TATA CONSULTANCY SERVICES

Experience certainty. IT Services
Business Solutions
Outsourcing

TATA

Information Integration: Metadata Management Landscape

White Paper, April 2009

Kamlesh Mhashilkar, Jaideep Sarkar

Executive Summary

In any organization, the successful operation and use of Business Intelligence (BI) heavily depends on the effective management of metadata. A well-defined Metadata facilitates a roadmap for all the data in a BI system and enables effective administration, change control, and distribution of the data supporting the BI components.

The most important part of metadata implementation is the integration between various components of metadata stored in the systems utilized. An established Metadata Paradigm assists the metadata implementation to achieve the strategic benefits of the data integration for BI system, which can be extended to the Enterprise Information Integration program. In a few implementations, the metadata architecture and its components need to be designed and built. In such cases, identification and separation of these components needs to be done in order to provide a robust metadata repository. This document provides a basis for the metadata architecture and design.

The Metadata Model for a BI program has been illustrated in this document, which can form the baseline for the metadata architecture activities. This paper presents the art of canvassing the landscape of the Metadata Management in the Information Integration solution. A selective adoption and orchestration of concepts and strategies as outlined in this paper will lead to a successful and value-driven Metadata Management within an Enterprise.

Table of Contents

Introduction

Organizations grow and change. Operational systems that run the day-to-day business and that provide management information to run the business (Business Intelligence systems) have to change along with the organization. Along with these changes the data that is generated within and Organization also grows and changes.

BI systems typically 'touch' a huge portion of the data in the enterprise in one way or another. The successful operation and use of BI heavily depends on the effective management of Metadata, which is usually referred as 'data about the data'. Metadata provides a roadmap of all the data in a BI system and enables effective administration, change control, and distribution of the data supporting the BI components. An extensive Metadata Management guarantees a high quality of the BI information and provides sufficient flexibility to extend the scope of the BI system to new information requirements and sources.

Metadata implementation is just one part of Information Intergation and its most important part is the integration between various components of Metadata stored in the tools utilized. In a few implementations, the Metadata Management architecture and its components need to be designed and built. In such cases, identification and separation of these components need to be done in order to provide a robust Metadata repository.

This document identifies the main factors that need to be considered and provides guideline to be followed during Metadata Architecture design and implementation. This paper only forms one part of a complete suite of papers available on Information Integration.

What is Metadata?

Metadata, commonly known as 'data about the data', is the data which describes other data. The term 'data' can be interpreted in various ways. Let us see that with an example:

'102250Richard King' can have many interpretations and a few of them have been given below.

- 10:22:50 EST appointment with Richard King
- 1022 is Order# and 50 is Line Item# of a shipment delivery to Richard King
- 10,2250C Temperature of a QUASARS called Richard-King
- 102250 is Richard King's employee number in TCS

How does one know which is the right interpretation? For that, some more information about this data is required and that is what Metadata is. Now, let us consider the last interpretation. Some examples of the data about the data '102250Richard King' can be:

- The format of this data is Employee Number – Number(6), Employee Name – Varchar(30)
- If the first digit of the Employee Number <> 9, then the Employee is not a Business Associate
- The Employee with Employee Number 102250 has joined TCS on 01JAN1997
- The Employee with Employee Number 102250 has worked in BIPM Services

If we analyze these data about the data, we can conclude that the first 2 examples of 'data about the data' are the rules that set the context for the data '102250Richard King'. However, the last 2 examples are really not setting the context for this data but rather are the detailed data related to the master data kept in the record '102250Richard King'.

Hence it is essential that when we say metadata is "data about the data", we need to be sure that we are talking about the context of the data and not the related / detailed data about the data. Metadata is the data describing context, content and structure of data and their management through lifetime. In short, Metadata is '**the context of the data**'.

Metadata Management Landscape spans across three areas:

- Metadata Model
- Metadata Topology
- Metadata Management Methodology

Let us now drill down into each of these areas to understand the Metadata Management in detail.

Metadata Model

Metadata is an important component of BI architecture. In a BI environment, the Metadata implementation primarily facilitates the integration of various metadata components/repositories used by the databases and the data modeling, ETL and OLAP tools. Metadata includes business rules, data sources, summarization levels, data aliases, data transformation rules, technical configurations, data access rights, data usage and much more. A well-designed Metadata Model enables effective administration, change control, and distribution of this Metadata to allow a seamless and end-to-end traceability.

Let us now see what a Metadata Model is.

What is a Metadata Model?

Let us look at our example from the previous section. If “102250Richard King” is the data then the following is the Metadata:

- Employee Number Number(6) – this tells us that the (first) 6 characters are numeric and represent the Employee Number.
- Employee Name Varchar(30) – this tell us that the (next) 30 characters are of alphanumeric of variable length and represent the Employee Name.

This Metadata can be further abstracted to Meta-Metadata, which stands for context of Metadata. From our example, we can see that our Metadata is actually telling us about the Element Name (Employee Number) and the Datatype (Number (6)) of the data. This information, which describes the Metadata in further detail, is called the Meta-Metadata. This is the Data based terminology.

Let us take a different view of this. The Metadata described above are clearly the elements / attributes in a logical or physical data model. So, we can say that the Data Model Metadata is actually the Metadata. This is the Model based terminology. The Metadata can be further abstracted to meta-meta data. The Data Model is created using the instances of the objects called tables and also the data is classified in the form of columns, primary keys, foreign keys, data types etc. This is the Meta-meta data or Metadata Model. The Metadata Model, itself, can be abstracted to another level, where the information is described in terms of Subject, Predicate and Object, where the Subject relates to the Object through the Predicate. This representation is known as the Meta-Meta Model.

These levels of abstraction across the two terminologies have been represented in the table below.

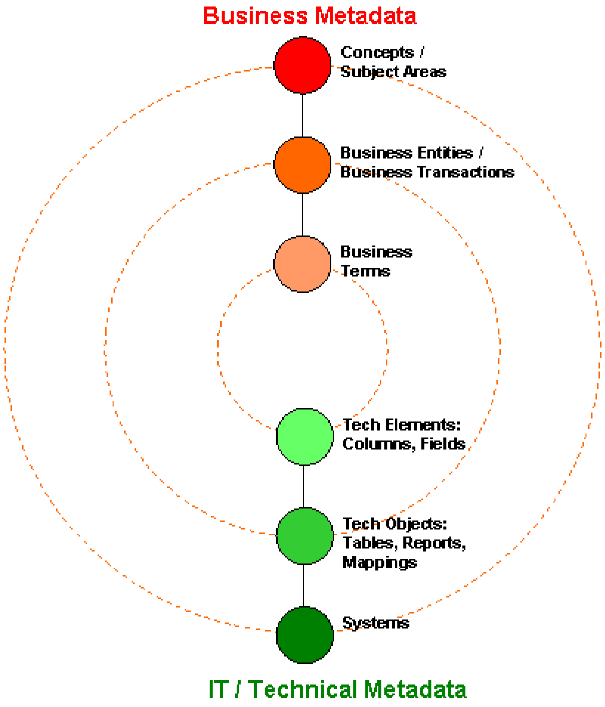
Example	‘Data’ Based Terminology	‘Model’ Based Terminology
102250Richard King	**Data**	
Employee Number Number(6), Employee Name Varchar(30)	**Metadata**	**Data Model**
Table, Columns, PK, Data Type	**Meta-Metadata**	**Metadata Model**
Subject, Predicate, Object		**Meta-Meta Model**

Hence, it is important that whenever Metadata is talked of, the level of abstraction is known. The Metadata or the Data Model tells us about the data and in order to understand the Metadata in detail, the Metadata Model should be known. Similarly, to know understand the Metadata Model the Meta–Meta Model should be understood. But in spite of different levels of abstraction, most of the times, we still refer to all these levels as metadata.

Let us now look at how we can model Metadata in an enterprise – the Enterprise Metadata Model, and subsequently, how we can enhance it to become our BI Metadata Model.

Enterprise Metadata Model

From an IT industry point of view, though in an enterprise, the Business and Technical (IT) areas are seen as separate, they cannot exist in isolation. The IT/Technical area is the backbone of the enterprise, which provides the foundation service and the necessary Applications / Tools to run and grow the Business, while there can be no IT/Technical area if there is no Business to run.


Business Metadata
Concepts / Subject Areas
Business Entities / Business Transactions
Business Terms
Tech Elements: Columns, Fields
Tech Objects: Tables, Reports, Mappings
Systems
IT / Technical Metadata

This One-to-One relationship is true for the Metadata paradigm, as well where the Business Metadata and the IT/Technical Metadata forms the basis of the Metadata model. The figure on the right side depicts these two limbs of Enterprise Metadata Model and the relations between the levels of abstractions.

The three levels of abstraction across these limbs have been explained below:

	Business Metadata	IT / Technical Metadata
High	Concepts / Subject Areas	Systems
Medium	Business Entities / Business Transactions	Technical Objects
Low	Business Terms	Technical Elements

High

Business Metadata can be represented as 'Subject Areas' or 'Concepts' at the highest level of abstraction. HR, CRM, billing and payment etc. are the examples of subject areas of a business which are defined at the time of gathering business requirements.

Corresponding to these business areas, technical systems are developed to meet the requirements of each subject area. Like Oracle HRMS can be developed for HR subject area and SIEBEL Systems may be implemented for CRM. These form the 'Systems' of the IT/Technical Metadata.

Medium

Each subject area can be further abstracted to Business Entities or Business Transactions. Customers, Vendors, Partners, Any Application Given by Customer and Business Transactions like Order Management etc. form the Business Entities of CRM. Corresponding to each Business Entity, there will be Technical Objects storing the details of these entities such as tables, reports, mappings etc.

