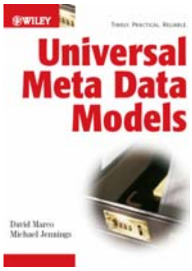
Managed Meta Data Environment (MME): A Complete Walkthrough

By David Marco

*This article is adapted from the book "**Universal Meta Data Models**" by David Marco & Michael Jennings, John Wiley & Sons*



WILEY

Universal Meta Data Models

David Marco
Michael Jennings

Almost every corporation and government agency has already built, is in the process of building, or is looking to build a Managed Meta Data Environment (MME). Many organizations, however, are making fundamental mistakes. An enterprise may build many meta data repositories, or "islands of meta data" that are not linked together, and as a result do not provide as much value (see "Where's my meta data architecture?" sidebar).

Let's take a quick meta data management quiz. What is the most common form of meta data architecture? It is likely that most of you will answer, "centralized"; but the *real* answer is "bad architecture". Most meta data repository architectures are built the same way data warehouse architectures were built: badly. The data warehouse architecture issue resulted in many Global 2000 companies rebuilding their data warehousing applications, sometimes from the ground up. Many of the meta data repositories being built or already in use need to be completely rebuilt.

WHERE'S MY META DATA ARCHITECTURE?

At EWSolutions one of our clients is a large pharmaceutical company. Since knowledge is the lifeblood of any pharmaceutical company, these types of firms tend to have very large meta data requirements and staffs. This company had decided to have a "Meta Data Day" and as such, they had asked me to come on-site and give a keynote address to kick the day off. Between 60-80 people attended "Meta Data Day".

After the keynote address were a series of workshops. We counted 4 separate meta data repositories in production and 3 other *separate* new meta data repository initiatives starting up -- a classic "islands of meta data" problem. This is not an approach that leads to long-term positive results. None of these islands are linked to each other and much of the most valuable meta data functionality will come from the relationships that the meta data has with itself. For example, it is highly valuable to view a physical column name (technical meta data) and then drill-across to the business definition (business meta data) of that physical column name.

MME Overview

The managed meta data environment represents the architectural components, people and processes that are required to properly and systematically gather, retain and disseminate meta data throughout the enterprise. The MME encapsulates the concepts of meta data repositories, catalogs, data dictionaries and any other term that people have thrown out to refer to the systematic management of meta data. Some people mistakenly describe an MME as a data warehouse for meta data. In actuality, a MME is an operational system and as such is architected in a vastly different manner than a data warehouse.

Companies that are looking to truly and efficiently manage meta data from an enterprise perspective need to have a fully functional MME. It is important to note that a company should not try to store all of their meta data in a MME, just as the company would not try to store all of their data in a data warehouse. Without the MME's components, it is very difficult to be effective managing meta data in a large organization. The six components of the MME, shown in Figure 1, are:

- Meta data sourcing layer
- Meta data integration layer
- Meta data repository
- Meta data management layer
- Meta data marts
- Meta data delivery layer

