
# The Future of Data Warehousing

## Integrating with Data Lakes, Cloud, and Self-Service

**By Dave Wells**

**September 2018**

Research sponsored by:

Infoworks

# About the Author

**Dave Wells** is an advisory consultant, educator, and research analyst dedicated to building meaningful connections along the path from data to business impact. He works at the intersection of information and business, driving value through analytics, business intelligence, and innovation. With nearly five decades of combined experience in information management and business management, Dave has a unique perspective about the connections of business, information, data, and technology.

Knowledge sharing and skills building are Dave's passions, carried out through consulting, speaking, teaching, research, and writing. He is a continuous learner—fascinated with understanding how we think—and a student and practitioner of systems thinking, critical thinking, design thinking, divergent thinking, and innovation.

# About Eckerson Group

Eckerson Group is a research and consulting firm that helps business and analytics leaders use data and technology to drive better insights and actions. Through its reports and advisory services, the firm helps companies maximize their investment in data and analytics. Its researchers and consultants each have more than 25 years of experience in the field and are uniquely qualified to help business and technical leaders succeed with business intelligence, analytics, data management, data governance, performance management, and data science.

# Executive Summary

*Recent developments in data management—self-service, big data, data lakes, NoSQL, Hadoop, and the cloud—raise questions about the role of the data warehouse in modern analytic ecosystems. Though pundits have declared the data warehouse dead, most organizations continue to operate at least one data warehouse, with the majority operating two to five, and expect to do so for the foreseeable future. Data warehousing is alive, but perhaps not alive and well.*

*Legacy data warehouses must modernize to fit gracefully into modern analytics ecosystems. They play an important role in data management as an archive of enterprise history and a source of carefully curated and highly integrated data for a broad scope of line-of-business information needs. To continue filling that role well, they must evolve both architecturally and technologically. Yet in many instances, data warehouse evolution is stalled due to uncertainty about what, how, and when to change.*

*This report provides guidance to break the logjam and begin moving to data warehouses that are agile, scalable, and adaptable in the face of continuous change. It describes how patterns of architectural restructuring, cloud migration, virtualization, and more can be used to combine data warehouses with big data, cloud, NoSQL and other recent technologies to resolve many of today's data warehousing challenges and to prepare for the future of data warehousing.*

# Data Warehousing Is Not Dead

Despite declarations by pundits, data warehousing is not dead. Recent surveys show that more than 60% of companies are operating between two and five data warehouses. Fewer than 10% have only one data warehouse or none at all. Nearly one-third of respondents work in an organization with six or more data warehouses. Although the vision from the past generation of BI and data warehousing-one data warehouse that serves as a single version of the truth-has not been realized, it is clear that data warehousing continues to provide value. We must sustain the concepts and practices of data warehousing-integrated, subject-oriented, non-volatile, and time-variant data-without being tied to legacy architectures and implementations.

## The Value of Data Warehousing

Data warehouses meet the information needs of people and continue to provide value. Many people use them, depend on them, and don't want them to be replaced with a data lake. Data lakes serve analytics

and big data needs well. They offer a rich source of data for data scientists and self-service data consumers. But not all data and information workers want to become self-service consumers. Self-service analytics does not replace data warehousing; it extends and complements. Published data (warehousing) and ad hoc data (self-service) work together to meet a broad spectrum of information needs.

***People continue to need well-integrated, systematically cleansed, easy-to-access data that includes time-variant history.***

Companies continue to operate data warehouses because they are needed. Business processes and information workers depend on warehouse data and information on a daily basis. Many people-perhaps the majority-continue to need well-integrated, systematically cleansed, easy-to-access relational data that includes a large body of time-variant history. They want to meet routine information needs with data that is prepared and published with those needs in mind. These people are best served with data warehousing that provides:

- **Subject-oriented data** that is organized around major business subjects such as customer, product, employee, etc., and that is readily mapped to business semantics.
- **Integrated data** where disparity among data sources is resolved to provide a consistent, reliable, and trusted source of data for reporting and analysis.
- **Time-variant history** that is captured at uniform time intervals, is kept beyond its lifespan in operational source systems, and is organized to support trending and time-series analysis.
- **Non-volatile data** where history is retained without revision, supporting reliable and repeatable reporting and analysis of past business events.
- **Cleansed data** transformed to mitigate the risks inherent in data quality defects.
- **Published data** that is delivered on a regular schedule and that is known, repeatable, and ready for use.

The characteristics that Bill Inmon set forth in his 1992 definition of a data warehouse-subject-oriented, integrated, time-variant, and non-volatile-continue to be desirable data qualities for reporting and for many of today's analysis use cases. The modern data warehouse may take many forms such as a physically distinct database, a rigorously structured and managed zone in a data lake, or a virtual warehouse with on-demand integration. Regardless of form, we continue to need the unique benefits of data warehousing.

## The Challenges of Legacy Data Warehousing

Data warehousing is alive, but perhaps not entirely well. Big data, NoSQL, data science, self-service analytics and demand for speed and agility all challenge legacy data warehousing. Traditional data warehousing-predicated on 1990s data management practices-simply can't keep up with the demands

of rapidly growing data volumes, processing workloads, and data analysis use cases. Data warehousing must evolve and adapt to fit with the realities of modern data management and to overcome the challenges of scalability and elasticity, data variety, data latency, adaptability, data silos, and data science compatibility.

***Data warehousing is alive, but perhaps not alive and well.***

### *Scalability and Elasticity*

Legacy data warehousing infrastructure addresses growth as a scale-up problem-buying bigger hardware as data and processing workloads grow. Scaling up is inadequate for today's data volumes, processing workloads, and query rates. Warehousing must evolve from scaling up to scaling out. Legacy infrastructure is typically designed to perform at a level that meets the demands of peak workloads. As data volume, processing complexity, and data use cases continue to expand, workload peaks and valleys become more extreme. Tooling up for peak workload is costly and much of the computing capacity goes unused. Elasticity-ability to dynamically provision and de-provision resources as needed-is on par with scalability as an important goal for data warehouse modernization.

