denodo

Whitepaper

Data Virtualization and ETL

Denodo Architecture Brief

A question that is frequently asked is "when should I use data virtualization and when should I use ETL tools?" Other variants of this question is "does data virtualization replace ETL?" or "I've already got ETL, why do I need data virtualization?" This Denodo architecture brief will answer these questions.

Summary

Data virtualization and ETL offer different capabilities to access, integrate, and deliver data. Although they can offer competing approaches for some specific scenarios, they are both useful additions to the data integration toolset of any organization. In general, data virtualization is more agile, flexible, versatile, and cost efficient than ETL. Therefore, a simple takeaway is not to use ETL when data virtualization is a viable approach.

However defining “viable approach” is not always as easy and clear cut as we would like - the most honest answer is “it depends on the circumstances of your particular project and your project constraints”. There are some things that ETL is very good at, and there are things where ETL is somewhat lacking. Similarly, there are many areas where data virtualization excels and there are some situations where data virtualization is not the best answer.

Extract, Transform, and Load (ETL) is a good solution for physical data consolidation projects which result in duplicating data from the original data sources into an enterprise data warehouse (EDW) or a new database. This includes:

- ETL tools that are designed to bulk copy very large data sets, comprising millions of rows, from large structured data sources.
- Creating historical records of data, e.g. snapshots at a particular time, to analyze how the data set changes over time
- Performing complex, multi-pass data transformation and cleansing operations, and bulk loading the data into a target data store.

However, applications using the data in the resultant data store are working with data that is as old as the last ETL operation. This might be data from yesterday’s end-of-day ETL operation. This may be acceptable for applications performing data mining or historical analysis in support of strategic planning or long term performance management, but is less acceptable for operational decision support applications, managing inventory levels or applications that need to use intraday updates to the data.

Data virtualization, on the other hand, abstracts, federates, and publishes a wide variety of data sources to consuming applications in an array of different formats. In doing so, it simplifies and accelerates the process of accessing, combining, and utilizing these disparate data sources and hides the complexity of the different data sources from the consuming applications. Data virtualization focuses on creating a unified common data model across diverse data sources rather than highly efficient data movement from data source to data warehouse.
The data virtualization platform creates logical views of the underlying data sources. These data sources can be structured, semi-structured (such data on the Internet, social media, pdf files, delimited files, etc.), or unstructured data in emails or documents – the complexity of accessing this data is hidden from the consuming applications. The logical views can be combined and enriched to create ‘business’ views (also called ‘derived views’), which can then be exposed as data services to consuming applications many different formats; SQL, Web Services (SOAP/XML and RESTful), Portlets, SharePoint Web Parts, JMS message queues, and so on.

Data virtualization is also different from ETL in that, with data virtualization, the data remains in place in the original data sources. When applications request data by querying a data service, the underlying data sources are queried in real time by the virtualization platform and the results are aggregated and combined before being returned to the application. This is a very important difference between the two solutions – data virtualization leaves the source data where it is and delegates the queries down to the source systems while ETL copies the data from the source system and stores it in a duplicate data store.

From a BI perspective, a data virtualization layer creates a flexible, easy to modify and manage ‘logical data warehouse’ on top of the existing data warehouse, databases and other data sources. From an application development and operational perspective, a data virtualization layer can be seen as a very versatile ‘shared data services layer’ that decouples the physical infrastructure that retains the data from the consuming applications and, in doing so, substantially simplifies data provisioning in IT departments.

Providing virtual views on to the underlying data sources also allows data virtualization to be more agile when adding new data sources and changing the underlying logical data model. Data virtualization supports rapid development iterations with value being added to the solution in each iteration (e.g. every week or two weeks). This is very different from the typical ETL/EDW project that takes many months of upfront planning and data modeling before any data can be consolidated in a data warehouse and, after deployment, it is very hard to change.

The **Table 1** provides a summary comparison of the differences between typical ETL/EDW projects and data virtualization projects.