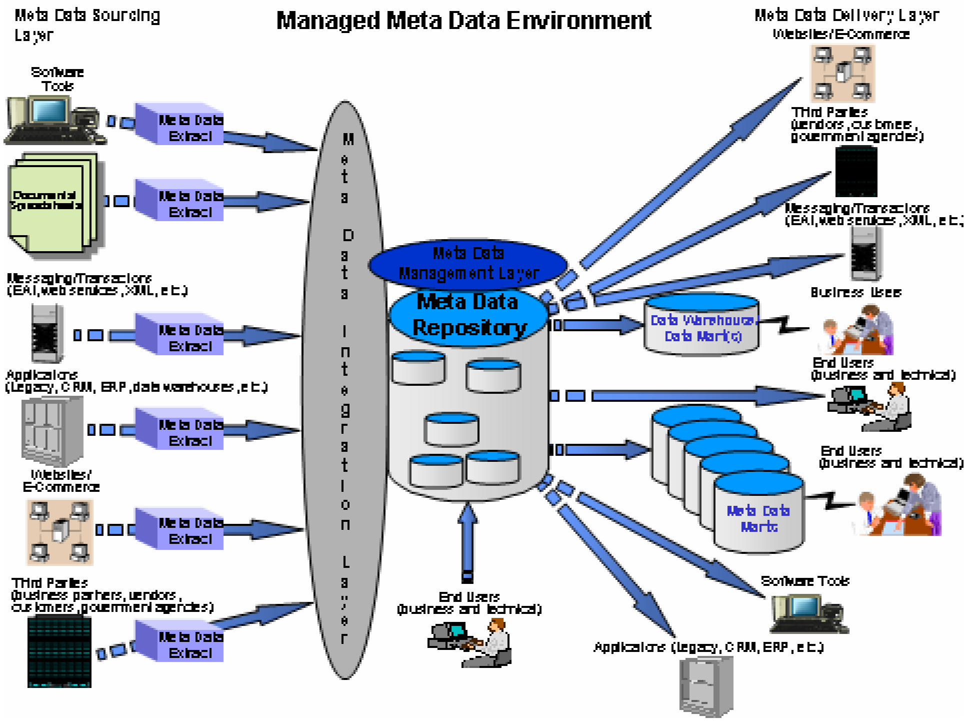

Meta Data Sourcing Layer
Managed Meta Data Environment
Meta Data Delivery Layer
Software Tools
Meta Data Extract
Documents/Spreadsheets
Meta Data Extract
Messaging/Transactions (EAI, web services, XML, etc.)
Meta Data Extract
Applications (Legacy, CRM, ERP, data warehouses, etc.)
Meta Data Extract
Websites/ E-Commerce
Meta Data Extract
Third Parties (business partners, vendors, customers, government agencies)
Meta Data Extract
Meta Data Integration Layer
Meta Data Management Layer
Meta Data Repository
End Users (business and technical)
Websites/E-Commerce
Third Parties (vendors, customers, government agencies)
Messaging/Transactions (EAI, web services, XML, etc.)
Business Users
Data Warehouse, Data Mart(s)
End Users (business and technical)
End Users (business and technical)
Meta Data Marts
Software Tools
Applications (legacy, CRM, ERP, etc.)

Figure 1: Managed Meta Data Environment

MME can be used in either the centralized, decentralized or distributed architecture approaches: *Centralized architecture* offers a single, uniform, and consistent meta model that mandates the schema for defining and organizing the various meta data stored in a global meta data repository. This allows for a consolidated approach to administering and sharing meta data across the enterprise. *Decentralized architecture* creates a uniform and consistent meta model that mandates the schema for defining and organizing a global subset meta data to be stored in a global meta data repository *and* in the designated shared meta data elements that appear in local meta data repositories. All meta data that is shared and re-used among the various local repositories must first go through the global repository, but sharing and access to the local meta data are independent of the global repository. *Distributed architecture* includes several disjointed and autonomous meta data repositories that have their own meta models to dictate their internal meta data content and organization with each repository solely responsible for the sharing and administration of its meta data. The global meta data repository will not hold meta data that appears in the local repositories, instead it will have pointers to the meta data in the local repositories and meta data on how to access it.[1] At EWSolutions we have built MMEs that use each of these three architectural approaches and some implementations use combinations of these techniques in one MME.

[1] See Chapter 7 of "Building and Managing the Meta Data Repository" (David Marco, Wiley 2000) for a more detailed walkthrough of these approaches.

Meta Data Sourcing Layer

The meta data sourcing layer is the first component of the MME architecture. The purpose of the Meta Data Sourcing Layer is to extract meta data from its source and to send it into the Meta Data Integration Layer or directly into the meta data repository (see Figure 1). Some meta data will be accessed by the MME through the use of pointers (distributed) that will present the meta data to the end user at the time that it is requested. The pointers are managed by the Meta Data Sourcing Layer and stored in the Meta Data Repository.

