

MDM Components and the Maturity Model



MDM is an evolution that involves more than just technology.

One common mistake that takes place at the highest levels of an organization is the assumption that any good idea arrives with everything it needs to be successful. But make no mistake: there is no silver bullet for *any* enterprise information initiative, let alone master data management (MDM). Information professionals recognize that master information consolidation is “the right thing to do,” but that does not necessarily imply that there are always going to be acute business requirements that support a drastic upheaval of an information management program.

The migration to an organization that relies exclusively on master data management does not take place overnight; the shift evolves through a number of transitional information management stages. Recognizing that the process involves more than purchasing a software package or engaging outside solution vendors is the first step towards achieving the MDM evolution. But it is more than that – it means understanding the essential capabilities necessary for a successful MDM deployment and the degree of maturity of those capabilities necessary to make MDM actionable.

No “functionality list” completely captures the inventory of services that a specific business requires from its master repository. However, it is worthwhile to explore a high-level enumeration of core MDM capabilities, and in this white paper we will provide a conceptual outline of technical MDM components. This white paper explores levels of maturity based on the ability to provide MDM services. By presenting the MDM component layers in terms of their maturity, enterprise architects can target a desired level of MDM maturity and develop a design and implementation roadmap that articulates the steps to take when assembling an MDM program.

MDM Basics

The proliferation of enterprise-level applications (along with expectation for shared, synchronized information) drives the need for the development of a single view of the key data entities in common use across the organization. At the technical level, the drivers and fundamentals of MDM can be summarized as processes for consolidating variant versions of instances of core data objects, distributed across the organization, into a unique representation. In turn, that unique representation is continually synchronized across the enterprise application architecture to make master data a shared resource. The result is a master repository of uniquely identified key data entity instances that are integrated through a service layer with applications across the organization.

Like many technology projects, the devil is in the details. To accomplish what may seem to be a relatively straightforward set of ideas, the organization must prepare for the technical, operational, and management challenges that will appear along the way. In fact, the deployment of an MDM solution could evolve through a number of iterations, introducing data object consolidation for analytic purposes as an initial step, then following on with increasing levels of integration, service and synchronization.

The end-state master data management environment exists as an enterprise resource that is integrated with the enterprise application architecture through a collection of provided services. At the very least, a mature MDM solution will encompass the capabilities and services displayed in Figure 1.

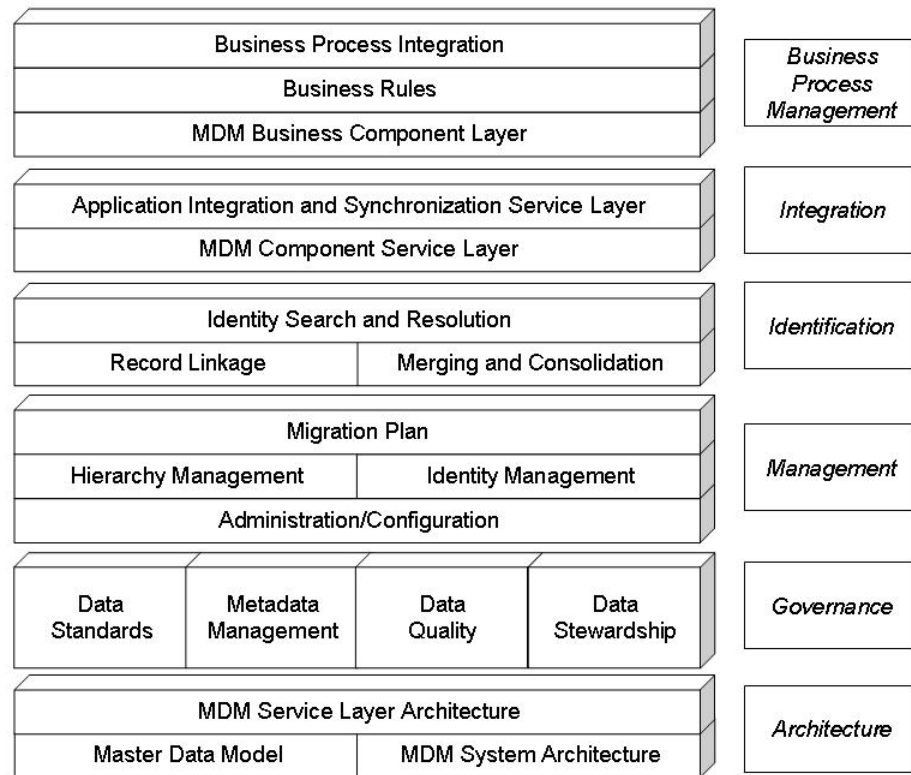


Figure 1: MDM Component and service model.