Low

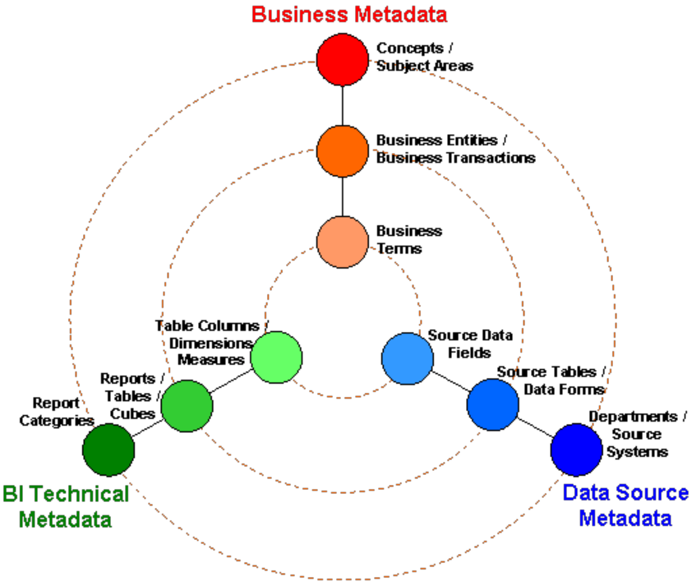
Business Terms form the lowest level of abstraction of Business Metadata. For Business Entities like Application, the Business Terms will be Customer ID, Customer Name and Product ID etc. The lowest level in IT/Technical group is Technical Terms. Any detailed information which exists at element level like columns, fields or any transformations etc. forms Technical Elements.

BI Metadata Model

In the BI space, the IT / Technical Metadata gets split into two categories, namely

- BI Technical Metadata and
- Data Source Metadata

In other words, the BI Metadata model has 3 limbs, as opposed to 2 in the Enterprise Model, The figure on the right side depicts the three limbs of BI Metadata model.



These limbs can be further abstracted to three levels as given below:

	Business Metadata	BI Technical Metadata	Data Source Metadata
High	Concepts / Subject Areas	Report Categories	Departments / Source Systems
Medium	Business Entities / Transactions	Reports / Tables / Cubes	Source Tables / Data Forms
Low	Business Terms	Columns / Dimensions / Measures	Source Data Fields

High (Area or concept level)

At the highest level, the subject areas of business can be directly applied to reports, analytics etc. of the BI Metadata and then can be mapped to the source systems of Data Source Metadata.

Medium (Entity level)

The business entities are connected to the technical entities like tables, cubes, and reports etc which obtain information directly from source tables or data forms available.

Low (Element level)

The most detailed level of Metadata exists at data element level. The business terms from business Metadata are mapped to the data fields in tables, reports and dimensions / measures in the multi-dimensional cubes which form the technical Metadata. The business users extensively use this Metadata information.

NOTE: The element level information from the three Metadata areas yields the "Glossary" for the Metadata implementation. This detailed Metadata information forms the base of the Metadata model with links to higher level of abstracts as well as other Metadata areas. This is the only zone through which the cross Metadata area search passes and hence it is important to design this zone for high performance search engines. Use of linked lists may aid in this case.

BI Technical Metadata

BI Technical Metadata contains all the Metadata, which relates to the different tiers within the BI environment and can be further split into three categories:

- Information Integration - ETL Metadata
- Information Storage - Data Warehouse Metadata
- Information Delivery - Reports Metadata

The terms ETL, DW and Reports metadata have been used only for simplicity / reference purpose. These should not be mistaken as then only components where metadata exists. For an instance, Information Integration Metadata can have components such as CDC, ETL, EAI and EII, but for simplicity we are calling it as ETL metadata.

At the three levels of abstraction, the BI technical Metadata can be classified as follows:

	ETL Metadata	Data Warehouse Metadata	Reports Metadata
Areas	ETL Categories	DW Schemas, MDDB	Report Categories
Entities	Extracts / Mappings	DW Tables / Cities	Reports / Dashboards / Adhoc Classes
Elements	Transforms / Parameters	DW Columns / Dimensions, Measures	Report Fields / Parameters

ETL Metadata

This category contains all the Metadata that relates to the Extraction, Transformation and Loading (ETL) of the data from the source systems into the BI environment.

At the highest level, the ETL jobs may belong to Categories which may be defined on the basis of the technology involved such as Oracle, Mainframe or Siebel or on the basis of the function of the source system such as Service Fulfillment / Assurance or Call Details. All the processes in a particular category will have some similarities. Metadata for these Categories such as source systems characteristics is captured at this level.

At the next level, the ETL Categories can be drilled down to individual ETL processes, which perform a particular task such as individual jobs, mappings etc. These processes typically are related to whole entities such as Customer Information, CDRs, Sales and hence they are named. Metadata about these Processes such as start time, dependencies etc. is captured at this level.

At the element level, when each element is to be loaded from source data fields into DW column, the Metadata about individual transformations such as normalization, de-normalization, aggregation rules and filter conditions etc. is captured at this level.

Data Warehouse Metadata

This category contains all the Metadata that relates to the storage tiers in a BI environment, which contain all the information extracted from the source systems.

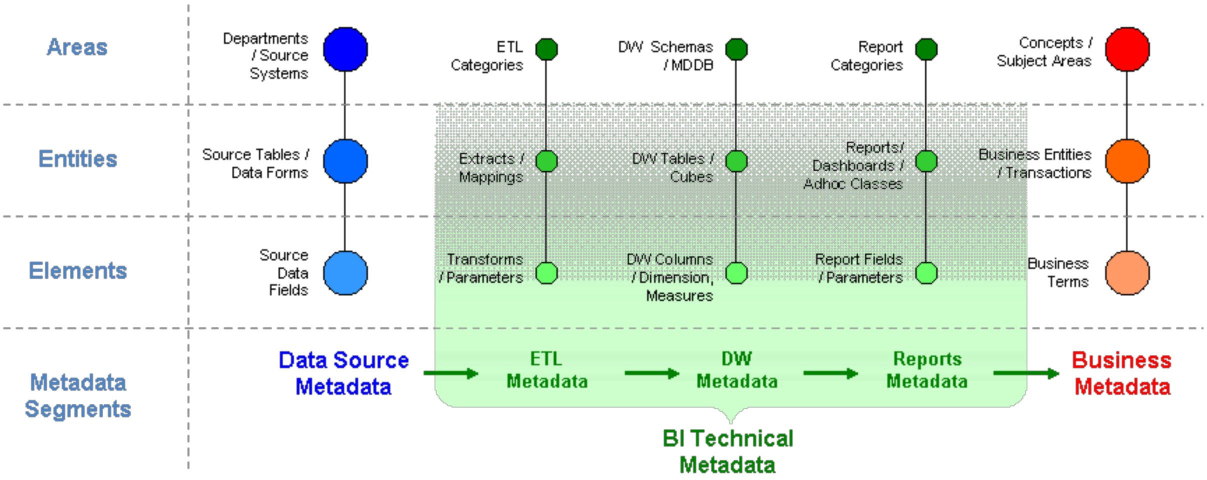
At the highest level, Metadata about the different relational DW Schemas and the Multi-Dimensional Databases (MDDBs) are captured. At the entity level, the metadata about the different objects such as tables, views, snapshots etc. in relational schemas or cubes in the MDDBs are captured. At the element level, metadata about the attributes such as columns of tables and views and dimensions/measures of the cubes are captured.

Report Metadata

This category contains all the Metadata that relates to the reporting and analytic tiers in a BI environment.

When DW is populated with Metadata, this data can be used for reporting categories by the end users at the highest level. As a part of end user layer, reporting categories can be created such as from business perspective there can be CRM universes, CRM framework etc and many reports can be generated and can be logically connected to DW schema or MDDB etc. At the medium level, there are dashboards, reports, various classes used in the universes. At the element or the lowest level, the Ad-hoc classes can have various folders for storing various reporting objects, fields, filters etc.

All these categories, discussed so far, can be mapped in a one to one manner. This has been depicted in the figure below.

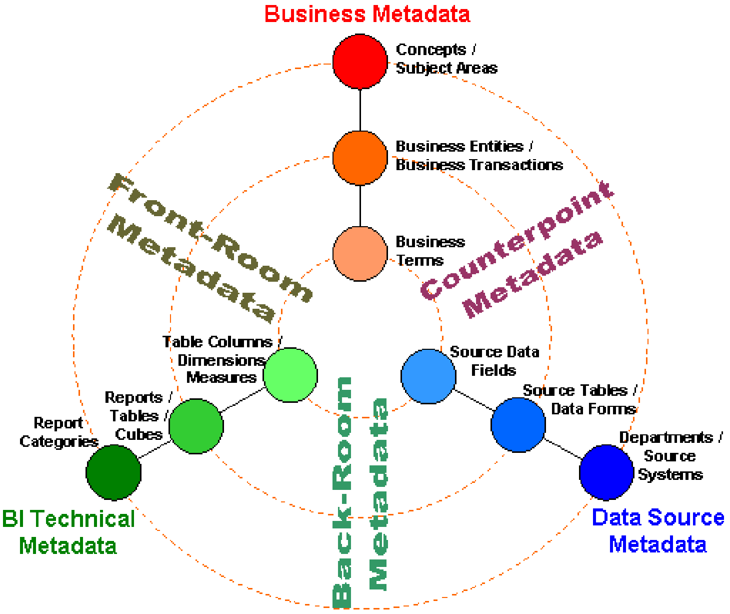


BI Metadata Implementation Domains

In a BI environment, the Metadata implementation takes place across the 3 principal limbs of the Metadata model, as explained in the previous section and depicted in the figure on the right. This implementation gets referenced as per the coverage of implementation. A few distinct implementation domains have been discussed in detail, subsequently.

