onebridge

denodo

Whitepaper

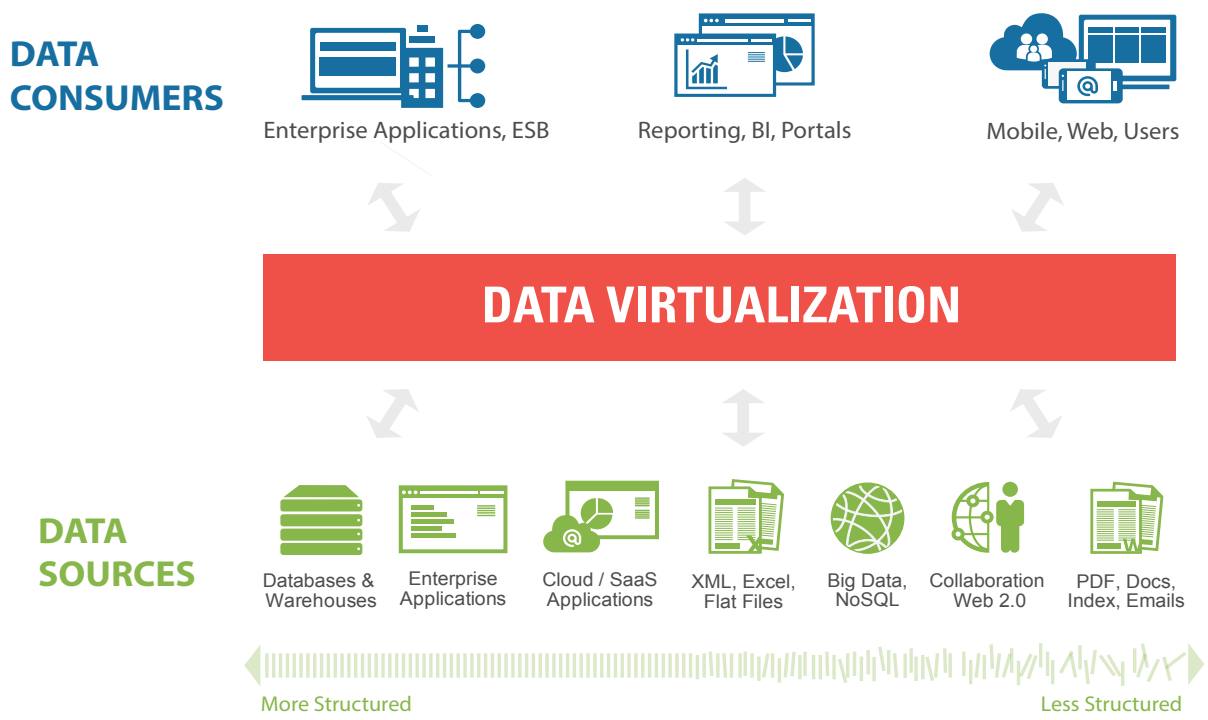
Data Virtualization Usage Patterns for Business Intelligence/ Data Warehouse Architectures

Modern organisations are having to react ever more quickly to competitive threats and new potential market opportunities and more than ever before need data which is up to date, comprehensive and easy to access. Business Intelligence has long been a key method for making this available and in recent years the most common method of serving data to this environment has been through the replication of data into a Data Warehouse architecture. Now these BI/DW architectures have to evolve fast to meet the rapidly increasing demands of today´s businesses. From a business perspective traditional BI faces the following challenges:

- Lack of business agility: rigidity in the traditional BI/DW architecture prevents new or ad-hoc data requests from being fulfilled on time.
- Lost revenue opportunities: high latency in processing and delivering important data.
- High costs: tightly coupled systems make the modifications required to align to business needs expensive.
- Incomplete information for decision making: challenges in accessing disparate enterprise, web and unstructured data.
- Limited self-service opportunities: multiple layers of processing make it difficult for business users to operate without significant IT assistance.

Today's users demand reports with better business insights, more information sources, real-time data, more self-service and want these delivered more quickly, making it hard for the BI professionals to meet such expectations.

In this document we show how the use of Data Virtualization can help BI professionals to accomplish such goals.