Category	ETL/EDW	Data Virtualization
Time to value	Long term project, typically 12 months or longer.	Need a solution in production in days to weeks, and then rapidly iterate to increase benefits to organization. Data models are dynamic and executable.
Project cost	$$$$$	$$
Data models stability	Requirements are very well defined and not predicted to change significantly after deployment or not to change at all.	Requirements are well understood but expected to evolve after deployment in ways that will require modifications.
Replication constraints	There are no policies or regulations governing replication. Cost of creating and managing data source replicas is not a limitation.	Privacy regulations, internal policies, or security concerns don't allow physical replication and consolidation of data. Replication minimization is a business goal.
Source data availability	Source systems are often offline, and/or the network connection between them is not reliable, making direct access to the data sources difficult.	Source systems are generally available, and the network is reliable. Direct access to the data sources is usually available. Note that caching by the data virtualization platform can alleviate source unavailability.
Source system load	Source Systems do not have capacity for additional load during 'business hours' resulting in need for 'out of hours' processing.	Source systems can handle additional controlled access. Caching (and sophisticated cache management policies) can reduce any additional load on the source systems.
Data cleansing	Complex multi pass data cleansing (e.g., matching, de-duping, conflict resolution with human intervention).	Usually single pass data cleansing operations (e.g., enrichment, normalization, redaction, etc.). APIs support integration to external data cleansing and data quality tools.
Data transformation	Complex and multi-step, often requiring workflow engines and/or a BPM pipelines.	Complex transformations, including hierarchical data structures (e.g. XML). API for integration to external transformation systems.

Category	ETL/EDW	Data Virtualization
Application uses	Heavy analytical BI – Historical analysis and/or data mining to facilitate strategic planning and long-term performance management.	Mixed of informational and operational uses. Tactical decision-making visibility into operational data (e.g., current risk, inventory levels, device status). Also operational applications with moderate transactional requirements.
Data formats	Limited to structured data.	Wide range of data source connectivity: Structured (RDBMS, XLS), Semi-Structured (File systems, PDF, Web, etc.), and Unstructured (email, documents, etc.)
Data freshness	End of day and/or End of last load	Near real-time (“right time data”)
Data volume	Each query needs to read and process a very large amount of data (millions of rows)	Queries can range from simple SELECT statements on single views all the way to complex queries returning result sets containing millions of rows
Data consumption	JDBC, ODBC, MDX	JDBC, ODBC, Web Services (SOAP/XML and REST) , Portlets, SharePoint Web Parts, HTML, etc.

Table 1 - **Comparison of ETL/EDW vs. data virtualization**

The reality is that, while the two solutions are different, data virtualization and ETL are often complementary technologies. Data virtualization can extend and enhance ETL/EDW deployments in many ways, for example:

- Extending existing data warehouses with new data sources
- Federating multiple data warehouses
- Acting as a virtual data source to augment an ETL process
- Isolating applications from changes to the underlying data sources (e.g. migrating a data warehouse)

The examples of how data virtualization can enhance existing ETL/EDW deployment are detailed later in this document.

Data Virtualization

Data virtualization makes data spread across physically distinct systems appear as a set of tables in a local database – a virtual view. The Denodo data virtualization platform consumes virtually any type of data, including SQL, MDX, CML, Web Services (REST and SOAP/XML), flat files, and unstructured data in Hadoop and NoSQL databases, and publishes the data as SQL tables or Web Services. When users submit a query, the data virtualization platform calculates the optimal way to fetch and join the data on remote heterogeneous systems. It then queries the relevant data, performs that necessary joins and transformations, and delivers the results to users – all on the fly without the users knowing about the true location of the data or the mechanisms required to access and merge it.

A couple of things are very different from the ETL/EDW data consolidation approach described below. Firstly, the Denodo Data Virtualization Platform can consume almost any type of data – structured, semi-structured, and unstructured data can all be consumed and federated by the platform. Secondly, the data stays in place. Data virtualization is not a data consolidation strategy. It leaves the data in its original systems and accesses them to retrieve the appropriate result set in response to user queries. The Denodo Data Virtualization Platform can utilize sophisticated data caching to alleviate any performance issues. For example, caching can be used to compensate for slower data sources (e.g. Web Services) or to minimize the impact on load sensitive production databases. Denodo's caching policies provide complete control over the data cache so that it can be fine-tuned for specific scenarios.

The fact the data virtualization does not move the data into a consolidated data warehouse, with a common enterprise data model, makes it much easier and quicker to make changes to both the data sources (e.g. add a new data source) or the exposed data e.g. creating an extended view from an existing view without disrupting the applications using the existing data view. This agility and flexibility results in shorter, more iterative development cycles and quicker time to value for a data virtualization project.

