G00276720

Adopt Logical Data Warehouse Architectural Patterns to Mature Your Data Warehouse

Published: 1 October 2015

Analyst(s): Mei Yang Selvage

Every data warehouse, whether new or old, should be built on or evolving toward an LDW to support business intelligence and analytics. This research provides proven LDW architectural patterns and targeted action plans to help you reap a high return on your LDW investment.

Key Findings

- The core of the logical data warehouse (LDW) is shared data access. It encourages reuse, improves governance and enhances the monitoring of diverse data sources. In addition, it provides trusted and reusable data services to a wide range of data consumers, as well as performing data preparation for self-service analytics.
- The maturity of business intelligence and analytics doesn't always correspond to the maturity of the data platform. Moreover, immature data platforms eventually become a bottleneck obstructing analytical advancement.
- Unclear definitions of LDW architectural components, such as the "data lake" have caused many issues in communication, design and implementation.

Recommendations

Technical professionals should:

- Start implementing shared data access right away. You can develop shared data access via custom coding or commercial tools, such as data virtualization and data preparation tools.
- Assess your data warehousing maturity level and take appropriate action to advance its maturity. Your organization may have varying maturity levels, depending on data domains.
- Adopt Gartner's LDW architectural patterns and component definitions as architectural standards to build and evolve your data warehousing initiative.

Table of Contents

List of Tables

List of Figures

Analysis

In the world of digital business and the Internet of Things (IoT), data warehousing is essential to gaining an edge over your competition because it delivers insights for business planning and operation. However, data warehousing initiatives by nature are complex and expensive, and pervasive misconceptions and poor practices have often increased their costs and risks. Although the advancement of technology has helped in some areas, a multitude of technology options often overwhelm and distract technical professionals from building a sound analytical data foundation. Even organizations that have had successful data warehouses in the past are finding themselves on shaky ground because the business and technology landscape is constantly changing. Consequently, traditional data warehouses, centered on repositories, are no longer sufficient to meet challenges in the age of big data.

To empower digital business, it is necessary to evolve the traditional data warehouse into the next-generation data warehouse: the LDW. It speeds up delivery, handles big data's challenges and improves value. It is not an exaggeration to say that every data warehouse — new or old — should be built based on an LDW or evolving toward it.

Unfortunately, misconceptions about LDW architectural components, such as the "data lake," have caused many issues in communication, design and implementation. To help you properly start or evolve your LDW initiative, this document provides proven architectural patterns and defines characteristics of components of three LDW styles: repository, virtualization and distributed process. Consider adopting them for your architectural practice.

Most importantly, an honest evaluation of your data warehousing maturity level is necessary before taking any action. The Guidance section describes scenarios of various maturity levels: beginner, intermediate and advanced, and offers targeted advice.

The three styles of the LDW differ in how they integrate, process and manage data (see "Embrace Sound Design Principles to Architect a Successful Logical Data Warehouse"):

- **Repository:** This style consolidates structured data into central repositories. In this style, analytical tools access data from centralized repositories via predefined schemas known as "schema on write," which means that you need to define schemas upfront when persisting data in the repositories. This style satisfies higher performance and data quality requirements, but it takes longer to change schemas. Key components include sandbox, staging, operational data store (ODS), enterprise data warehouse (EDW) and data mart.
- **Virtualization:** This style, also known as data federation, retrieves and processes data on demand. When designed properly, data virtualization (DV) can speed up data integration, lower data latency, offer flexibility and reuse, and reduce data sprawl. However, it is not suitable for data of poor quality or massive data volume. Key components include designer, optimizer, caching and wrappers.
- **Distributed process (DP):** This style offers a relatively cost-effective way of processing massive data volume in a variety of data types. In addition, analytical tools often define their own

schemas during analysis time, known as "schema on read." Key components of DP include streaming, data lake, enterprise DP data store and specialized DP data store.

The Importance of Shared Data Access

Shared data access is essential for all three LDW styles because it helps to meet the uncertainty brought by the IoT, the proliferation of analytical data stores and the Nexus of Forces (that is, converging forces of big data, cloud, mobile and social computing). Shared data access provides trusted and reusable data services to a wide range of data consumers, as well as helping the monitoring of data consumption patterns. In addition, it decouples data sources and consumers so that you can manage changes more easily. Gartner has seen that many clients have improved their data warehouse value as soon as their mindset moved from getting data into repositories to delivering information using shared data access.

Furthermore, shared data access provides data services to various applications easily and quickly through Java Database Connectivity (JDBC)/Open Database Connectivity (ODBC), Java Message Service (JMS), OData, Web services and REST services. In the long term, you can even create a data marketplace to allow external development of additional data services and products, similar to FICO Analytic Cloud.

Although you can develop shared data access via custom coding, commercial tools such as DV and data preparation tools have certain advantages due to their out-of-the-box connectors, performance-enhancing techniques, data catalog capabilities and enhanced manageability. Sample DV vendors include Cisco (Composite Software) and Denodo. Sample data preparation tools include Paxata, Datameer and Tamr. Keep in mind that embedded DV functions in analytic tools are unsuitable as a strategic solution for shared data access because they are coupled with a specific analytical tool, and other tools cannot easily consume integration logic and assets.

In addition, shared data access should be placed in front of the delivery/presentation component to reduce coupling of data sources and consumers, following a well-known architectural principle: "separation of concerns" (see "Embrace Sound Design Principles to Architect a Successful Logical Data Warehouse"). It is also beneficial to place shared data access in front of batch data movement tools like a façade to manage their footprint. For example, you can develop the integration of multiple data sources using DV tools, and then extract data from it and populate data into downstream repositories such as EDW and DP data stores. This is much more efficient than creating and managing duplicated data access and extraction processes (see "Adopt Data Federation/Virtualization to Support Business Analytics and Data Services").

Finally, you can further increase the value of shared data access by defining business glossaries, lineage, and traceability (see "Initiate and Sustain a Business Glossary to Improve the Value of Business Analytics and the Logical Data Warehouse"). Certifying the data access logic helps to create a healthy ecosystem to enable self-service integration and analytics.

Common Components

All LDW styles include three common components: data integration, delivery and presentation, and information governance (including data quality, metadata, security and monitoring).

Data Integration

There are four groups of data integration patterns: batch data movement, DV, messaging and data replication. Each group corresponds to distinct groups of commercial data integration products. "Use Data Integration Patterns to Build Optimal Architecture" presents a repeatable process to help you choose an appropriate data integration pattern for a given use case. In addition, integration platform as a service (iPaaS) has increasingly evolved into an alternative for on-premises application and data integration platforms. In terms of data integration patterns, iPaaS primarily leverages messaging and batch data movement. It offers many benefits, such as speed, elasticity and initial cost reduction (see "Assessing Data Integration in the Cloud"). Finally, Hadoop has its purpose-built integration tools: Apache Sqoop and Flume as a batch style, and Kafka as a messaging style (see Note 1).

Delivery and Presentation

The delivery/presentation component consumes integrated data from the LDW and provides business-centric views to business people. It provides the whole analytical capabilities continuum: descriptive, diagnostic, predictive and prescriptive (see "The Business Context and Technology Enablers of Business Analytics"). Most organizations have focused on descriptive capabilities, such as reports and dashboards, and diagnostic capabilities, such as data discovery. Both typically fall under shared IT services. On the other hand, predictive and prescriptive capabilities, if done at all, are performed within individual business units and are not leveraged across the organization. You can begin by leveraging analytic work in business units and extending it when applicable.

Besides traditional desktop, on-premises software, mobile and cloud applications are becoming important data consumers and providers. Begin an investigation into how to leverage mobile and cloud applications bidirectionally. This provides not only vehicles for delivering analytic content but also a platform for capturing data for business intelligence and analytics (referred to as "business analytics" for the rest of this document).

