Distribution of MDM Hubs

Addressing the MDM Challenge in Large Enterprises

Without proper planning, Multi-domain MDM implementations in large enterprises can quickly become unwieldy and inflexible. A distributed MDM architecture is often used to strike an optimal balance between enterprise-level and business-unit requirements.

In these cases, it means the Enterprise-level MDM Hub implementation will face requirements to distribute Hub processing across multiple Hub instances. These requirements are driven by a number of factors including:

- Business process differences between various business units, resulting in the need for business-unit specific data model extensions, lookup tables and workflow. *This is extremely common in large enterprises, and one of the most common drivers for a distributed MDM Hub architecture.*
- Global implementations requiring locale-specific data management within the Hub, either on a regional or country-by-country basis.
- Local privacy laws governing the use, protection and physical location of Personally Identifiable Information (PII).
- A need to consolidate data residing in multiple, existing MDM Hubs – for example, consolidating data from several operating companies in a large conglomerate where each operating company has its own MDM Hub.
- Global business processes, including business continuity processes, requiring the MDM Hub to remain active in one region in the world while inactive in other regions.

A distributed MDM architecture also addresses many practical issues surrounding resource management and the perceived/real business value MDM delivers across multiple business units.

Business units often have competing requirements/timing/priorities that are likely to conflict with one another. A distributed architecture helps ease development/support personnel contention issues (i.e. one business unit having to take a back-seat to another based on priority), which otherwise can seriously impact project implementation timelines, total cost of ownership (TCO) and overall business satisfaction with the MDM implementation.

These requirements drive a need to implement MDM Hubs in an architecture that distributes the master data management across multiple hubs and, in many cases, multiple layers of hubs. This document describes the strategies for distributing an MDM solution and the best practices for implementing and managing such a configuration.

Description of Architectural Approaches

There are several architectural approaches that may be used to distribute the MDM Hub, and they can generally be classified as follows:

- **Approach 1: Single Hub Instance with Single Repository**
  - Option A: Single Data Model per Repository
  - Option B: Multiple Data Models per Repository
- **Approach 2: Single Hub Instance with Multiple Repositories**
  - Option A: Single Data Model per Repository
  - Option B: Multiple Data Models per Repository
- **Approach 3: Multiple Hub Instances with Single Repository per Instance**
  - Option A: Single Data Model per Repository
  - Option B: Multiple Data Models per Repository
- **Approach 4: Hub of Hubs**
  - Option A: Registry of Hubs
  - Option B: Subset Master Hub

Choosing which architecture is most appropriate requires an understanding of the Customer requirements and an understanding of each of the architectures.

Distributed MDM Hub Architectures

Each architecture differs in the number of instances of the Hub that are deployed, the number of repositories per instance, the number of data models per repository and the MDM style used.

PLEASE NOTE: *Informatica licensing implications may be associated with a particular architecture choice, so please consult with your Informatica Account Management team to ensure compliance with your license agreement. The definitions below are used only for the purpose of describing the technical product architecture, and may not translate directly to your Informatica license agreement, which is governed by the applicable "Informatica Product Description Schedule".*

DEFINITIONS:

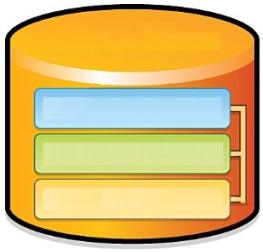
Instance: An MDM Hub with a single point of system management with the common MDM Hub configuration settings such as user accounts, security configuration, etc.

MRS (Master System Repository): The common MDM Hub configuration settings.

ORS (Operational Reference Store): An associated MDM repository containing a set of master data, content metadata, and the rules associated with mastering, processing and managing that data. A typical use case for a multiple ORS architecture is to separate Production, QA and Development each into their own ORS within the same MDM instance.

