O'REILLY®

Architecting Data Lakes

Data Management Architectures for Advanced Business Use Cases

Alice LaPlante
& Ben Sharma

Strata+ Hadoop WORLD

Make Data Work

strataconf.com

Presented by O'Reilly and Cloudera, Strata + Hadoop World is where cutting-edge data science and new business fundamentals intersect—and merge.

- Learn business applications of data technologies
- Develop new skills through trainings and in-depth tutorials
- Connect with an international community of thousands who work with data

Job # 15420

Architecting Data Lakes

Data Management Architectures for Advanced Business Use Cases

Alice LaPlante and Ben Sharma

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Architecting Data Lakes
by Alice LaPlante and Ben Sharma

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Shannon Cutt
Production Editor: Melanie Yarbrough
Copyeditor: Colleen Toporek
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

March 2016: First Edition

Revision History for the First Edition
2016-03-04: First Release

978-1-491-95257-3

[LSI]

Table of Contents

CHAPTER 1
Overview

Almost every large organization has an enterprise data warehouse (EDW) in which to store important business data. The EDW is designed to capture the essence of the business from other enterprise systems such as customer relationship management (CRM), inventory, and sales transactions systems, and allow analysts and business users to gain insight and make important business decisions from that data.

But new technologies—including streaming and social data from the Web or from connected devices on the Internet of things (IoT)—is driving much greater data volumes, higher expectations from users, and a rapid globalization of economies. Organizations are realizing that traditional EDW technologies can't meet their new business needs.

As a result, many organizations are turning to Apache Hadoop. Hadoop adoption is growing quickly, with 26% of enterprises surveyed by Gartner in mid-2015 already deploying, piloting, or experimenting with the next-generation data-processing framework. Another 11% plan to deploy within the year, and an additional 7% within 24 months.[1]

Organizations report success with these early endeavors in mainstream Hadoop deployments ranging from retail, healthcare, and financial services use cases. But currently Hadoop is primarily used

1 Gartner. "Gartner Survey Highlights Challenges to Hadoop Adoption." May 13, 2015.

as a tactical rather than strategic tool, supplementing as opposed to replacing the EDW. That's because organizations question whether Hadoop can meet their enterprise service-level agreements (SLAs) for availability, scalability, performance, and security.

Until now, few companies have managed to recoup their investments in big data initiatives using Hadoop. Global organizational spending on big data exceeded $31 billion in 2013, and this is predicted to reach $114 billion in 2018.[2] Yet only 13 percent of these companies have achieved full-scale production for their big-data initiatives using Hadoop.

One major challenge with traditional EDWs is their schema-on-write architecture, the foundation for the underlying extract, transform, and load (ETL) process required to get data into the EDW. With schema-on-write, enterprises must design the data model and articulate the analytic frameworks before loading any data. In other words, they need to know ahead of time how they plan to use that data. This is very limiting.

In response, organizations are taking a middle ground. They are starting to extract and place data into a Hadoop-based repository without first transforming the data the way they would for a traditional EDW. After all, one of the chief advantages of Hadoop is that organizations can dip into the database for analysis as needed. All frameworks are created in an ad hoc manner, with little or no prep work required.

Driven both by the enormous data volumes as well as cost—Hadoop can be 10 to 100 times less expensive to deploy than traditional data warehouse technologies—enterprises are starting to defer labor-intensive processes of cleaning up data and developing schema until they've identified a clear business need.

In short, they are turning to *data lakes*.

What Is a Data Lake?

A data lake is a central location in which to store all your data, regardless of its source or format. It is typically, although not always,

2 CapGemini Consulting. "Cracking the Data Conundrum: How Successful Companies Make Big Data Operational." 2014.

built using Hadoop. The data can be structured or unstructured. You can then use a variety of storage and processing tools—typically tools in the extended Hadoop family—to extract value quickly and inform key organizational decisions.

Because all data is welcome, data lakes are an emerging and powerful approach to the challenges of data integration in a traditional EDW (Enterprise Data Warehouse), especially as organizations turn to mobile and cloud-based applications and the IoT.

Some of the benefits of a data lake include:

The kinds of data from which you can derive value are unlimited.
: You can store all types of structured and unstructured data in a data lake, from CRM data, to social media posts.

You don't have to have all the answers upfront.
: Simply store raw data—you can refine it as your understanding and insight improves.

You have no limits on how you can query the data.
: You can use a variety of tools to gain insight into what the data means.

You don't create any more silos.
: You gain a democratized access with a single, unified view of data across the organization.

The differences between EDWs and data lakes are significant. An EDW is fed data from a broad variety of enterprise applications. Naturally, each application's data has its own schema. The data thus needs to be transformed to conform to the EDW's own predefined schema.

Designed to collect only data that is controlled for quality and conforming to an enterprise data model, the EDW is thus capable of answering a limited number of questions. However, it is eminently suitable for enterprise-wide use.

Data lakes, on the other hand, are fed information in its native form. Little or no processing is performed for adapting the structure to an enterprise schema. The structure of the data collected is therefore not known when it is fed into the data lake, but only found through discovery, when read.

The biggest advantage of data lakes is *flexibility*. By allowing the data to remain in its native format, a far greater—and timelier—stream of data is available for analysis.

Table 1-1 shows the major differences between EDWs and data lakes.

Table 1-1. Differences between EDWs and data lakes

Attribute	EDW	Data lake
Schema	Schema-on-write	Schema-on-read
Scale	Scales to large volumes at moderate cost	Scales to huge volumes at low cost
Access methods	Accessed through standardized SQL and BI tools	Accessed through SQL-like systems, programs created by developers, and other methods
Workload	Supports batch processing, as well as thousands of concurrent users performing interactive analytics	Supports batch processing, plus an improved capability over EDWs to support interactive queries from users
Data	Cleansed	Raw
Complexity	Complex integrations	Complex processing
Cost/ efficiency	Efficiently uses CPU/IO	Efficiently uses storage and processing capabilities at very low cost
Benefits	• Transform once, use many • Clean, safe, secure data • Provides a single enterprise-wide view of data from multiple sources • Easy to consume data • High concurrency • Consistent performance • Fast response times	• Transforms the economics of storing large amounts of data • Supports Pig and HiveQL and other high-level programming frameworks • Scales to execute on tens of thousands of servers • Allows use of any tool • Enables analysis to begin as soon as the data arrives • Allows usage of structured and unstructured content from a single store • Supports agile modeling by allowing users to change models, applications, and queries

Drawbacks of the Traditional EDW

One of the chief drawbacks of the schema-on-write of the traditional EDW is the enormous time and cost of preparing the data. For a major EDW project, extensive data modeling is typically

required. Many organizations invest in standardization committees that meet and deliberate over standards, and can take months or even years to complete the task at hand.

These committees must do a lot of upfront definitions: first, they need to delineate the problem(s) they wish to solve. Then they must decide what questions they need to ask of the data to solve those problems. From that, they design a database schema capable of supporting those questions. Because it can be very difficult to bring in new sources of data once the schema has been finalized, the committee often spends a great deal of time deciding what information is to be included, and what should be left out. It is not uncommon for committees to be gridlocked on this particular issue for weeks or months.

With this approach, business analysts and data scientists cannot ask ad hoc questions of the data—they have to form hypotheses ahead of time, and then create the data structures and analytics to test those hypotheses. Unfortunately, the only analytics results are ones that the data has been designed to return. This issue doesn't matter so much if the original hypotheses are correct—but what if they aren't? You've created a closed-loop system that merely validates your assumptions—not good practice in a business environment that constantly shifts and surprises even the most experienced businesspersons.

The data lake eliminates all of these issues. Both structured and unstructured data can be ingested easily, without any data modeling or standardization. Structured data from conventional databases is placed into the rows of the data lake table in a largely automated process. Analysts choose which tag and tag groups to assign, typically drawn from the original tabular information. The same piece of data can be given multiple tags, and tags can be changed or added at any time. Because the schema for storing does not need to be defined up front, expensive and time-consuming modeling is not needed.

Key Attributes of a Data Lake

To be classified as a true data lake, a Big Data repository has to exhibit three key characteristics:

Should be a single shared repository of data, typically stored within a Hadoop Distributed File System (HDFS)

Hadoop data lakes preserve data in its original form and capture changes to data and contextual semantics throughout the data lifecycle. This approach is especially useful for compliance and internal auditing activities, unlike with a traditional EDW, where if data has undergone transformations, aggregations, and updates, it is challenging to piece data together when needed, and organizations struggle to determine the provenance of data.

Include orchestration and job scheduling capabilities (for example, via YARN)

Workload execution is a prerequisite for Enterprise Hadoop, and YARN provides resource management and a central platform to deliver consistent operations, security, and data governance tools across Hadoop clusters, ensuring analytic workflows have access to the data and the computing power they require.

Contain a set of applications or workflows to consume, process, or act upon the data

Easy user access is one of the hallmarks of a data lake, due to the fact that organizations preserve the data in its original form. Whether structured, unstructured, or semi-structured, data is loaded and stored as is. Data owners can then easily consolidate customer, supplier, and operations data, eliminating technical—and even political—roadblocks to sharing data.