Finally, it is important to integrate insights generated by analytical clients into the data platform in a governed fashion. Without such oversight, business users tend to pass off datasets through emails or network drives. Also, many insights are only used in a siloed fashion.

Information Governance

The information governance component in the LDW includes data quality, metadata, security and monitoring. These touch on technologies, people and processes.

The usefulness of an LDW largely depends on how its data quality — aka information quality — addresses data's fitness for use. For example, financial reporting would have different data quality requirements than self-service or big data analytics. In addition, good data does not happen naturally. Hence, it is critical to establish a data quality program to transform raw operational data into high-quality data (see "Establish a Data Quality Program to Support Digital Business").

Considered as a special type of metadata, the business glossary is the semantic foundation for the LDW and business analytics. While the role of IT is to facilitate and enable, the business glossary is essentially a business discipline. In other words, the content needs to be built, used and maintained by business users. To learn how to generate interest from business people and sustain effort, see "Initiate and Sustain a Business Glossary to Improve the Value of Business Analytics and the Logical Data Warehouse."

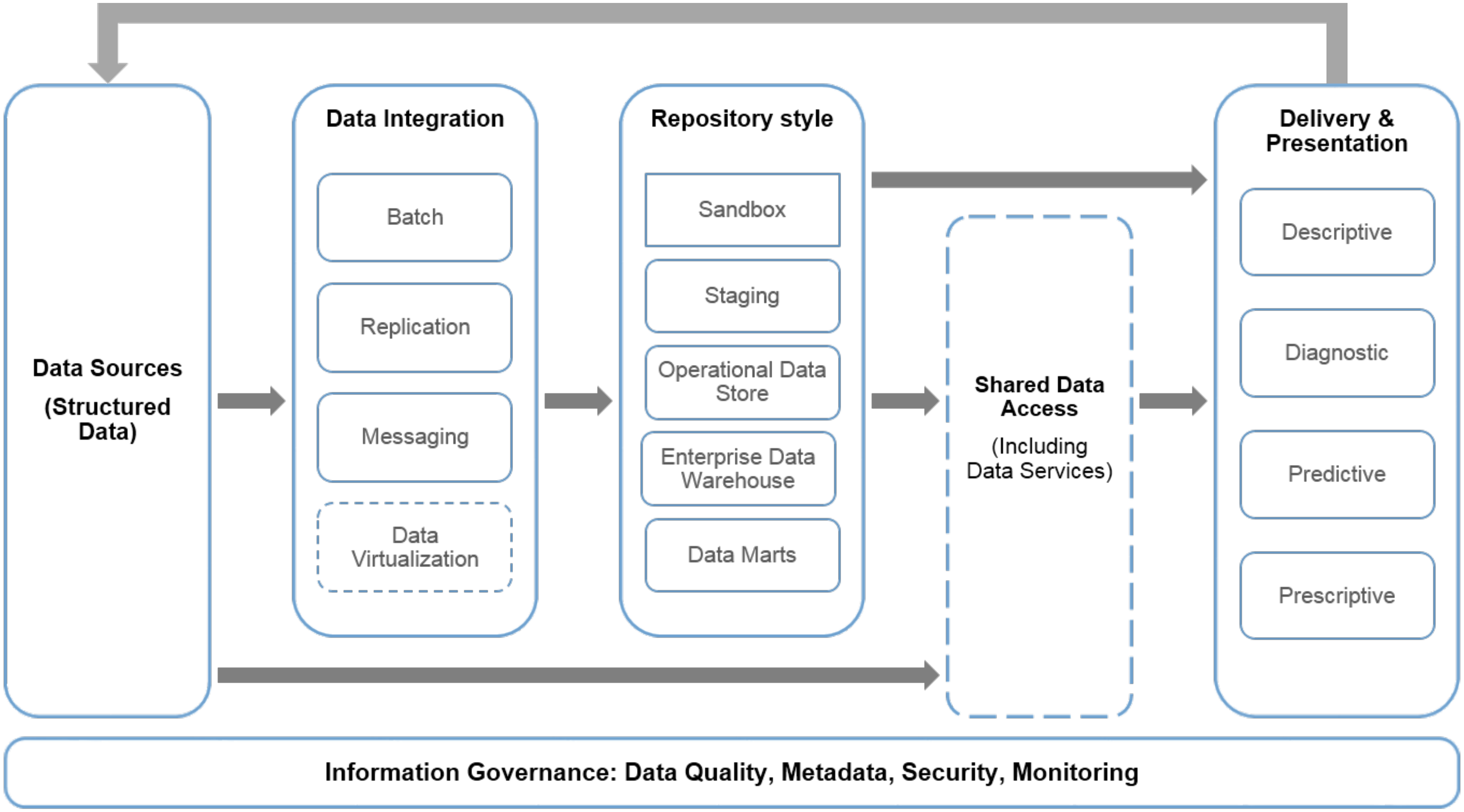
In addition to the business glossary, technical metadata management is also important. Both security and monitoring depend on the availability and integration of various metadata. Although vendors still lag on integrating and managing metadata, you still should define sensible and consistent policies to govern various LDW components.

The LDW: Repository Style

The repository style of the LDW consolidates structured data into central repositories, such as EDWs, ODSs, or data marts. Repositories have predefined schemas known as "schema on write." A repository style satisfies higher performance, concurrency and data quality requirements. It also stores historical data, which is necessary due to reasons such as regulatory compliance or trend analysis. However, it often has higher data latency due to batch data movement and takes more time to develop than the virtualization style. Once developed, schemas are also harder to change compared with the DP style because more effort is required for data modeling and ensuring data quality.

Various data stores in the repository style of the LDW reduce impact to original systems, integrate data from diverse sources and optimize performance for specific workloads. Ideally, data flows in an orderly way through the architectural components: from data sources to sandbox/staging, and then to the ODS, and finally to the EDW and data marts. However, not all organizations have adopted every component. When you need to retrofit a missing component or want to separate out the functions of an existing component, it is critical to carefully examine the actual functions of existing components and determine if you need to change the data flows, and how. For example, suppose that now your data flows from staging directly to an EDW without an ODS, and you have determined that an ODS is necessary because the business wants to access fresh, detailed data. After the ODS is set up, you may eventually route the data from an ODS to an EDW to perform data cleansing and transformation once, instead of performing them twice, from staging to an ODS, and then again from staging to an EDW.

Figure 1. The Architectural Pattern of the Repository Style



Source: Gartner (October 2015)

The following subsections describe the characteristics and functions of the architectural components in the repository style.

Data Sources

In the repository style of the LDW, data comes from a variety of internal and external data sources. Data sources are mostly structured data. For unstructured data sources, metadata such as a taxonomy can also be a data source because metadata is structured data. In addition, cloud data sources have increasingly become important in today's environment.

Sandbox

A sandbox in the repository style typically uses relational databases. It allows business users to experiment with datasets to research new ideas and identify potential for additional data value. It provides a relatively low-cost and quick method for data analysis, which complements the more governed data stores such as the ODS and EDW. Data in the sandbox is fresh, raw, structured data from transactional systems. Little data transformation or cleansing is performed here. Business users perform data analysis and exploration on top of the sandbox directly. Suitable use cases focus on self-service, such as ad hoc analysis, proof of concept, evaluation or information innovation. IT can also use it for development and testing.

One word of caution: Without clear data life cycle policies, a sandbox can quickly become a "data dump." To avoid such issues, define life cycle policies and adopt the approach used by Teradata sandboxes and eBay sandboxes. It is much easier to access and consume data in traditional database management systems — including both row- and column-oriented — than from the data lake using big data technologies.