Data Model: Each repository may contain one or more ***data models***, which describe how business entities are represented in the MDM Hub, and each set of related MDM Hub entities (for example Customer or Product) may be represented in its own data model, or it may be represented in a single data model within the repository.

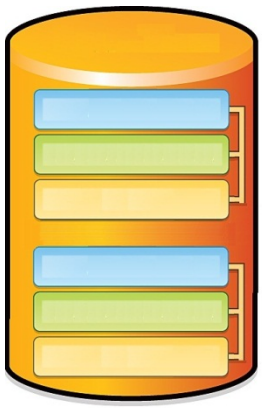
Approach 1: Single Hub Instance with Single Repository



Option 1A
Single Data Model

This architectural approach is often a good way to address two of the driving requirements for distributed hubs:

- Business process differences between various business-units which result in the need for business-unit specific data model extensions, lookup tables and workflow. For example, a retailer organized into Apparel, Electronics, and Automotive units might need to master size and color for the Apparel business-unit, but not for Electronics or Automotive. On the other hand, the Electronics business unit might require a process to add associated service plans when adding a new product, and the workflow for Electronics would need to deal with this, while Apparel and Automotive business units do not.
- Global implementations which need locale specific data management within the Hub, either on a regional or country-by-country basis.



Option 1B
Multiple Data Models

In this approach, the MDM Hub is configured to have one Master System Repository and one ORS with one (Option 1A) or more (Option 1B) data models defined. The choice of whether to have one or more data models is usually determined by looking at whether individual mastered entities will interact with other mastered entities, or whether locale specific data model extensions are required.

For example, one might choose to model Customer and Product as a related set of entities, which would look like Option 1A. Alternatively, one might chose to keep these models separate (but in the same ORS), and this would look like Option 1B. In either case, one might have a set of locale specific lookup tables, which would also look like Option 1B.

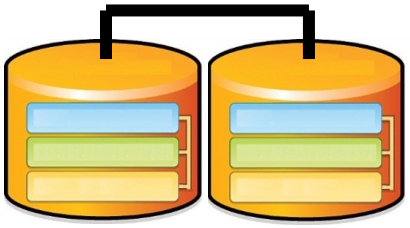
This is by far the simplest approach for supporting multiple user communities, but there are several limitations associated with this approach:

- Because MDM Hub configuration is common across the Hub instance, all users are subject to the configuration. That is, user properties, permissions and access control are common across all uses of the data. While this is usually acceptable, there are certain cases – for example, when Master Data contains PII for customers across multiple countries - where the data needs to have different security policies at various locations. If this is the case, multiple hub instances will be required.
- Data Stewards will have access to all data in the Hub (based on their role). In order to limit Data Steward access to portions of the data set in this architecture, the data steward role will need to be divided into multiple roles, each with specific access control capabilities.
- Rules and rules sets are applied consistently across all data. In this approach, different rules sets cannot easily be applied against a subset of the data.
- Repositories must be managed as a single unit – you cannot modify part of the Hub without affecting the rest of the Hub.

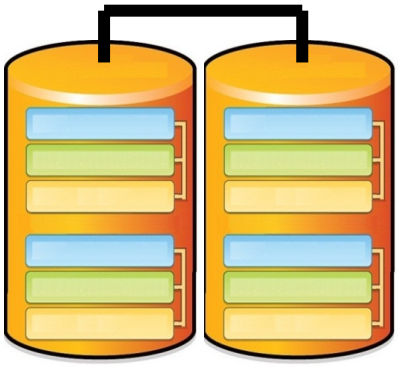
The advantages of this approach are:

- It is very easy to maintain a common core data model across the hubs.
- Adding additional business units or geographies is simplified.
- Administration of the overall Hub is simplified.

Approach 2: Single Hub Instance with Multiple Repositories



Option 2A
Single Data Model



Option 2B
Multiple Data Models

This architectural approach is often a good way to address three of the driving requirements for distributed hubs:

- Business process differences between various business-units which result in the need for business-unit specific data model extensions, lookup tables and workflow.
- Global implementations which need locale specific data management within the Hub, either on a regional or country-by-country basis.
- Local privacy laws governing the use, protection and physical location of Personally Identifiable Information (PII).