The Business Case for Data Lakes

EDWs have been many organizations' primary mechanism for performing complex analytics, reporting, and operations. But they are too rigid to work in the era of Big Data, where large data volumes and broad data variety are the norms. It is challenging to change EDW data models, and field-to-field integration mappings are rigid. EDWs are also expensive.

Perhaps more importantly, most EDWs require that business users rely on IT to do any manipulation or enrichment of data, largely because of the inflexible design, system complexity, and intolerance for human error in EDWs.

Data lakes solve all these challenges, and more. As a result, almost every industry has a potential data lake use case. For example,

organizations can use data lakes to get better visibility into data, eliminate data silos, and capture 360-degree views of customers.

With data lakes, organizations can finally unleash Big Data's potential across industries.

Freedom from the rigidity of a single data model

Because data can be unstructured as well as structured, you can store everything from blog postings to product reviews. And the data doesn't have to be consistent to be stored in a data lake. For example, you may have the same type of information in very different data formats, depending on who is providing the data. This would be problematic in an EDW; in a data lake, however, you can put all sorts of data into a single repository without worrying about schemas that define the integration points between different data sets.

Ability to handle streaming data

Today's data world is a streaming world. Streaming has evolved from rare use cases, such as sensor data from the IoT and stock market data, to very common everyday data, such as social media.

Fitting the task to the tool

When you store data in an EDW, it works well for certain kinds of analytics. But when you are using Spark, MapReduce, or other new models, preparing data for analysis in an EDW can take more time than performing the actual analytics. In a data lake, data can be processed efficiently by these new paradigm tools without excessive prep work. Integrating data involves fewer steps because data lakes don't enforce a rigid metadata schema. Schema-on-read allows users to build custom schema into their queries upon query execution.

Easier accessibility

Data lakes also solve the challenge of data integration and accessibility that plague EDWs. Using Big Data Hadoop infrastructures, you can bring together ever-larger data volumes for analytics—or simply store them for some as-yet-undetermined future use. Unlike a monolithic view of a single enterprise-wide data model, the data lake allows you to put off modeling until you actually use the data, which creates opportunities for better operational insights and data discov-

ery. This advantage only grows as data volumes, variety, and metadata richness increase.

Reduced costs

Because of economies of scale, some Hadoop users claim they pay less than $1,000 per terabyte for a Hadoop cluster. Although numbers can vary, business users understand that because it's no longer excessively costly for them to store all their data, they can maintain copies of everything by simply dumping it into Hadoop, to be discovered and analyzed later.

Scalability

Big Data is typically defined as the intersection between volume, variety, and velocity. EDWs are notorious for not being able to scale beyond a certain volume due to restrictions of the architecture. Data processing takes so long that organizations are prevented from exploiting all their data to its fullest extent. Using Hadoop, petabyte-scale data lakes are both cost-efficient and relatively simple to build and maintain at whatever scale is desired.

Data Management and Governance in the Data Lake

If you use your data for mission-critical purposes—purposes on which your business depends—you must take data management and governance seriously. Traditionally, organizations have used the EDW because of the formal processes and strict controls required by that approach. But as we've already discussed, the growing volume and variety of data are overwhelming the capabilities of the EDW. The other extreme—using Hadoop to simply do a "data dump"—is out of the question because of the importance of the data.

In early use cases for Hadoop, organizations frequently loaded data without attempting to manage it in any way. Although situations still exist in which you might want to take this approach—particularly since it is both fast and cheap—in most cases, this type of data dump isn't optimal. In cases where the data is not standardized, where errors are unacceptable, and when the accuracy of the data is of high priority, a data dump will work against your efforts to derive value from the data. This is especially the case as Hadoop transitions from an add-on-feature to a truly central aspect of your data architecture.

The data lake offers a middle ground. A Hadoop data lake is flexible, scalable, and cost-effective—but it can also possess the discipline of a traditional EDW. You must simply add data management and governance to the data lake.

Once you decide to take this approach, you have four options for action.

Address the Challenge Later

The first option is the one chosen by many organizations, who simply ignore the issue and load data freely into Hadoop. Later, when they need to discover insights from the data, they attempt to find tools that will clean the relevant data.

If you take this approach, machine-learning techniques can sometimes help discover structures in large volumes of disorganized and uncleansed Hadoop data.

But there are real risks to this approach. To begin with, even the most intelligent inference engine needs to start somewhere in the massive amounts of data that can make up a data lake. This means necessarily ignoring some data. You therefore run the risk that parts of your data lake will become stagnant and isolated, and contain data with so little context or structure that even the smartest automated tools—or human analysts—don't know where to begin. Data quality deteriorates, and you end up in a situation where you get different answers to the same question of the same Hadoop cluster.

Adapt Existing Legacy Tools

In the second approach, you attempt to leverage the applications and processes that were designed for the EDW. Software tools are available that perform the same ETL processes you used when importing clean data into your EDW, such as Informatica, IBM InfoSphere DataStage, and AB Initio, all of which require an ETL grid to perform transformation. You can use them when importing data into your data lake.

However, this method tends to be costly, and only addresses a portion of the management and governance functions you need for an enterprise-grade data lake. Another key drawback is the ETL happens outside the Hadoop cluster, slowing down operations and

adding to the cost, as data must be moved outside the cluster for each query.

Write Custom Scripts

With the third option, you build a workflow using custom scripts that connect processes, applications, quality checks, and data transformation to meet your data governance and management needs.

This is currently a popular choice for adding governance and management to a data lake. Unfortunately, it is also the least reliable. You need highly skilled analysts steeped in the Hadoop and open source community to discover and leverage open-source tools or functions designed to perform particular management or governance operations or transformations. They then need to write scripts to connect all the pieces together. If you can find the skilled personnel, this is probably the cheapest route to go.

However, this process only gets more time-consuming and costly as you grow dependent on your data lake. After all, custom scripts must be constantly updated and rebuilt. As more data sources are ingested into the data lake and more purposes found for the data, you must revise complicated code and workflows continuously. As your skilled personnel arrive and leave the company, valuable knowledge is lost over time. This option is not viable in the long term.

Deploy a Data Lake Management Platform

The fourth option involves solutions emerging that have been purpose-built to deal with the challenge of ingesting large volumes of diverse data sets into Hadoop. These solutions allow you to catalogue the data and support the ongoing process of ensuring data quality and managing workflows. You put a management and governance framework over the complete data flow, from managed ingestion to extraction. This approach is gaining ground as the optimal solution to this challenge.

How to Deploy a Data Lake Management Platform

This book focuses on the fourth option, deploying a Data Lake Management Platform. We first define data lakes and how they work.

Then we provide a data lake reference architecture designed by Zaloni to represent best practices in building a data lake. We'll also talk about the challenges that companies face building and managing data lakes.

The most important chapters of the book discuss why an integrated approach to data lake management and governance is essential, and describe the sort of solution needed to effectively manage an enterprise-grade lake. The book also delves into best practices for consuming the data in a data lake. Finally, we take a look at what's ahead for data lakes.

CHAPTER 2
How Data Lakes Work

Many IT organizations are simply overwhelmed by the sheer volume of data sets—small, medium, and large—that are stored in Hadoop, which although related, are not integrated. However, when done right, with an integrated data management framework, data lakes allow organizations to gain insights and discover relationships between data sets.

Data lakes created with an integrated data management framework eliminate the costly and cumbersome data preparation process of ETL that traditional EDW requires. Data is smoothly ingested into the data lake, where it is managed using metadata *tags* that help locate and connect the information when business users need it. This approach frees analysts for the important task of finding value in the data without involving IT in every step of the process, thus conserving IT resources. Today, all IT departments are being mandated to do more with less. In such environments, well-governed and managed data lakes help organizations more effectively leverage all their data to derive business insight and make good decisions.

Zaloni has created a data lake reference architecture that incorporates best practices for data lake building and operation under a data governance framework, as shown in Figure 2-1.

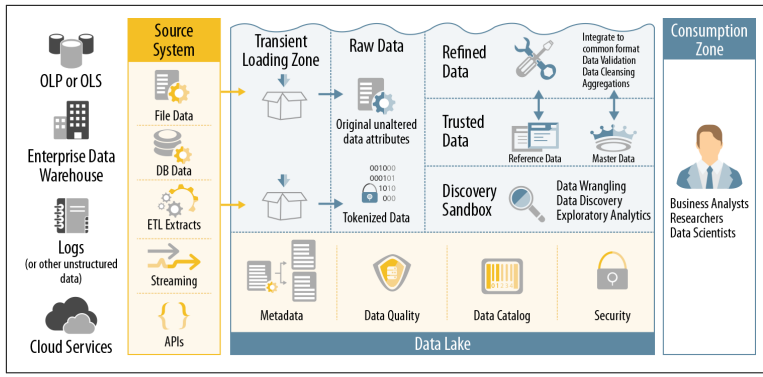

Figure 2-1. Zaloni's data lake architecture

The main advantage of this architecture is that data can come into the data lake from anywhere, including online transaction processing (OLTP) or operational data store (ODS) systems, an EDW, logs or other machine data, or from cloud services. These *source systems* include many different formats, such as file data, database data, ETL, streaming data, and even data coming in through APIs.

