LANTERN™ Dynamic Metadata

modemetric.com

modemetric

Abstract

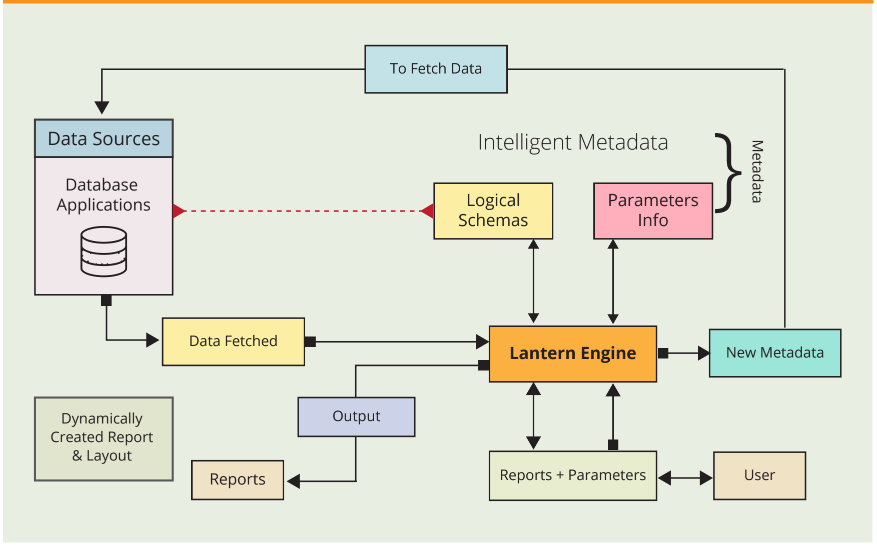
In the world of IT data, given the large number of systems, sources and data types, context is essential; and metadata is the key that puts all data into context. The "Metadata Lifecycle" is an integral part of any modern business intelligence (BI) solution. It involves multiple stages, which a technical resource can review in order to accurately and precisely develop an effective metadata strategy for a business where a BI solution is being implemented. Mismanaged and "static" metadata makes it extremely difficult to adapt to changes after a BI solution has been deployed. This translates into an immense drain in time and resources, due to the amount of effort needed to manage even the smallest change in the business requirement.

An effective solution for avoiding such a problem is to deploy a strategic or "dynamic" metadata with an added virtual intelligence, i.e., intelligence devised by a technical person, which can drive the automation of corporate and business level reporting and analysis needs. Such a solution reduces time in change requests and also improves data accuracy and governance. A dynamic and centrally managed metadata repository can speed up the implementation of new requirements as modification of existing ones. Additionally, it also supports the reusability of reporting and query algorithms, automation of business reporting, visualizations, and full data traceability.

The primary benefit of meta-dynamism is that all metadata, including transformational metadata, can be managed, validated and governed centrally, while facilitating faster, more consistent generation of reports and visualizations as well as analysis needs.

In this technical White Paper designed specifically for individuals charged with a successful BI strategy, we will describe how modern BI solutions, such as Modemetric's Lantern Platform utilize the concept of "Dynamic Metadata" and, identify the benefits of this approach compared to the constraints of conventional metadata used in traditional BI.

Dynamic Metadata Architecture



The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

Conventional Metadata: Background

Reports are visually available information generated from raw data, which are subsequently used to gain insight and support informed business decision making. The question of how to quickly generate a precise report with the most valid, accurate information is the challenge many of today's forward-looking organizations seek to resolve and optimize. This is where the role of metadata and its static or dynamic nature comes into play.

In the following sections, we will show how a dynamic metadata approach is much more powerful and effective than the implementation of a conventional static and rigid repository of configurations, queries, algorithms, tables, fields, etc.

A new metadata paradigm is required for the following situations, when:

- Configuration is changed for the host solution (ERP, CRM, SCM, HRM, etc.)
- User needs to add attributes, fields, and dimensions dynamically to existing data model
- Security rules need to be implemented and the same metadata will not work for every user

In most BI solutions and platforms, metadata is utilized to store logical schemas and end-user configurations. Creating metadata repositories and taxonomies that are optimized for an organization is very challenging, as every end user may have a different way of expressing the same or similar descriptor. The goal is to provide users with the right information as well as information distilled from a variety of distinct content, which can then be converted into actionable insight.

Traditionally, metadata repositories tend to create information silos, which makes it extremely difficult to integrate or communicate with other systems and adapt to changes in the organization, requirements, or the implemented logic. To cater to diversified requirements, of different users and business functions, it becomes necessary to create multiple independent metadata repositories, which are dependent on each other.