In this approach, the MDM Hub is configured to have one Master System Repository and multiple ORS with one (Option 2A) or more (Option 2B) data models defined. As with Approach 1, the choice of whether to have one or more data models is usually determined by looking at whether individual mastered entities will interact with other entities, or whether locale specific data model extensions are required.

While it is more complex than Approach 1, this approach removes some of the configuration limitations. In particular, different rules and rule sets may be defined in each ORS allowing a very easy way of defining and managing Business Unit or Geography specific rules.

The advantages of this approach are:

- Data can be physically segmented without complicating system management.
- Data Steward access may be limited to a particular ORS, which allows a reasonable level of access control without complicated configuration issues arising.
- Business Unit or Geography specific rules can easily be applied at the ORS level.
- It is relatively easy to maintain a common core data model across the hubs.
- Adding additional business units or geographies is straightforward.
- Repositories may be managed independent of each other.

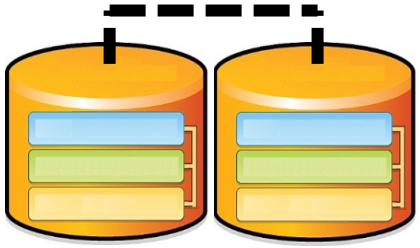
The disadvantages of this approach are:

- As with Approach 1, user properties, permissions and access control are common across all uses of the data.

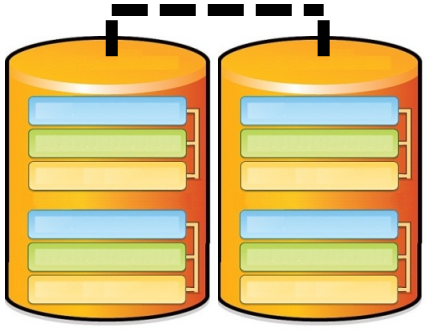
Approach 3: Multiple Hub Instances each with Single Repository

This architectural approach is often a good way to address all of the driving requirements for distributed hubs:

- Business process differences between various business-units which result in the need for business-unit specific data model extensions, lookup tables and workflow.
- Global implementations which need locale specific data management within the Hub, either on a regional or country-by-country basis.
- Local privacy laws governing the use, protection and physical location of Personally Identifiable Information (PII).
- A need to consolidate data residing in multiple, existing MDM Hubs – for example, consolidating data from several operating companies in a large conglomerate where each operating company has its own MDM Hub.



Option 3A
Single Data Model



Option 3B
Multiple Data Models

In this approach, the MDM Hub is configured to have multiple Master System Repositories, each with one ORS and one (Option 3A) or more (Option 3B) data models defined for each ORS. As with Approaches 1 and 2, the choice of whether to have one or more data models in each ORS is usually determined by looking at whether individual mastered entities will interact with other entities, or whether locale specific data model extensions are required at the ORS level.

From both an implementation and a production support point of view, this configuration is much more complex than either Approach 1 or Approach 2. However, Approach 3 provides capabilities for distributing the MDM Hub that neither of the previous approaches supports. Not only can data be physically segmented as with Approach 2, but the *management* of the data (both data stewardship and system management) can also be segmented.

The advantages of this approach are:

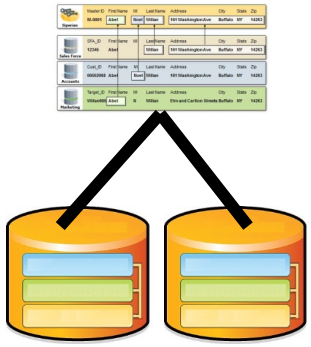
- Data can be physically segmented without complicating system management.
- Data Steward access may be limited to a particular ORS or to a particular instance of the MDM Hub, which allows complete access control at this level.
- User properties, permissions and access controls are defined independently for each instance of the Hub and therefore have no dependency on the other instances. These properties, permissions and access controls can, in fact, differ from those defined in other instances of the Hub.
- Business Unit or Geography specific rules can easily be applied at the ORS level or at the Hub instance level.
- Repositories and hub instances may be managed independent of each other, even to the extent of taking one instance off-line while other instances remain active.