The data is first loaded into a *transient loading zone*, where basic data quality checks are performed using MapReduce or Spark by leveraging the Hadoop cluster. Once the quality checks have been performed, the data is loaded into Hadoop in the *raw data* zone, and sensitive data can be redacted so it can be accessed without revealing personally identifiable information (PII), personal health information (PHI), payment card industry (PCI) information, or other kinds of sensitive or vulnerable data.

Data scientists and business analysts alike dip into this raw data zone for sets of data to discover. An organization can, if desired, perform standard data cleansing and data validation methods and place the data in the *trusted* zone. This trusted repository contains both *master data* and *reference data*.

Master data is the basic data sets that have been cleansed and validated. For example, a healthcare organization may have master data sets that contain basic member information (names, addresses,) and members' additional attributes (dates of birth, social security numbers). An organization needs to ensure that this reference data kept in the trusted zone is up to date using change data capture (CDC) mechanisms.

Reference data, on the other hand, is considered the single source of truth for more complex, blended data sets. For example, that healthcare organization might have a reference data set that merges information from multiple source tables in the master data store, such as the member basic information and member additional attributes to create a single source of truth for member data. Anyone in the organization who needs member data can access this reference data and know they can depend on it.

From the trusted area, data moves into the discovery sandbox, for wrangling, discovery, and exploratory analysis by users and data scientists.

Finally, the *consumption zone* exists for business analysts, researchers, and data scientists to dip into the data lake to run reports, do "what if" analytics, and otherwise consume the data to come up with business insights for informed decision-making.

Most importantly, underlying all of this must be an integration platform that manages, monitors, and governs the metadata, the data quality, the data catalog, and security. Although companies can vary in how they structure the integration platform, in general, governance must be a part of the solution.

Four Basic Functions of a Data Lake

Figure 2-2 shows how the four basic functions of a data lake work together to move from a variety of structured and unstructured data sources to final consumption by business users: ingestion, storage/retention, processing, and access.

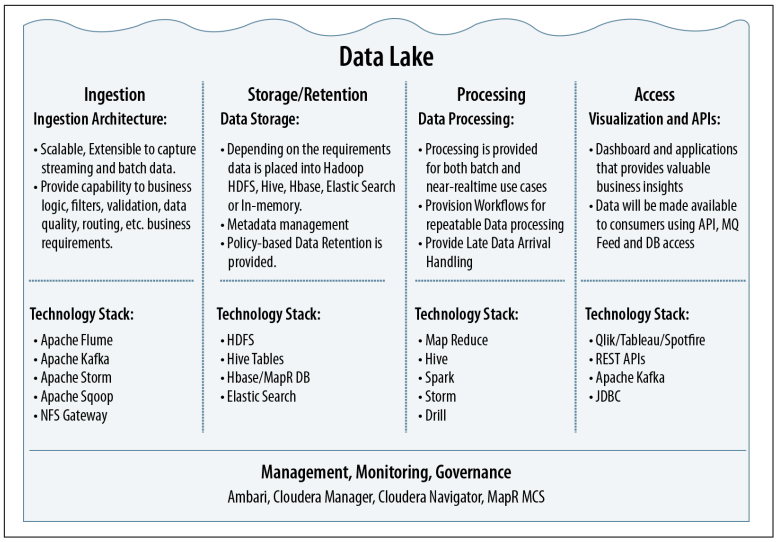

Figure 2-2. Four functions of a data lake

Data Ingestion

Organizations have a number of options when transferring data to a Hadoop data lake. *Managed ingestion* gives you control over how data is ingested, where it comes from, when it arrives, and where it is stored in the data lake.

A key benefit of managed ingestion is that it gives IT the tools to troubleshoot and diagnose ingestion issues before they become problems. For example, with Zaloni's Data Lake Management Platform, Bedrock, all steps of the data ingestion pipeline are defined in advance, tracked, and logged; the process is repeatable and scalable. Bedrock also simplifies the onboarding of new data sets and can ingest from files, databases, streaming data, REST APIs, and cloud storage services like Amazon S3.

When you are ingesting *unstructured* data, however, you realize the key benefit of a data lake for your business. Today, organizations consider unstructured data such as photographs, Twitter feeds, or blog posts to provide the biggest opportunities for deriving business value from the data being collected. But the limitations of the schema-on-write process of traditional EDWs means that only a small part of this potentially valuable data is ever analyzed.

Using managed ingestion with a data lake opens up tremendous possibilities. You can quickly and easily ingest unstructured data and make it available for analysis without needing to transform it in any way.

Another limitation of traditional EDW is that you may hesitate before attempting to add new data to your repository. Even if that data promises to be rich in business insights, the time and costs of adding it to the EDW overwhelm the potential benefits. With a data lake, there's no risk to ingesting from a new data source. All types of data can be ingested quickly, and stored in HDFS until the data is ready to be analyzed, without worrying if the data might end up being useless. Because there is such low cost and risk to adding it to the data lake, in a sense there is no useless data in a data lake.

With managed ingestion, you enter all data into a giant table organized with metadata tags. Each piece of data—whether a customer's name, a photograph, or a Facebook post—gets placed in an individual cell. It doesn't matter where in the data lake that individual cell is located, where the data came from, or its format. All of the data can be connected easily through the tags. You can add or change tags as your analytic requirements evolve—one of the key distinctions between EDW and a data lake.

Using managed ingestion, you can also protect sensitive information. As data is ingested into the data lake, and moves from the *transition* to the *raw* area, each cell is tagged according to how "visible" it is to different users in the organization. In other words, you can specify who has access to the data in each cell, and under what circumstances, right from the beginning of ingestion.

For example, a retail operation might make cells containing customers' names and contact data available to employees in sales and customer service, but it might make the cells containing more sensitive PII or financial data available only to personnel in the finance department. That way, when users run queries on the data lake, their access rights restrict the visibility of the data.

Data governance

An important part of the data lake architecture is to first put data in a *transitional* or staging area before moving it to the raw data repository. It is from this staging area that all possible data sources, external or internal, are either moved into Hadoop or discarded. As with

the visibility of the data, a managed ingestion process enforces governance rules that apply to all data that is allowed to enter the data lake.

Governance rules can include any or all of the following:

Encryption
: If data needs to be protected by encryption—if its visibility is a concern—it must be encrypted *before* it enters the data lake.

Provenance and lineage
: It is particularly important for the analytics applications that business analysts and data scientists will use down the road that the data provenance and lineage is recorded. You may even want to create rules to prevent data from entering the data lake if its provenance is unknown.

Metadata capture
: A managed ingestion process allows you to set governance rules that capture the metadata on all data before it enters the data lake's raw repository.

Data cleansing
: You can also set data cleansing standards that are applied as the data is ingested in order to ensure only clean data makes it into the data lake.

A sample technology stack for the ingestion phase of a data lake may include the following:

Apache Flume
: Apache Flume is a service for streaming logs into Hadoop. It is a distributed and reliable service for efficiently collecting, aggregating, and moving large amounts of streaming data into the HDFS. YARN coordinates the ingesting of data from Apache Flume and other services that deliver raw data into a Hadoop cluster.

Apache Kafka
: A fast, scalable, durable, and fault-tolerant publish-subscribe messaging system, Kafka is often used in place of traditional message brokers like JMS and AMQP because of its higher throughput, reliability, and replication. Kafka brokers massive message streams for low-latency analysis in Hadoop clusters.

Apache Storm
: Apache Storm is a system for processing streaming data in real time. It adds reliable real-time data processing capabilities to Hadoop. Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning, and continuous monitoring of operations.

Apache Sqoop
: Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured data stores such as relational databases. You can use Sqoop to import data from external structured data stores into a Hadoop Distributed File System, or related systems like Hive and HBase. Conversely, Sqoop can be used to extract data from Hadoop and export it to external structured data stores such as relational databases and enterprise data warehouses.

NFS Gateway
: The NFS Gateway supports NFSv3 and allows HDFS to be mounted as part of the client's local filesystem. Currently NFS Gateway supports and enables the following usage patterns:

- Browsing the HDFS filesystem through the local filesystem on NFSv3 client-compatible operating systems.
- Downloading files from the HDFS file system on to a local filesystem.
- Uploading files from a local filesystem directly to the HDFS filesystem.
- Streaming data directly to HDFS through the mount point. (File append is supported but random write is not supported.)

Zaloni Bedrock
: A fully integrated data lake management platform that manages ingestion, metadata, data quality and governance rules, and operational workflows.

Data Storage and Retention

A data lake by definition provides much more cost-effective data storage than an EDW. After all, with traditional EDWs' schema-on-write model, data storage is highly inefficient—even in the cloud.

