



MDM Concept in nutshell

OVERVIEW – Master Data Example

Multiple source systems might provide data in **inconsistent** format, often referring to a single instance but its format and quality.

GOAL: find a way to select the best values across all data sources to create a single "truth" – master record



MATCHING – group records together

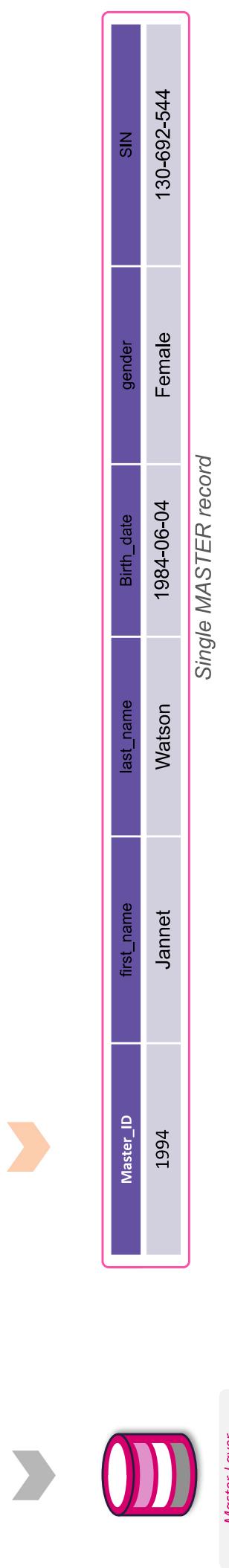
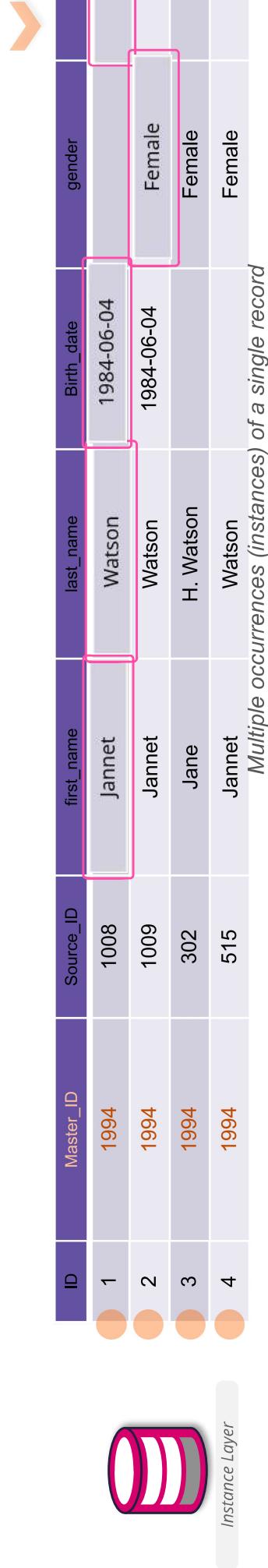
Configure **criteria** that would guarantee that **relevant records** are recognized and **linked together**.

ID	Source_ID	cio_first_name	cio_last_name	cio_Birth_date	cio_gender	cio_SIN
1	1008	Jannet	Watson	1984-06-04		130-692-544
2	1009	Jannet	Watson	1984-06-04	Female	130-692-544
3	1013	Anton	Lese	1950-09-08	Male	387-958-572
4	515	Jannet	Watson		Female	
5	1001	John	Smith	1978-12-16	Male	095-242-434
6	1018	Sandy	Hettinger	1965-09-21	Female	856-827-270
7	1003	Smith	John	1978-12-16	Male	095-242-434
8	1014	Anton	Lesse	1950-09-08	Male	387-958-572
9	1020	Thomas	Donathan	1966-02-04		961-085-248
10	1021	Tom	Donathan	1966-02-04	Male	961-085-248
11	1019	Sandy	Hettinger	1965-09-21	Female	
12	1038	Sandy	Hettinger	1965-09-21	Female	856-827-270
13	302	Jane	H. Watson		Female	130-692-544
14	316	Anton	Lese	1950-09-08	Male	387-958-572
15	1055	Tom	Donathan	1966-02-04		961-085-248
16	1002	John	W. Smith	1978-12-16	Male	095-242-434
17	1039	Sandy	Hettinger	1965-09-21	Female	856-827-270
18	1058	Anton	Lesse	1950-09-08	Male	387-958-572
19	317	Jane	Jordan	1982-05-01	Female	420-347-213

MATCHING – group records together

Conditions for a successful match:

- Identical value in a unique attribute (Strong/Primary Key)
- All identical values across a combination of multiple attributes



Topic Highlights

- Master data often comes from multiple sources with varying formats. The goal is to create a unified “by selecting the best values from these sources.”
- To match records effectively:
 - Use identical unique attributes (primary keys).
 - Match values across multiple attributes to ensure accuracy.



Memory
Refresher #1
MDM Concept in
nutshell

What is the primary goal of master data management (MDM)?

- 1.** To keep historical version of data
- 2.** To integrate data from various sources into a unified "truth"
- 3.** To enhance the security of data

What should you do to ensure accurate record matching across multiple attributes?

- 1.** Use primary keys only
- 2.** Match values across multiple attributes
- 3.** Use only manually verified records

Introduction to MDM



Master Data Management

Master data management (MDM) is a technology-enabled discipline in which business and IT work together to ensure the uniformity, accuracy, stewardship, semantic consistency and accountability of enterprise's official shared master data assets

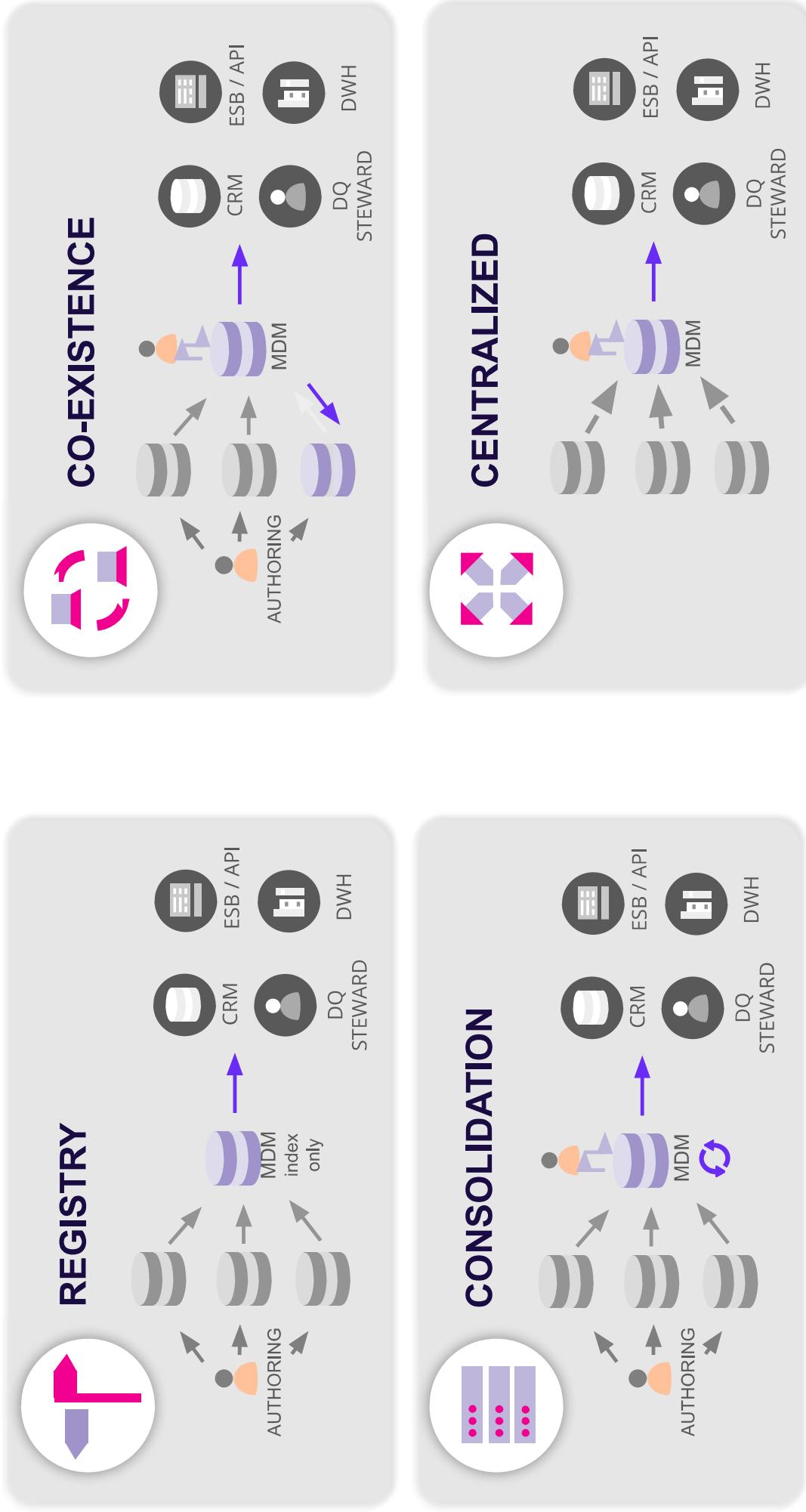
Master Data Management is the process of defining and maintaining how the critical data will be created, maintained and used throughout the enterprise as a single point of reference.

The data that is mastered may include:

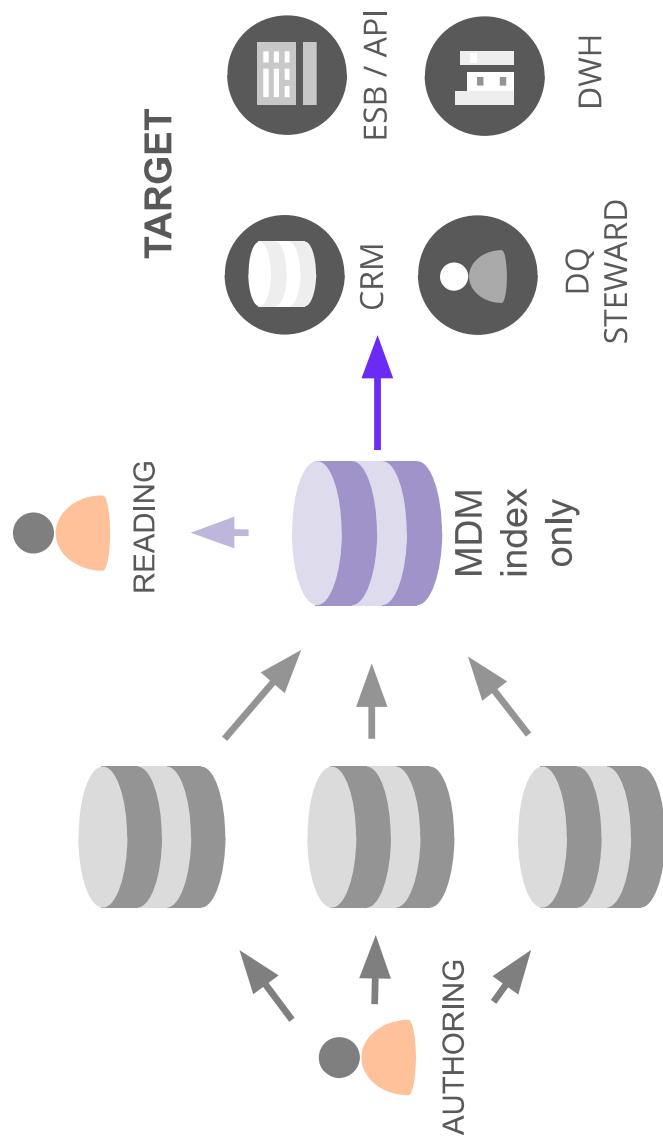
- Reference data- the set of permissible values
- Analytical data - that supports decision making

Master data management (MDM) is a technology-enabled discipline in which business and IT work together to ensure the uniformity, accuracy, stewardship, semantic consistency and accountability of enterprise's official shared master data assets

Master Data Management Styles



Master Data Management Styles



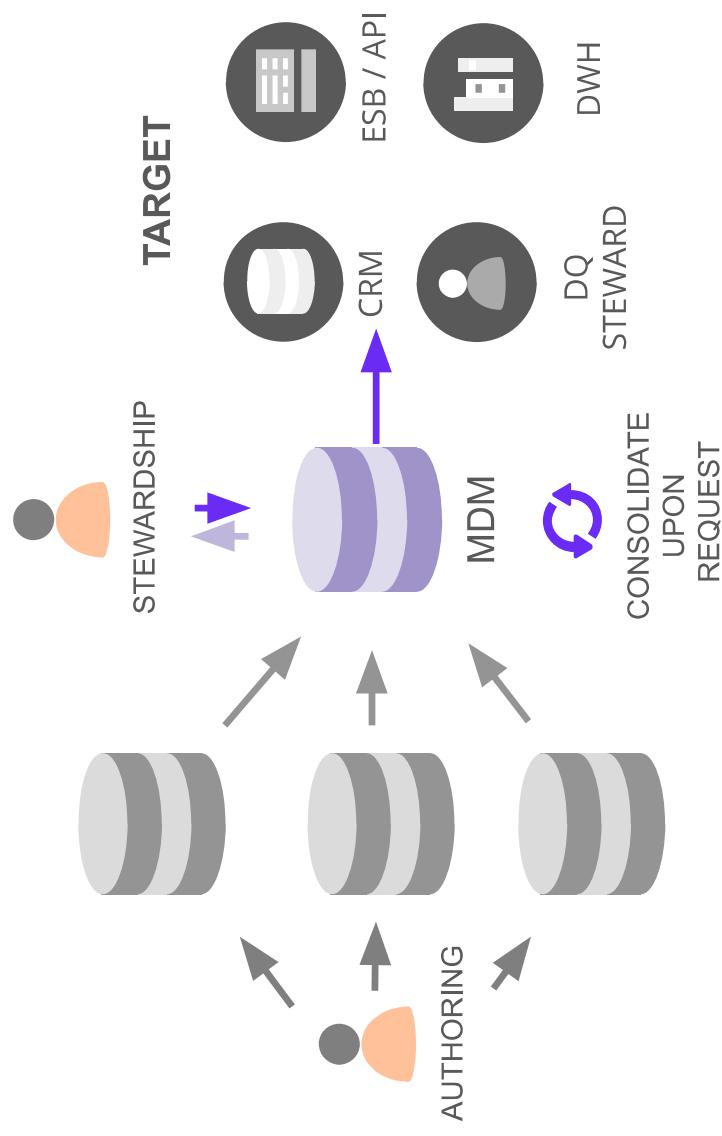
REGISTRY

- Non-intrusive
- No data consolidation
- Cross Ref Index
- Use of federated queries
- Read access only

LEGEND:

- | Icon | Master data | Original data | Data flow |
|-----------------|-------------|---------------|------------------|
| Purple cylinder | ↑ | → | Master data flow |
| Grey cylinder | ↑ | ↑ | On demand only |
| Grey arrow | → | → | Data flow |