The disadvantages of this approach are:

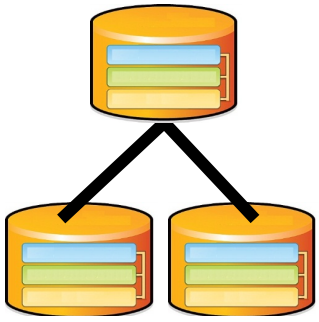
- System configuration and management is significantly more difficult using this approach, as each hub instance needs to be managed independently.
- Similarly, maintaining a common core data model across the hubs requires significantly more discipline than in the previous two approaches.
- Adding additional business units or geographies is more complicated as these additions need to be applied across independently operating Hub instances.
- Synchronizing data across the Hubs is a significant effort and will require discipline and appropriate process to maintain data consistency.

Approach 4: Hub of Hubs

In this approach, an MDM Hub provides the top level data that spans across the lower level hubs. This high level Hub is often referred to as the ***Hub of Hubs***, since it often provides additional mastering of data across several MDM instances. This Hub of Hubs is usually set up as a Registry style MDM Hub (Option 4a), but can also be any MDM style that instantiates a golden record (Option 4b) – Consolidation, Coexistence or Transaction.



Option 4a
Registry of Hubs



Option 4b
Subset Master Hub

As an example, if two operating companies within a conglomerate each have their own employee hub, the Hub of Hubs would provide data to the conglomerate that is common across the corporation, while each operating company would maintain their own hub containing master data that is used and managed locally.

In this architecture, a common core data model is defined for the Hub of Hubs, and the appropriate interfaces are defined to specify how data from each local hub will be transported and transformed into the Hub of Hubs, and how data from the Hub of Hubs would be consumed by each lower level hub.

This approach has significant complexity involved with its implementation, but in many cases may have less operational complexity than Approach 3. The Hub of Hubs approach assumes that the master data may be segmented across business unit or geographical boundaries, and therefore the operational need for synchronizing data found in Approach 3 is generally not required in this approach.

This architectural approach provides capabilities for distributing the MDM similar to those described in Approach 3. In addition, this approach supports heterogeneous Hubs – that is, the hubs do not need to have similar data model version or even vendor in order for this Approach to be useful. .

The advantages of this approach are:

- Data can be physically segmented without complicating system management.
- Data Steward access may be limited to a particular ORS or to a particular instance of the MDM Hub, which allows complete access control at this level.