Large amounts of data can be wasted due to the EDW's "sparse table" problem.

To understand this problem, imagine building a spreadsheet that combines two different data sources, one with 200 fields and the other with 400 fields. In order to combine them, you would need to add 400 new columns into the original 200-field spreadsheet. The rows of the original would possess no data for those 400 new columns, and rows from the second source would hold no data from the original 200 columns. The result? Empty cells.

With a data lake, wastage is minimized. Each piece of data is assigned a cell, and since the data does not need to be combined at ingest, no empty rows or columns exist. This makes it possible to store large volumes of data in less space than would be required for even relatively small conventional databases.

Additionally, when using technologies like Bedrock, organizations no longer need to duplicate data for the sake of accessing compute resources. With Bedrock and persistent metadata, you can scale-up processing without having to scale-up, or duplicate, storage.

In addition to needing less storage, when storage and compute are *separate*, customers can pay for storage at a lower rate, regardless of computing needs. Cloud service providers like AWS even offer a range of storage options at different price points, depending on your accessibility requirements.

When considering the storage function of a data lake, you can also create and enforce policy-based data retention. For example, many organizations use Hadoop as an active-archival system so that they can query old data without having to go to tape. However, space becomes an issue over time, even in Hadoop; as a result, there has to be a process in place to determine long data should be preserved in the aw repository, and how and where to archive it.

A sample technology stack for the storage function of a data lake may consist of the following:

HDFS
: A Java-based filesystem that provides scalable and reliable data storage. Designed to span large clusters of commodity servers.

Apache Hive tables
: An open-source data warehouse system for querying and analyzing large datasets stored in Hadoop files.

HBase
: An open source, non-relational, distributed database modeled after Google's BigTable that is written in Java. Developed as part of Apache Software Foundation's Apache Hadoop project, it runs on top of HDFS, providing BigTable-like capabilities for Hadoop.

MapR database
: An enterprise-grade, high performance, in-Hadoop No-SQL database management system, MapR is used to add real-time operational analytics capabilities to Hadoop. No-SQL primarily addresses two critical data architecture requirements:

  Scalability
  : To address the increasing volumes and velocity of data

  Flexibility
  : To store the variety of useful data types and formats

ElasticSearch
: An open source, RESTful search engine built on top of Apache Lucene and released under an Apache license. It is Java-based and can search and index document files in diverse formats.

Data Processing

Processing is the stage in which data can be transformed into a standardized format by business users or data scientists. It's necessary because during the process of ingesting data into a data lake, the user does not make any decisions about transforming or standardizing the data. Instead, this is delayed until the user *reads* the data. At that point, the business users have a variety of tools with which to standardize or transform the data.

One of the biggest benefits of this methodology is that different business users can perform different standardizations and transformations depending on their unique needs. Unlike in a traditional EDW, users aren't limited to just one set of data standardizations and transformations that must be applied in the conventional schema-on-write approach.

With the right tools, you can process data for both batch and near-real-time use cases. *Batch processing* is for traditional ETL workloads—for example, you may want to process billing information to generate a daily operational report. *Streaming* is for scenarios where the report needs to be delivered in real time or near real time and cannot wait for a daily update. For example, a large courier company might need streaming data to identify the current locations of all its trucks at a given moment.

Different tools are needed based on whether your use case involves batch or streaming. For batch use cases, organizations generally use Pig, Hive, Spark, and MapReduce. For streaming use cases, different tools such as Spark-Streaming, Kafka, Flume, and Storm are available.

At this stage, you can also provision workflows for repeatable data processing. For example, Bedrock offers a generic workflow that can be used to orchestrate any type of action with features like monitoring, restart, lineage, and so on.

A sample technology stack for processing may include the following:

MapReduce
: A programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster.

Apache Hive
: Provides a mechanism to project structure onto large data sets and to query the data using a SQL-like language called HiveQL.

Apache Spark
: An open-source engine developed specifically for handling large-scale data processing and analytics.

Apache Storm
: A system for processing streaming data in real time that adds reliable real-time data processing capabilities to Enterprise Hadoop. Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning, and continuous monitoring of operations.

Apache Drill
: An open-source software framework that supports data-intensive distributed applications for interactive analysis of large-scale datasets.

Data Access

This stage is where the data is consumed from the data lake. There are various modes of accessing the data: queries, tool-based extractions, or extractions that need to happen through an API. Some applications need to source the data for performing analyses or other transformations downstream.

Visualization is an important part of this stage, where the data is transformed into charts and graphics for easier comprehension and consumption. Tableau and Qlik are two tools that can be employed for effective visualization. Business users can also use dashboards, either custom-built to fit their needs, or off-the-shelf Microsoft SQL Server Reporting Services (SSRS), Oracle Business Intelligence Enterprise Edition (OBIEE), or IBM Cognos.

Application access to the data is provided through APIs, Message-Queue, and database access.

Here's an example of what your technology stack might look like at this stage:

Qlik
: Allows you to create visualizations, dashboards, and apps that answer important business questions.

Tableau
: Business intelligence software that allows users to connect to data, and create interactive and shareable dashboards for visualization.

Spotfire
: Data visualization and analytics software that helps users quickly uncover insights for better decision-making.

RESTful APIs
: An API that uses HTTP requests to GET, PUT, POST, and DELETE data.

Apache Kafka
: A fast, scalable, durable, and fault-tolerant publish-subscribe messaging system, Kafka is often used in place of traditional message brokers like JMS and AMQP because of its higher throughput, reliability, and replication. Kafka brokers massive message streams for low-latency analysis in Enterprise Apache Hadoop.

Java Database Connectivity (JDBC)
: An API for the programming language Java, which defines how a client may access a database. It is part of the Java Standard Edition platform, from Oracle Corporation.

Management and Monitoring

Data governance is becoming an increasingly important part of the Hadoop story as organizations look to make Hadoop data lakes essential parts of their enterprise data architectures.

Although some of the tools in the Hadoop stack offer data governance capabilities, organizations need more advanced data governance capabilities to ensure that business users and data scientists can track data lineage and data access, and take advantage of common metadata to fully make use of enterprise data resources.

Solutions approach the issue from different angles. A *top-down* method takes best practices from organizations' EDW experiences, and attempts to impose governance and management from the moment the data is ingested into the data lake. Other solutions take a *bottom-up* approach that allows users to explore, discover, and analyze the data much more fluidly and flexibly.

A Combined Approach

Some vendors also take a combined approach, utilizing benefits from the top-down and bottom-up processes. For example, some top-down process is essential if the data from the data lake is going to be a central part of the enterprise's overall data architecture. At the same time, much of the data lake can be managed from the bottom up—including managed data ingestion, data inventory, data enrichment, data quality, metadata management, data lineage, workflow, and self-service access.

With a top-down approach, data governance policies are defined by a centralized body within the organization, such as a chief data officer's office, and are enforced by all of the different functions as they build out the data lake. This includes data quality, data security, source systems that can provide data, the frequency of the updates, the definitions of the metadata, identifying the critical data elements, and centralized processes driven by a centralized data authority.

In a bottom-up approach, consumers of the data lake are likely data scientists or data analysts. Collective input from these consumers is used to decide which datasets are valuable and useful and have good quality data. You then surface those data sets to other consumers, so they can see the ways that their peers have been successful with the data lake.

With a combined approach, you avoid hindering agility and innovation (what happens with the top-down approach), and at the same time, you avoid the chaos of the bottom-up approach.

Metadata

A solid governance strategy requires having the right metadata in place. With accurate and descriptive metadata, you can set policies and standards for managing and using data. For example, you can create policies that enforce users' ability to acquire data from certain places; which users own and are therefore responsible for the data; which users can access the data; how the data can be used, and how it's protected—including how it is stored, archived, and backed up.

Your governance strategy must also specify how data will be audited to ensure that you are complying with government regulations. This can be tricky as diverse data sets are combined and transformed.

All this is possible if you deploy a robust data management platform that provides the technical, operational, and business metadata that third-party governance tools need to work effectively.

CHAPTER 3

Challenges and Complications

A data lake is not a panacea. It has its challenges, and organizations wishing to deploy a data lake must address those challenges head-on. As this book has discussed, data lakes are built as vehicles for storing and providing access to large volumes of disparate data. Rather than creating rigid and limited EDWs, all your data can be stored together for discovery, enabling greater leveraging of valuable data for business purposes. This solves two problems that have plagued traditional approaches to Big Data: it eliminates data silos, and it enables organizations to make use of new types of data (i.e., streaming and unstructured data), which are difficult to place in a traditional EDW.

However, challenges still exist in building, managing, and getting value out of the data lake. We'll examine these challenges in turn.

Challenges of Building a Data Lake

When building a data lake, you run into three potential roadblocks: the rate of change in the technology ecosystem, scarcity of skilled personnel, and the technical complexity of Hadoop.