While these layers represent a model of the necessary technical, operational, and management components for MDM, organizations can launch the program even with various levels of maturity for each of these components. The parts of this “component model” can be grouped into conceptual architectural levels: architecture, then governance, management, identity management, integration services and business process management. Examining those levels of maturity - and their relationship to the business requirements - will guide the MDM program manager in developing an implementation roadmap. Although the architectural levels are presented from the bottom up, the maturity model will provide insight into how selected pieces of the component model can begin to add value to the organization as the implementation grows.

The MDM architecture focuses on the model, systems and services required to support an MDM framework.

Architecture: The Foundation of MDM

Fundamentally, there are three aspects to the MDM architecture that correspond to the structure, power, and control of the environment. The structure is represented by the MDM master data model, the power reflects the MDM system architecture, and the control is encompassed by the MDM service layer.

Master Data Model

To create the conceptual master repository, all data elements in the various different formats and structures that exist across the enterprise need to be consolidated into a centralized resource that can accommodate those differences and, in turn, feed back into those different representations. That implies that there must be a consolidated master representation model to act as the core repository.

Source metadata details can be easily captured and managed in an enterprise metadata registry, and this information can be used to develop a representative master object model for every data object type. The master object model for each master data type should be resilient to the differences between existing replicated data instance models. This suggests creating a model to support all of the data in all of the application models.

The data model thus becomes a complex, but integral, part of any MDM effort. The set of the data attributes of the consolidated model must be a superset of all of the important attributes from each of the application models. And the format and structures for each attribute must support all the formats and structures used for that attribute across the variant models. This defines the fundamental challenge of the master data model: supporting variant structure and formats for both accumulating and publishing of master data objects.

MDM System Architecture

All data objects are subject to a “data life cycle,” with different systems requirements associated with, and affected by, each stage of that data life cycle. The MDM system architecture focuses on the aspects of that life cycle and incorporates the component methods to support them to higher levels of a service layer supporting applications across the enterprise. The MDM system architecture relies on a services-oriented framework, in which the functionality reflects the life cycle activities (create, access, update, retire) as they relate to the master object type.

The core functionality (e.g., creating a new master record, accessing/updating a master record) is presented as a set of low-level component services that can be adapted or enhanced for specific master data types (“customer” or “product”) or specific applications. For example, certain pieces of identifying information can be collected at different times and by different applications. If the different applications are allowed to create a new instance, the creation service may be adapted for each application to acquire what is necessary to complete the business process.

MDM Service Layer Architecture

The MDM system architecture focuses on the core technical components necessary to support the data life cycle. However, as the reliance of applications on the master repository increases, there are further requirements for data object services related to the level of service provided for application use, such as synchronization, serialization, embedded access control, integration, consolidation and access. Business applications are then layered on top of the data object service layer by deploying or possibly reusing specific components associated with business processes.

More comprehensive management activities for master data objects can be implemented at the system level. But because different types of applications may require different levels of service, it may be worthwhile to segregate those components with a role-based framework. For example, some applications that create new master records may have embedded timeliness requirements, such as a customer creation capability that must establish the customer record prior to allowing any purchase transactions. If a “quick-create” capability is needed within the sales organization, but not necessarily within the fulfillment organization, then the quick-create can be established at the service layer along with the service level requirements (e.g., the maximum time allowed between master object creation and its availability for use).

Governance and Oversight: The Policies of MDM

Because MDM is an enterprise initiative, there must be some assurance that stakeholders will adhere to the rules that govern participation and information sharing. A data governance program, applied across different business-level domains, will address issues of data stewardship, ownership, compliance, privacy, data risks, compliance, data sensitivity, metadata management, MDM and even data security. Each of these issues focuses on integrating technical data management with oversight, ensuring organizational observance of defined information policies. There are four aspects to governance, starting at the standardization of common use at the data element level, the consolidation of metadata into an enterprise management systems, managing data quality, and governance/stewardship.