- Back-Room Metadata
- Front-Room Metadata
- Counterpoint Metadata

The figure on the right depicts these three domains of Metadata implementation.



Back-Room Metadata

In a BI environment, data is extracted from data sources, put into DW and MDDB schemas and reports and dashboards are used to expose the data. This entire processing takes place at the back-end and the related Metadata is captured as Data Source and BI Technical Metadata, as described in previous section.

A few front-end processes too, work and store information such as ACLs, User Profiles and Scheduling, at the back-end to ensure that the front-end dissemination of information is done appropriately. Together, all this Metadata which is obtained from processes working and storing information at the back-end and those stored at the forms the Back Room Metadata. In other words, Back-Room Metadata encompasses the following components of Metadata and is primarily used by the Administrators, Supervisors, Developers and Designers:

- ETL Metadata (Control as well as Process Metadata)
- Data Models (primarily, data structures)
- Security Profiles (Roles and ACL's)
- Audit Trail (e.g. Data usage and actions)

These different types of Metadata are described in detail in the following sections.

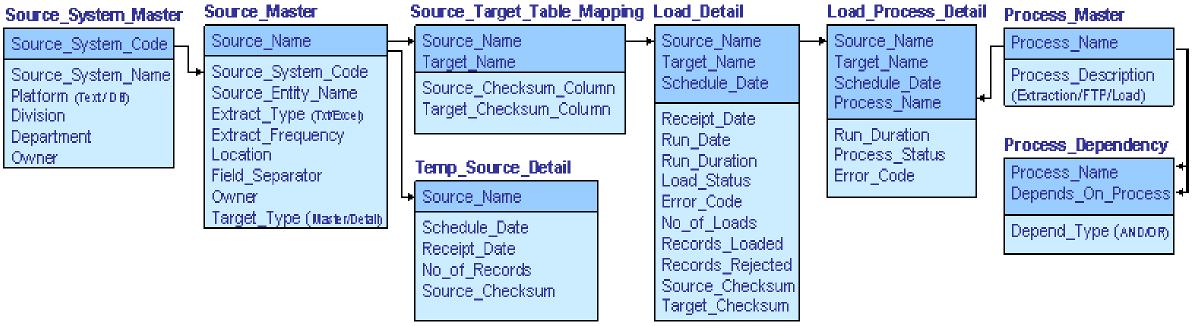
ETL Metadata

In ideal scenario, end-to-end data loading process should be Metadata driven. Otherwise a lot of manual intervention is needed to run the data load jobs and carry out the changes in the existing components. ETL Metadata principally gets divided into two further categories.

Inbound Integration – Metadata Management Landscape

- Control Metadata
- Process Metadata

Control Metadata: The Metadata deployed for controlling ETL processing is named as Control Metadata. The following diagram gives a basic control Metadata model, which can be enhanced as per the requirement.



Importance of Control Metadata:

- Business users can see the records loaded and records rejected in the DW and can hence decide the quality of data. To what extent they can rely on the data loaded is one of the important decisions taken by the business users with the help of control Metadata.
- Users can even create process dependency metrics to keep a track of entire ETL process flow and hence job execution can be automated and fully controlled.
- Control Metadata also aids in stating the data load status (i.e. data availability in BI system) to the end users. The history of input data errors and rejection also helps administrators to proactively enhance the system or suggest the source system changes.

Process Metadata: The Metadata that is used for processing purpose is called the Process Metadata. The complete ETL data transformation information is placed in this Metadata section. The following are the various processing activities, which are recorded in the Process Metadata.

Activity	Description
Parsing	All the TLV (Type, Length and Value) rules for structured data elements and Name-Value rules for semi-structured data elements (e.g. XML data elements) can be captured as metadata. And parsing programs / tools build parse trees using this metadata to separate the individual data elements.
Transformation	● **Data type conversion:** This involves lower-level transformations converting one data type or format to another. E.g. converting date, numeric, and character representations from one database to another. ● **Calculation and derivation:** These transformations need to apply the business rules identified during the requirements process, which would involve functions including string manipulation, date and time arithmetic, conditional statements, and basic mathematical functions. ● **Aggregation:** These involve summarization of low level data to the required granularity.

Activity	Description
	● **Special Transformations:** e.g. row to column conversions and column to row conversions
Cleansing	● Applying transformations (e.g. conversion, derivation, translation) for data standardization across different source systems. ● De-duping / match-merging is one of the important activities in the cleaning process, which is valuable for Customer centric applications. It aims at defining the business entity uniquely by removing its duplicate entries or by merging the multiple entries. e.g. duplicate customer records
Validation	Referential integrity (/lookups), constraint checks and also the reconciliation checks can be either implemented at the database level or during transformation process. These rules should be specified as a part of Metadata.

The ETL development could be in the form of specification through an ETL tool or development of custom built programs or mix of both these methods. The custom built scripts should use the Metadata rather than hard coding of validations so that maintenance is possible through Metadata. Also the Audit Trail (e.g. ETL process log) generated should display the Metadata references in order to improve the readability.

Importance of Process Metadata:

- The traceability / lineage of the source data to the BI environment is captured in the process metadata and hence it is easy for the developers / designers to perform impact analysis during change management.
- The easily available process metadata (and the lineage till front-room metadata) can help the business users in understanding the root of the data elements selected during decision making process. This helps in building the confidence of business users during decision making and also helps in identification of alternate data elements that can be used for better decision making.

Data Models

The data models are the backbone of the BI Technical Metadata. These models can encompass:

- DW schema and MDDB cube (Technical Metadata) and the corresponding Business Metadata
- Source system data structures and Metadata (Data Source Metadata)
- Mapping between source entities and DW entities (ETL Process Metadata)

NOTE: The availability of Data Source Metadata and ETL Process Metadata with the Data Model depends on the functionality offered by the data modeling tool e.g. ERWin. In many cases, the data models encompass only the DW schema and corresponding Business Metadata. The DW schema and corresponding Business Metadata can be fed to the RDBMS. E.g. tools like ERWin and Designer allow forward engineering to generate database structures and corresponding comments in the database.

This business Metadata may be imported into the Metadata repository of front-end tool using a few custom scripts. This can populate the front-end layer with the business Metadata which is an integral part of Front-Room Metadata.

The table on the right shows a basic data structure for storing the Metadata related to a data field in DW schema.

Also ETL processes (tool based or custom built) can be manually defined with the mapping information fed in the data modeling tools. The data source Metadata can be imported into the ETL repository, which can be utilized during the ETL development.

No.	Fields
1.	COLUMN_NAME
2.	DATA_TYPE
3.	LENGTH
4.	PRECISION
5.	DEFAULT_VALUE
6.	LOW_RANGE
7.	HIGH_RANGE
8.	UNIT_OF_MEASURE
9.	LEVEL
11.	BUSINESS_NAME
12.	BUSINESS_DESCRIPTION
13.	TABLE_NAME
14.	SUBJECT_AREA

Security Profiles

BI system administrators need to set and monitor the system security at various levels to ensure access restriction. The user profiles and their security policies are maintained at various regions, namely Back-End and Front-End Security Regions.

Tool administrators, developers and designers are part of the back-end security. They have privileges to modify the system and the data. Their area of operation is should be confined to the Level 100 security zone. It is recommended that they should not operate from the nodes other that this security zone. This information is maintained in Metadata of the database or Metadata repositories of various tools.

The end-users, who access the data, come under the Front-End Security Region. The administrators create these users and define their data access policies. ACL's are generated for this purpose and implemented in individual front-end tools of the system. These ACL's are stored in the Metadata repositories of the respective tools.

The following are the different levels / categories of users along with their region of control.

Security Region	Role	Region of Control
Back-End Security	Administrator	Access to Operating System, system tools
Back-End Security	Developer / Designer	Access to system tools (development environment)
Back-End Security	Supervisor	Access to system tools (production environment)
Front-End Security	Executive	Access to data from multiple departments
Front-End Security	Manager	Access to data from respective department
Front-End Security	Analyst	Access to data from respective division / business function in the Department
Front-End Security	Operator	Access to operational data (mostly from ODS).

Audit Trail

BI system usage Metadata indicating who is accessing which components of the DW, which reports are accessed at what frequency, what are the queries / ad-hoc reports requested, processing time for each report / query etc. should be recorded in the Metadata repository. The following table gives a generic audit trail table design.

