Modern Data Integration Paradigms

How Companies Enable Digital Transformation and Data-Driven Decisions

Matthew D. Sarrel
The Bloor Group

This study was prepared by the Bloor Group, an independent market analysis firm, and sponsored by IRI, The CoSort Company. Additional sections at the end reflect the roles that the IRI Voracity platform can play in these paradigms.

Table of Contents

Introduction

Digital Business

Business that blurs the digital and physical worlds. People, business, and things (IoT devices) intertwine and feedback to each other through technology.

Businesses of all sizes and industries are rapidly transforming to make smarter, data-driven decisions. To accomplish this transformation to **digital business**, organizations are capturing, storing, and analyzing massive amounts of structured, semi-structured, and unstructured data from a large variety of sources. The rapid explosion in data types and data volume has left many IT and data science/business analyst leaders reeling.

Digital transformation requires a radical shift in how a business marries technology and processes. This isn't merely improving existing processes, but rather redesigning them from the ground up and tightly integrating technology. The end result can be a powerful combination of greater efficiency, insight and scale that may even lead to disrupting existing markets. The shift towards reliance on data-driven decisions requires coupling digital information with powerful analytics and business intelligence tools in order to yield well-informed reasoning and business decisions. The greatest value of this data can be realized when it is analyzed rapidly to provide timely business insights. Any process can only be as timely as the underlying technology allows it to be.

Even data produced on a daily basis can exceed the capacity and capabilities of many pre-existing database management systems. This data can be structured or unstructured, static or streaming, and can undergo rapid, often unanticipated, change. It may require real-time or near-real-time transformation to be read into business intelligence (BI) systems. For these reasons, data integration platforms must be flexible and extensible to accommodate business's types and usage patterns of the data.

Inherent Challenges of a Changing Technical Landscape

IT leaders have an unprecedented number of tools and techniques at their disposal, many of which are based on complex, open source and distributed technologies. Popular solutions like Hadoop and NoSQL databases are scalable platforms that provide powerful advantages for narrow applications. As such, they are specialized for specific tasks and must be used appropriately. For example, using the Neo4j graph database in a transaction-heavy environment would be like using an exotic sports car to shuttle kids to soccer practice.

Organizations that rely on individual best-in-class databases are struggling to integrate disparate systems in a way that will unlock the insights that are lie isolated and untapped within each. Most data solutions speak their own language with their own APIs and their own design concepts.

IT leaders should therefore try to integrate data across systems in a way that exposes them using standard and commonly implemented technologies such as SQL and REST. Integrating data, exposing it to applications, analytics and reporting improves productivity, simplifies maintenance, and decreases the amount of time and effort required to make data-driven decisions.

Data Integration Platforms and Decision Criteria

It's critically important for decision makers to realize that this complex technology problem requires making optimal decisions up and down the stack. Developers need a single datastore with clearly defined access methodologies in order to realize the greatest value of data-driven initiatives. This is much more than simply firing up a data pipeline and connecting a bunch of tools to it.

There are other important technology selection criteria to consider as well, especially around securely discovering, integrating, migrating, and analyzing big data.

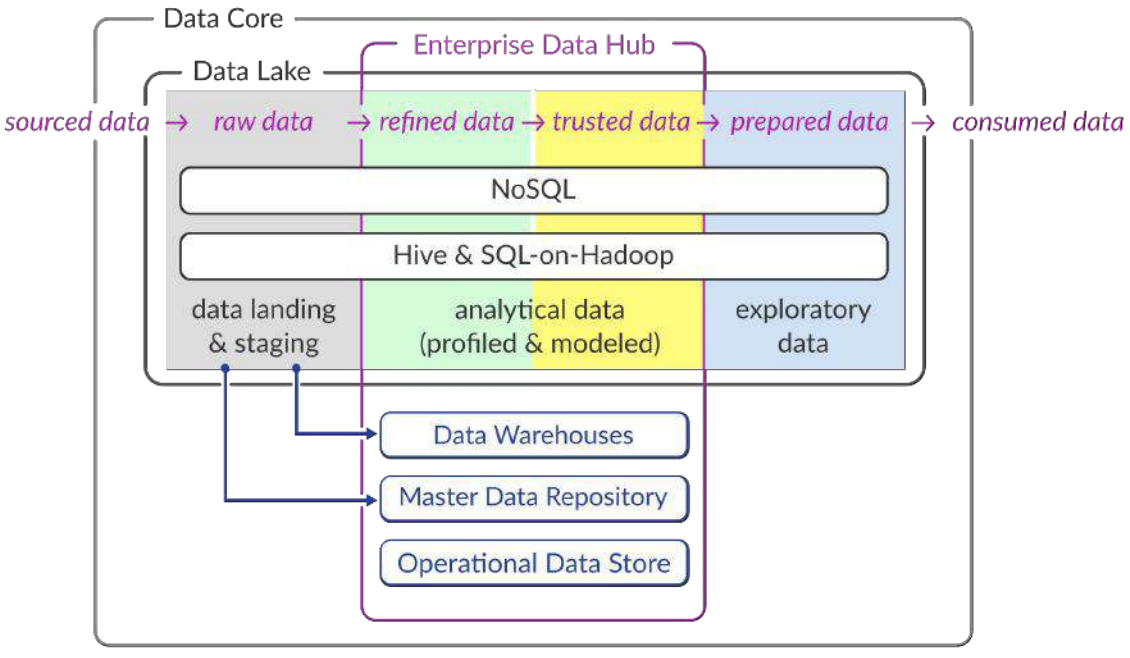
This paper takes a deeper look into common industry data integration and storage architectures that enable data-driven decisions:

- Operational Data Store (ODS) / Enterprise Data Hub (EDH)
- (Enterprise) Data Warehouse (EDW)
- Logical Data Warehouse (LDW)
- Data Lake

The data lifecycle and the data warehouse model shown features the six-stage data lifecycle as a basic architectural principle, as used by David Wells. The data warehouse supports two stages: *refined* and *trusted* data. The data lake exists as a landing and staging area for raw data. It can also contain experimental data stores.

The Future of the Data Warehouse
Eckerson Group

The Data Lifecycle and the Data Warehouse

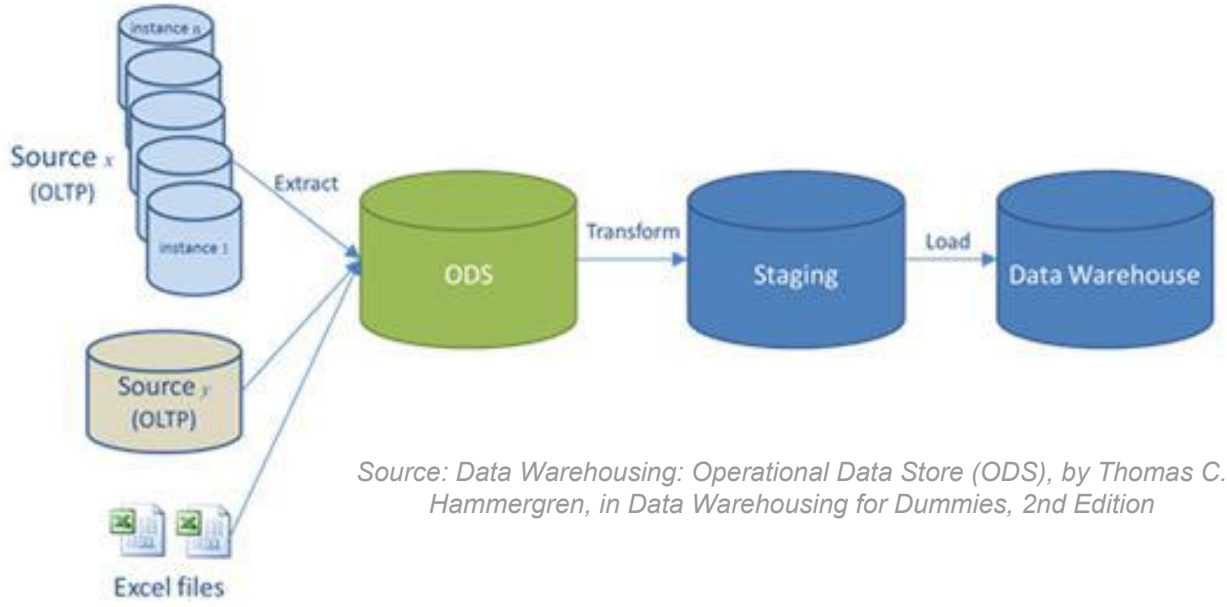


Source: Eckerson Group. *The Future of the Data Warehouse*

Operational Data Store (ODS)/ Enterprise Data Hub (EDH)

An operational data store (or "ODS") is the simplest paradigm for integrating enterprise data. The ODS is a central database (DB) where data that has been integrated from disparate sources undergoes specific functions. The ODS is often a component of a data warehouse (DW).

Data in the ODS can come from batch inputs, and is processed by extract, transform, and load (ETL), and data quality, operations:



Source: Data Warehousing: Operational Data Store (ODS), by Thomas C. Hammergren, in Data Warehousing for Dummies, 2nd Edition

The ODS is also a multipurpose structure that enables transactional and decision support processing. Because its data originates from multiple sources, integration often involves cleaning, resolving redundancy, and checking the data against business rules for integrity.