Metadata based solutions are not easy to build. For each customer, the requirements are different and the implementation of the host system is unique. Therefore, the specifications of the metadata vary greatly and, sometimes, become too complex and unmanageable, which makes it difficult to achieve the desired output in a timely manner. This requires having a metadata that meets the users' requirements, which change frequently, and can be modified on the fly with no development effort.

Metadata Approach by BI Vendors and Lantern: Static vs Dynamic

Metadata is typically required for maintaining:

- Schema information of the underlying database of the host solution
- Configuration of the host solution
- Information about logical schemas, which are required to retrieve information desired by a user

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

Most BI solutions create metadata paradigms that are static in nature. For example, for any ERP, CRM, SCM, HRM or any database solution, a unique metadata needs to be defined for each unique implementation of that metadata. However, dynamic metadata is characterized by the ability to change itself automatically based on the underlying configuration of the host solution.

Lantern's metadata approach, besides being dynamic, is also intuitive. It can modify itself on the fly, based on the parameters selected by an end-user. This functionality equates to less time needed on design and development; therefore, more time can be spent on analysis and solving the business problems.

For example, Oracle EBS (E-Business Suite, also known as Oracle Financials), is very rich in functionality and provides extensive customization options for its end-users. It even facilitates the addition of custom fields even after deployment has been completed. Therefore, as can be expected in such a situation, the configuration may continue to change long after deployment. This Oracle EBS flexibility makes it almost impossible for conventional BI products to dynamically identify and incorporate changes, without extensive effort and customization.

In the real business world, we can expect dynamic requirements from a user for changing the output. For example, at runtime, a user may want to add a new dimension or convert the existing data in different dimensions. Such changes are challenging and cannot be managed with static metadata-based solutions. Some static metadata-based BI solutions use an approach where the tool fetches all the data from the data source and, subsequently, at runtime modifies the output based on the user's requirement. Using this approach causes performance issues and impacts both hardware and network resources. Also, this approach cannot address many complex and advanced user requirements.

The three major challenges for conventional and static metadata implementations are that metadata is not:

1. Dynamic
2. Intuitive
3. Flexible

Due to these three limitations, the full power of such BI solutions is not realized.

Static Metadata Challenges Explained

Unique Metadata for Each Implementation

When metadata is static, it becomes necessary to develop a unique metadata for each configuration for the same host solution. Since configuration may greatly vary from one deployment to another, it entails developing metadata from scratch for each deployment. This requires additional resources, time, and development costs, which most organizations either cannot afford or are unwilling to incur.

Solution Upgrades

Another challenge involves upgrading existing solutions. Since each deployed solution has its own metadata, upgrade efforts (development, testing, and debugging) will be required for each single deployment.

Modifying Base Configurations

Another challenge with static metadata is that it cannot sense any changes in the base

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

configuration. Therefore, whenever a change is made to the host solution, the metadata needs to be modified as well. A typical example of this scenario is a connectivity situation such as an ETL (Extract, Transform & Load) configuration. In most organizations, after a data warehouse is implemented, it becomes very cumbersome to make any changes, even as minor as adding/deleting a field, to the ETL process.

Dashboards or Report Layout Manipulations

When the layout of any dashboard component is changed, such as adding or deleting a column, the effort required can be significant if a conventional BI tool is used.

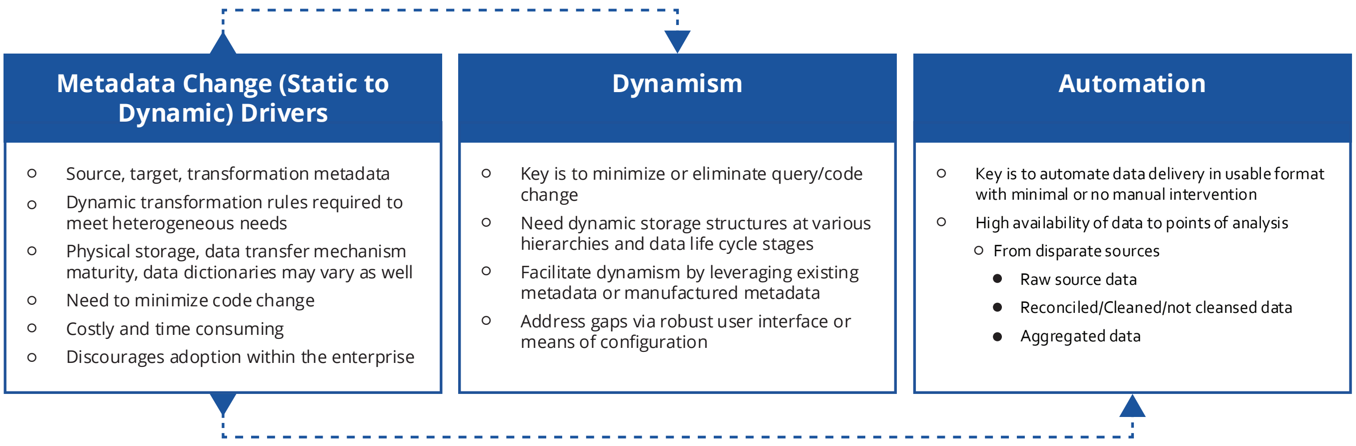
Updating/Changing Metadata as per User Requirement

