Data Abstraction Best Practices White Paper

Project Name	AS Assets Data Abstraction Best Practices
Document Location	This document is only valid on the day it was printed. The source of the document will be found in the ASAssets_DataAbstractionBestPractices folder (https://github.com/TIBCOSoftware)
Purpose	Self-paced instructional

This document presents a detailed explanation of the open source Data Virtualization Data Abstraction Best Practices capability. It provides detailed overview of the architecture and usage.

TIBCO Software empowers executives, developers, and business users with Fast Data solutions that make the right data available in real time for faster answers, better decisions, and smarter action. Over the past 15 years, thousands of businesses across the globe have relied on TIBCO technology to integrate their applications and ecosystems, analyze their data, and create real-time solutions. Learn how TIBCO turns data—big or small—into differentiation at www.tibco.com.

www.tibco.com

Global Headquarters
3303 Hillview Avenue
Palo Alto, CA 94304

Tel: +1 650-846-1000
+1 800-420-8450
Fax: +1 650-846-1005

Revision History

Version	Date	Author	Comments
V1	04/28/2010	Mike Tinius	Initial revision
V3	08/23/2010	Mike Tinius	Added SOA Read/Write white paper
V5	11/05/2011	Mike Tinius	Added Business Views in Business Layer
V7	04/08/2013	Mike Tinius	Modified major layers to 3. Enhanced technical guidance.
V7.1	08/20/2013	Mike Tinius	Updated documentation to Tibco format.
V8.0	11/07/2013	Mike Tinius	Updated for release 8.0
V8.1.2	04/09/2014	Mike Tinius	Removed version number from title page
V8.1.6	05/20/2015	Mike Tinius	Updated White papers and Powerpoints to Cisco format.
V8.1.9	12/06/2017	Mike Tinius	Transitioned to Tibco for release 8.1.9
2018Q1	03/20/2018	Mike Tinius	Release 2018Q1 – no changes.

Related Documents

Name	Version
Tibco Data Abstraction Best Practices White Paper.pdf	2018Q1
Tibco Data Abstraction Best Practices.pptx	2018Q1
Tibco Data Abstraction Best Practices Technical Guide.ppt	2018Q1
Tibco SOA-Centric Read/Write Methodology Technical Note.pdf	2018Q1

Supported Versions

Name	Version
TIBCO® Data Virtualization	7.0 or later
AS Assets Utilities open source	2018Q1 or later

© Copyright 2000-2017 TIBCO Software Inc.

Table of Contents

© Copyright 2000-2017 TIBCO Software Inc.

1 Complexity and Agility Require Data Abstraction

Business and IT Challenges

All large enterprises and government agencies today are looking to improve on their bottom-line, cut costs or reduce their risk by providing better access to information assets.

Towards these objectives, every organization struggles with the challenges presented by significant volumes of complex, diverse data spread across various technology and application silos. Further complicating matters are a range of problems such as each source having its own access mechanism, syntax, security, etc., few structured properly for business user or application consumption, let alone reuse, incomplete data or duplicate data, as well as a mix of latency issues.

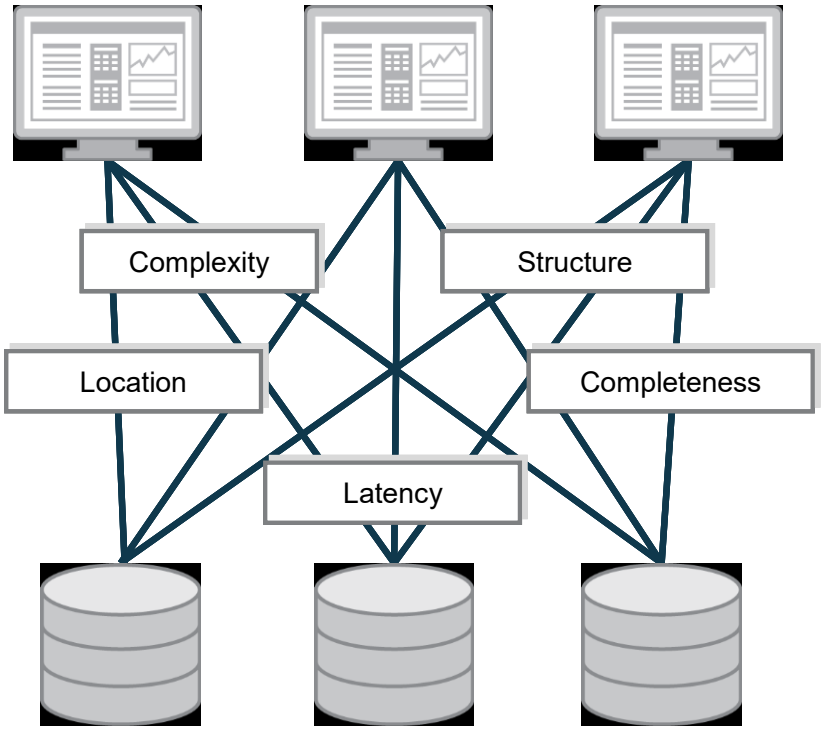


Figure One: Data Abstraction Challenges

Data Abstraction Overcomes These Challenges

Data abstraction overcomes source to consumer incompatibility by transforming data from its native structure and syntax into reusable views and data services that are easy for application developers to understand and solutions to consume.

Some data abstraction approaches enterprises use today work better than others.

For example, some organizations build data abstraction by hand in Java or use business process management (BPM) tools. Unfortunately, these are often constrained by brittleness and inefficiencies.

© Copyright 2000-2017 TIBCO Software Inc.

Further, such approaches are not effective for large data sets since they lack the robust federation and query optimization functions required to meet data consumers' rigorous performance demands.

Data warehouse schemas can also provide data abstraction. Data modeling strategies for dimensions, hierarchies, facts and more are well documented. Also well understood is the high cost and lack of agility in the data warehousing approach. Further, data warehouse based schemas don't include the many new classes of data (big data, cloud data, external data services and more) that reside outside the data warehouse.

Data Virtualization is a Superior Solution for Data Abstraction

Data virtualization is an optimal way to implement data abstraction at enterprise scale. From an enterprise architecture point of view, Tibco's data virtualization solution forms semantic abstraction or a data services layer in support of multiple consuming applications. This middle layer of reusable services decouples the underlying source data and consuming solution layers. This provides the flexibility required to deal with each layer in the most effective manner, as well as the agility to work quickly across layers as applications, schemas or underlying data sources change.