A key difference between a DW and an ODS is temporality. Unlike in the warehouse, new data coming into the ODS overwrites existing data. That is because the business unit always needs to work with the most current data (versus aggregate data typical of a DW used for analytics); e.g., when a bank needs to cover, notify, and charge a customer whose account is overdrawn.

The ODS in More Detail

Master Data

Sometimes called *reference data*. Refers to critical business data that comes from and supports transaction data, but it is not usually transactional in nature.

The ODS typically provides a non-historical, integrated view of data in legacy applications. It enables the business unit to complete transactional functions and/or operational reporting with current snapshots of data at a specific level of granularity (atomicity).

ODS data is processed through a series of ETL operations to integrate, transform, and comply with a set of standards where data quality and uniformity are the goals. This data is usually kept in a relational database so the business unit can access it immediately. That database has specific constraints, including referential integrity, to make sure the data is relatable.

The ODS database is usually designed for low-level or atomic (indivisible) data, such as transactions and prices, with limited history that is captured "real time" or "near real time," as opposed to the much greater volumes of data stored in the data warehouse, generally on a less-frequent basis. The pace of updates in a batch-oriented DW is usually too slow for operational requirements.

Unlike data in a **master data** store, ODS data is not passed back to operational systems. It may be passed into further operations and onto the DW for reporting. The ODS is an alternative to a decision support system application that accesses data directly from an online transaction processing (OLTP) system.

Unlike a DW, the ODS tends to focus on the operational requirements of a particular business process (for example, customer service). The ODS must also allow updates and propagate them back to the source system. A DW architecture, on the other hand, helps decision makers access and analyze historical and cross-functional (non-volatile) data, while supporting many different kinds of applications.

As the ODS is set of logically related data structures, data exists in an integrated, volatile state and at a non-historical granular level, so operational functions can be performed to meet specific business goals. Because the results of its operations are mission critical, and because it shares data and potential ETL workflows with a larger DW, the ODS must also run with the same data governance and management standards in place enterprise-wide.

When the ODS Makes Sense

To understand the purpose of the ODS and its appropriateness as a data integration paradigm, consider its four primary attributes:

1. Subject-Oriented
   The ODS contains data that is unique to a set of business functions in a given subject area.

2. Integrated
   Legacy application source data undergoes a set of ETL operations, which includes cleansing and transformation processes based on the business' rules for data quality and standardization.

3. Current (non-historical)
   The data in the ODS is up-to-date and shows the current status of data from the source applications.

4. Granularity
   Data in the ODS is primarily used to support operational business functions, and so it must contain the specific level of detail the business requires for those functions to be performed.

The best way to determine if an ODS is an appropriate solution is for business analysts and the data management team to jointly assess the processes involved in completing transactions and providing operational reports. These assessments are most effective if they:

- are performed in a business process management (BPM) framework
- focus on data-dependent functions
- recognize the inefficiencies and ineffective aspects of current or alternative approaches

Through this analysis, and an understanding of what an ODS can do, the team can clearly articulate their issues and requirements. If the business unit's business process management (BPM) analysis reveals transactional or operational issues, missed deadlines, data quality errors in the legacy sources, or an aged or poorly designed supporting application, chances are that an ODS is appropriate. The analysis should also help the team design and use the ODS to meet their specific business goals, while adhering to corporate data governance standards.

ODS Example

360 Degree View of the Customer

Integrating customer and client data, including transaction history and preferences, to create a holistic account of the individual. Leveraging master data, it emphasizes a unique and total store of information of a person.

An ODS in a bank has, at any given time, one account balance for each checking account (courtesy of the checking account system), and one balance for each savings account (per what's provided by the savings account system). The various systems send the account balances periodically (e.g., at the end of each day) to the ODS where functions act on that data to generate alerts, fees, statements, and so on.

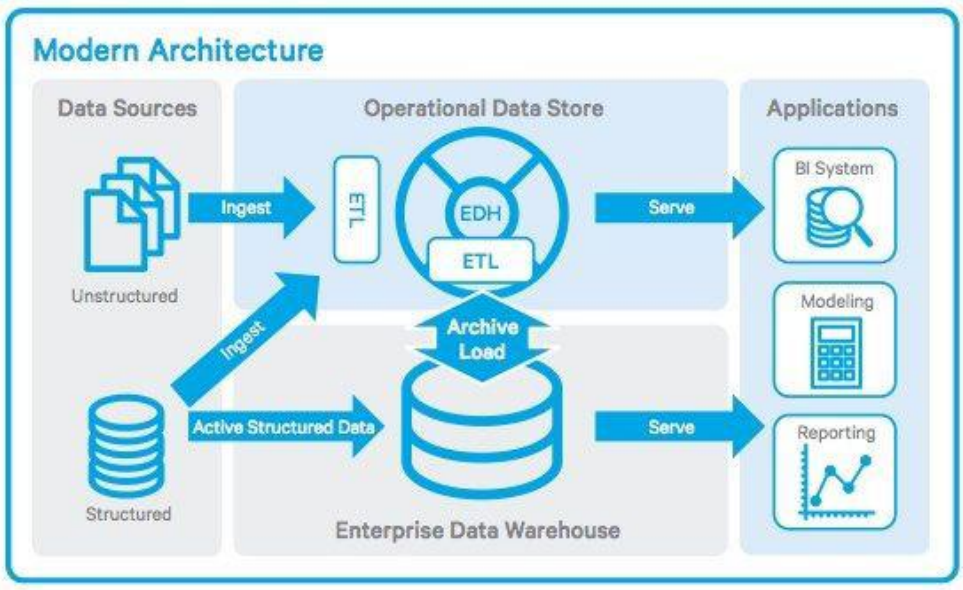
In this case, the ODS user also gets a central and complete point of reference for each customer's profile (such as his/her basic information and account balances). It's not necessarily a **360 degree view of the customer**, since it only speaks to the information and transactions joined in this particular database, but it's still an ODS.

This example compares and contrasts the ODS and DW as well. Here, the ODS is acting as a batch-oriented DW, updating and replacing each datum that resides in it (and adding new data). But it is not keeping a running history of the measures it stores. You can implement this kind of ODS with batch-oriented middleware, reporting and OLAP tools. Or you can use a single platform to connect to the sources, administer the DB(s), do ETL, cleanse, and report.

Modern ODS Manifestations

A more advanced version of an ODS uses the push-pull approach of DW-enabled applications. That allows an informational database to be refreshed in or close to real time. However, as you might imagine, this architecture is harder to implement.

In the era of big data, with newer (e.g., unstructured) data sources and Hadoop processing paradigms, the ODS has also been called an Enterprise Data Hub (EDH). Here is how a Hadoop distribution provider illustrates the EDH:



Source: Cloudera

In this model, the EDH is a more robust form of an ODS because it uses the Hadoop File System (HDFS) as a repository for structured and unstructured data, along with an elastic, multi-node computation environment for very high volumes. Big Industries founder Matthias Vallaey explains the benefit of the EDH in terms of both operational performance and analytics:

> *The implementation of an enterprise data hub (EDH), powered by Apache Hadoop, provides enterprises an ODS that unlocks value by processing and storing any data type at massive volumes—eliminating the need to archive data—while allowing for quick, familiar data access to end users and applications.*
>
> *Operational Data Store: First Step Towards an Enterprise Data Hub*

Enterprise Data Warehouse (EDW)

The data warehouse (DW) or Enterprise Data Warehouse (EDW) is a core component of many effective analytics and BI programs, yet this venerable architecture often has a hard time meeting the new challenges thrown at it by big data. This centrally-accessible repository, and its simpler, singular, limited form called a data mart, are the oldest and still most widely-used solutions for managing data, conducting business analysis and creating reports. The EDW can be integrated with the more basic operational data store (ODS) for additional data operations.

Originating in the 1980s, data warehouse architecture was engineered to facilitate data flow from operational systems requiring analysis of massive accumulations of enterprise data. In the process referred to as ETL, data is extracted from heterogeneous sources (usually on-premise databases and files) into a staging area, **transformed** to meet decision support requirements, and loaded into the warehouse for storage.