Another very challenging issue is to modify existing metadata, as per the user requirements, so that the tool retrieves only the required content. In most conventional BI solutions, when a metadata is defined it remains static. Therefore, static and all data elements defined in the metadata will be retrieved from the host solution, even in the case where a user may need only 10% of the defined fields.

Take the example where a business schema has 20 fields that are defined and can be retrieved. If a user only needs 5 fields, then most solutions will always retrieve 20 fields, even though they will only display the 5 requested fields. This has an impact on the processing requirement, network bandwidth, as well as performance. In this case, if 15 fields are dropped, it can create another problem. In a complex model, selecting 5 fields may also force the selection of another 2 fields in order to enable joins or filters, which may be required to be applied for obtaining the desired data. This challenge becomes even greater with cloud based solutions, since customers are charged on usage basis.

Another aspect of this problem pertains to the case where a user wants, at runtime, to increase the number of fields that are defined in the metadata by making different choices. For example, if the metadata is defined for 20 fields, and the user wants to select only 5 fields but wants to split one of the 5 fields in 4 unique fields based on some logic, conventional metadata practices will not support this requirement.

The solution to all the challenges identified in this paper is a dynamic, intuitive metadata model. Such intuition cannot be derived from the underlying data model or by merely reading the relationship among the various data objects. Lantern's metadata meets all these requirements, it is dynamic, intuitive, and flexible. This flexibility allows for easy modification and enhancement of the metadata. The following figure provides a bird's eye view of Lantern's dynamic metadata and its drivers:

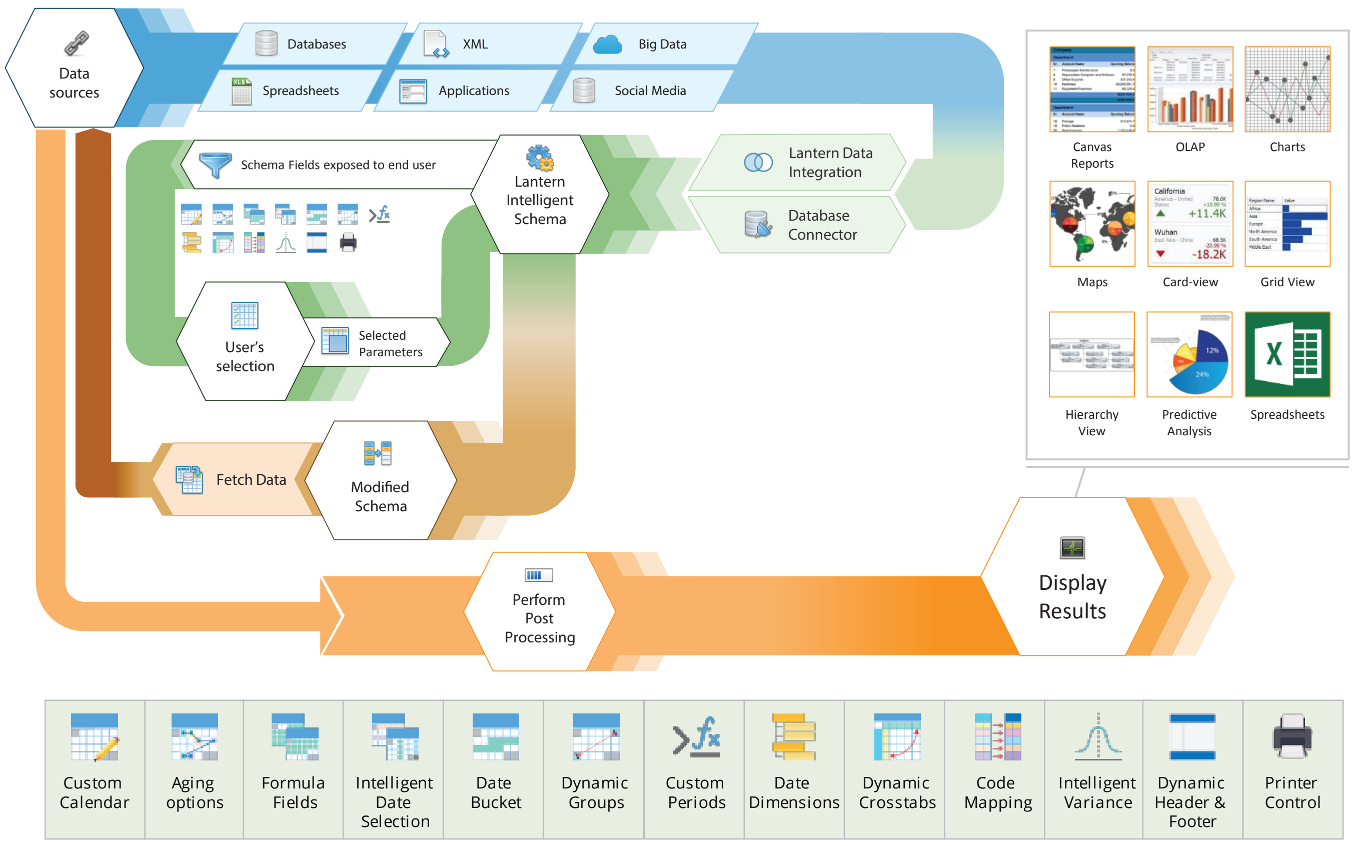

Metadata Change (Static to Dynamic) Drivers
Source, target, transformation metadata
Dynamic transformation rules required to meet heterogeneous needs
Physical storage, data transfer mechanism maturity, data dictionaries may vary as well
Need to minimize code change
Costly and time consuming
Discourages adoption within the enterprise
Dynamism
Key is to minimize or eliminate query/code change
Need dynamic storage structures at various hierarchies and data life cycle stages
Facilitate dynamism by leveraging existing metadata or manufactured metadata
Address gaps via robust user interface or means of configuration
Automation
Key is to automate data delivery in usable format with minimal or no manual intervention
High availability of data to points of analysis
From disparate sources
Raw source data
Reconciled/Cleaned/not cleansed data
Aggregated data

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

Lantern Dynamic Metadata - An Introduction

Lantern’s metadata approach is extremely powerful in the context of BI in that enables the end user to perform analysis without being encumbered by the limitations expected from traditional BI tools which rely on static metadata. Starting from a simple to a complex query, sub-query or program and ranging to the icons, buttons and interfaces used in the Lantern application-- all are very strategically and flexibly programmed into the Metadata architecture and utilized as per user calling or requirement.

The following diagram displays a high-level view of Lantern’s metadata workflow and architecture:



The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

Essentials of Dynamic Metadata

Patented & Unique

Lantern's metadata model is based on a unique patented technology, which is programmable and customizable—essentially what makes the metadata dynamic. With this unique metadata model, intelligent BI/reporting solutions can be developed that are host configuration independent and do not require redevelopment for different configurations. This technology makes Lantern unique among Traditional BI architectures.

Centralized

Lantern's metadata model is centralized, which means the same metadata works for desktop, client/server, mobile, cloud, web, and appliance mode deployment. This architecture enables great scalability as a standalone deployment can be upgraded to an enterprise level deployment without changing a single report or a dashboard.

Database Independent

Lantern is database independent, which enables it to work with a database from any vendor. Therefore, its metadata can be ported from one database to another without any modification. This makes Lantern future-proof and allows organizations to upgrade, or standardize database platforms, without having to go through a difficult, costly, and time consuming process.

Programmable

Lantern's metadata is programmable. This feature enables development of complex and intelligent applications that require almost no customization, as modification is automatic. For example, Lantern can develop a reporting solution for a specific ERP, CRM, or other application, without requiring any customization when deployed. The solution will automatically configure itself, based on the configuration of the host application

Object Oriented

Lantern's metadata model is object oriented and extremely granular, which enables development flexibility and highest efficiency, due to the reusability of objects

Programming Standards

Lantern's metadata supports both ANSI and native programming language support as per requirement. It does not require any additional routines, queries or procedures to implement and execute the desired report query.

Data Segregation & Identification

Each primary entry within the Lanterns metadata has a unique APP ID associated with it for accurate data identification and segregation.

Query / Sub-Query

Lantern metadata flexible data model enables users to write the most complex and nested queries required for complex report development. It is possible to easily embed N number of queries within the main query, which the Lantern engine can decipher and process for returning the required result.