A BI/DW Architecture based on Data Virtualization

Data Virtualization technology greatly enhances the traditional BI/DW architecture offering the agility that business users demand.

The following picture shows the typical chain of data replication nodes that is commonly used in today´s BI/DW environments [Ref.1].

Figure 1 – **Chain of data replication nodes in a traditional BI/DW environment**

Either using the Hub & Spoke or the Bus Architecture, data replication is massively applied and multiple repositories have to be managed and maintained ending in a highly inflexible and costly environment. It typically requires considerable analysis and design effort before delivering results to users.

It is difficult to accommodate changes either as a result of changes in the production systems or in the final user requirements, as many ETL scripts have to be rewritten affecting the whole data delivery chain. Data is not up-to-date hindering operational reporting needs. Only structured content is typically handled and it is difficult to accommodate semi-structured and unstructured data types from either internal or external repositories (e.g. partner data).

Through Data Virtualization an abstraction layer is added between the consuming business intelligence tools and the underlying information sources (see Figure 2), that is characterized by the following aspects:

- The Data Virtualization layer exposes a Virtual Data Model that it is built by creating links to the information repositories (operational systems, staging, DW, data marts, etc), importing the required data schema from them in the form of base views, and creating the needed data transformations and combinations to generate composite views as final data services that are ready to be consumed by the business intelligence tools.
- It maintains a virtual schema meaning that only metadata is kept in this layer and not actual data. Data is delivered through this layer on demand, in real-time, as it is gathered from the underlying sources and combined according to the specific data combination required by each view.
- The Data Virtualization query engine makes use of sophisticated query optimization techniques to guarantee good performance, pushing down processing to the data sources as much as possible to minimize processing needs at this layer and the amount of data received from the sources, and by selecting the best execution strategy for each query parallelizing query execution as much as possible.

- A cache is used to keep temporal copies of data to improve performance when needed. The cache can work in several modes which can be mixed for different views:
    - It is possible to materialize complete copies of the data from some views. This data can be refreshed on an incremental basis or with scheduled data pre-loads.
    - In most advanced Data Virtualization tools such as Denodo, it is also possible to have in the cache only a subset of the content of the view. In this mode, when a query is executed against the view, the Data Virtualization tool will check whether the cache contains the data required to answer the query, otherwise it will query the data source directly to obtain the required data. This way, only the most important and/or frequently used data can pre-load in the cache and/or simply cache the results of the last queries, to benefit from temporal locality.
- The Data Virtualization layer isolates business intelligence tools from the details for accessing the underlying data sources. Through the Data Virtualization layer a business intelligence tool can get access to any data available either in the data warehouse, staging area or in the production systems, enabling for instance Single Views of Customer with information about a customer coming from the CRM system, historical information about this customer retrieved from the data warehouse, billing information from the billing system, etc., so historical data as well as operational data can be provided to the business intelligence tools.
- As data access is enabled through the Data Virtualization layer, data governance can be handled in a centralized way in this layer. The data architect can trace any piece of data from data sources to final data services that are exposed to business intelligence tools. Advanced Data Virtualization tools such as Denodo offer end-to-end data lineage, data source schema refresh and change impact analysis for this purpose.
- Unified management of security: as data access is enabled through the Data Virtualization layer, security can be managed in a centralized way from this layer, granting access to users to whatever information resource they should get access to, and protecting other information resources otherwise. Advanced Data Virtualization tools such as Denodo provide a very low level of granularity with regard to user access control offering, for instance, the possibility of specifying for every column and row in a given view whether the user has a given access right to it or not, and masking the most sensitive fields depending on user privileges if needed.
- The Data Virtualization layer allows multiple sources and multiple consumers:
    - Denodo Data Virtualization allows access to any kind of information source: relational databases, multidimensional databases (e.g. SAP BW, Mondrian, etc.), packaged applications (e.g. SAP ERP), columnar databases, Web Services (REST and SOAP), mainframes (through 3270 / 4250), Web applications (through Denodo Web Automation technology), SaaS applications (e.g. salesforce.com), JMS queues, Big Data sources (e.g. Hadoop), Excel spreadsheets, text files, logs.
    - It exposes data services through multiple standard interfaces making it possible for virtually any tool to consume those services: JDBC/ODBC, Web Services (REST, SOAP), JMS, Widgets (JSR standards, Sharepoint Webparts), Java API.