### *Data Variety*

Most data warehouses are implemented using relational database management systems. Relational technology was the predominant database technology of the day and most warehouse data was sourced from relational databases used by enterprise operational systems. The big data phenomenon radically expanded the variety of available data sources and the ways in which data is organized and structured. Modern data warehouses must be able to ingest data from graph databases, key-value stores, document stores, XML files, JSON files, and a variety of other sources. They may also get advantage from storing warehousing data in NoSQL databases when working with unstructured, semi-structured, and multi-structured data.

### *Data Modeling*

Relational database management relies on the practice of rigid data modeling. Legacy data warehouses are built upon a relational model, and all data must fit into the model. Change becomes burdensome when new data, new business requirements, and changing data sources require data model refactoring and modification of existing ETL processes. Rigid data models also limit the analytic value of the data warehouse. The one-size-fits-all data model that works for OLAP doesn't work well for modern analytics where every use case may have unique data requirements.

### *Data Latency*

A typical data warehouse acquires most of its data through batch ETL processing with periodical warehouse refreshes, most commonly with daily and weekly ETL processing. Batch processing is inherently latent. Intra-day refresh and micro batching reduce latency but they don't deliver real-time data. As the speed of business accelerates, data drives process automation; and digital transformation

intensifies dependency on data. These factors combine to expand and amplify the demand for real-time and very-low-latency data. The modern data warehouse must be able to ingest and process data at the right frequency for each data source. Data ingestion modes must span the continuum from periodically scheduled ETL to real-time stream processing.

### *Adaptability and Self-Service*

Legacy data warehouses aren'tdon't readily adaptable to change. Every change is time consuming and costly, yet source systems, business needs, and technologies change frequently. Too much manual effort, too few tools, and absence of reliable documentation create narratives of, "it takes too long, costs too much, and isn't what I really want." These experiences result in frustrated business people, overloaded and demoralized technical staff, and a data warehouse of diminishing value. Ability for business people to self-serve new data sources and support new analytic use cases, architectural refinements, and automation are core concepts for an adaptable data warehouse.

### *Data Silos*

The early vision of data warehousing-a single place to go for integrated and trusted data-has clearly not become reality when 90% of companies operate two or more data warehouses. The reasons for multiple data warehouses are many, including mergers and acquisitions, independently developed departmental and line-of-business warehouses, geographically specialized warehouses for multi-national companies, and more. Regardless of the causes, multiple warehouses create the very data silos that warehousing is intended to eliminate. Modern warehousing through virtualization, federation, or consolidation seeks to break down these silos.

Silos also proliferate where data warehouses and data lakes are both present. Questions arise about the right source or the best source of data for reporting, analysis, and other use cases. In a modern data ecosystem, data lakes and warehouses should coexist, be compatible and complementary, and prevent the confusion caused by conflicting information.

### *Data Science Compatibility*

Data science often uses data differently than other applications do. Outliers, exceptions, anomalies, and inconsistencies are typically considered to be data quality deficiencies that are cleansed as part of data transformation. But the very things that are viewed as defects for reporting and OLAP applications may provide the best opportunities for insight through data science. A modern data warehouse avoids cleansing away analytic opportunities by keeping raw or "as-received" data as well as refined and "as-cleansed" data.

# Data Warehouse Modernization

Data warehousing badly needs modernization. The challenges described here can only be overcome by rethinking the architecture, design, and implementation of data warehouses. Modernization is essential if data warehousing is to keep pace with changes in business, compress data-to-insight cycles, and respond to GDPR and other regulatory pressures.

Considering the many challenges of legacy data warehousing, it is tempting to declare the death of the data warehouse and move to a data lake where integration, subject orientation, and time variance are not defining characteristics. Avoid that temptation because it is not a real and viable solution. The data lake is not a silver bullet. It is a valuable source of data for many analytics use cases, but a typical data lake that is not designed to be subject-oriented, integrated, non-volatile, and time-variant is not well positioned to deliver the value of data warehousing. The ideal solution is a data lake and data warehousing working together in a data management ecosystem that provides the right data for the full spectrum of use cases from basic reporting to advanced analytics and data science. Achieving that ideal solution requires architectural, operational, and technological modernization of legacy data warehouses.

## The Challenges of Modernization

Data warehouse modernization is necessary but it can be challenging. Major challenge areas include knowledge and understanding, uninterrupted operations, and architectural deficiencies.

### *Knowledge and Understanding*

Knowledge of data, processing, and implementation details of legacy warehouses can be elusive. They are often poorly documented and poorly understood. They have operated for years with occasional patches applied when data sources change or other maintenance needs arise. With each patch the gap between documentation and reality grows. Finding subject matter experts is equally challenging. Original developers of in-house data warehouses have likely moved on to other jobs or roles, or perhaps retired. Knowledge transfer is rare when data warehouses are acquired through merger and acquisition.

### *Uninterrupted Operations*

Data warehouse modernization is likely to occur as a series of changes implemented over a period of time. Throughout the modernization journey it is important to implement changes without disrupting day-to-day operations. People and processes depend on warehouse data for information to do their work. Minimizing downtime and user impact are important considerations when planning to modernize.

### *Architectural Deficiencies*

The underlying architecture of legacy data warehousing is predicated on 1990s concepts and principles.