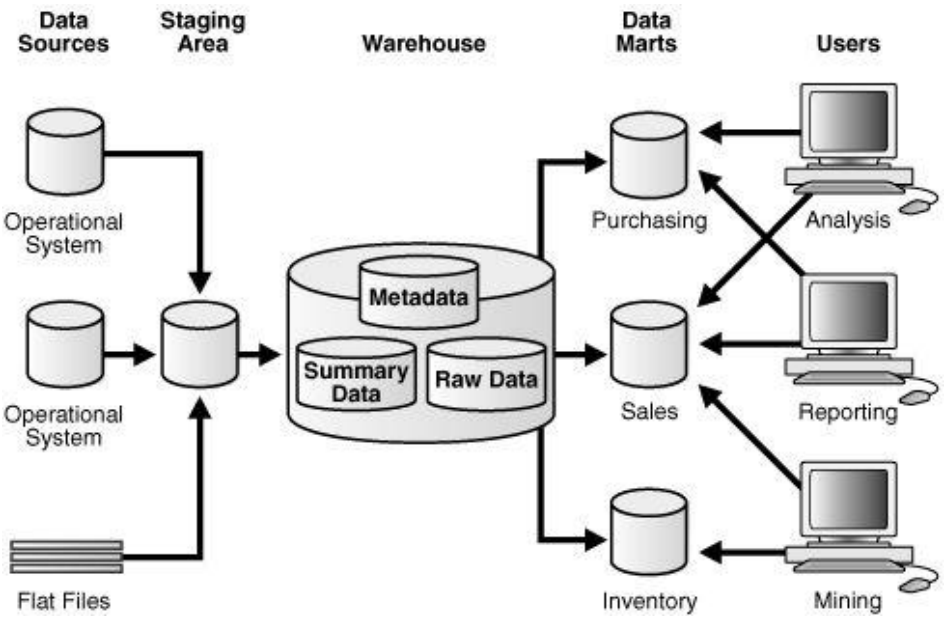
Data Transformation

The process of converting or reformatting data. Typically this is done to change source file types into a file type accepted by a destination system.

The typical EDW environment includes:

- disparate storage and systems that provide the source data
- data integration and staging through extract-transform-load (ETL) processes
- data quality and governance processes to ensure the DW fulfills its purposes
- tools and applications to profile sources, feed the DW DB, and analyze the results

The basic architecture of the EDW has remained more or less as follows:



Traditional data sources are relational DB tables, flat files, and web services; but now CRM, ERP, IoT, NoSQL, social media, web log, public, and other "big data" sources are in the mix.

Unlike source OLTP DBs with normalized tables optimized for complex queries and modeled in E-R diagrams, the DW DB is denormalized for simple joins, and thus faster OLAP queries. Its data models reflect more advanced ***star*** and ***snowflake*** **schema**. They are also considered "nonvolatile" and "time-variant" because they produce the same reports for different periods in time. Modern EDWs and logical data warehouses (LDW) are more volatile.

Data marts are smaller, departmental-level DWs that either use subsets created from the main DW (dependent), or they are designed for one business unit (independent). The operational data store (ODS), which we'll cover separately, is an interim DW DB, usually for customer files.

Star/Snowflake Schema

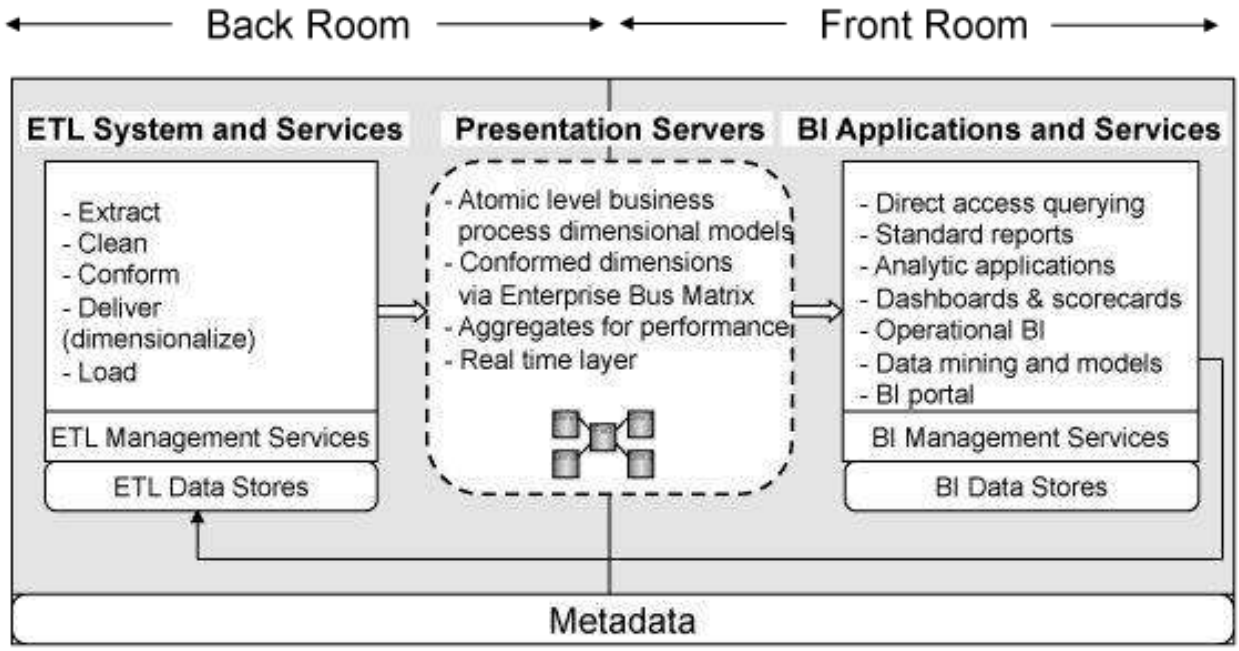
Star schema and *snowflake schema* are ways to organize data marts or entire data warehouses using relational databases. Both of them use *dimension* tables to describe data aggregated in a *fact* table.

Although they store the exact same data, they differ in some critical areas, like normalization and query complexity.

Emil Drkušić
Star Schema vs Snowflake Schema.
Vertabelo

(Slightly) Deeper Dive

Though there are variations of "back room" DW/BI architectures to stage and integrate data (including extract-load-transform (ELT) and hybrids of either), IRI subscribes to the Ralph Kimball ETL convention, with BI data stores in the "front room" and presentation services in between:



Source: *The Data Warehouse Lifecycle Toolkit, Second Edition*

Beyond basic ETL vs. ELT decisions is a long list of other considerations, including the hardware and software systems running in the bottom (DW/ETL), middle (OLAP), and top (BI) job tiers of the EDW. As a threshold matter, EDWs mostly use SQL-driven relational DBs; though with data pushing into petabyte ranges, mainframes, multi-core Unix servers, and Hadoop data nodes are now the norm, along with SQL layers on NoSQL DBs.

In the bottom tier vendor offerings handle ETL and related data delivery issues, including change data capture, migration and replication, and various 'types' of slowly changing dimension updates.

In the middle tier, the choice of **Multidimensional Online Analytic Processing (MOLAP)** or **(Relational Online Analytical Processing) ROLAP** is usually made, where the DB is either multi-dimensional and stores "facet" views (like sales by time) in arrays, or relational where similar results require SQL queries. MDDBs are faster at analytic processing, but RDBs are more common in EDWs, where BI tools feed off them in the top tier instead. Choices of partitioning strategy and normalization form are often made to speed the RDB's queries.

In the top tier, the choices for BI and data mining are virtually endless, and dictated by reporting or analytic (diagnostic, predictive, prescriptive, etc.) requirements. Here, an external data preparation solution removes integration overhead from the BI layer. In addition to creating centralized, reusable data, this approach can speed the time-to-display for several popular analytic and data visualization platforms by up to 20X.

Multidimensional Online Analytic Processing (MOLAP)

Online analytical process that indexes directly into a multidimensional database. Users are able to view different aspects or facets of data aggregates.

MOLAP
TechTarget

Relational Online Analytic Processing (ROLAP)

A form of online analytical processing that performs dynamic multidimensional analysis of data stored in a relational database rather than in a multidimensional database.

ROLAP
TechTarget

EDW Uses and Benefits

A data warehouse is best defined by the type of data it stores and the people who use it. Designed for decision support and BI activities, the data warehouse is separated from the day-to-data online transactional processing (OLTP) applications that drive the core business, thereby reducing contention for both operational transactions and analytic queries.

Craig Mullins
The Benefits of Deploying a Data Warehouse Platform

Data warehouse consultant Craig Mullins delineated that an EDW can:

- track, manage, and improve corporate performance
- monitor and modify a marketing campaign
- review and optimize logistics and operations
- increase the efficiency and effectiveness of product management and development
- query, join, and access disparate information culled from multiple sources
- manage and enhance customer relationships
- forecast future growth, needs, and deliverables
- cleanse and improve the quality of your data

Specific use cases show that DWs and EDWs are used to:

- Assess call, click, and other consumption habits
- Detect insurance fraud or set rates
- Evaluate treatment outcomes and recommend drug therapies
- Manage goods inventories and shipments
- Monitor device/equipment health and service levels
- Optimize pricing and promotion decisions
- Streamline staffing, fleet, and facilities.

A key technical benefit of EDWs is their separation from operational processes in production applications and transactions. Mullins explained that performing analytics and queries in the EDW delivers a practical way to view the past without affecting daily business computing. This, in turn, means more efficiency, and ultimately, profit.

Also, from a financial point of view, the EDW is a relative bargain among data delivery paradigms, especially compared to less open, stable, or governed ones, like appliances, Hadoop, and data lakes. And thanks to competitive tool and talent markets, the barriers to entry have dropped for those who previously found EDWs too expensive or complex to implement.

EDW Evolution

The EDW emerged from the convergence of opportunity, capability, infrastructure, and the need for converting transactional data into information, all of which have increased exponentially in the last twenty years. As related information technologies evolved, many process-oriented business rules were changed or broken to make way for data-driven rules. Processes fluctuated from simple to complex, and data would shrink or grow in an ever-changing enterprise environment.