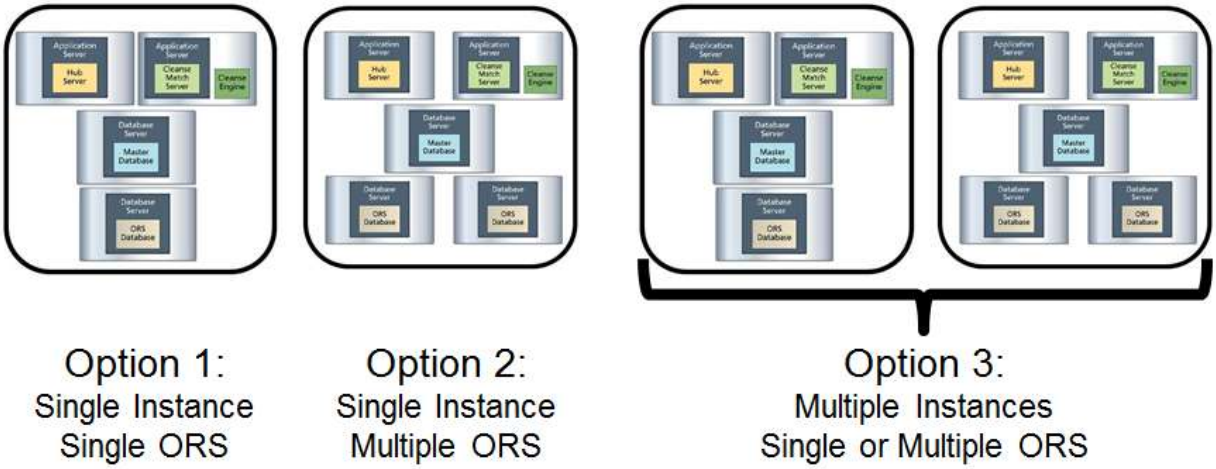
- User properties, permissions and access controls are defined independently for each instance of the Hub and therefore have no dependency on the other instances. These properties, permissions and access controls can differ from those defined in other instances of the Hub.
- Business Unit or Geography specific rules can easily be applied at the ORS level or at the Hub instance level.
- Repositories and hub instances may be managed independent of each other, even to the extent of taking one instance off-line while other instances remain active.
- Heterogeneous Hub environments may be supported.
- Synchronizing across Hubs is not required.
- Adding additional business units or geographies is easier than in Approach 3 as these additions will be in the form of additional Hub instances and should require little to no change to the hub of hubs (i.e. new lower level hubs will typically be expected to meet the existing SLA and interfaces of the Hub of hubs).

The disadvantages of this approach are:

- System configuration and management is significantly more difficult using this approach, as each hub instance needs to be managed independently, along with the Hub of Hubs.
- Defining a common core data model across the hubs requires business owners across multiple parts of the organization to agree on the definition of the common core elements and the survivorship rules associated with them, and this can be a significant political challenge.

Physicalizing the Distributed MDM Hub Architectures

There are many possible ways in which you can set up the various Informatica MDM Hub components in your environment. This section provides three examples of Informatica MDM Hub installations. Their purpose is *instructive* rather than prescriptive—they illustrate some general principles to consider while designing your installation. However, each architecture shown below is live today in production at an Informatica customer.



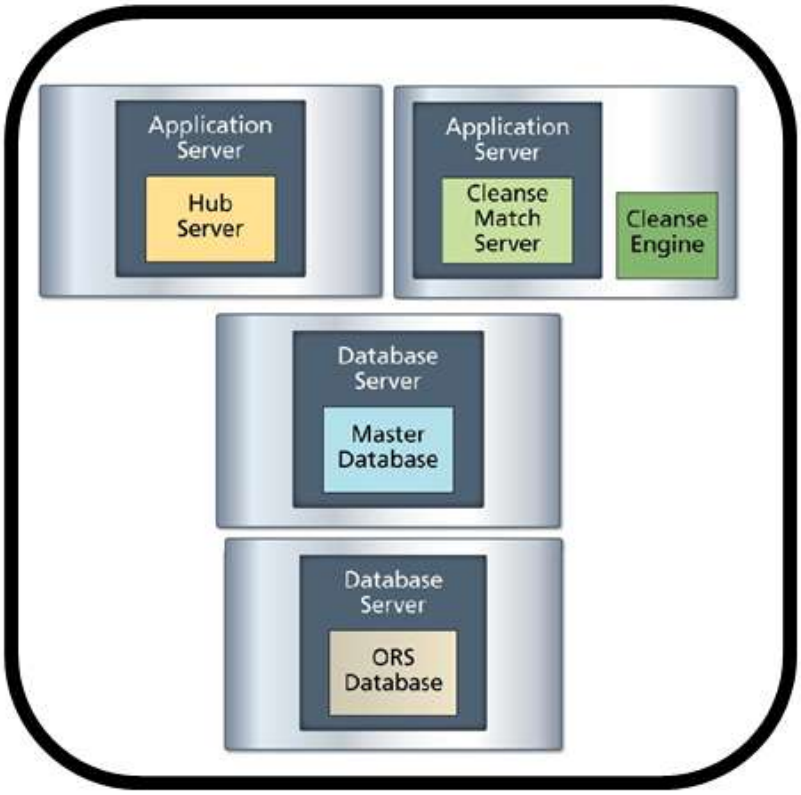
Option 1 is a single instance of Informatica MDM Hub with a Single ORS hosted in the same environment.