Master Data Management Styles



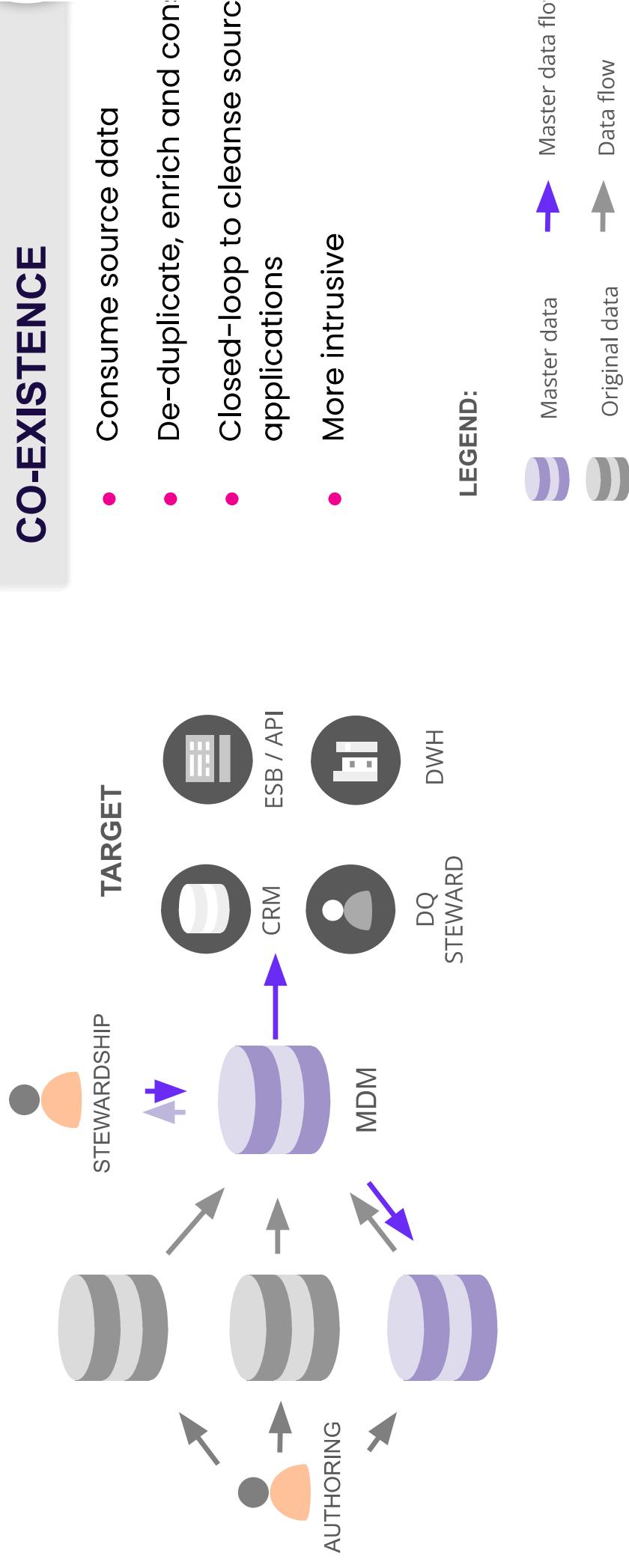
CONSOLIDATION

- Consume source data
- De-duplicate, enrich and consolidate
- Publish data to downstream app
- Non-intrusive

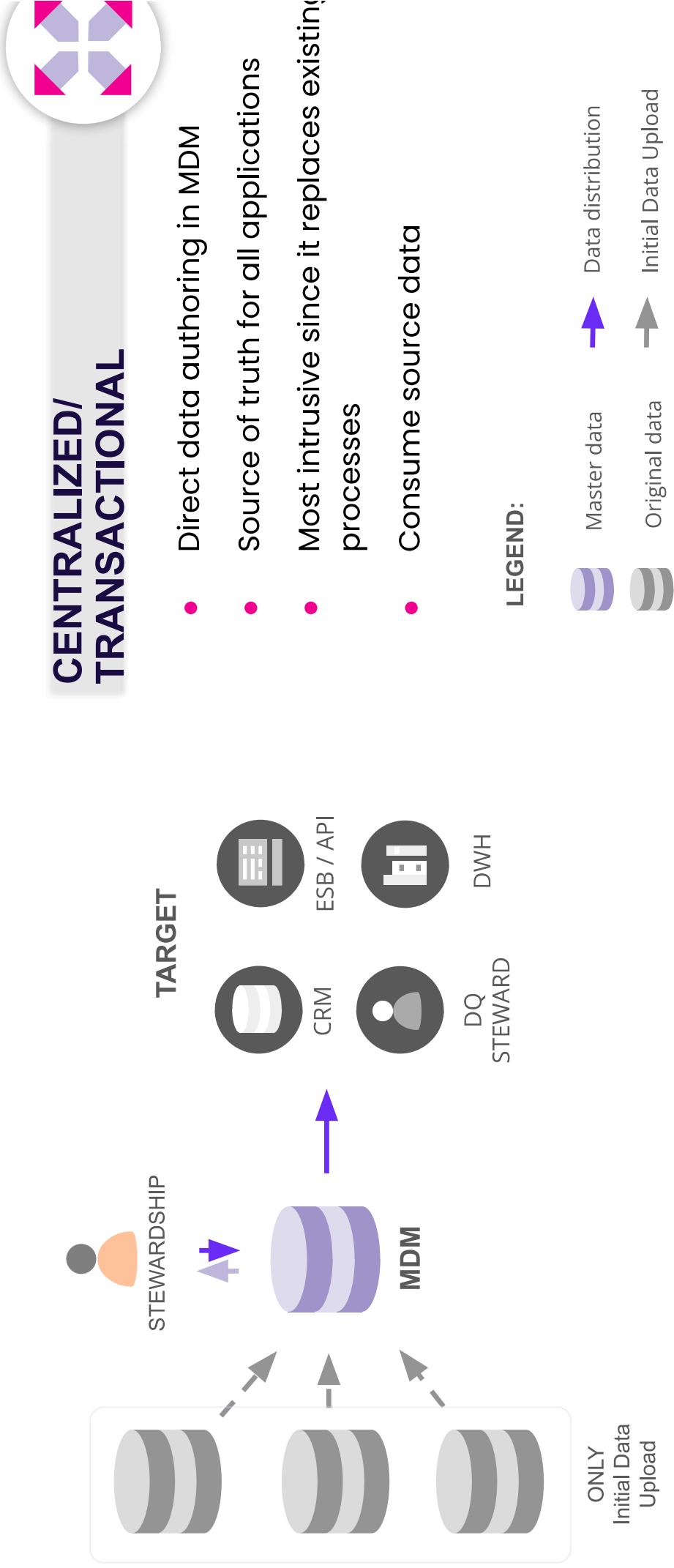
LEGEND:

- Master data flow (purple arrow)
- Original data (grey arrow)
- Data flow (grey arrow)

Master Data Management Styles

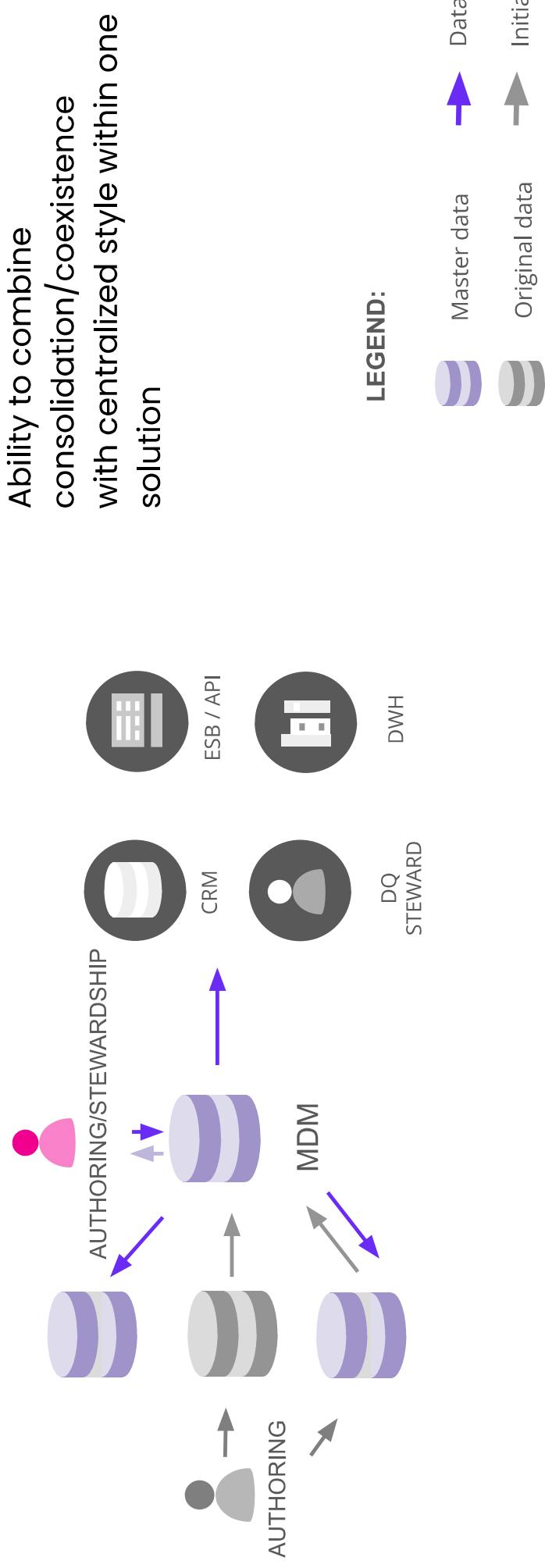


Master Data Management Styles

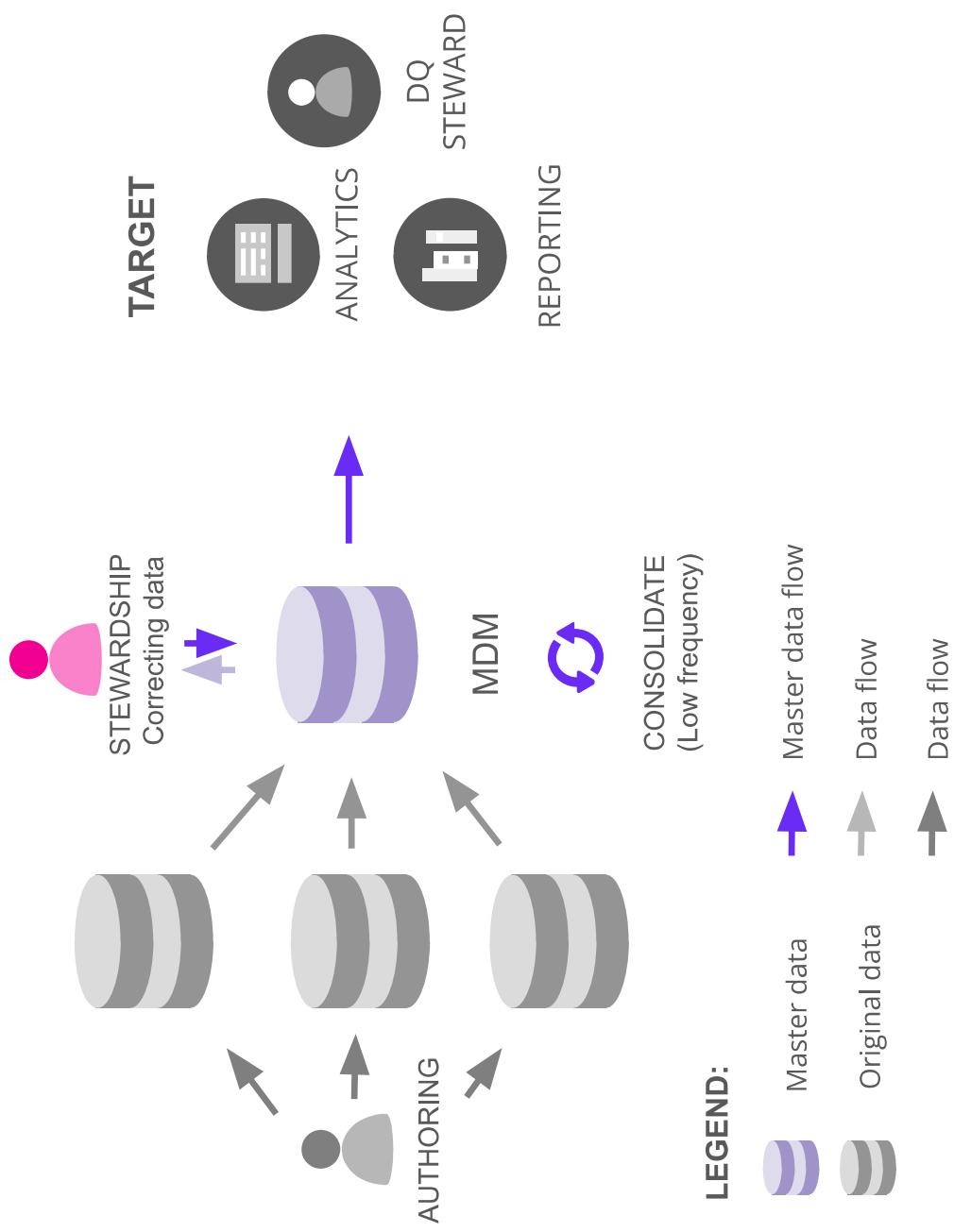


Master Data Management Styles

MIXED



Master Data Management Styles



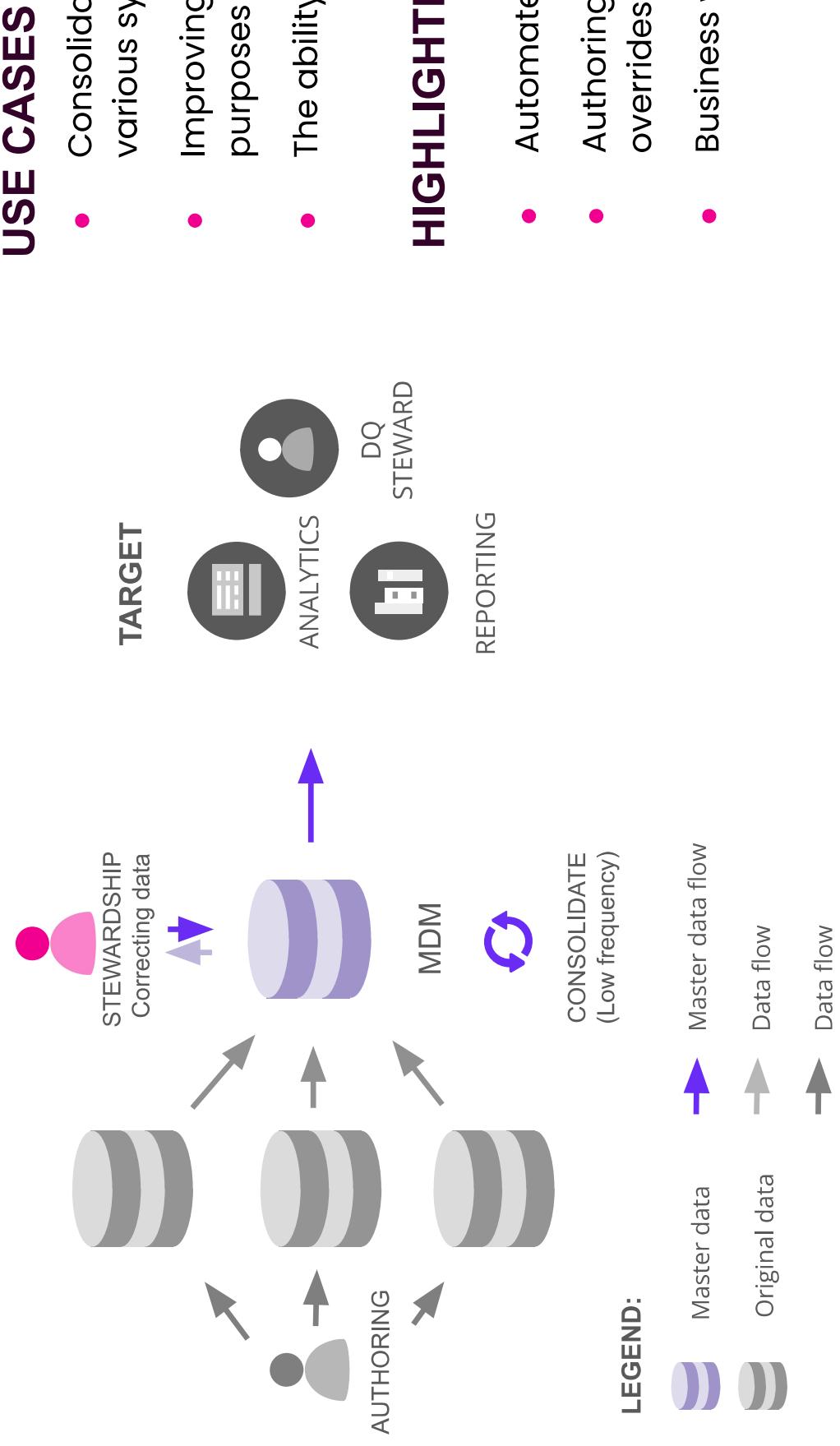
USE CASES

- Consolidation of data across various systems
- Improving the data for reporting purposes
- The ability to add missing

