Gartner®

Building Data Lakes Successfully — Part 2 — Consumption, Governance and Operationalization

Refreshed 12 April 2022, Published 7 October 2020 - ID G00733127 - 43 min read

FOUNDATIONAL This research is reviewed periodically for accuracy.

By Analyst(s): Sumit Pal

Initiatives: Data Management Solutions for Technical Professionals; Chief Data and Analytics Officer Leadership

Data lakes need proper governance, operationalization and data catalogs to facilitate data consumption. This research helps data and analytics technical professionals understand the challenges and best practices around these three important pillars of a data lake.

Overview

Key Findings

- Organizations that have successfully developed data lakes have focused on the cross-cutting concerns around a data lake — primarily ensuring governance, leveraging data catalogs and carefully building end-to-end orchestration of the data lake.
- Though data governance is the key to successfully delivering modern data lakes, most organizations are struggling to accomplish end-to-end data governance in a data lake. Existing tools only solve a part of the governance, and integrating different tools is cumbersome and brittle.
- Organizations repeatedly run into the last-mile problem with data lakes. They face challenges — such as lack of a single product or framework — in building end-to-end, automated, agile and repeatable data-driven systems. This is causing organizations to duct-tape different products and frameworks together to build their solutions and delivery processes.

This research note is restricted to the personal use of .

Recommendations

Technical professionals responsible for data management strategies should consider the following:

- Incorporate effective governance early in the process to avoid potential pitfalls of data lakes. Pitfalls include poor accessibility, poor metadata management, poor data quality, and lack of lineage and data security.
- Enable data democratization by establishing a metadata layer and data catalogs to facilitate governed data discovery, which is imperative to address data accessibility and glean insights into context and analytics that will be delivered to the enterprise.
- Implement comprehensive automation of deployment and be able to reproduce and roll back deployed components in the data lake, from code to data and infrastructure.
- Incorporate a configuration-driven process by leveraging containerization of images for various environments. This is especially recommended in hybrid and multicloud-based data lake architectures.

Analysis

This document was revised on 15 October 2020. The document you are viewing is the corrected version. For more information, see the Corrections page on gartner.com.

Beyond the basic components of a data lake — ingestion, storage and data processing, there are other critical requirements that organizations should address before data lakes are ready for the different personas who would like to use the data lake. Data lakes should allow organizations to:

- Securely consume the ingested and processed data with the ability to easily search, discover, query the data in place and integrate the data in a data lake with other downstream systems. Data lakes should provide access to the data through a broad and deep portfolio of analytics, data science, machine learning and visualization tools.
- Enable end-to-end orchestration with a configuration-driven approach to deploy data pipelines, orchestrate, schedule and manage the workflow. Continually monitor and provide tools to manage, maintain, provision and execute the data operations on the data lake. Perform ongoing operations and incorporate performance management, metrics gathering from data-intensive applications and infrastructure usage.

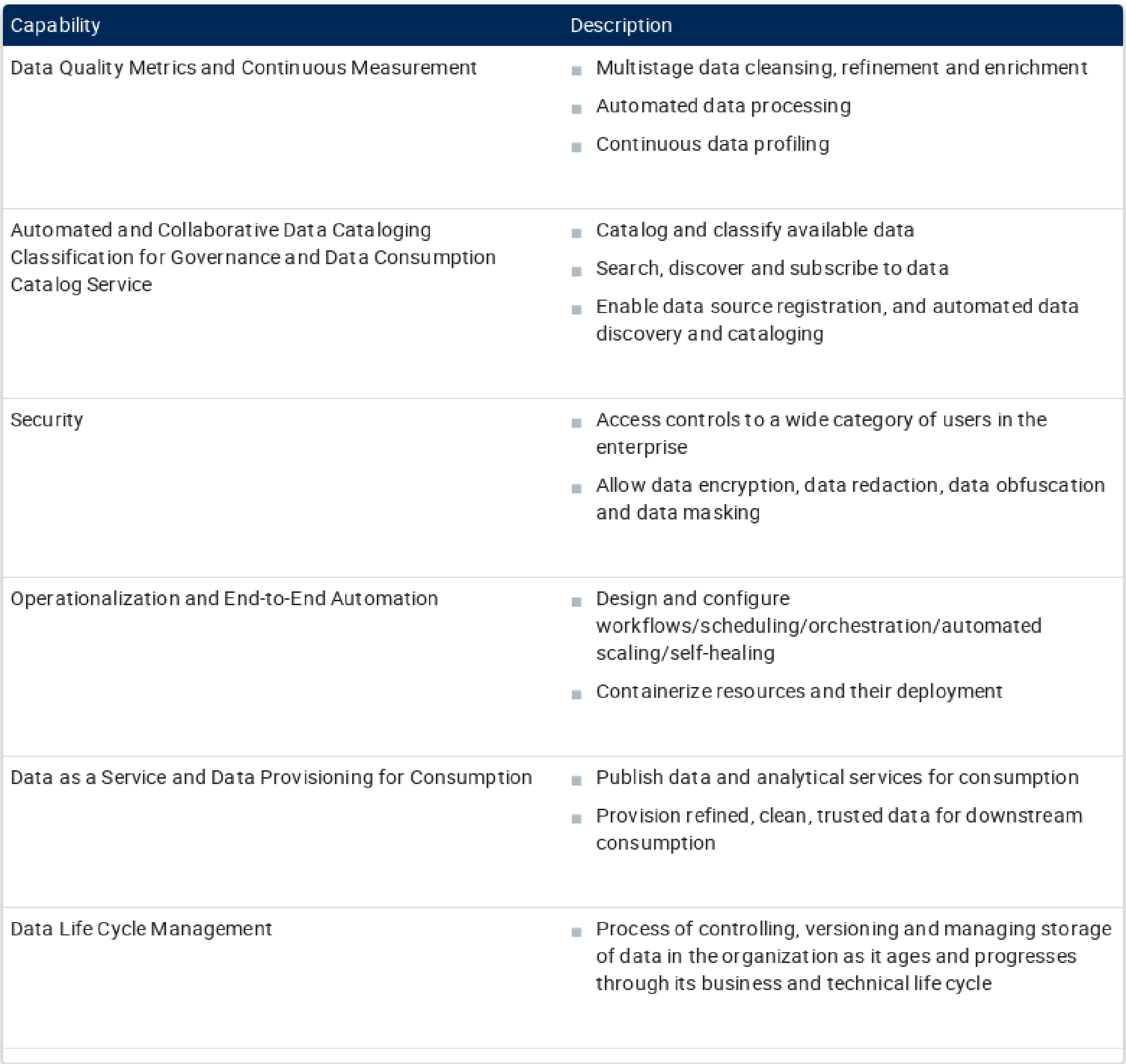
This research note is restricted to the personal use of .

- Govern access to the data lake. This includes managing data quality, data catalog, data security, data lineage and auditing, and the data life cycle.

The overall goal of the data lake should be to allow ingestion of any data, perform any analytics and allow access to any user. While the first part of this research addressed data lake architecture, data ingestion, storage and processing (see Building Data Lakes Successfully — Part 1 — Architecture, Ingestion, Storage and Processing), this document focuses on the bulleted items above and the best practices, challenges and pitfalls associated with these steps. When architecting a data lake, note some of the key capabilities around governance and operationalization listed in Table 1.

Table 1: Key Capabilities of a Data Lake

(Enlarged table in Appendix)



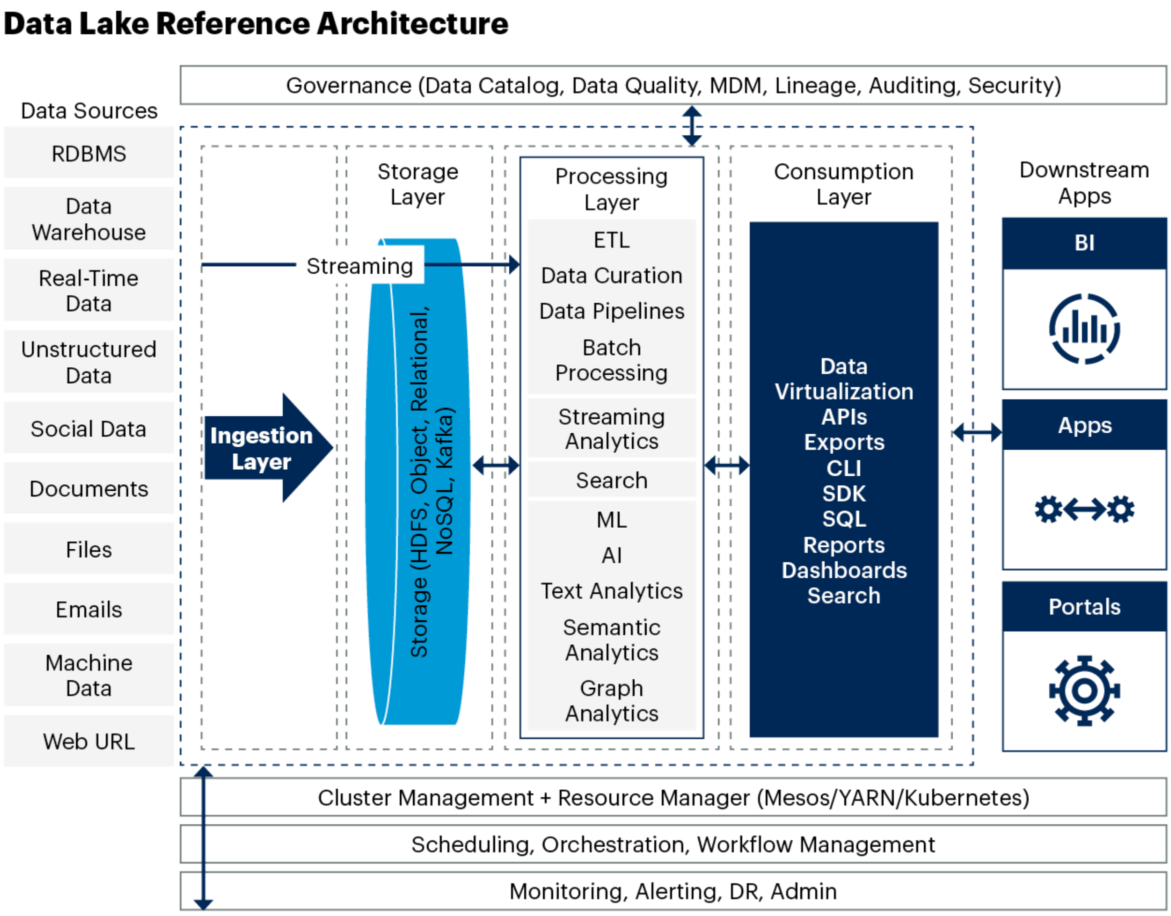
Capability	Description
Data Quality Metrics and Continuous Measurement	- Multistage data cleansing, refinement and enrichment - Automated data processing - Continuous data profiling
Automated and Collaborative Data Cataloging Classification for Governance and Data Consumption Catalog Service	- Catalog and classify available data - Search, discover and subscribe to data - Enable data source registration, and automated data discovery and cataloging
Security	- Access controls to a wide category of users in the enterprise - Allow data encryption, data redaction, data obfuscation and data masking
Operationalization and End-to-End Automation	- Design and configure workflows/scheduling/orchestration/automated scaling/self-healing - Containerize resources and their deployment
Data as a Service and Data Provisioning for Consumption	- Publish data and analytical services for consumption - Provision refined, clean, trusted data for downstream consumption
Data Life Cycle Management	- Process of controlling, versioning and managing storage of data in the organization as it ages and progresses through its business and technical life cycle