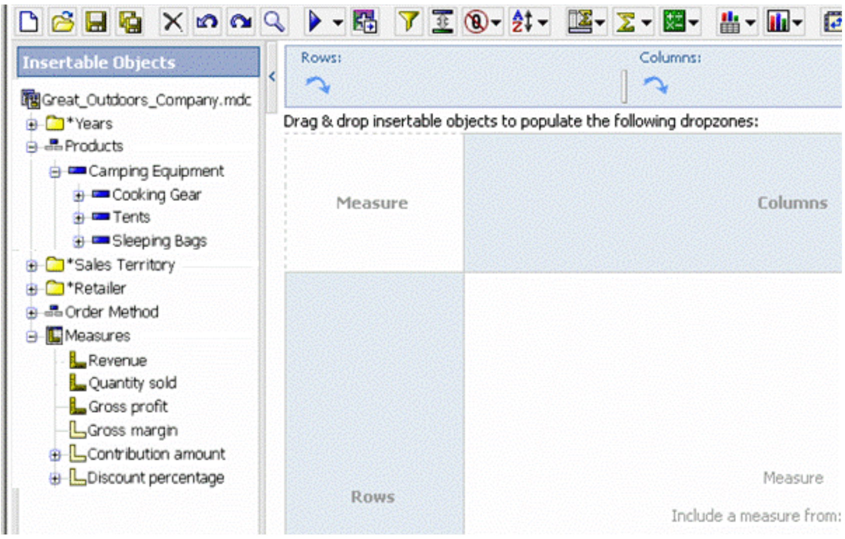
Field	Description
USER_ID (PK)	User identification
ACTION (PK)	Action specification e.g. Login, Logout, Open Report, Create Report, Save Report, Refresh Report, Publish Report to Corporate, Send Report to another User, Error etc.
TIMESTAMP (PK)	Time when the action was initiated
EXECUTION_TIME	Average execution time for the action. This can be NULL if the action was aborted.
MODULE	Details of the module for which the action is taken e.g. Report Name, Table Name, Subject Area, Department, Error Code etc.
USER_IP	The machine identification (IP Address) from which the user is accessing the system

Front-Room Metadata

Front-Room Metadata spans across the BI Technical Metadata and the Business Metadata, which is primarily used by the business users. Hence it needs to be developed in conjunction with business users. It has primarily 2 components as given below.

- **Standard Reports / Dashboards / Data Service Metadata:** This canned metadata is available to the business users in standard form. With the help of this metadata the users can select the appropriate (report / dashboard / data service) objects and view the necessary information during the decision making process.

- **Business Semantic Layer:** This metadata is typically derived from the Logical Data Model (LDM) of the data warehouse. This is presented to the business users so that they can drag and drop the elements of their choice and create the reports / queries. E.g. Universes in Business Objects, Models in Cognos Framework Manager. The figure shows an interface which gives the Business Semantic Layer on the left and report creation area on the right side. Once the report is created, the Layer, which is mapped to the DW elements, creates the SQL and fetches the necessary data from DW.



The unstructured data like user documents (e.g. manuals, glossary and business information documents, listing of special business events for justifying data trends) used in reporting and other front end processes is also captured in Front-Room Metadata.

Importance of Front-Room Metadata

Since the Front-Room metadata is directly available to the business users, it has high importance from the traceability perspective. It also contains a lot of internal (to be specific vertical) traceability scenarios. Consider the following examples:

- When the reports are created, lots of manipulations are done on the data as per the requirements of the reports like normalization, de-normalization, aggregation, encapsulation etc.
- In Business Objects, data can be obtained from several data providers and can be summarized in a report.

The data related to such manipulations which helps in tracing back the manipulations are captured in Front-Room Metadata and is highly useful in impact analysis or data lineage analysis.

Counterpoint Metadata

Counterpoint Metadata, as the name suggests, ensures the harmony in the BI system by establishing a trace for the handshake between Back-Room and Front-Room Metadata. Back-Room Metadata ensures that the data from various data source systems is moving into appropriate BI system elements. Further, Front-Room Metadata aids in presenting this data to the business users in their native language i.e. business terminology.

By the time the source data reaches the business users, it passes through a large number of processes and components and hence it is difficult to trace the data along this long path. In order to shorten this path without sacrificing quality and also ensure speedy design and impact analysis, Counterpoint Metadata is implemented. This spans across the Data Source Metadata and the Business Metadata giving traceability from source data to the business information (or vise versa) at various levels of abstraction

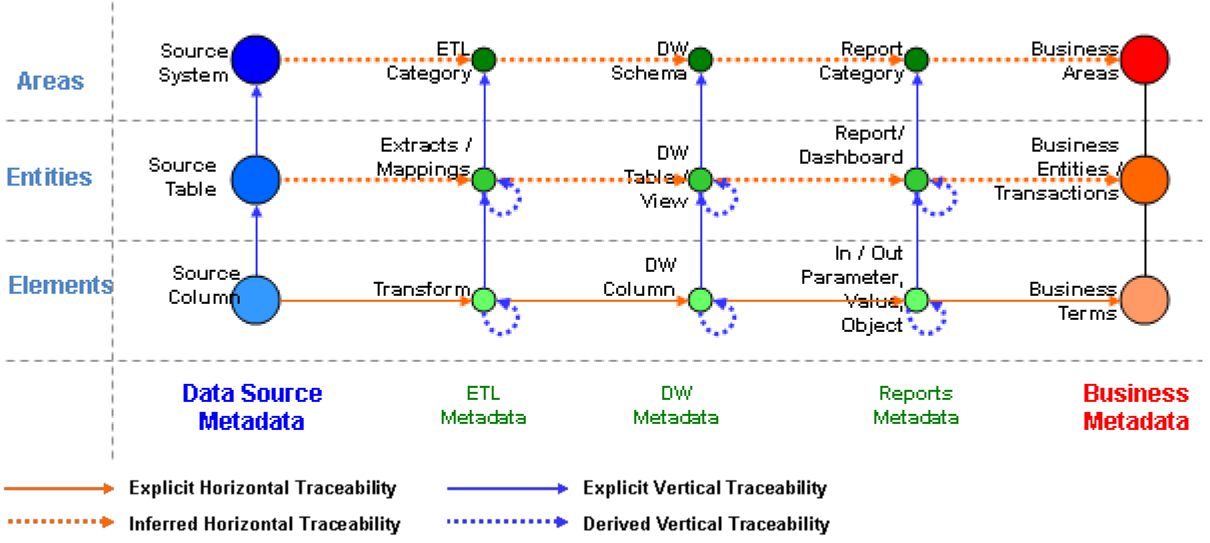
Counterpoint Metadata becomes very important in processes like requirement analysis, strategy developments, gap analysis etc. It balances the Front room Metadata and Back room Metadata effort. The effort of creating entire BI system is balanced with the help of Counter point Metadata.

Importance of Counterpoint Metadata

- In the development of strategy and requirement phase of any business, situations may come when only partial data is available say 60%-70%. Definitely, a BI system cannot be generated from such data, so to fill this gap of unavailable data, counter point Metadata is used.
- Impact Analysis: In maintenance, enhancement phase or in production support, impact analysis is often done. To gauge the impact of any change in the flow of entire process execution, like changes in business definition, changes in data source, addition of a new data source mapping to existing business functionality etc, counter point Metadata is used.
- Gap Analysis: This data gives the feasibility of bridging the gap between what is to be implemented from business perspective and what is available in IT /Technical System (which element can be populated from the data source and which element cannot i.e. there is a gap).

Horizontal and Vertical Traceability

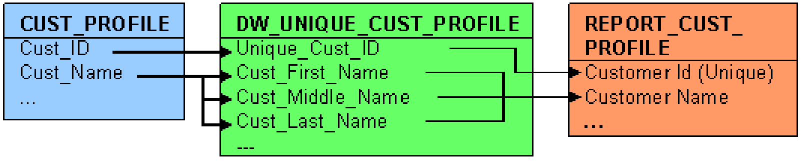
The following figure shows the horizontal and vertical traceability at all levels of a Metadata model:



Horizontal Traceability

At the lowest level of abstraction in the BI Metadata Model, a source column undergoes transformation and is then loaded into a DW column. This DW column is then exposed through reports, dashboards etc and associated with business terminology for effective exploitation by the business.

This can be better understood by an example. Suppose we have a source table named CUST_PROFILE, having customer details such as Customer ID (CUST_ID), Customer Name (CUST_NAME) etc. The columns of this table undergo some transformations before being loaded onto the DW. Using a CDI (customer data integration) system, a unique Customer ID (UNIQUE_CUST_ID) is generated and the column CUST_NAME is split into first name, middle name and last name. So, the DW will have the UNIQUE_CUST_ID, CUST_FIRST_NAME, CUST_MIDDLE_NAME, CUST_LAST_NAME and other columns with Customer details. These columns are input parameters to a report which generates customer profile. It is further linked to business terms i.e. unique identification number of the customer.

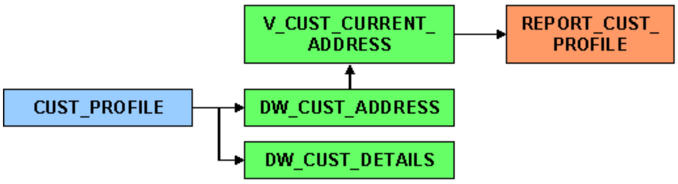


The end-to-end traceability at the element level that can be provided across the DW components i.e. the source system, ETL and finally the business terms is called **Explicit Horizontal Traceability**.

At the middle level of abstraction in Metadata model, let, there is a table called CUST_PROFILE in the data source. The transformations are done at the column level and the transformed columns are stored in table DW_UNIQUE_CUST_PROFILE in the DW. This table is used in the report named REPORT_CUST_PROFILE. This kind of horizontal traceability at the entity level is called **Inferred Horizontal Traceability**. To infer this kind of mapping, the element levels are used. E.g. the columns CUST_ID in source table CUST_PROFILE will undergo transformation to UNIQUE_CUST_ID into the DW table DW_UNIQUE_CUST_PROFILE.

At the highest level of abstraction, let there be a SIEBEL System in the source system which gets split to say SIEBEL1 and SIEBEL2 in the ETL categories, loaded onto various schemas and used in reports in various subject areas by the User. This level of traceability at the subject area level is also Inferred Horizontal Traceability.