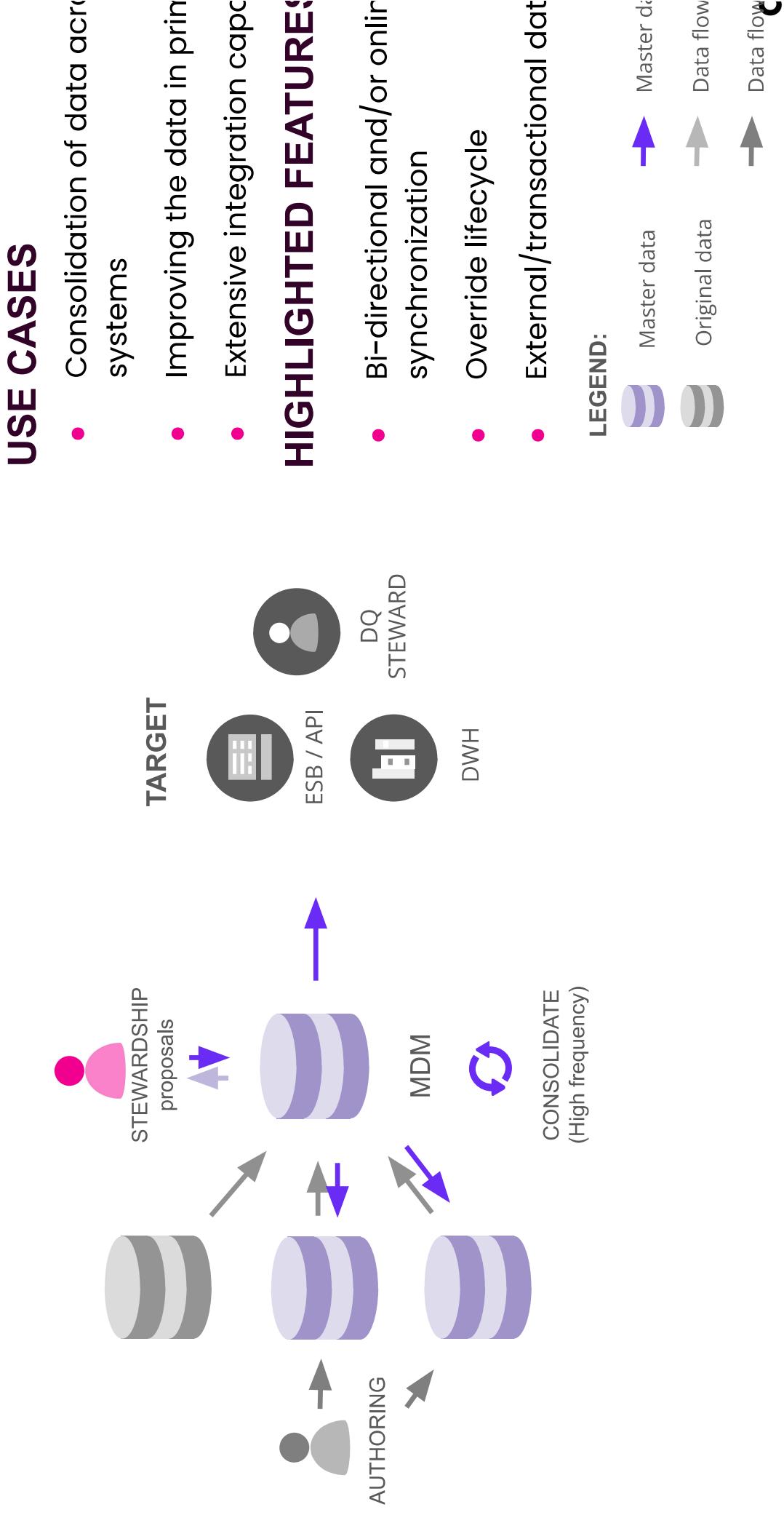
HIGHLIGHTED FEATURE

- Automated consolidation
- Authoring new records in overrides
- Business workflow

Example 1: Pharma Customer



Example 1: Pharma Customer



Example 2: Bank Customer

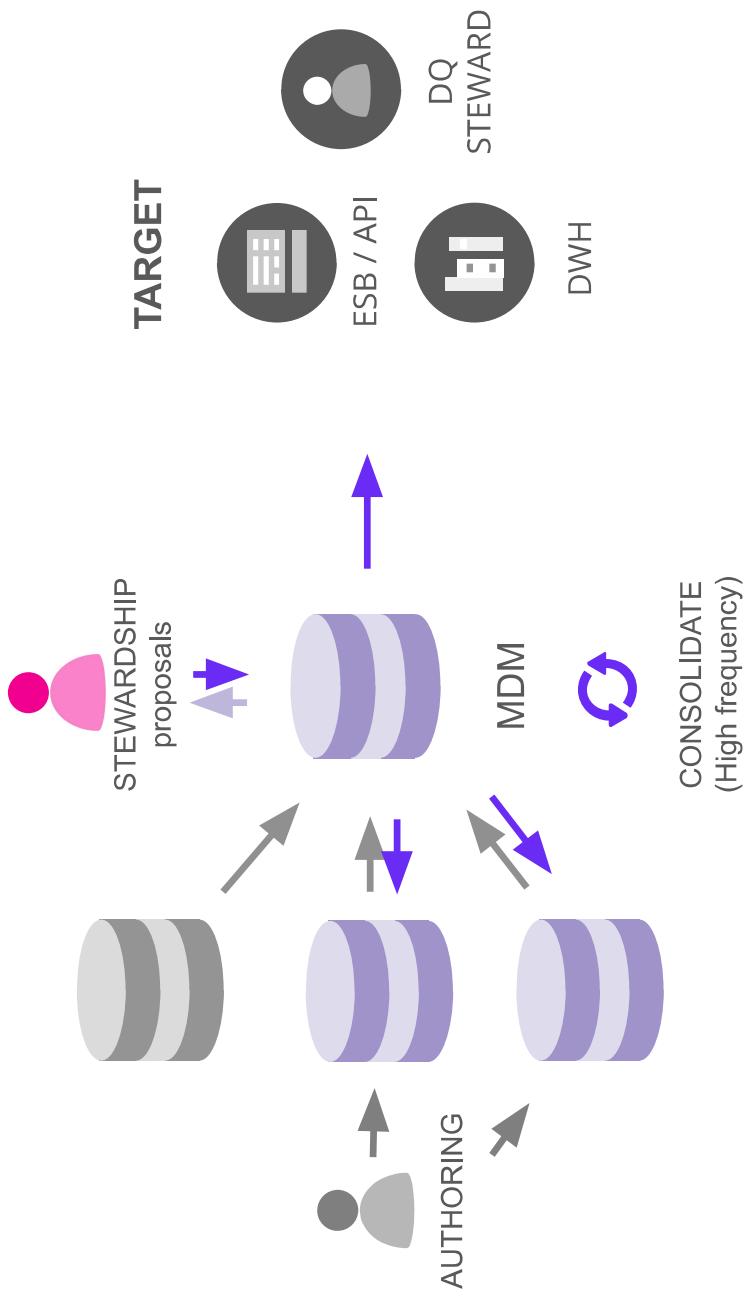
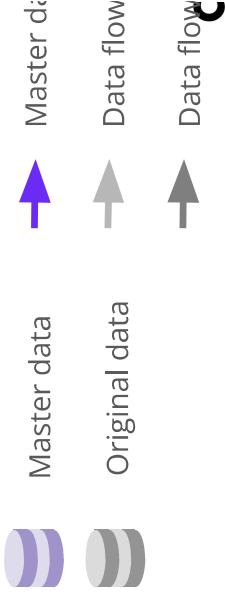
USE CASES

- Consolidation of data across systems
- Improving the data in production
- Extensive integration capabilities

HIGHLIGHTED FEATURES:

- Bi-directional and/or online synchronization
- Override lifecycle
- External/transactional data

LEGEND:



Example 3: Bank Customer

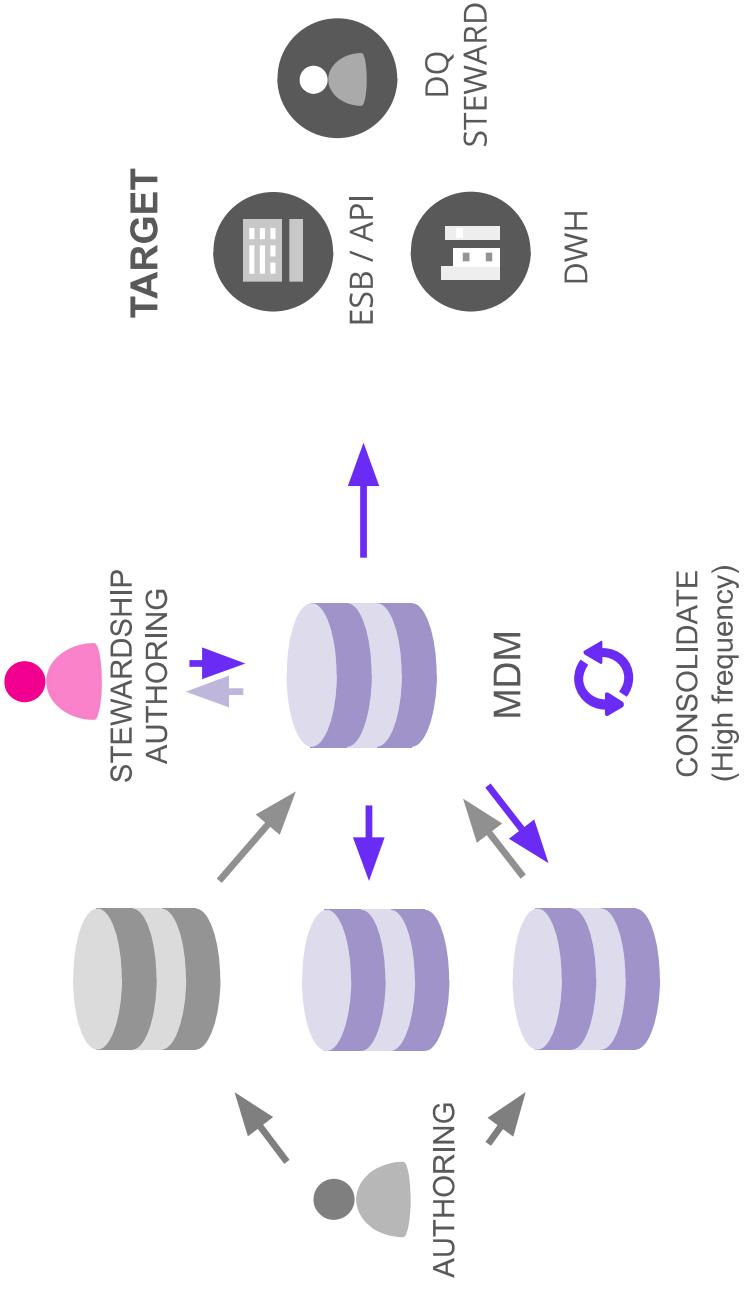
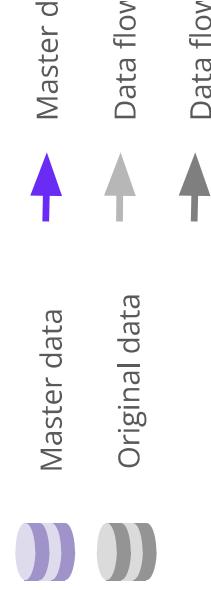
USE CASES

- Consolidation of data across various systems
- Improving the data in primary systems
- Authoring data centrally

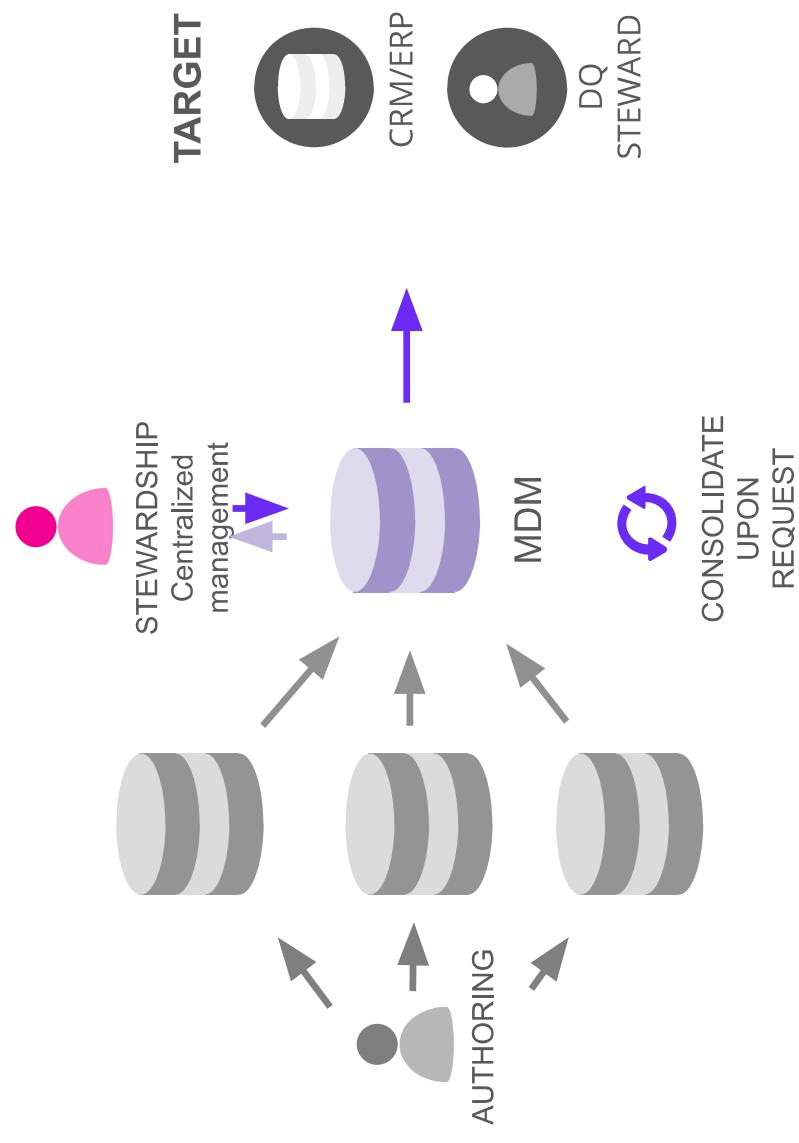
HIGHLIGHTED FEATURES

- Authoring on both MDM and DWH
- Bi-directional and/or online synchronization
- Override lifecycle

LEGEND:



Example 4: Retail Customer



USE CASES

- Centralized master data management
- Data migration
- Stewardship

HIGHLIGHTED FEATURE

- Initial/migration load with consolidation
- Multi domain mastering
- Business workflow with master data

LEGEND:

Master data	↑	Master data flow
Original data	↑	One time/One off
	→	Data flow

Example 5: Retail Customer

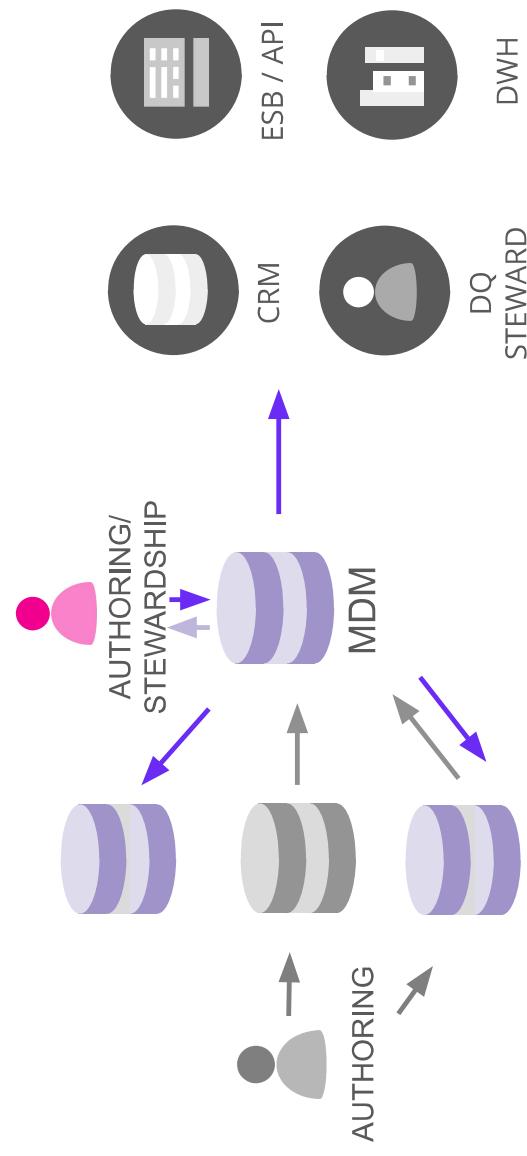
USE CASES

- Mixed master data management
- Data migration
- Stewardship

HIGHLIGHTED FEATURES

- Authoring on both MDM and DWH
- Bi-directional and/or online synchronization
- Override lifecycle

LEGEND:



Topic Highlights

Registry:

- Definition: A registry MDM style maintains a centralized index or registry of master data without physically consolidating it. It acts as a reference or directory that maps and links records from various systems.
- Key Feature: No data consolidation; instead, it uses federated queries to access and reference data from source systems.