Standardized Definitions

While humans can resolve ambiguous definitions and linguistic differences, application systems do not have this capability. People can resolve missing information or potentially conflicting definitions, although each individual’s translation of a business term may differ slightly from any other person’s definition. This becomes an issue during integration and consolidation when data element instances that may share a name do not share a meaning. Alternately, differently named data elements may not be recognized as representing the same concept.

A process for assessing organizational data element information and weaving that information into business metadata provides standardized definitions that ultimately drive and control the determination of master data objects. With these definitions in place, the organization has an understanding of how these definitions are resolved into the unique view.

Consolidated Metadata Management

A by-product of the process for identifying and clarifying data element names, definitions, and other relevant attribution is the discovery and documentation of enterprisewide business metadata. Aside from collecting standard technical details regarding the numerous data elements that are potentially available, enterprises need to determine:

- Business uses of each data element
- Which data element definitions refer to the same concept
- The applications that refer to manifestations of that concept
- How each data element and associated concepts are created, read, modified, or retired by different applications
- Data quality characteristics, inspection and monitoring locations within the business process flow
- How all the uses are tied together

Because the use of the data elements and their underlying concepts drive how the business applications operate using master data, the enterprise metadata repository becomes the “control center” driving the business applications. Therefore, a critical component of an MDM environment is an enterprise business metadata management system to facilitate the desired level of control.

At an even higher level, the metadata management framework supports the definition of the master data objects themselves (which data objects are managed within the MDM environment and which application data sources contribute to their consolidation and resolution). The framework also manages the frequency of, and processes used for, consolidation – everything necessary to understand the complete picture of the distributed use of master data objects across the enterprise.

Data Quality

Data quality impacts MDM in two ways. First, the concept of the unique representation for each real-world object requires a high level of trust in the data – otherwise there would be little incentive for business clients to participate. Second, data quality tools and techniques are employed in the integration and consolidation processes.

As a result, instituting a data quality management program will ultimately change the way that management, and in turn individual staff members, relate to and assess the information value. Instead of considering data as only the raw input to the operation of the business, individuals grow to understand how information becomes an asset to be

The management function establishes the underlying processes that build a master data repository.

used in many ways for improving the business. As business practices continue to rely on a master repository, they will become more reliant on high-quality data. The recognition that business performance and operational productivity depend on high-quality data – at the organizational and personal levels – becomes a core competency of any MDM program.

Data Governance and Stewardship Program

One of the major side-benefits of an MDM program is that a successful effort will be accompanied by a data governance program. As more lines of business integrate with core master data object repositories, there must be some assurance of that end users will follow the rules that govern data quality.

Yet while MDM success relies on data governance, a governance program can be applied across different operational domains, providing economies of scale for enterprise-wide deployment. The aspects of governance are critical, as the ownership models and oversight mechanisms ensure that the participants in the MDM environment are aware that the quality of the information is actively managed.

Management: The Logic behind MDM

By definition, a master data repository manages the unique index of data entities dealt with by the various business applications. There is a requirement for providing the components for maintaining the special characteristics of these master data objects through the data life cycle while supporting each application's corresponding ongoing needs. This includes the unique identification of each object, and the connectivity between the replicas, instances, and usage points of those objects.

The management function also involves maintaining the ways that different master data objects are possibly connected, such as customer names in a household. Aside from the expected administration and configuration management components, the MDM stack must provide "specialty" management services, including identity management for unique key entities, hierarchy management to track association, lineage, and relationships as well as migration management as part of the transition to the MDM platform.

Identity Management

Every instance of each master data object type must represent a unique real world object, implying the constraint that there is one, and only one, uniquely identifiable record for any specific customer (or product, employee, etc.). This means that any time a process seeks a specific individual from the master repository, enough identifying information must be provided to both determine that either:

- A record for that individual exists and that no more than one record for that individual exists, or
- No record exists and one can be created that can be uniquely distinguished from all others

Identity management addresses these requirements by enabling and managing the determination of the attributes necessary for unique identification. Identity management also includes the search and match capabilities used to locate both exact and approximate matches as well as maintaining the master index based on the identifying attributes.

Hierarchy Management

The first aspect of hierarchy management essentially focuses on the lineage and process of resolving multiple records into a single representation. Since there may be records representing the unique entity in different application systems, part of the consolidation will focus on documenting which application data sources contribute to the master consolidation. In certain types of MDM architectures, this will provide links from the master index to the original source records to materialize master information on demand. Hierarchy management becomes especially important as a data control if it is determined that there are false positive matches (e.g., identifying information for two individual objects incorrectly resolved into a single entry) or false negatives (e.g., more than one master record exist for the same unique entity).