Rate of Change

The Hadoop ecosystem is large, complex, and constantly changing. Keeping up with the developments in the open-source community can be a full-time job in and of itself. Each of the components is continually evolving, and new tools and solutions are constantly

emerging from the community. For an overview of the Hadoop ecosystem, check out *The Hadoop Ecosystem Table* on GitHub.

Acquiring Skilled Personnel

As a still-emerging technology, Hadoop requires skilled development and architecture professionals who are on the leading edge of information management. Unfortunately, there's a significant skill gap in the labor marketplace for these skills: talent is scarce, and it is expensive. A CIO survey found that 40 percent of CIOs said they had a skill gap in information management.[1]

Technological Complexity

Finally, you've got the complexity of deploying the technology itself. You've got to pull together an ecosystem that encompasses hardware, software, and applications. As a distributed filesystem with a large and ever-changing ecosystem, Hadoop requires you to integrate a plethora of tools to build your data lake.

Challenges of Managing the Data Lake

Once you've built the data lake, you have the challenge of managing it (see Figure 3-1). To effectively consume data in the lake, organizations need to establish policies and processes to:

- Publish and maintain a *data catalog* (containing all the metadata collected during ingestion and data-quality monitoring) to all stakeholders
- Configure and manage *access* to data in the lake
- Monitor PII and regulatory *compliance* of usage of the data
- Log access requests for data in the lake

1 CIO. 2016 CIO Agenda Survey. 2016.

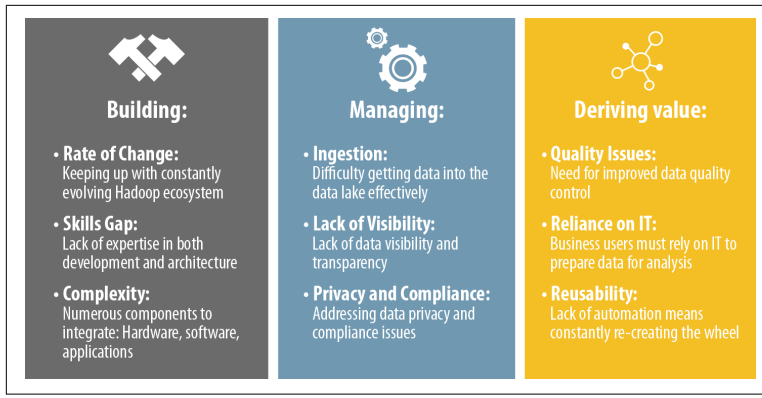

Figure 3-1. Tackling data lake complications

Ingestion

Ingestion is the process of moving data into the distributed Hadoop file system. Deploying a solution that can perform *managed* ingestion is critical, because it supports ingestion from streaming sources like log files, or physical files landed on an edge node outside Hadoop. Data quality checks are performed after data is moved to HDFS, so you can leverage the cluster resources and perform the data-quality checks in a distributed manner.

It's important to understand that all data in the lake is not equal. You need governance rules that can be flexible, based on the *type* of data that is being ingested. Some data should be certified as accurate and of high quality. Other data might require less accuracy, and therefore different governance rules.

The basic requirements when ingesting data into the data lake include the following:

- Define the incoming data from a business perspective
- Document the context, lineage, and frequency of the incoming data
- Classify the security level (public, internal, sensitive, restricted) of the incoming data
- Document the creation, usage, privacy, regulatory, and encryption business rules that apply to the incoming data
- Identify the data owner (sponsor) of the ingested data

- Identify the data steward(s) charged with monitoring the health of the specific data sets
- Continuously measure data quality as it resides in the data lake

Lack of Visibility

Without the proper tools, you lack visibility and transparency into the data lake. Ideally, a solution will organize and catalog data as it arrives, and then provide simple user interfaces to search it. A solution might also create Hive tables for the data, which can be queried using SQL.

Privacy and Compliance

As always, when you deal with enterprise data, you run into issues of privacy and compliance. How do you protect the data so that only authorized users can see it? How do you comply with the increasing number of privacy rules and regulations? An ideal solution will let you apply masking and tokenization for sensitive data like Social Security numbers, birthdates, and phone numbers.

Without a managed data lake solution, data can be placed into the data lake without any oversight, creating significant risk exposure for the organization.

Deriving Value from the Data Lake

Finally, there are the challenges of deriving *value* from the data lake. The most important challenge is occurs when you have a data lake that is *not* managed. In this case, you have no way of determining data quality or the lineage of data sets discovered by other business users. Without this metadata, users must start from scratch every time they attempt data analysis.

Furthermore, if a data lake is not managed, business users must rely on IT to prepare data for analysis. IT departments typically get overwhelmed with requests, and the queue for extracting data for analysis can be long and slow. This also adds to the overall operational costs of deploying a data lake.

Reusability

Without the right tools to automate the many aspects of ingesting, storing, and processing data, you will be constantly reinventing the wheel. With a managed data lake, once you've put a rule or policy in place on an action (for example, how to provision data for internal applications), a workflow can be scheduled every day to export the latest dataset to the correct location in HDFS so that tools can consume and report it. This entire process can be automated and the code can be reused for different data sets.

CHAPTER 4
Curating the Data Lake

To leverage a data lake as a core data platform—and not just an adjunct staging area for the EDW—enterprises need to impose proper governance. Organizations that possess many potential use cases require the mature controls and context found in traditional EDWs, before they will trust their business-critical applications to a data lake. Although it is exciting to have a cost-effective scale-out platform, without controls in place, no one will trust it. It might work, but you still need a management and governance layer that organizations are accustomed to having in traditional EDW environments.

For example, consider a bank doing risk data aggregation across different lines of business into a common risk-reporting platform for the Basel Committee on Banking Supervision (BCBS) 239. The data has to be of very high quality, and have good lineage to ensure the reports are correct, because the bank depends on those reports to make key decisions about how much capital to carry. Without this lineage, there are no guarantees that the data is accurate.

Hadoop makes perfect sense for this kind of data, as it can scale out as you bring together large volumes of different risk data sets across different lines of business. From that perspective, Hadoop works well. But what Hadoop lacks is the metadata layer, as well as quality and governance controls. To succeed at applying data lakes to these kinds of business use cases, you need rigorous controls in place.

To achieve the balance between the rigid and inflexible structure of a traditional EDW and the performance and low-cost of the so-called

"data swamp," organizations can deploy integrated management and governance platforms that allow them to manage, automate, and execute operational tasks in the data lake. This saves them both development time and money.

Data Governance

It's important to note that in addition to the tools required to maintain governance, having a *process*—frequently a manual process—is also required. Process can be as simple as assigning stewards to new data sets, or forming a data lake enterprise data council, to establish data definitions and standards.

Questions to ask when considering goals for data governance:

Quality and consistency
: Is the data of sufficient quality and consistency to be useful to business users and data scientists in making important discoveries and decisions?

Policies and standards
: What are the policies and standards for ingesting, transforming, and using data, and are they observed uniformly throughout the organization?

Security, privacy, and compliance
: Is access to sensitive data limited to those with the proper authorization?

Data lifecycle management
: How will we manage the lifecycle of the data? At what point will we move it from expensive, Tier-1 storage to less expensive storage mechanisms?

Integrating a Data Lake Management Solution

A data lake management solution needs to be integrated because the alternative is to perform the best practice functions listed above in siloes, thereby wasting a large amount of time stitching together different point products. You would end up spending a great deal of resources on the plumbing layer of the data lake—the platform—when you could be spending resources on something of real value to the business, like analyses and insights your business users gain from the data.

Having an integrated platform improves your time-to-market for insights and analytics tremendously, because all of these aspects fit together. As you ingest data, the metadata is captured. As you transform the data into a refined form, lineage is automatically captured. And as the data comes in, you have rules that inspect the data for quality—so whatever data you make available for consumption goes through these data quality checks.

Data Acquisition

Although you have many options when it comes to getting data into Hadoop, doing so in a managed way means that you have control over what data is ingested, where it comes from, when it arrives, and where in Hadoop it is stored. A well-managed data ingestion process simplifies the onboarding of new data sets and therefore the development of new use cases and applications.

As we discussed in Chapter 3, the first challenge is ingesting the data—getting the data into the data lake. An integrated data lake management platform will perform managed ingestion, which involves getting the data from the source systems into the data lake and making sure it is a process that is repeatable, and that if anything fails in the daily ingest cycle, there will be operational functions that take care of it.

For example, a platform implementing managed ingestion can raise notifications and captures logs, so that you can debug why an ingestion failed, fix it, and restart the process. This is all tied with post-processing once the data is stored in the data lake.

Additionally, as we see more and more workloads going toward a streaming scenario, whatever data management functions you applied to batch ingestion—when data was coming in periodically—now needs to be applied to data that is streaming in continuously. Integrated data lake management platforms should be able to detect if certain streams are not being ingested based on the SLAs you set.

A data lake management platform should ensure that the capabilities available in the batch ingestion layer are also available in the streaming ingestion layer. Metadata still needs to be captured and data quality checks need to be performed for streaming data. And you still need to validate that the record format is correct, and that

the record values are correct by doing range checks or reference integrity checks.