Consolidation:

- Definition: In a consolidation MDM style, data from multiple sources is aggregated and consolidated into a repository. This involves cleansing, deduplicating, and enriching data to create a unified view.
- Key Feature: Data is physically combined in a central location, improving data quality and consistency for downstream applications.

Topic Highlights

Co-Existence:

- Definition: Co-existence involves maintaining master data in multiple systems while ensuring that these can share and synchronize data. It balances between central control and local system autonomy.
- Key Feature: Master data exists in both centralized and local systems, with mechanisms for synchronization to ensure consistency.

Centralized:

- Definition: A centralized MDM style involves managing all master data within a single central repository. creation, maintenance, and access are controlled from this central point.
- Key Feature: All master data is stored and managed centrally, providing a single source of truth and often to a more controlled and consistent data environment.

Use Cases: Examples from pharma, banking, and retail show different MDM needs such as consolidation, data improvement, and synchronization.



Memory Refresher #2 Introduction to MDM

Which type of data might be managed as part of MDM?

- 1.** Only financial data
- 2.** Reference data and analytical data
- 3.** Mostly personal information

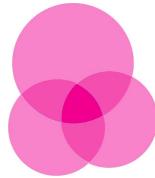
How does the Consolidation MDM style handle data?

- 1.** By maintaining data in multiple systems
- 2.** By aggregating and consolidating data into a central repository
- 3.** By querying data from multiple databases without merging the data into a single location.

MDM Overview



MASTER DATA MANAGEMENT – Overview



ALL PURPOSE MDM

Multi-Domain Mastering
All common MDM Styles including
Mixed style
High performance engine

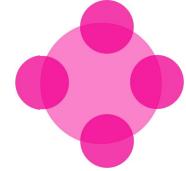
AUGMENTED

Built-in DQ, Enrichment and DQ
Firewall
Deal with DQ issues and
matching proposals
AI-assisted Matching

POWERFUL BUSIN||

Browse master and exte
data, discover data
relationships
Create and edit
Export Data

MASTER DATA MANAGEMENT – Overview



MASTER DATA HUB

Key architectural component for Master Data centralization and integration that reduces silo & point-to-point integration with Data Quality as an integral part



ALL PURPOSE MDM

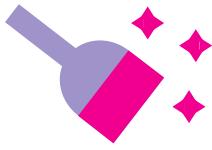
All common MDM styles e.g. consolidation, coexistence, systems of record are supported including mixed mode



MULTI-DOMAIN MASTERING

Consolidate, cleanse, store and distribute any master data entity. Among other, distribution is performed via out-of-the-box generated online interfaces.

DATA QUALITY – Fully integrated



DATA CLEANSING AND ENRICHMENT

Identify locations, cleanse and enrich customer data
(individuals, legal entities)



MATCH & MERGE

Correctly match related records, create representative “golden records”



DQ FIREWALL

Prevent poor quality data from entering your system using MDM as a validation authority

MDM WebApp Browse & Edit



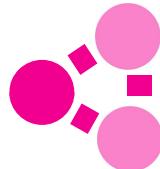
BROWSE

Browse your master data, discover related entities and instance records



CREATE & EDIT

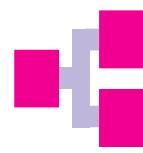
Turn the MDM HUB into System of Record mode and maintain your master data



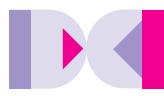
RELATIONSHIP MANAGEMENT

Discover your data and mutual relationships in a hierarchical view

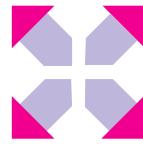
MASTER DATA MANAGEMENT – Quick Facts



Model- and metadata-driven



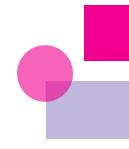
Quick deployment



Integration, SOA, messaging



Built-in data quality



From DQ to MDM



Cost-effective

Topic Highlights

Key MDM Components:

- Master Data Hub: Centralizes and integrates master data.
- Multi-Domain Mastering: Supports various data domains.
- Data Quality Integration: Includes cleansing, enrichment, and validation.

MDM Features:

- AI-Assisted Matching: Improves record accuracy.
- Data Quality Firewall: Prevents poor-quality data entry.

Memory Refresher #3 MDM Overview



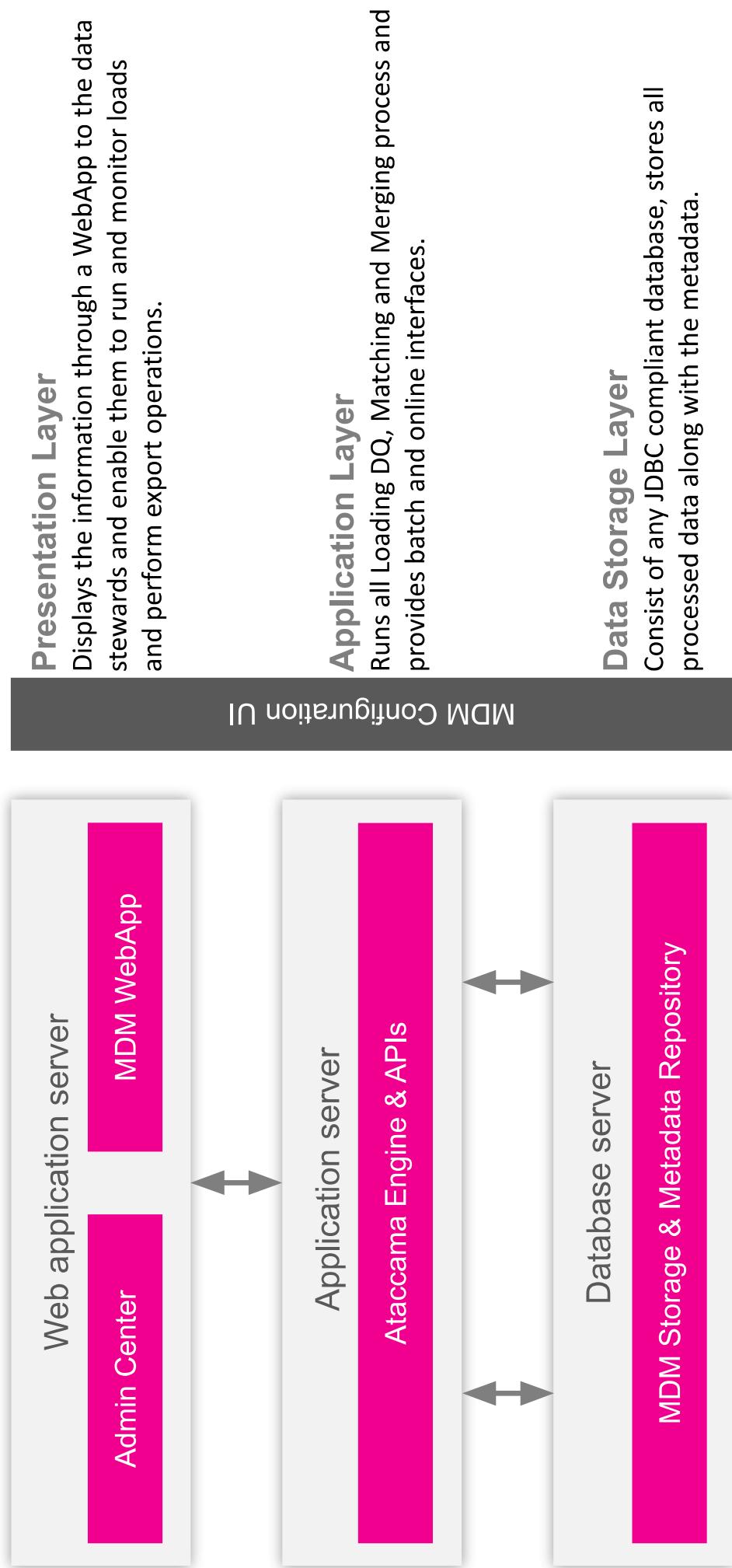
Which MDM component is responsible for centralizing and integrating master data?