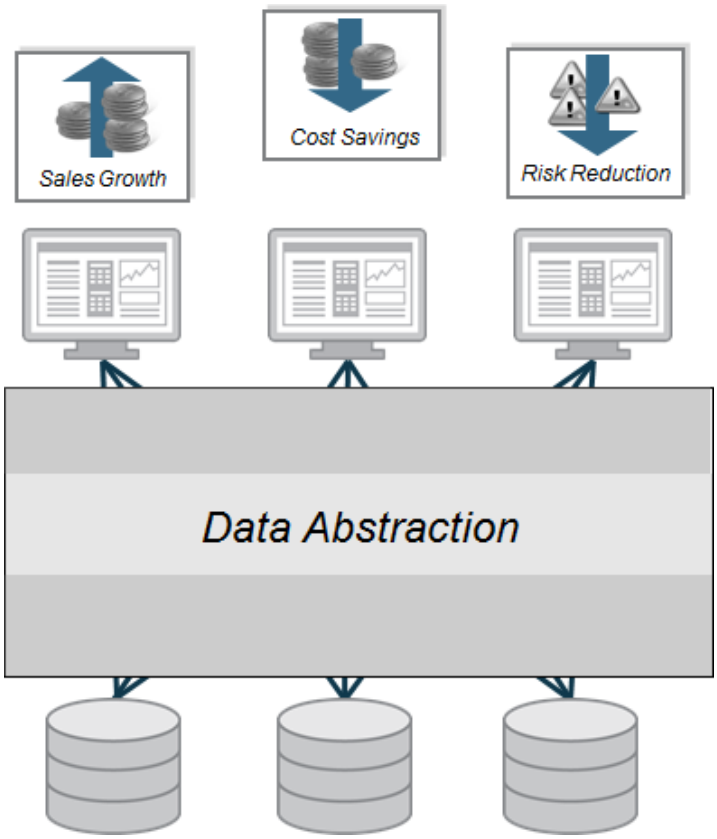


Figure Two: Data Abstraction

Data abstraction helps enterprises and government agencies achieve a number of key objectives including:

1. **Right business information at the right time** – Fulfill complete information needs on demand by linking multiple, diverse data sources together for delivery in real-time.

© Copyright 2000-2017 TIBCO Software Inc.

2. **Business and IT model alignment** – Gain agility, efficiency, and reuse across applications via an enterprise information model or logical business model. Often times, known as the canonical model, this abstracted approach overcomes data complexity, structure, and location issues.

3. **Business and IT change insulation** – Insulate consuming applications from changes in the source and vice versa. Developers create their applications based on a more stable view of the data. Allow ongoing changes and relocation of physical data sources without impacting consumers.

4. **End-to-end control** – Use a single platform to design, develop, manage and monitor data access and delivery processes across multiple sources and consumers.

5. **More secure data** – Consistently apply data security rules across all data sources and consumers via a unified security methods and controls.

Selected Examples

Simplify Multiple Reporting Needs via Data Abstraction – A major money-center bank implemented a data abstraction layer to simplify prime brokerage, reconciliation, risk management, and more (over 25 applications in all) from over 200 unique sources. This Tibco data virtualization use case accelerated new reporting development and reduced IT costs.

Accelerate Time to Market using Data Abstraction – Pharmaceutical giant, Pfizer, Inc., implemented a data abstraction layer to simplify and accelerate access to a wide range of research and clinical trials data across its R&D, marketing and manufacturing teams. Using Tibco data virtualization in this way provides decision makers with the information required to bring new drugs to market faster, while adhering strictly to FDA regulations.

© Copyright 2000-2017 TIBCO Software Inc.

2 Tibco's Data Abstraction Reference Architecture

The diagram below outlines the layers that form Tibco's Data Abstraction Reference Architecture. Architects and analysts can use this as a guide when abstracting data using Tibco's data virtualization platform. The various layers included in this reference architecture are described below:

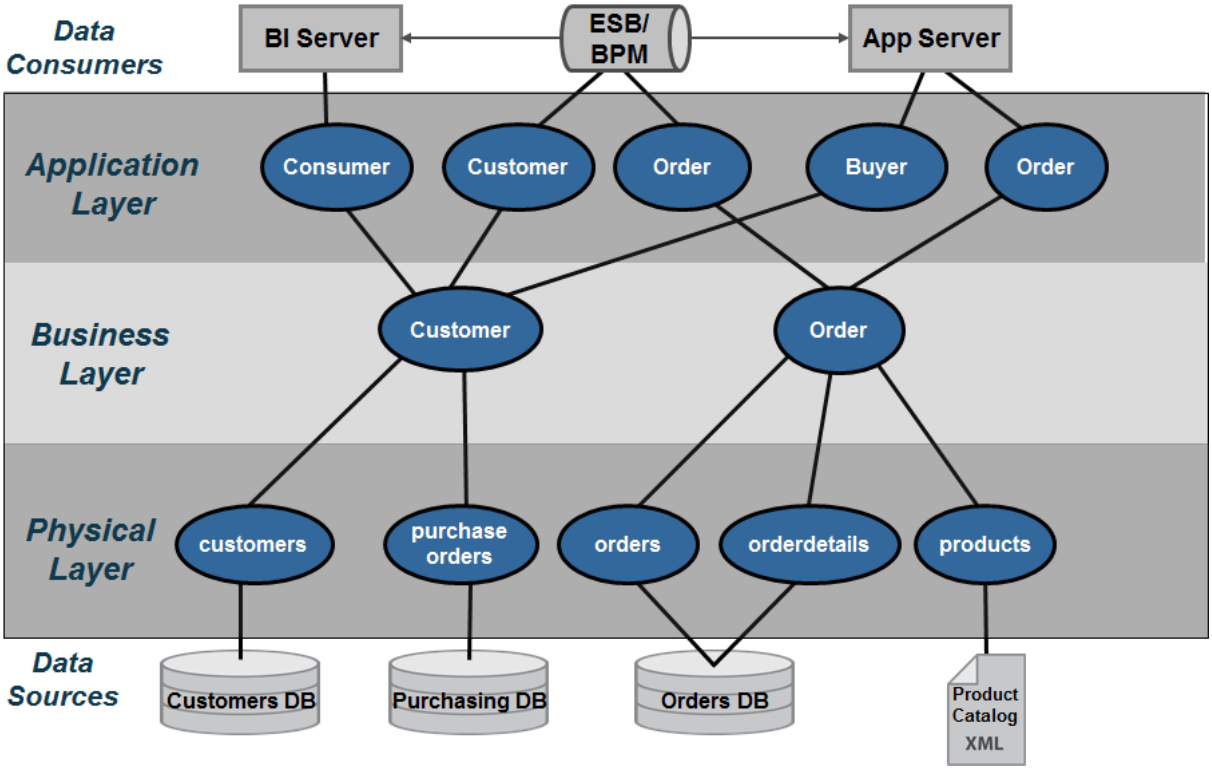


Figure Three: Tibco's Data Abstraction Reference Architecture

- **Data Consumers** – Client applications need to retrieve data in various formats and protocols that they understand. Tibco delivers the data to consumers using the most popular standards including SOAP, REST, JDBC, etc.

- **Application Layer** – The "Application Layer" serves to map the Business Layer into the format which each application data consumer wants to see. It might mean formatting into XML for Web services or creating views with different alias names that match the way the consumers are used to seeing their data.

- **Business Layer** – The "Business Layer" is predicated on the idea that the business has a standard or canonical way to describing key business entities such as customers and products. In the financial industry, one often accesses information according to financial instruments and issuers amongst many other entities. Typically, a data modeler would work with business experts and data providers to define a set of "logical" or "canonical" views that represent these business entities.

© Copyright 2000-2017 TIBCO Software Inc.

These views are reusable components that can and should be used across business lines by multiple consumers.

- **Physical Layer** – The “Physical Layer” provides access to underlying data sources and performs a physical to logical mapping by integrating physical metadata and formatting views.
    - The physical “Metadata” is essentially imported from the physical data sources and used as way to onboard the metadata required by the data abstraction layer to perform its mapping functions. As an “as-is” layer, entity names and attributes are never changed in this layer.
    - The “Formatting” resources provide a way to map the physical metadata into the Data Virtualization layer by aliasing the physical names to logical names. Additionally, the formatting views can facilitate simple tasks such as value formatting, data type casting, derived columns and light data quality mapping. This layer is derived from the physical sources and performs a one-to-one mapping between the physical source attributes and their corresponding “logical/canonical” attribute name. This layer serves as a buffer between the physical source and the logical business layer views. As such, caching may be introduced at this level if and when it makes sense. Rebinding to different physical views during deployment is another role these views take on. Naming conventions are very important and introduced in this layer.
- **Data Sources** –The data sources are the physical information assets that exist within and without an organization. These assets may be databases, packaged applications such as SAP, Web services, Excel spreadsheets and more.

© Copyright 2000-2017 TIBCO Software Inc.

3 Enabling Forrester's Data Virtualization Vision

Forrester Research provides the following guidance for data abstraction in their "Data Virtualization Reaches Critical Mass" report.

According to Forrester "Leading firms have implemented a layered architecture combining both physical and virtual data stores, choosing the appropriate mix based on different areas' performance requirements. In the most-successful cases, the firms create an hourglass-shaped architecture that funnels mappings of disparate source data through canonical business information models. As a result, one large drug manufacturer is able to successfully operate with both IBM Cognos and SAP Business Objects BI tools for different customer groups, using data that has been reconciled against common metadata using virtualization.

In addition to the use of canonical models in the middle, we have identified two other noteworthy characteristics of this architecture: 1) virtual/physical modality choices tend to be more physical in the staging layers close to the actual data sources and more virtual as data moves closer to the users, and 2) a final virtual mapping layer ensures that the solution provides data to consumers in just the required format."

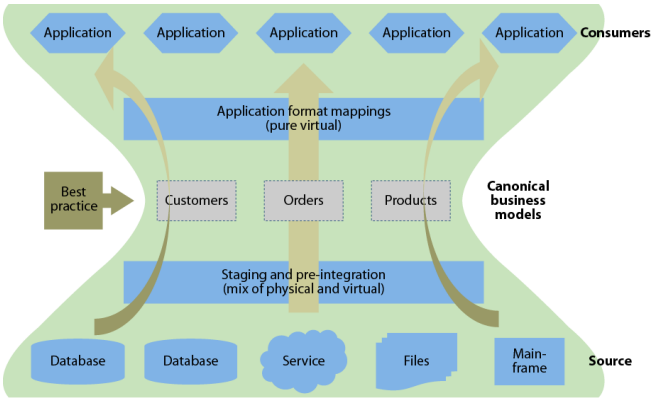


Figure Four: Source: Forrester Research Data Virtualization Reaches Critical Mass

There is a striking resemblance between Forrester and Tibco's best practice architectures:

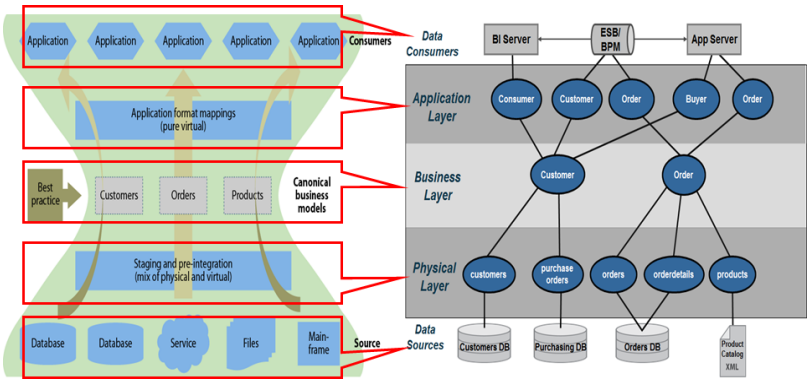


Figure Five: Comparing Forrester and Tibco's best practices architectures

© Copyright 2000-2017 TIBCO Software Inc.

4 Enabling Gartner's Discipline of Data Integration

Since 2005, Gartner has been researching the concept of data services in relation the broader business and IT evolution. More recently, Gartner discussed the "Discipline of Data Integration" at the Business Intelligence Summit as shown in Figure Six.

