FOREWORD

NUMBER ONE
Understand what data federation is.

NUMBER TWO
Understand the business value of data federation.

NUMBER THREE
Understand what data federation doesn’t do.

NUMBER FOUR
Understand data federation use cases for data warehousing.

NUMBER FIVE
Understand data federation use cases for business intelligence.

NUMBER SIX
Understand additional data federation applications.

NUMBER SEVEN
Evaluate data federation middleware.

ABOUT OUR SPONSOR

ABOUT THE AUTHOR

ABOUT TDWI RESEARCH
FOREWORD

Data federation is an important tool in today’s data integration portfolio. Data and application architects use the middleware to query and join data from multiple sources on the fly and deliver the results to data-hungry decision makers. It makes a lot of sense to use data federation tools when it takes too long or costs too much to create a persistent store of consolidated data, such as a data warehouse or data mart.

Data federation is not a new technique. The notion of virtualizing multiple back-end data sources has been around for decades, always playing a complementary role to more traditional batch-processing-based data integration techniques. In the early 1990s, when data warehousing was gaining prominence, data federation tools were ambitiously touted as a way to create “virtual” data warehouses. But by the early 2000s, with more powerful computing resources, the technology was positioned as a general-purpose data integration tool, adopting the moniker “enterprise information integration” or Eii. The three-letter acronym was designed to mirror and complement ETL—extract, transform, and load—which is the predominant method for integrating data in data warehouses.

Today, data federation tools have broadened their capabilities and appeal and go by many labels, including data virtualization, data services, and distributed query. They are used in a variety of situations that require unified access to data in multiple systems via high-performance distributed queries, such as data warehousing, reporting, dashboards, mashups, portals, master data management, data services in a service-oriented architecture (SOA), post-acquisition systems integration, and cloud computing.

This Checklist Report will help you understand when and how to use data federation tools to deliver optimal solutions.

NUMBER ONE

UNDERSTAND WHAT DATA FEDERATION IS.

Data federation integrates data from multiple, disparate sources inside or outside the organization on the fly. It offers business users and application developers a single, easy-to-use interface to access heterogeneous sources, making remote data appear as if it resides in a single local database. When users submit a query, data federation software calculates behind the scenes the optimal way to fetch and join the remote data and return the result. Its ability to shield users and application developers from the complexities of distributed SQL query calls and back-end data sources is why some vendors call this technology “data virtualization” software.

Data federation software consists of four basic components:

1. **Data discovery tools** that assist developers in understanding the structure and contents of remote data sources.
2. **Business modeling or “abstraction” tools** that create business-oriented models of data that reside on different systems in various locations.
3. **A distributed query optimizer and execution engine** that calculates the most efficient way to join remote data sets, perform needed transformations, and deliver the results back to the user or application.
4. **Data adapters** that provide native (and thus optimized) access to major databases and applications.
Reduces Time, Cost, and Risk. Data federation software is designed for query-based applications that require current data from multiple systems. It’s ideal when the business needs a solution fast, doesn’t have a sizable budget for infrastructure and staffing, and wants to minimize the risk involved in deploying a new solution.

The traditional way to build query-based applications is to create a data warehouse or data mart. This practice off-loads queries from core transaction systems, which can’t suffer performance degradation from an unpredictable query workload. It also enables data architects to pre-integrate and model the data in a form optimized for query processing.

However, creating a new data mart—or sometimes even extending an existing one—takes at least three months. The process involves understanding user requirements, creating target data models, and building and testing ETL transformations as well as purchasing, deploying, and testing server hardware and database software. This takes significant staff time and capital to get the solution up and running. And what happens if the design fails to meet user expectations? Or if users change their requirements and want something different?

Fast Deployment. Data federation can minimize the risks, costs, and time needed to deliver query-based solutions because it doesn’t require a lot of upfront coding and doesn’t need an additional database to store source data. All you do is install the data federation development and runtime software on an industry standard server, create the views and services that form the global semantic layer, and tune the major queries. There are no ETL programs to develop or staging areas, data warehouses, or data marts to instantiate.

Rapid Prototyping. With quicker deployments, you can get user feedback earlier in the process. If users change their minds or want a different view, a developer rewrites the query (often entirely via a point-and-click interface) and generates a new data set. By querying data instead of encoding it in a database, data federation tools minimize the risk, cost, and time involved in deploying many types of applications.

Ideal Scenario. Data federation software is ideal when source systems are consistently available and have enough capacity to handle streams of ad hoc queries without slowing down transaction processing tasks. The intervening network should be robust and not prone to outages or delays, and individual queries shouldn’t request large volumes of data. Also, it’s best if source data doesn’t require significant transformation or cleansing, and if the business application consumes mostly current data in relational or XML formats.

