Three Architecture Styles for a Useful Data Lake

Published 15 July 2016 - ID G00303817 - 40 min read

ARCHIVED This research is provided for historical perspective; portions may not reflect current conditions.

By Analysts Svetlana Sicular,

Supporting Key Initiative is Analytics and BI Strategies

A data lake extends processing and analytics capabilities to unrefined data in its native or near-native format. Implementers of big data solutions should select a proper data lake architecture to derive the most value from this data store.

Overview

Key Findings

- A data lake contains unrefined data when the data structure is unknown in advance, or when organizations want to increase analytics and operational agility by complementing their systems of record with systems of insight.
- Today, three complementary data lake architecture styles reflect three main data lake purposes: an inflow data lake to bridge information silos, an outflow data lake to get to the data faster, and a data science lab data lake to enable innovation in new ways.
- A successful data lake starts with a single data lake style, then adds characteristics of other styles as it matures.

- The new data store and new technologies do not change the fundamentals — the value of data is in deriving insights from it.

Recommendations

- Architect your data lake with a clear understanding of how your organization will derive value from the data you store in the lake. Establish a roadmap for implementing a data lake at the pace your business is capable of, to consume the outcomes from the new use cases.
- Apply data modeling and address metadata management capabilities immediately.
- Ensure proper information governance (better described as data advocacy) to prevent or minimize mistakes due to misinterpretation of data. Implement data policies and standards to avoid pollution in the lakes, change management for tracking versions of data and models, and data lineage to avoid disasters with data derivatives.
- Evolve different skills through training and practices that extend teams' current experience, and hire short-term mentors to hone those skills. Engage infrastructure specialists — cloud, storage, network and security experts — in the data lake project from its inception.

Comparison

Gartner's Definition of a Data Lake

> **A data lake is a collection of storage instances of various data assets. These assets are stored in a near-exact, or even exact, copy of the source format and are in addition to the originating data stores.**

The purpose of a data lake is to present an unrefined view of data. The data lake is a concept, not a technology, and it can rely on any scalable tool that can store data in its exact or near-exact format. It is up to the users of the lake to interpret the data and to determine the best data applicability for the identified use cases. For example, the same log files can be interpreted for cybersecurity in one way and for process automation in another way. The data in the lakes is highly contextual, and its interpretation requires sound metadata management and information governance.

Data Lake Architecture Styles

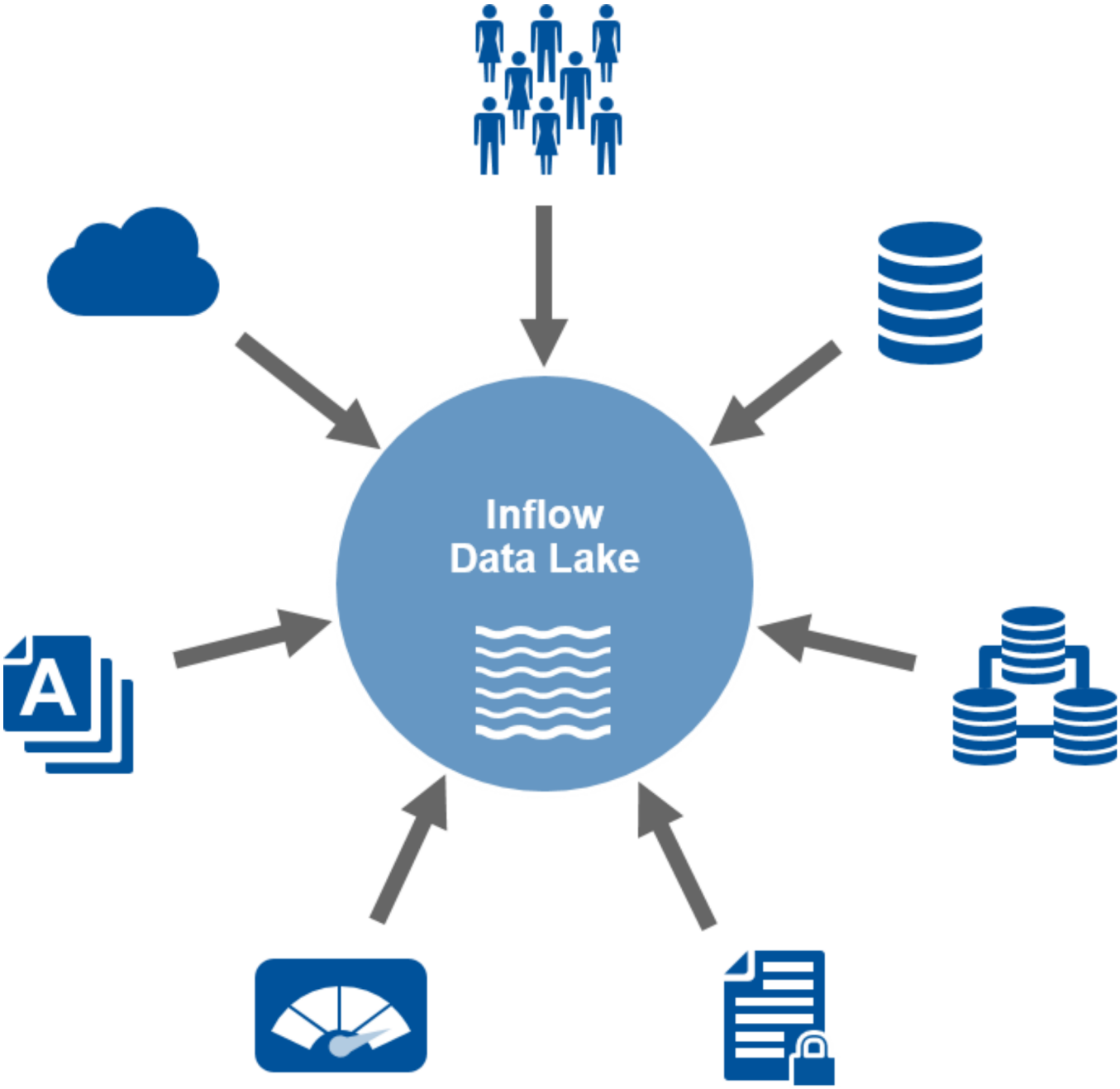
Today, three data lake architecture styles have crystallized: an inflow data lake, an outflow data