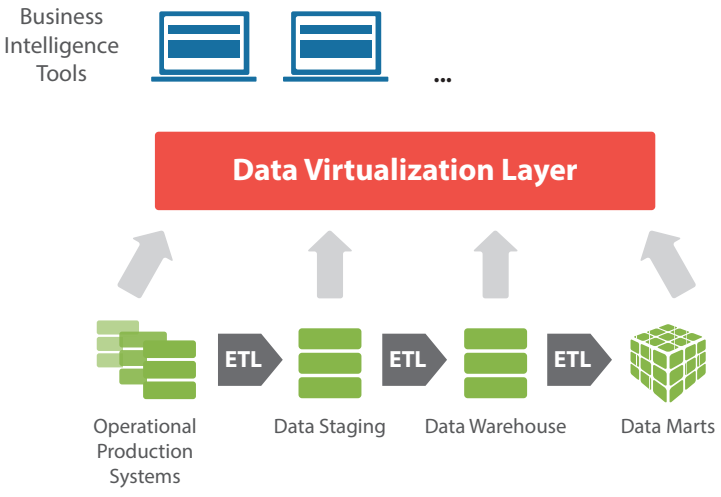


Figure 2 - **A BI/DW Architecture based on Data Virtualization**

The advantages of the Data Virtualization approach are many:

- It significantly minimises data replication, subsequently simplifying the architecture and reducing storage set-up and maintenance costs:
  - Data architects can decide whether to materialize the production data in some repository (e.g. a data warehouse or a data mart), or to get access to data in real-time directly from the operational systems. Depending upon the specific report, it might be possible to provide real-time data directly from the operational systems instead of having to materialize it in an external repository.
  - Data marts can be built virtually from either the operational systems or from the Data Warehouse (depending on the selected DW architecture). The cache can be enabled in this case for improving performance.
  - Data can be taken from any node within the BI/DW architecture (see Figure 2), so for instance, it can be provided from a staging area to the business tools directly without needing to build a Data Warehouse or a data mart. The needed data transformations in this case can be carried out in the Data Virtualization layer.
- A common data model is used by multiple business intelligence tools, avoiding the need to build complex data models in every tool, and therefore eliminating inconsistencies in data specification and minimizing the design effort as most of the needed transformation/aggregation will be done in this layer and not in the consuming business intelligence tool.
- It is easy to accommodate any change in a data source or in the final reporting requirements. When a data source changes the Data Virtualization layer will notify the designer whether to incorporate the change in the data model or not (e.g. a new column in a table), and will help the designer to know which views will be affected by such change. As a result the designer can decide whether to propagate the change or not in every affected node in the view tree. Changes are therefore handled directly by this layer and will avoid the need to modify the ETL scripts across all the data replication process which can be very cumbersome. The same applies for a change in user requirements due to a mistaken report specification from the end-user, which is a very frequent situation. The Virtual Data Model can be easily changed in this layer to accommodate the change in requirements without impacting the underlying sources and ETL scripts.
- Access to real-time data: as data can be directly retrieved from operational systems, those reports needing completely up-to-date data can benefit from this layer enabling much better operational reporting. Also it is possible to expose data views combining real-time data (e.g. current stock level) from operational systems and historical data from a data warehouse (e.g. medium stock level during last week).
- Shorter time-to-market for delivering reports to users: preparing data views to serve the needs of users is very easy to accomplish using Data Virtualization, as it is usually of process of hours (if not minutes) to prepare a proper view for a reporting tool.
- Data Virtualization enables access to any data type from semi-structured to unstructured sources, internal or external as was stated above, including recently arising sources such as Non-SQL databases (e.g. Hadoop). Including external information sources allows for a more comprehensive Business Intelligence solution as things such as competitive information, partner data, social media content, can provide significantly better insights than using internal data alone.