Option 2 is a single instance of Informatica MDM Hub with multiple ORS. A variant of Option 2 (not shown) has a single instance of Informatica MDM Hub with multiple ORS, where at least one ORS is not hosted in the same environment as the Master System Repository (but *is* in the same database instance).

Option 3 is multiple instances of Informatica Hub connected via a Data Integration tool such as PowerCenter, with each instance either single ORS or multiple ORS.

Single Installation with Single ORS

In the following example, all Informatica MDM Hub core components are installed in a single host environment.



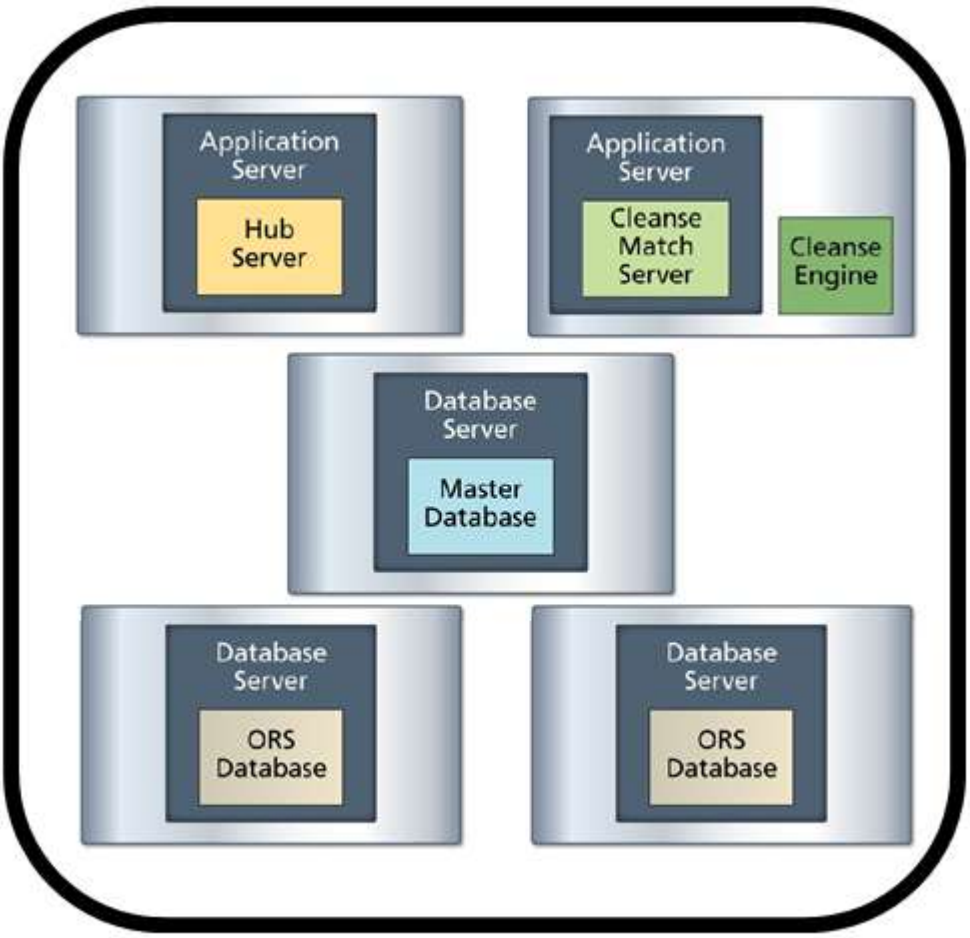
This layout simplifies communication among the components within a single host environment. The application server contains the Hub Server and the Cleanse Match Server while the Database Server contains the Master System Repository and a single ORS. The Cleanse Engine (for example, Informatica Data Quality) is separately installed into the same environment.