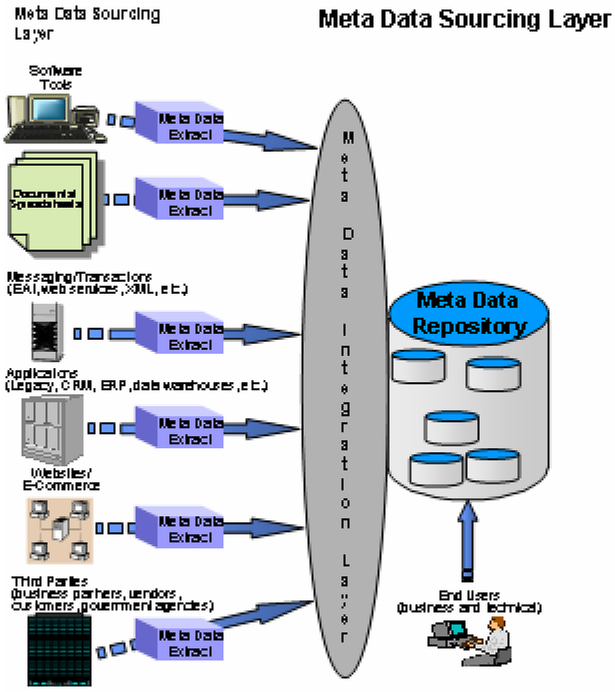

Meta Data Sourcing Layer
Meta Data Sourcing Layer
Software Tools
Documents/ Spreadsheets
Messaging/Transactions (EAI, web services, XML, etc.)
Applications (Legacy, CRM, ERP, data warehouses, etc.)
Websites/ E-Commerce
Third Parties (business partners, vendors, customers, government agencies)
Meta Data Extract
Meta Data Integration Layer
Meta Data Repository
End Users (business and technical)

Figure 2: Meta Data Sourcing Layer

It is best to send the extracted meta data to the same hardware location as the Meta Data Repository. Often meta data architects incorrectly build meta data integration processes on the platform that the meta data is sourced from (other than record selection, which is acceptable). This merging of the meta data sourcing layer with the meta data integration layer is a common mistake that causes a whole host of issues.

As sources of meta data are changed and added (and they will), the meta data integration process is negatively impacted. When the meta data sourcing layer is separated from the Meta Data Integration Layer only the meta data sourcing layer if impacted by this type of change. By keeping all of the meta data together on the target platform the meta data architect can adapt the integration processes much more easily.

Keeping the extraction layer separate from the sourcing layer provides a tidy backup and restart point. Meta data loading errors typically happen in the meta data transformation layer. Without the extraction layer, if an error occurred the architect would have to go back to the source of the meta data and re-read it. This can cause a number of problems. If the source of meta data has been updated it may become out of sync with some of the other sources of meta data that it integrates with. Also the meta data source may currently be in use and this processing could impact the performance of the meta data source. The golden rule of meta data extraction is:

Never have multiple processes extracting the same meta data from the same meta data source.

In these situations, the timeliness and consequently the accuracy of the meta data can be compromised. For example, suppose that you have built one meta data extraction process (Process #1) that reads physical attribute names from a modeling tool's tables to load a target entity in the meta model table that contains physical attribute names. You also built a second process (Process #2) to read and load attribute domain values. It is possible that the attribute table in the modeling tool has been changed between the running of Process #1 and Process #2. This situation would cause the meta data to be out-of-sync.

This situation can also cause unnecessary delays in the loading of the meta data with meta data sources that have limited availability/batch windows. For example, if you were reading database logs from your enterprise resource planning (ERP) system you would not want to run multiple extraction processes on these logs since they most likely have a limited amount of available batch window. While this situation doesn't happen often, there is no reason to build in unnecessary flaws into your meta data architecture.

The number and variety of meta data sources will vary greatly based on the business requirements of your MME. Though there are sources of meta data that many companies commonly source, I've never seen two meta data repositories that have exactly the same meta data sources (have you every seen two data warehouses with exactly the same source information?), but following are the most common meta data sources:

- Software tools
- End users
- Documents and spreadsheets
- Messaging and transactions
- Applications
- Web sites and E-commerce
- Third parties

Meta Data Integration Layer

The meta data integration layer (Figure 3) takes the various sources of meta data, integrates them, and load it into the meta data repository. This approach differs slightly from the common techniques used to load data into a data warehouse, as the data warehouse clearly separates the transformation (what we call integration) process from the load process. In a MME these steps are combined because, unlike a data warehouse, the volume of meta data is not nearly that of data warehousing data. As a general rule the MMEs holds between 5-20 gigabytes of meta data; however, as MME's are looking to target data audit related meta data then storage can grow into the 20-75 gigabyte range and over the next few years you will see some MME's reach the terabyte range.