Staging

Data in the staging area is mostly as-is data and reflects transactional system schemas. The purpose of staging is to feed integrated, harmonized data into the downstream stores: ODSs, EDWs and data marts. The use of staging largely depends on whether you adopt extraction, transformation and loading (ETL) or extract-load-transform (ELT, see "Use Data Integration Patterns to Build Optimal Architecture"). ETL is usually implemented in a stand-alone environment to provide extra processing power and to reduce the impact during the transformation and cleansing stages. In comparison, ELT extracts and loads data into staging areas in target systems first, and then transforms and cleanses data using the computing power of the target systems.

The ODS

The ODS is a subject-oriented, volatile, integrated and current-valued collection of data in support of tactical or operational decisions. The ODS reduces the impact of reporting on transactional systems. It is often modeled in the third normal form and contains one or more subjects, such as customer, product and sales information. The ODS has detailed current data but minimal historical data. If data flows from an ODS to an EDW, the cleansed data not only reduces cleansing work performed by the EDW, but also provides high-quality data to various clients, such as operational reports, ad hoc queries and drill-down.

The EDW

The EDW is a subject-oriented, nonvolatile, integrated and time-variant collection of data in support of management or strategic decisions. An EDW contains one or more subjects, such as customer, product and sales information. The data updates to an EDW are less frequent and less detailed than those to an ODS and staging. Data is integrated, transformed and cleansed. An EDW serves multiple business units. Finally, modern EDW software is capable of processing semistructured and unstructured data such as JavaScript Object Notation (JSON) or geospatial data.

The Data Mart

Typically modeled in a star schema, data marts improve performance and provide easier data access to business users. Compared with EDWs, data marts hold only a subset of data oriented toward specific subject areas, business units or countries. They may be implemented as dependent marts (where the mart is derived from an EDW), or independent marts (where the solution is populated as a stand-alone product with no dependence on an EDW).

One word of caution: Without thoughtful information governance, data marts can quickly get out of hand — known as "marting to death" — which creates redundant effort and inconsistent results. Consequently, lifetime cost is high and returns on investment are low. To avoid the explosion of data marts, you can use the conceptual data model to identify overlapping data and subject areas. Whenever possible, encourage users to leverage shared data access, the ODS and the EDW.

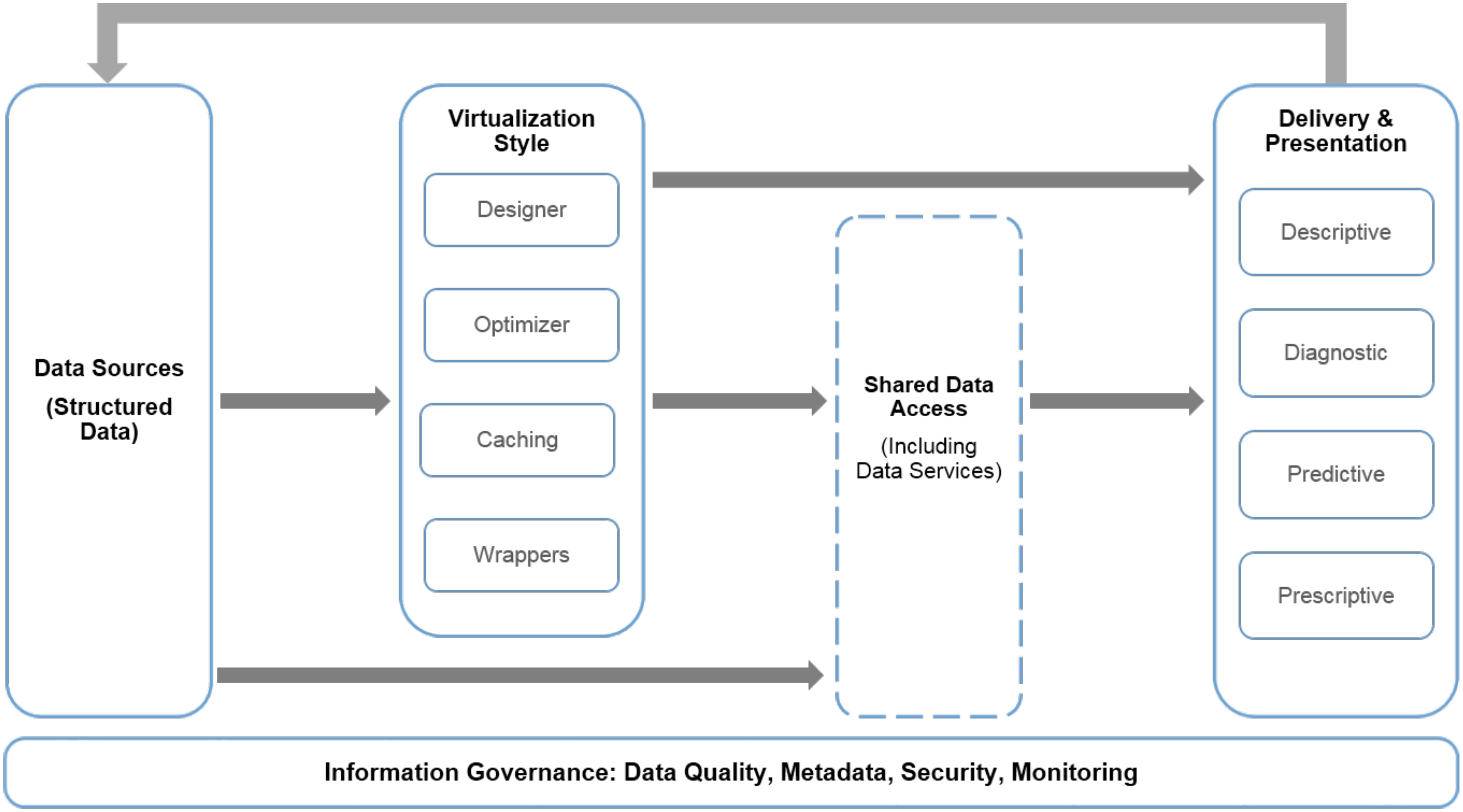
The LDW: Data Virtualization Style

DV accelerates LDW development and delivers data services for operational applications. DV creates integrated virtual views and retrieves data from multiple data sources on the fly. Sample DV vendors include Cisco (Composite Software), Denodo, IBM, Informatica, Information Builders, Oracle and Red Hat. (See Note 2 on mixed terminologies and different approaches to design and marketing in the DV space.)

When designed properly, DV can speed data integration, lower data latency, offer flexibility and reuse, and reduce data sprawl across dispersed data sources. Due to its many benefits, DV is often the first step for organizations evolving a traditional, repository-style data warehouse into an LDW (see "Embrace Sound Design Principles to Architect a Successful Logical Data Warehouse"). On the down side, DV is not suitable for data with poor quality or massive data volume. It also adds some overhead to source systems (see "Adopt Data Federation/Virtualization to Support Business Analytics and Data Services" and "Four Critical Steps for Successful Data Virtualization").

In Figure 2 below, the architectural pattern of the virtualization style merges the data integration and LDW components into one layer. This is because DV doesn't have the physical data stores to persist data permanently. Essentially, its data integration and processing functions are merged.

Figure 2. The Architectural Pattern of the Virtualization Style



Source: Gartner (October 2015)

Moreover, DV can provide governed, shared data access to multiple data sources for a variety of applications. However, not all DV is intended for shared data access. In other words, you can use DV to wrap data sources and expose siloed, as-is data for tactical purposes, such as data migration or the front interfaces for ETL jobs.

The following subsections describe the characteristics and functions of the architectural components in the virtualization style.