Data Virtualization Usage Patterns for BI/DW

As was stated above Data Virtualization helps to simplify the BI/DW architecture and therefore provides the agility and flexibility needed to deliver timely results to business users.

Among the many different Data Virtualization usage patterns that we can find in the BI/DW world we have reproduced the most typical ones hereinafter (a good information source for this purpose is [Ref.2]).

Data Virtualization Layer for BI/DW

The first scenario resembles the architecture depicted in Figure 2. A Data Virtualization layer allows access to any data within production systems, the staging area, the data warehouse and data marts.

This layer offers the flexibility that we described above. It is worthy of note that the Data Virtualization layer complements the existing infrastructure offering more agility, although this approach does not preclude that existing BI processes can still be working in the same way accessing the DW / data marts directly as they were before introducing Data Virtualization.

In a Bus Architecture the Data Virtualization layer can offer composite views of data over the different data marts, emulating the functionality of a Virtual Data Warehouse in this case.

Virtual Data Marts

Regardless of the DW architecture chosen, Hub & Spoke or Bus [Ref.1], the data virtualization layer can help reducing the number of data marts needed in the architecture, or even eliminating them completely.

In a Hub & Spoke architecture data marts can be built in the virtual layer by combining and transforming imported views from the data warehouse.

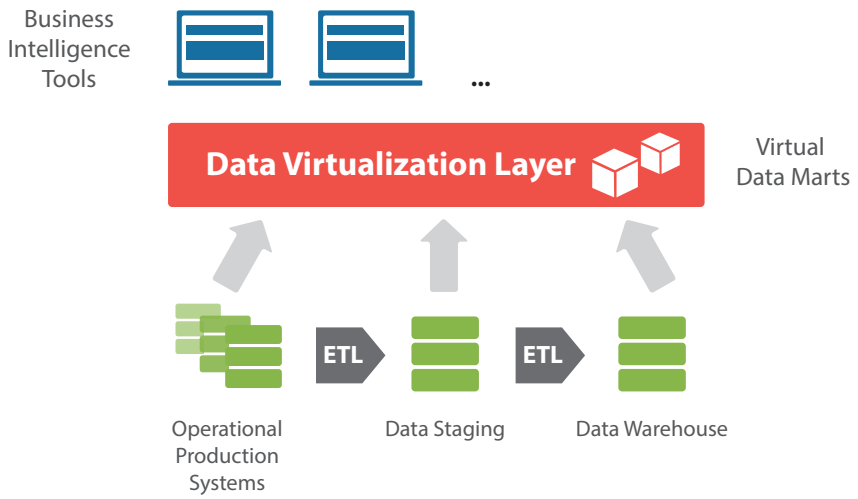


Figure 3 - **Creating virtual data marts through Data Virtualization in a Hub & Spoke Architecture**

In the Bus architecture, data marts can also be built in the virtual layer, but now combining and transforming imported views either directly from operational systems or from an intermediate persistent staging area.

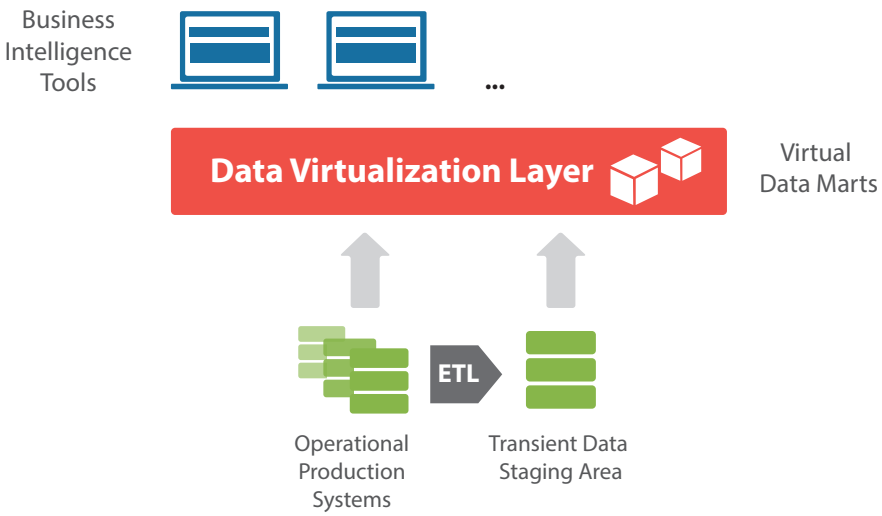


Figure 4 - **Creating virtual data marts through Data Virtualization in a Bus Architecture**