Source: Gartner (October 2020)

This research note is restricted to the personal use of .

A high-level data lake architecture, along with key functional capabilities and major components needed to build a data lake, is shown in Figure 1.

Figure 1: Data Lake Reference Architecture



Source: Gartner
733127_C

Consumption

Data lakes can ingest and process volumes of data, but the real value of a data lake is realized only when the processed data is consumed to build data-driven applications. Making the data available to downstream applications and consumers in the data lake lays the foundations for innovation and building data-driven applications. Easing the process of accessing the data lake allows business transformations to come from anyone in any line of business, not just your data scientists. One of the main requirements from a data lake is to cater to a variety of consumers. Each consumer has different requirements regarding the attributes they want and in what format.

This research note is restricted to the personal use of .

A data lake is successful only when it allows rapid access and consumption with the right governance onboarding of users, irrespective of their location and which tools they use. Data lakes should support a wide variety of users from data scientists, data engineers, data and business analysts, and product teams. These consumers should be able to point their tools and access the data without development or operational skills. Consumers of data from a data lake consist of two types:

- Processes and workloads *within* the data lake. For example:
  - Extraction, transformation and loading (ETL) workloads
  - Machine learning (ML) algorithms
  - Data-lake-based data marts
  - Search-based access
  - Data processing and analysis tools
  - Data governance, data catalog, security, metadata and MDM tools within the data lake
- Processes and workloads *outside* the data lake. For example:
  - Data pushed to data warehouses and/or data marts
  - Access to data visualization tools and business intelligence (BI) tools
  - Search-based access
  - Data analysis tools
  - Data governance, data catalog, security, metadata and MDM tools outside the data lake

Data from a data lake can be accessed in two ways:

- Data push:
  - Data exports
  - Data publish to message queue to be picked up by downstream processes

This research note is restricted to the personal use of .

- Data pull:
  - Data services
  - Data views
  - SQL access (BI tools)
  - REST API
  - GraphQL
  - Data virtualization tools
  - Data marketplaces
  - Data services that provide a mechanism to deliver data based on a contract to a consuming application over lightweight protocols, as a pull-based mechanism from the consuming applications

Questions to Ask

Data lake architects and engineers should ask the following questions before architecting the consumption layer of a data lake:

- What are the data access throughput and latency requirements to the consuming applications?
- What are the SLAs for the throughput and low-latency reads and updates?
- What kind of access is required by the consuming application — random reads, sequential reads, random writes or sequential writes?
- What indexing capability allows for random reads and serving of a small portion of a larger dataset quickly?
- How will you architect the data consumption layer to handle scalability (of data push or pull) and concurrency?
- How will you ensure fault tolerance and handle both hardware and software failures?
- How will you ensure that the data lake is capable of serving multiple data models?

This research note is restricted to the personal use of .

- What are the data structures or data stores that need to be created to make the access easier, faster, scalable and performant?
- Can the consumers shop for and prepare the data just like in a data marketplace?
- Can the consumers search the data lake comprehensively either directly or through a catalog?
- Is the data access automatically governed for data security with the right kind of access policies?
- Is the data access logged, audited and monitored, and are metrics around these captured and stored?
- How will you expose the data quality and data profile results to the data consumers to gain their trust of the data in the data lake?

Best Practices

This section provides some of the best practices that organizations, data lake architects and data engineers should follow when building the consumption layer of the data lake:

- Integrate the consumption layer with a data catalog/data governance and security tools to control access policies/authorization/authentication. Setting and enforcing security policies are essential for successful use of data within a data lake. The data owners configure the security access requirements of the data, detailing who can access what data. The security details are stored as part of metadata, which is used by data stewards to enforce these policies.
- Support the most common ways of data access from the data lake, including SQL, APIs, search, exports and bulk access.
- Ensure self-service capabilities, which are 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.
- Track lineage, auditing and logging of what data is consumed by whom as well as when, where and how.

This research note is restricted to the personal use of .

- Maintain a data catalog that describes the content, data definition and all the metadata that is captured as the data moves in the data pipeline from data ingestion to data enrichment, integration, transformation and consumption. This should be kept up to date with automated mechanisms and published by the data steward to the stakeholders.

Challenges

Some of the challenges that data lake architects and engineers face when designing the consumption layer include:

- Designing for high concurrent access to the data stored with a data lake across the data stores
- Managing the data model and schema, and avoiding data, model and schema inconsistencies
- Ensuring data findability by implementing the right kind of indexing, tagging and search capabilities
- Supporting self-service across the different consumers of the data lake
- Ensuring security of the data being accessed both by internal and external users
- Ensuring continuous maintenance of the quality of the data being accessed
- Enforcing logging, tracking and auditing of the data access, as well as integrating with tools that support these capabilities

Operations

Building a data lake may be difficult and complex, but sustaining it is even more difficult. Managing a production data lake can easily become complex and foggy if the right kind of tools and an efficient monitoring-and-alerting framework are not in place. Key elements you must consider are monitoring the operations of the data lake, making sure that it meets performance expectations and SLAs, analyzing utilization patterns, and using this information to optimize the cost and performance of your data lake. Develop a roadmap of the operational requirements and prioritize those requirements for successful management and operations of the data lake environment. Customers can incorporate the requirements incrementally as they continue to mature their data lake environment.

This research note is restricted to the personal use of .

Due to the enormous complexity of building a data lake from ideation to production, organizations should realign their focus in building data lakes from a component-based architecture to an application-centric approach. The idea is to reduce the time to deployment, reduce the time to market and increase the repeatability of the process.

This section briefly discusses the operationalization aspects of a data lake. For more details, please refer to Operationalizing Big Data Workloads.

Questions to Ask

Data lake architects and engineers should ask the following questions before operationalizing a data lake:

- Where is the deployment going to happen — cloud or on-premises?
- What resource management tool is to be used for the data lake cluster — Yarn, Mesos, Kubernetes?
- What are the capacity requirements of the data lake — for storage, types of storage, compute, networking and IO?
- What is/are the data pipeline tools to be used — Airflow, Prefect or others?
- What are the tools to manage CI/CD pipelines for the data lake?
- What are the tools to manage automation of the deployments?
- What are the tools to be used for artifact repository and configuration management?

Best Practices

- Improve efficiency by using a repeatable and replayable workflow that is configuration-driven, without needing to write code.
- Employ automated deployments with single-click deployments and the capability to roll back easily.
- Introduce proactive monitoring and alerting to keep constant check on platform availability and critical metrics. Provide operational intelligence and publish metrics to highlight availability, trends, SLA violations and so on.
- Perform incident analysis and a health checkup of the data lake in an automated way.

This research note is restricted to the personal use of .

- Before building the data lake, perform the exercise related to cluster planning, capacity planning, and cluster design and component layouts. Have a map of the resources available on the data lake and the different tools, software and frameworks that are deployed.

- Ensure that primary processes for components utilizing a primary/secondary architecture are distributed across racks to minimize risk due to rack failures. Deploy multiple gateway nodes for load balancing and client operations.

- Automate the provisioning and deployment processes through tools like Chef, Jenkins, Puppet and Red Hat Ansible.

- Deploy all security components in high-availability (HA) mode.

- Leverage DevOps and “DataOps” teams to use continuous integration, continuous delivery and continuous deployment principles to ensure a seamless data lake deployment process.

- Automate boilerplate code and configuration generation — ensuring that infrastructure, permissions and deployment setup are abstracted away from users.

- Use containerization of workloads and apply container orchestration tools like Kubernetes to ensure better resource utilization, multitenancy, resource isolation, autoscaling and self-healing where appropriate.

- To enable the best out of the data lake, thoroughly integrate and coordinate different layers of the data lake. This can be extremely challenging, especially for hybrid data lakes.

- For data lakes on the cloud, ensure you have processes in place for ongoing performance and cost management. The data lake technology stack is continually evolving across multiple data ingestion, storage and processing engines. Understanding the cost to performance requirements and implementation should be carefully managed and controlled.