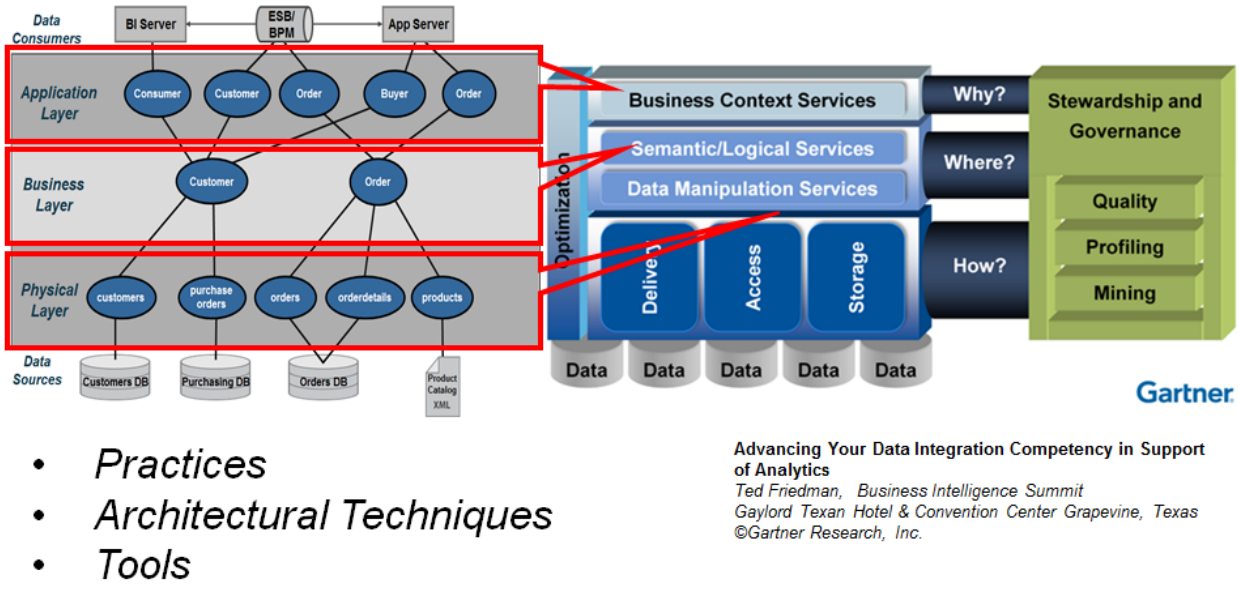


Figure Six: Tibco's Data Abstraction Architecture Implements Gartner's Discipline of Data Integration

According to Gartner, **"**The discipline of data integration comprises the practices, architectural techniques and tools for achieving consistent access to, and delivery of, data across the spectrum of data subject areas and data structure types in the enterprise, to meet the data consumption requirements of all applications and business processes. As such, data integration capabilities are at the heart of the information-centric infrastructure and will power the alignment and delivery of data in support of BI and performance management. Contemporary pressures are leading to an increased investment in data integration in all industries and geographic regions. Business drivers, such as "the imperative for speed to market" and "agility to change business processes and models", are forcing organizations to manage their data assets differently. Simplification of processes and the IT infrastructure are necessary to achieve transparency, and transparency requires a consistent and complete view of the data, which represents the performance and operation of the business."

Tibco's data abstraction reference architecture can be used to implement Gartner's "Discipline of Data Integration" as follows:

- **Practices** – Tibco has shaped the best practices that customers use today to implement data virtualization in their organizations. Tibco also has influenced the practices of recommended by leading IT analysts and system integrators. This thought-leadership and real-world experience helps users gain confidence when deploying data virtualization in their organization.

© Copyright 2000-2017 TIBCO Software Inc.

- **Architectural Techniques** – Tibco Professional Services brings a wealth of knowledge and skills to help users architect their data virtualization solutions. Tibco's architectural techniques are including in Tibco Professional Services' Quick Start program. This program is designed to help customers get a project up and running quickly and maximize their return. During this program, customers are introduced to the "Data Abstraction Best Practices Technical Guide" that Tibco Professional Services Consultants use an architectural techniques blueprint.

- **Tools** – The Tibco Data Virtualization Platform provides a complete and proven tool to implement Gartner's "Discipline of Data Integration".

- **Business Context Services** – In Tibco's reference architecture the Application Layer provides the mechanisms for mapping and publishing views or web services in the context of the applications. Tibco's Application Layer maps into Gartner's Business Context Services. Application consumers require delivery of data via different protocols. Within Tibco's reference model, data consumers use a variety of standard protocols including JDBC, ODBC, SOAP/HTTP, REST and ADO/.Net to access needed data. These standard protocols support the BI, MDM, Web service API's and Enterprise Objects consumers included by Gartner.

- **Semantic/Logical Services** – Gartner's "Semantic/Logical" services provide for the transformation of the physical model into the business context view of the information. The term logical and semantic are often referred to as canonical. It is a way of defining a common data dictionary across the business. The terms or attributes from this data dictionary are grouped together into semantically similar entities. Tibco supports these needs with its formatting views.

- **Data Manipulation Services** – Gartner's "Manipulation" functions include Access, Storage, and Delivery which align with Tibco's physical layer. This is where Tibco's introspection, discovery, and source data access tools expose the physical layer. Increasingly, Tibco is providing access to a wide array of data sources including relational, service oriented, file, packaged applications and big data.

- **Optimization** – Both Gartner and Tibco view optimization as spanning the entire architecture from source to consumer, both during design and runtime, perfectly matching how Tibco's optimizers work.

Gartner's recent research on the Logical Data Warehouse extends and enhances this guidance.

© Copyright 2000-2017 TIBCO Software Inc.

5 User Roles and Responsibilities

Implementing an enterprise scale data abstraction layer involves a variety of IT staff as can be seen in below. In this section, we identify the roles and responsibilities needed for successful data abstraction layer development and operation.