Now, in the era of big data, many more sources and targets are in play. There are well-known challenges of data volume, variety, velocity, veracity, and value putting pressure on traditional EDWs. These concepts and their consequences are creating shifts in enterprise data management architecture from older paradigms like the operational data store to newer ones, like the Enterprise Data Hub (EDH), logical data warehouse (LDW), and data lake, which were designed to also accommodate more modern data stores and analytic needs.

As a result, some data management experts now consider EDWs to be a legacy architecture, but one still able to perform routine workloads associated with queries, reports, and analytics.

The Bottom Line on EDW

Analytics and BI tools connect to the EDW so that business users and executives can base their decisions on the insights gleaned from that data. The EDW has a long and proven history as a reliable data integration and storage paradigm for enabling analytic insight. However, some businesses have become dissatisfied with this paradigm for large-scale information management due to its overhead.

The time-consuming processes of legacy ETL tools negates the EDW's ability to inform real-time, data-driven decisions. More than a few enterprises have discovered that there aren't enough hours in the day to run timely batch ETL and backup. In addition, companies relying on existing data warehouses are growing reluctant to maintain their relatively high hardware, software, and design/support costs.

For these reasons, the EDW architectural model has evolved in recent years from the initial focus on the physical consolidation and management of data to incorporate more logical and virtual paradigms. EDWs are well-suited for historical data analysis and batch-based loading, processing and reporting on transaction-based workloads.

Logical Data Warehouse (LDW)

The traditional or enterprise data warehouse (EDW) has been at the center of data's transformation to business intelligence (BI) for years. An EDW involves a centralized data repository (traditionally, a relational database) from which data marts and reports are built. However, the EDW paradigm of physical data consolidation has been shifting in recent years to a more *logical* one.

The logical data warehouse (LDW) also serves analytic ends in a managed information management architecture, but accesses and integrates data in place (i.e., virtually). Specifically, a logical data warehouse is an architectural layer that sits atop the EDW's store of persisted data. The logical layer provides (among other things) mechanisms for viewing data in the warehouse store (and elsewhere) without relocating and transforming data ahead of time.

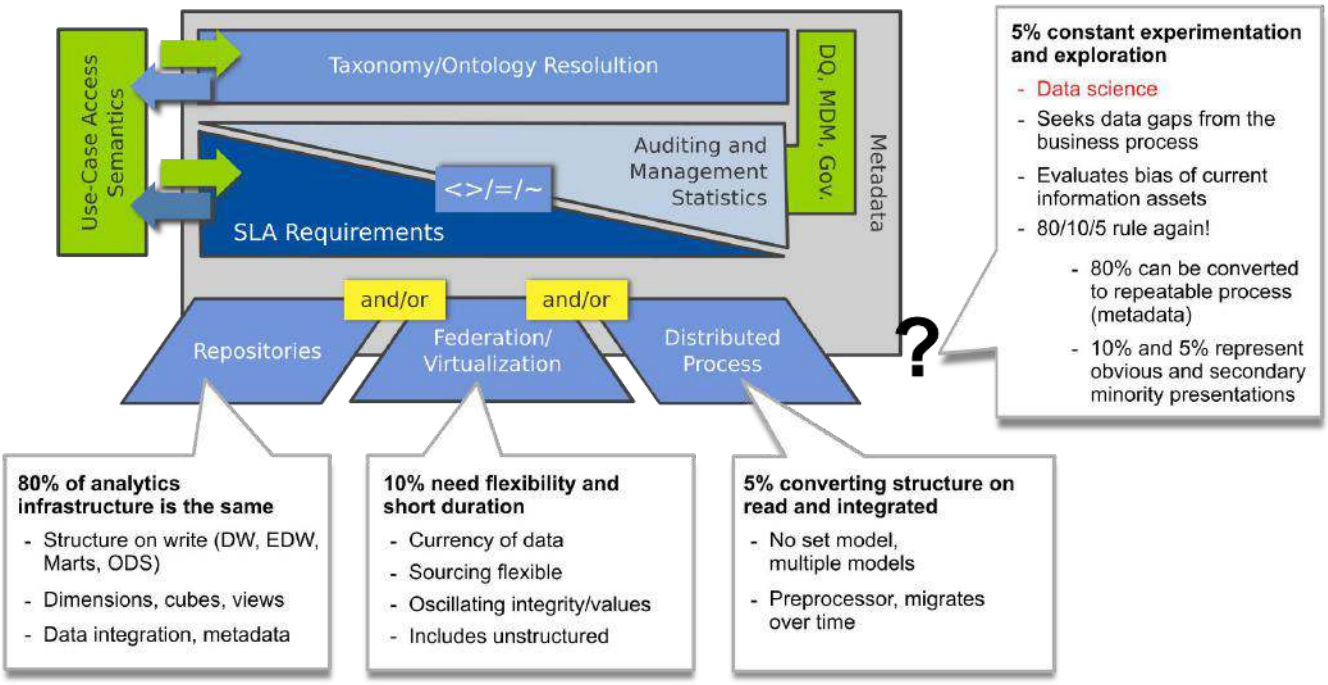
LDW #trending

As more and more businesses realize the value of data-driven decision making, data analytics and BI programs have grown in importance. This change in business analytics requirements, plus the additional demands created by a need for real-time analytics, has created a more modern data virtualization paradigm driven by the demand to easily access and federate data.

To complement the traditional data warehouse, the logical data warehouse (LDW) adds an architectural layer to view enterprise data without having to relocate and transform it beforehand.

LDWs are capable of retrieving and transforming data in real time, and producing fresher data without the limitations imposed by the pre-built structures of traditional DWs

Figure 1. Understanding the 80/10/5 Rule for SLAs



DW = data warehouse; EDW = enterprise data warehouse; ODS = operational data store; DQ = data quality; MDM = master data management; Gov. = governance

Source: Gartner

Source: *Avoid a Big Data Warehouse Mistake by Evolving to the Logical Data Warehouse*. Gartner. December 2014, Gartner Foundational April 2016, G00252003

Several trends are contributing the EDW-to-LDW evolution, including:

1. Growth of data volume, variety, velocity, and veracity in the big data age of Web 2.0, mobile apps, and the Internet of Things.
2. New data silos and formats that legacy ETL/ELT and BI tools cannot readily consume.
3. Analytic demands of digital businesses that need actionable information in real time.
4. Lack of data quality, compliance, and reliability in an ungoverned data lake or BI tool.
5. cloud-based deployments of analytic environments (which enable LDWs)

As LDWs become more prevalent, there will be more disruption in the data warehouse market as well as increased expectations for the LDW. In today's highly-competitive business environment, the digital transformation race is on and those organizations that can get out in front are likely to enjoy the competitive advantage of making timely data-driven decisions for many years.

There are many cloud services available that decrease the amount of time needed to get analytics programs up and running. As the number of organizations looking to adopt cloud analytics grows, the potential exists to shift the competitive landscape of entire industries and leave some conventional companies behind.

Workloads that are well-suited for LDWs include data mining, real-time BI and analytics, and combining data housed in disparate systems to provide a complete picture of a specific aspect of business, like a customer 360 program that integrates ERP, CRM, web and mobile apps, online marketing, social media marketing and other data sources.

> ***LDWs Will Become the Heart of the Modern Data Management Architecture***
>
> *If there were ever a moment that would force IT to modernize its data and analytics architecture, it is the IoT. The Volume, velocity, and variety generated are creating a deluge with which we are forced to reckon. The demands brought on by the IoT require that technical professionals build the data management and analytic architecture to accommodate changing and varied data and analytic needs. The environment must accommodate not only the traditional static, structured data and analyses, but also allow for access to less well-defined data using more advanced, iterative analytical techniques.*
>
> *Gartner, Planning Guide for Data Management Analytics; Carlie Idoine,*

Certain “administrative issues” involved in EDW planning and investment are also well known, and further motivate movement to LDWs in some cases. For example:

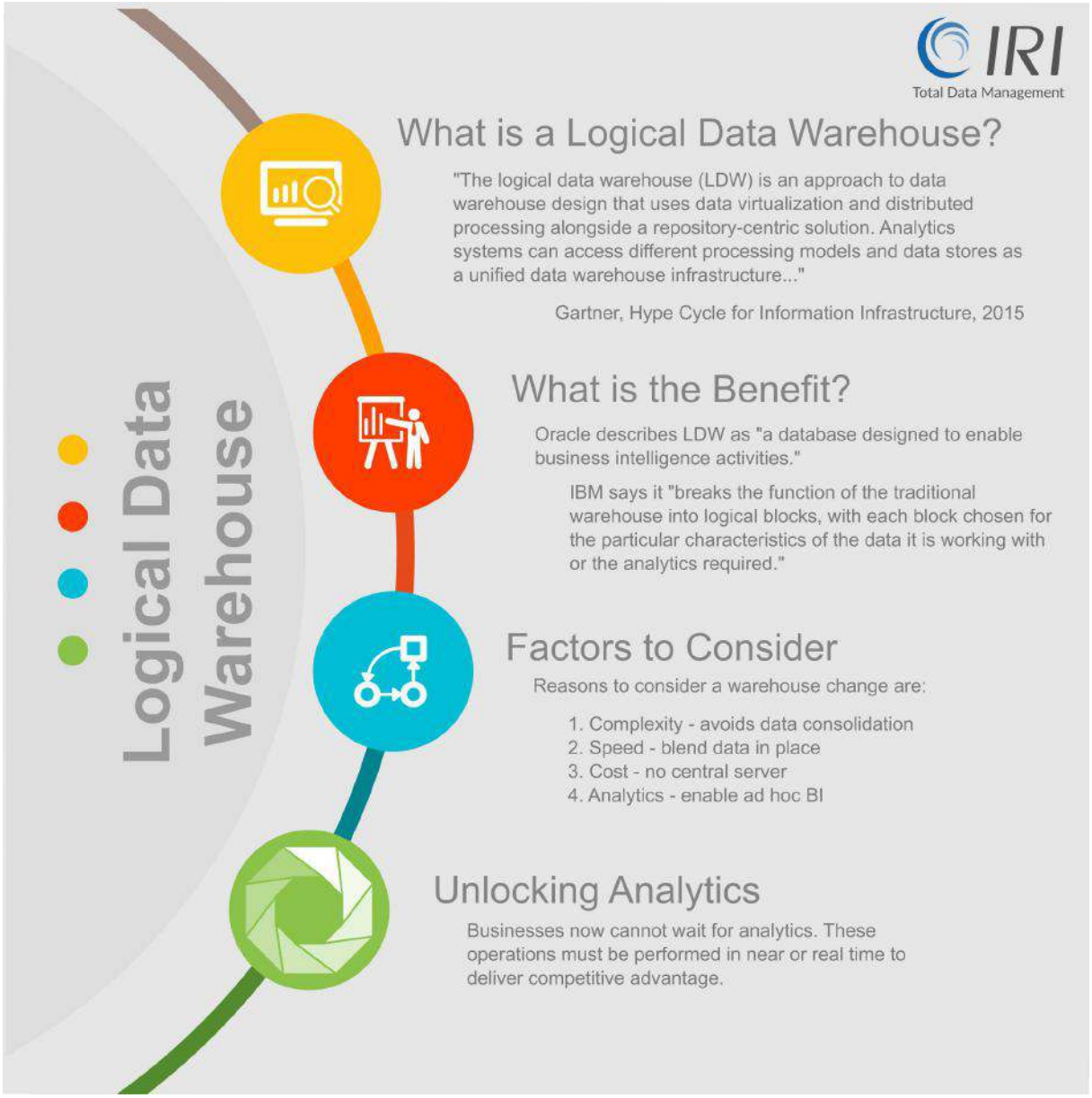
1. An EDW can host enough data to provide decision makers with the information they need to spot trends and drive strategies, but by the time the EDW is implemented, it may no longer meet current business needs.
2. Dragging all that data into a central staging area before leveraging it, rather than leveraging it logically where it lives, is not always necessary, and much less efficient.
3. EDWs are also encumbered by skill gaps and the costs of mega-vendor hardware and software; i.e., legacy ETL tools building and running those jobs are complex, slow, and expensive (even before a separate BI platform enters the picture).

Regardless of whether an EDW or LDW is used, data from transactional DBs, CRM systems, line-of-business applications, and other sources must be cleansed and transformed before, or within, the warehouse to ensure reliability, consistency, and accuracy downstream. Thus, careful thought must go into how data is extracted, transformed and loaded (ETL), actually or virtually.

Even though “small” structured data still drives most business decisions, the growing mess of unstructured data is what many business users and data scientists want to cull for decision value. So modern information architecture solutions must be flexible, nimble, affordable, and relatively painless to implement in order to work. Those qualities are everything the EDW is not, even though it still provides a vital purpose.

LDW Details

The Logical Data Warehouse (LDW) is a newer data management architecture for analytics that combines the strengths of traditional repository warehouses with an alternative data management and access strategy. Unlike a Virtual Data Warehouse (VDW) designed to work with multiple relational databases across a virtual network, the LDW is better positioned to deal with big data coming in from multiple silos in structured and unstructured formats.



The LDW has also been compared to the data lake (next chapter), a repository for storing massive amounts of data, usually within a Hadoop infrastructure. The data lake can be particularly useful for handling the distributed processing component of the LDW. That said, differentiating the LDW from the VDW or the data lake does not give us a definitive picture of the LDW. In fact, arriving at that picture is no easy task because the LDW, as a whole, is as much a conceptual effort as it is a physical implementation.

Perhaps the best way to view the LDW is as a logical structure that's defined by the sum of its parts. Those parts are the EDW, cloud services, Hadoop clusters, data lakes, and other elements, some of which include their own capacity to virtualize data and distribute processing. There is no one architecture that defines how the LDW should be built. It is changeable and malleable, with the essential ingredients necessary to handle all the enterprise data present.

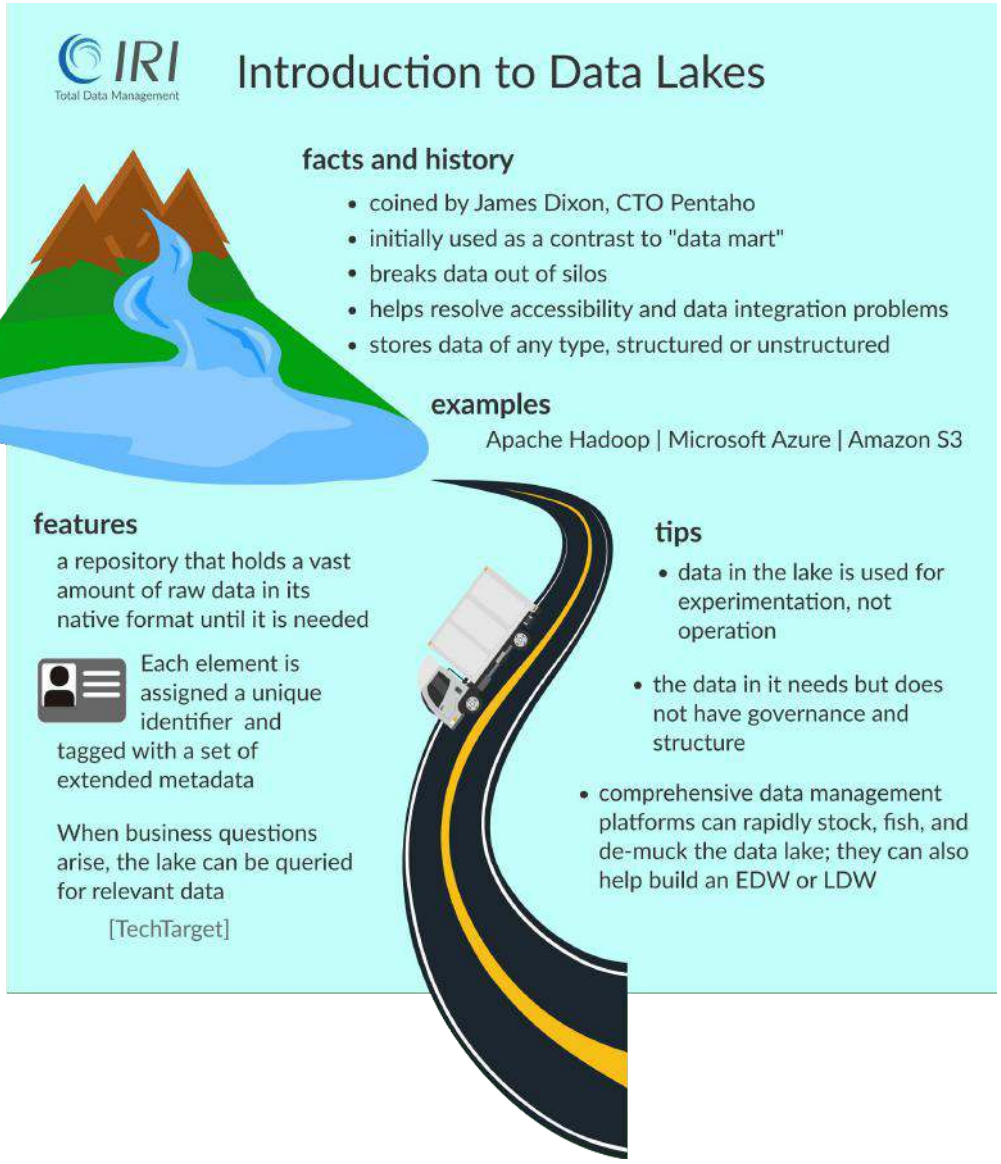
Data Lake

The contents of the data lake stream in from one or more sources to fill the lake, and various users of the lake can come to examine, dive in, or take samples.

James Dixon

A new and increasingly popular approach to working with big data stored on-premises or in the cloud is the enterprise data lake. Despite being the newest option, this data repository method can be considered a more natural approach to the data gathering, storage and analysis problem, since vast amounts of raw data are stored in native format until needed.

The term data lake was coined by Pentaho CTO James Dixon. He used the term to compare data that was cleansed, packaged, and structured — like what was found in a data mart (or bottled water) — to data in its more natural state (in a large natural body of water). Its purpose is to be an environment for gathering and storing data that will be used for experimentation.



