What is Transparent Data Virtualization?

Data virtualization is a popular topic in today's world of real-time enterprise data management. The goal is to solve the problem of data sharing and data access across various platforms throughout the enterprise, without needing to know the location of the data or the underlying storage technology. With the ever-growing volumes of data being generated and collected today, the cost to store, transform, and share it throughout an organization is swelling. But the opportunity cost of not having immediate access to potentially valuable business information could be even greater. In any enterprise, data becomes locked and hidden away in various database silos across various departments and business units. For years, the standard approach has been to copy the data from silo to silo in order to provide access to these disparate data sources. The data replication and extract, transform, and load (ETL) processes implemented to support data sharing often become too costly and cumbersome to build and maintain. Enter data virtualization, where data isn't moved, but is made *virtually* accessible.

Standard Data Federation

Often used as a synonym for each other, data federation is actually a subset of data virtualization. Other types can include storage virtualization and database virtualization. Data federation software provides the ability to join and aggregate disparate data sources in a federation engine for virtual access by an application or query. While this approach does meet the overall goal of accessing various siloed data sources without using ETL, there are still challenges that remain.

One complication with *standard* data federation is that the users of the data are affected immediately once the data federation software is introduced. When a federated query is run, an intermediate federation engine performs the translation of query syntax between multiple different types of database engines. Applications, analytical reports, and power user queries need modification to use the new proprietary SQL query syntax required by the data federation service. Virtual access to data occurs without writing costly ETL, however, loads of production code must be rewritten to support the new syntax introduced by the federation engine. With data federation, the query processing remains in the same database silos, but developers and end users must update their existing codebase.

Another challenge is the potential for performance bottlenecks that might plague this form of data virtualization. First, the federation engine must translate SQL queries from disparate systems, on the fly, to join, aggregate, and return the final query results. Second, the original source of the data must handle all of the data processing within its local database engine. Often, there may not be sufficient resources available for high performance query results that are expected for near real-time enterprise data access. The federated approach requires the source database engines to take on the additional load of any new queries that are pushed down as a part of the federation service. More queries against the RDBMS can introduce added resource constraints on database servers that may already be close to CPU processing and memory usage capacity with their current workload.

“Data federation is not end to end data virtualization”

Now, what if the data federation implementation were reversed? Federation can only virtualize the *location* of the data, but the data virtualization approach must also virtualize *access* to the data. Rather than modifying existing code and processing data on existing database servers, the process can be switched around. Keep the existing code intact and unchanged, and offload the data and processing to a new, high-performance and scalable environment. This approach to data virtualization, data sharing, and data access is transparent to all end users.

Going Transparent with Data Virtualization

Transparent data virtualization uses an approach that is different from traditional data virtualization technologies. The implementation is seamless and completely transparent to the end user of the data, regardless of whether the end user is an application, analytical reporting system, or a business analyst running custom SQL queries. That means zero code changes are required to utilize transparent data virtualization with an existing application.

The key to high performance query results with transparent data virtualization is the use of a distributed storage and computation backend, such as Apache Hadoop or Spark, to store and access the data. While sales demos for other standard virtualization technologies work very well while retrieving small datasets, large queries which join big datasets across various disparate sources will eventually grind to a halt. The solution is to store all, or most, required data in the centralized data lake - the virtualization backend. This allows the query heavy-lifting to be transparently pushed down into a single, scalable computation platform, while freeing up resources on the relational database server.

The centralized data lake can contain datasets from many different sources, including relational databases, IoT devices, streaming operation output, and even the results of machine learning and ETL processes. Proprietary database engines are no longer required to for data access. Once centralized, it is stored in open data formats, therefore the data can be queried directly or virtually accessed from any other database using transparent data virtualization. As soon as the data lands in the data lake tables, it is immediately available for virtual query from any database application, in real-time.

	Standard Data Virtualization	Transparent Data Virtualization
Data storage location	Silo	Silo or data lake (centralized)
Data access	New proprietary query engine	Existing platforms or direct access
Application rewrite required	Yes	No
Data processing location	Silo	Data lake
Transparent pushdown of queries	No	Yes

Ultimately, transparent data virtualization is the reverse of data federation. Using a centralized data store allows data access to remain transparent, while the data storage location and query processing engine changes automatically. The centralized data lake is a cornerstone for enterprise data sharing and is the foundation for an enterprise data fabric solution.

Gluent's Transparent Data Virtualization

At Gluent, we believe that transparent data virtualization is the best approach for enterprise data sharing. That is why Gluent's data virtualization technology, Gluent Data Platform, was designed to be transparent from the ground up. It virtualizes relational database tables so that enterprise applications can continue running on their existing databases unchanged, with no re-platforming or code rewrites, while most of the data resides in the Cloud or Hadoop. Gluent will push down queries so the heavy-lifting is performed by these modern, distributed compute backends, freeing up resources on the relational database.

Unlike other data virtualization products on the market, Gluent Data Platform will automatically sync relational database management system (RDBMS) data to Hadoop without users having to build any ETL code or data pipelines. A single command will sync all or a portion of an RDBMS table data to the centralized data store, place the data in a columnar, compressed format, create the Hadoop table, automatically partition and index the data, and prepare the data for querying from the RDBMS. Gluent can even track changes and perform incremental updates from the RDBMS to the data lake. Thanks to data synchronization and transparent data virtualization, transparent data access becomes possible with Gluent Data Platform. Virtualizing access to the data to make it truly transparent to the end user is a key differentiator between transparent data virtualization and standard data virtualization.