A typical data warehouse is built using hub-and-spoke architecture as promoted by Bill Inmon, bus architecture as popularized by Ralph Kimball, or a hybrid that blends both. These architectural frameworks share some common characteristics that limit the value of data warehousing as a modern data management component:

- **Slow development and change processes.** Legacy data warehouse architecture requires data to fit into a data model. When new kinds of data arrive the model must be updated, and the ripple effects necessitate processing changes. The change processes for legacy warehousing systems are very slow and don't support the realities of today's data-driven economy.
- **Linear data flow and work flow.** Data moves from sources, through ETL processing, to integrated data stores, and then is accessed by applications. The multi-directional data flow of modern data pipelines is not readily supported.
- **Structured enterprise data.** Data sources include OLTP systems and legacy databases along with a limited amount of external data. All data has known schema and most data sources are relational database tables. Semi-structured data, unstructured data, and data without rigid schema are not easily accommodated.
- **Batch processing.** Data is processed and published for consumption through scheduled ETL processes that run as batch jobs. Data warehouse refreshes happen daily or weekly. Data streams are difficult to process in real time. Batch schedules limit the compression of data-to-insight cycle times.
- **Data latency.** Batch processes inherently create data latency. Real-time data is an exception that requires special processing.
- **Rigid infrastructure.** Introducing new technologies, data subjects, and data sources is impractical without careful impact analysis and labor-intensive implementation projects. In fact, changes to upstream data sources such as adding a column to a source table will break ETL processes and create downstream problems for the data warehouse.

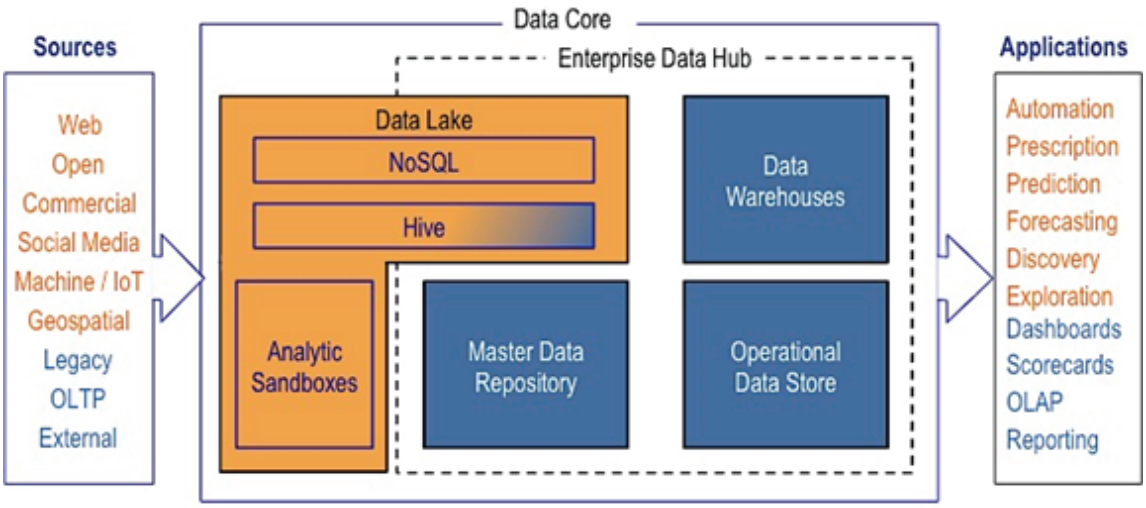
## The Goals of Modernization

Data warehouse modernization is a purposeful undertaking to achieve specific and tangible benefits that improve data management, enhance data value, and drive positive business impact. Those benefits include:

- **A complete and cohesive analytics ecosystem.** Analytics processes and projects depend on data of many types-transactional data, event data, and reference data-that comes from enterprise systems and databases as well as from big data sources. Big data, in fact, has little context or meaning for analytics until it is connected with enterprise data. Existing data warehouses need to integrate into the analytics ecosystem, working together with a data lake, to provide the full range of data needed for analytics.
- **Complete and cohesive data management architecture.** In today's complex data management

world, each kind of data store-data lake, data warehouse, master data repository, etc.-has a specific role and purpose. An architectural framework that designates all of the components and illustrates how they work together is important to break down data silos, minimize data redundancy, and maximize data reuse. (See Figure 1.)

**Figure 1. An Architectural Framework for Modern Data Management**




• **Governed self-service.** Modern data management enables self-service data and analytics without loss of essential governance controls. Balancing the need for self-service freedom with imperatives for data privacy, data security, and access controls requires a proper governance framework as a core characteristic of modern data architecture.

• **Maximum reusability and reuse.** Fitting data warehousing into comprehensive data management architecture fosters reuse of data and reduces data duplication.

• **One stop shopping for data.** Fitting legacy warehouses into a cohesive analytics ecosystem makes it easy for analysts to find and access data of all types without needing to know where and how it is stored. Data analysts focus on getting the right data for each use case without needing to search separately in a data lake, multiple data warehouses, MDM, and other data stores. Warehouse modernization is an important step along the path to building an Enterprise Data Marketplace.

• **Data for all.** Comprehensive data management architecture brings together data warehouses, the data lake, MDM, and other data stores to be presented as a single logical data resource that supports all users and use cases from reporting to data science.

• **Technological updates.** Modernizing legacy data warehouses includes migrating to recent technologies that improve scalability, elasticity, performance, data variety, and data freshness.

• **Architectural updates.** Integrating data warehousing into modern data management architecture improves agility, adaptability to change, and maintainability of legacy data warehouses. Data warehousing architecture should be modernized to readily accept a variety of

new data sources, process data streams in real time, and substantially reduce data latency. Underlying technology architecture should be built for evolution. Recognize that the pace of technology change is accelerating and that continuous technology refresh is an essential part of modern data management.

- **Automation opportunities.** Keeping up with the ever-increasing demand for data isn't practical when all data engineering is manual effort. As demand grows and availability of data engineers shrinks, automation becomes increasingly important. Automation cuts the time, cost, and risk of development and deployment.
- **Hyper-converged infrastructure.** Data management infrastructure that tightly integrates computing, storage, networking, virtualization, and analytics with a cohesive, software-centric architecture improves performance, takes full advantage of scale-out technologies, advances IT agility, and meets the ever-rising demand for analytics and data science capacity.
- **Deployment independence.** As organizations move to the cloud at an accelerating rate, separating architecture and deployment platform is important. Ability to migrate between on-premises, cloud, and hybrid implementations is practical only when architecture is separated from underlying processing and storage technologies. In legacy data warehousing environments architecture and technology choices are tightly coupled. Decoupling them reduces complexity and enhances agility when keeping up with continuously evolving technologies.