Vertical Traceability



Continuing with the example from the previous section, suppose the CUST_PROFILE table gets loaded into two tables DW_CUST_DETAILS and DW_CUST_ADDRESS in the DW where the address is a Type 2 SCD (Slowly changing Dimension). Thus the history is maintained for the Customer's address. Now, a report on the customer profile is to be generated. For this, the two tables have to be linked and only current address is required. A view is created on these tables in the DW which filters the current address and gives all details of the customer along with current address. The required report is then generated from this view. In order to know that report generated is actually linked to DW_CUST_DETAILS and DW_CUST_ADDRESS tables in the DW, the traceability that is required is called Vertical Traceability. It can be at table, column, and even at report level when dashboards are created using many reports.

Similarly to Horizontal Traceability, Vertical Traceability too can be of two types. The traceability provided within the DW component itself is called **Explicit Vertical Traceability.** E.g. the traceability between the DW columns to DW tables to DW schemas. In contrast, suppose in ETL, a DW stored procedure uses 5 DW tables and say 3 of these tables are used as lookup tables and the other 2 tables as transaction tables. Aggregation is done on these transaction tables and the final data is loaded onto the DW. This is similar to Explicit Horizontal Traceability within a component itself, but the Horizontal traceability is lost because of this kind of derivation within the component itself. The traceability available in this situation is called **Derived Vertical Traceability**. Similarly, materialized views, views and trigger based data population are difficult to trace, unless this vertical traceability is maintained.

Metadata Management Topology

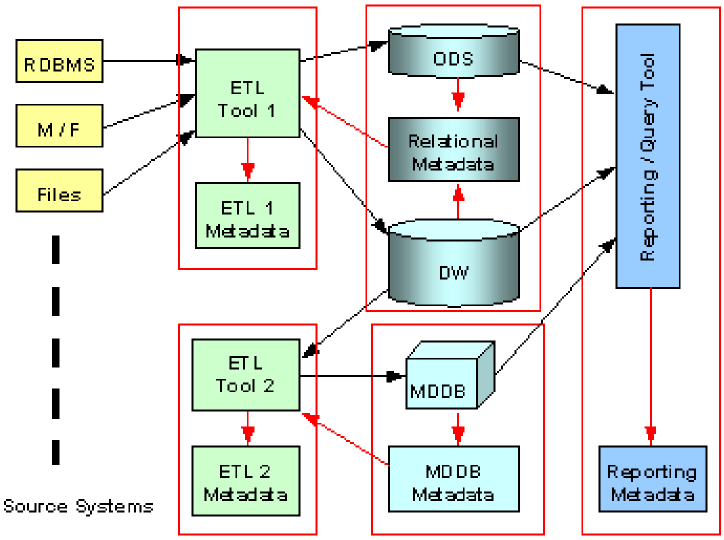
Topology, in the context of Metadata Management can be defined as the arrangement of its different logical and virtual architectural components of the solution. The Metadata Management Topology too can be classified into three types:

- Distributed
- Centralized
- Federated

Each of the above types presents different considerations. E.g. when thinking about a Centralized Architecture, the automation of object metadata linking and configuration management becomes important. When thinking of a Federated Architecture, the Functions, the Degree of Federation and the Technology Features become important.

Distributed Metadata Management

A Distributed Metadata Architecture results in a distributed metadata distribution mechanism, which is against the Data Warehousing principle of possessing 'a single version of truth at a centralized location'. This is seen as a major drawback. While metadata changes less frequently compared to the data, metadata updates are more complex to deal with. This is because metadata updates not only affect the data that is described (e.g. deletion / insertion of a data



element) but also related date due to metadata inter-relationships (e.g. referential integrity constraints). Additionally, the distributed metadata architecture requires an activity of synchronization of repositories, which share Metadata with each other. In particular, updates of replicated Metadata need to be detected and propagated in order to keep this metadata consistent. Subsequently, updated metadata needs to be applied (i.e. integrated) within a repository, e.g. to maintain the consistency of inter-relationships with metadata from other repositories.

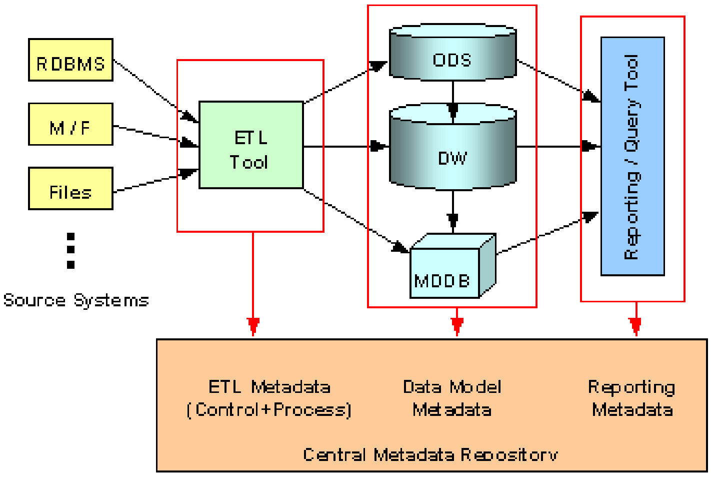
Why Distributed Metadata Management?
Consider the following example. An enterprise has multiple ETL and Reporting Tools in use but it does not have any specialized Metadata Management Tool in its one or many BI environments. Each ETL Tool such as Informatica, Ab Inito and DataStage, and Reporting Tool such as Cognos and Business Objects have their own repositories. In such architectures, it can be expected that the metadata is completely distributed, where the number of repositories is same as the number of tools in use. Having mentioned that, centralized metadata management gives a lot of benefits over the distributed one.

Centralized Metadata Management

A Centralized Metadata Architecture ensures,

- Standardized Metadata across different systems
- No replication of Metadata across systems and hence no need for synchronization of Metadata across the components used
- No need for maintaining bi-directional connections to be between various tools for Metadata exchange
- Minimal effort in system integration.
- Optimal hardware resource requirements.

Why Centralized Metadata Management?
To understand the necessity of Centralized Metadata Management, a question needs to be asked - How can the Metadata across the different BI tools be exchanged? The answer to this question may be very important, as can be understood if we take an example. Along with presenting data, it may be expected of the reporting tool to report on the reliability of the data that has been loaded onto the DW. However, without any exchange of metadata between the ETL and the Reporting Tool, this will not be possible. To resolve this kind of a problem, a centralized metadata management is required.



Centralized Metadata Warehouse v/s Centralized Metadata Management

Many a times, various BI tool vendors give a guarantee that they can manage Metadata inherently. E.g Informatica has its Metadata Manager and IBM has MetaStage. In actual practice, they give bridges to extract the information from RDBMS such as Oracle, MDDB such as Hyperion Essbase and reporting tools such as Business Objects or even from the Data Modeling Tools such as ERWin and then they load this extracted information into a Centralized Metadata Repository. However, this does not qualify as true Centralized Metadata Management because of the multiple repositories where Metadata is generated, propagated between and finally used from. This is simply the **Centralized Metadata Warehouse**.

It is expected that in the future vendors like IBM, Oracle, SAP will consolidate everything under a single platform. So if someone opts for oracle platform then he will have OWB or ODI as ETL Tool, Oracle database as RDBMS, Hyperion Essbase as MDDB, OBI EEE as the reporting tool, and all the Metadata repositories will be internally inbuilt into a single repository which will have a plug and play kind of mechanism, where an entire end-to-end Metadata Repository can be chosen. There will be no further need of any metadata linkage or replication as all linkages will be automatically maintained within the Centralized Metadata Warehouse, ensuring the **TRUE** definition of **Centralized Metadata Management.**

To qualify as a truly centralized architecture:

- There is only one self-maintaining repository
- There is no replication of data as there is only one repository.
- Metadata synchronization and Bidirectional Metadata is not required
- Hardware requirement is optimum
- The ability to perform impact analysis and provide end-to-end linkage is available

Distributed v/s Centralized Metadata Management

The aim of Metadata management is always to achieve the centralized Metadata architecture but due to the limitations of the tools and their functionality it may not be possible to achieve it in present time. Hence distributed Metadata architecture is seen in most of the implementations. Following table gives a comparison of the two architectures discussed so far.

Aspects	Centralized Architecture	Distributed Architecture
Number of repositories	Only one centralized repository is needed.	All the tools possess their own Metadata repositories.
Replication of Metadata	None.	Sometimes this it is necessary to replicate the Metadata across multiple tools e.g. user profiles.
Tool independence	The BI system architecture fully depends upon the tool chosen, as there exists only a single tool to take care of the entire BI system.	As multiple tools can be involved in this architecture, a set of tools can be chosen to get maximum functionality / facility coverage. But this is usually at the cost of seamless system integration.
System Integration	As only one tool or a set of tools from a single vendor is involved in this architecture, the system integration is	Integrating tools from various vendors is one of the greatest challenges in this architecture. The number of tool-to-tool connections / compatibility issues and the mapping overhead are

Aspects	Centralized Architecture	Distributed Architecture
	seamless and hence compatibility issues of various components do not arise.	significant. Usually a POC is recommended to ensure the compatibility among various tools before the implementation begins.
Metadata synchronization	No synchronization is necessary.	Synchronization of repositories sharing Metadata with each other needs to be accomplished. In particular, updates of replicated Metadata need to be detected and propagated automatically in order to keep this Metadata consistent.
Metadata exchange	No Metadata exchange is necessary.	All tools communicate with each other to exchange Metadata generating numerous bi-directional connections. But a few of the tools may not be able to communicate at all or may need tools to provide a channel to communicate with others.
Hardware capacity requirements	Optimum hardware is required.	Various tools demand different hardware capacities. Accumulated hardware requirement is always larger than what is needed for centralized Metadata architecture.
Example	Informatica Suite of Products.	Architecture using various products like Informatica PowerCenter, Business Objects, Hyperion Essbase, ASG Rochade etc.