By using a data management solution purpose-built to provide these capabilities, you build the foundation for a well-defined data pipeline. Of course, you need the right processes, too, such as assigning stewards for new data sets that get ingested.

Data Organization

When you store data, depending on the use case, you may have some security encryption requirements to consider. Data may need to be either *masked* or *tokenized*, and protected with proper access controls.

A core attribute of the data lake architecture is that multiple groups share access to centrally stored data. While this is very efficient, you have to make sure that all users have appropriate permission levels to view this information. For example, in a healthcare organization, certain information is deemed private by law, such as PHI (Protected Health Information), and violators—organizations that don't protect this PHI—are severely penalized.

The data preparation stage is often where sensitive data, such as financial and health information, is protected. An integrated management platform can perform *masking* (where data from a field is completely removed) and *tokenization* (changing parts of the data to something innocuous). This type of platform ensures you have a policy-based mechanism, like access control lists, that you can enforce to make sure the data is protected appropriately.

It's also important to consider the best format for storing the data. You may need to store it in the raw format in which it came, but you may also want to store it in a format that is more consumable for business users, so that queries will run faster. For example, queries run on columnar data sets will return much faster results than those in a typical row data set. You may also want to compress the data, as it may be coming in in large volumes, to save on storage.

Also, when storing data, the platform should ideally enable you to *automate* data lifecycle management functions. For example, you may store the data in different zones in the data lake, depending on different SLAs. For example, as raw data comes in, you may want to store it in a "hot zone" where data is stored that is used very fre-

quently, for a certain amount of time, say 30 days. Then after that you may want to move it to a warm zone for 90 days, and after that, to a cold zone for seven years, from which the queries are much more infrequent.

Data Catalog

With the distributed HDFS filesystem, you ingest information that is first broken up into blocks, and then written in a distributed manner in the cluster. However, sometimes you need to see what data sets exist in the data lake, the properties of those data sets, the ingestion history of the data set, the data quality, and the key performance indicators (KPIs) of the data as it was ingested. You should also see the data profile, and all the metadata attributes, including those that are business, technical, and operational. All of these things need to be abstracted to a level to where the user can understand them, and use that data effectively—this is where the data lake *catalog* comes in.

Your management platform should make it easy to create a data catalog, and to provide that catalog to business users, so they can easily search it—whether searching for source system, schema attributes, subject area, or time range. This is essential if your business users are to get the most out of the data lake, and use it in a swift and agile way.

With a data catalog, users can find data sets that are curated, so that they don't spend time cleaning up and preparing the data. This has already been done for them, particularly in cases of data that has made it to the trusted area. Users are thus able to select the data sets they want for model building without involving IT, which shortens the analytics timeline.

Capturing Metadata

Metadata is extraordinarily important to managing your data lake. An integrated data lake management platform makes metadata creation and maintenance an integral part of the data lake processes. This is essential, as without effective metadata, data dumped into a data lake may never be seen again.

You may have a lot of requirements that are defined by your organization's central data authority, by your chief data officer or data

stewards in your lines of business, who may want to specify the various attributes and entities of data that they are bringing into the data lake.

Metadata is critical for making sure data is leveraged to its fullest. Whether manually collected or automatically created during data ingestion, metadata allows your users to locate the data they want to analyze. It also provides clues for future users to understand the contents of a data set and how it could be reused.

As data lakes grow deeper and more important to the conduct of daily business, metadata is a vital tool in ensuring that the data we pour into these lakes can be found and harnessed for years to come. There are three distinct but equally important types of metadata to collect: technical, operational, and business data, as shown in Table 4-1.

Table 4-1. Three equally important types of metadata

Type of metadata	Description	Example
Technical	Captures the form and structure of each data set	Type of data (text, JSON, Avro), structure of the data (the fields and their types)
Operational	Captures lineage, quality, profile and provenance of the data	Source and target locations of data, size, number of records, lineage
Business	Captures what it all means to the user	Business names, descriptions, tags, quality and masking rules

Technical metadata captures the form and structure of each data set. For example, it captures the type of data file (text, JSON, Avro) and the structure of the data (the fields and their types), and other technical attributes. This is either automatically associated with a file upon ingestion or discovered manually after ingestion. *Operational metadata* captures the lineage, quality, profile, and provenance of the data at both the file and the record levels, the number of records, and the lineage. Someone must manually enter and tag entities with operational metadata. *Business metadata* captures what the user needs to know about the data, such as the business names, the descriptions of the data, the tags, the quality, and the masking rules for privacy. All of this can be automatically captured by an integrated data management platform upon ingestion.

All of these types of metadata should be created and actively curated —otherwise, the data lake is simply a wasted opportunity. Addition-

ally, leading integrated data management solutions will possess file and record level watermarking features that enable you to see the data lineage, where data moves, and how it is used. These features safeguard data and reduce risk, as the data manager will always know where data has come from, where it is, and how it is being used.

Data Preparation

Making it easier for business users to access and use the data that resides in the Hadoop data lake without depending on IT assistance is critical to meeting the business goals the data lake was created to solve in the first place.

However, just adding raw data to the data lake does not make that data ready for use by data and analytics applications: *data preparation* is required. Inevitably, data will come into the data lake with a certain amount of errors, corrupted formats, or duplicates. A data management platform makes it easier to adequately prepare and clean the data using built-in functionality that delivers data security, quality, and visibility. Through *workflow orchestration*, rules are automatically applied to new data as it flows into the lake.

For example, Bedrock allows you to automatically orchestrate and manage the data preparation process from simple to complex, so that when your users are ready to analyze the data, the data is available.

Data preparation capabilities of an integrated data lake management platform should include:

- Data tagging so that searching and sorting becomes easier
- Converting data formats to make executing queries against the data faster
- Executing complex workflows to integrate updated or changed data

Whenever you do any of these data preparation functions, you need metadata that shows the lineage from a transformation perspective: what queries were run? When did they run? What files were generated? You need to create a lineage graph of all the transformations that happen to the data as it flows through the pipeline.

Additionally, when going from raw to refined, you might want to *watermark* the data by assigning a unique ID for each record of the data, so you can trace a record back to its original file. You can watermark at either the record or file level. Similarly, you may need to do format conversions as part of your data preparation, for example, if you prefer to store the data in a columnar format.

Other issues can arise. You may have changes in data coming from source systems. How do you reconcile that changed data with the original data sets you brought in? You should be able to maintain a time series of what happens over a period of time.

A data management platform can do all of this, and ensure that all necessary data preparation is completed before the data is published into the data lake for consumption.

Data Provisioning

Self-service consumption is essential for a successful data lake. Different types of users consume the data, and they are looking for different things—but each wants to access the data in a self-service manner, without the help of IT:

The Executive

An executive is usually a person in senior management looking for high-level analyses that can help her make important business decisions. For example, an executive could be looking for predictive analytics of product sales based on history and analytical models built by data scientists. In an integrated data lake management platform, data would be ingested from various sources—some streaming, some batch, and then processed in batches to come up with insights, with the final data able to be visualized using Tableau or Excel. Another common example is an executive who needs a 360-degree view of a customer, including metrics from every level of the organization—pre-sales, sales, and customer support—in a single report.

The Data Scientist

Data scientists are typically looking at the data sets and trying to build models on top of them, performing exploratory ad hoc analyses to prove or come up with a thesis about what they see. Data scientists who want to build and test their models will find a data lake

useful in that they have access to all of the data, and not just a sample. Additionally, they can build scripts in Python, and run it on a cluster to get a response in hours, rather than days.

The Business Analyst

Business analysts usually try to correlate some of the data sets, and create an aggregated view to slice and dice using a business intelligence or visualization tool.

With a traditional EDW, business analysts had to come up with reporting requirements and wait for IT to build a report, or export the data on their behalf. Now, business analysts can ask "what-if" questions from data lakes on their own. For example, a business analyst might ask how much sales were impacted due to weather patterns, based on historical data and information from public data sets, combined with in-house data sets in the data lake. Without involving IT, he could consult the catalog to see what data sets have been cleaned and standardized and run queries against that data.

A Downstream System

A fourth type of consumer is a downstream system, such as an application or a platform, which receives the raw or refined data. Leading companies are building new applications and products on top of their data lake, so they are also consumers of the data. They may also use RESTful APIs or some other API mechanisms, on an ongoing manner. For example, if the downstream application is a database, the data lake can ingest and transform the data, and send the final aggregated data to the downstream system for storage.

Benefits of an Automated Approach

Taking an integrated data management approach to a data lake ensures that each business unit does not build a separate cluster—a common practice with EDWs. Instead, you build a data lake with a *shared enterprise cluster*. An integrated management platform provides the governance and the multi-tenant capabilities to do this, and to implement best practices for governance without impacting the speed or agility of the data lake. This type of platform enables you to:

Understand provenance
: Track the source and lineage of any data loaded into the data lake. This gives you *traceability* of the data, tells you where it came from, when it came in, how many records it has, and if the data set was created from other data sets. These details allow you to establish *accountability*, and you can use this information to do impact analysis on the data.