In order to improve performance, the cache in the Data Virtualization layer can be populated with pre-loaded data on a per-view basis, avoiding as a result, operational system latencies (data can be retrieved from the data virtualization cache instead) and avoiding overloading operational systems with reporting derived workload. The Data Virtualization cache plays an important role in this scenario to guarantee an appropriate performance for every kind of report.

Virtual Data Warehouse

In a scenario where we replace any intermediate storage node (i.e. staging, data warehouse, data marts, etc) and get access to information directly from operational systems through the Data Virtualization layer, a Virtual Data Warehouse functionality is offered.

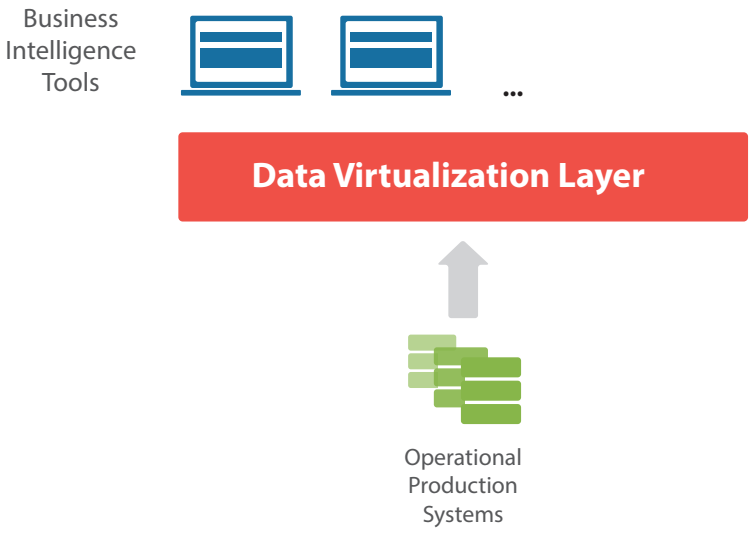


Figure 5 - **Virtual Warehouse approach**

The Cache in the data virtualization layer can be used to store any data to keep track of historical information when needed.

Data Warehouse Extension with Real-time / External data

Data Warehouse information can be enhanced with real-time data for serving operational reporting needs. The Data Virtualization layer can offer the possibility of creating data views that can combine pre-loaded information in the data warehouse with up-to-date information from operational systems.

Another way to improve the existing data warehouse functionality is making use of the Data Virtualization layer to incorporate data from other sources that otherwise would be very difficult to consider. Given the powerful capability of the Data Virtualization platform to integrate data from virtually any kind of data source, this can be used to enhance data warehouse information with external content.