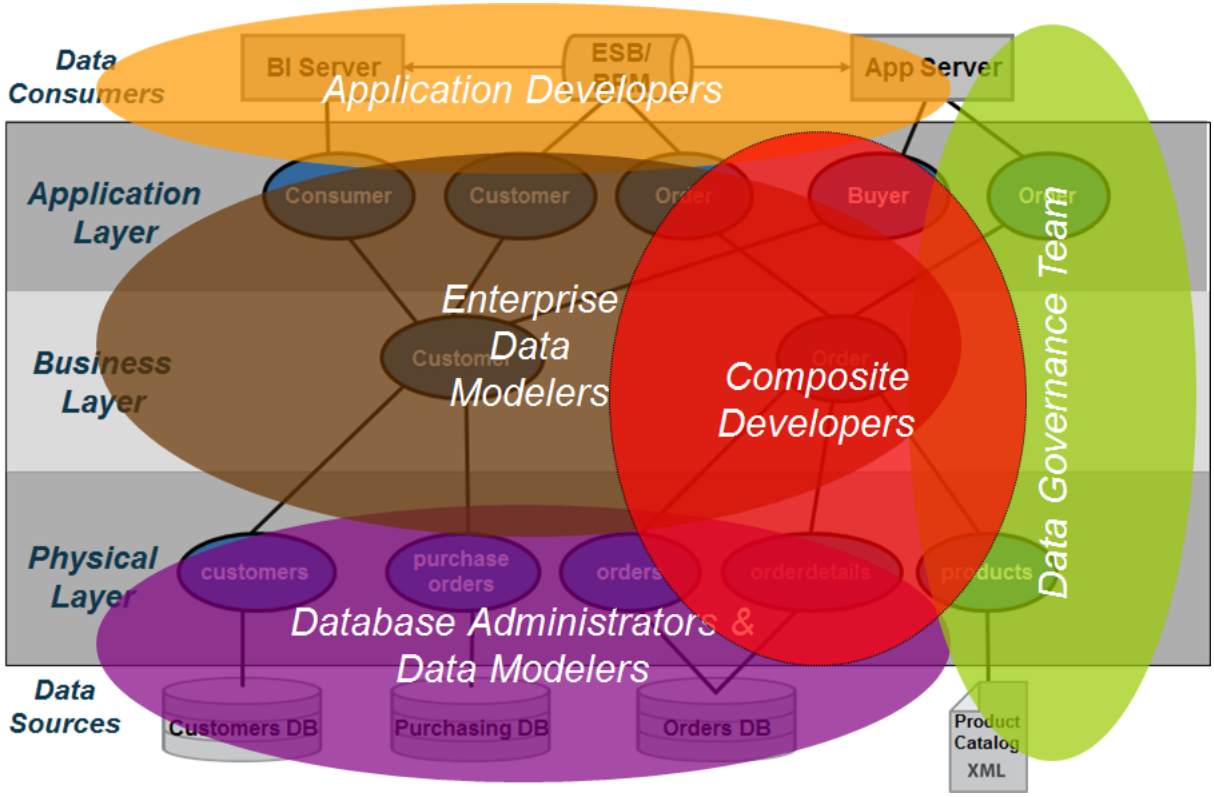


Figure Seven: Data Abstraction Roles and Responsibilities

The following roles and responsibilities should be considered when implementing a data abstraction layer:

- **Application Developers** – Application developers are concerned with the API's and the various mechanisms by which they can access the data from the data abstraction layer. Often times, data lineage is important to the client-side developers because they need to know the originating source. Tibco can provide data lineage from the client services views down to the physical metadata.

- **Enterprise Data Modelers and Data Architects** – Enterprise data modelers and data architects work with subject matter experts who know the business and use their expertise to craft logical data models. These logical data models can be further refined into something that is closer to a logical database design which could be used as the basis for building views in the business layer. Enterprise data modelers work with Tibco developers to design effective views and services.

© Copyright 2000-2017 TIBCO Software Inc.

- **Tibco Developers** – Tibco developers are responsible for the implementation of the layers using the Tibco data virtualization middleware. This implementation is based on the overall architecture and concepts laid forth in this document and performed in conjunction with the other teams involved. They need to understand the application developer requirements so they can construct the API layers and views above the business layer as well as the underlying sources. Beyond lineage tools and folders available within Tibco, these developers often use additional control systems to understand and manage key mappings across the various layers.

- **Database Administrators (DBAs) and Database Modelers** – DBAs and database modelers provide access to the physical data sources. Further they help define logical database designs from their corresponding physical database designs and implementations. They work with Tibco developers to ensure are properly tuned and assist with creating indexes if necessary.

- **Data Governance Teams** – Data governance staff are involved throughout the process making sure that enterprise data access and data mapping rules are followed. Further, they provide guidance on how data should be modeled and often help to resolve data quality issues.

© Copyright 2000-2017 TIBCO Software Inc.

6 Data Service Design Guidance

In this section, the data service design and development efforts required to implement Tibco's data abstraction reference architecture are described starting with the physical layer and moving up the consumer. This is representative of how a typical Tibco developer would approach this activity.

Physical Layer

The physical "**Metadata**" is imported from the customer's data sources. It provides a physical representation of the data source's metadata which structurally looks exactly like the data sources; however, it does not store any data. In essence, it is a virtual representation of the data source.

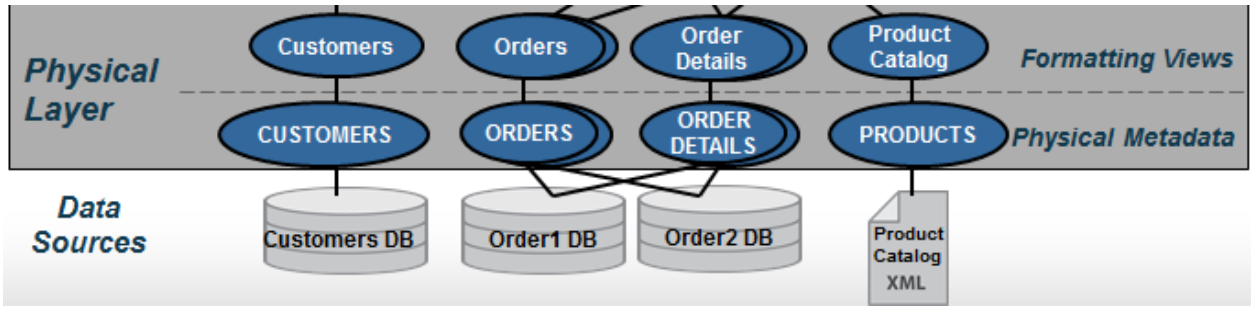
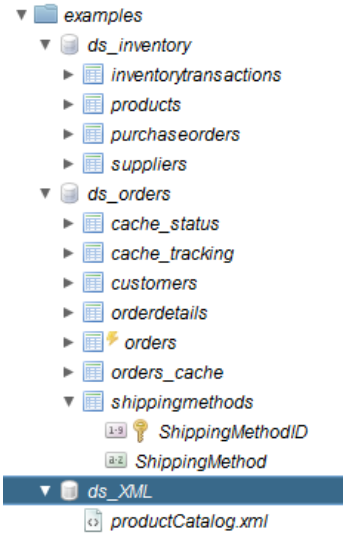