- Look to leverage third-party SaaS vendors like Cazena or Zaloni to streamline data orchestration and self-service access to a data lake, both for cloud and on-premises deployments.

Challenges

- Failed jobs more often than not leave data in a corrupted state, and it is challenging to recover the data.

This research note is restricted to the personal use of .

- Lack of schema enforcement within a data lake can create inconsistent and low-quality data.

Data Governance

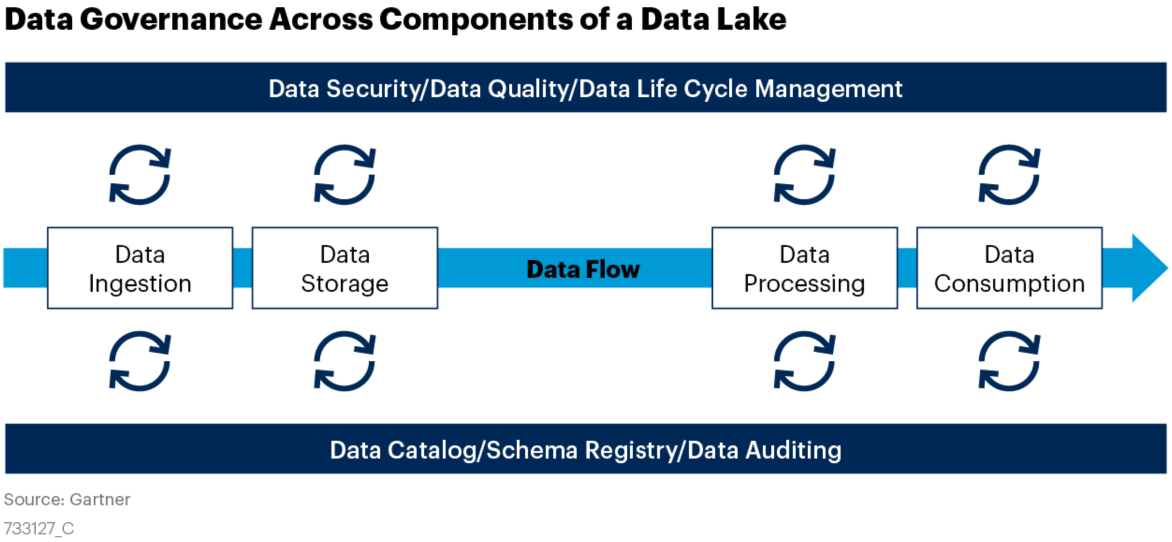
Data governance gives the right control and trust in the data within the data lake. This gives confidence to the consumers of data in the data lake to make sound business decisions. Data governance is a set of formal processes that ensure that data within the enterprise meets expectations, such as:

- Data is acquired from reliable sources.
- Data meets quality standards.
- Data conforms to well-defined business rules.
- Data is accessed and modified with the right policies, guidelines and access controls.
- Data follows a well-documented change control process.
- Trustworthiness of the data remains intact as the data flows through the data lake.

Figure 2 shows the different forms of data governance as it is applied across the different components of a data lake.

This research note is restricted to the personal use of .

Figure 2: Data Governance Across Components of a Data Lake


Data Governance Across Components of a Data Lake
Data Security/Data Quality/Data Life Cycle Management
Data Ingestion
Data Storage
Data Flow
Data Processing
Data Consumption
Data Catalog/Schema Registry/Data Auditing
Source: Gartner
733127_C

At a high level, the best practices for data governance for a data lake should include:

- Ensure that data access policies and governance solutions are not based on the storage system or analytics engine being used. Instead, the solution must be data-centric and enable the consistent enforcement of policies using the tools you have deployed today, as well as those you may deploy in the future.
- Ensure governance across all components of the data lake. Use data catalogs to ensure data quality at ingestion time, secure communication and every component within the data lake, and ensure the data life cycle is managed with the right kind of policies. Also, automate, configure, coordinate and control the operationalization of data lakes (see Operationalizing Big Data Workloads).

Data governance has four major pillars: data quality, data catalog, data security and data life cycle management. The following sections discuss details related to each of these pillars.

Data Quality

The need for data quality is not new nor specific to data lakes. Data lakes make fixing data quality issues both urgent and challenging for a few reasons:

- The unstructured nature of the data ingested into the data lake

This research note is restricted to the personal use of .

- Volume
- Velocity

These factors make ensuring data quality in a data lake a more challenging proposition and sometimes more elusive. Correctness is difficult to determine when using data from external sources, and structural integrity can be difficult to test with unstructured and differently structured (nonrelational) data. Validating all data entering the lake is hard. Each source features unique technical challenges and a particular set of data issues, necessitating the development of specific data validation functions for each source.

Ensuring data quality involves:

- Data quality detection
- Data quality remediation (ignore/correct)
- Automated detection, flagging
- Setup rules — configuration-driven
- Data rejection
- Profiling and classification of data based on rules (good, bad or ugly)
- Automated data quality detection and remediation

Questions to Ask

- How should you determine the data quality level for the data sources in the data lake?
- What tools should you use for data quality detection, data quality correction and automation?
- How often is the data quality to be assessed?
- Is the data accurate, complete, valid and/or consistent?
- How can you identify and implement data quality rules for monitoring and improvement?

Best Practices

This research note is restricted to the personal use of .

- Ensure data quality across all layers of the data lake.

- Avoid data lake indigestion. Identify, fix and flag data quality problems at the point of ingestion by building a robust set of data validation rules for each data source as it is loaded into the data lake. Validation rules compare the contents of each field in each record with a set of parameters and thresholds to determine whether that record contains quality data.

- Ensure you maintain an exception queue, dead-letter queue and error-queue-based pattern to handle data oddities.

- Do not skip automatic data validation and profiling during the ingestion process. This is the single root cause of failure of many data lake projects. Just because it is easy to ingest data into a data lake, do not fall into the trap of dumping bad quality data into a data lake for experimentation purposes.

- Integrate data quality and profiling closely with data cataloging tools by specifying data quality, profiling and distribution metrics in a data catalog and by cross-checking data during and after ingestion.

- Implement rules and systems to continually monitor data quality.

- With the help of data stewards, have controls in place and steps to be taken when data quality is not to the required level.

- Always ensure data quality as a precursor to the correlation of data across different siloed sources and before scheduling long-running data pipelines.

- Automate data quality detection when orchestrating the data pipelines. It does not make sense to run expensive data processing — which consumes time, resources and money — on bad quality data.

- Automate data quality rules just like operators in a programming language. Organize rules into rule sets.

Challenges

- If the data in the data lake is not validated and profiled, it results in data liability, where the ingested data is questionable to use in terms of its quality and usefulness.

- Data quality issues in a data lake are generally tied to delays in making data available and (ultimately) the delivery of bad data.

- Not ensuring proper data quality within a data lake erodes consumer confidence.

This research note is restricted to the personal use of .

Data Catalog

The single most important reason data lake implementations fail is because data lakes degenerate into data swamps due to a lack of proper metadata around them. Data lakes are reincarnated by making data catalogs an integral part of the ecosystem that is integrated across all layers and components. A data catalog should be able to answer the following question for a data lake:

What data is there, what does it mean, where is it used and who has responsibility for it?

Context is everything when it comes to data in the data lake. A data catalog helps companies build a map to organize and locate data stored in the data lake. It contains details on each dataset, both at a high level — in terms of its origin, date and time of ingestion, and who ingested — and on a granular level regarding each piece of data, such as the data's profile and quality metrics, its data types and their lineage. The catalog is the first tool anyone looking to work with the data lake should use to find data to build insights.

A data catalog solves multiple problems, such as:

- Gives a comprehensive view of each piece of data across databases
- Makes the data easy to find
- Puts guardrails on the data and governs who can access it
- Enables the data lake to support different use cases such as self-service analytics, self-service data preparation, data virtualization and multicloud-based data architectures

Data catalog vendors are innovating by building catalogs powered by AI, ML and knowledge graphs that:

- Generate technical metadata
- Enable semantic inference and recommendations
- Catalog business context
- Discover entities within unstructured data
- Manage highly complex relationships, data quality and data lineage

This research note is restricted to the personal use of .

Metadata management tools and business glossaries are being integrated and are evolving as holistic data catalog tools. The data catalog becomes an indispensable guide to data, unlocking its potential and enabling organizations to be data-driven.

The data catalog guides the producers and consumers of a data lake to find the right data for their interest. It provides clarity in the form of agreed definitions, supplied by subject matter experts, with a high-level view of the interconnectedness between data and the business processes that it supports. Data can be both a liability and an asset. A data catalog is an important component to becoming a data-driven organization. It is not enough to simply have lots of data. The goal should be to maximize its findability and proper usage through ownership, understanding and trust.

Questions to Ask

- What are the data catalog access mechanisms?

1. REST APIs to integrate with other tools
2. Self-service capabilities
3. Searching and findability capabilities

- What kind of search capabilities are provided by the data catalog? With the volume and complexity of data in a data lake, it is extremely important to have a data catalog that can be searched efficiently in multiple ways across different tools and applications. It is best if the search capabilities of a data catalog can be integrated with other applications in an organization that needs to access the data lake. Nontechnical users need to search the catalog to find the data useful for their specific purpose.
- What kind of automated data discovery is possible?
  - Automated data profiling
  - Automated derivation of lineage
  - Automated field-level tagging
  - Automated data quality detection based on rules and thresholds

This research note is restricted to the personal use of .

- What are the relationships/dependencies that exist between the various datasets and the entities of the datasets?
- How do you determine the current data quality of the data in the data lake?