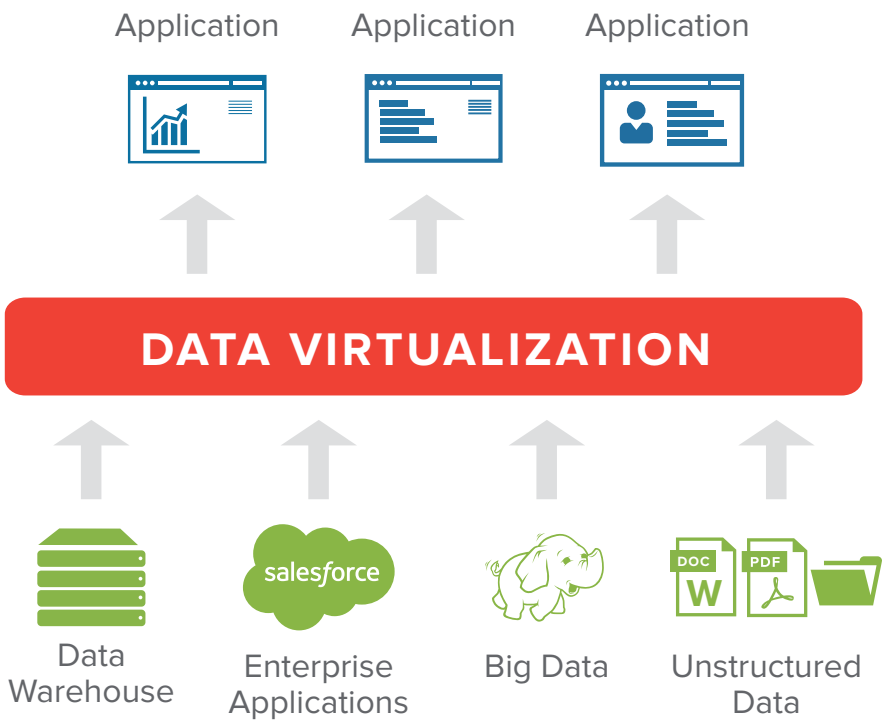


Figure 1 - **Data virtualization**

What Is Data Virtualization Good For?

Data virtualization is an excellent solution, and many times the only option in scenarios such as:

- Structured, semi-structured, and unstructured data from disparate data sources need to be combined and queried. The target applications consuming applications are isolated from the intricacies of accessing and formatting this disparate data as the data virtualization platform presents it via standard interfaces, such as SQL, Web Services (REST and SOAP/XML), and so on.

- Data needs to be accessed and delivered in real time. Data virtualization also provides near real-time ('right time') access to the underlying data. This is very important for decision support applications, such as those managing inventory levels or providing intraday portfolio risk analysis. In these applications, using stale data is not acceptable. Additionally, accessing the data sources through the data virtualization layer means that there are no data duplication and synchronization issues. The data source stays where it is and only the results are transferred to the consuming application. This is important in highly regulated environments where data duplication can be perceived as a security and privacy risk.

- The requirements to expose business data entities demand multiple diverse consuming formats that decouples applications from data sources.

The Denodo Platform can also support data transformations and data cleansing. The data transformations can be complex and can include hierarchical data structures, such as XML. Integration with external transformation tools are also supported via an API. The Denodo platform can also perform data cleansing and quality operations, such as data enrichment, value normalization via mapping tables, data redaction in support of privacy requirements, and so on. As the Denodo platform is extensible, custom data quality routines can be created and added to the platform or the platform APIs can be used to invoke external data cleansing and data quality tools.

Applications Benefiting from Data Virtualization

A wide variety of applications can benefit from the agility and flexibility provided by data virtualization – even some that are traditionally thought of as ETL/EDW type applications. Obvious applications include those requiring real time (or near real time) access to the most up-to-date data available. These applications cannot use the 'last available' data provided by ETL/EDW systems – they need current data and they need it now! Examples of these applications are operational decision support systems, such as inventory control, risk management, and so on.

Applications that are subject to changing requirements and the addition of new data sources – including unstructured data sources that are typically not handled by traditional data tools – are also good candidates for data virtualization. The flexibility and agility provided by the data virtualization platform to easily connect to new data sources of all types and to combine these new sources into the existing data views enables a rapid iteration process which allows the development team to react quickly to new data demands from the business.

BI and analytics applications that are traditionally the preserve of data warehouse deployments can also use data virtualization to extend the types of data being analyzed to include unstructured data sources typically not supported by data warehouses. Examples of this include pulling social media data from the web to analyze influencer behavior for consumer buying patterns. This combines the normal transaction data (purchases) with social media influencer or influenced data pulled from, for example, Twitter streams or Facebook posts.

Finally, new web and mobile applications that need to access corporate data sources are ideal candidates for data virtualization. Typically these applications need to be isolated from the underlying data schemas and traditional access methods such SQL queries. Mobile applications especially are more likely to use REST Web Services to access any data from the corporate data stores and the ability of the data virtualization platform to expose the underlying data as REST Web Services makes them better suited for newer web and mobile applications.