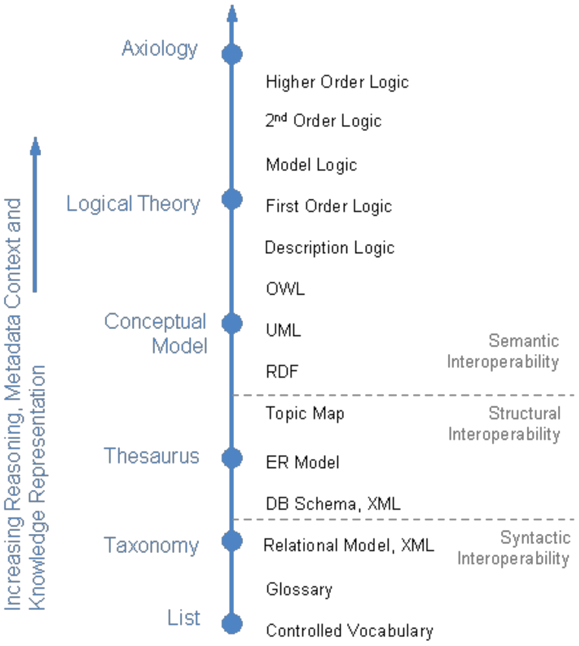
The above comparison actually shows us that a centralized architecture yields a more cost effective solution as compared to a distributed architecture. It can be said that the Centralized Metadata Architecture is the desired objective in most cases but due to the limitations of the tools and their functionality it may not be possible to achieve at present times. Hence, a distributed Metadata architecture is seen in most of the implementations.

Automation in Configuration Management

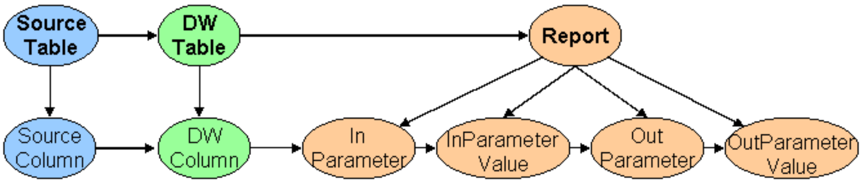
Configuration Management is important not only from the perspective of Centralized Metadata Management but also Distributed Metadata Management. The concept of configuration management in Metadata Management is actually quite simple. It is as simple as treating the Metadata as Type 2 SCDs. The metadata elements need to be associated with validity period (START_DATE and END_DATE) and if necessary, a STATUS_FLAG to maintain the versions. While constructing the end-to-end lineage along with the versions, these dates will need to be used to show how the lineage has changed over a period.

Automation in Metadata Linking

With reference to the figure, the degree of sophistication that is desired in a Metadata Context and Knowledge Representation has evolved over time. It started with providing **Syntactical Inter-Operability** of various elements like control vocabulary, glossary of terms with meaning etc. In taxonomy creation, the Relational Models and XML were used. As the next step in the evolution, a more **Structural Inter-Operability** was required where Database schemas, XSDs, Entity Relationship (ER) Models and Topic Maps were used. At present, the stress is on **Semantical Inter-Operability**, where RDL, UML, OWL, etc. are used. Domain Ontologies allow greater inferences and cognition. The further concepts are deemed to be more advanced than the scope of this document and are hence not described in further detail.



Semantics can be used to automate the element level metadata linking and also infer the relations at higher levels of abstraction such as entity and system. In this, the typical tuples can be formulated to construct the relations among the various metadata objects of interest. The following figure shows formation of such tuples. The linkages represented in terms of Subject-Predicate-Object at element level (mentioned as Horizontal Traceability) can be populated and then the auto inference building can be performed at Entity level using the Vertical Traceability set of tuples. OWL-DL (Web Ontological Language – Description Logic) can be used to program this. This mechanism can help only in providing the automation in metadata linking. If metadata versioning needs to be achieved along with the linking then it gets complicated and looses the efficiency. In such scenario, we need to fall back to Relational Tables for representing these tuples in terms of recursive relations and formulate the linkages. It does help in creating metadata versioning along with the linkages.



	Subject	Predicate	Object
Vertical Traceability	SourceTable	hasSourceColumn	SourceField
	DWTable	hasDWColumn	DWColumn
	Report	hasReportInParam	InParameter
	Report	hasReportOutParam	OutParameter
Horizontal Traceability	SourceField	OutRefSDW	DWColumn
	DWColumn	OutRefDWR	OutParameterValue
	InParameter	inParameterValue	InParameterValue
	InParameterValue	inParamValueOutParamValue	OutParameterValue
	OutParameter	outParameterValue	OutParameterValue

Federated Metadata Management

In Federated Metadata Management, a Central Metadata Repository is deployed at the EDW or Integration Layer (IL) level and local repositories are deployed at the Subject Area, Data Mart or Project level. In this architecture, even though there is more than one repository, common data elements are defined consistently even when they are stored in multiple systems/databases and data can be shared between applications. It ensures:

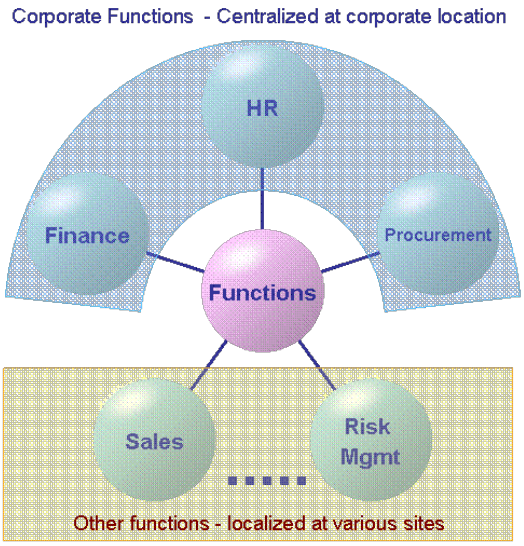
- Uniform representation of shared metadata across different systems
- Relative autonomy for local repositories
- Controlled replication of metadata across systems
- Reduced number of connections to be maintained between repositories (as compared to the Distributed Metadata Architecture) leading to reduced complexity of the system.

The definition of the different components within the Federated Metadata architecture can be defined on the basis of three different factors:

- Business Functions e. g. HR, Sales, Finance and Procurement
- Degree of Federation e. g. Hub and spoke, Distributed Federation, Centralized Federation, Multi -tier Federation
- Technology Features

Business Functions Based Federation

In an enterprise, there can be various business functions such as Finance, HR, Procurement, Sales, and Risk Management. The challenge in defining the federated metadata architecture lies in identifying Corporate Functions as opposed to the Localized ones. E.g. in TCS, Finance and HR are handled in a corporate manner and Sales and Customer Delivery are handled by various Verticals Units, which can be operated across various geographies, are localized functions.


Corporate Functions - Centralized at corporate location
HR
Finance
Procurement
Functions
Sales
Risk Mgmt
.....
Other functions - localized at various sites

Corporate Functions, typically, have to be centralized and these functions should be arranged in a Hub and Spoke manner and not as a Multi-tier Federation (described below). By centralization, it is meant that all the metadata from various geographies are moved into a single centralized repository. For Sales, Risk Management there can be individual Data Warehouses where analytics can be performed or Multi-tier Federation can be created as per the need. This gives autonomy to the local operations.

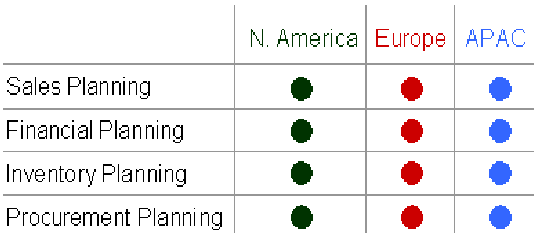
Degree Based Federation

There are primarily 3 different kinds of scenarios in federation based upon its degree.

- Distributed Federation
- Hub and Spoke Federation
- Multi-tier Federation

Distributed Federation

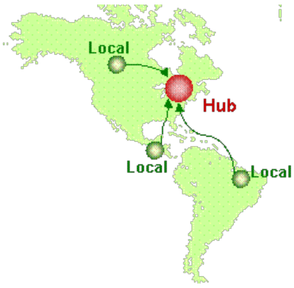
Distributed Federation uses Cartesian Product of Business Functions and Geographies for implementation. A company has implemented individual geography based DW solutions at various locations such as in the UK for entire Europe, in Australia for APAC countries and in the USA for North America. Now, consider the planning process in which they need to implement Sales Planning, Financial Planning, Inventory Planning and Procurement Planning, each of which require separate data marts and associated metadata solutions across all geographies. This can be seen as a Cartesian product between the Business Functions and the Geographies.



	N. America	Europe	APAC
Sales Planning	●	●	●
Financial Planning	●	●	●
Inventory Planning	●	●	●
Procurement Planning	●	●	●

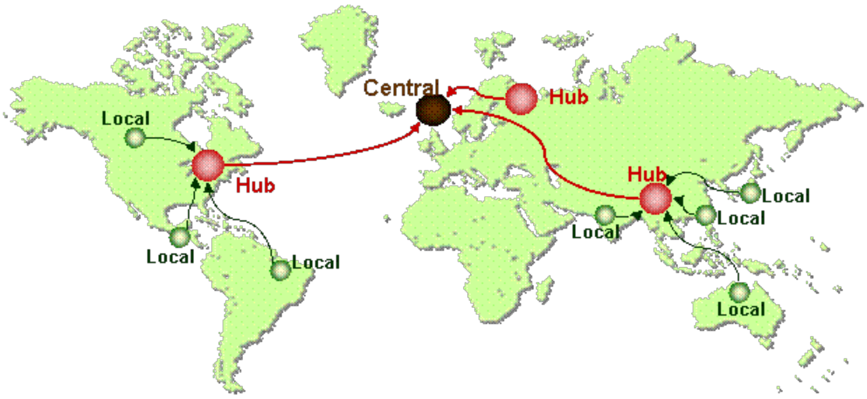
Hub and Spoke Federation

Within the Hub and Spoke implementations, Cartesian Products do not come into the picture. E.g. the data from various subsidiaries in an individual country will be consolidated for the country specific reporting and that will form the local instances. This data from various countries will be passed to the corporate for financial consolidation. Here the corporate acts as a hub. This is a typical scenario of Centralized / Corporate Data Warehouse.