Understand context
: Record data attributes, such as the purpose for which the data was collected, the sampling strategies employed in its collection, and any data dictionaries or field names associated with it. This pieces of information makes your organization much more productive as you progress along the analytics pipeline to derive insights from the data.

Track updates
: Log each time new data is loaded from the same source and record any changes to the original data introduced during an update. You need to do this in cases where data formats keep changing. For example, say you are working with a retail chain, with thousands of point-of-sale (POS) terminals sending data from 8,000-plus stores in the United States. These POS terminals are gradually upgraded to newer versions, but not everything can be upgraded on a given day—and now you have multiple formats of data coming in. How do you keep track as the data comes in? How do you know what version it maps to? How do you associate it with the right metadata and right structures so that it can be efficiently used for building the analytics? All of these questions can be answered with a robust integrated data lake management platform.

Track modifications recording
: Record when data is actively changed, and know by whom and how it was done. If there are format changes, you are able to track them as you go from version 1 to version 2, so you know which version of the data you are processing, and the structure or schemes associated with that version.

Perform transformations
: Convert data from one format to another to de-duplicate, correct spelling, expand abbreviations, or add labels. Driven by metadata, these transformations are greatly streamlined. And

because they are based on metadata, the transformations can accommodate changes in a much more dynamic manner. For example, you have a record format with 10 fields and perform a transformation based on metadata of 10 fields. If you decide to add an additional field, you can adjust that transformation without having to go back to the beginning implementation of the transformation. In other words, the transformation is driven by and integrated with the metadata.

Track transformations
: Performing transformations is a valuable ability, but an additional, essential requirement involves keeping track of the transformations you have accomplished. With a leading integrated data management platform, you can record the ways in which data sets are transformed. Say you perform a transformation from a source to a target format: you can track the lineage so that you know, for example, that this file and these records were transformed to a new file in this location and in this format, which now has this many records.

Manage metadata
: Manage all of the metadata associated with all of the above, making it easy to track, search, view, and act upon all of your data. Because you are using an integrated approach, much of technical metadata can be discovered from the data coming in, and the operational data can be automatically captured without any manual steps. This capability provides you with a much more streamlined approach for collecting metadata.

CHAPTER 5

Deriving Value from the Data Lake

The purpose of a data lake is to provide value to the business by serving users. From a user perspective, these are the most important questions to ask about the data:

- What is in the data lake (the catalog)?
- What is the quality of the data?
- What is the profile of the data?
- What is the metadata of the data?
- How can users do enrichments, clean ups, enhancements, and aggregations without going to IT (how to use the data lake in a self-service way)?
- How can users annotate and tag the data?

Answering these questions requires that proper architecture, governance, and security rules are put in place and adhered to, so that the right people get access to the right data in a timely manner. There also needs to be strict governance in the onboarding of data sets, naming conventions have to be established and enforced, and security policies have to be in place to ensure role-based access control.

Self-Service

For our purposes, *self-service* means that non-technical business users can access and analyze data without involving IT.

In a self-service model, users should be able to see the metadata and profiles and understand what the attributes of each data set mean. The metadata must provide enough information for users to create new data formats out of existing data formats, using enrichments and analytics.

Also, in a self-service model, the catalog will be the foundation for users to register all of the different data sets in the data lake. This means that users can go to the data lake and search to find the data sets they need. They should also be able to search on any kind of attribute—for example, on a time window such as January 1st to February 1st—or based on a subject area, such as marketing versus finance. Users should also be able to find data sets based on attributes—for example, they could enter, "Show me all of the data sets that have a field called discount or percentage."

It is in the self-service capability that best practices for the various types of metadata come into play. Business users are interested in the business metadata, such as the source systems, the frequency with which the data comes in, and the descriptions of the data sets or attributes. Users are also interested in knowing the technical metadata: the structure and format and schema of the data.

When it comes to *operational data*, users want to see information about *lineage*, including when the data was ingested into the data lake, and whether it was raw at the time of ingestion. If the data was not raw when ingested, users should be able to see how was it created, and what other data sets were used to create it. Also important to operational data is the *quality* of the data. Users should be able to define certain rules about data quality, and use them to perform checks on the data sets.

Users may also want to see the ingestion history. If a user is looking at streaming data, for example, they might search for days where no data came in, as a way of ensuring that those days are not included in the representative data sets for campaign analytics. Overall, access to lineage information, the ability to perform quality checks, and ingestion history give business users a good sense of the data, so they can quickly begin analytics.

Controlling and Allowing Access

When providing various users—whether C-level executives, business analysts, or data scientists—with the tools they need, security is critical. Setting and enforcing the security policies, consistently, is essential for successful use of a data lake. In-memory technologies should support different access patterns for each user group, depending on their needs. For example, a report generated for a C-level executive may be very sensitive, and should not be available to others who don't have the same access privileges. In addition, you may have business users who want to use data in a low-latency manner because they are interacting with data in real time, with a BI tool; in this case, they need a speedy response. Data scientists may need more flexibility, with lesser amounts of governance; for this group, you might create a sandbox for exploratory work. By the same token, users in a company's marketing department should not have access to the same data as users in the finance department. With security policies in place, users only have access to the data sets assigned to their privilege levels.

You may also use security features to enable users to interact with the data, and contribute to data preparation and enrichment. For example, as users find data in the data lake through the catalog, they can be allowed to clean up the data, and enrich the fields in a data set, in a self-service manner.

Access controls can also enable a collaborative approach for accessing and consuming the data. For example, if one user finds a data set that she feels is important to a project, and there are three other team members on that same project, she can create a workspace with that data, so that it's shared, and the team can collaborate on enrichments.

Using a Bottom-Up Approach to Data Governance to Rank Data Sets

The bottom-up approach to data governance, discussed in Chapter 2, enables you to rank the usefulness of data sets by crowdsourcing. By asking users to rate which data sets are the most valuable, the word can spread to other users so they can make productive use of that data. This way, you are creating a single source of truth from the bottom up, rather than the top down.

To do this, you need a rating and ranking mechanism as part of your integrated data lake management platform. The obvious place for this bottom-up, watermark-based governance model would be the catalog. Thus the catalog has to have rating functions.

But it's not enough to show what others think of a dataset. An integrated data lake management and governance solution should show users the rankings of the data sets from *all users*—but it should also offer a personalized data rating, so that each individual can see what they have personally found useful whenever they go to the catalog.

Users also need tools to create new data models out of existing data sets. For example, users should be able to take a customer data set and a transaction data set and create a "most valuable customer" data set by grouping customers by transactions, and figuring out when customers are generating the most revenue. Being able to do these types of enrichments and transformations is important from an end-to-end perspective.

Data Lakes in Different Industries

The data lake provides value in many different areas. Below are examples of industries that benefit from using a data lake to store and access information.

Healthcare

Many large healthcare providers maintain millions of records for millions of patients, including semi-structured reports such as radiology images, unstructured doctors' notes, and data captured in spreadsheets and other common computer applications. A data lake is an obvious solution for such organizations, because it solves the challenge healthcare providers face with data storage, integration, and accessibility. By implementing a data lake based on a Hadoop architecture, a healthcare provider can enable distributed big data processing, by using broadly accepted, open software standards, and massively-parallel commodity hardware.

Hadoop allows healthcare providers' widely disparate records of both structured and unstructured data to be stored in their native formats for later analysis; this avoids the issue of forcing categorization of each data type, as would be the case in a traditional EDW. Not incidentally, preserving the native format also helps maintain data *provenance* and *fidelity* of the data, enabling different analyses to be performed using different contexts. With data lakes, sophisticated data analyses projects are possible, including those using predictive analytics to anticipate and take measures against frequent readmissions.

Financial Services

In the financial services industry, data lakes can be used to comply with the Dodd-Frank regulation. By consolidating multiple EDWs into one data lake repository, financial institutions can move reconciliation, settlement, and Dodd-Frank reporting to a single platform. This dramatically reduces the heavy lifting of integration, as data is stored in a standard yet flexible format that can accommodate unstructured data.

Retail banking also has important use cases for data lakes. In retail banking, large institutions need to process thousands of applications for new checking and savings accounts on a daily basis. Bankers that accept these applications consult third-party risk scoring services before opening an account, yet it is common for bank risk analysts to manually override negative recommendations for applicants with poor banking histories. Although these overrides can happen for good reasons (say there are extenuating circumstances for a particular person's application), high-risk accounts tend to be overdrawn and cost banks millions of dollars in losses due to mismanagement or fraud.

By moving to a Hadoop data lake, banks can store and analyze multiple data streams, and help regional managers control account risk in distributed branches. They are able to find out which risk analysts were making account decisions that went against risk information by third parties. The net result is better control of fraud. Over time, the accumulation of data in the data lake allows the bank to build algorithms that detect subtle but high-risk patterns that bank risk analysts may have previously failed to identify.