Typical projects where data virtualization is a 'must' includes:

- **BUILDING A LOGICAL DATA WAREHOUSE**
  This can be augmenting and enhancing an existing data warehouse deployment – whether by extending an existing data warehouse with new data sources or by federating multiple data warehouses – or it can be creating a 'virtual' (or logical) data warehouse from existing data sources (and avoiding the time and expense of creating a physical data warehouse). In either scenario, augmenting existing data warehouse deployments or building a logical data warehouse, data virtualization is a critical piece of the solution for connecting, combining, and delivering the data in a consumable format for the dependent applications.

- **BIG DATA INITIATIVES**
  Big Data initiatives are everywhere – in fact, it almost seems that if your organization is not investing in Big Data then you are getting left behind. However, Big Data cannot, and should not, existing in isolation of the rest of the data architecture. Once you have assembled your datasets and run them through the Map/Reduce engine (using Hadoop, Amazon EMR, Cloudera, and so on), what then? How do you take those results and integrate them with you 'mainstream' data residing in your databases, data warehouses, and enterprise applications? This is where data virtualization come in – allowing you to integrate the data with the information pulled from your CRM, from your DSR, from your data warehouse, and so on. Not only that, data virtualization can also pull in and combine the Big Data with unstructured data sources, such as social media (Twitter, Facebook), web logs, etc.

  Without this ability to integrate the data coming from Hadoop and other Big Data systems, you end up with data silos – one for the business and one for the data scientists – and you won't be able to realize the full potential of Big Data.

- **PRACTICAL MDM**
  MDM projects are complex and costly and many fail to deliver their expected value because they are too ambitious is scope and create too much change and uncertainty throughout the data infrastructure environment. MDM vendors are trying to overcome these drawbacks by moving towards a 'registry' MDM solutions rather than the more traditional repository style MDM solutions (in which the data is replicated and cleansed in a central MDM repository). However, data virtualization provides flexibility and time-to-value for any MDM project, whether you are using an MDM tool or not. For projects not using an MDM tool, a data virtualization layer allows you to create a virtual MDM repository by pulling the 'master data' from the relevant source systems (e.g. to create a composite master view of a customer from multiple sources). This virtual master data can then be used by consuming applications to give a single consistent view of the data entity (e.g. customer). It is also important to note that the data virtualization platform can quickly adapt to new 'master' sources of data. For example, if you find that your CRM data for customer contacts is getting stale, you can use the data virtualization platform to access new data sources (e.g. social media) to refresh the virtual master data from this new source. (Of course, you can feed this data back to your other data sources if you need to update their data). Alternatively, if you already have an MDM solution in place, you can extend and enrich the data from the MDM solution by using the data virtualization layer to access other data sources, such as unstructured data from social media and the web.

- **ENTERPRISE INFORMATION GOVERNANCE INITIATIVES**
  Information governance is increasingly important to organizations, especially as regulations control the access and usage of private and confidential data – with significant penalties for organization and people who breach these regulations. However, a key to effective information governance is understanding who is accessing the underlying data and how they are using it. Without this basic knowledge, governance and control of the information is impossible and, if users and applications are accessing the data source directly, it is often a difficult task to determine who has access to the data and when and how they use it.

  A data virtualization platform acts as an abstraction layer between the data sources and the consumers. Consumers of the data connect to the data virtualization platform rather than the data sources directly and this provides a point of control for monitoring and enforcing data access policies. These policies allow you to control who accesses certain data, how and when they access it (e.g. office hours and from office locations), and how they can use the data. The data virtualization platform also allows you to configure different views of the same data for different users or user roles. This can be by simply not allowing access to certain data elements or by redacting parts of the data (e.g. all but the last four digits on a social security number). With the data virtualization layer providing a control point for monitoring and managing access to the underlying data sources, it is easier for organizations to implement information governance programs to bring them into compliance with industry regulations.

The data virtualization platform can also provide additional benefits such as data lineage reporting, so that it is easy to determine the originating source of information in case of errors or other problems at the consuming end. The ability to trace the information back to its data source and see how it was modified or manipulated between the source and consumer is invaluable. The data virtualization platform can also provide a reverse view of this lineage, showing where data is consumed i.e. going from the source to the consumers. This is important for impact analysis when planning a change to a data source. It is easy to see what consuming applications are affected by changes to a data source and, therefore, to plan for these impacts.

Overall, data virtualization can be used in many scenarios and its ubiquity and benefits make it a critical component within any enterprise data architecture.