Data Sources

Data in the virtualization style of the LDW comes from a variety of internal and external data sources. Data sources are mostly structured data. For unstructured data sources, metadata, such as taxonomy, can also be a data source because metadata is structured data. Furthermore, DV provides a façade to retrieve data in the cloud as cloud data sources have become increasingly important.

Designer

The designer component is a graphic design tool to discover data sources and design virtual views. It is geared toward data-centric developers, such as data modelers or data integrators. Virtual views can be modeled in normalized or denormalized ways. After virtual views are created, you can test and publish results to diverse clients through JDBC/ODBC, JMS, OData, Web services and REST services.

Optimizer

The optimizer component is the intelligence of DV software. There are several ways to perform optimization. A cost-based optimizer is similar to a database query optimizer, which uses statistics to create an optimal execution path. A rule-based optimizer allows users to specify rules to run certain queries. Finally, optimizers can also be performed based on network topology.

Caching

Besides optimizers, caching is another performance enhancer in DV. Caching can use memory or materialized tables, which store data on disks, and DV software manages the data life cycle automatically. In addition, full and partial refreshing of caching can be triggered by schedules or events.

Wrappers

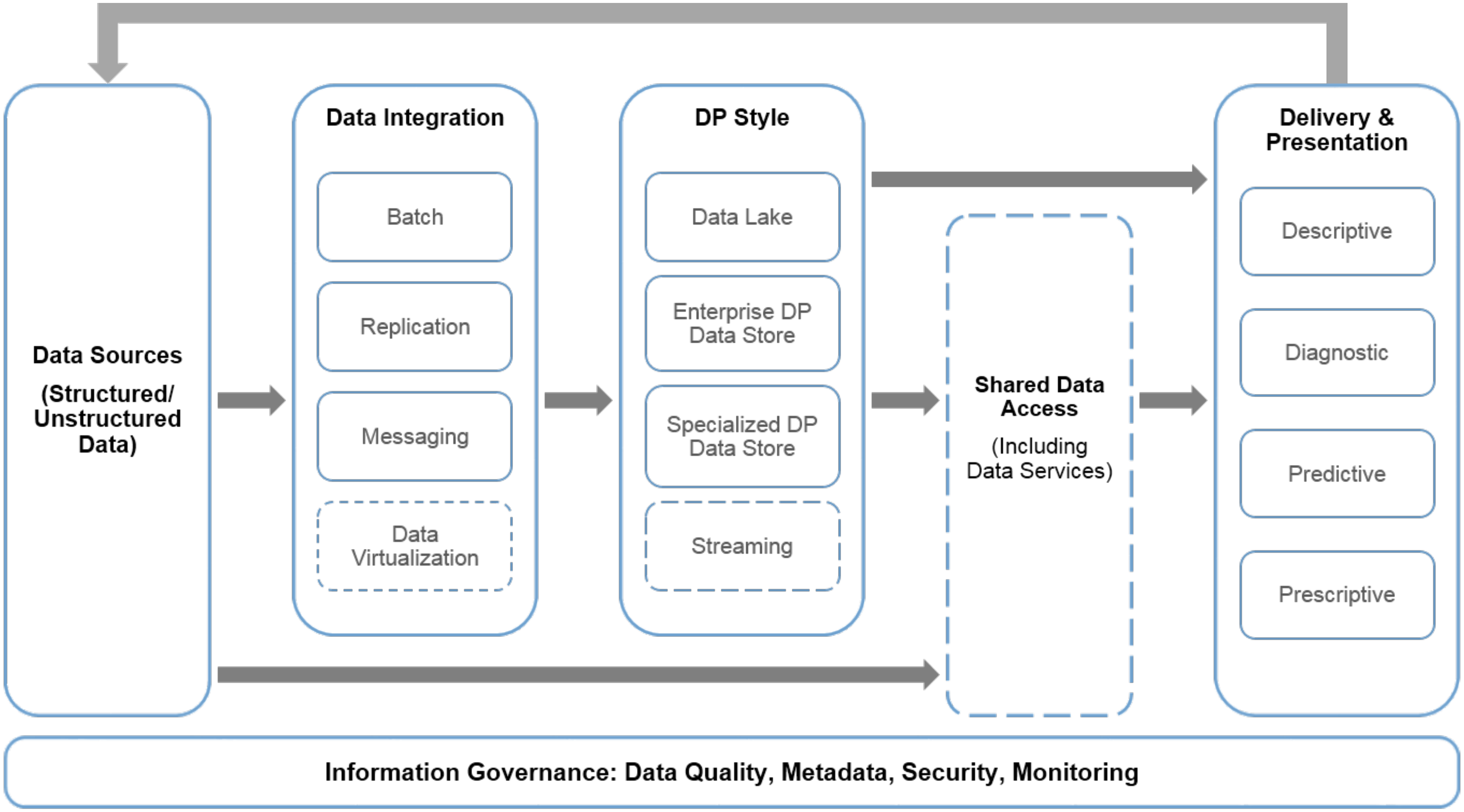
Wrappers are essentially connectors and adopters to original data sources such as relational databases, message queues, appliances, cloud data sources, packaged applications, Hadoop and other NoSQL databases. Wrappers can integrate multiple data sources and present data to consumers as if it came from a single data source.

The LDW: Distributed Process Style

The DP style of the LDW offers a relatively cost-effective way to provide massively parallel processing of diverse data types. Data is distributed among many nodes. In addition, analytical developers and end users define their own schemas at a later time, known as "eventual schema" and "schema on read." Technology examples include Hadoop, NoSQL DBMS and content management systems. Although these technologies do not share a common technical architecture, they do provide a common set of SLAs for a big data environment, such as data volume, schema variety and flexibility of data analysis.

Naturally, data redundancy occurs between repository and DP styles because both require copying and storing data. Such redundancy is inevitable, but redundancy within each style should be properly managed.

Figure 3. The Architectural Pattern of DP Styles



Source: Gartner (October 2015)

The following subsections describe the characteristics and functions of the architectural components in the DP style.

Data Sources

Data in the DP style of the LDW comes from a variety of internal and external data sources. Data can be at rest or in motion, ranging from structured to unstructured. Examples of structured data are tabular data stored in relational databases. Examples of semistructured and unstructured data are graph, sensor data (see "Technical Guidance on Sensor Data Management"), text, geospatial and social media. Like the other two LDW styles, cloud data has become increasingly important in today's DP environment.

The Data Lake

At its core, the data lake is a storage concept that is designed to handle massive and highly variable data (aka big data). Gartner defines the data lake concept as a collection of storage instances of various data assets additional to the originating data sources (see "Defining the Data Lake"). These assets are stored in a near-exact, or even exact, copy of the source format. The purpose of a data lake is to present an unrefined view of data to only the most highly skilled analysts, to help them explore their data refinement and analysis techniques independent of any of the system-of-record compromises that may exist in a traditional analytic data store (such as a data mart or data warehouse).

Despite the hype generated by Hadoop vendors, the data lake is like a sandbox for a big data environment. Consequently, it tends to have less restrictive policies in security, data quality, data life cycle and data ownership. The data lake is commonly built on top of distributed data stores such as Hadoop, AWS S3 and MongoDB. Data is minimally modeled and transformed per the "schema on read" principle. The primary users are advanced analytic users such as statisticians, data miners and data scientists. The data lake complements the EDW and data marts.

Enterprise DP Store

Like the EDW, the enterprise DP store serves a variety of analytical use cases for multiple business units. Although Hadoop is a popular choice, the enterprise DP store can also be implemented using other types of NoSQL databases, such as MarkLogic, MongoDB and AWS S3.

Popular IT-centric use cases for enterprise DP stores include data archiving and offloading ETL jobs. Sample business-centric use cases include customer view, operations, product marketing and management, compliance and risk management, and new business models (see "Decision Point for Practical Big Data Use Cases").