Figure Eight: Physical Layer

In Tibco, the "New Data Source" wizard tool walks the Tibco developer through a series of screens to import metadata for a source to generate this physical metadata. There are no manual steps involved here. Physical sources can be relational (tables, views, stored procedures, SQL Statements), Web services, Flat delimited files, XML files, Java functions, packaged applications such as SAP and much more. Besides capturing table and column structures, this wizard performs introspection of indexes, primary keys and foreign keys. This metadata also includes cardinality statistics on source tables used by Tibco's cost-based optimizers.



© Copyright 2000-2017 TIBCO Software Inc.

Figure Nine: Introspected Objects within the Physical Layer Example

Tibco allows for re-introspection of these sources either manually or automatically. This is important where sources change over time, thereby providing application insulation. With Tibco in place and a canonical model established for applications, source changes typically only impact next layer up, the formatting layer.

The job of the "**Formatting**" sub-layer is two-fold. It standardizes the data elements into the terms or data dictionary established by an organization's business logical views. This is often referred to as "data canonicals." This layer also insulates applications and upper level views from lower level change for example when new columns are added to a data source. For applications to take advantage of those changes, the columns must be percolated to the upper views and published externally. However, this kind of change can be managed in the form of new versioned data services that provide a new signature of data while maintaining the integrity of existing applications. These views are typically SQL-based which provides for easier maintainability and understanding by the data architects.

The general guidelines for this layer are as follows:

- **One-to-one mapping** – These views provide a one-to-one correspondence between the physical and the formatting views. Generally, the names and types of data elements the physical layer are mapped to their corresponding "logical/canonical" name and type. An example of one-to-one mapping is shown below:

```
/project/Physical/Physical_Metadata/ORDER_SYSTEM → /project/Physical/Formatting_Views/Orders
        OID                 → OrderId
        CID                 → CustomerId
        ORDER_DT            → OrderDate
        PONUM               → PurchaseOrderNumber
        SHIP_ID             → ShipToId
        FR_CHRG             → cast (FR_CHRG as numeric(12,2)) FreightCharge
        WEIGHT              → Weight
        (FR_CHRG/WEIGHT)    →  CostPerWeight
        MTHD_ID             → case MTHD_ID
                                    when 1 then 'AIR'
                                    when 2 then 'GROUND'
                                    else 'OTHER'
                              end as ShipToMethod
        "ORDERS1"           → cast ('ORDERS1' as varchar) DataSourceName
        CURRENT_TIMESTAMP   → cast (CURRENT_TIMESTAMP as timestamp) StartTime
        NULL                → cast (NULL as varchar) EndTime
```

- **Simple mappings** – Actions that may be performed in this layer should be kept simple and should refrain from performing joins. The following is a list of actions that may be accomplished at this layer:
    - **Name aliasing** – Mapping the physical name to its logical/canonical counterpart. E.g. OID is mapped to the alias OrderId.
    - **Data type casting** – Data type casting is a form of transformation whereby the type of the physical column is cat to a different type as in "cast (FR_CHRG as numeric(12,2)) FreightCharge".
    - **Simple derived columns** – Derived columns are typically columns that can be calculated from existing columns. In the example provided above, the CostPerWeight is calculated

© Copyright 2000-2017 TIBCO Software Inc.

from the Freight Charge and the container Weight. Another example would be the concatenation of two or more fields to create a derived column.

- **Value formatting** – Value formatting provides conditional logic to return a different value in place of the original value. An example would be to asses an ID field and return a description. Refer to the "case" statement in the above example.
- **New columns** – An example of a new column introduced at this level is one where the data does not exist in the source. The data is provided at the time of this view creation through a "static" definition or a "system function".
  - An example of a static definition is to provide the data source name. For example, if you are mapping more than one Order Entry System, it might be advantageous to the Application Developers to know where a particular Order row is coming from (its data lineage).
  - Another example is the use of a system function such as CURRENT_TIMESTAMP. A custom function could also be invoked to populate the new column with data.
- **Null mapping** – It may be necessary to establish the alias column in this layer, yet it has no corresponding physical data element to map to. In this case, it is permissible to map the alias to a NULL value.
- **Light Data Quality** – Cleaning up known bad data using SQL functions, validation tables, etc.

An example of the above concepts is shown in the figure below:

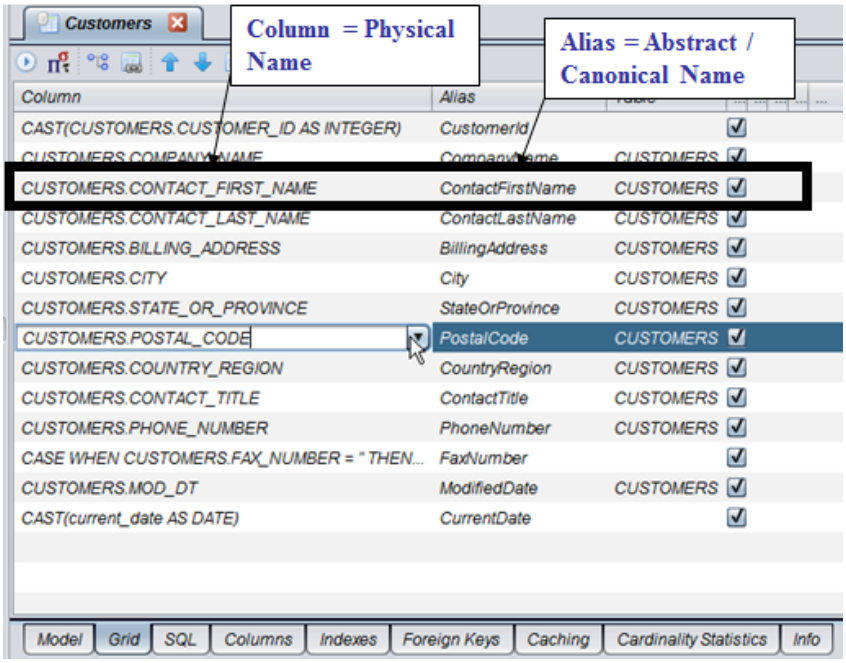


Figure Ten: Column Mapping Example

© Copyright 2000-2017 TIBCO Software Inc.