The second aspect of hierarchy management for MDM revolves around the interconnectedness of master objects across multiple systems. For example, customers may be related to each other (e.g., same family, work for the same business), or different master data types may be related (e.g., the products associated with a specific supplier). These relationships are reflected in linkage hierarchies, and the hierarchy management layer will also provide service components supporting the management of these connections.

Migration Management

The transition toward application integration with an MDM system is an interesting contrast to general approaches to application modernization. Whether incrementally or drastically modernizing a standalone application, the migration plan typically will have the version to be retired running simultaneously with the modernized version for some time period to ensure a high confidence that the new version properly addresses the business requirements.

For an MDM program, one objective may be to replace the application's underlying data interactions, which would complicate the ability to have different version operating simultaneously. Therefore, a necessary operational component is the ability to manage application migration and transition to using the master repository.

Administration/Configuration

As there are different architectures and frameworks for deploying the master repository, a master index of identities that is mapped to the repository, and the ways that applications interface and use the master repository, there will be a need for the MDM technical team to configure and administer application.

Identification and Consolidation: Establishing Master Data

The wide spectrum of applications that deal with each type of master data object will eventually need to be integrated into our single virtual master resource. That requires three capabilities: search and match for identity resolution, linkage to connect records together within their appropriate hierarchies, and merging and consolidation of multiple record attributes into a single “best” version of each entity.

Identity Search and Resolution

Identity resolution refers to the ability to determine that two or more data representations can be resolved into one representation of a unique object. This is not limited to people’s names or addresses, since even though the bulk of data (and consequently, the challenge) is person or business names or addresses, there is a growing need for resolution of records associated with other kinds of data. Identify resolution is becoming more important in helping resolve product names, product codes, object descriptions, reference data, etc.

For a given data population, identity resolution can be viewed as a two-stage process. The first stage is one of discovery, combining data profiling activities with manual review of data. Typically, simple probabilistic models can be evolved that then feed into the second stage: scoring and matching for the purpose of record linkage.

Record Linkage

After developing the scoring processes and models as part of identity resolution, the algorithms are then applied to a much large population of records. These records are taken from the different sources, to link – and presumably to automatically establish within predefined bounds – that some set of records refer to the same entity.

Usually, there are some bounds to what can be deemed an automatic match, and these bounds are not just dependent on the quantification of similarity, but must be defined based on the application. For example, there is a big difference determining if the same person is being mailed two catalogs and discerning if the individual boarding the plane is on the terrorist list. The record linkage component services both the identity management capability as well as the processes for merging and consolidation.

Merging and Consolidation

Enterprise data sets are reviewed using identity resolution to distinguish records representing unique entities, and then are loaded into the canonical representation. Record linkage is applied to seek out similar representations, paving the way for the merging and consolidation process. Similar records are subjected to algorithms to qualify the values within each data attribute.

At the integration layer, applications are integrated with master data components.

Integration: The Reality of MDM

The objectives of MDM are not only achieved through data integration. Value is added when the consolidated master data is integrated back into operational and analytical use by the participating applications to truly provide a single, synchronized view of the customer, product, or other master data entity.

The abstraction of the data integration layer as it relates to business application development exposes two ways that master data is integrated into a services-based framework. Tactically, a services layer must be introduced to facilitate the transition of applications to the use of a master repository. Strategically, the core master entities at a data integration layer provide the foundation for establishing a hierarchical set of information services to support the rapid and efficient development of business applications. Fortunately, both of these imperatives are satisfied by a services-oriented architecture (SOA), and these concepts form the next layer of the component model.

Application Integration with the Master Repository

An MDM program that solely accumulates data into a consolidated repository without allowing for the use of that data is essentially worthless. One driving factor for establishing the single point of truth is establishing a high-quality asset that can be shared across the enterprise.

This goal requires a bi-directional flow of information: data must easily enter the master repository and must be just as easily accessible by enterprise applications. Production applications can be expected to migrate to access the master data repository as each application's data sets are consolidated within the master. Therefore, part of the MDM framework must accommodate existing application infrastructures in ways that are minimally disruptive yet provide a standardized path for transitioning to the synchronized master.