At its core, the enterprise DP store should have the following abilities:

- To store and process massive amounts of structured and unstructured data in parallel
- To combine data of various types, ranging from relational data to sensor data and text
- To affordably perform comprehensive analysis on a large scale

- To delay schema definition during analysis time, aka "eventual schema" or "schema on read"

The Specialized DP Store

Like data marts, specialized DP stores satisfy specific analytical use cases and business units. They are often bundled by business analytic SaaS (baSaaS) providers, which have deep knowledge of specific analytical or industry domains such as compliance, fraud detection, online sales or supply chain management. Data is typically cloud- or Web-centric, such as social media or clickstreams. Specialized DP stores complement the enterprise DP store as well. They can provide business value faster because you don't have to set up information infrastructure from the ground up. They also help to fill skill gaps, including both business and technology skills.

Because business units are primary adopters to baSaaS providers, you may wonder about IT's roles here. Although baSaaS providers handle the entire life cycle of their specialized DP stores, IT still plays important roles, such as:

- Integrating data between on-premises and baSaaS environments
- Managing metadata, including business glossary, across the systems
- Supporting the appropriate level of governance, such as data ownership and life cycles
- Validating baSaaS-promised technical capabilities

Streaming

Streaming components process data in motion, such as stock feeds or weather data. Streaming uses in-memory technologies to handle extremely high data volume with low response time and data latency. It is memory- and CPU-intensive. Moreover, choices about streaming hardware architecture (including storage, memory and CPUs) are typically influenced and determined by streaming technology vendors. Common technologies are complex-event processing (CEP), distributed-stream computing platforms (DSCPs) and in-memory analytics or in-memory DBMSs.

Architectural Considerations

Deployment

Architectural patterns described in this research are on the logical level, which means that you can deploy components on-premises, or as cloud as IaaS or PaaS. For the on-premises environment, you can deploy various components on the same or separate physical servers. Confusion between logical and physical layers often creates unnecessary arguments within an organization. For example, some technical professionals map the architectural patterns to the physical deployment on a one-to-one basis. They believe that the EDW should be a megarepository residing on a single server. This utopian approach forces everyone to agree on data models, integration styles and SLAs, which results in unhappy stakeholders who are forced to compromise on the lowest common denominators.

The better approach is to carefully evaluate the requirements of use cases — for example, user types, latency, performance and data quality — and then choose the appropriate data modeling approach, software, hardware, appliances and cloud hosting. (See "Decision Point for Logical Data Warehouse Implementation Styles" for a complete list of evaluation criteria.) Finally, continued advancement of technologies makes the physical deployment even more of a moving target. For instance, various flavors of cloud offerings are increasingly popular due to their fast-paced technology breakthroughs and ability to jump-start initiatives quickly for low cost.

Finally, Gartner has seen the following key components receiving increased cloud adoption:

- **Database platform as a service (dbPaaS):** dbPaaS is a database management system or data store engineered as a scalable, elastic, multitenant service, with a degree of self-service, and sold and supported by a cloud service provider (CSP), or a third-party software vendor on CSP infrastructure (see "Market Guide for Database Platform as a Service"). dbPaaS includes relational, NoSQL and in-memory databases (see "Is the Cloud Right for Your Database?").
- **iPaaS:** iPaaS is a suite of cloud services enabling development, execution and governance of integration flows connecting any combination of on-premises and cloud-based processes, services, applications and data within individual or across multiple organizations (see "Assessing Data Integration in the Cloud").
- **Business analytics platform as a service (baPaaS):** Solutions delivered as baPaaS include infrastructure, tools, applications and best practices. They are delivered from cloud provider data centers and enable access to, and analysis of, information to improve and optimize decisions and performance (see "Assessing Business Analytics in the Cloud" and "Why Cloud Business Analytics Makes Sense and How to Go About It").
- **baSaaS:** Unlike the above options, baSaaS vendors manage and support much of the entire analytic technology stack, which includes hardware, databases, data integration, system management and security. In some instances, the vendor even manages the analytic content for baSaaS solutions, although the customer may still be able to customize it. Since business units work with baSaaS vendors directly, IT's role here is to help business to evaluate baSaaS, and then integrate data and analysis. Sample baSaaS providers include Google Analytics for Web analytics, Microsoft Azure for stream analytics, FICO Analytical Cloud and LexisNexis. (For more information, see "Who's Who in Cloud Business Analytics.")

Comparing the ODS, Sandbox and Data Lake

There are common misconceptions about the definitions and roles of the ODS, sandbox and data lake. To clear up the confusion, Table 1 below compares the three:

Table 1. Comparison of ODS, Sandbox and Data Lake

Comparison Criteria	Operational Data Store	Sandbox (using relational database)	Data Lake (using big data technologies)
Key Objective	Deliver trustworthy information	Enable information exploration to a wide range of business users	Provide big data foundation to enable advanced analytics
Data Modeling	Schema on write; data in third normal form	Schema on write; minimal data modeling	Schema on read; as-is data with minimal data modeling
Data Quality	Cleansed, transformed, integrated data	As-is data; minimal data modeling and transformation	As-is data; minimal data modeling and transformation
Primary Audience	Business intelligence (BI) specialists and users who are comfortable with SQL and BI tools such as business analysts, BI technical specialists and business specialists	BI users who are comfortable with SQL and BI tools, such as business analysts, BI technical specialists and business specialists	Advanced analytic users such as data scientists, data miners and statisticians
Focus of Analytical Spectrum	Descriptive, diagnostic	Descriptive, diagnostic	Diagnostic, predictive and prescriptive
Ecosystem	Mature ecosystem in data integration and analytical space	Mature ecosystem in data integration and analytical space	Immature ecosystem in data integration and analytical space
Data Structure	Mostly structured data	Mostly structured data	Range from structured to unstructured data

Source: Gartner (October 2015)

One common question to Gartner is whether the data lake — commonly built on top of Hadoop — can replace the sandbox built on top of the relational database. The answer depends on many factors, such as user types and their skills, analytical use cases, data characteristics, and analytical tools' usability and accessibility. In many situations, the data lake cannot replace the sandbox sitting on top of the relational database.

Another common question to Gartner is whether the data lake can replace the ODS. The answer is no. To reiterate, the ODS is a subject-oriented, volatile, integrated and current-valued collection of detailed data in support of tactical or operational decisions. It not only supports operational reports and queries, but also provides high-quality, detailed data for other analytics. The ODS and the data lake satisfy entirely different sets of requirements and workloads.

Finally, many organizations have not realized the value of the ODS, sandbox and data lake. Without them, business users need to extract data to their desktops or departmental data stores to integrate various data together. Consequently, data is replicated all over the enterprise, data access is point-to-point, information governance is down the drain, and the centralized IT team doesn't know how data is used and who has access. A better approach is to engage with business users upfront and offer a sandbox, ODS and/or the data lake, based on their needs.

Strengths

When implemented properly, the LDW delivers the following results:

- **Faster and more effective delivery:** LDW architectural patterns are one of the most important design artifacts in the data warehousing project. They help you improve communication, design and implementation. Moreover, action plans in this research offer targeted advice to help you progress your data warehousing toward the next level.
- **Added agility:** The LDW helps you build out incrementally, starting with what you have. It leverages what is in place and provides an incremental approach for moving forward based on priorities.
- **Improved decision making:** The LDW improves decision making in a fast-changing environment. It connects diverse data sources and delivers powerful insights to diverse clients in both operational and strategic contexts.
- **Improved flexibility:** The LDW improves the flexibility of the data warehouse by using the right solutions for the right problems. For example, a Hadoop solution reduces the need for an upfront, intensive data-modeling and cleansing effort, thus empowering data scientists to analyze data quickly. Likewise, DV reduces data sprawl, allowing the IT budget to be freed up for information innovation.
- **Increased responsiveness to big data and the IoT:** Newer, distributed computing such as Hadoop and streaming helps organizations meet the unprecedented information challenges brought about by big data and the IoT. Organizations can respond better and achieve a higher and more consistent quality of service despite the increased challenges.