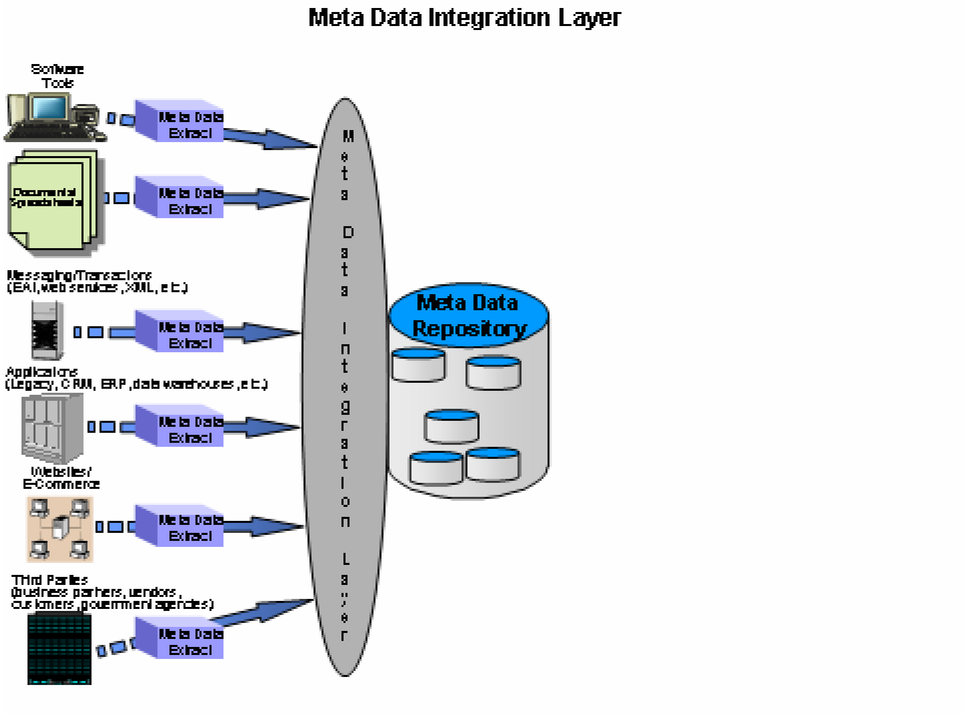

Meta Data Integration Layer
Software Tools
Documents/Spreadsheets
Messaging/Transactions (EAI, web services, XML, etc.)
Applications (Legacy, CRM, ERP, data warehouses, etc.)
Websites/ E-Commerce
Third Parties (business partners, vendors, customers, government agencies)
Meta Data Extract
Meta Data Extract
Meta Data Extract
Meta Data Extract
Meta Data Extract
Meta Data Extract
Meta Data Integration Layer
Meta Data Repository

Figure 3: Meta Data Integration Layer

The specific steps in this process depend on whether you are building a custom process or if you are using a meta data integration tool to assist your effort. If you decide to use a meta data integration tool, the specific tool selection can also greatly impact this process.

Meta Data Repository

A meta data repository is a fancy name for a database designed to gather, retain, and disseminate meta data. The meta data repository (Figure 4) is responsible for the cataloging and persistent physical storage of the meta data.

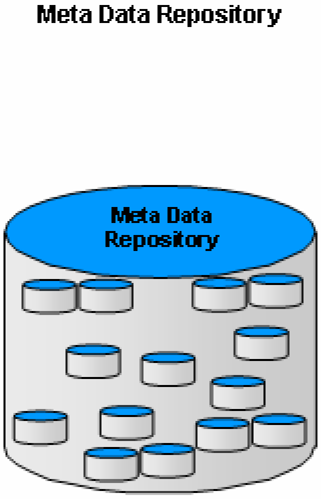


Figure 4: Meta Data Repository

The Meta Data Repository should be *generic*, *integrated*, *current* and *historical*. Generic means that the physical meta model looks to store meta data by meta data subject area as opposed to application-specific. For example, a generic meta model will have an attribute named "DATABASE_PHYS_NAME" that will hold the physical database names within the company. A meta model that is application-specific would name this same attribute "ORACLE_PHYS_NAME". The problem with application-specific meta models is that meta data subject areas change. To return to our example, today Oracle may be our company's database standard. Tomorrow we may switch the standard to SQL Server for cost or compatibility advantages. This situation would cause needless additional changes to the change to the physical meta model.[2]

A Meta Data Repository also provides an integrated view of the enterprise's major meta data subject areas. The repository should allow the user to view all entities within the company, and not just entities loaded in Oracle or entities that are just in the customer relationship management (CRM) applications.