# Design Patterns for Modern Data Warehousing

There is no one-size-fits-all solution for data warehouse modernization. Every legacy data warehouse is unique, thus every modernization plan is unique. There are, however, several design patterns for modern data warehousing that help to bridge the gap between current state and future-state goals. Common patterns include architectural frameworks, cloud data warehousing, automation and virtualization, data warehousing with Hadoop, and adoption of recently emerged technologies. Follow these patterns individually or in combination to develop a modernization plan and frame your modern data warehouse design.

## Architectural Frameworks

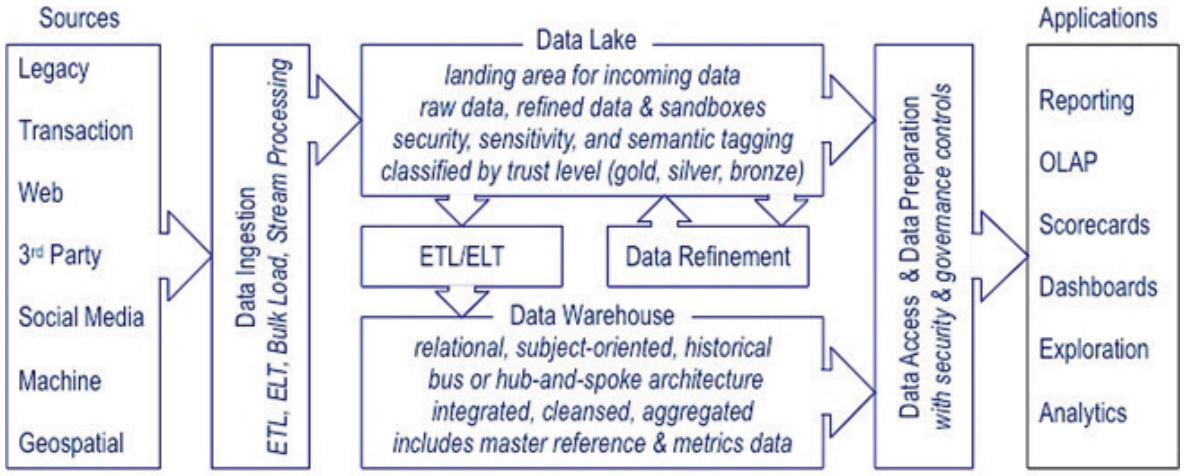
As previously described, data warehousing and the data lake need to work together as complementary components of cohesive data management architecture. These frameworks describe different perspectives on positioning the data warehouse relative to the data lake.

### *Data Warehousing Outside the Data Lake*

This variation treats the data lake and the warehouse as separate data stores without overlap. The data lake is the landing zone for all incoming data, and warehouse ETL draws data directly from the lake. (See Figure 2.) The data lake's landing zone serves as warehouse data staging. Sharing a common landing

zone for all incoming data reduces redundancy, retains raw data as received, and supports fully traceable data lineage.

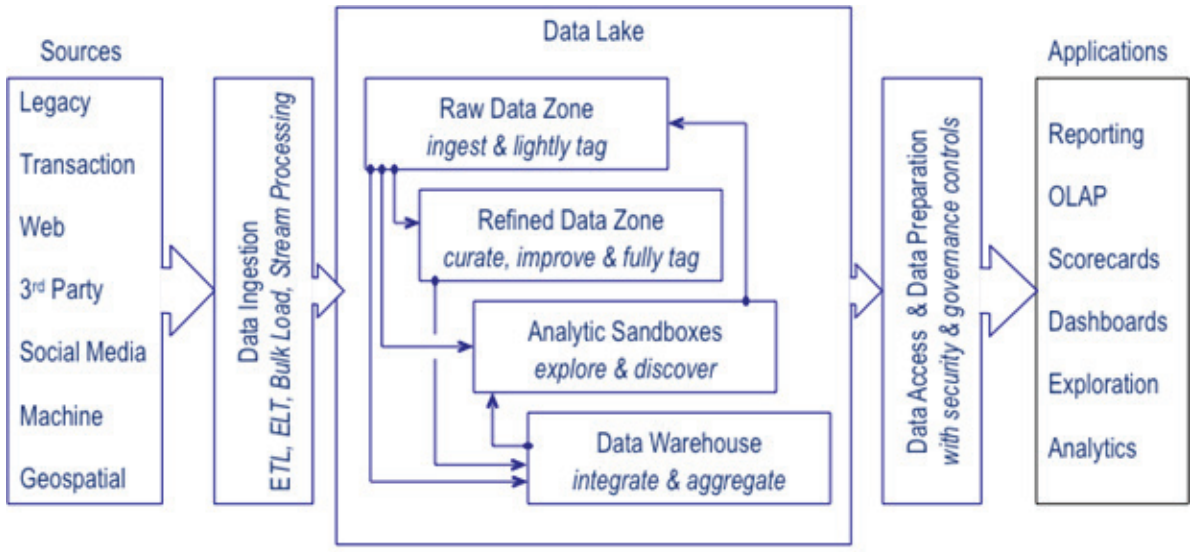
**Figure 2. Data Warehousing Outside the Data Lake**




## *Data Warehousing Inside the Data Lake*

This framework positions the warehouse as part of the data lake. (See Figure 3.) The warehouse may acquire data from a raw data zone (data staging) and from a refined data zone where some cleansing and transformation work has already been performed.