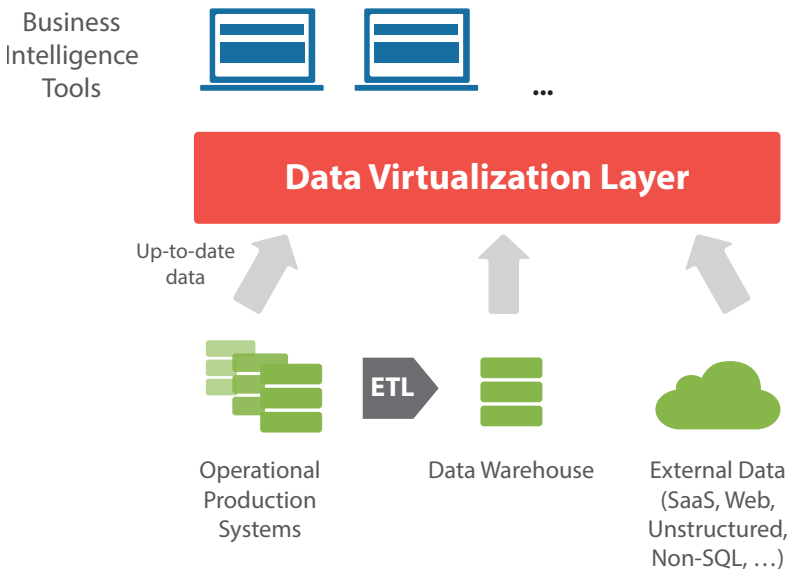


Fig. 6 - **Data Warehouse Enhancement**

Federation of Data Warehouses

A very frequent situation is federating the access to different warehouses through a Data Virtualization layer. This situation can happen, for example, as a result of a Merger & Acquisition process where the involved companies were using different DW solutions.

The Data Virtualization layer in this scenario hides the existence of different warehouses to the business intelligence tools.

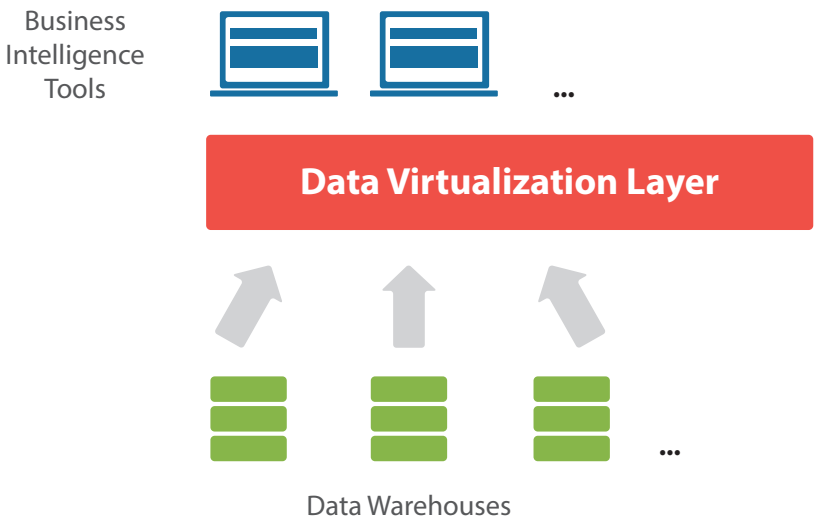


Figure 7 - **Federation of Data Warehouses**

Self-Service Reporting

The Data Virtualization layer facilitates self-service reporting. Data services can be created for the end users with all the information they might need. On top of them, end-users can run dashboards / reports which they can customize according to their needs. The BI architect pre-defines such data services in advance with the Data Virtualization tool. Adding a new data service is a very quick process as it can be easily created in a matter of minutes in the virtual data layer. The BI architect might have a library of data services that serve the most common end-user requests.

The unified security management of the Data Virtualization layer is very important in this scenario, allowing a fine-grained control over who has access to what data.

