

# What is Master Data Management (MDM)?

MDM is an approach to reducing data redundancy by maintaining a definitive "record of truth," or **master data**, for critical data in order to supply a single source as a reference. Ideally, MDM organizes data sharing among multiple applications or departments.

## It Starts with Data Integration

It's common for organizations to have duplicate information on different systems. For example, customer information could be stored in both a CRM and accounting system:

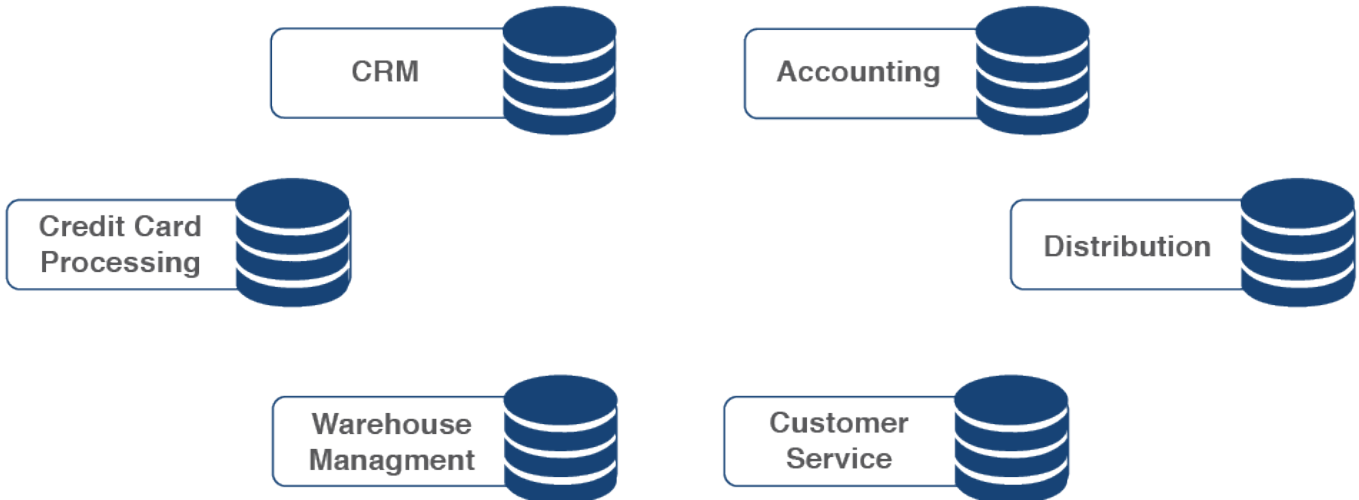


While some data in both systems is identical, other data may be similar but not the same.

With small organizations, it can be straightforward to keep just a few systems up to date. But as an organization brings more systems online, their business data often gets increasingly duplicated, which poses a problem when keeping all of the systems up to date. The process of keeping data up to date between disparate systems is called **data integration**.

For example, imagine a delivery service that stores information about its customers in several systems:

## Delivery Service Example



As customer information changes over time (i.e. customer addresses or phone numbers), keeping all the systems up to date becomes exponentially more difficult.

Consider the diagram above:

- **Where is the truth?**

A customer's email address may be stored on several systems, which may result in conflicting values. If so, which system holds the true/correct value?

- **Which system gets notified of a change and what data elements do other systems need?**

In the example above, if the Credit Card Processing system receives updated customer address, which of the other systems also need to have that data? The Accounting and Distribution systems. If only the street address changed, but the city, state, and zip code information remain the same, then only the street address **data element** of the customer's data record needs to be updated in the

other two systems.

**Note:** The most recent version of a given record is called the **data record**.

If all systems need to be updated, then each system needs to know how to transform the data to be consumed by each of the other systems. This requires programming many transformations. In an extreme case, the example above could take five transformations for each of the six systems, requiring 30 transformation modules or adaptors.

**Note:** An **adaptor** is software located within a system that connects to, and shares data through, the YOUnite Data Hub.

The adaptor functionality focuses on:

- Extract, Transform, and Load (**ETL**) functions at the system level, ensuring any system's "outbound data" is extracted and transformed to meet defined format requirements before it gets loaded onto the message bus sent through the YOUnite Data Hub sent back out to the message bus again so that the receiving system's router then transforms the data into an "inbound data" format that system requires.
- Create, read, update, and delete (**CRUD**) functions at the system level,
- The **YOUnite Data Hub** sits between the various systems, routing data between them based on which systems have access to which data (**data governance**).

- **Are changes handled in real time or in batch?**

The easiest and most common change handling is via batch updates. However, latency between batch updates may cause business process issues.

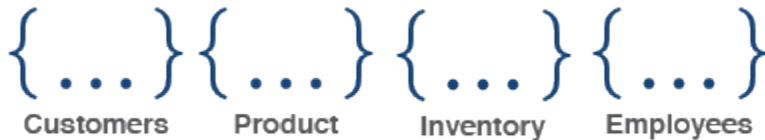
- **How is DI handled?**

Transferring data between systems can be a daunting task. Some applications have built-in adaptors to handle transformations, but these generally handle only a subset of the required data. Where the built-in adaptors fall short or don't exist, an organization may spend resources developing "one-off" adaptors to meet an ongoing transformation need.