Sub-Report

Lantern metadata enables users to design sub-reports and call them in the main report. For example, in the case of a group report, you can design a sub-report (template) that

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

contains the representation of the grouped data to be shown on the report. This sub-report can subsequently be called whenever a grouped data report is to be generated. The same concept also applies to the multi-data source report, where multiple report outputs are generated, viewed, and printed as a single detailed report.

Debugging

Lantern Metadata enables the user to very easily trace the source of the error within the executed query and, as required, correct and/or optimize it. Every metadata element or attribute is made traceable with the help of GUID's, primary keys, and foreign key relations. So if a problem occurs, whether it is in the menu, report, dataset or the query, you can quickly find the problem and take corrective action.

Dynamic Metadata Use Cases

Lantern's Dynamic Metadata opens a world of endless possibilities for the users. In this section, we are going to layout 3 specific use cases that aim to build the understanding towards our unique approach to BI. We will take into consideration the following Use Cases:

- Applications with Configurations available through API
- Applications where Configuration is available but not exposed through API
- Real Time and Dynamic Report Output Manipulation

Applications with Configurations exposed through API (Metadata, WEB Service, DLL)

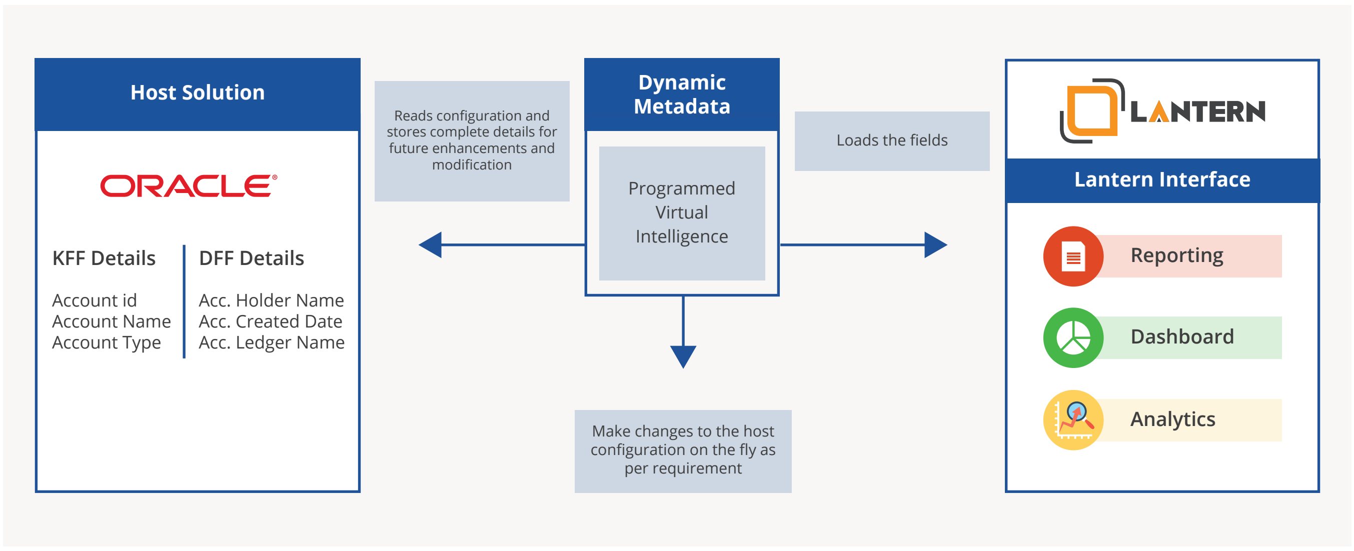
Commonly, BI Solutions are unable to modify their base configurations based on the configuration of the host solutions. Oracle EBS offers two primary options to its users known as Oracle KFF (Key Flex Fields) and DFF (Descriptive Flex Fields). KFF are basically required to define the primary structure of the module for instance, Account Name, Item Name, etc. whereas, DFF are additional descriptive fields like Company MD Name, Department Head Name.

In most cases, KFF are defined during the implementation of the system and remain unchanged during the lifecycle of the product, whereas DFF can be added, updated, or modified at any time or as per user requirement.
Now the complete definition of the KFF and DFF fields allows the user to fetch the available data and run the required reports in various supported formats. For Instance, the KFF contains complete information field like Account Name, Account Id, Account Type, Transaction Type and DFF contains supportive fields like Account Holder Name, Account Created Date, Account Ledger Name. This seems like a very simple configuration but in reality its quite complex and the query built behind this involves data being pulled from various fields and later presented to the user.