The picture above is a logical representation of the architecture. It is a design decision as to how many physical servers are used. Typically, most implementations of Informatica MDM Hub have one physical Database Server with the Master System Repository and the ORS on that server and one physical Application Server, with the Hub Server and Cleanse/Match Server deployed. However, the Master System Repository and ORS do not need to be on the same physical box (although they do need to be in the same database instance), and the Hub Server and Cleanse server may also be separated.

Distribution across multiple physical nodes in this option is performed by the underlying application server and database technologies; Informatica MDM Hub views these nodes logically and acts as if they are in a single logical instance.

Single Installation with Multiple ORS

In the following example, all Informatica MDM Hub core components are installed in a single host environment.



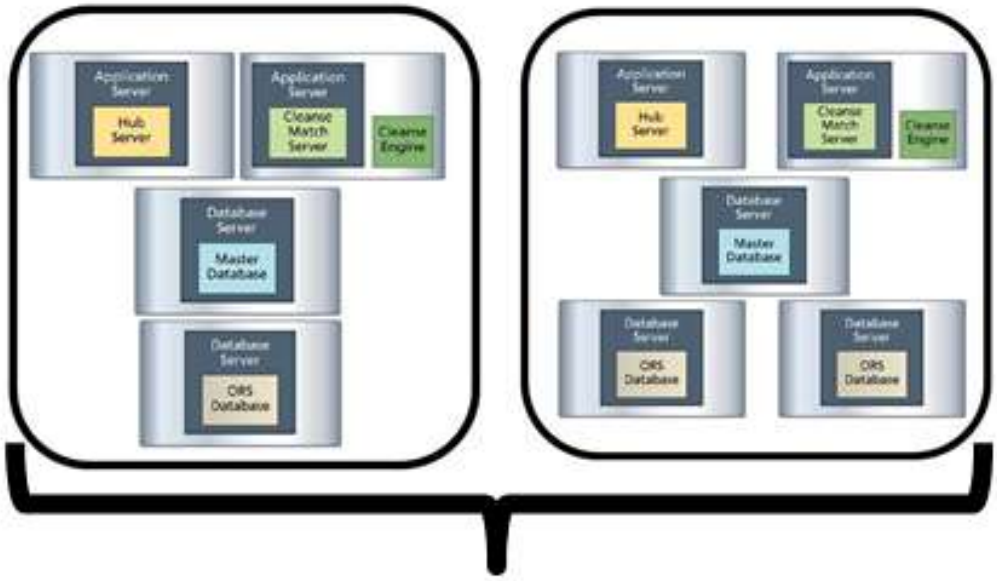
This layout is the same as the previous layout, but has multiple ORS associated with the same Master System Repository to allow for separation of operations.

Distribution across multiple physical nodes in this option is performed by the underlying application server and database technologies; Informatica MDM Hub views these nodes logically and acts as if they are in a single logical instance.

PLEASE NOTE: Multiple ORS architectures may require additional Informatica licensing.

Multiple Installations with Single or Multiple ORS

In the following example, the Informatica MDM Hub core components are installed in multiple host environments.



This layout has complicated communication between the components, but maximizes the separation of environments such that one instance can be taken completely off-line while the other instances continue to operate. Each single instance in this layout is one of the two physical layouts described above.