Planning Your Data Virtualization Project

Given the flexibility of adding new data sources to the data virtualization platform at any time, it is very easily to incrementally grow the size and scope of data virtualization projects – unlike the ETL/EDW projects that require considerably upfront cost and effort to create the common enterprise data model before any ETL processes can be defined to consolidate the data in the data warehouse. With data virtualization, projects can start as a pilot project to easily and rapidly prove the value and agility of the solution to the business. The scope of the project can grow as more data sources are added and more applications use the data virtualization layer for accessing the data. As value is added at each stage of the project, it is easier to justify extending the use of data virtualization to new projects.

This incremental approach to data virtualization projects minimizes the risk inherent in all data integration projects. This, in turn, makes it easier for the business to buy-in to the plans, as real value is delivered back to the business before the next stage of the project is funded. Compared to ETL/EDW projects, the upfront costs and efforts are limited by the scope of the pilot project and grow incrementally as the scope of the project grows.

ETL (Extract, Transform, and Load)

ETL is designed to process very large amounts of data as it bulk copies complete data sets from the source systems, translates and, often, cleanses the data to improve its quality, and loads to resultant data set into a target data warehouse.

ETL is a critical part of a physical data consolidation strategy which replicates data in an enterprise data warehouse. Data replication might not be an acceptable solution in highly regulated scenarios where governance and privacy policies restrict or prevent data duplication in an attempt to control and manage data access.

Consolidating data from multiple sources into a shared data warehouse requires that any source data is mapped to the schema of the target data warehouse. This in itself necessitates the upfront modeling of the different data sources into a common enterprise data model – a process that is usually a long and taxing as it involves multiple stakeholders and data owners, each with their own perspective on the final data model.

However, ETL does provide benefits to the implementing organization and provides a suitable platform for building a single source (albeit is a replicated source) of data to the organization. Additionally, it improves productivity by supporting the reuse of data objects and transformations. It imposes a strict methodology on the management of the data assets and greatly improves metadata management, including better impact analysis of changes in the underlying data source schemas.

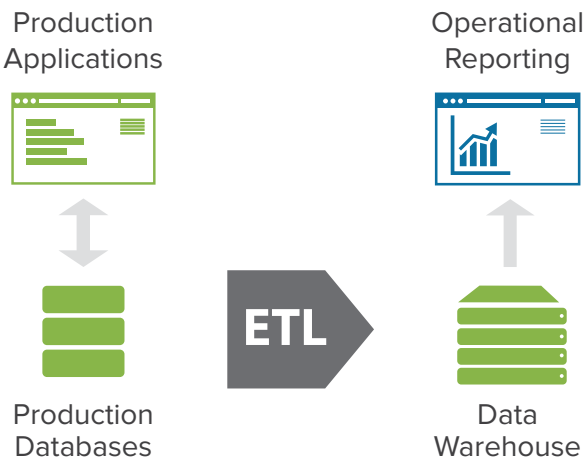


Figure 2 - **Typical ETL scenario**

What Is ETL Good For?

ETL is an excellent tool for the bulk copying of complete data sets and transforming them to a predefined enterprise data model (EDM). It is design and optimized to handle very large datasets with millions (or even billions) of rows. Typically ETL handles very structured data sets, such as relational databases, or XML. However, ETL does not handle semi-structured or unstructured data very well, if at all.

If the data needs to be transformed prior to loading in the target data warehouse, ETL process support complex data transformations. These can be complex multi-pass transformations that even require a rules engine or workflow process to orchestrate the transformation. Data can also be staged by the ETL process to support the multiple passes over the data. Similarly, ETL supports complex cleansing operations, including value matching, de-duplication, conflict resolution with human intervention, etc. These transformation and data cleansing capabilities usually result in better quality data in the target data warehouse.

Applications Benefiting from ETL

ETL (and the resulting data warehouse) is best suited for applications that require access to the complete consolidated data set. For example, historical trend analysis, data mining operations, etc. These applications need to process (not just access) complete data sets to perform their analysis.

Applications that can use 'old' data – for example, yesterday's 'end of day' data set – are also candidates for ETL/EDW. Examples are end of day sales reconciliation processes in retail, or close of business updates on portfolio positions in finance. These applications can use the 'last available' data provided by ETL operations.

Finally, applications that are 'read only' are candidates for ETL/EDW. ETL is typically in a single direction i.e. from the original source to the data warehouse. This provides read only access to the consolidated data in the data warehouse, but any updates to the data (if even permitted) are typically not transferred back to the original source. Data updates need to be made at the source so that subsequent ETL processes (or any Change Data Capture processes) will pick up the changes and replicate them to the consolidated data warehouse.