Gartner refers to the data lake as “a collection of storage instances of various data assets additional to the originating data sources. These assets are stored in a near-exact, or even exact, copy of the source format.”

Thus, the data lake is a single store of enterprise data that includes both raw data (which implies an exact copy of source data) and transformed data used for reporting and analytics. Some want the data lake to replace the traditional data warehouse, while others see it as more of a staging area to feed data into existing data warehouse architectures.

Data Lake Architecture

A data lake uses a flat architecture such as a file system for storage. This is in contrast to the data warehouse, which stores data in a database or hierarchical file system.

Data Warehouse	vs.	**Data Lake**
Structured, processed	**Data**	Structured / semi-structured / unstructured / raw
Schema-on-write	**Processing**	Schema-on-read
Expensive for large data volumes	**Storage**	Designed for low-cost storage
Less agile, fixed configuration	**Agility**	Highly agile, configure and reconfigure as needed
Mature	**Security**	Maturing
Business Professionals	**Users**	Data scientists *et. al.*

Many companies are launching data lake initiatives, perhaps as a component of their first major Hadoop implementation. As Hadoop is an open source data management platform, many are attracted to its promise of decreased expense and increased ecosystem. There is no doubt that Hadoop brings immediate value as a "Data-as-a-Service" platform. However, like many other promising platforms, there are inherent problems with using Hadoop for building the enterprise data lake.

Some of the crucial elements required for optimal performance and data storage are to regularly sectionize and streamline Hadoop-based data, keep the data lake clean and well-organized, and periodically eliminate irrelevant and unusable data. These are not trivial tasks.

Benefits of the Data Lake

Ideally, data lakes can help a business achieve data storage and analytics goals by offering great value and flexibility to business teams. With an adequate amount of foresight and planning, as well as strong data governance practices, organizations can automate data lake initiatives to handle anticipate size and scope, and thus maximize their value towards making data-driven business decisions.

Data lakes are being used successfully for data mining using ad-hoc queries to fully explore hypotheses in real time across multiple data types and to analyze streaming data alongside historical data (a common need to maximizes results from IoT implementations). Data lakes are growing in popularity as solutions for off-loading historical data from other systems, such as a data warehouse or OLTP system because their flexible nature allows them to run on cheap storage either in the cloud or on-premise.

Yet data lakes are not without challenges, including knowing the data they contain, who is using that data, and how they're using it. Data governance is critically important to successful data lake implementations. Businesses also require centralized or shareable metadata that persists and can be readily modified. The ability to automate the processes that prepare and report on data is also critical to obtaining timely insight.

Data Lake Governance and Efficiency in Focus

There are naysayers who relegate the data lake to a mere notion, particularly since many organizations are unsuccessful with their deployments. Cambridge Semantics CTO Sean Martin said, "We see customers creating big data graveyards, dumping everything into HDFS and hoping to do something with it down the road. But then they just lose track of what's there."

As with any major data-driven initiative, the data lake must be sold across the enterprise. Data lakes absorb data from a variety of sources and store it all in one place, and by definition, without the usual requirements for integration (like quality and lineage) or security. Someone must be accountable for governance. Data Vault inventor Dan Linstedt warns, for example:

> *Users of self-service BI tools trolling the lake have to be governed. Think about who gets to use which tool, who gets to log in where and access what data, or who can open a spreadsheet and upload data directly to Hadoop, and then make it available to the rest of the enterprise. That can be a serious problem.*

David Weldon's April 2017 article in Information Management magazine, "Many Organizations Struggling to Manage Lakes," affirmed the issue in this quote from Zaloni's CEO Ben Sharma:

> *Perhaps the biggest challenge organizations are facing is "finding, rationalizing and curating the data from across an enterprise for analytics solutions … the ability to easily access data, refine data and collaborate on data needs continues to be a large roadblock for many analytic applications.*

Governing data in a lake is of the utmost importance. Furthermore, it is critical to address veracity, security and metadata lineage issues, to name a few. For more information, see De-Mucking the Data Lake in the IRI section.

The other issue to consider is performance. Most tools and data interfaces cannot ingest, process, or produce information in an unmanaged lake as well as data in fit-for-purpose (e.g., query optimized) environments. Thus, consistent semantics and an engine like CoSort will help.

Preparing for the Future

Big data keeps getting bigger, faster, and flows in many different formats from increasingly more sources. This is driven equally by new advancements in technology and escalating business requirements. The Internet of Things (IoT) is transmitting ever-increasing amounts of information and this is a strong driver of storage size and architecture requirements. Many new data storage options rely on the cloud to achieve performance and scale. Yet, data storage is merely the tip of the iceberg when it comes to the business and IT demands created by today's demanding data management and analytics programs.

The magnitude of the problem grows dramatically when IT departments start building overly complicated multi-layered technology stacks out of what could amount to dozens of different software packages. Freedom to choose the best technology is a mixed blessing because each new point solution makes building a fully integrated data storage architecture that much more arduous.

The rapidly shifting big data environment requires support for a variety of data types as well as full-functioned APIs and support for common languages, no matter how specialized they are. Finally, the requirement to run on any infrastructure- cloud, on-premise or both – can help future-proof data management and analytics platform decisions.

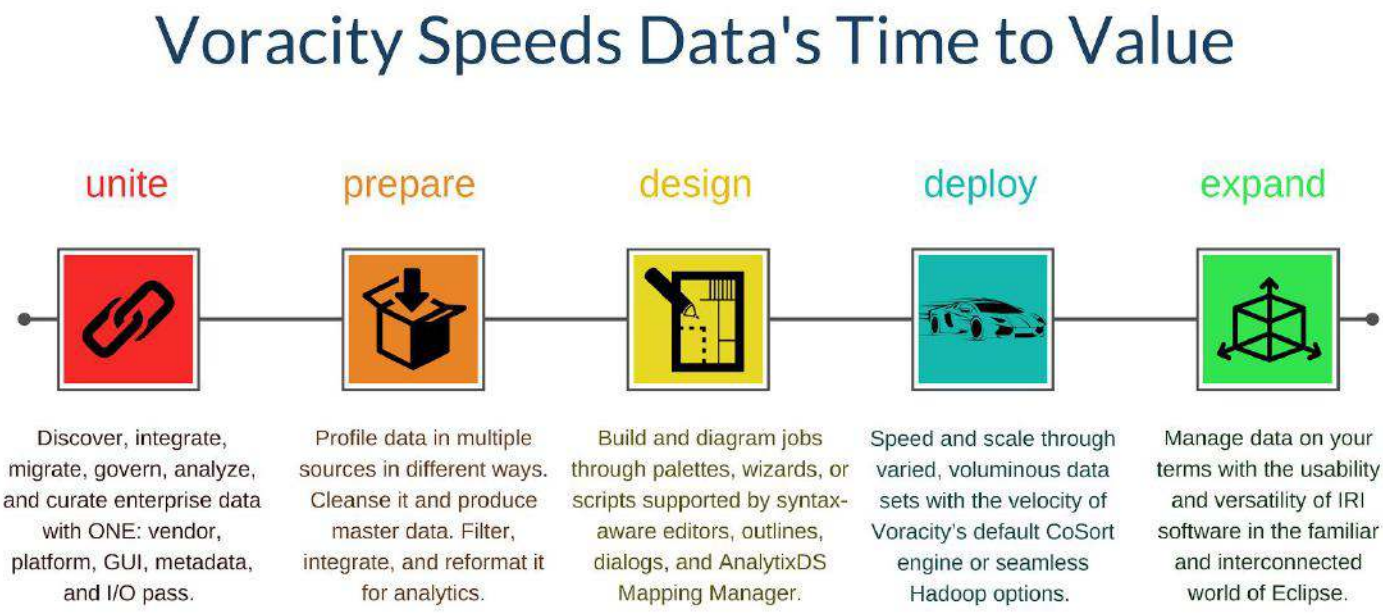
IT leaders empower organizations by simplifying the stack and anchoring it on top of a single centralized platform for data discovery, integration, migration, governance and analytics that is secured and tightly managed by comprehensive policy.

Voracity's Role in Data Integration

Voracity is a comprehensive data management platform product. Powered by CoSort or Hadoop engines, Voracity performs and consolidates the discovery, integration, migration, governance, and analytics of data big and small.

Voracity is designed to be a central marshalling area, built on Eclipse, that consolidates the key activities of the data management lifecycle, while supporting present and future data:

- source volumes through multiple engines
- variety and velocity through multiple APIs and brokers
- security concerns through PII discovery, masking, and auditing
- veracity issues through multiple cleansing and enrichment features.



The pages that follow describe the ways in which data, BI, and data warehouse architects can leverage Voracity in each of the data integration paradigms described in this white paper.

Voracity and the ODS/EDH

IRI Voracity is an end-to-end data management platform built on Eclipse and powered by IRI CoSort or Hadoop engines for data discovery, integration, migration, governance, and analytic operations. As an infrastructure for connecting to, integrating, and managing data sources and DBs, Voracity is thus a central, ergonomic place to build and run an ODS or EDH.

In a traditional ODS, Voracity will:

- discover and integrate both relational and flat-file **sources**
- optimize the ETL components as needed
- administer source and operational DBs via Eclipse DTP
- cleanse and mask the data
- create multiple targets in different formats at once, including: updates to the ODS (and DW) database, custom reports, and hand-offs for BI and analytic tools.