For any other BI solution, it is very difficult to read all these fields and complexity or any

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

modification made to these on the fly as per user requirement. However, with Lantern's Dynamic Metadata, the complete configuration of any solution can be dynamically programmed into the metadata and the respective changes are automatically reflected on the interface. Now, the user can very easily interact with the data of their choice without having to learn the complexity and the relationships at the backend.



Similarly, updating Lantern metadata and its design is handled through programmed virtual intelligence and very uniquely segregated and kept in numerous tables with easy access to complete trail of modifications. Ultimately, making it very easy and simple to dynamically make any existing or newly defined fields available to the user.

Lantern can read the configuration and design of any implemented host solution and load the respective fields for the user to utilize for reporting, visualizations and analysis purpose. In addition, it is very simple to duplicate the newly read and loaded configuration for other installations of the same nature. One more important aspect of Lantern's Dynamism is that it can be configured to consume the security model of the host solution, if it is exposed via API.

Applications where Configuration is available but not Exposed through API

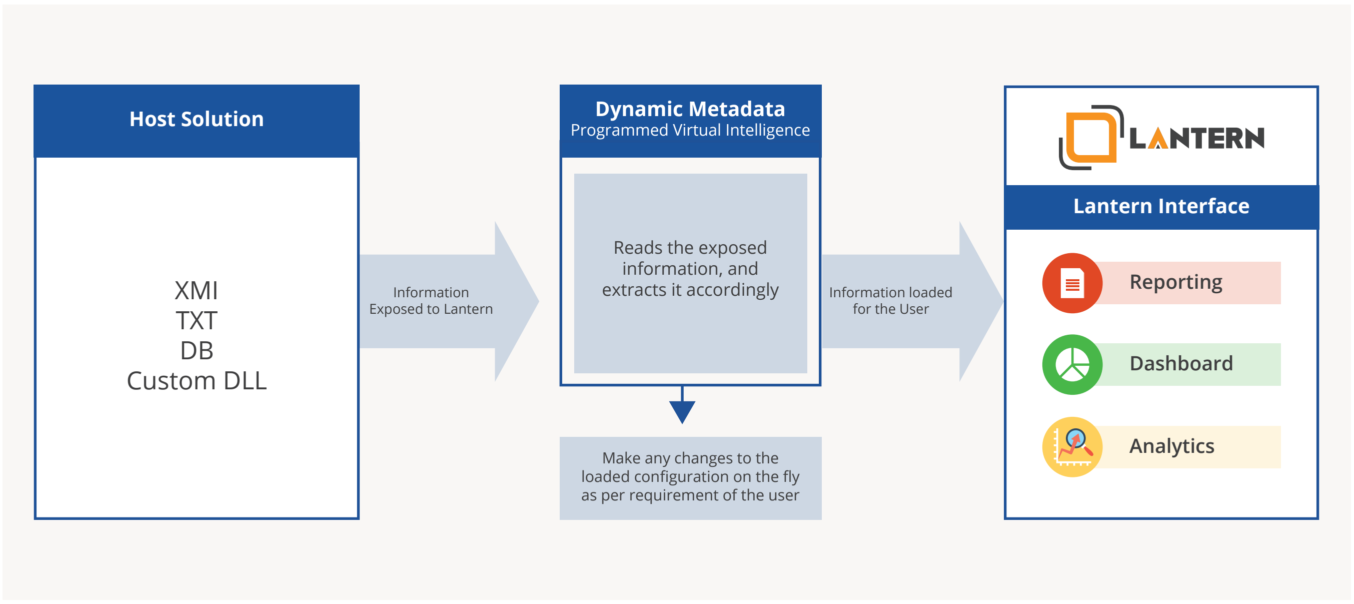
Sometimes, organizations have implemented a system whose configuration is not readily available through API's but rather stored in XML, TXT, DB, Custom DLL's or in other non API formats. In such cases, reading, duplicating and loading such configurations can become a hectic process for almost all BI solutions.

However, configurations kept in any format mentioned above can be easily exposed to and consumed by Lantern Intelligently programmed metadata, which will accordingly change itself on the fly. Ultimately, reflecting the necessary functionality to the user for reporting, dashboards and analysis need.

Let us take an example of an organization that is storing all its accounts and departments

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

information in text format. The text file must have a specific structure like Account Title, Account Name, Account Type, Account Budget, Department Name, Department ID etc. Now, if reporting is required on this data, most BI solutions would need to write special programs first to read the data and then extract it in the desired manner; ultimately presenting it to the user in a static standard format. But Lantern can read the exposed information, extract It, transform it load it intelligently into the metadata and present it to the user for reporting, dashboards and analysis purposes without having the need to write any special codes or programs.