Unreliable, Overloaded Systems and Networks. Conversely, data federation isn’t appropriate when source systems are frequently offline or overloaded with existing workloads, and the network connection between systems isn’t reliable. In this case, it’s better to deploy a physical architecture that moves data in a batch process to a data warehouse or data mart to support the business applications.

Complex Transformations. Data federation also isn’t advisable if the source data requires multiple passes or steps for cleansing or transformation. Data federation software can apply nominal transformations on the fly using SQL (e.g., transform a NUM to a VARCHAR or validate against a cross-reference table), but it doesn’t perform the multi-step transformations needed to do matching, de-duping, conflict resolution, denormalization, rollups, or dimensional calculations. Consequently, data federation won’t deliver data in a format usable for OLAP cubes or data mining tools. These types of transformations are best done with ETL or data cleansing tools.

Very Large Result Sets. Although data federation tools impose only about 10 to 20% overhead on queries, it’s not ideal to use them to analyze millions of rows of historical data on a single query that traverses more than one system. The software must parse the query, analyze the data in both systems, copy and move the smaller table to the system with the bigger table, perform the join, and return the result.

With caching, 64-bit operating systems, and higher-bandwidth networks, the ceiling on query size is always rising. However, the best bet is to use data federation to support short, tactical queries...
on current data rather than strategic queries against large volumes of historic data. You want to avoid response time delays that business users might notice.

The bottom line is that data federation tools depend on the availability, reliability, and processing capacity of source systems and the intervening network. Selecting applications with the ideal query footprint given the technical constraints of your environment is the best way to guarantee the success of a data federation deployment.

Data federation is an ideal way to augment existing data warehousing environments. If you haven’t deployed data federation software, your environment probably lacks the flexibility it needs to meet ever-changing business demands. Here are four use cases for data warehousing:

1. **Extend a data warehouse with additional data sources.** Every data warehousing manager knows that it’s impossible to get all the data that users want into a data warehouse. Often, it’s difficult to keep up with user requests for new data in a timely manner. But sometimes regulation or ownership prevents the data warehousing team from copying and consolidating the data in a data warehouse. Data federation can overcome these limitations by giving business intelligence (BI) users a consolidated view of data in both the data warehouse and other systems. For example, DW managers might augment a data warehouse with real-time data in an operational system. Or they might deliver a 360-degree view of customers by combining customer data in a data warehouse with billing data from a financial system and service data from a technical support system.

2. **Federate multiple data warehouses.** What do you do when your organization sprouts multiple data warehouses and data marts? Or your organization buys another company with its own set of data warehouses? Using data federation software, you can create a virtual data warehouse that combines data from each physical data warehouse. This technique might make it possible to give the CEO a view across reporting systems after a merger or allow executives to analyze processes that cut across departments, each of which is served by its own data warehouse.

3. **Augment ETL processes.** ETL tools can use existing virtual views and data services as inputs to their batch processes. This practice can expedite ETL development and enable ETL tools to support data sources they can’t easily access, such as SAP or cloud-based applications.

4. **Migrate data warehouses.** If you need to migrate or replace a data warehouse, data federation can help minimize the impact on downstream reports, reducing costs and risks. To migrate the data warehouse, you need to rewrite report queries to run against data federation software instead of directly against the old data warehouse. Once the new warehouse is in place, you modify the semantic layer in the data federation tool to point at the new source. This insulates your reports and queries from source system changes.
Data federation is an ideal way to enhance existing business intelligence (BI) environments. Data federation provides a robust abstraction layer that accelerates deployment and minimizes the impact of changes. Here are four use cases for BI:

1. **Accelerate the deployment of BI solutions.** Business executives often have an urgent need for information when there is not enough time to adhere to established standards and architectures. In some cases, the information is a one-time request and doesn’t warrant the expenditure of time and money to build a permanent solution. In these cases, it’s quicker and easier to build a BI solution using data federation tools. If the organization decides to make the application permanent, then administrators can easily flip a switch in the data federation software to make it persist data rather than pull it on the fly. Or they can extend the schema in their physical architecture to support the new application.

2. **Prototype new BI solutions.** Rather than build a data warehouse or data mart from scratch based on requirements gathered from interviews and joint application development (JAD) sessions, you can use data federation software to prototype a solution, building data views on the fly, and get quick feedback from business users. Once the users sign off on the proposed views, you can build the physical data warehouse using the schema built into the prototype. This procedure saves time and money and reduces the risk of creating a solution that doesn’t meet user requirements.