Best Practices

Some of the best practices around data catalogs in a data lake are outlined below:

- Make sure all datasets are labeled, tagged and classified, which is imperative for successful usage of the data within the lake.
- Ideally, have a single, enterprisewide, scalable, multicloud data catalog. If this is not possible and your organization has multiple catalog tools, research into integrating them to reduce duplication and ambiguity.
- Have an open API where tools, applications and other components of the data lake or logical data warehouse (LDW) can seamlessly integrate with each other.
- Keep your data catalog current. Data lakes are constantly changing and evolving, with new data sources being added continually. A data catalog should be an updated navigation map for the entire data lake and must be in sync with the data that currently exists in the data lake.
- Ideally, have a data catalog that integrates with data quality tools to measure data quality, enable correction of errors and provide data that is fit for usage.

This research note is restricted to the personal use of .

- When choosing a data cataloging tool for a data lake, consider these important capabilities:
    - Support native big data processing for performance and scalability.
    - Automate data discovery and classification, both for initial data loads and for ongoing discovery.
    - Catalog the data ingestion, data processing, data lineage and data security details and profiles.
    - Integrate with other enterprise metadata repositories.
    - Establish collaboration capabilities across a variety of users and the ability to annotate the contents of the data catalog at different granularity.
    - Keep an inventory of what data is there, where it is, and its format, sensitivity, profile and time stamp — all built from the technical metadata.
    - Select a solution that can connect to the widest range of data sources.
    - Equip the data steward with tools to maintain data compliance.
    - Search for data across the data lake based on various search criteria, including keywords, similarity searching, and facet-based and tag-based searching.
    - Select a data catalog fueled by ML and AI, with automated data discovery and autogenerated recommendations to view and explore similar datasets.

Challenges

Organizations that neglect to build and integrate a data catalog within a data lake will eventually create a data swamp and underutilize the data within a data lake. This leads to:

- Lack of trust in data
- Poor awareness of what data actually exists
- Confusion around the meaning of data
- Data becoming obsolete due to lack of active ownership

Data Security

This research note is restricted to the personal use of .

Organizations building data lakes with large amounts of data from a variety of sources need to carefully protect their data assets just as they would with natural lakes, which have rules and regulations to ensure the safety of swimmers.

Important aspects of privacy and security of the data in a data lake include:

- A greater volume of data in a data lake implies a larger surface area of risk and attacks.
- Newer data types like sensor data, connected home data, fitness data, telematics and so on have different privacy and risk implications than traditional datasets.
- Performing data integration, enrichment and linkage with sensitive data in the lake can often unknowingly result in exposure of private data.
- Combining data can inadvertently identify individuals in a way that the different types of aggregate data cannot.
- Data discovery and experimental and exploratory analysis in the data lake can create unanticipated risk exposure to private data.

Security solutions in a data lake should provide, at minimum, the following capabilities:

- Richness of access policies
- Built for scale and automation
- Data obfuscating and redaction capabilities
- Data auditability
- Unified visibility — alerting, monitoring and logging
- Hybrid and multicloud-ready, governance, risk assessment, risk management and compliance capabilities (a data lake would typically retain data for a long period of time, and that data might be sent across to multiple cloud providers and back and forth with on-premises solutions)
- Data disposal

Security in a data lake can be broadly split into three areas:

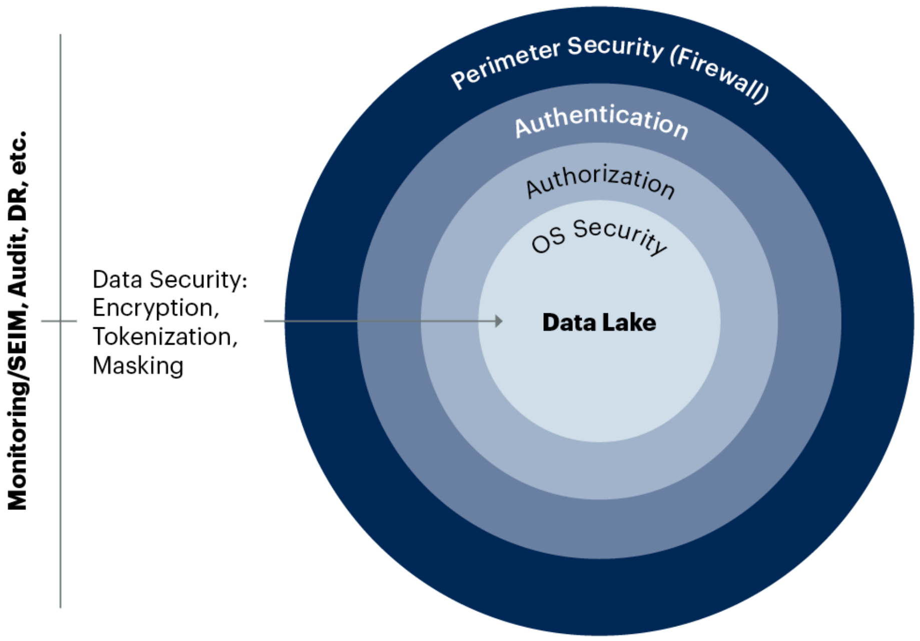
This research note is restricted to the personal use of .

- **Authentication:** Determining the legitimacy of the identity of a user
- **Authorization:** The privileges that a user holds to perform specific actions
- **Access:** The security mechanisms used to protect data, both in transit and at rest

Figure 3 shows the different security layers in a data lake.

Figure 3: Security Layers in a Data Lake

Security Layers in a Data Lake


Monitoring/SEIM, Audit, DR, etc.
Data Security: Encryption, Tokenization, Masking
Perimeter Security (Firewall)
Authentication
Authorization
OS Security
Data Lake

Source: Gartner
733127_C

The ingestion layer poses a high security risk because:

- Incoming raw data has not been evaluated for usefulness, and in its raw state is devoid of any classification about its eventual usage, authorizations and privacy.
- This raw data may contain personally identifiable information, and since it is untouched and no data masking has been applied, there is a higher propensity of exposing sensitive data.
- This raw data with sensitive information can be combined, linked and enriched with data in the data lake, resulting in security breaches.

This research note is restricted to the personal use of .

Questions to Ask

Organizations, data architects and security engineers should ask these questions before building the data lake:

- Is the incoming data sensitive? Does it contain personally identifiable information (PII), protected health information (PHI) or Payment Card Industry (PCI) information?
- Who is entitled to read and/or write the data? Is the user allowed to see all the data? Are they limited to certain rows, or even certain parts of certain rows?
- Does this data need anonymization, sanitization or encryption?
- What kind of encryption is required? Is the same encryption to be used across all layers in the lake?
- Where does the data need to be protected/encrypted: at rest, in motion and/or in use?
- How and when can the data be disposed of?
- How do you dispose of the keys associated with encryption?
- Is the data safe when the user runs analytics?
- Is the data safe once the user has completed the data access? For example, you might believe there is no point in ensuring the data is safe across all layers of the data lake, only to write plain text results to an unsecure area.
- Can conclusions be made from aggregated data?

Best Practices

Organizations, data architects and security engineers should follow these best practices when building the data lake:

- Design security from the beginning.
- Apply the best security practices you would apply to legacy database implementations.
- Be aware of data types going into your data lake — sources, dependencies and levels of sensitivity.

This research note is restricted to the personal use of .

- To mitigate the security risks in the ingestion layer, perform raw data analysis in a quarantined landing zone where only a small number of authorized users are provided access to all the data.
- Ideally, encryption should be done to:
  - Data at rest
  - Data in transit (shuffle/internode data transfers).
- Log and monitor data access across all layers in the data lake.
- Automate the infrastructure provisioning, and make it configuration-driven. Have security settings incorporated within the automation process rather than rely on manual intervention processes.
- Have separate security privileges across all personnel with access to the data lake, including data engineers, data scientists, QA engineers, DevOps and DataOps teams, and data and business analysts.
- Select the right type of encryption based on performance, data size and application transparency.
- Ensure that you encrypt ephemeral data storage across the cluster.
- For data on object storage, turn on object-level auditing.
- The key to successful data security is to implement the metadata-driven governance approach across all layers of the data lake. Use this approach to classify or tag data based on different security and protection requirements of the multiple types of data deposited in the data lake.
- Provision Apache Hadoop and Apache Spark deployments on identity and access management (IAM) concepts. For structured data, use role-based access control (RBAC), roles and the data policy based on organizational policies. For unstructured data (text, audio and video), use coarse-grained platform security or add third-party layers that govern access.
- Limit the entry points into your cluster to specified edge nodes and authorized middleware.

This research note is restricted to the personal use of .

- Ensure correct disposal of data. This is the most overlooked area of data security. Use policies with life cycle management tools to identify if data is simply out of date and no longer has value. Also, if data is encrypted and is no longer required, simply throw away the keys.
- Ensure continuous security (CS) — based on the same principles of continuous integration/continuous delivery (CI/CD).
- Because, setting up a cloud data lake with enterprise-level security, compliance and governance controls takes significant time, effort coordination, configuration, customization and integration, ensure resources can be accessed by users with the right credentials. Have a dedicated team to ensure ongoing security, compliance, monitoring, controlling sensitive data and user access of the data lake.
- Decouple data access policies and governance; use storage, processing and analytics engines to enable consistent enforcement of policies.
- Always use SSL/TLS protocols to exchange data across different locations.
- Avoid making assumptions that the frameworks are secure out-of-the-box by default.

Challenges

Ensuring end-to-end security in a data lake can be extremely challenging. Some of the challenges include:

- Any system is as secure as its weakest link. An end to end data driven organization relies on multiple tools, products, frameworks and libraries. Ensuring that all the pieces involved are at the same security compliance level is extremely challenging.
- Frameworks, tools and libraries in a data engineering pipeline are constantly being updated and innovated. Ensuring that the security compliance of the newly released versions matches or exceeds the older versions can be very challenging and difficult to ascertain.
- Lack of security tools, products that integrate seamlessly across the different tools in an data engineering pipeline can often cause loopholes that can be difficult to detect and fix.