**Figure 3. Data Warehousing Inside the Data Lake**

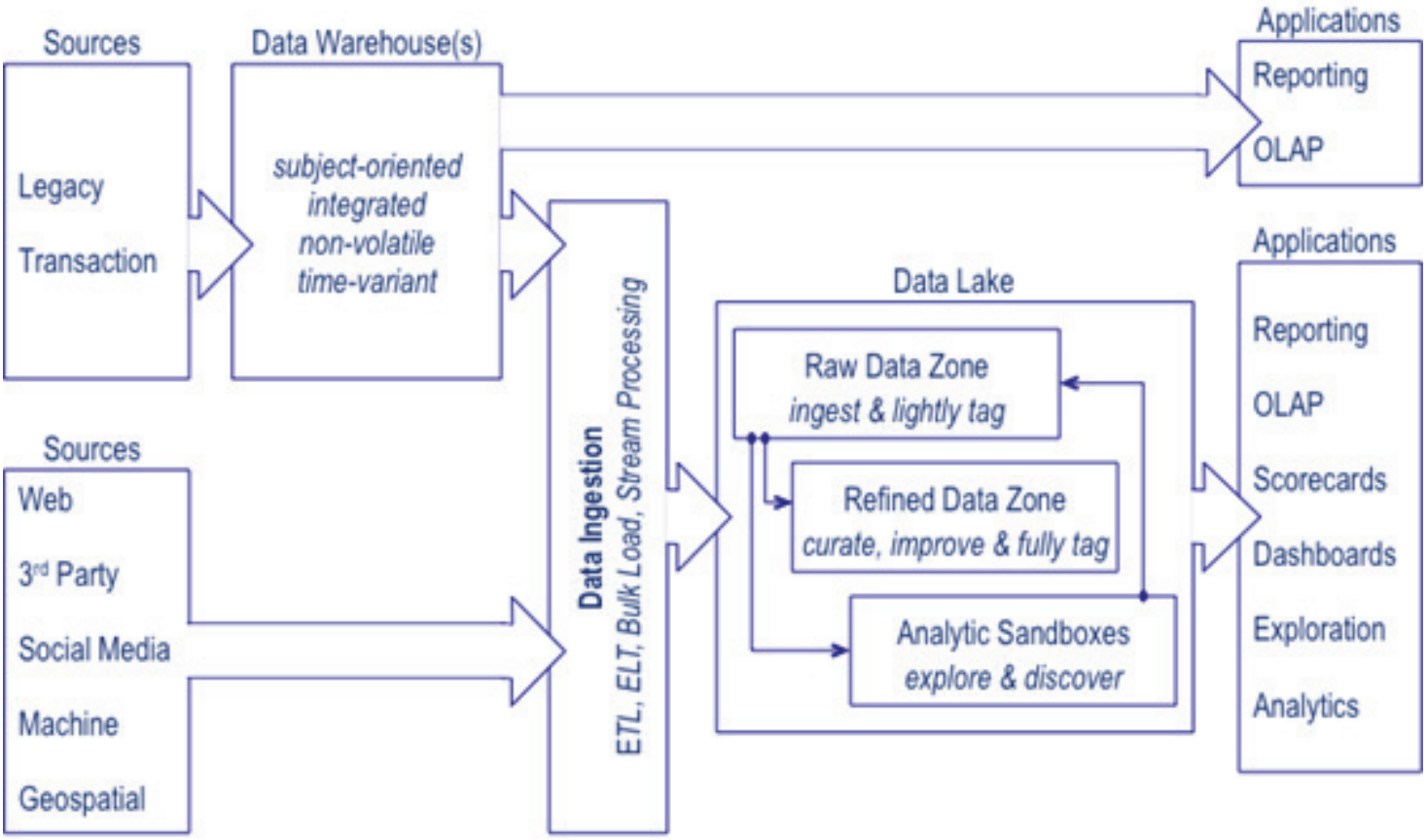



It may be especially desirable to position a data warehouse as a subset of the data lake when that warehouse is expected to have a long lifespan with a significant number of users who need to work with raw data, refined data, and integrated and historical warehouse data.

### *Data Warehousing in Front of the Data Lake*

In this variation, one or more data warehouses continue to operate independently, but they also become sources for data ingested into the data lake. The modernization advantage here is limited because the data warehouses remain unchanged. Pushing warehouse data to the data lake creates an additional copy of the data, but it also eliminates the silo effect of multiple data warehouses and the data existing separately and in isolation. Although advantages are limited, complexity and effort are relatively small and there is no visible impact to data warehouse users. This may be a practical first step of a multi-phase modernization process.