Third, the Meta Data Repository contains current and future meta data, meaning that the meta data is periodically updated to reflect the current and future technical and business environment. Keep in mind that a Meta Data Repository is constantly being updated and it needs to be, in order to be truly valuable.

Lastly, meta data repositories are historical. A good repository will hold historical views of the meta data, even as it changes over time. This allows a corporation to understand how their business has changed over time. This is especially critical if the MME is supporting an application that contains historical data, like a data warehouse or a CRM application. For example, if the business meta data definition for "customer" is "anyone that has purchased a product from our company within one of our stores or through our catalog". A year later a new distribution channel is added to the strategy. The company constructs a Web site to allow customers to order our products. At that point in time, the business meta data definition

[2] See Chapters 4 - 8 of "Universal Meta Data Models" (David Marco & Michael Jennings, Wiley 2004) to see various physical meta models

for customer would be modified to "anyone that has purchased a product from our company within one of our stores, through our mail order catalog or through the web". A good Meta Data Repository stores both of these definitions because they both have validity, depending on what data you are analyzing (and the age of that data). Lastly, it is strongly recommended that you implement your Meta Data Repository component on an open, relational database platform, as opposed, to a proprietary database engine.

Meta Data Management Layer

The Meta Data Management Layer provides systematic management of the Meta Data Repository and the other MME components (see Figure 5). As with other layers, the approach to this component greatly differs whether a meta data integration tool is used or if the entire MME is custom built. If an enterprise meta data integration tool is used for the construction of the MME, than a meta data management interface is most likely built within the product. This is almost never the case; however, if it is not built in the product, than you would be doing a custom build. The Meta Data Management Layer performs the following functions:

- Archive
- Backup
- Database modifications
- Database tuning
- Environment management
- Job scheduling
- Load statistics
- Purging
- Query statistics
- Query and report generation
- Recovery
- Security processes
- Source mapping and movement
- User interface management
- Versioning

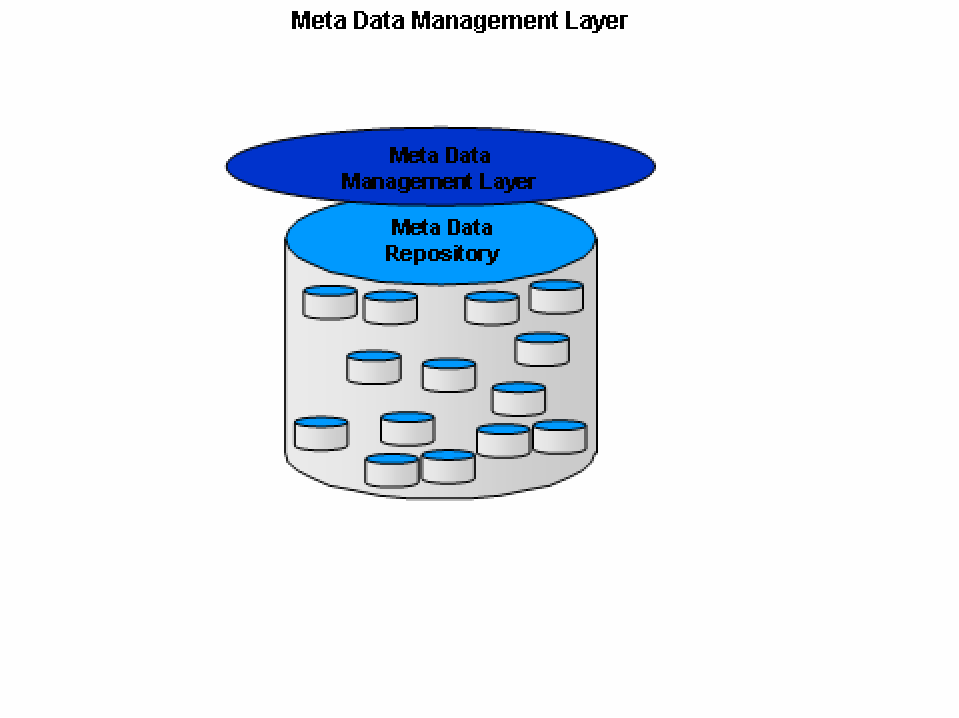


Figure 5: Meta Data Management Layer

Meta Data Marts

A Meta Data Mart is a database structure, usually sourced from a Meta Data Repository, designed for a homogenous meta data user group (see Figure 6). “Homogenous meta data user group” is a fancy term for a group of users with like needs.