In addition to fast ETL and other data management activities, Voracity's Eclipse IDE, IRI Workbench, also supports development and execution of SQL and 3GL applications.

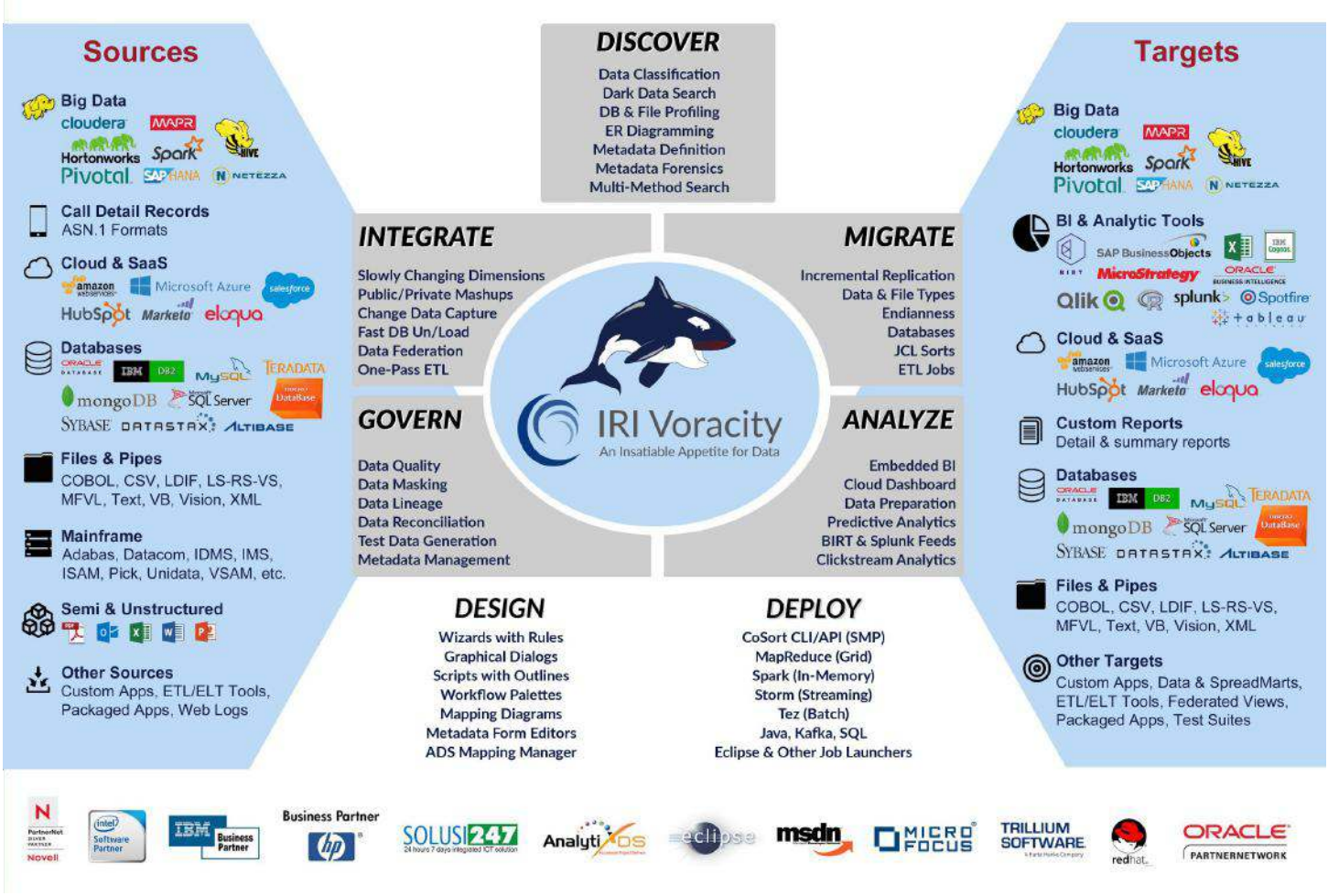
The table below reflects the merits of a virtual approach to data warehousing, and what the Voracity data integration platform specifically provides:

Benefit	Description	Implementation
Access More Data	Business users want quick access to new information, from inside and outside the enterprise, while using existing tools and applications. An EDH allows enterprises to ingest, process, and store any volume or type of data from multiple sources.	Connect to legacy and modern data sources from the same Eclipse UI, and perform multiple types of data discovery (including profiling and classification), to help Voracity find the data and apply field rules for integration, masking, and business operations.
Optimized Data Processing	ETL workloads that previously ran on storage systems can migrate to the EDH and run in parallel in order to process any volume of data rapidly. Optimizing the placement of these workloads frees capacity on traditional systems, allowing them to focus processing power on business-critical OLAP, reporting, etc.	Migrate existing (or design new) ETL workflows in Voracity, and execute them in parallel with the multi-threaded CoSort engine, OR seamlessly in HDFS using MapReduce 2, Spark, Storm or Tez. This allows for a choice of scalable processing of high volumes, in either existing or new Hadoop file systems with the same code.
Automated Secure Archive	An EDH offers a secure place to store all your data, in any format and volume, as long as it is needed. This allows you to process and store data without archiving it, and thus re-use it when needed. This puts historic data on-demand to satisfy internal and external analytic needs.	Direct Voracity to pull (push) data from (to) secure repositories, controlled by access code specified in the connection registry, or other authorization task (block in the job flow). This could mean big data secured in cloud repositories, like S3 or smaller data, or metadata assets in systems like EGit.

Enterprises that go this route will see a strong ROI, especially if commodity hardware is used. The EDH can free up existing database licenses and servers for other uses, increase the volume and variety of data collected, and retain that data in active (not archived) storage. By gaining the flexibility and scalability that is limited in a traditional ODS, EDH users can view operational data processing in a new way, one with more possibilities.

Voracity and the EDW

The IRI Voracity data management platform supports traditional EDW architectures, as well as operational data stores, LDWs, and data lakes. Voracity is powered by the multi-threaded IRI CoSort transformation engine by default, or by Hadoop MR2, Spark, Spark Stream, Storm, or Tez engines for data in HDFS. Both use the same metadata and Eclipse IDE for job management; engine choice is just a (seamless "map once, deploy anywhere") click-to-run option.



Voracity handles a wide range of data sources and targets, and addresses a number of data warehouse-related requirements, including: data profiling and classification, ETL diagramming, plus wizards for slowly changing dimensions, change data capture, pivoting, and master data management. Scripts support complex transformations, data quality, and elaborate reporting. All of its design and deployment facilities are exposed in the one GUI.

Governance-minded data warehouse architects should appreciate Voracity's data discovery and protection wizards, which automate PII data identification and masking. Its subsetting and test data generation wizards facilitate the database and EDW prototyping as well. Metadata management is available through cloud-enabled asset repositories, with graphical lineage impact analysis through AnalytiX DS.

For users of existing DB, BI, and DW software — and ETL tools in particular — Voracity can either accelerate or replace them. For example, Voracity engines can be called into existing workflows to optimize unloads, transforms (especially sorts, joins, and aggregations), and loads (through flat-file pre-sort). Alternatively, AnalytiX DS software can automate a re-platforming process by converting most of the code in a legacy ETL tool into the equivalent jobs in Voracity. This effort can be undertaken and validated before any Voracity costs are incurred, but once complete, would free up hundreds of thousands of dollars and CPU cycles no longer needed.

The Voracity platform was launched, in part, to resolve the performance problems of traditional ETL tools through better engines for each step and through task consolidation.

Voracity and the LDW

The LDW Bottom Line

Faster insights and smaller footprints are possible through the federated query environment of the LDW. IRI Voracity is an ideal LDW platform technology because it connects to and rapidly integrates numerous data sources in place. It combines the key governance and analytic activities that businesses need.

IRI Voracity plays into the LDW as any data integration platform would. It can rapidly and reliably extract, load, and transform data components that reside in the traditional EDW architecture. But Voracity can also discover data accumulating in the storage repositories, effectively integrate that data with data from other sources, and also govern, protect, and serve up results for analysis. It creates 2D reports, feeds BIRT in Eclipse, or prepares hand-offs to Business Objects, Cognos, Microstrategy, OBIEE, QlikView, R, Splunk, SpotFire, and Tableau.

Voracity can also include a self-service BI tool (cloud dashboard) with the platform. The collocation of, and metadata integration between, a traditionally managed DW infrastructure with a business user's sandbox combines data governance with rapid-query mashups and 'what if' analyses.

The table on the next page reflects the merits of a virtual approach to data warehousing, and what the Voracity data integration platform specifically provides.