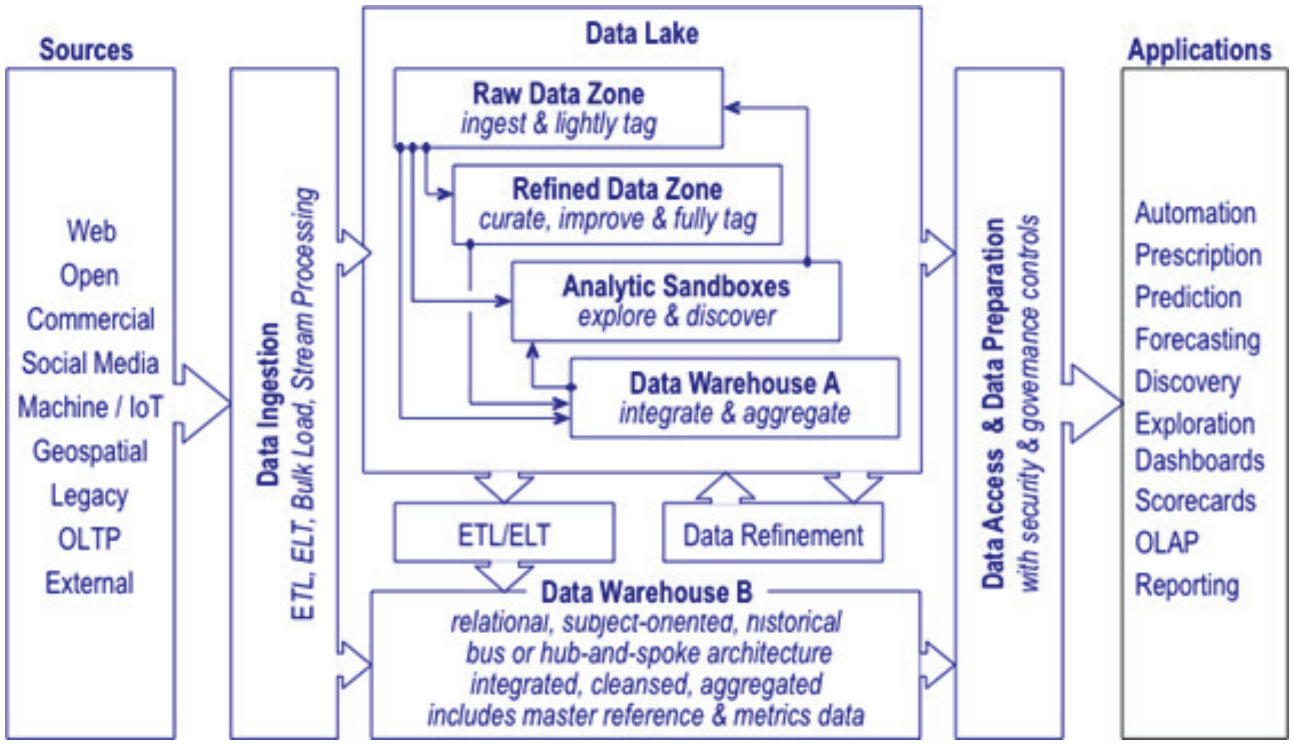
**Figure 4. Data Warehousing in Front of the Data Lake**




### *Data Warehouse and Data Lake Inside/Outside Hybrid*

With multiple data warehouses, it can be practical to implement a hybrid. (See Figure 5.) Data warehouses with high analytics usage and significant overlap with other data lake contents are positioned inside the data lake. Those with limited user base, primarily used for inquiry and reporting, remain outside the data lake.

**Figure 5. Data Warehouses Inside and Outside the Data Lake**



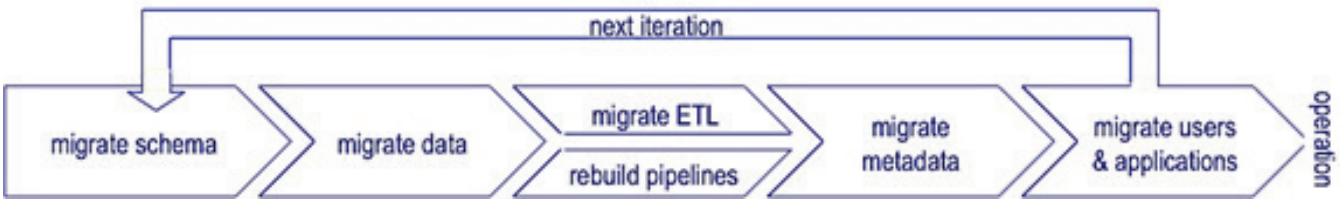

### *Architectural Decisions*

The architectural frameworks shown here are concepts, not prescriptions. Choose among them or blend and adapt them to fit your data management needs. Consider the number of data warehouses needing modernization, the maturity and stability of your data lake, and your organization's capacity for architectural change. Big changes may offer big benefits, but they involve a lot of work. Make architectural modernization practical with an incremental approach. First create the vision of your future architecture, then build the plan to get there one step at a time.

## Migrating Legacy Data Warehouses

Migrating an existing data warehouse to a new architecture or a modern data management platform offers substantial benefits and is a practical step to modernization, but it is not quick or easy. Tactically and technically, data warehouse migration is a challenging multi-step process to move many different warehousing components. (See figure 6.)

**Figure 6: Migrating a Data Warehouse to the Cloud**



Migrating a data warehouse to the cloud requires all of the following activities:

- **Migrating Schema.** Before moving warehouse data, you'll need to migrate table structures and specifications, possibly with structural and indexing changes.
- **Migrating Data.** Moving very large volumes of data can be process-intensive, network-intensive, and time-consuming. Don't underestimate resources needed to move the data. If you're transforming data as part of migration, decide whether to transform in stream or pre-process and then migrate.
- **Migrating ETL.** Moving data may be the easy part compared to migrating ETL processes. You may need to change the code base to optimize for platform performance, and change data transformations to sync with data restructuring. This is also an opportunity to reduce data latency.
- **Rebuilding Data Pipelines.** With any substantive change to data flow or data transformation, rebuilding data pipelines may be a better choice than migrating existing ETL.
- **Migrating Metadata.** Source-to-target metadata is a crucial part of managing a data warehouse, knowing data lineage, and tracing and troubleshooting when problems occur. Consider how you'll move metadata to the cloud platform and how to maintain continuous data lineage that blends pre- and post-migration data movement.
- **Migrating Users and Applications.** The final step in the process is migrating users and applications to the new cloud data warehouse with little or no interruption of business operations. You'll need to migrate security and access authorizations, reconnect BI and analytics tools, and openly communicate with stakeholders throughout.

Whether migrating your data warehouse to the cloud, to Hadoop, or another data management environment, plan a step-by-step approach. Consider data integration and data management tools carefully with preference for tools that are architecture independent.

## Data Warehousing in the Cloud

Data warehousing in the cloud has become popular as companies are challenged with growing data volumes, higher service-level expectations, and the need to integrate structured warehouse data with unstructured data in a data lake. The movement to SaaS for enterprise applications also makes cloud data warehousing an inviting option.

### *Why Cloud Data Warehousing?*

Cloud data warehousing responds to many of the legacy data warehouse challenges previously discussed, offering a targeted and direct response to the challenges of scalability, elasticity, performance, and workload management. Less direct but equally important benefits include ready access to technologies designed for non-relational and unstructured data, improved adaptability and agility through instant infrastructure, and reduced dependency on on-premises data centers (with associated cost savings).