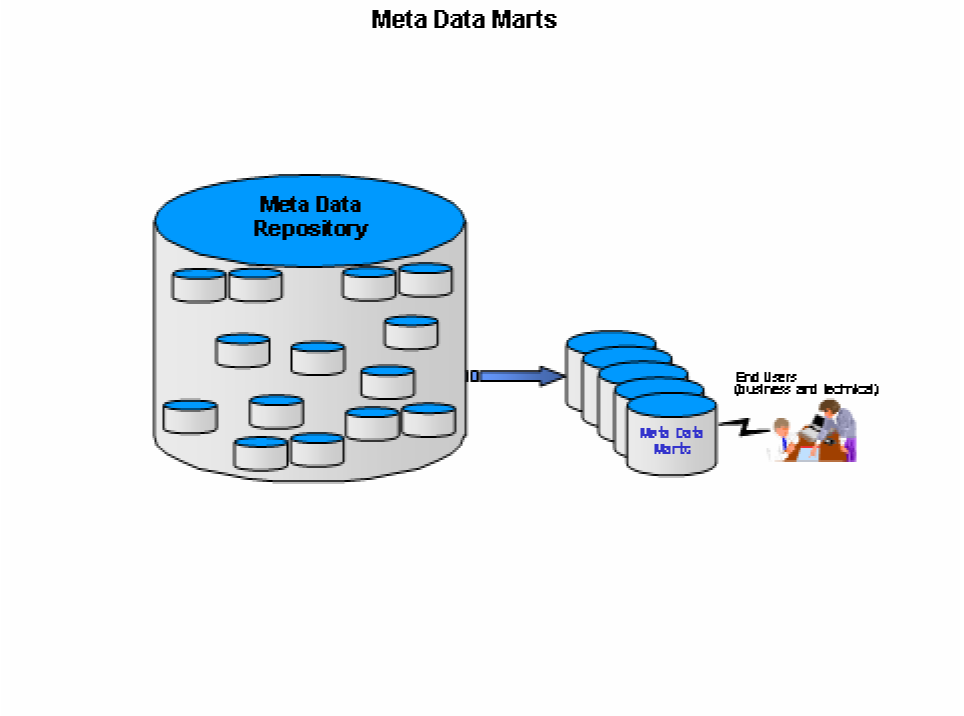


Figure 6: Meta Data Marts

There are two reasons why an MME may need to have meta data marts. First, a particular meta data user community may require meta data organized in a manner other than what is in the Meta Data Repository component. Second, an MME with a larger user base often experiences performance problems because of the number of table joins that are required for the meta data reports. In these situations it is best to create meta data mart(s) targeted specifically to meet those user’s needs. The Meta Data Marts will not experience the performance degradation because they will be modeled multi-dimensionally. In addition, a separate meta mart provides a buffer layer between the end users from the Meta Data Repository. This allows routine maintenance, upgrades, and backup and recovery to the repository without impacting the availability of the Meta Data Mart.

Meta Data Delivery Layer

The Meta Data Delivery Layer is the sixth and final component of the MME architecture. It delivers the meta data from the Meta Data Repository to the end users and any applications or tools that require meta data feeds to them (Figure 7).[3]