MDM Component Service Layer

As MDM becomes more fully integrated into the enterprise architecture, new applications can increasingly rely on the abstraction of the conceptual master data objects and their corresponding functionality to support newer business architecture designs. Standardizing master data object representations reduces the need for application architects to focus on traditional data-oriented issues (e.g., data access and manipulation, security and access control, or policy management). Instead, they can use abstracted functionality to develop services that rely on the lower level data-directed services whose format and design is dictated through the MDM services layer architecture.

The ability to consolidate application functionality (e.g., creating a new customer or listing a new product) using a services layer supplements multiple application approaches favored by most enterprise. This approach will also provide additional value across both existing and future applications.

Business process management allows the MDM framework to directly improve day-to-day business operations.

Business Process Management: The Goal of MDM

The highest level of abstraction, business process management, is the one that exposes the requirements for making application design decisions. Often, application designs are technology-driven, with implementation decisions made based on technical recommendations rather than business needs.

A key (and perhaps, ironic) factor in MDM system design is to ensure that the system is business-driven. Despite the fact that MDM is a technology, it is widely recognized that deploying the technology without linking its functional components to a corresponding business process model is a useless activity. At this component level, the architects incorporate business process modeling with system architecture. Clearly, MDM is differentiated from other types of technology-driven consolidation efforts because of the desire to more closely couple technology inclusion, and that is made possible through business process integration and the use of rules-based operational systems that rely on formally-defined business rules.

Business Process Integration

All business applications should reflect the implementation of business process requirements specified – explicitly or implicitly – as the way the business operations are performed. A business process model is a logical presentation that incorporates the descriptions of a business process in a way that communicates the right details to the right people at the right time. This typically enumerates the processes involved, their inputs, aspects that control the process, the types of events or triggers that emerge as a result of the process, and the expected output of the process. The model's visual representation relies on the underlying metadata, such as activity purpose, timing attributes, operational triggers, process inputs, process duration, generated events, resources used, and the desired outputs.

As individual activities are linked together, the model shows how the outputs of one activity coupled with triggered events from other activities control or influence the behavior of the system. In turn, these business process model descriptions are annotated with the references to the master data objects necessary to complete the procedure. This effectively integrates the business process with the MDM solution, exposing the strict and implicit data dependencies and validating the identification and selection of master data object classes.

Business Rules

Within any business process model, the logic employed for executing a particular operation combines the evaluation of the values of shared data objects and the values expressed by defined controls. The values are examined to determine the actions to take that will create new values and trigger new controls.

The MDM maturity model aligns current IT strategies with an ultimate goal of a single view of the enterprise.

There are two ways to look at a specific implementation. The first is explicit: embedding the logic within application program code to evaluate the data values and specifically executing the actions. The second, more abstract approach is to systematically use descriptive rules to examine variable values, or trigger actions, used to establish the consistency of overall system state.

The way that staff interacts with the events, controls and inputs associated with the business process model provides the details of the business logic that will ultimately be deployed as formal business rules. Reviewing the business process model enables the application designers to identify key triggers for specific rules. This process also exposes the conditions that need to be addressed during the business process. This review process leads to a more complete model and its corresponding master data dependencies.

MDM Business Component Layer

Underlying the definitions and requirements exposed through the business process modeling and integration component and the implementation of business rules through a rules-based system is the business component layer. At this layer, we begin to see the creation of more sophisticated reusable business services (as opposed to the functional services that address interaction with the master data). We also start to see reliance on more interesting master data objects.

For example, in addition to referring to master customer records, we might also begin to integrate master customer profiles within predictive analytics embedded within operational applications. The migration towards the use of the master model will open up opportunities for creating analytic-oriented master data object types and combine their use with traditional operational applications.

The MDM Maturity Model

Our objective in defining a maturity model is not to provide a benchmark against which all MDM implementations are measured. Rather, many organizations have already designed, architected, coded and deployed various versions of the described capabilities. Therefore, the level of maturity describes both how the use of already-deployed components and/or services can be exploited for the purposes of a master data repository. And it suggests the missing capabilities that should be acquired to advance to more sophisticated application reliance on master data.

Initial

The **initial** level of maturity is characterized more by the absence of capabilities than the alternative. At the initial level, there are limited possibilities for exploiting master data, but there is some degree of recognition that there are replicated copies of certain data sets that are relevant to more than one application. At the initial level, some business and technical managers are prepared to explore ways to consolidate data sets for some analytic purposes.