1. Data Integration Layer
2. Master Data Hub
3. MDM central server

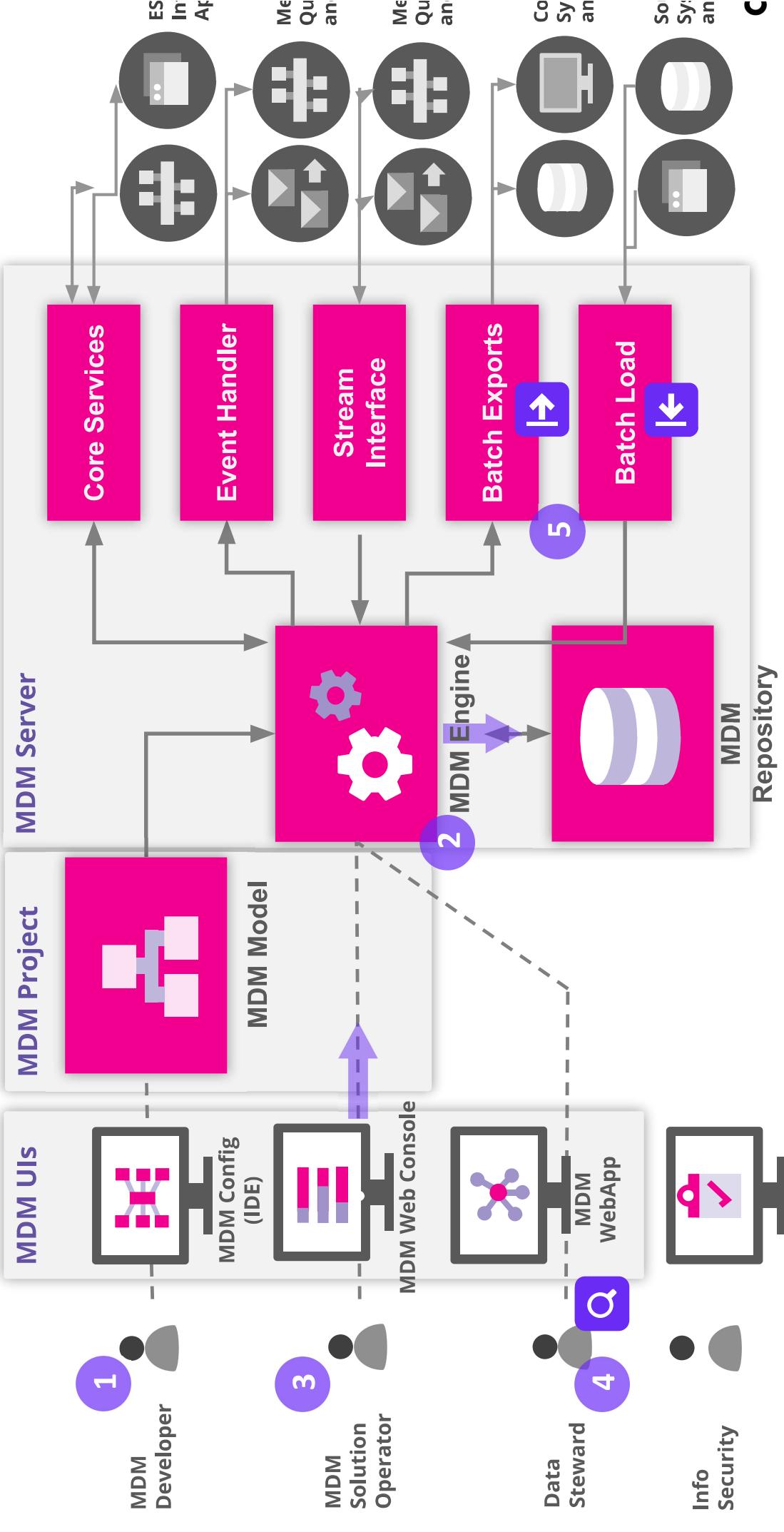
MDM Principles



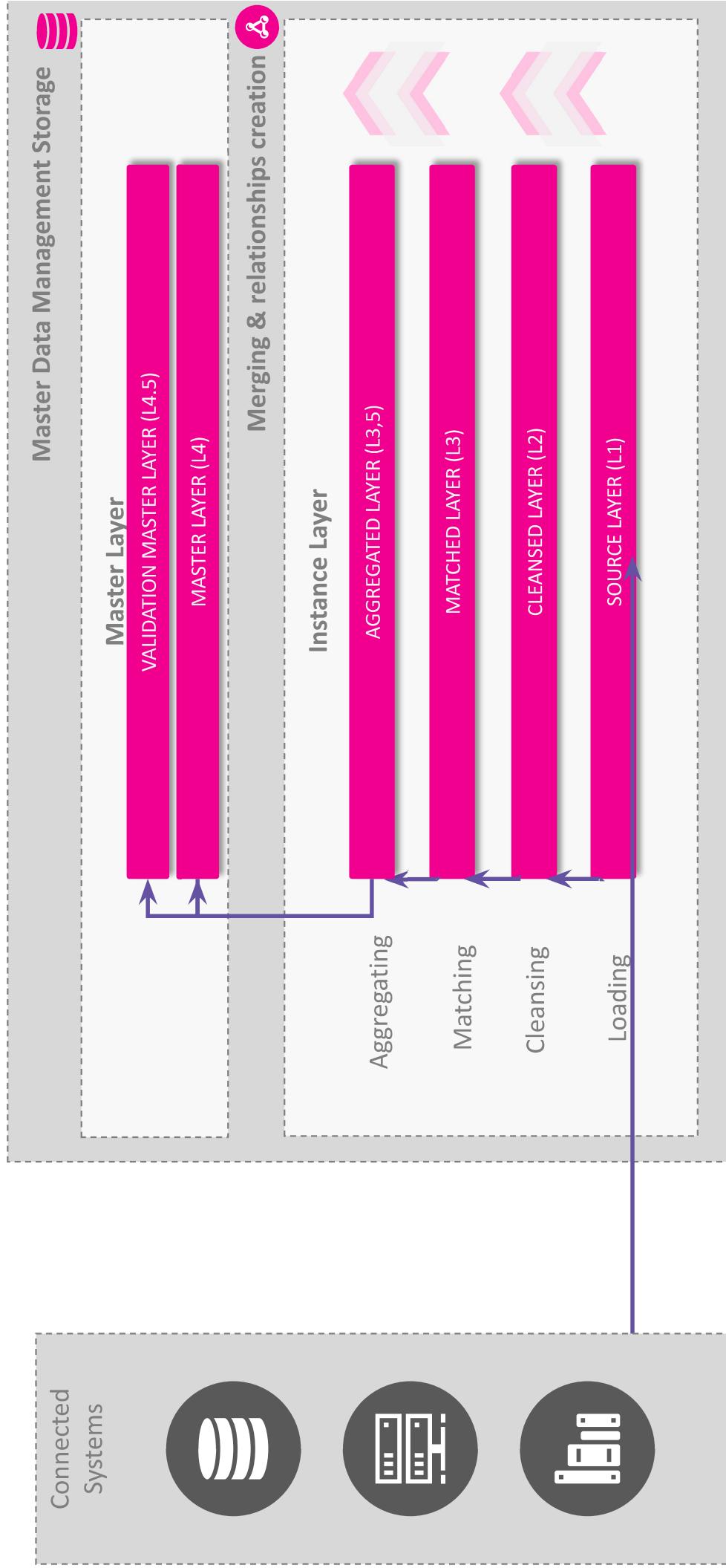
MDM – Technical Architecture



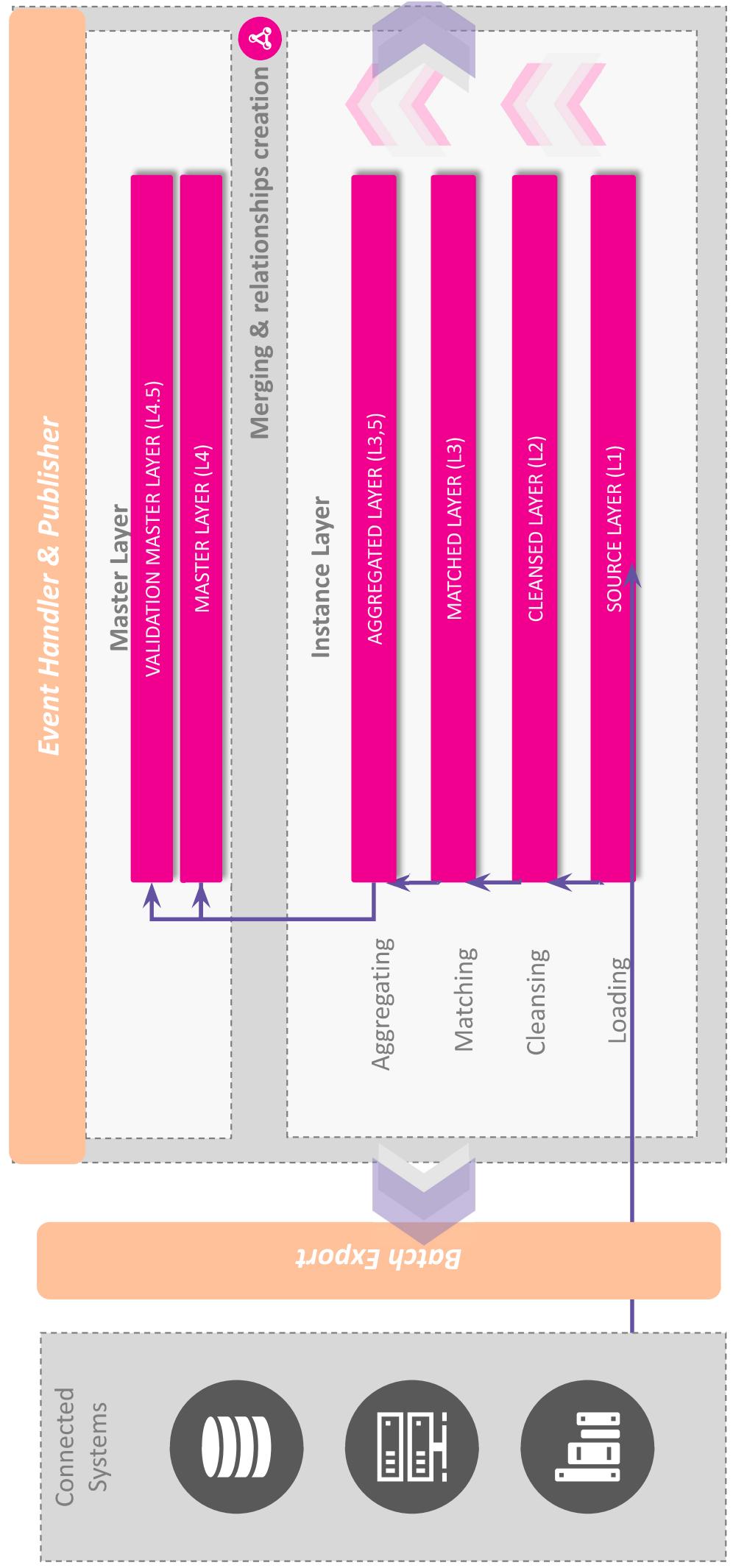
ONE MDM – Architecture



MDM – Data Storage Layers



MDM – Output Interfaces



MDM Processes VS. Data Layers

Source data layer:

- Data as it appears in source systems
- Canonical format (“common” format for all systems)
- No other transformations

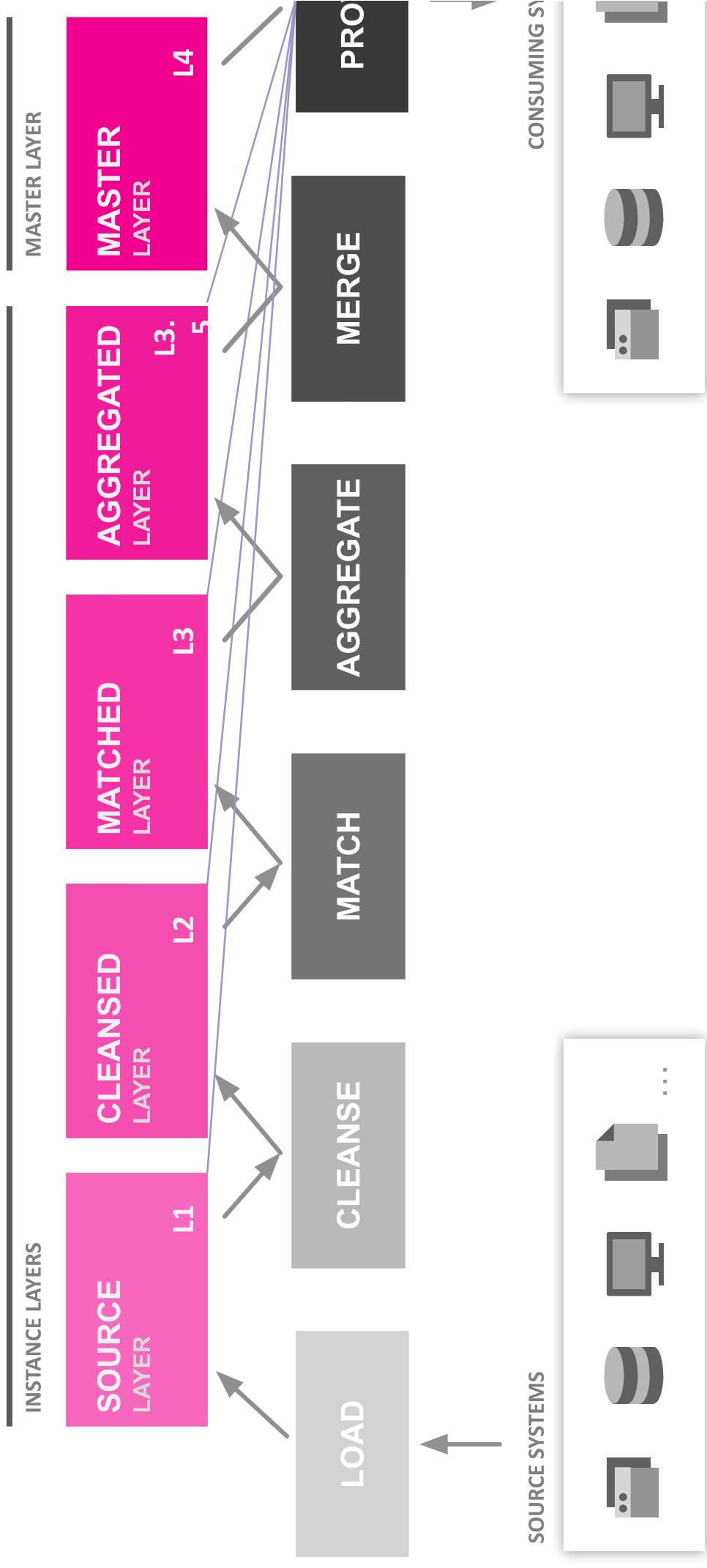
Cleansed data layer:

- Cleansed and standardized data
- Data quality metadata

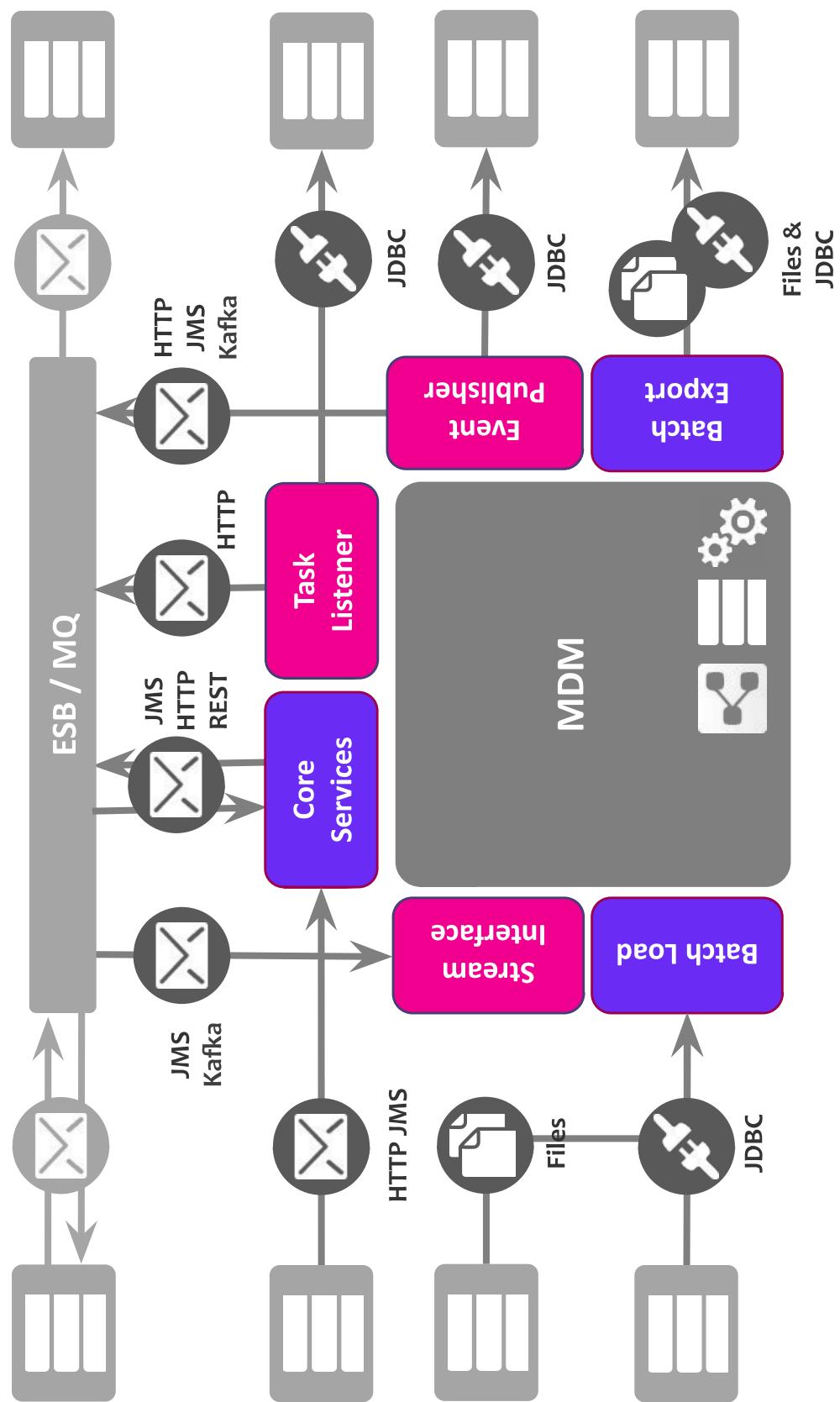
Matching data layer:

- Identifies which instance records correspond to a single object/person/etc.
- Based on cleansed data and associated DQ metadata