Data Life Cycle Management

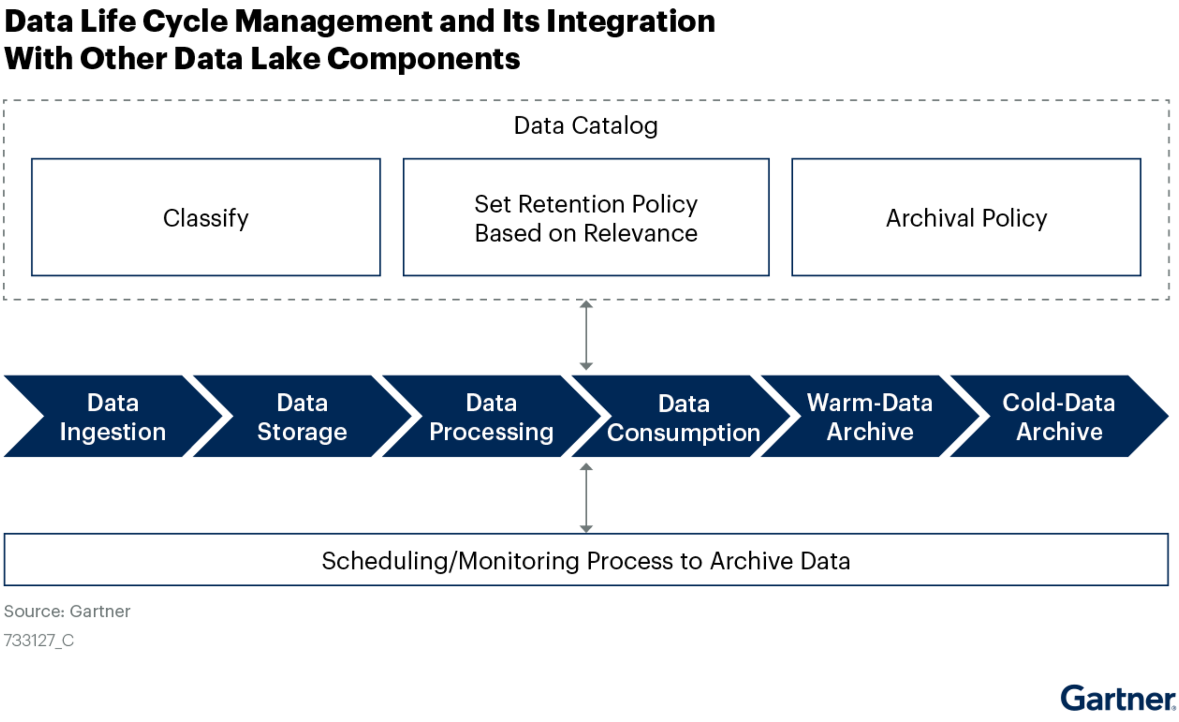
This research note is restricted to the personal use of .

Data life cycle management (DLM) is a subprocess of data governance. The DLM layer strives to achieve strategy and policies for classifying which data is valuable and how long you should store a particular dataset in the data lake. The life span of data can be partitioned into multiple phases, categorized by different patterns of usage. During different phases, the data can be stored differently. DLM helps to identify the true value of data over its lifetime and classifies it so data is stored, migrated or deleted according to its value. DLM provides a structured capability to classify data based on relevance and how the data evolves or grows over time.

Over periods of time, the value of data tends to decrease and the risks associated with storage increase.

Hence, it does not make sense to keep the data in a data lake continuously without plans and policies in place to delete data that is past its "use by" date. Figure 4 shows how the DLM strategy and process should be integrated with the data flow, data catalog and orchestration tools.

Figure 4: Data Life Cycle Management and Its Integration With Other Data Lake Components


Data Life Cycle Management and Its Integration With Other Data Lake Components
Data Catalog
Classify
Set Retention Policy Based on Relevance
Archival Policy
Data Ingestion
Data Storage
Data Processing
Data Consumption
Warm-Data Archive
Cold-Data Archive
Scheduling/Monitoring Process to Archive Data
Gartner

Source: Gartner

733127_C

Questions to Ask

This research note is restricted to the personal use of .

- What changes happen to the data over a period of time? Will its value diminish? Is it OK to hold it in the data lake?
- What is the extent of the data protection and data availability needed for this data?
- What are the applicable legal policies?

Best Practices

- DLM should be defined across each layer of the data lake:
  - Ingestion Tier
  - Storage Tier
  - Consumption Tier

Because the Consumption Tier deals with the distribution of data from the data lake to internal and external data customers, every transaction that distributes data is tracked, logged and monitored so that it adheres to the information life cycle management (ILM) policies of the organization.

- Data disposal — Secured data should have an agreed life cycle, set by a data authority or data steward who is knowledgeable both with the data and with the business and commercial context. A dataset labeled as “sensitive” may require encryption for the first year and “no encryption” thereafter before finally being disposed of. DLM ensures that everyone knows exactly how the data is to be treated. The life span of the data is based on usage frequency, classification and relevance. For each classification of data, the age parameter may differ by the data’s relevance or governmental norms.
- Data prioritization — Selection and classification of data should be the first exercise of DLM strategy.
- Ensure rules governing what can or cannot be stored in the data lake.
- Define your archival policy, and use automated rules based on data life cycle management, integrated with the data catalog. The frequency of access is one of the methods to determine whether or not the data is relevant to the business.
- It is not a good approach to put a hard timeline for all objects in the data lake. That could have an adverse impact on the business relevance of data.

This research note is restricted to the personal use of .

- Monitor the data usage over time (continuously) to avoid creating data.
- Integrate DLM with the data catalog to avoid any inconsistencies and data loss.
- Look to leverage vendors like Solix for your end to end data lifecycle management for data lakes.

Challenges

- Velocity and variety of data, as well as its source and veracity, adds to complexity in defining and understanding the governance and life cycle policies on data in a data lake.
- Unstructured data has challenges in terms of the eventual usability from an analytical insights perspective. Most of this data is not bound by legal, regulatory and privacy norms. These are typically used in conjunction with the organization's own internal customer, financial and transaction data to produce insights, after data enrichment.
- Sheer volume of data in a data lake often poses challenges in enforcing DLM policies because volume provides a huge surface area for policy breaches and risks.

Evolution of Data Lakes and Their Future

Two trends are driving innovation and the future of data lakes:

1. Data Lake House

2. Data Fabric

These trends are driven by some of the challenges of current data platforms and systems, which include:

- Data warehouses, data discovery tools and data lakes separate semantics from the data.
- The idea of movement and consolidation of data in and of itself will not solve critical business needs. The missing piece is the lens through which to view and better understand that data.

Data Lake House

This research note is restricted to the personal use of .

Lake house is a new paradigm that combines the best elements of data lakes and data warehouses. This approach removes the requirement to have to load the data onto any of the data warehouses to process and get the analysis or BI done. You can directly query the data underlying in your data lakes made of object storages or Hadoop. This method decreases the operational overhead on data pipelining and maintenance.

Most organizations build a data lake to store all their data, and a data warehouse to support fast business queries. Merging the two is challenging because their operating ETL characteristics are completely different. A data lake house aims to combine these characteristics into a single platform.

Though the tools to enable data lake houses are in their infancy, there is an increasing effort by data management vendors to have some of the capabilities of a data lake house. These vendors include Databricks Platform, Microsoft's Azure Synapse Analytics (which integrates Azure Data Lake Storage and Azure SQL Data Warehouse into a seamless environment, enabling a lake house pattern) and other managed services such as Google BigQuery and Amazon Redshift Spectrum.

Some of the benefits that data lake houses are trying to address include:

- Elimination of ETL jobs
- Reduction of data redundancy
- Ease of data governance and schema management
- Enablement of transaction support

Data Fabric

Though the whole idea of data lakes was to consolidate and centralize all the data in an organization, it increasingly looks like the future of data within an organization is **decentralized.** This is very different from what we see in almost any company currently. Data fabric is a new way to manage and integrate data. Data fabric tries to overcome some of the limitations of data warehouses and data lakes by leveraging semantic standards to describe data and enable better integration. It does this by providing context and mapping the data to the language of business, thereby making it less opaque.

This research note is restricted to the personal use of .

Data fabric is a platform that supports access and processing across all the data in an organization. It is built using an enterprise knowledge graph to create a unified data environment.