At the initial level, organizations have only the basic components of MDM in place.

Table 1 - Characteristics of the initial level.

Component Layer	Capabilities
Architecture	<ul style="list-style-type: none"> • Application architectures are defined for each business application • Limited enterprise consolidation of representative models • No master data models • Collections of data dictionaries in various forms
Governance	<ul style="list-style-type: none"> • Limited data cleansing by application/line of business, for specific purposes (e.g., address standardization) • Absence of defined ownership or stewardship models • Recognition of need for oversight
Management	<ul style="list-style-type: none"> • Identity management by application when needed (e.g., customers) • Some application configuration, but not coordinated through centralized management
Identification	<ul style="list-style-type: none"> • Limited use of identity management by line of business • "Tiger team" attempts at customer data consolidation as required by applications (e.g., software upgrades or transitioning of accounting applications)
Integration	<ul style="list-style-type: none"> • Replicated copies of reference data • Limited data reuse • No application services reuse
Business process management	<ul style="list-style-type: none"> • Limited or no business involvement except at highest level of requirements definition

Reactive

At the **reactive** level, not only is there recognition that the existence of replicated copies of data causes, but there are some attempts are resolving the issue. Invalid or unusable data is deemed an Information technology problem. Data quality tools are purchased as a prelude to "fixing the data," although the actual business needs may lie unanalyzed while a technical team acquires tools. Initial uses of the tools satisfy some line-of-business application needs, but lessons learned are not shared, leading to duplication of effort.

Some attempts are made at consolidating metadata from across different applications, and tools are reviewed and purchased but still are managed as technical resources. Application needs for data sharing are attacked by vigorous and uncoordinated XML schemas and corresponding services, although there is a great need for fine tuning the variant implementations.

Reactive organizations understand the value of MDM and begin to build the infrastructure that can support MDM efforts.

Table 2 - Characteristics of the reactive level.

Component Layer	Capabilities
Architecture	<ul style="list-style-type: none"> Attempts to collect data dictionaries into a single repository Initial exploration into low-level application services Review of options for information sharing (e.g., enterprise information integration or enterprise application integration)
Governance	<ul style="list-style-type: none"> External applications used to manage metadata Introduction of data quality management for parsing, standardization and consolidation
Management	<ul style="list-style-type: none"> Resources are assigned to manage the use of introduced tool sets Training for enterprise roll-out of tools and technology make capabilities available on a more widespread basis Centralized administration of metadata and master indexes
Identification	<ul style="list-style-type: none"> Identity search and match used to reduce duplication Identity search and match used for rudimentary record linkage for householding purposes
Integration	<ul style="list-style-type: none"> Initial exploration of consolidation of data for newly-developed analytic (e.g., CRM) applications Data warehouse used as a core repository for master data No integration back into contributing applications
Business process management	<ul style="list-style-type: none"> Conceptual business process models are described Analytic application integration of consolidated data Initial use of business rules embedded within applications

Managed

Once analytic applications have been created that rely on some level of consolidation, individuals within the organization can establish a value proposition for continued use and growth of consolidated master repositories. Gaining senior management buy-in enables more comprehensive enterprise modeling activities, which are supplemented by the MDM program.

While at the reactive level the focus may have been on a single area such as customers, the **managed** level sees the ability to use master data become a repeatable process. The managed level also allows an enterprise to incorporate both new applications as well as existing applications, as the consolidation and synchronization services are available as part of the migration package.

For managed companies, initial efforts prove successful, creating a business case for increased use within the enterprise.

Table 3 - Characteristics of the managed level.

Component Layer	Capabilities
Architecture	<ul style="list-style-type: none"> Defined core data model for persistence Fundamental architecture for shared master repository Identified operational framework for low-level master data lifecycle activities Defined services for integration with master repository
Governance	<ul style="list-style-type: none"> Data quality tools in place Policies and procedures for data quality management Data quality issues tracking Data standards processes in place Line of business data stewardship
Management	<ul style="list-style-type: none"> Identity management centralized in master index Identity management utilized across numerous applications Identified hierarchies (households, relationships within a data class) used by analytic applications Advanced configuration and administration of application use of master data A migration plan is available for selected applications
Identification	<ul style="list-style-type: none"> Identity search and match services available to applications Record linkage integrated within the MDM service layer Rules for merging and consolidation standardized and managed under centralized control Merging and consolidation processes established and repeatable
Integration	<ul style="list-style-type: none"> Component services available for application integration Services synchronize applications with the repository
Business process management	<ul style="list-style-type: none"> Integration of business rules with master data operations Fundamental connectivity between business applications and core data objects Business process analysts participate in master data engineering requirements