Caching – The formatting layer makes the most sense to perform initial caching of a source if necessary because the logical names, type casting and any transformation have already been applied thus reducing the downstream processing requirements. Caching of data sources can be done for various reasons:

- Protect against maintenance windows and downtime of underlying source
- Protect underlying data sources from excessive utilization
- Location transparency – guard against slow network bandwidth by caching data local to the processing

- **Rebinding** – When promoting resources from development to test/QA/Staging and production, it may be necessary to rebind the formatting views to a different physical metadata source. This may be as a result of a schema or path change in the data source.

Business Layer

Business layer views and services are grouped into subject areas defined by the organization's enterprise information model. The business layer is a logical, canonical representation of the key business entities and supports federation of data across multiple data sources. Often data modeling tools such as ER/Win and ER Studio are used to create a logical data design. These models can be used as the basis for the views and data dictionary at this level. Naming of objects should reflect the logical entity and attribute names determined by data modelers. Because this layer serves to federate multiple, like views together to form a single unified result set, naming used in the underlying formatting layer should be consistent.

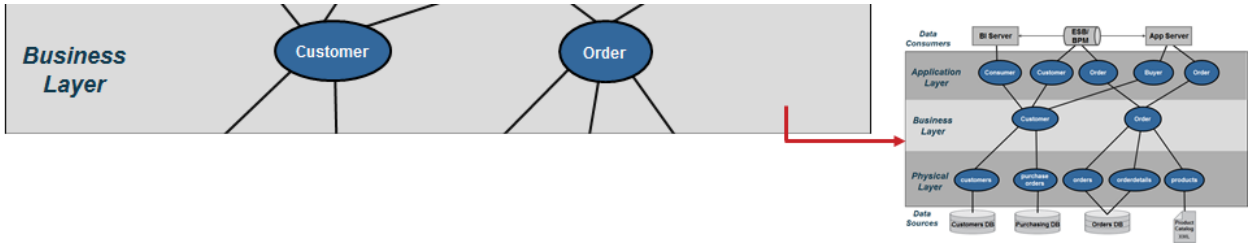


Figure Eleven: Business Layer

It is vital that this layer only access other logical views or formatting views and 'never' access the physical views. Logical views may be Tibco views, parameterized queries, or perhaps SQL procedures. The business layer in general may be a combination of views and SQL procedures. Views do not have logic other than joins, filters or parameters. SQL procedures provide logic and are much like stored procedures in databases. SQL procedures can access logical views or abstract views.

© Copyright 2000-2017 TIBCO Software Inc.

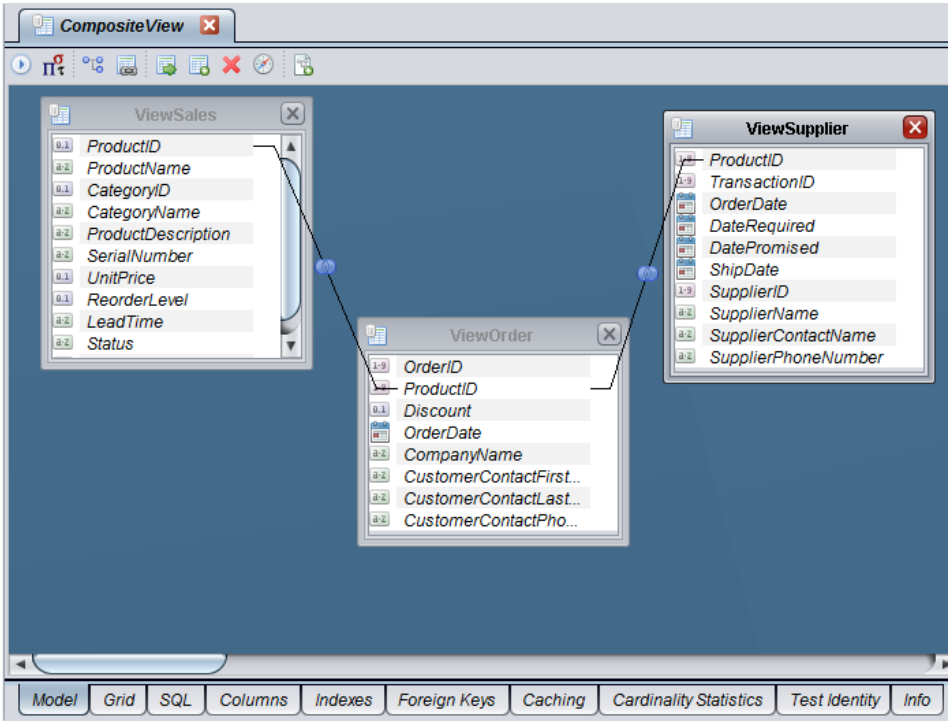


Figure Twelve: Federated Business View Example

Application Layer

Application layer views and services prepare the final output into the context required by the client application. It is recommended that these views and services match up to client API structures (object model, XML complex types and SQL tables). In other words, the client structure dictates the naming conventions and view contents. If XML shaping is required, then Tibco provides XQuery, XSLT or SQL procedures with XML functions for shaping XML. If SQL result sets are needed, then a SQL procedure can be used to provide the output as a cursor. Tibco SQL views or SQL procedures can be published as database resources. Web services can be published through top-down (aka contract-first) design or bottom-up (auto-generation) design methods.