In this option, changes in one instance of the hub need to be replicated into the other instances. In order to do this, Hub publishing needs to be configured to determine which events should cause an update to occur in the other hubs. Typical events that should be considered include Add New Data, Update Existing Data, Merge Records, Unmerge Records, and Delete Data. Configuration is through a drag-and-click user interface. Once the system is running, configured events will be published and may be consumed by the other Hubs in several ways. The two most popular ways of doing this are:

- Publish changes via JMS queues to the other Hubs. The Hub puts the changes in a change table which indicates the changed record and the target to whom it will be published, and message queues are used to pass these changes to the appropriate target (in this case, other hubs).
- The change table can also be read and managed by a data integration tool such as PowerCenter and moved between hubs as a standard ETL job.

Once the data arrives at the target Hub, standard processing can be used to load this data into the target Hub, either via SIF or via standard batch jobs called from PowerCenter as appropriate.

Choosing a Distributed MDM Architectural Approach

The previous section described the four approaches for distributing the MDM Hub, the advantages/disadvantages of each, and an approach for physicalizing these architectures. This section describes Informatica's best practice to determine which architecture to choose.

Consider the Usage Requirements

There are a number of specific points of interest which will help refine which approach is most appropriate:

- Are there any special security concerns?
  - Security concerns usually require the special treatment of one or more classes of users with their own access control. If this is the case, Approaches 1 and 2 are probably not viable since they leverage the same security infrastructure across all users.
- Are there any requirements for physical location of the data?
  - If there is a requirement to physically segment the data such that some of it stays within a particular geographic boundary (for example, restricting certain PII from leaving the borders of the European Union), Approach 1 is not viable, and Approach 2 is only viable if a single database instance (with multiple ORS located in multiple countries) can be used.
- Will System Management/Configuration be done locally or globally?
  - If local administration is required, Approaches 1 and 2 will not work.

Identify what needs to be unique between hubs

Points that should be considered are:

- Are there business unit or country specific extensions or lookup tables required?
  - Any of the approaches will work for this, but in the case of Approach 1 and Approach 2, using multiple data models will optimize the solution.
- Are there any requirements to use different match populations in different parts of the world?
  - If the populations need to be different for the same record in different locales, then Approach 1 should not be used.
- Are there any interfaces (system level or user interface) unique to a particular locale?
  - If the interface is unique to a specific locale (other than localization of the interface), the hub will need to have appropriate connectivity developed. While Approach 1 and Approach 2 can be made to integrate in such architecture, Approaches 3 and 4 are better suited to it.
- Are there business workflow differences that materially affect the design of your consolidated data or the access to it?
  - If there are such differences, Approach 1 will likely not be suitable for your implementation.

Choose the Least Complicated Approach

Informatica recommends that you evaluate your implementation against the following chart and determine the simplest approach (i.e. the one furthest to the left in the chart) that supports your requirements.

REQUIREMENT	APPROACH							
	1a	1b	2a	2b	3a	3b	4a	4b
Common Data Model (core attributes)	√	√	√	√	√	√	√	√
Locale specific configuration for access control	√	√	√	√	√	√	√	√
Privacy laws governing use of data	√	√	√	√	√	√	√	√
Common Data Model (all attributes)	√	√	√	√	√	√		
Relationships across all entities	√		√		√			
Relationships across a subset of entities		√		√		√		
Locale specific data model extensions			√	√	√	√	√	√
Locale specific lookup tables			√	√	√	√	√	√
Locale specific workflow			√	√	√	√	√	√
Regional level data management			√	√	√	√	√	√
Privacy laws governing protection of data			√	√	√	√	√	√
Locale specific configuration for permissions			√	√	√	√	√	√
Country level data management					√	√	√	√
Privacy laws governing physical location of data					√	√	√	√
Stay active in a region and inactive in another					√	√	√	√
Locale specific configuration for user properties					√	√	√	√
Link data from existing heterogeneous hubs							√	
Consolidate from existing heterogeneous hubs								√
May require additional MDM software licensing			√	√	√	√	√	√