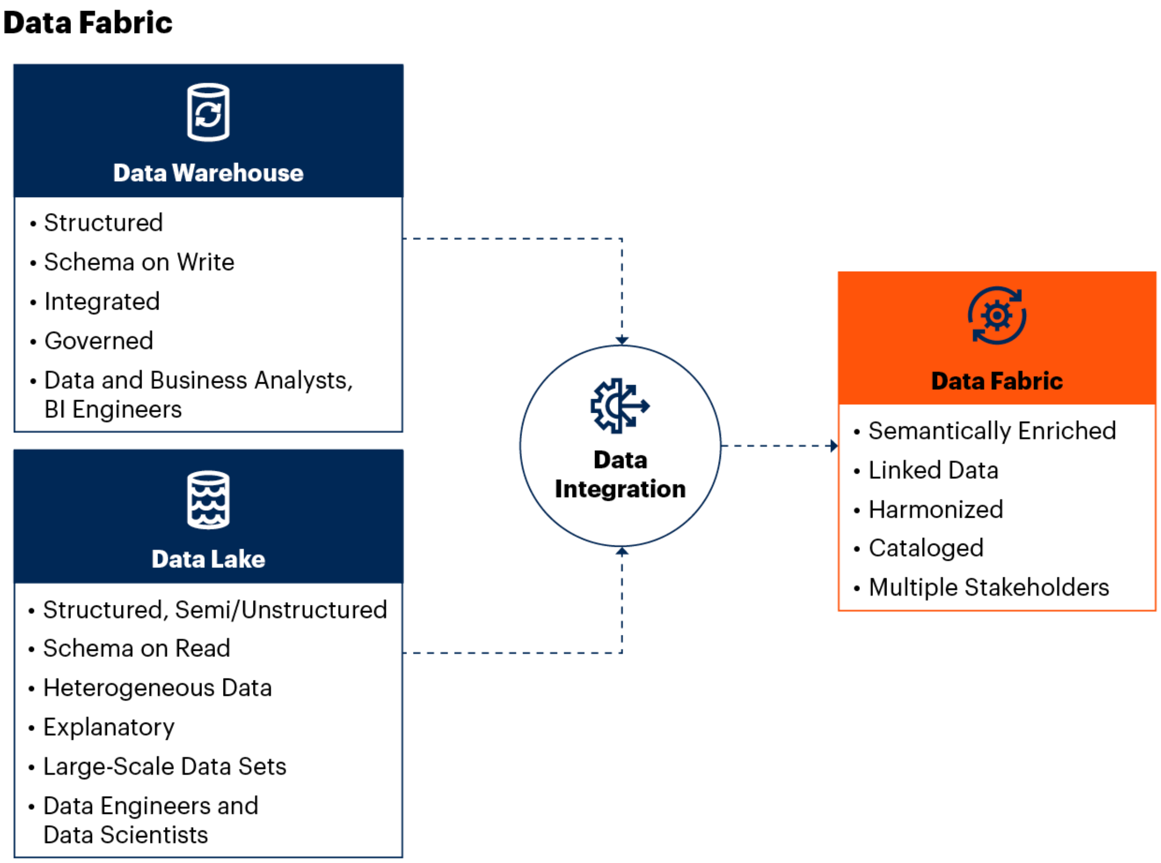
A data fabric allows overlaying data applications access across data sources where data lakes and data warehouses become nodes in a network. Data fabric combines the power of graph technology to capture and represent data complexity with rich semantics that allow end users to understand what the data means as it is ingested, stored and delivered where and when needed. Data graphs enhanced with semantics have the power to capture and represent the complexity and heterogeneity of data available within an organization.

Data fabric aims to future-proof data architecture by providing a semantic buffer over data acquisition, storage, distribution and underlying data sources. If data changes at sources, data fabric can mitigate the changes without breaking applications or requiring downtime. Data fabric can also elegantly query data across disparate data silos without data movement.

Figure 5 shows how the concept and capabilities of data fabric build on the concepts and capabilities of data warehouses and data lakes.

This research note is restricted to the personal use of .

Figure 5: Data Fabric Components



Source: Gartner
733127_C

None of these are ready for prime time but could become the paradigm of this decade. Anzo data lake from Cambridge Semantics has been building data lakes based on a knowledge graph platform.

Strengths

Some of the strengths of data lakes around data consumption, governance and operationalization include:

- Data lakes are based on distributed scale-out architecture for compute, storage and I/O. This facilitates fault tolerance, HA and replication out of the box.
- Data lakes allow organizations to better-utilize their resource consumption and utilization across different users and workloads.

This research note is restricted to the personal use of .

- Tools for data governance, operationalization and data cataloging are rapidly being enhanced and innovated to ensure organizations are able to leverage their data lake capabilities successfully.

Weaknesses

Some of the weakness of data lakes around data consumption, governance and operationalization include:

- If data lakes are built without the right governance, as had been the case a few years back when organizations were building data lakes without any governance, they easily create data swamps. Although a large volume of data is available to users in the data lake, problems can arise when this data is not carefully managed, including:
  - Lack of data governance: Without the structure and controls to manage and maintain the quality, consistency and compliance of data, a data lake can rapidly devolve into a data swamp.
  - Poor findability: Although the data might be available, its value is limited because users are unable to find or understand the data.
- Data lakes lack a logical data model, hence each application needs to transform and process the data — duplicating the data cleansing and modeling efforts in many contexts.
- Operationalization of data lakes is extremely challenging, complex and often error-prone. Promoting data pipelines from development into full enterprise production can be fraught with perils, including wrong packaging, wrong configuration and wrong infrastructure. However, emerging tools in the DataOps space are helping to remediate this situation (see Operationalizing Big Data Workloads).

Guidance

To be successful with data lakes, technical professionals need to clearly define, understand and outline their requirements across the dimensions discussed in this section.

This research note is restricted to the personal use of .

Data and Governance

Understanding the data to be ingested in a data lake, irrespective of whether it is external or internal data, is extremely imperative. Determine the following before building a data lake:

- Clearly identify the nature of the data in terms of volume, velocity, structure, source type and location.
- Define the data quality requirements, lineage, frequency of updates, and consistency and completeness.
- Understand the security constraints of the data in terms of its usage and regulations.
- Identify data owners, data encryption requirements, access controls, and authorization and authentication policies.
- Document the policies for data life cycle management.

Operations

The last mile of a data lake implementation is the operationalization. Some best practices for operationalizing a data lake are outlined below.

- Define and develop the data lake hosting strategy, whether it is in the cloud, on-premises, hybrid or multicloud. Plan and develop an end-to-end automation strategy across the different layers of the data lake, with self-service capabilities, and determine how you will leverage containerization and container-orchestration-based approaches.
- Invest in the right tools for building workflows while orchestrating and scheduling the end-to-end data flow processes. Invest in tools for monitoring, management, alerting and notifications.
- Operationalize end-to-end governance and continually collect and use operational metrics.
- Define and implement the nonfunctional requirements — such as disaster recovery (DR), HA, fault tolerance and backup strategies.

This research note is restricted to the personal use of .

The Details

This section takes a deeper dive into each of the components of a data lake and discusses the architectural underpinnings of these components.

Data Governance

Data governance comprises data quality, data catalog, data security and privacy, data lineage tracking, and data life cycle management components. These components cut across all the layers of the data lake. This section explores the data quality, data catalog and data security aspects of data governance from an architectural lens.

Data Quality

For data democratization to be successful, organizations must ensure the data lake is filled with high-quality, well-governed data that is easy to find, easy to understand, and easy to determine for quality and fitness. Some of the characteristics for ascertaining data quality are:

- Correctness/accuracy
- Completeness/coverage
- Consistency
- Timeliness
- Data lineage

Some common data quality issues that routinely arise in a data lake include:

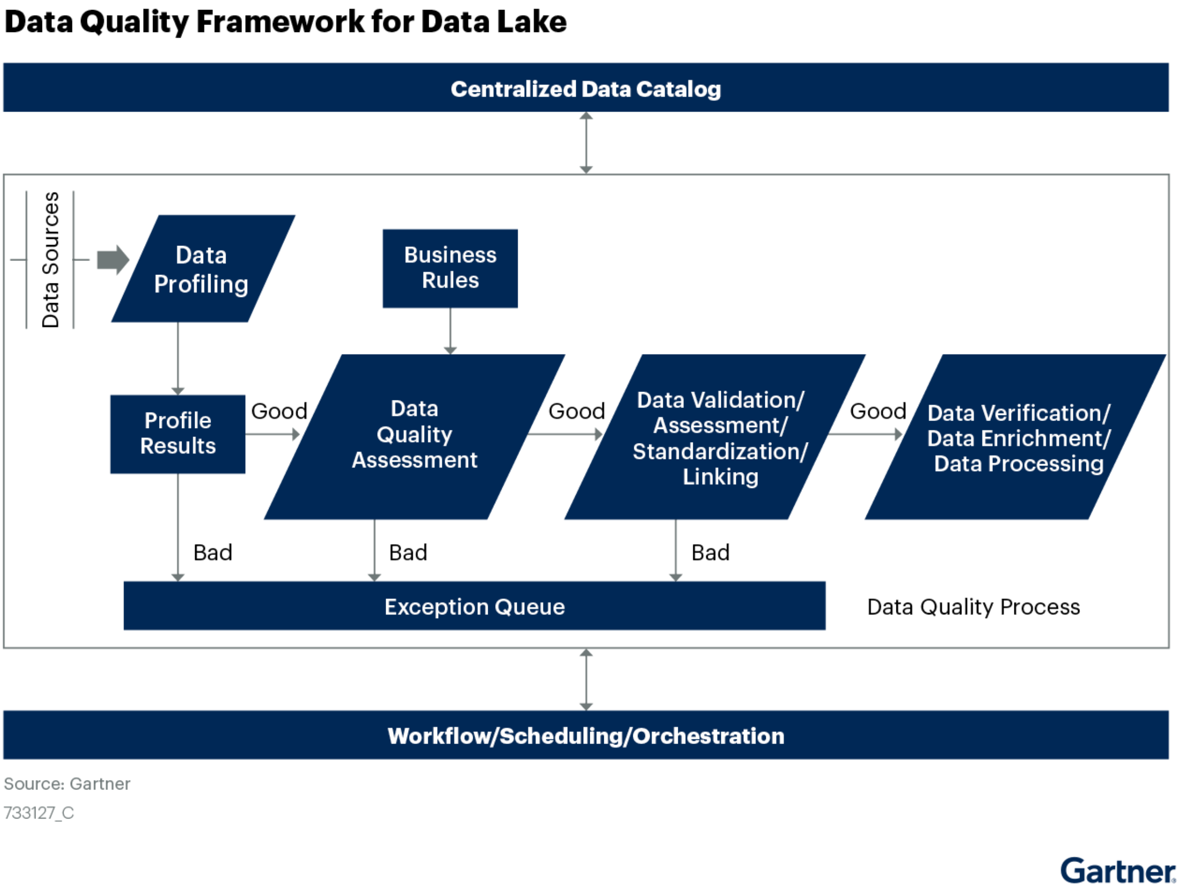
- Embedded delimiters, where a value contains the delimiter that separates fields (e.g., commas)
- Corrupted records where an operational system may have inadvertently put control characters in values
- Data type mismatches, such as alphabetic characters in a numeric field
- Nonstandard representations of numbers and dates
- Multiple record types in a single file

This research note is restricted to the personal use of .