3. **Create analytic sandboxes.** A problem that plagues most organizations is rogue data marts created by analysts who combine data from multiple systems and dump it into a spreadsheet or desktop database to do their analysis. These analysts generate silos of fragmented, inconsistent data, making it impossible to get a single version of truth and potentially causing executives and managers to base their decisions on questionable data. To alleviate this problem, developers can use data federation software to build consistent views of analytical data (in data warehouses and marts), operational data, and external data.

4. **Consolidate BI tool sets.** There’s nothing worse for a business user than to have to use different tools to access different systems and data. Organizations can overcome this problem by augmenting a single BI tool with data federation software that points to all the sources supported by the existing BI tools.

Data federation can be applied in a number of situations to speed deployment, reduce costs, or both. Following are some common applications of data federation:

**360-Degree Views.** Data federation makes it possible to consolidate data on the fly from multiple systems to deliver a 360-degree view of customers, employees, products, or suppliers as a complement to MDM. For example, you can use data federation to pull customer data from billing, sales, service, and provisioning systems to deliver to customer service representatives a 360-degree view of a customer’s profile and history of interactions with the company—while talking with the customer on the phone. This practice spares companies from having to physically consolidate this data in an expensive data warehouse.

**SOA.** Data federation tools are often used to provide data services within a service-oriented architecture. By abstracting back-end systems via a business semantic layer and common data access interface, data federation software can provide the data services within an SOA.

**Cloud Computing.** Data federation can integrate data from information silos, such as cloud computing applications, with internal data. For example, a company can use data federation to create a report with data from a software-as-a-service (SaaS) provider, such as Salesforce.com, and an internal customer data warehouse.

**Real-Time Data.** In the same way, data federation can augment historical data in a data warehouse with real-time data from an operational application. For example, a dashboard displays yesterday’s sales derived from the data warehouse in one pane and a running total of today’s sales updated every 15 minutes from an order entry system.
The three most important considerations when evaluating a data federation tool set are:

1. **Robust, easy development environment.** (Will my developers use it?) A data federation tool should have a development environment that is easy to learn and use. It should leverage standards that developers already understand and support both SQL and XML approaches to data.

2. **High performance.** (How fast will my queries run?) A good data federation tool has a sophisticated query optimizer that exploits the latest hardware, network, operating system, and relational database management system capabilities. It should support flexible data caching and balance workloads across a clustered environment.

3. **Reliable operations.** (Can I sleep at night?) The best data federation software has a proven track record in corporate data processing environments. The software aligns with data center standards, provides backup and failover capabilities, and works with various systems management tools, such as IBM Tivoli, HP OpenView, and BMC Atrium.
Composite Software, Inc., is the only company that focuses solely on data virtualization.

Global organizations faced with disparate, complex data environments, including ten of the top 20 banks, six of the top ten pharmaceuticals, four of the top five energy firms, major media and technology companies as well as government agencies, have chosen Composite’s proven data virtualization platform to fulfill critical information needs, faster with fewer resources.

Scaling from project to enterprise, Composite’s middleware enables data federation, data warehouse extension, data sharing, real-time and cloud computing data integration.

Founded in 2002, Composite Software is a privately held, venture-funded corporation based in Silicon Valley.

For more information, please visit www.compositesw.com. Or follow Composite on http://twitter.com/compositesw.

Wayne Eckerson is the director of TDWI Research. Eckerson is an industry analyst and educator who has covered DW and BI since 1995. Eckerson is the author of many in-depth, groundbreaking reports, a columnist for several business technology magazines, and a noted speaker and consultant. He is the author of Performance Dashboards: Measuring, Monitoring, and Managing Your Business (John Wiley & Sons, 2005) and the creator of TDWI’s BI Maturity Model and Benchmarking Assessment service. He can be reached at weckerson@tdwi.org.

TDWI Research provides research and advice for BI professionals worldwide. TDWI Research focuses exclusively on BI/DW issues and teams up with industry practitioners to deliver both broad and deep understanding of the business and technical issues surrounding the deployment of business intelligence and data warehousing solutions. TDWI Research offers reports, commentary, and inquiry services via a worldwide Membership program and provides custom research, benchmarking, and strategic planning services to user and vendor organizations.

TDWI Checklist Reports provide an overview of success factors for a specific project in business intelligence, data warehousing, or a related data management discipline. Companies may use this overview to get organized before beginning a project or to identify goals and areas of improvement for current projects.