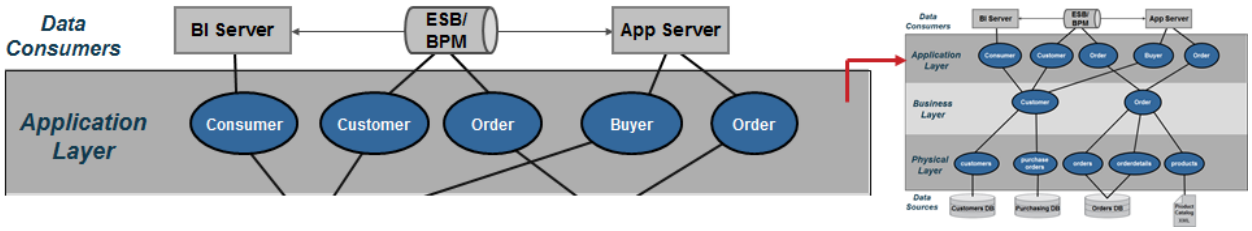


Figure Thirteen: Application Layer

Application layer guidelines include:

- **Client API Abstraction** – The application layer is the client API abstraction layer and names of items should reflect the needs of the client API object.

© Copyright 2000-2017 TIBCO Software Inc.

- **Business to Client Mapping** – This layer serves as the business to client mapping and may involve a projection of joins, transformations or views from the business layer.
- **Data Manipulation** – Data from the application layer views can be manipulated using “order by”, “group by” and aggregation but the names from the client views should be retained for consistency throughout the organization.
- **Narrow Results** – Narrow results by selection criteria and parameterized queries
- **Client** – Data can be materialized (cached) intelligently at this layer to provide the best possible consumer response and throughput.
- **Publish Database Views or Web Services** – The application layer contains candidates for publishing as a view or web service

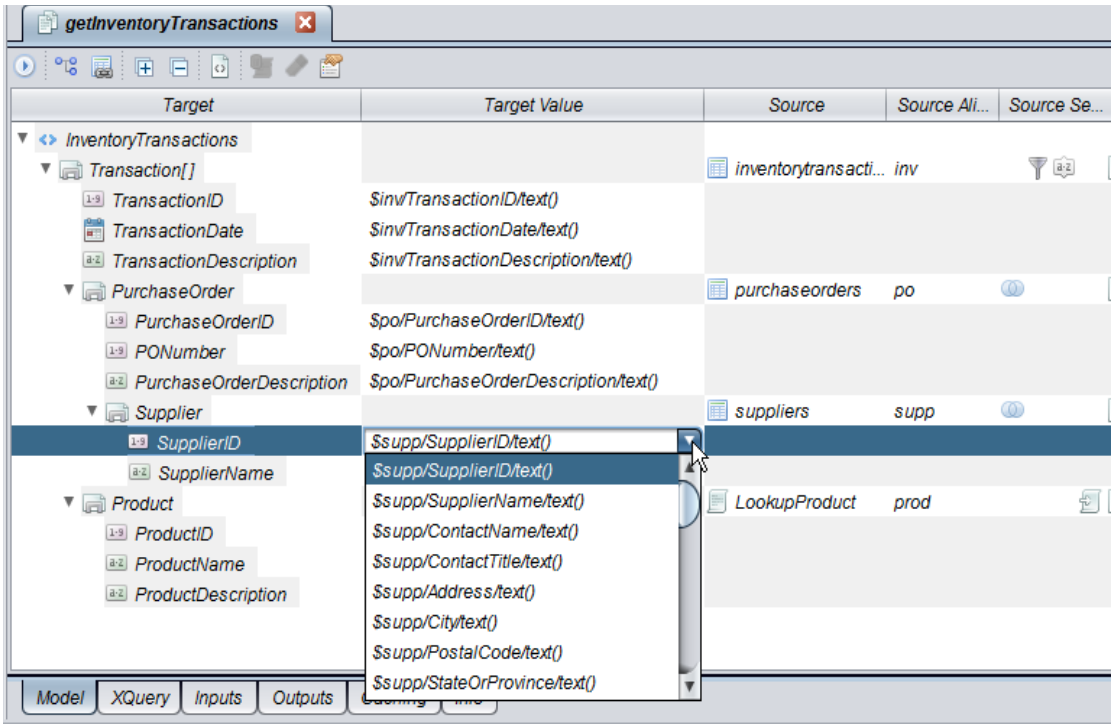


Figure Fourteen: Mapping Service Example

© Copyright 2000-2017 TIBCO Software Inc.

7 Summary of Key Benefits

Data abstraction bridges the gap between business needs and source data's original form. This best practice implementation of data virtualization provides the following benefits:

- **Simplify information access** – Bridge business and IT terminology and technology so both can succeed.
- **Common business view of the data** – Gain agility, efficiency and reuse across applications via an enterprise information model or "Canonical" model.
- **More accurate data** –Consistently apply data quality and validation rules across all data sources.
- **More secure data** – Consistently apply data security rules across all data sources and consumers via a unified security framework.
- **End-to-end control** – Use a data virtualization platform to consistently manage data access and delivery across multiple sources and consumers.

Business and IT change insulation –Insulate consuming applications from changes in the source and vice versa. Business users and applications developers work with a more stable view of the data. IT can make ongoing changes and relocation of physical data sources without impacting information users.

© Copyright 2000-2017 TIBCO Software Inc.

8 Practical Next Steps

Enterprises can get started on the journey to achieving the key agility and total cost of ownership benefits described above with a few simple steps. The key is getting started quickly with a manageable project that enables learning and a foundation for progress.

- **Set achievable goals** – Start with projects and a focused team. With success, broaden Business and IT team involvement to expand usage across departments toward the ultimate enterprise level deployment.

- **Determine levels of abstraction** – Are the four recommended layers right for your organization? Do you need greater depth within one or more layers? Tibco Professional Services can help answer these questions and get you started on the right path.

- **Determine modeling and mapping approach** – Should you use top-down, bottom-up, or some of both?

    - Top down – You have a vision and you want to find the data to fulfill it. This is often referred to as Contract-First design. In this approach Tibco Data Virtualization allows you to start with your own WSDL and map Tibco services to your contract.

    - Bottom up – You know what your data looks like, now how do you make it usable by others. In this approach, Tibco Data Virtualizationallows you to generate or publish resources such as SQL view and Web services directly from the Tibco Data Virtualization introspected sources.

    - Both – Mix and match appropriately according to domains and needs.

Start now! – Don't over analyze. Getting started now with small steps is the best way to learn, progress, and gain value.

© Copyright 2000-2017 TIBCO Software Inc.