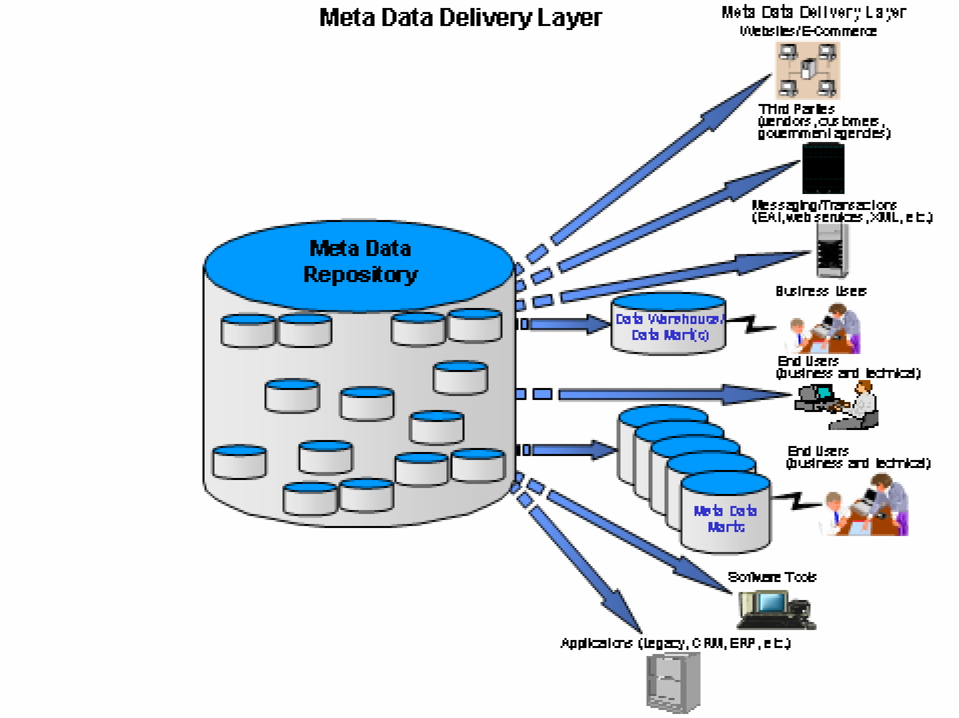

Meta Data Delivery Layer
Meta Data Delivery Layer
Websites/E-Commerce
Third Parties (vendors, customers, government agencies)
Messaging/Transactions (EAI, web services, XML, etc.)
Business Users
Meta Data Repository
Data Warehouse/ Data Mart(s)
End Users (business and technical)
End Users (business and technical)
Meta Data Marts
Software Tools
Applications (legacy, CRM, ERP, etc.)

Figure 7: Meta Data Delivery Layer

The most common targets that require meta data from the MME are:

- Applications
- Data warehouses and data marts
- End users (business and technical)
- Messaging and transactions
- Meta data marts
- Software tools
- Third parties
- Web sites and e-commerce

For professionals that have built an enterprise meta data repository they realize that it is so much more than just a database that holds meta data and pointers to meta data. Rather it is an entire environment. The purpose of the MME is to illustrate the major architecture components of that managed meta data environment.

Author Biography

*Mr. Marco is an internationally recognized expert in the fields of data warehousing, enterprise architecture and business intelligence, and is the world's foremost authority on meta data. He is the author of the widely acclaimed books **"Universal Meta Data Models"** (John Wiley & Sons, April 2004) and **"Building and Managing the Meta Data Repository"** (John Wiley & Sons). These groundbreaking books have been broadly endorsed by many of the largest software companies in the industry and by several major magazines. In addition, he is a coauthor of **"Impossible Data Warehouse Situations and Solutions***

[3] See Chapter 10 of "Building and Managing the Meta Data Repository" (David Marco, Wiley 2000) for a detailed discussion on meta data consumers and meta data delivery

From The Experts" *(Addison-Wesley) and* ***"Data Resource Management"*** *(DAMA). Mr. Marco has published hundreds of articles, is a regular columnist for several technology magazines and has served as a judge in dozens of industry awards. In addition, in 2004 Mr. Marco was selected to the very prestigious* ***Crain's Chicago Business "Top 40 Under 40"****.*

Mr. Marco is a highly sought after speaker and has presented over 100 keynote addresses and courses at the major business, data warehousing, and meta data conferences throughout the world. Mr. Marco has taught at the ***University of Chicago*** *and* ***DePaul University****, and is on the* ***Advisory Council for DePaul University's College of Commerce****. In addition, he is the founder and President of EWSolutions, a GSA schedule and Chicago-headquartered strategic partner and systems integrator dedicated to providing companies and large government agencies with best-in-class, award-winning solutions using enterprise architecture, data warehousing, and managed meta data environment technologies (866) EWS-1100 or visit www.EWSolutions.com. He may be reached via email at DMarco@EWSolutions.com*