Planning Your ETL Project

ETL/EDW projects are typically very large enterprise-wide projects with all of the encumbrances that these types of projects entail. They are usually of long duration – 12 months and often much longer – and are very expensive, both in terms of the cost of the tools and resources necessary to deliver the project.

ETL projects require significant upfront investment in building the common enterprise data model – a task that involves getting agreement from all of the data owners and stakeholders of the target applications. Often, getting agreement on the common enterprise data model is the most difficult part of the whole project. These enterprise data modeling activities are notorious for resulting in a lot of 'churn' with little actual progress being made. It requires strong enterprise data architects and project managers to bring these to a successful conclusion.

The need for the upfront data modeling also means that the originating data sources need to be stable and well documented. The ETL transformation and data cleansing processes will be defined based on the known source data schemas – if these are not well understood or are subject to change, this will result in reworking these processes as testing activities start.

Extending ETL/EDW with Data Virtualization

Frequently an ETL/EDW system already exists and there is a lot of pressure on new projects to make use of this existing infrastructure. The organization might have spent a lot of money and time to set up the ETL/EDW infrastructure and projects are pushed to make use of these tools to justify the investment. These results in projects that don't fit into the classic ETL/EDW use case being force fitted into the infrastructure, which usually results in overly complex designs that fail to meet the user's expectations. In these circumstances, a better solution would be to extend and enhance the existing system capabilities with data virtualization. The following are examples of how data virtualization can coexist and enhance existing ETL/EDW systems. This is not intended to an exhaustive list of the ways in which ETL/EDW can be enhanced with data virtualization – it simply provides some examples of these types of scenarios.

Extending Existing Data Warehouses with New Data Sources

In situations where a data warehouse already exists within the organization, but the business users need to add new data sources to enhance their reporting or analytics. In this situation, a data virtualization platform is layered over the data warehouse and also connects to the new data source(s). The reporting tools use the data virtualization platform as the data source for their reporting needs, effectively including the existing data (in the data warehouse) with the data from the new data sources. This scenario is illustrated in **Figure 3.**