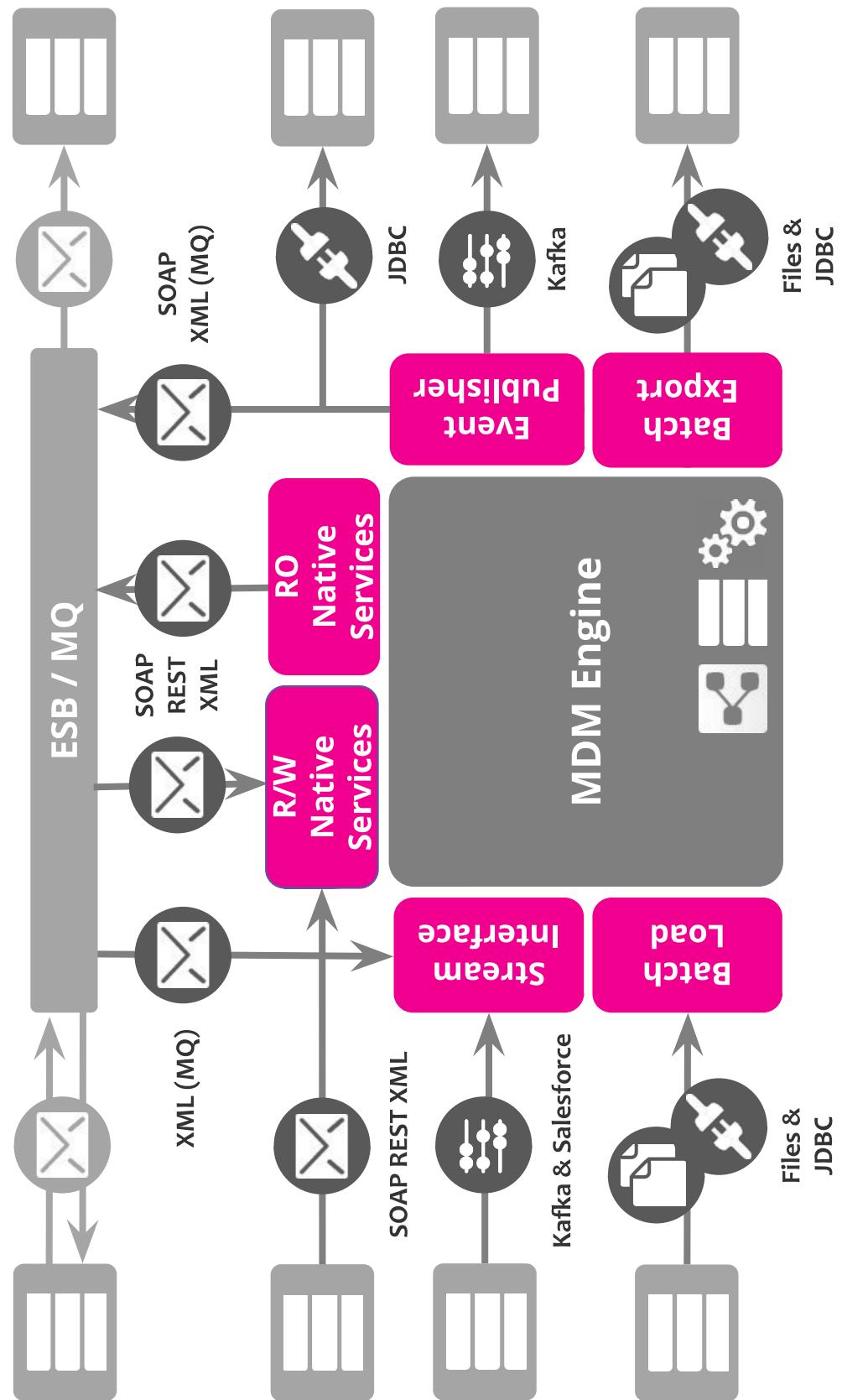
ONE MDM – Layers & Processes



MDM Integration Schema

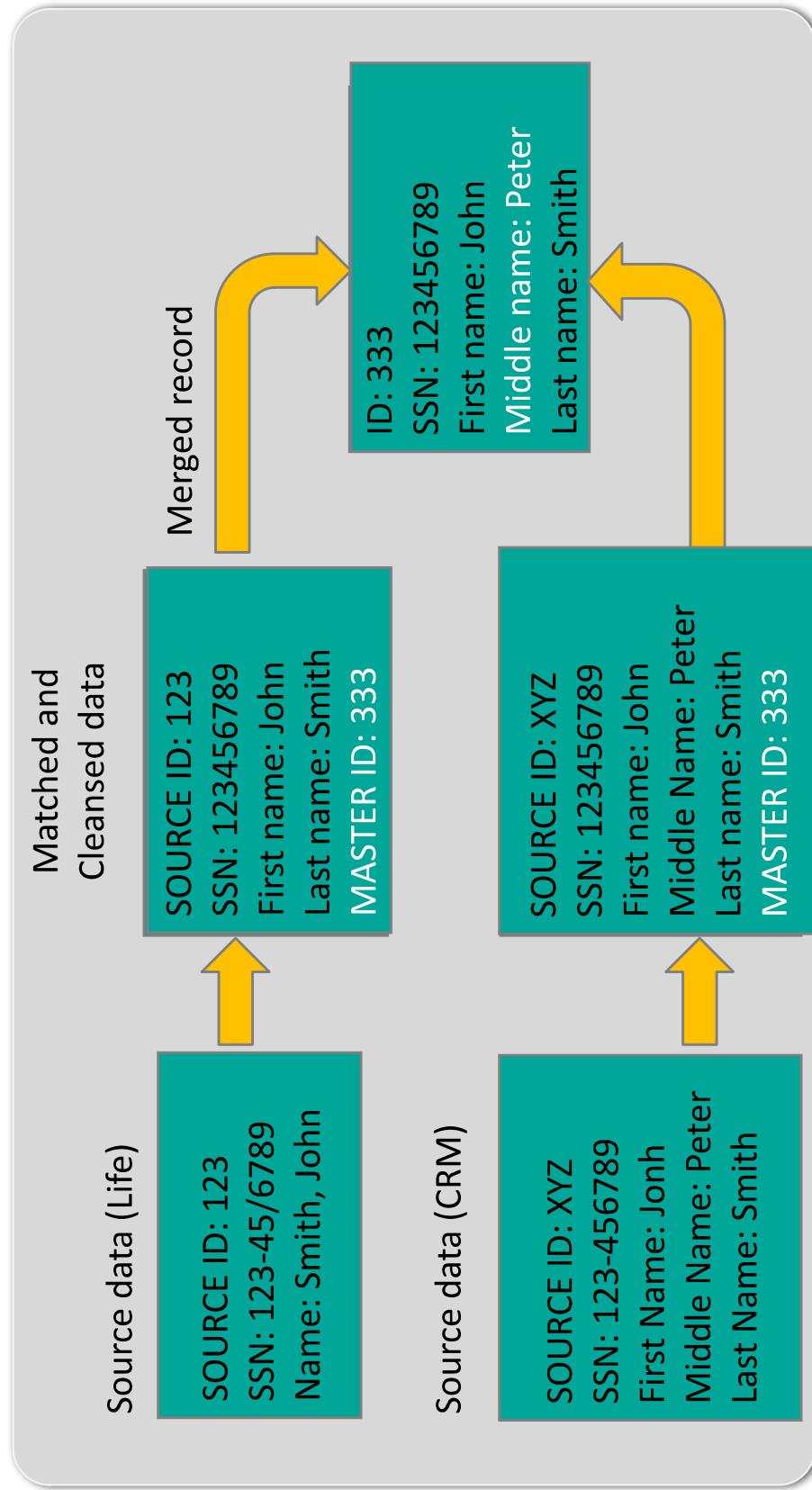


MDM Integration Patterns



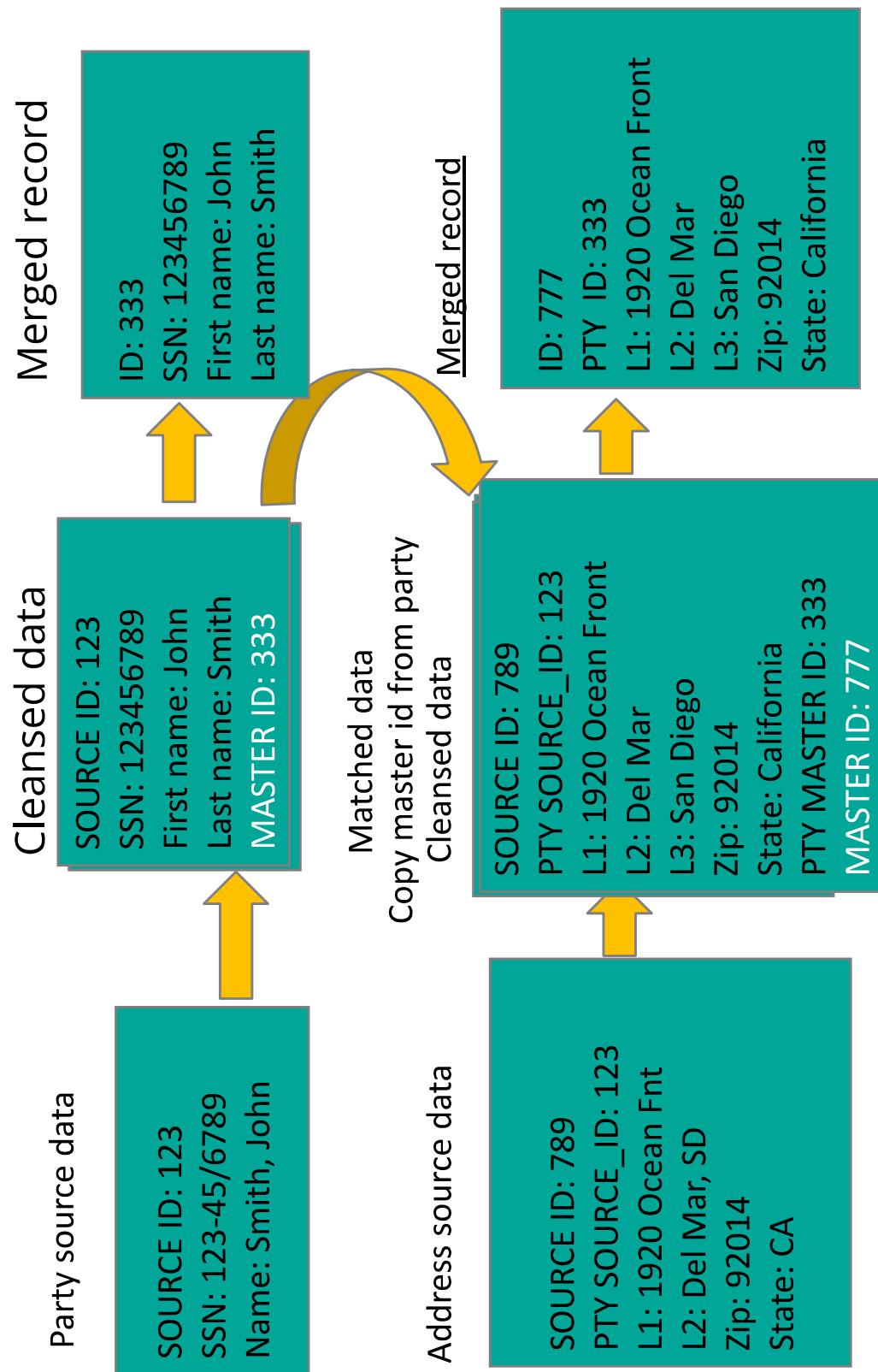
Cleanse, Match and Merge Principle

single entity and multiple systems



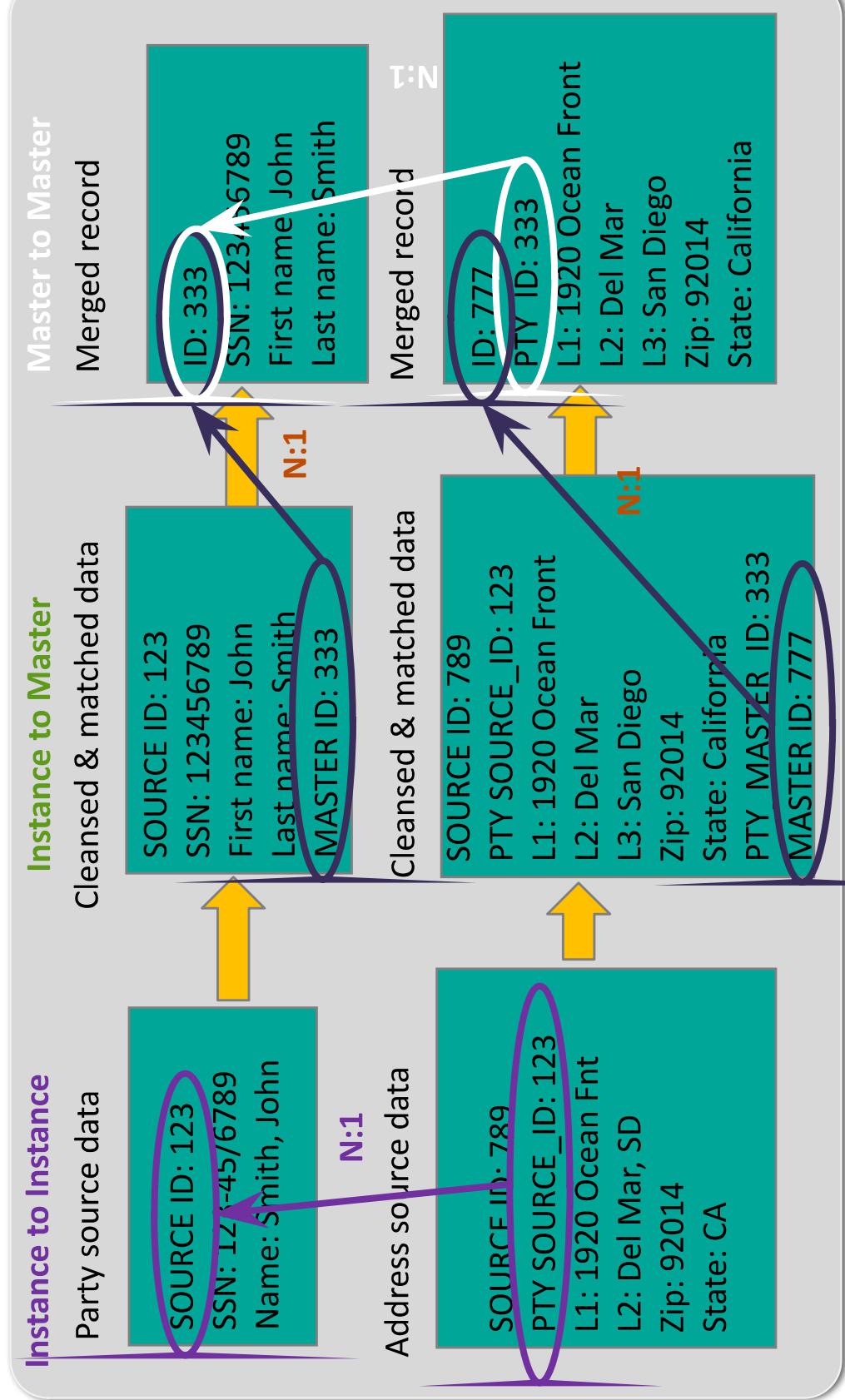
Cleanse, Match and Merge Principle

single system and multiple entities



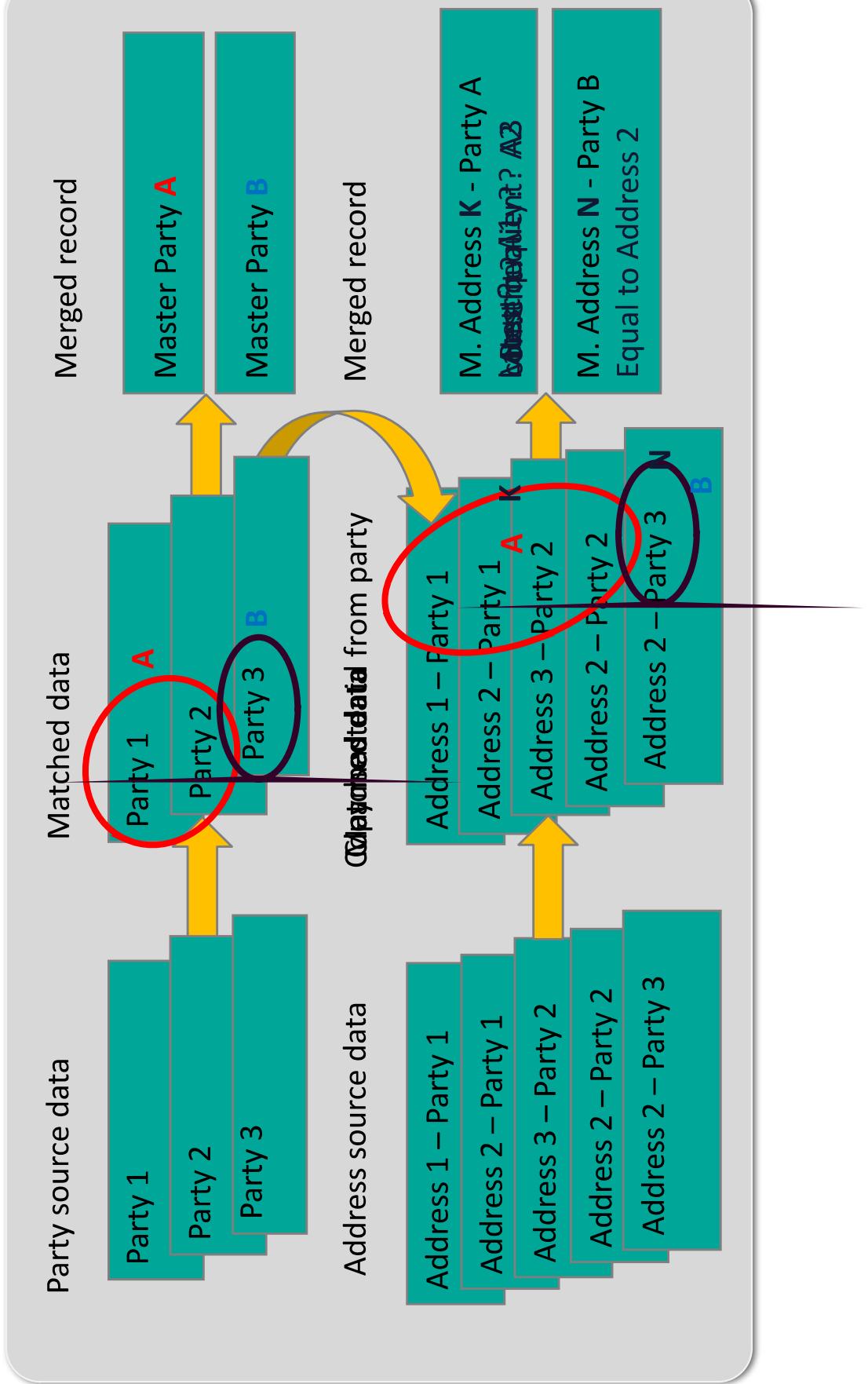
Cleanse, Match and Merge Principle

Highlighted relationship keys



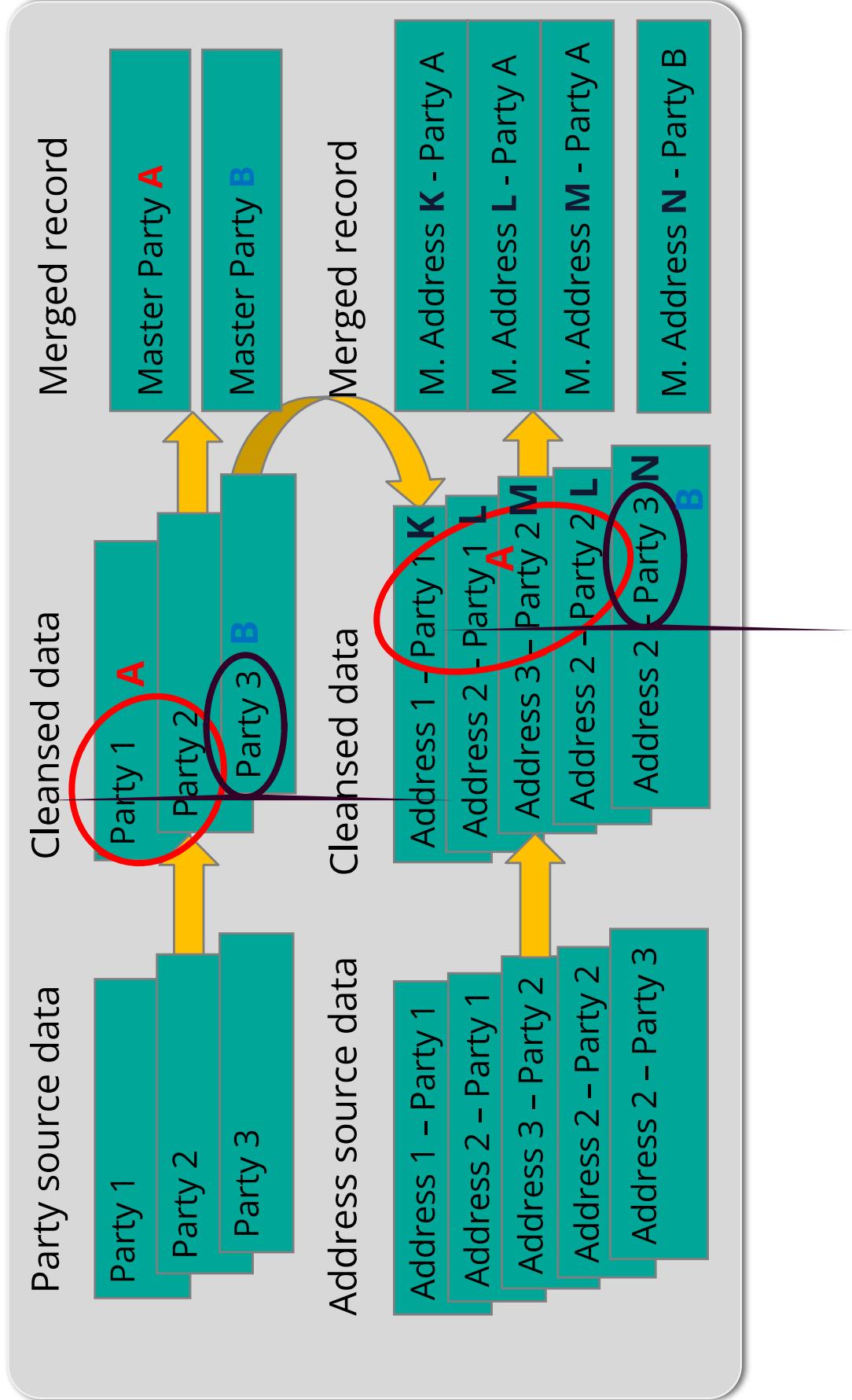
Cleanse, Match and Merge Principle

Rare case: mastered party and address (1:1)



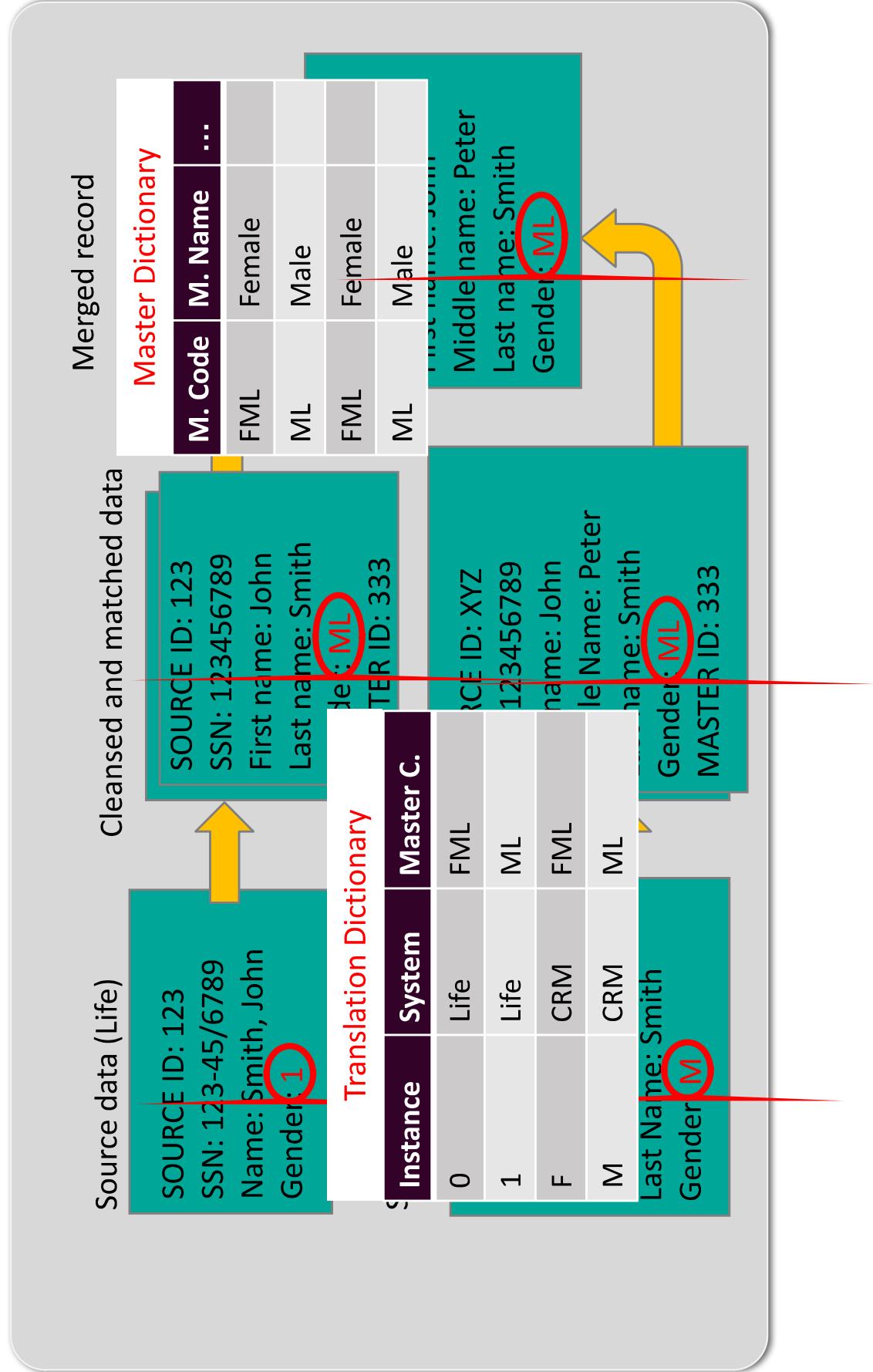
Cleanse, Match and Merge Principle

Real case: mastered party and de-duplicated address (1:N)



Reference Data Principle

single entity and multiple systems



Topic Highlights

Data Storage Layers:

- Source Data Layer: Raw data from source systems.
- Cleansed Data Layer: Standardized and cleansed data.
- Matching Data Layer: Identifies records as single entities.

Integration Patterns: Covers merging principles and integration schema for effective data handling.

Memory Refresher #4 MDM Principles

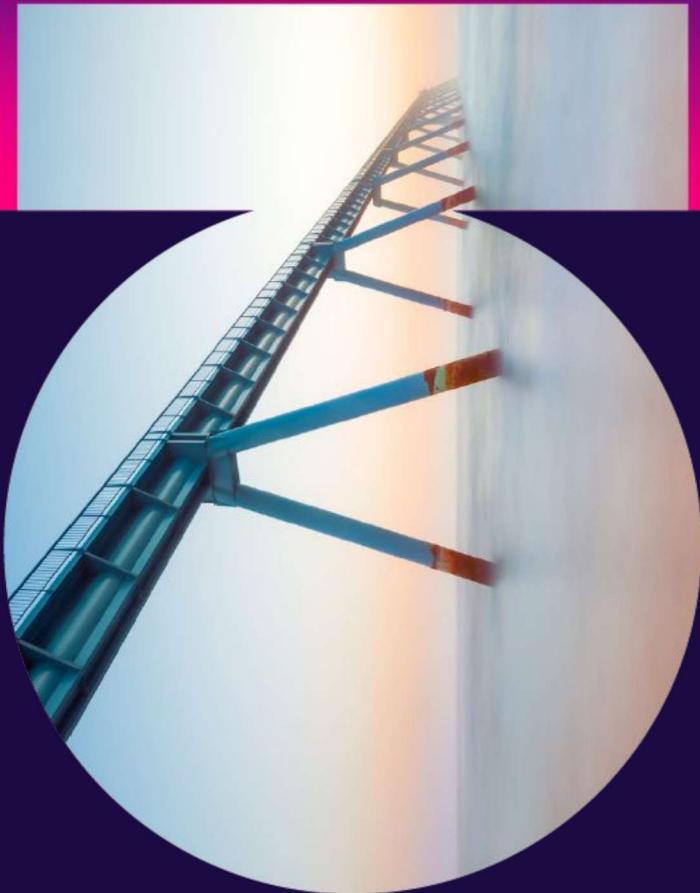