### *Cloud-Optimized Data Warehousing Architecture*

Cloud-optimized architecture amplifies the benefits of cloud data warehousing beyond what is possible by simply migrating on-premises data warehousing to a cloud native environment. Taking full advantage of cloud platform features and capabilities increases scalability and elasticity and delivers improvements in both data pipeline and query performance. With a cloud-native data warehouse the cloud platform simply hosts the warehouse. A cloud-optimized data warehouse does more than hosting. The cloud platform actively manages the warehouse. Cloud-optimized data warehouse architecture:

- Continuously optimizes resource allocation through monitoring and machine learning.
- Separately and independently scales storage and computing.
- Minimizes administration and management needs.
- Works with data of all types from traditional relational to NoSQL.
- Works with data at all velocities from batch ETL to streaming data.
- Optimizes data storage and data operations using technologies such as high-performance columnar databases and object stores.
- Delivers the performance needed for high-volume, compute-intensive data science projects.
- Is the foundation for data warehousing as a service (DWaaS), offering a fully managed, pay-as-you-go model to outsource data warehouse administration and operation.

## Data Warehouse and Big Data Automation

Automation technologies eliminate much of the manual effort from data management. They accelerate development and deployment, improve agility and response to changes, promote reuse, and increase standardization and consistency. Data warehouse automation is mature technology that has been available for more than a decade and is now proving valuable in modernization efforts. Big data automation is more recent technology that is especially valuable in architecture where data warehousing and data lakes converge.

### *Data Warehouse Automation*

Data warehouse automation tools deliver efficiencies and improve effectiveness in data warehousing processes. More than simply automating ETL and development processes, automation encompasses all of the core processes of data warehousing including design, development, testing, deployment, operations, impact analysis, and change management.

**Agility through automation** - Using an integrated development environment, automation tools help developers and business stakeholders collaborate through a process of iterative requirements discovery, design, and development when building, enhancing, or modifying a data warehouse. Automation is an essential part of agile data warehousing.

**Automation and reverse engineering** - When the design and logic of an existing data warehouse need

to be systematically extracted and understood in preparation to modernize, metadata-based automation tools provide real value. Capturing the hidden design as metadata exposes relationships and dependencies among data warehouse components is easy to understand and to verify.

**Automation and ETL modernization** - When the design and logic of ETL processing is described as metadata in an automation tool, ETL reengineering is accelerated and simplified. By adjusting the metadata to include reengineering and modernization goals, desired new ETL processes can be quickly generated, tested, and deployed.

### *Big Data Automation*

Big data automation technologies extend automation benefits beyond data warehousing to encompass data lakes and the full scope of data engineering. Big data and data lakes expand data management processes from a relatively small number of ETL jobs to potentially thousands of data pipelines. Development, operation, and maintenance of data ingestion, change data capture, data transformation, and data preparation processes are accelerated, simplified, and actively managed with big data automation technologies.

**Automated data lake management** - Beyond the capabilities of data warehouse automation tools, big data automation handles complexities of the big data world such as streaming data ingestion, changed data capture, schema change detection, varied non-relational data sources, and simultaneous and asynchronous execution of multiple data pipelines. Managing data lakes and big data without automation typically requires an army of big data specialists. Automated data lake management eliminates the high level of staffing and skills demand.

**Automation of data ops** - Building and operating a few data pipelines to populate a data warehouse is a relatively small challenge. Building and managing hundreds or thousands of data pipelines is a substantially more complex job. Automating pipeline operations as well as development is a key to sustainable and adaptable modern data management. Without automation data engineers are consumed with the challenges of day-to-day operations, which limits their capacity to respond to new data requirements and new use cases.

**Automation and platform independence** - Data lake and data warehousing technologies and implementations are evolving at a breakneck pace. Automated data engineering and data ops processes alleviate many of the challenges of migrating to new architectures and technologies as they emerge.

## Data Virtualization

Data virtualization technology enables applications to access and retrieve data without the need to know implementation and technical details, such as where the data is stored or how it is formatted. Virtualization hides the technical details of the data, allowing applications and users to focus on data meaning rather than processing.

In data warehousing and data integration applications, virtualization can be contrasted with materialization. The goal of materialization is to provide a single source of rationalized data, where source describes a physical database. The goal of virtualization is to provide a single view of rationalized data, where view implies a logical, but not physically instantiated data structure. A virtual query navigates through levels of data abstraction—generally from business view, to integration view, and then to physical view. It retrieves data from sources based on the physical view, and then reverses the path to transform data for integration, and to deliver query results in business context. Data virtualization can contribute to data warehouse modernization in several ways.

### *Logical Data Warehouse Architecture*

Logical data warehouse architecture applies data virtualization technology to create a non-physical abstraction layer between data sources and data consumers. The primary advantage of a logical data warehouse is adaptability. Connecting new data sources—both traditional relational data and the variety of big data types—is relatively quick and easy. Adjusting for changes to existing data sources is also quick and easy. The disadvantages may be loss of history (virtualization can only return as much history as is retained in the source databases) and performance issues (complex data transformation means a lot of heavy lifting between query and response). If your challenge is source volatility, retention of history is not a concern, and data transformations are relatively simple, then the logical data warehouse may be a good fit.

### *Federated Data Warehousing*

Virtualization can be a good choice to break down the silos and resolve disparities between multiple data warehouses. Federating multiple data warehouses to provide a system-of-record view of data creates new value opportunities with all of the data and eases the pain that data consumers experience when searching for data. Normally the source warehouses retain needed history, so loss of history is not a concern. Source warehouses have typically performed the complex and difficult data transformations, mitigating the risk of poor query performance.

### *Isolating Users from Impact of Change*

**Minimizing user impact** - One of the big challenges of modernization is minimizing the impact on data warehouse users as the environment evolves. Whether cloud migration, architectural update, or other approaches, modernization is an evolutionary process that frequently introduces changes to warehouse implementation. Introducing a virtual data access layer as the consumer-facing component of the data warehouse will help to isolate users from the impacts of frequent change.