Multi-tier Federation

Let's take an example to understand the Multi-tier Federation. The diagram given below represents the implementation of a DW solution for a multi-national company. There is a central DW at the corporate location in UK and Hubs have been placed in three different continents, namely North America, Asia Pacific (APAC) and Europe, to store the data from local countries.



The Corporate Functions (e.g. Finance and HR) have to be centralized in UK and hence the data has to be extracted from all the locations and should be loaded directly into a repository in the UK through the hubs. But centralization is not required for the Localized Functions (e.g. Sales). This may be because of different considerations such as that the information relating to individual Retail customer cannot be sent out of the European Landscape. Hence in the true federation or Third degree of federation, a lot of information is collected from the various local countries and consolidated into the

Hub and subsequently, only consolidated information is moved to the centralized location in UK. So, we have Metadata being managed locally, at Hubs and also centrally, which can be operated through various Geographies.

Federation usually leads to complexity in metadata. The typical issue is related to scenarios of Synonyms and Homonyms of various elements. E.g. there can be a definition for a measure called profit which can be defined in specific manner (assume it to be Profit after Tax) at a central location. The local centers and Hubs will have to send the data in compliance with the central definition. If the local analysts want to see profit as Profit before Tax they can have this definition, however, when profit is required by the central analysts, it must be Profit after Tax. Hence such homonyms need to be maintained in Metadata Management. Same is the case with Synonyms.

Technology Features Based Federation

It becomes difficult to choose the right technology for the requirements due to the availability as well as limitations of features in various Technologies. The different technology features, which may define the nature of federation, have been listed below.

- **Metadata Repositories**: A few tools provide the synchronization among metadata repositories to enable the federation. E.g. For example, in Informatica, the global and local repository configuration can be used to push the data from local into the Hub and from Hub to the Centralized location. With the help of this the processing, storage and information delivery can be distributed along the degree of federation without loosing coherency.
- **Metadata Bridges for Exchange:** Most of the tools provide Metadata Bridges to extract the metadata information from different tools (through API or directly) and centralize at one location (Centralized Metadata Warehouse). E.g. Informatica Metadata Manager, IBM Metadata Workbench. This enables the visibility into the metadata of various tools / components along the different degrees of federations. A few bridges are in the form of views on top of the metadata structures of the tools. This allows the real-time exchange of metadata without transferring the metadata. Hence the metadata exchange features through bridges can be categorized as
  - Exchange Mechanism (Metadata transfer v/s view)
  - Metadata Repository Coverage (Which tools and which versions can be connected with for the exchange)
- **Metadata Lineage:** A few platform specific tools are in the process of integrating the metadata repositories. This will facilitate the centralized metadata management and hence giving the end-to-end lineage. A few questions still arise.
  - Is the metadata lineage available horizontally as well as vertically?
  - Is the metadata lineage also going to display the versioning (/configuration management)?
  - Are the higher level inferences appropriate or erroneous?
- **Metadata Publishing / Services:** This is the other side of the Metadata Bridges i.e. the metadata APIs / services, which are invoked by the Bridges. The downward / upward compatibility of the APIs can enable the sustained handshake with the downstream applications seeking for metadata. In spite of changes in the tool across the tiers of federation, the metadata API calls can remain same. Also the functional overloading of such APIs for satisfying advanced requirements can be looked into.
- **Metadata Access / Reporting:** How Access Control List (ACL) can be created and what kind of flexibility is available in defining the ACL needs to be checked. Localized metadata access v/s

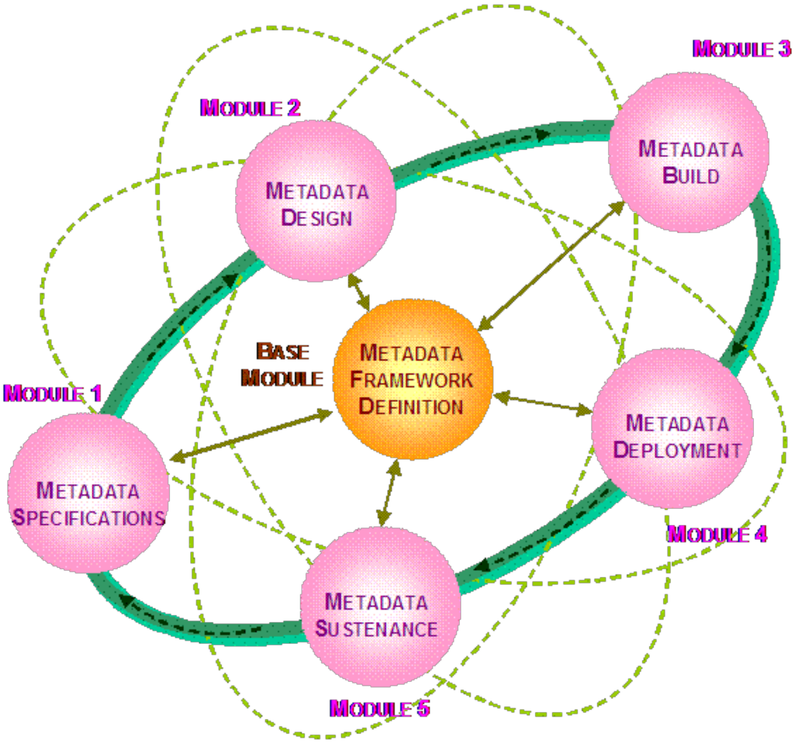
corporate metadata access configuration parameters and features can help in defining the tiers in the federated architecture. The availability of different reporting and search features on the metadata is important. The availability of standard reports for control metadata and data load statistics, end-to-end lineage and search (keyword based, content based, synonym/homonym based) capabilities on the metadata needs to be checked. The drillable metadata reports are very useful on top of the search features.

- **Metadata ACL:** Different features can be made available as a part of the metadata ACL. Consider the following scenarios.
  - It is advisable that individuals should have unrestricted access to the entire metadata. This enables them to understand that 'such an information' is available. The ACL can give access to the metadata of the information even if it is in the confidential region. If the user tries to seek that information (/ open the report), it will not open, but it indicates the availability of the report.
  - How homonyms can be handles by users across different functions? The users in the corporate have access to a metric (e.g. Profit) with a specific definition (Profit after Tax). However in the local area they want to create a report where profit actually means profit before tax not after tax. Can this kind of Duality be maintained, When reporting is being done and who is going to access this kind of Metadata ,if Corporate is going to access the Metadata they should actually see profit before tax instead of profit terminology over there which can contradict their definition of profit i.e. profit after tax.
- **Search Span:** i.e. levels of penetration. At which level the penetration can be done or only it will display the metadata at the element (columns, fields) level only. Is a drill-down from higher level of abstraction to the lower level (while displaying the end-to-end lineage) possible and vise versa?
- **Invoking other programs/services through Metadata Portal:** There can be new features available which can involve services like data services, report services or invoke program directly from Metadata portal. In the reporting tools, the portal provides keyword based search for the report. The new features will be able to allow metadata search on metrics / elements / concepts and give lineage to reports and provide links to directly invoke the reports / dashboards.

Such technical features may help the federated metadata architecture to satisfy the requirements or also put restriction a certain requirements, hence determining the degree of federation that can be achieved.

BIDS™ Metadata Management Methodology

A well-defined metadata management program guarantees a high quality of the information and provides sufficient flexibility to extend the scope of the BI system to new information requirements and sources. BIDS™ provides Metadata Management Methodology as one of its Process Solutions. The methodology consists of 6 modules as depicted in the figure below.



The following sections give an overview of the framework definition, specifications and design phases. The other phases resemble any other standard project. Further details are available as the BIDS™ Metadata Management Methodology document.

Framework Definition

Metadata management primarily aims at standardized and centralized metadata yielding flexible and robust metadata architecture. The Metadata Framework Definition involves analyzing the current state of metadata and metadata processes and developing a blueprint for Metadata Management Program. It starts with the goal, objectives and high level requirements for the metadata management which have been given below.

Goals

The overall goals of the Metadata Management Program may be described as follows.

- Standardization in metadata as well as data handling
- Centralization of metadata management
- Elimination of duplication of metadata information
- Transition adaptive metadata architecture

Objectives

The objectives of the Metadata Management Program may be described as follows.

- To develop metadata and data standards.
- To centralize the administration as well as usage of BI system.

- To improve data integrity and accuracy through non-redundant / non-duplicated metadata information.
- To reduce the effort in development, enhancement, implementation and maintenance of the BI system components.
- To establish a flexible metadata architecture to incorporate alterations in the BI architecture.

High Level Requirements

The high level requirements for generation and management of metadata can be perceived as given in the following table.