- **How does an organization manage access to the data?**

Using the delivery service example above, the Warehouse Management system should get access to only a subset of the customer data for security reasons. And it shouldn't get any level of access to the Credit Card Processing system. This level of access is defined by the **Access Control Lists** or "**ACLs**". Data governance is used to describe managing the ACLs and also defines where the data records are stored.

The challenge doesn't end with just the customer data. Additional data, such as product, inventory, and employee data may need to be kept up to date on several systems:



In MDM, the set of fields or properties (e.g. name, address, phone number) that define a set of data (e.g. Customer) is called a **data domain** (or sometimes just called "**domain**").

## Migrating to MDM

MDM solves the problem of keeping interrelated systems up to date by creating a separate system where data domains are defined for all systems inside the organization. The domains provide neutral data formats or schemas for all systems, facilitating data sharing.

The data for a data domain defined in YOUnite can:

1. Be stored in the MDM Data Store or,
2. Remain in the organization's systems and, through adaptors, share data with other systems via the YOUnite Data Hub. This is called **federated MDM**. With federated MDM, YOUnite's data store does not store the data.

Using the delivery service example above:

- **MDM Data Store:** The customer data records are stored in YOUnite's data store and can be retrieved, in whole or in part, by the delivery service company's system applications that have appropriate access.
- **Federated MDM:** If the customer domain is federated, then the customer data record is NOT stored in YOUnite's data store but is created in real-time by referencing customer domain properties as they reside inside the various systems (i.e. the customer name, address, and phone number properties as stored in the Customer Service, Accounting, and CRM systems).

In the federated model, data records can be stored in the YOUnite Data Store or in one or more systems connected to YOUnite. Many systems may hold similar data but generally the organization as a whole decides which system(s) hold the data records. Note

that with YOUNite's federated model, different groups inside of an organization can designate which system holds the master data. YOUNite's governance model can manage who can access the data.

In the federated example, governance can be set so that:

For the Warehouse Management System...	For the Accounting System...
data access is restricted to data stored in the Distribution and CRM systems	data access is allowed to all the other systems
data record lookups return only information that is appropriate for the division's needs	data record lookups return information from all systems, which may include Credit Card processing, for example

Implementing MDM involves a process of determining where the data records are stored (whether it's the MDM Data Store or Federated MDM) and managing who (a person or a system) has read, write, update, and delete privileges to those systems.

## Data Records vs Master Data

**Data records** are the latest, most recent version of a record stored in many systems connected to YOUNite. **Master data** is data in a particular domain or a particular element that has been declared the "Record of Truth." It's not always necessary or appropriate for systems to access an organization's master data. Many data access requests are for data records that may or may not contain master data. However, YOUNite has the ability to propagate changes from a system that contains data records to others in the YOUNite ecosystem on a permission-appropriate basis (i.e. governance).

## Reviewing New Terms

Several terms have been introduced and it may be helpful to review them before moving on:

- **Access Control Lists (ACLs)** Data controls that control inbound/outbound messages for any given system, application, or role.
- **Adaptor** Software located within a system that shares data through the YOUNite Data Hub and acts as the connection point between a system and the YOUNiteData Hub. It focuses on ETL (Extract, Transform, and Load) functions, ensuring the outbound data from a system meets the YOUNite Data Hub format requirements, transforming inbound data from the hub into what other systems require.
- **Data Domain (Domain)** A domain refers to a data entity such as student, course, or customer, etc. and is defined by the parties responsible for data governance. It is a set of fields or properties that define a set of data (i.e. a domain may be a "Student" that includes data (properties) such as student name, address, etc.).
- **Data Governance** Managing data access (i.e. who accesses certain data sets based on role, application, etc. defining where the *Master Data* is stored).
- **Data Integration (DI)** The process of keeping data up to date between disparate systems.
- **Data Record** The latest (most recent) version of a given record. More technically, JSON objects in motion that follow the data domain model schema of the MDM ecosystem.
- **Federated MDM** An MDM model in which the data handled through MDM is not stored centrally in the Data Hub but instead a reference to the data is stored that points to the the source system where the data hub can retrieve the data.
- **Master Data (MD)** The master or golden version of a record (for a customer, for example). The "record of truth" as declared by a Data Governance Steward (DGS) or Zone Data Steward (ZDS).
- **Master Data Management (MDM)** An approach to reducing data redundancy across systems by maintaining a master file for critical data. MDM is the process of describing and cataloging data inside of an organization and understanding which stakeholders value which sources of data.
- **YOUNite Data Store/MDM Data Store** A centralized store natively connected to the YOUNite Data Hub that holds data records. The domains configured for this data store are locked in as the source of master data for that domain in the Tenant.
- **YOUNite Data Hub** The scalable web application that handles the API and message broker requests through which adaptors and the YOUNite API consumers communicate.

## Further Reading

An good source for more MDM background is Mark Allen and Cervo Dalton's *Multi-domain Master Data Management: Advanced MDM and Data Governance in Practice*. Waltham, MA: Morgan Kaufmann, 2015. (ISBN 978-0-12-800835-5).

## Next

[An Introduction to YOUNite](#)