## Data Warehousing and Hadoop

Hadoop has been widely adopted as big data and data lake technology. For organizations that have invested in Hadoop implementation and skills development, it is natural to ask how Hadoop can address data warehouse modernization needs. Hadoop is a viable solution when scalability, elasticity, and database capacity are among the legacy data warehousing challenges.

**Hadoop for scalability and elasticity** - Hadoop can be used as a high-performance transformation engine. It has advantages when processing capacity and performance are key concerns for data warehouse modernization. Large volumes of data and complex, compute-intensive transformations often lead to ETL performance problems. Speed, parallelism, fault tolerance, and scalability of data transformations are among the Hadoop benefits.

**Hadoop for database offloading** - Mature and aging data warehouses often contain data that is valuable but infrequently accessed. Although the data must be retained for occasional use, it can lead to high costs, database management issues, and query performance problems that are associated with very large databases. Hadoop and NoSQL offer an alternative for retaining aging and rarely used data in a way that is cost effective, performance friendly, and still readily accessible.

## Recently Emerged Technologies

Several recently emerged technologies can address or resolve some of the legacy data warehouse challenges. Among these technologies are the following:

**In-memory columnar storage** - Both columnar storage and in-memory data have been available for years. Neither can be described individually as recently emerged. But the combination of columnar and in-memory is the recent development—perhaps within the past three years—that delivers measurable performance gains when working with large volumes of structured data and compute-intensive analytics applications.

**Big data OLAP cubes and in-memory models** - Accessing and analyzing data in a big data repository with self-service visualization and analysis tools is challenging. The self-service tools are not designed to process exceptionally large volumes of data efficiently, and data storage technologies such as Hive are not designed for sub-second responses to complex queries. In-memory data models and big data OLAP cubes are designed specifically to meet these challenges.

**In-database analytics** - Though the earliest developments for in-database processing date back several years, recent developments have matured the technology to meet today's data management challenges. In-database processing helps to modernize a data warehouse by building on a database platform that provides parallel processing, partitioning, scalability, and optimization features to maximize analytic functionality. This approach moves processing to the database and eliminates the work of moving large volumes of data to a processing platform.

**GPU databases** - A GPU database uses a graphics processing unit (GPU) to perform database operations. Conventional databases perform their operations using the central processing unit (CPU). The GPU, originally designed for fast and intensive processing to render 3D graphics, is now being used to accelerate the computational workloads of big data and analytics. The parallel processing capabilities

and raw computing power of GPUs deliver exceptionally fast database performance for both relational and non-relational data. When working with large volumes of data and compute-intensive analytics, GPU database performance reduces the need for careful indexing, database partitioning, and data sampling to reduce the size of the data population.

**Snapshots, columnar storage, and virtualization** - A unique combination of source data snapshots, columnar storage, and virtualization offers an innovative alternative to conventional data warehousing. Moving data directly from sources to columnar storage without aggregation or transformation removes the "T" from ETL and eliminates the need for rigid and prescriptive data models. Date and time stamping of incoming data provides the foundation for snapshots and time-variant history. Automated discovery and mapping of relationships in the data—both explicit and inherent—is the foundation for integration. Schema on demand with minimal data modeling offers rapid response for a variety of analytics use cases without compromising warehouse value. Hyper-converged analytics is enabled with all of the data—enterprise data, big data, and external data—available in one place, in its raw form. Query processing uses virtualization techniques to integrate and aggregate on demand at exceptionally high speed. Rethinking data warehousing from the ground up, with attention to purpose before process, has produced technology that supports data warehousing without a fully integrated data warehouse. With multiple deployment options—on-premises and cloud—this may be the solution for those seeking a fresh start in data warehousing.

# Getting Started with Modernization

Data warehouse modernization is a journey, not an event. Done well it is a planned and incremental step-by-step process of moving from warehousing of the past to the future of data warehousing. Plan your journey and navigate the course with these tips in mind:

- Assess your current state of data warehousing. Define your needs and challenges and prioritize them to know which are most pressing and need earliest attention.
- Define your future state of data warehousing. Define and describe your goals for modernization with enough clarity to know when the goals have been achieved.
- Choose the modernization patterns that are best suited to your goals. Don't hesitate to mix and match patterns. You might, for example, want to use data warehouse automation to reverse engineer a legacy data warehouse and then migrate that warehouse to the cloud. Or you might migrate a high-use data warehouse to the cloud and federate other warehouses to break down the silos.
- Plan for the future, not for today. Look ahead three to five years. Planning only for today leads to being outdated before you're fully implemented.
- Expect and prepare for change. The data management world is changing rapidly. Expect that your three- to five-year plan will need regular updating as new architectures and technologies emerge. Be prepared to reevaluate decisions and adjust your plans as new concepts and technologies come to market.
- Execute one step at a time, not "big bang." Then repeat the entire process. After each step your current state will be different, your priorities may have changed, and you may want to refine your future-state thinking. And, of course, the technology will continue to evolve.

# About Infoworks

Over 80% of data lake projects fail to deploy to production because project implementation is a complex, resource-intensive effort taking months or even years. The Infoworks.io agile data engineering software automates and accelerates data analytics projects and has been adopted by some of the largest enterprises in the world. Using a code-free environment, Infoworks.io allows organizations to quickly create and manage data pipeline processes from source to consumption. Customers deploy projects to production within days with fewer people, dramatically increasing analytics agility and time to value.

Infoworks.io leverages our founder's experiences at Google and Zynga building and deploying world-class big data environments to create an autonomous data engine that requires no coding or specialized "big data" skills.

Infoworks.io customers use the Infoworks Autonomous Data Engine platform to implement big data solutions both on premises and in the cloud for:

- Automated Data Lake Creation and Management
- Automated Enterprise Data Warehouse Offload and Migration
- Automated Data Workflows for Business Intelligence and Analytics