What does the Source Data Layer represent?

- 1.** Data that has been cleansed and standardized
- 2.** Raw data from source systems
- 3.** Matched and merged records

What information does the Matching Data Layer contain?

- 1.** Cleansed and standardized data
- 2.** Records identified as referring to the same entity (e.g., party, product) through cleansing and matching
- 3.** Matched and merged records



MDM Data Modelling

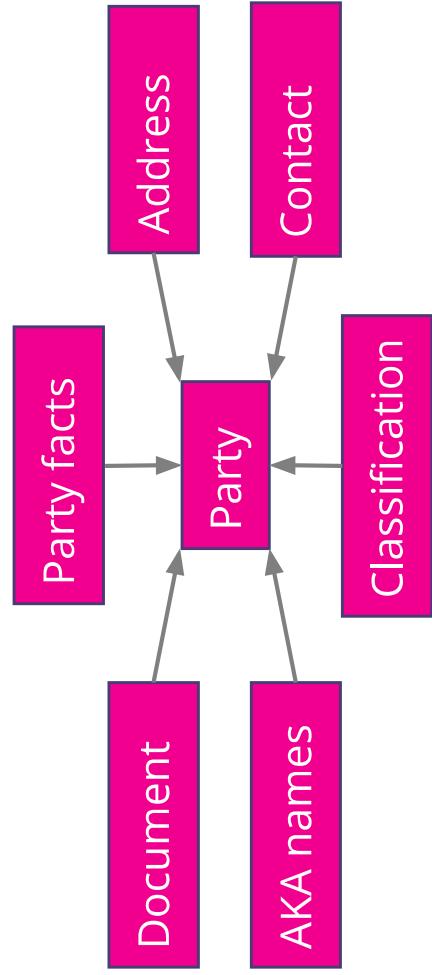
MDM Data Model – Form and Shape

Logical model is sufficient:

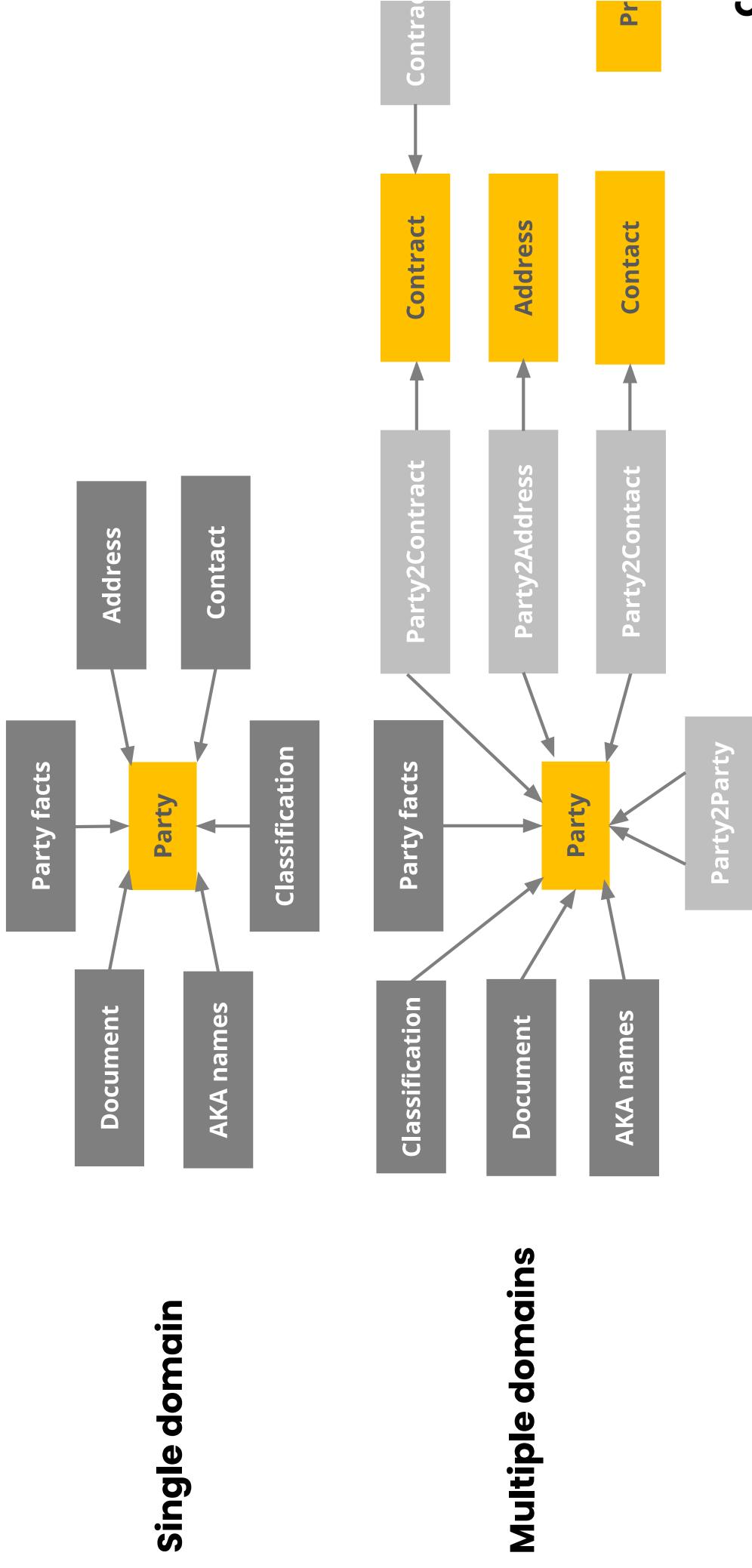
- Physical model may be used as well, but only a part of the information is actually utilized

Star model is easier to implement and provides better performance of the solution:

- Should fit most model requirements
- The top-level entity has a 1:N relationship with satellite entities



MDM Data Model – Domains



Data Stored in MDM

General rule

- Core information identifying master data only, not all available data
- Related data or data which needs to be interconnected in each query

Example for MDM (Customer Data Information) implementation

- Data identifying a customer
- Customer data which needs to be mastered/de-duplicated
 - Addresses, contacts, IDs, classifications etc.