Retail

Data lakes can also help online retail organizations. For example, retailers can store all of a customer's shopping behavior in a data lake in Hadoop. By capturing web session data (session histories of all users on a page), retailers can do things like provide timely offers based on a customer's web browsing and shopping history.

CHAPTER 6
Looking Ahead

As the data lake becomes an important part of next-generation data architectures, we see multiple trends emerging based on different vertical use cases that indicate what the future of data lakes will look like.

Ground-to-Cloud Deployment Options

Currently, most data lakes reside on-premises at organizations, but a growing number of enterprises are moving to the cloud because of the agility, ease of use, and economic benefits of a cloud-based platform. As clouds—both private and public—mature from security and multi-tenancy perspectives, we'll see this trend intensify, and it's important that data lake tools work across both environments.

As a result, we're seeing an increased adoption of cloud-based Hadoop infrastructures that complement and sometimes even replace on-premises Hadoop deployments. As data onboarding, management, and governance matures and becomes easier, data needs to be accessible in cloud-based architectures the same way it is available in on-premises architectures. Most data lake vendors are extending their tools so they work seamlessly across cloud and physical environments. This allows business users and data scientists to spin up and down clusters in the cloud, and create augmented platforms for both agile analytics and traditional queries.

With a cloud-to-ground environment, you have a hybrid architecture that may be useful for organizations that have yet to build their

own private clouds. It can be used to store sensitive or vulnerable data that organizations can't trust to a public cloud environment. At the same time, other, less-sensitive data sets can be moved to the public cloud.

Looking Beyond Hadoop: Logical Data Lakes

Another key trend is the emergence of *logical data lakes.* A logical data lake provides a *unified view* of data that exists across multiple data stores and across multiple environments in an enterprise.

In early Hadoop use cases, batch processing using MapReduce was the norm. Now in-memory technologies like Spark are becoming predominant, as they fit low-latency use cases that previously couldn't be accopm;ished in a MapReduce architecture. We're also seeing hybrid data stores, where data can be stored not only in HDFS, but also in object data stores such as S3 from Amazon, or Azure Elastic Block storage, or No-SQL databases.

Federated Queries

Federated queries go hand-in-hand with logical data lakes. As data is stored in different physical and virtual environments, you may need to use different query tools, and decompose a user's query into multiple queries—sending them to on-premises data stores as well as cloud-based data stores, each of which possess just part of the answer. Federated queries allow answers to be aggregated and combined, and sent back to the user so she gets one version of the truth across the entire logical data lake.

Data Discovery Portals

Another trend is to make data available to consumers via rich metadata data catalogs put into a data-as-a-service framework. Many enterprises are already building these portals out of shared Hadoop environments, where users can browse what data is available in the data lake, and have an Amazon-like shopping cart experience where they select data based on various filters. They can then create a sandbox for that data, perform exploratory ad hoc analytics, and feed the results back into the data lake to be used by others in the organization.

In Conclusion

Hadoop is an extraordinary technology. The types of analyses that were previously only possible on costly proprietary software and hardware combinations as part of cumbersome EDWs are now being leveraged by organizations of all types and sizes simply by deploying free open-source software on commodity hardware clusters.

Early use cases for Hadoop were trumpeted as successes based on their low cost and agility. But as more mainstream use cases emerged, organizations found that they still needed the management and governance controls that dominated in the EDW era. The data lake has become a middle ground between EDWs and "data dumps" in offering systems that are still agile and flexible, but have the safeguards and auditing features that are necessary for business-critical data.

Integrated data lake management solutions like Zaloni's Bedrock and Mica are now delivering the necessary controls without making Hadoop as slow and inflexible as its predecessor solutions. Use cases are emerging even in sensitive industries like healthcare, financial services, and retail.

Enterprises are also looking ahead. They see that to be truly valuable, the data lake can't be a silo, but must be one of several platforms in a carefully considered end-to-end modern enterprise data architecture. Just as you must think of metadata from an enterprise-wide perspective, you need to be able to integrate your data lake with external tools that are part of your enterprise-wide data view. Only then will you be able to build a data lake that is open, extensible, and easy to integrate into your other business-critical platforms.

A Checklist for Success

Are you ready to build a data lake? Here is a checklist of what you need to make sure you are doing so in a controlled yet flexible way.

Business-benefit priority list
: As you start a data lake project, you need to have a very strong alignment with the business. After all, the data lake needs to provide value that the business is not getting from its EDW. This may be from solving pain points or of creating net new

revenue streams that you can enable business teams to deliver. Being able to define and articulate this value from a business standpoint, and convince partners to join you on the journey is very important to your success.

Architectural oversight
: Once you have the business alignment and you know what your priorities are, you need to define the upfront architecture: what are the different components you will need, and what will the end technical platform look like? Keep in mind that this is a long-term investment, so you need to think carefully about where the technology is moving. Naturally, you may not have all the answers upfront, so it might be necessary to perform a proof of concept to get some experience and to tune and learn along the way. An especially important aspect of your architectural plans is a good data-management strategy that includes data governance and metadata, and how you will capture that. This is critical if you want to build a managed and governed data lake instead of the much-maligned "data swamp."

Security strategy
: Outline a robust security strategy, especially if your data lake will be a shared platform used by multiple lines of business units or both internal and external stakeholders. Data privacy and security are critical, especially for sensitive data such as PHI and PII. You may even have regulatory rules you need to conform to. You must also think about multi-tenancy: certain users might not be able to share data with other users. If you are serving multiple external audiences, each customer might have individual data agreements with you, and you need to honor them.

I/O and memory model
: As part of your technology platform and architecture, you must think about what the scale-out capabilities of your data lake will look like. For example, are you going to use decoupling between the storage and the compute layers? If that's the case, what is the persistent storage layer? Already, enterprises are using Azure or S3 in the cloud to store data persistently, but then spinning up clusters dynamically and spinning them down again when processing is finished. If you plan to perform actions like these, you need to thoroughly understand the throughput requirements from a data ingestion standpoint, which will dictate throughput

for storage and network as well as whether you can process the data in a timely manner. You need to articulate all this upfront.

Workforce skillset evaluation
: For any data lake project to be successful, you have to have the right people. You need experts who have hands-on experience building data platforms before, and who have extensive experience with data management and data governance so they can define the policies and procedures upfront. You also need data scientists who will be consumers of the platform, and bring them in as stakeholders early in the process of building a data lake to hear their requirements and how they would prefer to interact with the data lake when it is finished.

Operations plan
: Think about your data lake from an SLA perspective: what SLA requirements will your business stakeholders expect, especially for business-critical applications that are revenue-impacting? You need proper SLAs in terms of lack of downtime, and in terms of data being ingested, processed, and transformed in a repeatable manner. Going back to the people and skills point, it's critical to have the right people with experience managing these environments, to put together an operations team to support the SLAs and meet the business requirements.

Communications plan
: Once you have the data lake platform in place, how will you advertise the fact and bring in additional users? You need to get different business stakeholders interested and show some successes for your data lake environment to flourish, as the success of any IT platform ultimately is based upon business adoption.

Disaster recovery plan
: Depending on the business criticality of your data lake, and of the different SLAs you have in place with your different user groups, you need a disaster recovery plan that can support it.

Five-year vision
: Given that the data lake is going to be a key foundational platform for the next generation of data technology in enterprises, organizations need to plan ahead on how to incorporate data lakes into their long-term strategies. We see data lakes taking over EDWs as organizations attempt to be more agile and gen-

erate more timely insights from more of their data. Organizations must be aware that data lakes will eventually become hybrids of data stores, include HDFS, no-SQL, and Graph DBs. They will also eventually support real-time data processing and generate streaming analytics—that is, not just rollups of the data in a streaming manner, but machine-learning models that produce analytics online as the data is coming in and generate insights in either a supervised or unsupervised manner. Deployment options are going to increase, also, with companies that don't want to go into public clouds building private clouds within their environments, leveraging patterns seen in public clouds. Across all these parameters, enterprises need to plan to have a very robust set of capabilities, to ingest and manage the data, to store and organize it, to prepare and analyze, secure, and govern it. This is essential no matter what underlying platform you choose—whether streaming, batch, object storage, flash, in-memory, or file—you need to provide this consistently through all the evolutions the data lake is going to undergo over the next few years.

About the Authors

Ben Sharma, CEO and cofounder of Zaloni, is a passionate technologist with experience in solutions architecture and service delivery of big data, analytics, and enterprise infrastructure solutions. His expertise ranges from business development to production deployment in technologies including Hadoop, HBase, databases, virtualization, and storage.

Alice LaPlante is an award-winning writer who has been writing about technology and the business of technology for more than 30 years. Author of seven books, including *Playing For Profit: How Digital Entertainment is Making Big Business Out of Child's Play* (Wiley), LaPlante has contributed to *InfoWorld*, *ComputerWorld*, *InformationWeek*, *Discover*, *BusinessWeek*, and other national business and technology publications.