Weaknesses

Although the LDW brings significant business and technological benefits, it also carries some risks. The good news is that many can be avoided or minimized.

- **High investment, stakes and expectations:** The LDW requires a significant investment. When investment is high, stakes and expectations are also high. Different data management and integration styles often force parallel investment. Therefore, to balance investment portfolios, organizations need to actively seek suitable technology and business strategies, such as adopting cloud solutions and outsourcing certain analytical capabilities.
- **A technology-centered approach:** Many organizations have so far taken a technology-centered approach for business analytics and their data warehouses. Oversight of data, people, and process has caused many failures of analytic/data warehouse projects. To avoid a technology-only approach, create a holistic LDW roadmap based on the four critical success factors: data, people, process and technology (see "Why Business Analytics Projects Succeed:

Voices From the Field"). Organizations should also assess the priority of information governance and balance value, reusability, compliance and risks (see "Information Governance in the Age of Big Data").

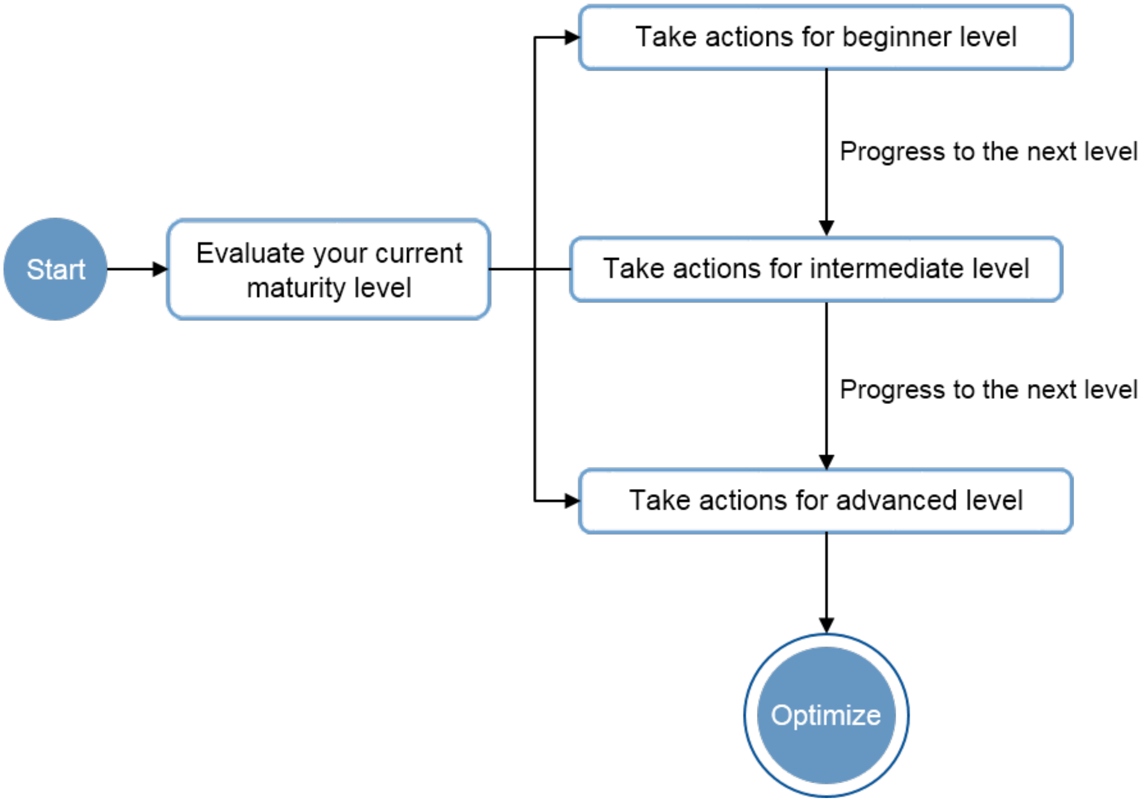
- **Lack of solid data integration and data quality:** The LDW requires extra attention for data integration and data quality. Increasing adoption of cloud software escalates issues in data silos and can perpetuate poor quality data. End users can become overwhelmed, and business decisions can become less effective when they are based on siloed and poor quality data. To avoid these risks, you need to strengthen data integration and data quality practices (see "Use Data Integration Patterns to Build Optimal Architecture" and "Establish a Data Quality Program to Support Digital Business").
- **Skills gap:** Fast-changing and complex technologies have generated a high demand for certain skills, such as Hadoop and its associated analytical skills. Many organizations find it difficult to find the right candidates to fill the positions supporting an LDW. Gartner's 2015 big data survey has revealed that skills are considered as the No. 2 challenge to handling big data.[1] Skills required in big data include technology skills and business skills.
- **Immature business:** A data-driven culture is more about a business mindset than technology adoption. Most organizations do not realize the necessity of investing in business analytics and data warehousing until they have acquired some business maturity. To persuade your business about the necessity of investment, gather competitors' stories, share Gartner research, and create a defensible business case (see "Guidance Framework for Creating a Defensible Business Case").

Guidance

Before you take any action, it is critical to honestly assess your data warehousing at the beginner, intermediate or advanced level. Your organization may have a different maturity level depending on data domains.

During your LDW journey, be realistic about the goals: Leverage what you have and deliver incrementally. You also need to periodically step back and fine-tune the direction. When you have reached the next level, adopt actions for that level. Once you have reached an advanced level, the LDW will have become a part of the digital business fabric but will still need to be continuously monitored and optimized for best business results. As you progress, avoid the "big-bang" approach and do not get distracted by the multitude of technology options. Figure 4 shows an overall approach to advance your LDW maturity level.

Figure 4. Overall Approach to Advance LDW Maturity Level


Start
Evaluate your current maturity level
Take actions for beginner level
Progress to the next level
Take actions for intermediate level
Progress to the next level
Take actions for advanced level
Optimize

Source: Gartner (October 2015)

Beginner: Build a Sound Data Platform

At the beginner level, business users access and query transactional databases directly. There are no strategic data repositories, such as an ODS or EDW. Business users download and store data in highly siloed data stores — for example, Excel, Access databases or shared network drives — which do not allow easy access for other uses. Data integration is ad hoc, and data is often exchanged via emails or Dropbox. Occasionally, the centralized IT team sets up a sandbox, but they have no idea how and what data business users are using. It is indeed a "Wild West." Regulation and business competition are the strongest motivators to push beginner organizations forward and get them to start changing.

It is not uncommon for a beginner organization to have developed siloed advanced business analytics, such as predictive analytics using SAS or SPSS, dispersed throughout the organization. However, these insights are not being leveraged by other parts of the organization. Business users may have realized that their analytical efforts are unsustainable or limited in their ability to promote advancement without a solid data platform. Users spend a fair amount of time to acquire, integrate and cleanse data on their own, instead of analyzing data and applying insights. Their data efforts often are duplicated, and the results often are contradictory.