LDW Platform Components	What Data Virtualization Does	What Voracity Delivers
Repository Management	supports a broad range of data warehouse extensions	full management (sharing, lineage, version control, and security) for data and metadata in asset hubs, like EGit, SVN, and CVS
Data Virtualization	virtually integrates data within the enterprise and beyond	ergonomic, cross-platform connection, discovery, integration, migration, governance and analytics of heterogeneous data sources; i.e., federated queries in SortCL syntax
Distributed Processes	integrates big data sources such as Hadoop, and enables integration with distributed processes running in the cloud	ingests and targets HDFS and Hive sources, direct or via ODBC, and can optionally execute IRI data transformation, masking, and test data generation programs in MapReduce 2, Spark, Storm, or Tez (vs. using only CoSort SortCL)
Auditing Statistics and Performance Evaluation Services	provides the data governance, auditability, and lineage required	discovers and defines metadata, unifies and buckets, finds and masks sensitive data, validates, cleanses and standardizes data, tracks lineage, and secures metadata in a hub
SLA Management	scalable query optimizers and caching delivers the flexibility needed to ensure SLA performance	source-native API hooks or IRI FACT extract data into memory for consolidated CoSort (or Hadoop) transforms and reports, stream-feeds BIRT or Splunk, or preps a new set for R or other analytic tools
Taxonomy / Ontology Resolution	provides an abstracted, semantic layer view of enterprise data across repository-based, virtualized and distributed sources	automatically defines simple, Eclipse-supported metadata (DDFs) for all flat-files and JDBC connected sources, as well as COBOL copybooks and values found in dark data discovery
Metadata Management	leverages metadata from data sources, as well as internal metadata needed to automate and control key logical data warehouse functions	all jobs are based on SortCL syntax, which incorporates the above data definition file (DDF) layouts directly, or by reference, to their reusable/central location

Voracity and the Data Lake

IRI Voracity facilitates the population, exploitation, and governance of data lakes in much the same way it would integrate and manage data in the other paradigms. It takes a streamlined approach to semantic consistency through a metadata management system that automates creation and maintenance of a common, self-documenting syntax with multiple integrations for lineage, security, and version control.

The platform's built-in task scheduler automates common data lake tasks such as integration, cleansing, masking, reporting and any other job organizations require. Another way to look at these tasks might be in three phases:

1. **Stocking** the data lake -- population and staging
2. **Fishing** the data lake -- integration and analytics
3. **De-Mucking** the data lake -- data cleansing, masking, and governance

Stocking the Data Lake

Data enter the lake from various sources, including structured data from files and databases (rows and columns), semi-structured data (ASN.1, XML, JSON, etc.), unstructured data (emails, documents, and pdfs), or possibly images, audio, and video … thereby creating a centralized store for all forms of data.

IRI Voracity is a data connection and curation platform that can be used to populate a data lake by connecting to, profiling, and moving data in different **sources** into the lake, including Internet of Things (IoT) and web logs.

During or after movement, you can select, transform, reformat, and report on data from those disparate sources using jobs defined by scripts, wizards, dialogs, or diagrams in Eclipse. You can use Voracity to govern and test-analyze data in the lake, and to move data out of the lake.

Fishing the Data Lake

Once you've defined the lake's location and what to pump into it, spend some time now and again to see what's currently in it. Consider what experiments can be run on that data. Use your own data discovery tools — or Voracity's flat-file, ODBC, and dark (document) data search, statistical, and relationship checking and diagramming tools — before "testing the water."

Think of this aspect of Voracity as sonar, where you're trying to find different kinds of data and at different depths (various sources in the lake). Voracity discovery tools classify data, and allow you to fuzzy-search for values from, ODBC and file sources. They also search those and unstructured sources for: explicit strings, values conforming to canned or custom RegEx patterns, and values in a set (lookup) file. Those tools are actually free, since they only require Voracity's GUI (IRI Workbench), not an underlying CoSort or Hadoop transformation license.

After identifying data in the lake that looks worthwhile, even data scientists can struggle deriving value from it without the benefit of semantic consistency or managed metadata. It is much harder to manipulate or analyze data without them. Voracity wizards auto-create metadata for the collections within, and build ETL, federation, masking, reformatting, and/or reporting jobs that filter relevant data from the lake, transform it into useful information, and display it.

Voracity manipulates data with the CoSort engine (by default) or, optionally, in Hadoop with the same metadata. Plugins to Voracity's Eclipse "Workbench" GUI also run: Python, R, SQL, Java, shell scripts, SQL procedures, and C/C++ or Java programs. These tools enable you to do more with lake-related data and apps in the same GUI.

De-Mucking the Data Lake

Recall that a key problem with data lakes, as with real lakes, is that people don't know what's in them, or how clean they are. In nature, unknown things in the water can kill the ecosystem. Unknown data dumped into a data lake can kill the project. Dan Linstedt again advises that:

> *Users of self-service BI tools trolling the lake have to be governed. Think about who gets to use which tool, who gets to log in where and access what data, or who can open a spreadsheet and upload data directly to Hadoop, and then make it available to the rest of the enterprise. That can be a serious problem.*

In short you have to have enough trust in the data to trust your analysis. So it's better to know and manage what's in the water. If you use Voracity, you can discover, integrate, migrate, govern and analyze data in the lake — or prepare test or production-ready targets for other architectures, like a data warehouse, mart, or ODS — all within a managed metadata infrastructure.

You also want to be able to dredge the data lake clean, at least as much as you can, through various data cleansing operations. You can use Voracity to improve data quality in the lake in these ways:

- *Find* - discover, profile, and classify data from a quality standpoint
- *Filter* - remove or save conditionally selected or duplicate items
- *Unify* - data found by fuzzy match algorithms and set probabilities
- *Replace* - data found in pattern searches with literal or lookup values
- *Validate* - identify null values and other data formats by function
- *Regulate* - apply rules to find and fix data out of range or context
- *Synthesize* - custom composite data types and new row or file formats
- *Standardize* - use field-function APIs for Melissa Data or Trillium

With less garbage in the lake, less garbage will come out in your analytic results, and the water will be cleaner for everyone else, too.

Also from a governance perspective is the issue of finding, classifying, and de-identifying personally identifiable information (PII) in the data sets. Voracity addresses these problems as well, and offers a wide range of rule (and role)- based encryption, redaction, pseudonymization and related data protection functions that can be applied ad hoc or globally to like columns.

Other Conservation Programs

For advanced information architects, Linstedt advocates combining Data Vault with Voracity:

> *As far as IRI is concerned, I like their solution because we can govern the end-to-end processing in a central place. With that governance comes the ability to manage. Wrap the Data Vault architecture into that mix, and all of a sudden you have standards around your IT, data and information processes, and around the data modeling constructs that are behind the scenes of a future warehouse iteration of this data.*

Helpful administrative management should also feature centralized or shareable metadata that persists and can be readily modified. And if you can automate processes that prepare and report on data, then you can leverage process repeatedly for what-if analysis and thus get to improved results sooner.

Voracity's approach to metadata management is simplified by virtue of its automatic creation, self-documenting syntax, hub support in Eclipse systems like Git for lineage, security, and version control. For more advanced metadata management and automation, Voracity users can leverage a seamless bridge to the AnalytiX DS graphical lineage and impact analysis environment. Built-in task scheduling allows you to sequence — and fine tune the repetition of — integration, cleansing, masking, reporting, and/or other jobs you might want to run on lake data.

A data lake can be a helpful place to test new theories about data now in silos. So stock it, mind your visitors, and see what good can surface from the muck.

The Bloor Group

The Bloor Group

Austin, TX

+1.512.426.7725

info@bloorgroup.com

www.bloorgroup.com

The Bloor Group is an independent research firm that provides objective, high-quality analysis of enterprise technology products, services, and markets via new media outlets and traditional research methods.

The company focuses on quality content and research. It produces webcasts and publications that analyze the enterprise software industry. Regularly publishing meaningful commentary and research, the Bloor Group prides itself on the independency and transparency of its work.

Matthew Sarrel is a Certified Information Systems Security Professional (CISSP). He is currently a Contributing Editor to PCMag.com and was a Technical Director at PC Magazine Labs where he led all testing conducted by the Applications, Enterprise and Development Software, OS and Utilities, and Network Infrastructure teams. Other publications for which he writes or has written include eWeek, TechWeb, Intelligent Enterprise, Information Week, InfoWorld, IGN, GigaOm Research and YRB Magazine.

Matthew D. Sarrel
matt@sarrelgroup.com

IRI

IRI (Innovative Routines International)

2194 Highway A1A,
3rd Floor
Melbourne, FL 32937
USA

+1.321.777.8889

info@iri.com

www.iri.com

IRI is a leader in the data management software industry. Founded in 1978, the company grew from a data movement and manipulation utility called CoSort into a multifaceted enterprise platform player.

Represented in more than 40 cities worldwide, IRI software serves BI/DW and data architects, data governance and IT managers, and DBAs and application developers in every industry who must handle big and/or sensitive data collections in faster, more affordable ways.

Powered by the data definition and manipulation program in its CoSort data transformation and reporting utility, IRI products include: FieldShield for data masking, NextForm for data migration, RowGen for test data generation, and Voracity, a total data management platform built on Eclipse that can run many jobs seamlessly in Hadoop.

David Friedland is the COO and Sr. VP at IRI, The CoSort Company. He started with IRI in 1988 to direct technology strategy and partner development. He now also holds daily management responsibility for the company, while continuing to oversee customer and channel growth, product line enhancement, marketing collateral, licensing agreements, and new projects. He is also focused on increasing global awareness and adoption of Voracity.

David Friedland
davidf@iri.com