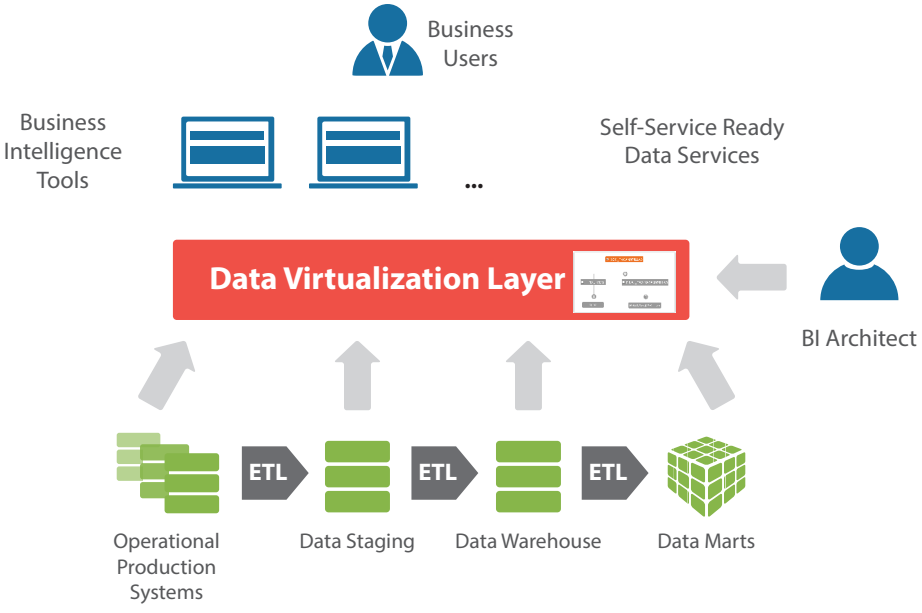


Figure 8 - **Data Virtualization enables Self-Service Reporting**

ETL / Web ETL

As was stated above the high capability of the Denodo Data Virtualization platform for integrating data from virtually any kind of data source can be used to build ETL-like processes where the data sources are not the ones typically supported by an ETL tool:

- Web Services with compound parameters (arrays, registers).
- Web applications (either internal or external).
- Non-SQL databases.

An important advantage in that regard is Denodo's ability to deal with data sources which offer an API-based query interface, such as web services (SOAP and REST), websites and packaged applications. In this case, data is typically accessed by invoking operations requiring mandatory parameters. Imagine for instance an external web service, which offers credit information about a certain customer through its tax id as a mandatory input parameter. If we had wanted to obtain such information for all of our customers, we would need to invoke such a web service once for each customer. Denodo provides transparent support for this type of invocation, including the ability to obtain the list of tax ids from any other data source in the first place, iterate through them invoking the external web service with each tax id as the input parameter in each iteration, and allowing the configuration of features such as: the desired degree of parallelism, transparent retries, and stateful executions (so that the process may be split in several runs) through the Denodo Task Scheduler, a job scheduling tool suitable for this purpose.

Another key advantage is Denodo's state-of-the-art Web Automation and Integration capability, which allows bi-directional integration (read/write) with dynamic, hidden Web applications. This is done by emulating navigation through a Website's standard, published interface, including the completing of forms, executing of mouse actions, and automated retrieval of data in a completely structured way, for subsequent combination and integration with other structured content.

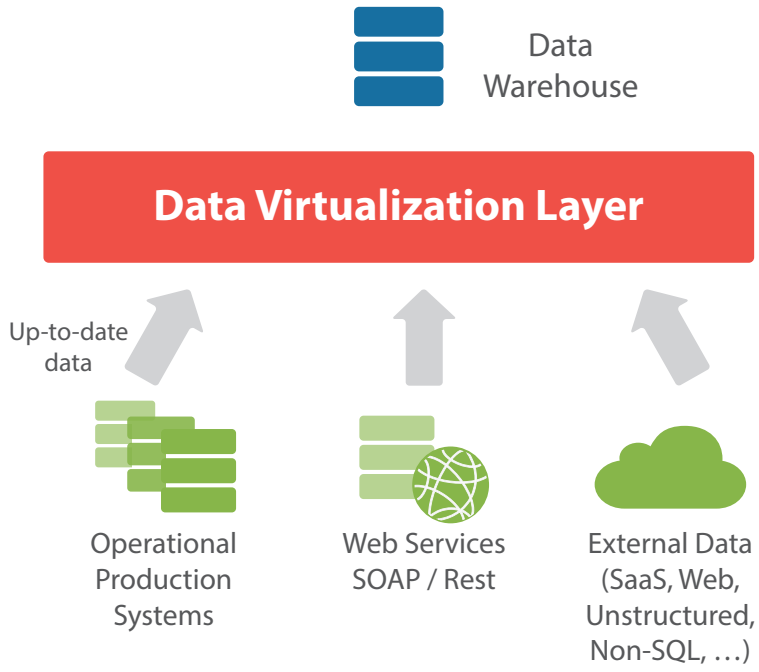


Figure 9 - **Data Virtualization for ETL / Web ETL**

Sandboxing & Prototyping

A virtual sandbox can be easily created as a separate virtual database in the Data Virtualization layer, avoiding any interference with other development or production data bases. Users can run their tests in this virtual sandbox, avoiding the need for building and maintaining another data store for this purpose.