Normalized vs. De-Normalized Model

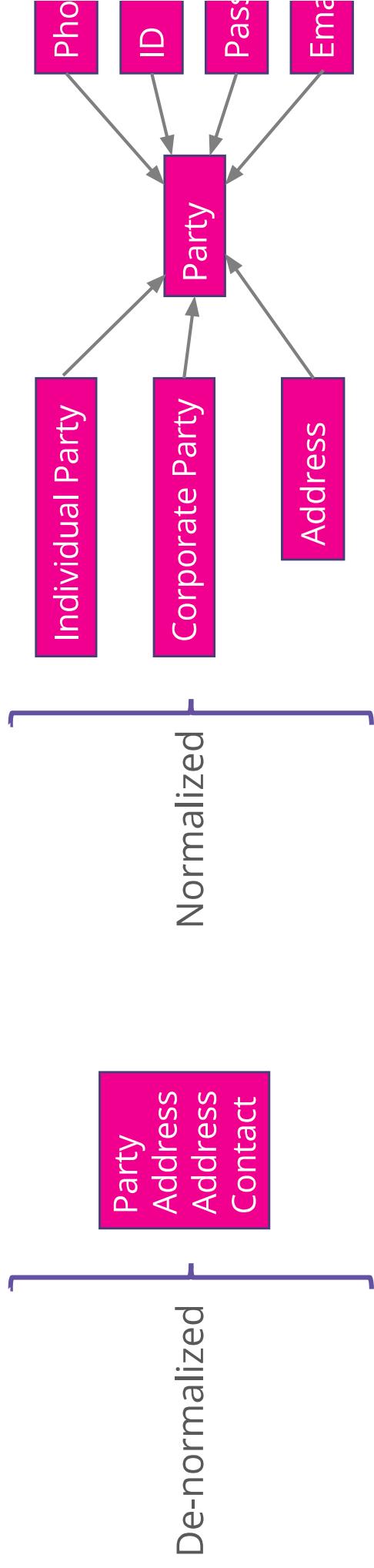
Extreme examples:

Fully Denormalized

- One table containing all attributes from party, address, contact etc.

Fully Normalized

- Multiple tables interconnected via relationships

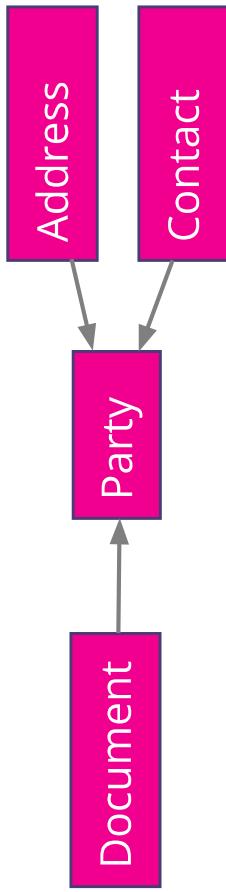


Normalized VS. De-Normalized Model

Should sit somewhere between the two extremes

- Should be divided based on attribute usage and logical entity information (address, contact, ...)
- Too many normalized structures can affect performance because many copy columns are required to take the matching step into account)
- Too many denormalized structures can make it hard or impossible to accurately reflect reality (party usually has multiple addresses/contacts)

Shouldn't be closely related to a source/target system model



Instance vs. Master Model

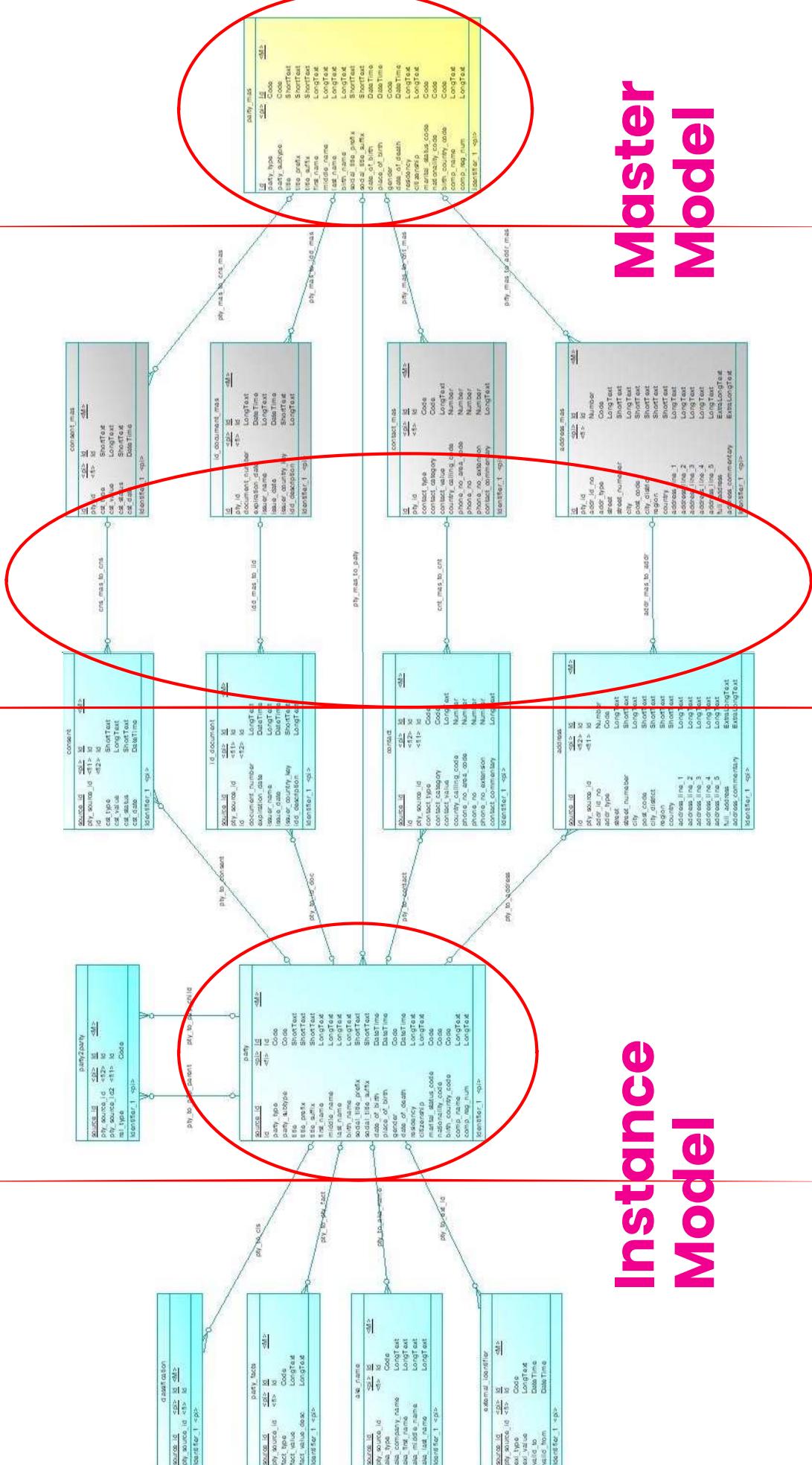
Instance model

- So called 'Canonical Form'
- Should be able to accommodate all incoming source attributes, e.g.,
 - First name
 - Last name
 - Full name

Master model

Usually one common data layout, e.g., First name and Last name

Instance vs. Master Model



Model Relationships

Instance relationships

- Created based on source data
- Enables copy column definition from parent to child and vice versa

Master relationships

- Mandatory between master entities
- Created based on matching
- Relationships between master and instance entities are optional
 - Useful for some of the atomic services only

Cross-instance and master model relationship

- Provided and maintained by Ataccama MDM engine by default

Topic Highlights

Data Models:

- Star Model: Provides better performance and fits most requirements.
- Normalized vs. De-Normalized: Balances between performance and data accuracy.

Instance vs. Master Models:

- Instance Model: Accommodates all incoming source attributes.
- Master Model: Standardizes data layout for consistency.



Memory Refresher #5 MDM Data Modelling

What is a key advantage of the Star Model in data modeling?

- 1.** It allows for a high level of data normalization
- 2.** It simplifies implementation and provides better performance
- 3.** It supports complex data relationships

Which statement accurately describes the relationship types in an MDM system?

- 1.** Instance relationships are created based on matching and are mandatory between master entities.
- 2.** Master relationships are optional and created based on source data, enabling column definitions to be copied between parent and child entities.
- 3.** Instance relationships are created based on source data, enabling the copy of column definitions between master entities and are mandatory between master entities and are created based on matching.



MDM Storage

MDM Storage

- Created automatically when the server starts for the first time
- The generation of the structures for the storage is based on the model defined
- The MDM storage structure is transparent to users

Implementation details of this structure are completely hidden from the users of the **MDM hub**

- Logical transaction is one of the reasons, but not the most important one
- Currently an open relational structure
- Both Instance and Master data are stored in the Master Data Hub

It is not allowed to modify an existing entity or records in repository!

MDM Storage – Data Types

Created automatically when the server starts for the first time

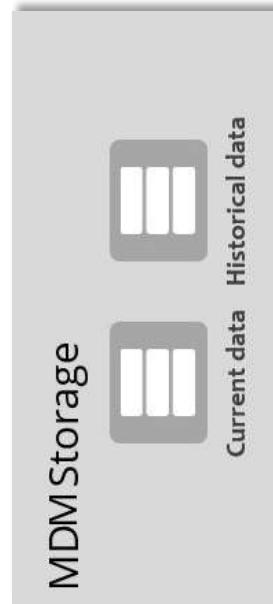
- The generation of the structures for the storage is based on the model defined
- Includes a repository of cleansed, matched and mastered data

Historical data

- Repository stores previous versions of both instance and master data
- Contains data in a defined scope

Current data

- Repository stores the instance records, or source records, from various source systems
- All records from the source systems are stored in their cleansed form
- The Matching Key tables for all matched entities
- Other technical tables



Topic Highlights

Types of Data:

- Historical Data: Stores previous versions and historical records.
- Current Data: Contains cleansed, current instance records from various sources.

Storage Details:

- The structure is hidden from users and includes technical tables for managing data.

Memory Refresher #6 MDM Storage

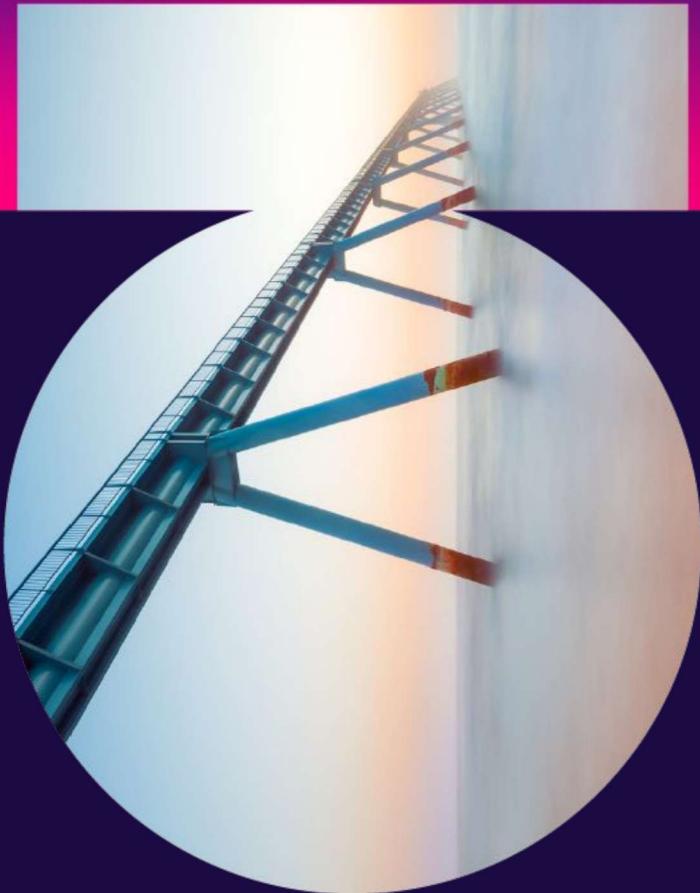


What type of data in MDM storage keeps previous versions of records?

- 1.** Transactional Data
- 2.** Historical Data
- 3.** Reference Data

Which statement accurately describes the relationship types in an MDM system?

- 1.** Instance relationships are created based on matching and are mandatory between master entities.
- 2.** Master relationships are optional and created based on source data, enabling column definitions to be copied between parent and child entities.
- 3.** Instance relationships are created based on source data, enabling the copy of column definitions between master entities and are mandatory between master entities and are created based on matching.



Operational Configuration

MDM Server Configuration

Spring-boot server for MDM engine

- References Runtime Configuration
- References Logging Configuration
- Components
 - MdaBackendComponent
 - HTTP Listeners
 - Health Status (HealthStateComponent)
 - Workflows, Schedulers
- Issue Management (DqitsServerComponent, DqitmServerComponent)
- VirtualFileSystem
 - Ports
 - Paths

MDM Runtime Configuration

- Database Connections
- Path Variables
- Additional Server Connections Server (SMTP, SFTP, etc.)
- Workflow Resources
- Use MDM Runtime Parameters for parallelism tuning

Start/Stop MDM Server

- Requires DB permissions (DDL rights)
- Requires configuration files (incl. generated project specific files)
- Run Locally from MDM ONE Desktop:
 - Files/bin/start-keycloak.bat
 - Files/bin/start-db-postgres.bat
 - Files/bin/start-mdm-server.bat (! wait until started !)
 - Files/bin/start-mdm.bat

MDM Development Procedure

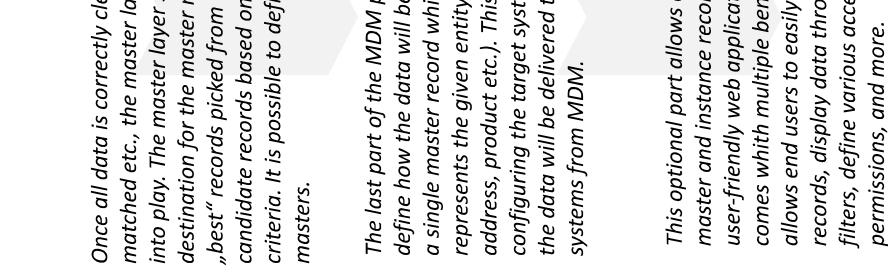


The first step of any MDM project is the creation of the Instance Model (IM). The model defines how the data is structured, how the entities are related, and where the data comes from. The ONE Desktop provides an easy-to-use GUI for this which does most of the configuration automatically.

This part deals with defining source systems and their entities and relationships. These can be different from the instance entities and relationships, so it is necessary to map each source system to the instance layer. Another important part is the generation of load plans which are used to load data from the source systems to the MDM Database.

Cleansing is a common part of any project. The incoming data usually needs to be cleansed before it can be processed further (matched, merged etc.). Default cleansing plans are created automatically for every entity. They serve as a decent basis for adding custom data transformations required by your project.

Matching is essential for any MDM project. Creating and maintaining effective matching rules can be tricky, as it needs to be precise enough to eliminate incorrect matches, but also simple enough to not slow down the performance of the whole MDM solution.



Topic Highlights

Configuration Aspects:

- Server Components: Includes HTTP listeners, health status components, and issue management.
- Runtime Configuration: Involves database connections and server settings.

Development Procedures:

- Includes starting and stopping the MDM server, with specific commands and configuration requirement

Data Storage:

- Core information identifying master data.
- Related data for integration in queries.

Modeling Approaches:

- Instance Relationships: Created based on source data.
- Master Relationships: Created based on matching; optional relationships with instance entities.



Memory Refresher #7 Operational Configuration

What does the runtime configuration for the MDM server involve?

- 1.** HTML project layout
- 2.** Security protocols
- 3.** Server and database connections

What must you do to start the MDM server locally from the ONE Desktop?

- 1.** start-mdmn-server-cloud-local.bat
- 2.** start-mdmn-server.bat
- 3.** start-keycloak.bat