Host Solution
XMI
TXT
DB
Custom DLL
Information Exposed to Lantern
Dynamic Metadata
Programmed Virtual Intelligence
Reads the exposed information, and extracts it accordingly
Make any changes to the loaded configuration on the fly as per requirement of the user
Information loaded for the User
LANTERN
Lantern Interface
Reporting
Dashboard
Analytics

Furthermore, in the specific case of Oracle, organizations have misused the implementation of its Quality Assurance module by defining a certain account payable related field in it and then linking it with the AP module of Oracle to cater a special reporting need. User can store such configuration on a text file, expose it to Lantern, which can then intelligently consume it, update its base configuration and load the same user configuration on the application for reporting, creating dashboards and analysis purposes.

Hence, intelligently updating Metadata saves time, resources and provides a more desirable and precise output.

Real Time and Dynamic Report Output Manipulation

Multiple Nested Crosstabs and Date/Time buckets generation in real-time is one of most primary requirements a BI solution should fulfill dynamically that is without the need to write complex codes and queries.

Let us take an example of an organization that globally sells a certain product in 20 different cities (City 1 to City 20). Now, a senior executive wishes to view the sales of that product in 5 specific cities (City 5 to City 10) cross tabbed by Customers (Customer 1 to Customer 50). Let us assume that the total number of records behind this request are 50 million.

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

Below is how the data is stored on the DB server

Customer 1	City 1	200.00
Customer 2	City 1	400.00
Customer 3	City 5	500.00
Customer 4	City 5	1000.00
Customer 5	City 8	2000.00
Customer 10	City 8	1500.00
Customer 15	City 4	1200.00
Customer 25	City 4	2200.00
Customer 35	City 20	2700.00
Customer 50	City 20	3600.00

Following is the output required by the client that is sales in 5 cities cross tabbed by customer

Customer	City 1	City 2	City 3	City 4	City 5
Customer 1	Sale	Sale	Sale	Sale	Sale
Customer 2	Sale	Sale	Sale	Sale	Sale
Customer 3	Sale	Sale	Sale	Sale	Sale
Customer 4	Sale	Sale	Sale	Sale	Sale
Customer 5	Sale	Sale	Sale	Sale	Sale

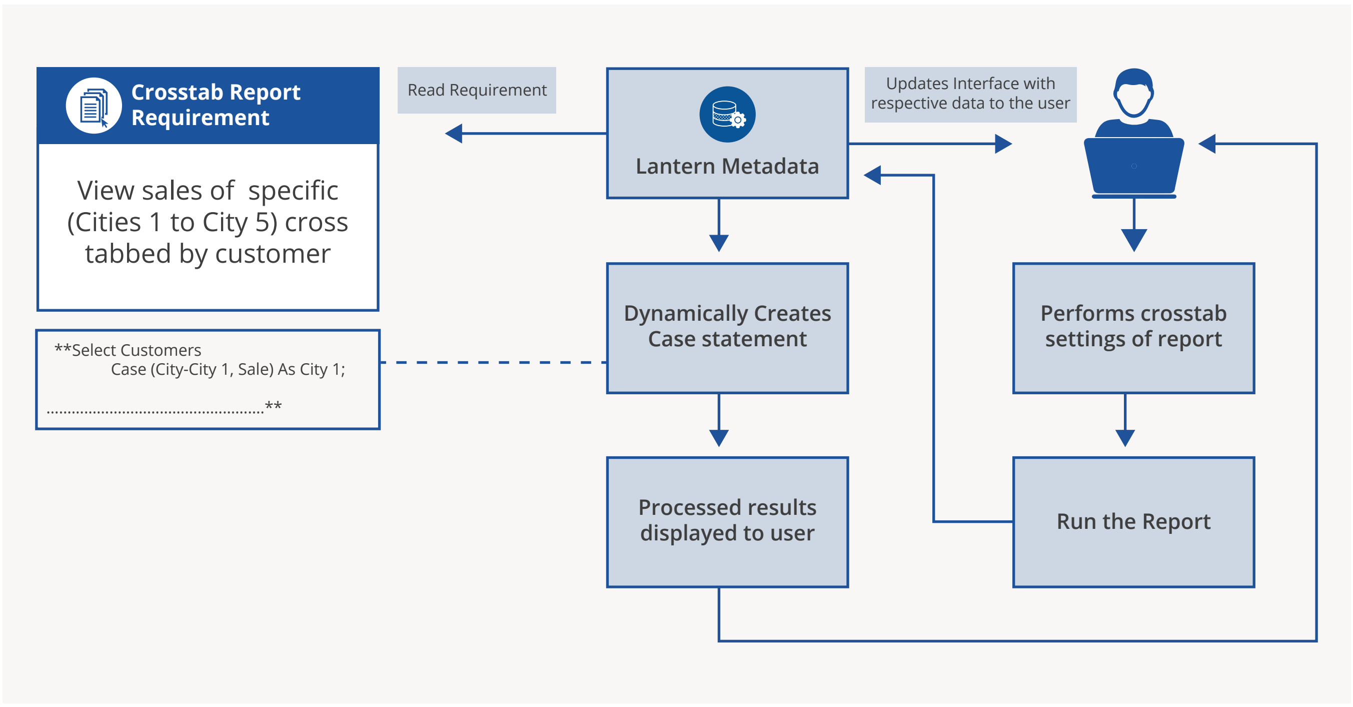
To achieve this, most BI solutions will fetch all the available data (50 million records) of the 20 cities from the DB server, snapshot it in memory and process the data to show the desired Sales of the 5 specific cities.