- Mainframe file formats that use different character sets on legacy systems, which Hadoop does not know how to recognize or process

A high-level architecture of how a data quality system should be architected is shown in Figure 6.

Figure 6: Data Quality Framework for Data Lake


Data Quality Framework for Data Lake
Centralized Data Catalog
Data Sources
Data Profiling
Business Rules
Profile Results
Good
Data Quality Assessment
Good
Data Validation/ Assessment/ Standardization/ Linking
Good
Data Verification/ Data Enrichment/ Data Processing
Bad
Bad
Bad
Exception Queue
Data Quality Process
Workflow/Scheduling/Orchestration
Gartner

Source: Gartner
733127_C

For more details on data quality, see Enabling Data Quality for Machine Learning and Artificial Intelligence.

Data Catalog

For the data in the data lake to be used effectively with the downstream applications, the consumers need to understand the data and its structure, content and context. Various techniques can be applied on the data in the Raw Zone by the data catalog tools to identify semantic metadata and to enable efficient data discovery.

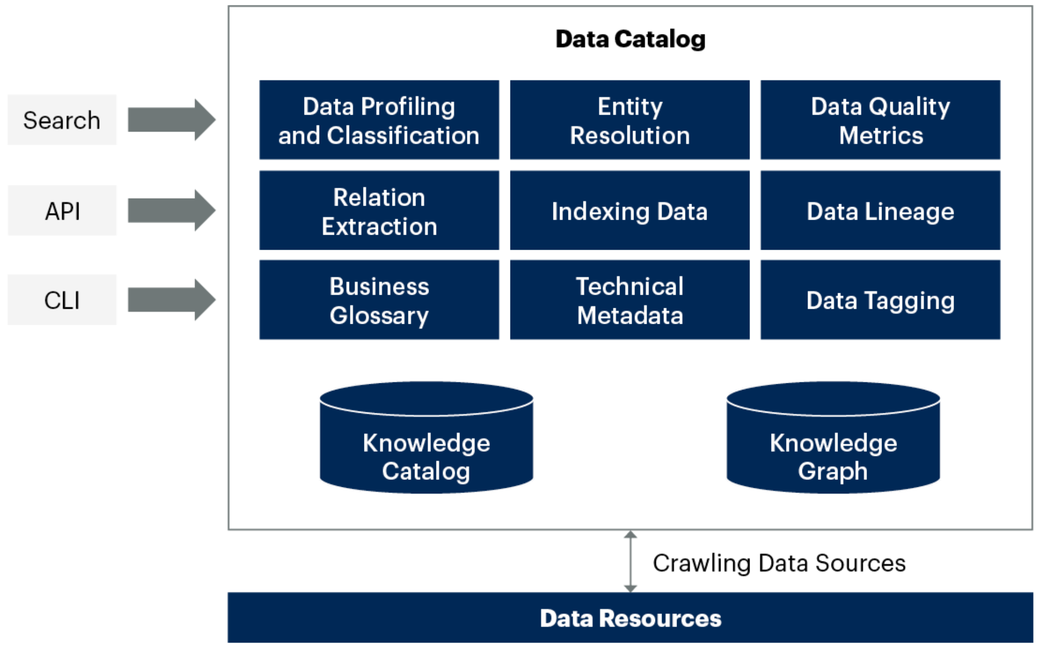
This research note is restricted to the personal use of .

Data catalog tools should be able to extract entities from multistructured and unstructured data to build semantic metadata and to be able to identify relationships between entities. Additionally, data catalogs should also index the data to optimize and speed up data exploration and data discovery. Data catalog tools should leverage semantic technologies to identify relationships and standardize methods expressing relationships to improve visibility of the data and to reduce time to discover, integrate and analyze the data. Data catalog tools should identify latent topics from documents and then classify and annotate them based on the topics.

After data ingestion, data catalog tools crawl the datasets in the Transient Zone and Raw Zone to build the metadata. Enterprisewide data catalog tools that build a holistic enterprise catalog also crawl data sources outside a data lake and can also crawl to edge devices to gather technical and operational metadata. A high-level architecture of a data catalog is shown in Figure 7.

Figure 7: Data Catalog Architecture for Data Lakes

Data Catalog Architecture for Data Lakes



Source: Gartner
733127_C

Some tools and vendors in the data catalog:

- Alation

This research note is restricted to the personal use of .

- Alex Solutions
- Boomi (formerly Unifi)
- Cloudera Hortonworks Data Steward Studio
- Cloudera Navigator
- Collibra
- Hitachi Vantara (formerly Waterline Data)
- IBM Watson Knowledge Catalog
- Infogix
- Informatica Enterprise Data Catalog
- Oracle Enterprise Metadata Management
- Reltio

Data Security

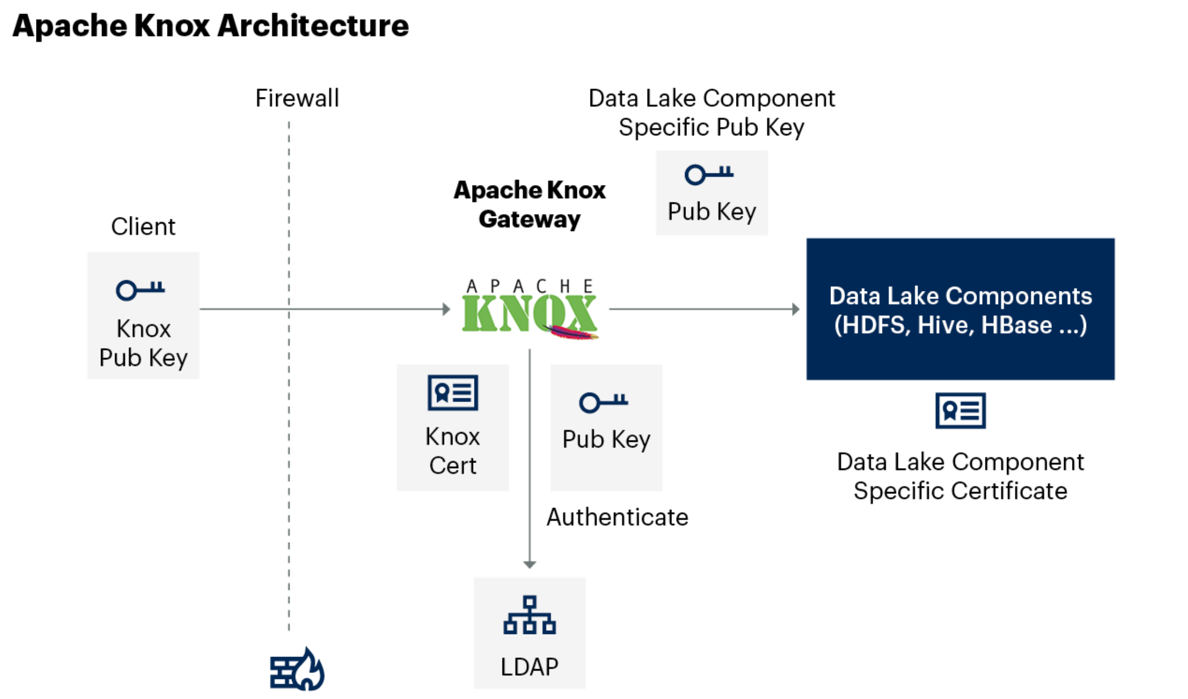
This section details some of the commonly used products and provides an overview of their architecture to implement security across the different layers of the data lake.

Apache Knox is a reverse proxy that provides perimeter security to Hadoop clusters. It supports different policy enforcement like authentication, authorization and host mapping by chaining these as specified in a topology deployment descriptor.

Figure 8 shows how Apache Knox fits into the overall architecture of a data lake.

This research note is restricted to the personal use of .

Figure 8: Apache Knox Architecture


Apache Knox Architecture
Firewall
Data Lake Component Specific Pub Key
Pub Key
Apache Knox Gateway
Client
Knox Pub Key
APACHE KNOX
Data Lake Components (HDFS, Hive, HBase ...)
Knox Cert
Pub Key
Data Lake Component Specific Certificate
Authenticate
LDAP