To progress your data warehousing efforts from beginner to intermediate, take the following actions:

- Engage business early and often. This is critical when you start your LDW journey. Assure business users that you want to help them succeed, and listen to their needs and concerns.
- Establish a basic information governance body that has a sponsoring business executive and a working group including both business and IT people. Refer to "EIM 1.0: Setting Up Enterprise Information Management and Governance" for practical advice for getting started with information governance.
- Discover what business users have done so far. Although it's unlikely that you can refactor proprietary data integration and data stores created by business users, you can treat them as field-based prototypes and a starting point. Examples of valuable insights include processes, use cases, logics and requirements for data integration and quality. Discovering existing work helps you not only jump-start your LDW but also identify gaps and redundancies.
- Adopt LDW architectural patterns described in this research. Develop repository and virtualization styles iteratively in combination. Curate data in higher quality into repositories selectively. Use DV as the prototype tool, and DV also helps build the foundation for shared data access later. Avoid a common development mistake: spending years to build a "perfect," mega-EDW, but ending up with little adoption.
- Decide the suitable deployment options: on-premises and/or cloud. Data gravity (see Note 3) is the one key indicator of adopting on-premises or cloud deployment. Other factors to consider include specialized functions, data consumers, workload, network bandwidth, scalability, governance and skills (see "Assessing Data Integration in the Cloud").
- When you have narrowed down the high-priority use cases, deliver iteratively. The following questions will help you think through various stages of the development life cycle (they apply to the intermediate and advanced levels, too):
  - Analysis: What data is required? Which data sources are the best ones to use? What are the data quality requirements? Which metadata — such as business glossaries — is required?
  - Design: What is the best way to integrate and manage data? How does one make data accessible and consumable?
  - Build: Which deployment platforms are most suitable, on-premises or cloud? Which parts are best developed by shared IT vs. decentralized IT?

Intermediate: Evolve Traditional Data Warehousing Toward the LDW

Many Gartner clients belong to the intermediate LDW maturity level. They have set up certain strategic repositories such as the EDW and minimized proliferation of data marts. They have found that the traditional approach is increasingly insufficient to support their digital business. Even successful data warehousing initiatives have found their state of equilibrium disrupted by the IoT and the Nexus of Forces. In addition, these initiatives need to modernize information infrastructure or repeat success from one initial area to another while containing cost. For example, one large oil and gas company started a successful data mart in the U.S. and now needs to expand it to a global level.

For unsuccessful data warehouse initiatives, common problems are low adoption, dissatisfied end users and high costs. One of the root causes is an abuse of batch data movement, which has resulted in a long development cycle, inflexible systems, long data latency and data sprawl. Finally, for unsuccessful data warehouse initiatives, the development processes tend to be rigid, and information governance tends to be top-down, heavy and inflexible.

To progress your data warehousing from intermediate to advanced, take the following actions:

- Develop shared data access right away if you have not done so already. Use it to expose data to more data consumers, especially mobile devices. Shared data access helps you increase the value and contains the footprint of existing data repositories.
- Systematically inventory not only official IT-managed assets but also work that is developed informally by business people and decentralized IT. Discover gaps and pain points that have not been addressed by current analytical systems. Map your systems to a high-level conceptual data model. This will help you grasp and communicate the data flows and the overall data landscape, as well as the overlaps (see "Data Modeling: A Necessary and Rewarding Aspect of Data Management").
- Separate the operational and innovation budgets. IT operational budgets have remained flat for years, and most resources are tied up with operational maintenance, leaving little room for innovation. Since there is little innovation budget, IT is often forced to be in a reactive mode with technology disruptions. To break this negative cycle, consider a results-driven funding model and fund as you go. One example is Union Pacific, the largest railroad company in the U.S., which separates operational and innovation budgets. Start small, prove value and then get more funding. Most importantly, directly tie innovation investment to strategic business outcomes, such as customer experience and revenue growth. This is key to sustaining the efforts and getting more funding.
- Expand cloud options — including dbIaaS, dbPaaS, iPaaS and baPaaS — to complement your on-premises data warehousing. They will help you accelerate the delivery, reduce the initial cost, improve ease of use and improve productivity. They also set the stage for scaling in the future.
- Create a sandbox using proven database technologies, if you have not done so already. A sandbox allows average business users to easily explore and manipulate data. It provides some basic governance for a self-service analytical environment.
- Facilitate self-service analytics and baSaaS as a complement to existing business intelligence. Self-service analytics leverage the existing infrastructure to serve a broader set of business customers (see "Serving Up Self-Service Business Analytics"). Similarly, advocate for adoption of baSaaS for specialized use cases, for example, Web analytics and log analysis. baSaaS is especially beneficial when your organization lacks skills in these areas, and data gravity is cloud-centric. Help your business people evaluate baSaaS. A proactive approach is much better than being told to support baSaaS after the fact.
- Mindfully pilot and invest in a DP style for big data. Gartner predicts that through 2017 60% of big data projects will fail to go beyond piloting and experimentation and will be abandoned (see "Predicts 2015: Big Data Challenges Move From Technology to the Organization"). If you have

concrete business needs for big data technologies, start small and contain the effort. Set the proper expectations concerning the risks and benefits. It's a bad idea to channel the majority of investment into big data projects while overlooking improvement of the data management foundation.

- Evolve information governance in the age of big data. Define policies and responsibilities for each of the analytical environments. "Information Governance in the Age of Big Data" provides a six-step guidance framework and a heat map of information governance priorities based on business value, reusability, compliance and risk.
- Work with your management to formalize data steward (see "Toolkit: Data Stewardship Role Descriptions") and chief data officer (CDO) roles (see "Chief Data Officers' Handbook"). While CIO and business leaders are responsible for making a business case for a CDO, technical professionals can help them gather evidence, trends and best practices to make the business case (see "Business Case for the Chief Data Officer"). Moreover, a CDO can increase data management visibility and funding, whereas technical professionals can prepare a CDO to transform business via an LDW initiative.

Advanced: Optimize the LDW to Support Digital Business

Organizations at the advanced level have at least two LDW styles in production. However, implementation of LDW styles is siloed or has limited business value — for example, using Hadoop just for data archives and ETL extension. Alternatively, organizations may have both repository and virtualization styles now but need to expand into the big data space to support the full range of analytical capabilities, including descriptive, diagnostic, predictive and prescriptive (see "The Business Context and Technology Enablers of Business Analytics").

Compared with the beginner and the intermediate, the advanced level has a balanced technology portfolio and has kept a close watch on business needs. However, the advanced level is a dynamic state because the IoT and digital business are adding unprecedented challenges and opportunities, as well as demanding constant innovation and new investment. Today's Hadoop deployment may become obsolete in a few years due to rapidly changing technologies.

To sustain and optimize your LDW to continuously support digital business, take the following actions:

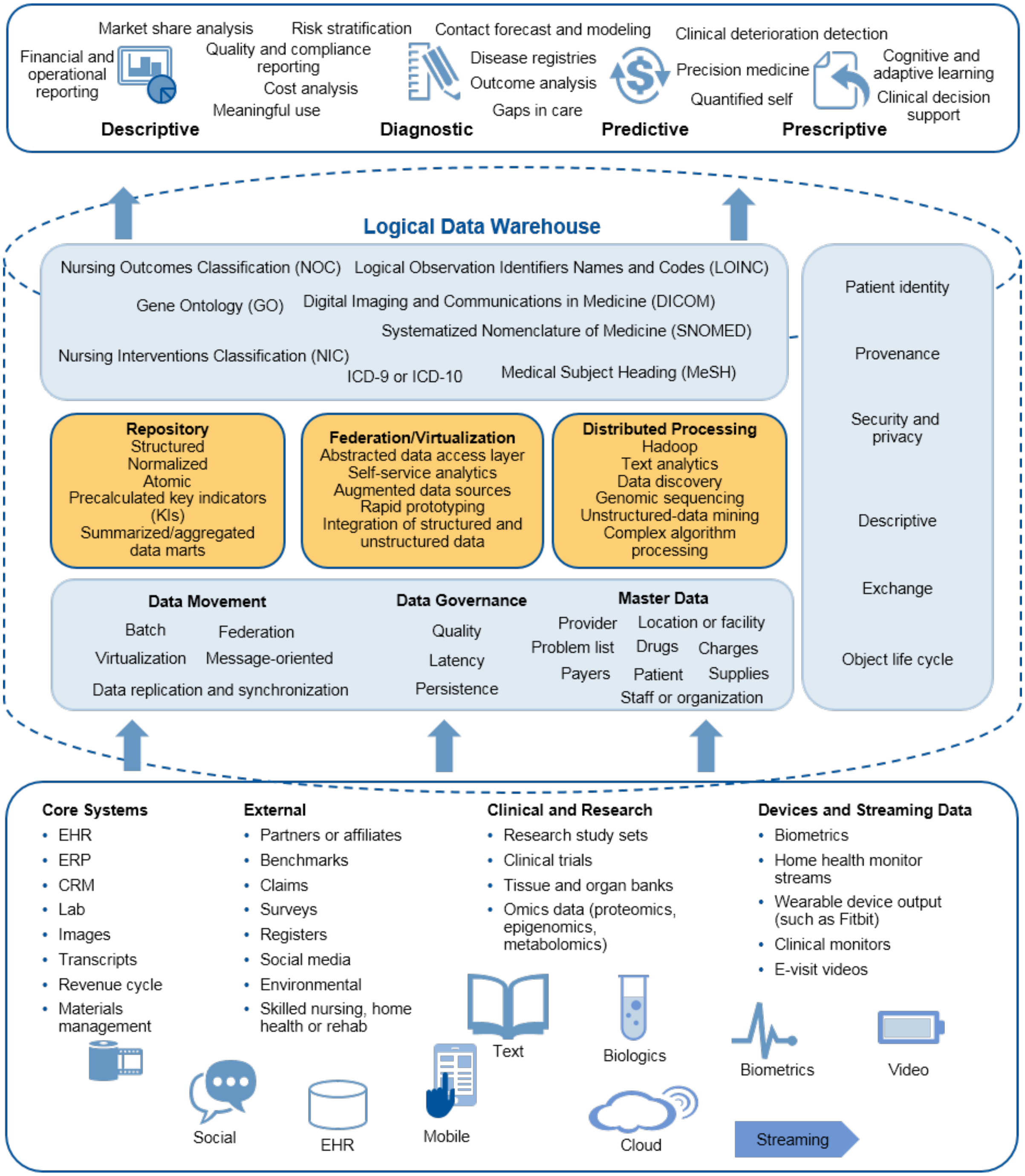
- Improve synergy between LDW styles to deliver higher business value, for example, saving big data insights back to the EDW or injecting them into business processes. In another example, use DV to extend to additional data sources and consumers. To drive higher synergy of LDW styles, you need to link different data sources holistically: internal and external, transactional systems and social media, and on-premises and cloud (see "Decision Point for Practical Big Data Use Cases").
- Optimize data integration components. Integrate data following a systematic approach to choose from four data integration styles: batch data movement, DV, messaging and data replication (see "Use Data Integration Patterns to Build Optimal Architecture").

- Start a shared business glossary to build the semantic foundation (see "Initiate and Sustain a Business Glossary to Improve the Value of Business Analytics and the Logical Data Warehouse"). Initiate a systematic data quality program (see "Establish a Data Quality Program to Support Digital Business"). Tie the business glossary and data quality directly to business objectives and metrics.
- Create a data marketplace based on shared data access. Allow consumers to configure data delivery options, subscribe to data channels, and share and contribute to data assets like commodity trading. To take it a step further, you can monetize data to partners, customers and communities (see "How Organizations Can Best Monetize Customer Data").
- Work with the business to identify potential game-changing use cases while maintaining value delivered by business-extension use cases. Business extenders have predictable costs and benefits, while game-changers have great potential for breakthroughs (see "A Framework for Evaluating Big Data Initiatives").
- Create a smart-sourced analytics and data platform. Adopt suitable cloud solutions following the data gravity principle. Outsource certain specialized analytical capabilities where your organization lacks skills and capital funding. Maintain a complete investment portfolio that supports a variety of business analytics.
- Assess the priority of information governance and balance value, reusability, compliance and risks to support information governance in the age of big data (see "Information Governance in the Age of Big Data").
- Modernize information infrastructure. This requires legacy data migration. Many organizations are ill-prepared for legacy migration. To help you achieve a greater success and reduce risks, "Legacy Data Migration Is a High-Risk Project — Be Prepared!" provides a four-phase roadmap.

The Details

If you want to see how the LDW applies to vertical industry contexts, "Adopt the Logical Data Warehouse to Meet Today's Healthcare Provider Data Challenges" illustrates LDW reference architecture in the healthcare context:

Figure 5. LDW Reference Architecture in the Healthcare Industry



EHR = electronic health record

Source: Gartner (October 2015)

Gartner Recommended Reading

Some documents may not be available as part of your current Gartner subscription.

"Embrace Sound Design Principles to Architect a Successful Logical Data Warehouse"

"Adopt Data Federation/Virtualization to Support Business Analytics and Data Services"

"Use Data Integration Patterns to Build Optimal Architecture"

"Legacy Data Migration Is a High-Risk Project — Be Prepared!"

"Solution Path for Creating a Business Analytics Strategy"

"Decision Point for Logical Data Warehouse Implementation Styles"

"Architecture Options for Big Data Analytics on Hadoop"

"How to Use Tai Chi to Find and Act on Business Intelligence and Analytics Opportunities That Are Hidden in Plain Sight"

"Establish a Data Quality Program to Support Digital Business"

Evidence

[1] Gartner conducted a big data adoption survey in 2015 (see "Survey Analysis: Practical Challenges Mount as Big Data Moves to Mainstream"). Of the 437 people who responded to the survey, 333 (76%) of them have invested in or plan to invest in big data. The top three challenges are business value, skills and risk/governance.

Note 1 Hadoop Integration Software

The Hadoop ecosystem provides some integration software, such as Apache Sqoop, Flume and Kafka. Apache Sqoop efficiently transfers bulk data between Hadoop and relational databases. Apache Flume transfers bulk data from many data sources, such as logs, documents and Twitter, to Hadoop. Apache Kafka provides publish-subscribe messaging. However, Sqoop, Flume, and Kafka lack enterprise critical capabilities, such as the ability to perform complex transformation, data quality and governance. Therefore, they are often used as an extension mechanism to integrate data in Hadoop on an ad hoc basis rather than as a replacement for traditional integration software.

Note 2 Terminologies

The DV market can be confusing because of mixed terminologies and different approaches to design and marketing. Vendors sometimes refer to DV as data federation or enterprise information integration (EII). Also, vendors have taken different approaches to design and market their products. For example, IBM InfoSphere Federation Server leverages DB2's database optimizer and focuses on providing relational database views. Other vendors, such as Oracle Data Service Integrator (ODSI) or Red Hat JBoss Data Services Platform, have taken a service-oriented architecture (SOA) path and focus on delivering data services. They frequently market their DV products as "data service" and package them as a part of their SOA platforms.

Note 3 Data Gravity

"Why Cloud Business Analytic Makes Sense and How to Go About It" defines data gravity as the term for data's "pull" on, for example, services and applications. Data's pull increases with its volume. This means that it often makes sense to perform analytics close to the source of most of the data to be analyzed. So if you own data in the cloud, it makes sense to analyze that data in the cloud.

GARTNER HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
USA
+1 203 964 0096

Regional Headquarters
AUSTRALIA
BRAZIL
JAPAN
UNITED KINGDOM

For a complete list of worldwide locations,
visit http://www.gartner.com/technology/about.jsp