This methodology is resource hungry and raises the cost of the solution especially if its implemented in a cloud based subscription model.

In comparison, Lantern Dynamic Metadata has the unique ability where it can be modified on the fly based on the user requirement and dynamically form data fetching requests (like SQL Query), which are optimized to retrieve only desired data in required format. To fetch the result, Lantern Dynamism enables the user to select the required fields, apply the crosstab settings, and its metadata intelligently builds the following case statement for retrieving the data

```
**Select Customers
        Case (City= City 1, Sale) As City 1;
                        ......................................................**
```

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.


Crosstab Report Requirement
View sales of specific (Cities 1 to City 5) cross tabbed by customer
**Select Customers
Case (City-City 1, Sale) As City 1;
....**
Read Requirement
Lantern Metadata
Updates Interface with respective data to the user
Dynamically Creates Case statement
Performs crosstab settings of report
Processed results displayed to user
Run the Report

Dynamic metadata is intelligent enough to only fetch the required data and is not resource hungry at all; ultimately improving speed, performance and accuracy, and saving precious time.

Features & Benefits

- ✓ Lantern metadata definition and maintainability is very efficient from a resource (human, hardware and network) perspective
- ✓ Lantern metadata can be modified on the fly; minimal or no development effort is required, depending on user requirement
- ✓ Enables simple, advanced, complex customization and management through an intelligent interface
- ✓ Promotes usability; updates or enhancements made to the metadata during a deployment can easily be implemented for customers using the same solution
- ✓ Solutions built for various customers can be made part of the Lantern metadata and, subsequently, made available as new product features and functionality
- ✓ Advanced object oriented design effectively manages all metadata objects (datasets, queries, sub-queries, sub-reports, interface icons etc.) separately
- ✓ Lantern metadata integrity, confidentiality and availability is not impacted due to user requirement changes
- ✓ Metadata can intelligently adapt to the host system configuration
- ✓ Detects programmed changes and refreshes itself automatically
- ✓ Complete trail of data manipulation can be accessed for debugging and traceability purposes

The information contained in this presentation is confidential and proprietary to Modemetric. The recipient of this document agrees that they will not disclose its contents to any third party or otherwise use this presentation for any purpose other than an evaluation of Modemetric's business or its offerings. Reproduction or distribution is prohibited.

Conclusion

In the world of business, change is a constant—BI solutions need to scale accordingly. Traditional BI architectures are simply "behind the times" when it comes to quickly and effectively meeting change requirements—often times even with the simplest of upgrades. One of the main reasons for the inability to respond quickly to changes is the static metadata foundation of Traditional BI, which means even minor changes can require manual efforts, which equate to time and costs—and even then they are restricted to the latest build which presents a potential burden on overall IT flexibility and performance.

Lantern resolves the challenges which come from ever-changing data models from the host and other dependent systems through its patented dynamic metadata architecture. It provides organizations with the flexibility to make changes on the fly which equates to an overall savings in costs and resources, but above all, it provides organizations with the actionable insight based on the most relevant and real-time data.

You can also refer to our overview White Paper, titled "**Lantern: BI and Advanced Analytics for Today's Enterprise,**" which provides additional information pertaining to other key capabilities and features of the Lantern platform.

Lantern: BI and Advanced Analytics for Today's Enterprise
An Overview for IT and Business Leaders
From Any Source to Actionable Insight
Modemetric Inc. White Paper Series

For more information, please visit us at
www.modemetric.com or contact us at:
555 California Street, Suite 4925, San Francisco, CA 94104
Phone: +1 833.663.8742 (MMETRIC)
Email: info@modemetric.com