The Data Virtualization approach fosters data-driven modelling where data views can be created in a matter of minutes and shown to the users so they can provide quick feedback about the actual accomplishment of their requirements. This avoids having to re-build a number of times complex ETL scripts in order to properly meet user requirements. This is a very frequent situation that can be avoided with the use of Data Virtualization.

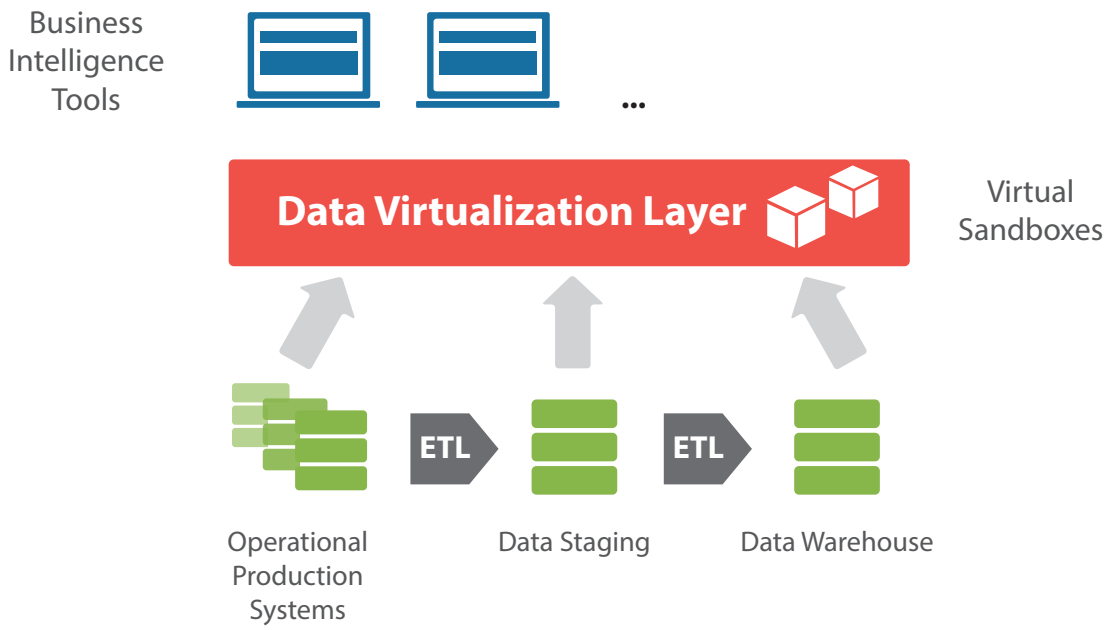


Figure 10 - **Data Virtualization for Sandboxing & Prototyping**

Conclusion

A Data Virtualization platform can greatly enhance a BI/DW architecture as has been stated and explained in this document. The use of a Data Virtualization layer brings a huge number of benefits that makes this technology well worth being considered as a key asset in the armoury of any BI professional:

- Abstraction and encapsulation.
- Significantly reduces and minimizes replication.
- Performance is guaranteed by advanced query optimization techniques and the use of caching when appropriate.
- Centralized management of security.
- Centralized data governance.
- Support for multiple-sources and multiple-consumers: integration of virtually any data source, internal or external; delivery of data services to multiple reporting tools and applications avoiding any tie to any specific technology or vendor.
- Less design effort as common data specifications are used for every reporting tool.

- Much less analysis time through the use of the prototyping and sandboxing approach.
- More flexibility to accommodate changes in data sources or in end-user requirements.
- Enables real-time data for serving operational reporting needs.
- More comprehensive Business Intelligence thanks to the integration of external content (competitive information, partner data, social media content, etc.) and also unstructured data.
- Shorter time-to-market for delivering reports to users.
- Greater agility.

References

[1] TDWI Data Warehousing Concepts and Principles: An Introduction to the Field of Data Warehousing. Chris Adamson, June 2012.

[2] Data Virtualization in Business Intelligence Architectures. Revolutionizing Data Integration for Data Warehouses. Rick F. van der Lans, 2012.