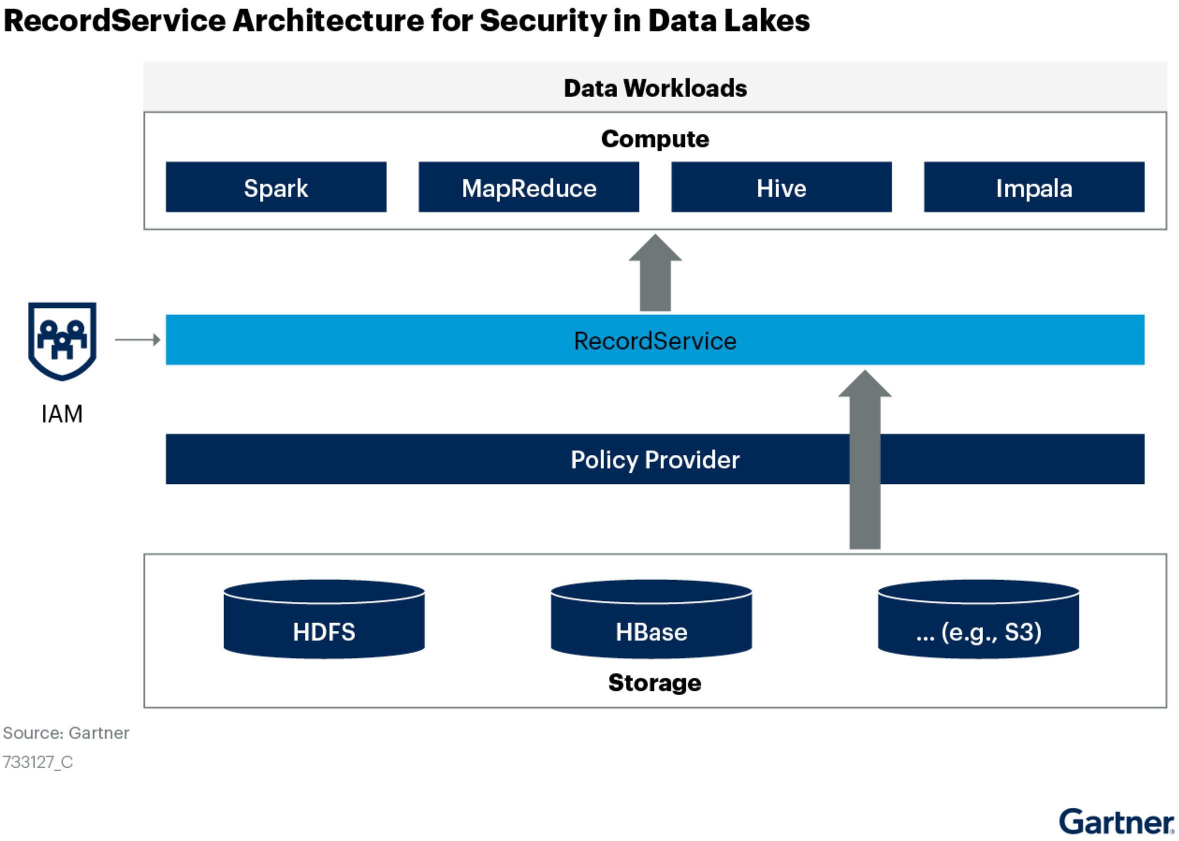
Source: Gartner
733127_C

As the data lake ecosystem grows, different products have different security implementations. Duplicate policies are often needed to provide the same user with seamless access to different tools in the ecosystem. Cloudera's RecordService is a new security layer that sits between the storage managers and compute frameworks to provide a unified data access path, fine-grained data permissions and uniform enforcement across the stack to minimize duplication of policies.

Figure 9 shows how RecordService fits into the architecture.

This research note is restricted to the personal use of .

Figure 9: RecordService Architecture for Security in Data Lakes



Source: Gartner
733127_C

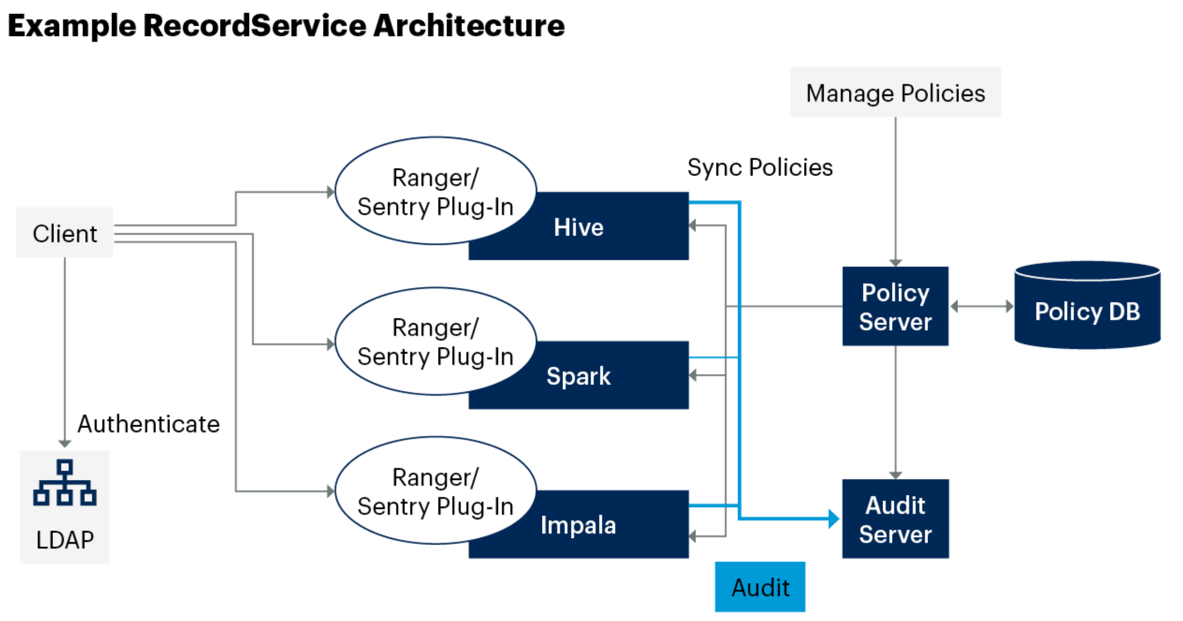
Apache Ranger’s concept of a RecordService is a framework to enable, monitor and manage security across a Hadoop ecosystem. It provides authorization for a range of technologies in the ecosystem. It is based on attribute-based access control (ABAC). The Ranger plug-in is installed with the product — for that, authorization needs to be enforced. It synchronizes user data with the enterprise directory (where user credentials are stored) and uses that to set up appropriate security policies.

When a user tries to access data for products for which the Ranger plug-in is installed, it retrieves the policies stored and does appropriate checks before users gain access to the data that they require.

Another example of a record service is Apache Sentry. Sentry is a system for enforcing fine-grained, role-based authorization to data and metadata. Sentry’s plug-in is installed on any of the data processing technologies. Access to data is intercepted by the plug-in, and if it meets all the criteria defined in the policies, metadata access is allowed. Sentry’s server manages authorization metadata. Figure 10 shows a high-level architecture of how Apache Ranger or Sentry integrates with the different components in a data lake.

This research note is restricted to the personal use of .

Figure 10: Example RecordService Architecture



Source: Gartner
733127_C

Most components in the Hadoop ecosystem now include a built-in key management service (KMS) to secure the transport protocol over HTTP. It provides both client and server REST APIs for securing the communication channel. It is a Java application that includes support for the Java key store to hold multiple keys and an API to access and manage key metadata. It also includes an access control list (ACL)-based support for multiple authentication and authorization protocols like Kerberos, Microsoft Azure Active Directory and Lightweight Directory Access Protocol (LDAP) with Secure Sockets Layer (SSL). The Hadoop Key Management Server (KMS) includes end-to-end encryptions covering data at rest and in motion. Data written into Hadoop Distributed File System (HDFS) is immediately encrypted using a specific algorithm and assigned a security zone.

Data lakes contain a large number of files, and setting permissions for each manually is not possible. Tag-based security in products like Cloudera Navigator and Ranger support tag-based policies. Instead of defining ACLs at a file level, these tools allow you to set up policies using tags by simply tagging files and folders instead of manually creating ACLs. Catalog tools allow you to set these tags, and they can get automatically applied by the policy-based access control tools.

This research note is restricted to the personal use of .

This approach provides a powerful way to manage and organize the data without trying to shoehorn the organizational, hierarchy-based access rules into the file management structure. Tagging can be applied to data at the ingest layer and policies can be applied to restrict access to files.

Automation is the ideal solution for handling sensitive data and the access control management around it. Tools from vendors like Informatica and Waterline Data automatically scan ingested files and detect sensitive data with advanced ML algorithms and tag them. These tags are then used by the data catalog tools to enforce tag-based policies.

Vendors in the security space for data lake include Apache Metron, BigID, Dataguise, Delphix (Data Masking), Immuta, Microsoft (BlueTalon), Okera, Privitar, Protegrity and Thales (Vormetric).

For more details, see Securing the Data and Advanced Analytics Pipeline.

Recommended by the Author

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

Operationalizing Big Data Workloads

Building a Comprehensive Data Governance Program

Selecting SQL Engines for Big Data Workloads

Enabling Data Quality for Machine Learning and Artificial Intelligence

Securing the Data and Advanced Analytics Pipeline

Data Modeling to Support End-to-End Data Architectures

This research note is restricted to the personal use of .

© 2023 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. It consists of the opinions of Gartner's research organization, which should not be construed as statements of fact. While the information contained in this publication has been obtained from sources believed to be reliable, Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although Gartner research may address legal and financial issues, Gartner does not provide legal or investment advice and its research should not be construed or used as such. Your access and use of this publication are governed by Gartner's Usage Policy. Gartner prides itself on its reputation for independence and objectivity. Its research is produced independently by its research organization without input or influence from any third party. For further information, see "Guiding Principles on Independence and Objectivity."

This research note is restricted to the personal use of .

Table 1: Key Capabilities of a Data Lake

Capability	Description
Data Quality Metrics and Continuous Measurement	■ Multistage data cleansing, refinement and enrichment ■ Automated data processing ■ Continuous data profiling
Automated and Collaborative Data Cataloging Classification for Governance and Data Consumption Catalog Service	■ Catalog and classify available data ■ Search, discover and subscribe to data ■ Enable data source registration, and automated data discovery and cataloging
Security	■ Access controls to a wide category of users in the enterprise ■ Allow data encryption, data redaction, data obfuscation and data masking
Operationalization and End-to-End Automation	■ Design and configure workflows/scheduling/orchestration/automated scaling/self-healing ■ Containerize resources and their deployment

This research note is restricted to the personal use of .

Data as a Service and Data Provisioning for Consumption	■ Publish data and analytical services for consumption ■ Provision refined, clean, trusted data for downstream consumption
Data Life Cycle Management	■ Process of controlling, versioning and managing storage of data in the organization as it ages and progresses through its business and technical life cycle

Source: Gartner (October 2020)

This research note is restricted to the personal use of .