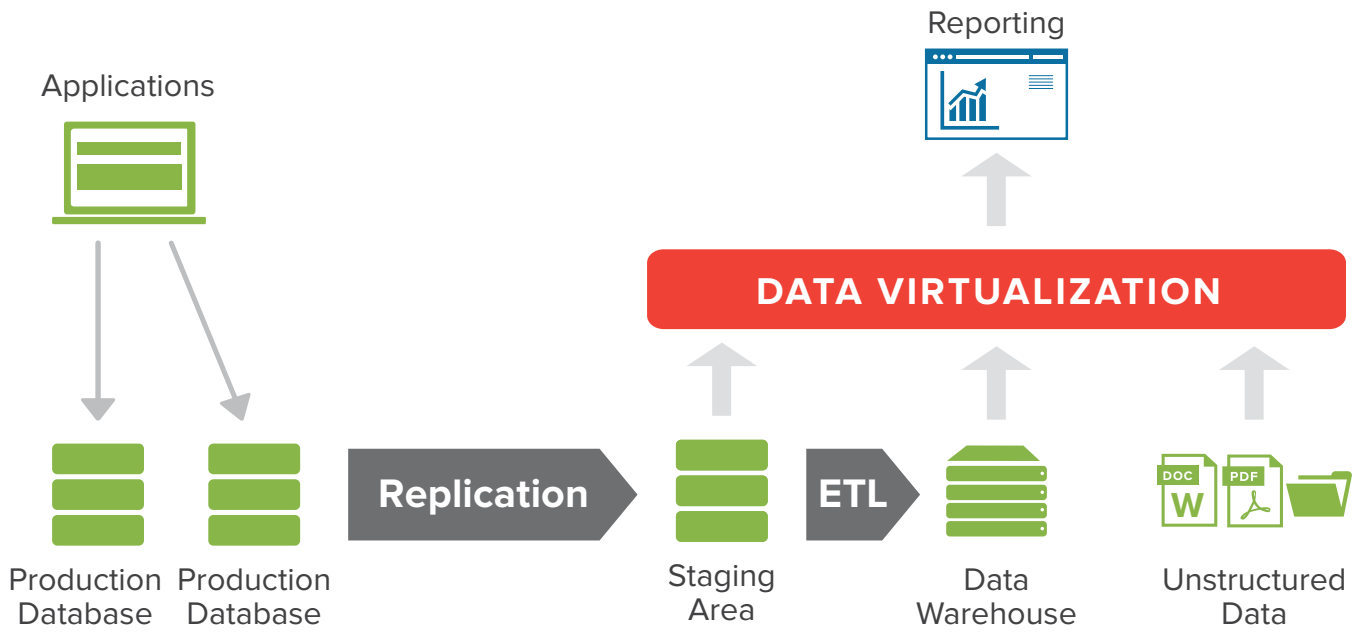


Figure 3 - **Extending an existing data warehouse with new data sources**

Federating Multiple Data Warehouses

In a situation where there are multiple existing data warehouses within an organization – for example, regional data warehouses – and the business needs a single view of the data, a data virtualization platform can be used to provide this single view. This scenario is illustrated in **Figure 4**.

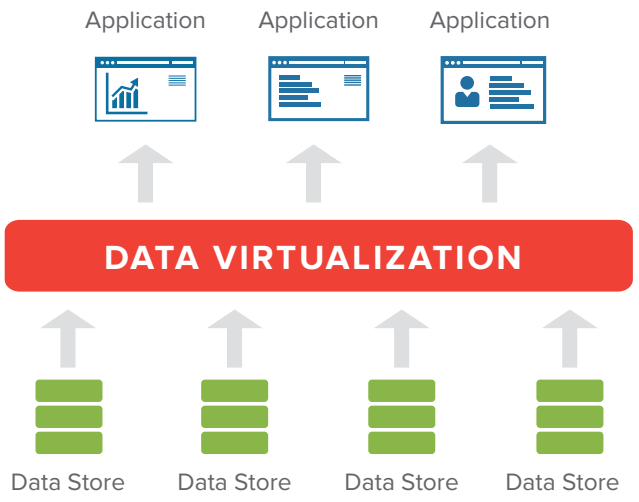


Figure 4 - **Federating multiple data warehouses**

Acting as a Virtual Data Source to Augment ETL Processes

In the same way that data virtualization can provide a virtual view onto multiple data sources, it can also become a virtual data source for an ETL process. In this situation, the data virtualization platform accessing and federates the underlying data sources – including those containing unstructured data that is typically not handled by an ETL process – and delivering the data as derived views that can be access via SQL by the ETL tools.

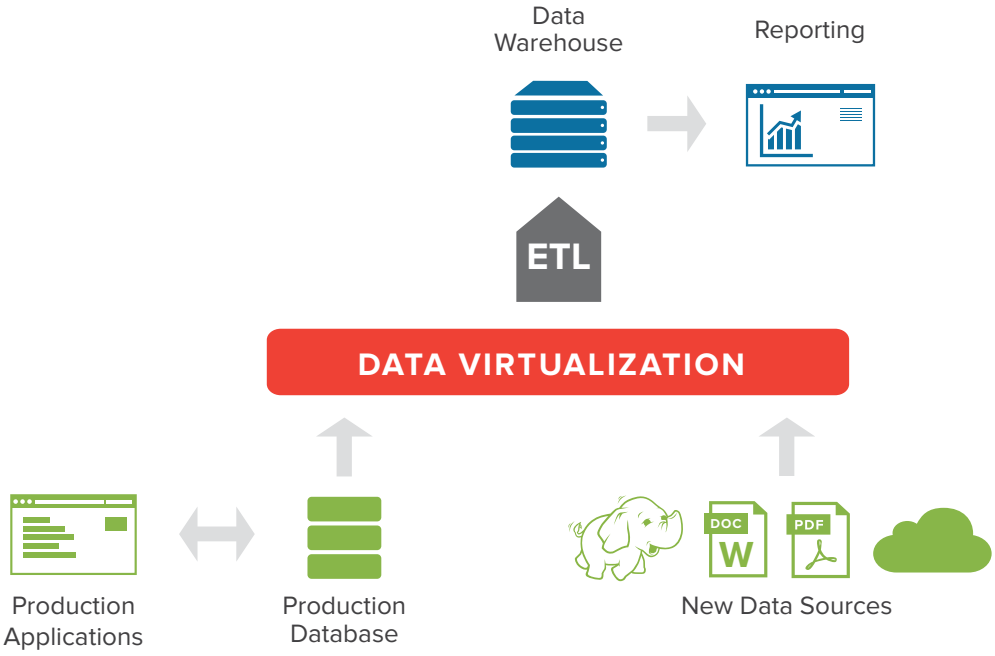


Figure 5 - **Data virtualization as virtual data source for ETL**

Isolating Applications from Changes to Underlying Data Sources

One of the core benefits of data virtualization is that is isolates the consuming application from the complexities and location of the underlying data sources. The applications do not know, nor should they care, where the data comes from and how it is retrieved. In addition to making application development quicker and easier (by hiding the vagaries of accessing different data sources), the data virtualization platform also isolates the applications from changes to the underlying data sources. If new data sources are introduced to the system, or an existing data source is changed (e.g. migrating a data warehouse), the data virtualization platform hides this from the applications. Consequently, any such changes can be quickly implemented with less impact on the rest of the application ecosystem than if the applications accessed the data sources directly.