No.	Requirements
1.	**Metadata Standardization**
1.1	Unique terminology and standardized communication within the enterprise: The availability of a metadata as a unique source for users should bring various benefits. It should ensure a consistent vocabulary for users to communicate, understand, and interpret business information. It should eliminate ambiguity and guarantee consistency of information within the enterprise, and enable sharing of knowledge and experience.
1.2	Seamless system integration: ETL processes, especially integration, rely on metadata of the various data sources and BI system. The standardized metadata should aid in integration of data from various systems and give unique meaning to the data elements loaded in the BI system. Furthermore, the integration of different applications as well as tools is only possible if their metadata is shared through a standard mechanism.
1.3	Data quality improvement: Standard quality assurance rules should be defined. This forms an integral part of ETL metadata.
2.	**Metadata Centralisation**
2.1	Improvement of analytics as well as user interaction with the BI system: Analytics covers a wide range of techniques starting with a simple query based reporting, continuing through OLAP analysis and up to complex data mining. The user interaction with these techniques is highly guided by the metadata layer. All the different kind of analytics should be metadata driven. Metadata should provide the user with the centralised information about the meaning of the data, the terminology and business concepts used within the enterprise and their relationship to the data. Hence metadata should allow posing precise, well-directed queries and reduces the costs for users accessing, evaluating and using appropriate information.
2.2	Data integrity and accuracy: Centralized metadata should assure non-redundant / non-duplicated metadata information. In addition, high data quality requires data traceability and reconciliation. ETL procedures should manage the metadata traceability by capturing the data heritage (e.g. source, schedule information, receipt timestamps) and the reconciliation through methods like checksum. Centralizing all this information aids in speedy resolution of data integrity issues and the well management of accuracy of captured information.
3.	**Reduced efforts in BI system management**
3.1	Support for the development of new applications: Metadata provides the information related to the meaning of the data, its structure and origin.

No.	Requirements
	This aids in the requirements gathering and design phase yielding control and reliability of the application development process. Furthermore, metadata regarding design decisions adopted for existing applications may be reused.
3.2	Automated administration processes: Metadata should drive the execution of the diverse DW processes (like ETL, batch reporting). Information about the process execution (logs, DW data load status etc.) should also be stored in the repository for easy access by the administrator. These metadata driven processes should automate the complete BI administration, reducing the manual intervention and hence the effort in maintaining the BI system.
3.2	Sophisticated security mechanisms: ACLs and user profiles should be well managed in metadata layer in order to provide a sophisticated security mechanism. The different granularity of information with department wise / geography wise restrictions needs to be appropriately maintained with the user roles. Security breaches need to be detected through a robust audit trail process.
4.	**Flexible metadata architecture**
	Extendibility and adaptability of metadata: Metadata should be extendable and adaptable to changes. E.g. semantic aspects likely to change frequently can be explicitly stored as metadata outside the application programs yielding flexibility in extending the system and adaptability for including new metadata objects without difficulty. Also generic metadata models can enable reusability of various code fragments.

It is necessary to create a Metadata Management Team, which is typically divided into the Program and Project Levels and comprises of Metadata Administrators, Coordinators, Information Analysts and DBAs. Once this team is in place, the high level Metadata Requirements are established in discussion with Business and Technical Beneficiaries.

Specifications

After the Framework Definition phase is completed, the next step would be defining the metadata specifications, which include the following activities and sub-activities.

- Metadata Current State Inventory: Creation of inventory of Functional information requirements, Data models, process models, data dictionaries, dictionaries of business terms, Existing metadata environments and System documentation
- Metadata Requirements
  - Industry Standards to be followed
  - Metadata Model Requirements: Nomenclature, structure, element details, relationships
  - Metadata Interface Requirements: Metadata Repositories and details, Bridges, owner, system access, metadata lineage
  - Metadata System Requirements
  - Metadata Reporting Requirements
  - Security Requirements
  - Change Management Requirements
  - Training Requirements
  - Governance Requirements

Design

This phase includes establishing the following:

- Metadata Standards
  - Developing Data Standard for Data Elements
  - Technical and Cross-functional Review of Data Element Standards
  - Establishing Data Element Design Rules and Nomenclature
- Inbound Interface Mechanism
  - Source Metadata API and Bridges
- DW Metadata Schema
  - Metadata Classification Dimensions
  - Using Metadata Dimensions for designing the Metadata model
  - Data Element Definition Procedure
  - Configuration management
- Interoperability (metadata delivery) Mechanism
  - File Exchange
  - Repository API
  - Metadata Services
- Metadata Synchronization Mechanism
  - Degree of Federation
  - Replication Control and Update Propagation
  - Replication Control with a Shared Repository

Metadata Management Maturity Model

This Metadata Management Maturity Model presents a perspective to understand the metadata based knowledge management culture in the organization. In order to benchmark the maturity of the Metadata Management, we need to analyze its association with the People, Process and Technology.

- How People start using and sharing the metadata and then eventually set discipline to manage it as a part of the business operations?
- How Process governance matures to remove redundancy and achieve automation through the metadata driven activities?
- How Technology investment matures from distributed to centralized metadata management environment with better knowledge representation and reasoning.

There are five levels of metadata management maturity as described below.

Early

At the early stage, the metadata knowledge mainly resides with people and is created, stored and consumed locally using documents, modeling tools or even application specific tools. Any sharing is purely by accident and in an ad-hoc manner such as conversations and emails. Though Metadata exists at this level, there is absolutely no standardization or automation across the enterprise and hardly any awareness of the same.

Emerging

At this level, people are slightly more aware about the importance of metadata and consciously add information to any central repository that is available. Any sharing of information happens through the repository and sometimes interested applications can use the information available in this repository. Typically, the capturing of information in the repository is only done at the logical level (business metadata) and as a post-design activity. Any technical metadata is captured using isolated custom and bespoke processes, often implemented post-deployment.

Established

At this level, an enterprise wide drive exists to educate people about the importance of metadata, define governance processes for management of metadata and ensure that the processes are followed adequately. Metadata changes are tracked and audited and workflow is setup to control these changes. Metadata versioning is seriously taken into account. A large portion of the metadata related processes are near-real-time and a proper Metadata Management tools begin to be implemented. These implementations evolve over time and start interfacing with data integration / ETL tools.

Optimized

At this level, the people have got used to using Metadata and are constantly looking to optimize the Metadata Management processes. They work together and conscious effort is spent in improving the quality (as opposed to quantity) of Metadata and enterprise standards are fully defined. The reliability and trust-worthiness increases greatly and metadata becomes an integral part of any process. The Metadata across the enterprise is standardized, business vocabulary and taxonomies are defined. This gives the common reference model, which can be accessed by various people and processes to optimize their activities. Enterprise-wide implementation of Metadata Management tool is carried out.

Automated

This is the final level of maturity, as of now and at this level, Metadata Management starts to happen automatically as a part of any business or technical process. Metadata is deemed critical to any work but it totally engrains itself into the wider scheme of things. Semantic interoperability becomes possible. Domain Ontologies begin to be created, which allows greater inferences and cognition. Different schemas of data can be related without human intervention and any integration requires minimum effort. It becomes an important part of the Knowledge Management system with enhanced reasoning capabilities.

The metadata management maturity levels described above have been summarized in the table below.

Level->	Early	Emerging	Established	Optimized	Automated
People					
Awareness	Just Aware	Aware of Importance	Disciplined	Seeking Optimization	Sub-conscious and Dependant
Usage	Individual based and Uncontrolled	Conscious	Controlled	Authoritative	Inherent
Process					
Management	Locally	Disparate but Discoverable	Workflow Managed	Standardized	Ubiquitous
Interoperability	Conversational	Syntactic	Structural	Structural	Semantic
Technology					
Tool	Documents and Modeling Tools	Application Specific Repositories	Metadata Management Tool	Enterprise Metadata Management Tool	Enterprise Metadata Management Tool
Stress	None	Completeness and Correctness	Reliability and Low Latency	Efficiency and Quality	Seamless Integration

This metadata management maturity model presents a perspective to understand the ‘AS IS’ state of the Metadata Management program in the organization. In the vibrant market space, it provides a path to Experience Certainty, while continuing with the journey towards the perfection.

References

1. BIDS™ Metadata Management Solutions – Tata Consultancy Services Ltd
2. Metadata Management Paradigm – Kamlesh Mhashilkar

About the Authors

Kamlesh Mhashilkar is the Head of Business Intelligence and Performance Management (BIPM) Services under Technology Excellence Group at Tata Consultancy Services Ltd. (TCS). He holds graduation (B. Tech. in Electrical Engineering) from IIT, Bombay. Along with the thrust on bringing domain experience into analytical computing, his expertise span across Business Strategy, Practice Management, Corporate Research & Development, Consulting and Coaching. His prime contribution is in terms of TCS' Business Intelligence solutions branded as BIDS™ and bringing delivery specific rigor to it. He has led many BI solution deliveries in the domain of financial services, telecommunications and logistics.

Jaideep Sarkar is a Solution Architect within the BIPM Services under the Technology Excellence Group (TEG) at Tata Consultancy Services Ltd. (TCS). He holds a graduation degree (in Civil Engineering) from NIT Jamshedpur, India. He is currently leading the Solution Design and Assurance Functions within the TCS managed BIPM platforms at a leading Telecommunications Company in the UK, which is one of the domains in which he has an extensive experience. In the past, he has held Solution Design, Business Analysis, Data Modeling and Implementation roles in BIPM implementations in the Utilities, Transportation and Government industry domains.

For more information contact

BIPM Services

Tata Consultancy Services Ltd.
5H89, Yantra Park, Subhash Nagar, Unit VI, Pokharan Road No. 2,
Thane (W): 400 601, India.
Phone: +91-22-6778-2899

Kamlesh Mhashilkar

kamlesh.mhashilkar@@tcs.com

Jaideep Sarkar

Jaideep.sarkar@tcs.com

TATA CONSULTANCY SERVICES

www.tcs.com