems.

3. BLUEPRINT



(*): RELATIONAL DATABASE (OLTP)
(**): KNOWLEDGE GRAPHS DATABASE

4. YOUR SITUATION & OBJECTIVES