Conclusion

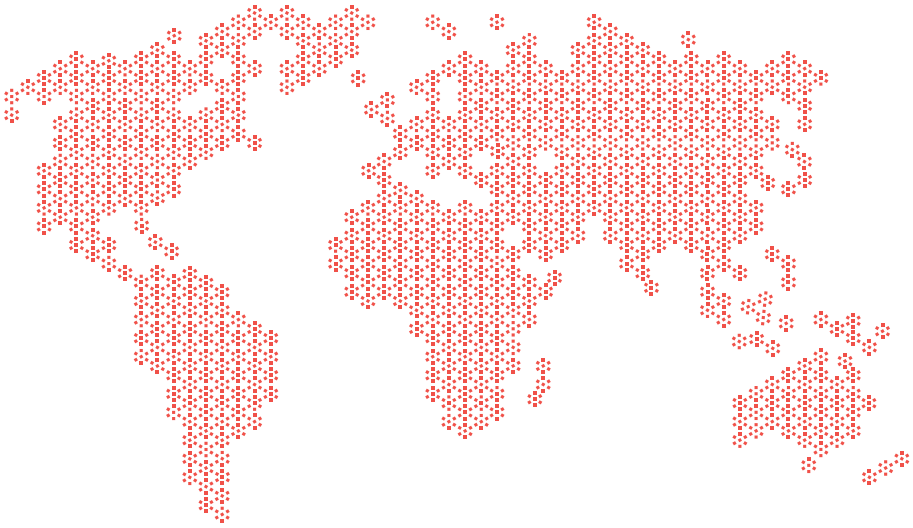
As shown in the previous sections, data virtualization plays a key role in modern enterprise integration stacks to cover some strategic needs. Factors such as exponential data growth, new data source types that create new information silos (NoSQL, Cloud), and the extensive use of Big Data require a new infrastructure that is not covered with traditional ETL solutions alone.

The answer to the original question of “when should I use data virtualization and when should I use ETL tools?” really is “it depends on your circumstances”. The comparison table above can be used to determine how project needs map to the operational strengths of both ETL/EDW and data virtualization.

As a summary:

- In many cases, especially those combining structured data with unstructured data or requiring real-time access to up-to-date data, then data virtualization is a better option.
- In some case, where it is really necessary to copy massive amounts of data for complex analytics or historical data marts with no concerns about data freshness, ETL is still the best option.
- Very often, the line is not that clear. Cost of data storage, operational costs of the solution\, and time to market can tip the balance to data virtualization, even for projects that have traditionally used ETL solutions.

Having said this, data virtualization can often be used to increase the value of existing ETL/EDW deployments by extending and enhancing the ETL/EDW tools and processes. Some typical scenarios of a hybrid solution – combining data virtualization and ETL/EDW – were described earlier in this document.



Denodo

Denodo Technologies is the leader in data virtualization – the only platform that delivers Information-as-a-Service across disparate structured, Web and unstructured sources. Innovative leaders in every industry use Denodo to dramatically increase flexibility and lower costs of data integration for agile BI and reporting, call center unified desktops, customer portals, Web and SaaS data integration, and enterprise data services. Founded in 1999, Denodo is privately held.

Denodo North America & APAC
530 Lytton Avenue,Suite 301
Palo Alto, CA 94301, USA
Phone (+1) 650 566 8833
Email info.us@denodo.com & info.apac@denodo.com

Denodo EMEA
Portland House, 17th floor, Bressenden Place
London SW1E 5RS, United Kingdom
Phone (+44) (0) 20 7869 8053
Email: info.emea@denodo.com

Denodo DACH
3rd Floor, Maximilianstraße 13
Munich, 80539, Germany
Phone (+49) (0) 89 203 006 441
Email info.emea@denodo.com

Denodo Iberia & Latin America
C/ Montalbán, 5
28014 Madrid, Spain
Tel. (+34) 912 77 58 55
Email: info.iberia@denodo.com & info.la@denodo.com

Visit www.denodo.com Email info@denodo.com twitter.com/denodo