Proactive

As managed organizations establish data models and service architectures, they become more adept at reducing individual application dependence on replicated data. At this level, applications are integrated through the service layer with the master repository. Data synchronizations are embedded within the component service layer, as are identity resolution, hierarchy management, and identity management.

A **proactive** business is able to better establish relationships at the customer, supplier and vendor level, as full profiles based on aggregated and consolidated data is managed as a core enterprise resource. Data governance is in effect across the organization with hierarchical organization down the management chain.

Proactive organizations realize better relationships with customers and suppliers due to a more responsive information framework.

Table 4 - Characteristics of the proactive level.

Component Layer	Capabilities
Architecture	<ul style="list-style-type: none"> • Master models are established • Capability to move from index framework to transaction-based MDM framework • Services-oriented architecture (SOA) in place for application architecture • Centralized management of business metadata
Governance	<ul style="list-style-type: none"> • Enterprise data governance program in place • Enterprise data standards and metadata management in place • Proactive monitoring for data quality control feeds into governance program
Management	<ul style="list-style-type: none"> • Identity management fully integrated across the enterprise • Unique identification of all master object instances • Full-cycle hierarchy management supports both analytic and operational activities • Hierarchy management enables roll-back of false positive consolidation errors
Identification	<ul style="list-style-type: none"> • Services for data life cycle embed identity search, match, and resolution • All data life cycle operations structured on top of merging and consolidation services • Consolidation occurs in background
Integration	<ul style="list-style-type: none"> • Synchronization completely embedded within life cycle services • Component layer supports application integration at master object level • SOA drives business application integration
Business process management	<ul style="list-style-type: none"> • Business logic is reused • Business rules are integrated within a rules engine and made available at the business process level • Business analysts integral to application development • Personalized customer relationships • Automated business processes

Strategic Performance

MDM, coupled with a services-oriented architecture, will ultimately enable rapid development of high-quality applications that support both the operational and analytic requirements of enterprise business applications. Business analysts work closely to enumerate expectations for outward-facing process implementations. Analytic results associated with business intelligence processes will be managed as master objects, enabling more effective and consistent predictive analytics to be embedded within customer-facing applications.

At the strategic performance level, applications - both operational and analytic - are fueled by a single, comprehensive MDM data repository.

Table 5 - Characteristics of the strategic performance level.

Component Layer	Capabilities
Architecture	<ul style="list-style-type: none"> • Complete transaction integration available to internal applications • Published API interfaces enable straight-through processing involving master data repository
Governance	<ul style="list-style-type: none"> • Cross-organization data governance assures high quality information sharing
Management	<ul style="list-style-type: none"> • Seamless identity management of all data objects synchronized to both internal and external representations • Migration of legacy applications complete
Identification	<ul style="list-style-type: none"> • Identity resolution services exposed externally to the organization • Business performance directly tied to master dimensions
Integration	<ul style="list-style-type: none"> • All application development is driven by business process models and their interaction with core master object models
Business process management	<ul style="list-style-type: none"> • Businesses completely drive application design and development • Applications largely integrate business rule engines • Data instance profiles (customer or vendor profiles) managed within master repository • MDM enables embedded predictive analytics

Summary

The transition to MDM is viewed as a revolution, but it is more effectively developed as an *evolution*. We have looked at the different components necessary to implement a mature master data management program, as well as investigated levels of maturity through which organizations may grow. And while no functionality list completely captures the inventory of services that a specific business requires from its master repository, by exploring the core MDM capabilities and a conceptual outline of technical MDM components, we have provided a framework to determine where any organization's capabilities lie.

When faced with the opportunity to assemble an MDM program, one should evaluate the business requirements and then review how those requirements can be addressed at the different levels of the maturity model. The presentation of the MDM component layers in terms of their maturity enables enterprise architects to target a desired level of MDM maturity. With that initial assessment, they can then develop a design and implementation roadmap that articulates the steps to take when assembling a program that effectively meets the business needs.