lake and a data science lab data lake. These styles are additive: Data lake implementations start with a single style. Later, when maturing, the data lake adds elements of other styles.

Architecture Style No. 1: An Inflow Data Lake

The inflow style of a data lake architecture is best for bridging information silos. It accommodates a collection of data ingested from many different sources that are disconnected outside the lake but can be used together by being colocated within a single place (see Figure 1).

Figure 1. The Inflow Data Lake Architecture Style

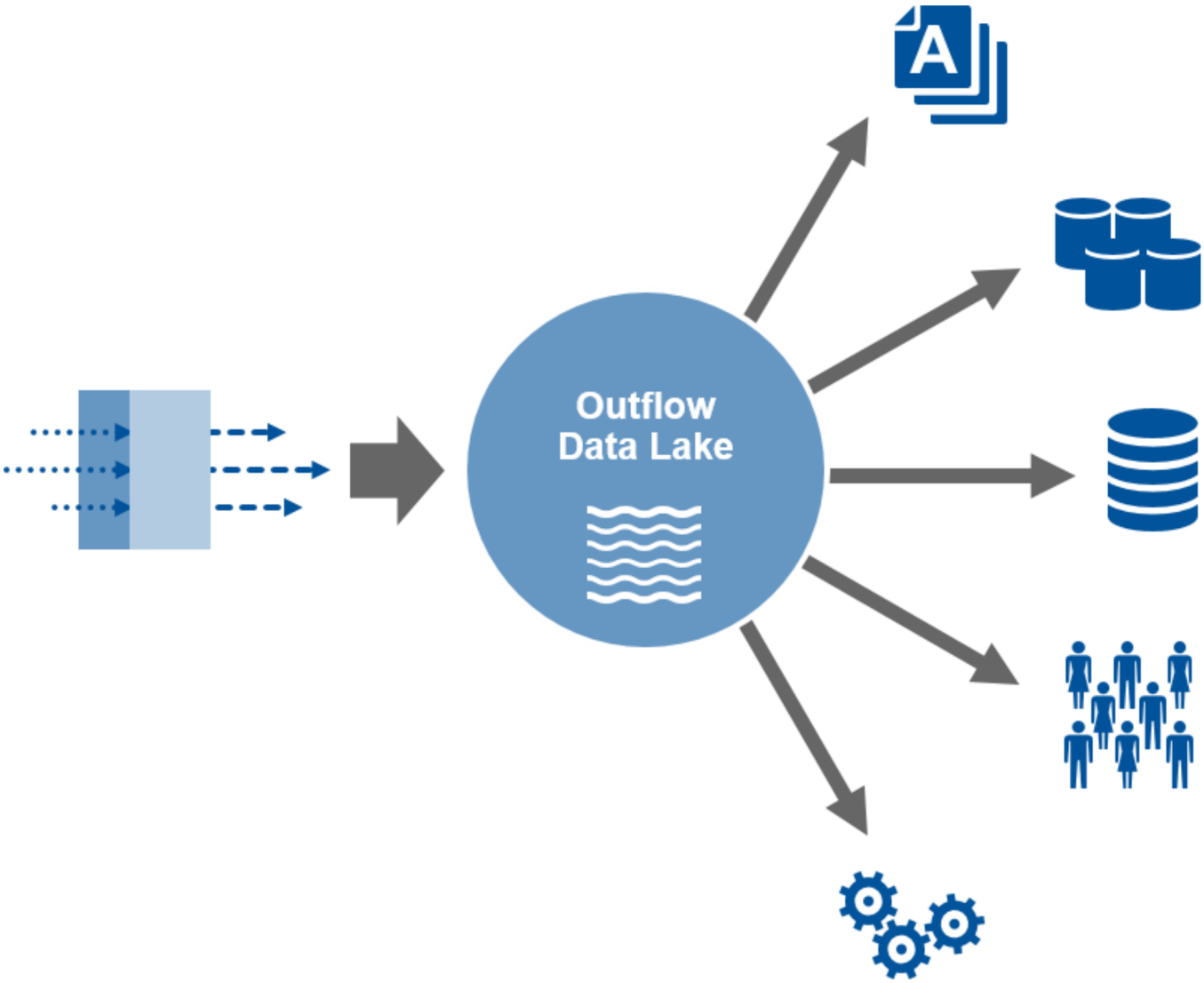


Source: Gartner (July 2016)

Architecture Style No. 2: An Outflow Data Lake

The outflow style of the data lake architecture is best for getting to the data faster. It is a landing area for freshly arrived data available for immediate access or via streaming. It employs schema-on-read for the downstream data interpretation and refinement (see Figure 2). The outflow data lake is usually not the final destination for the data, but it may keep raw data long term to preserve the context for downstream data stores and applications.

Figure 2. The Outflow Data Lake Architecture Style

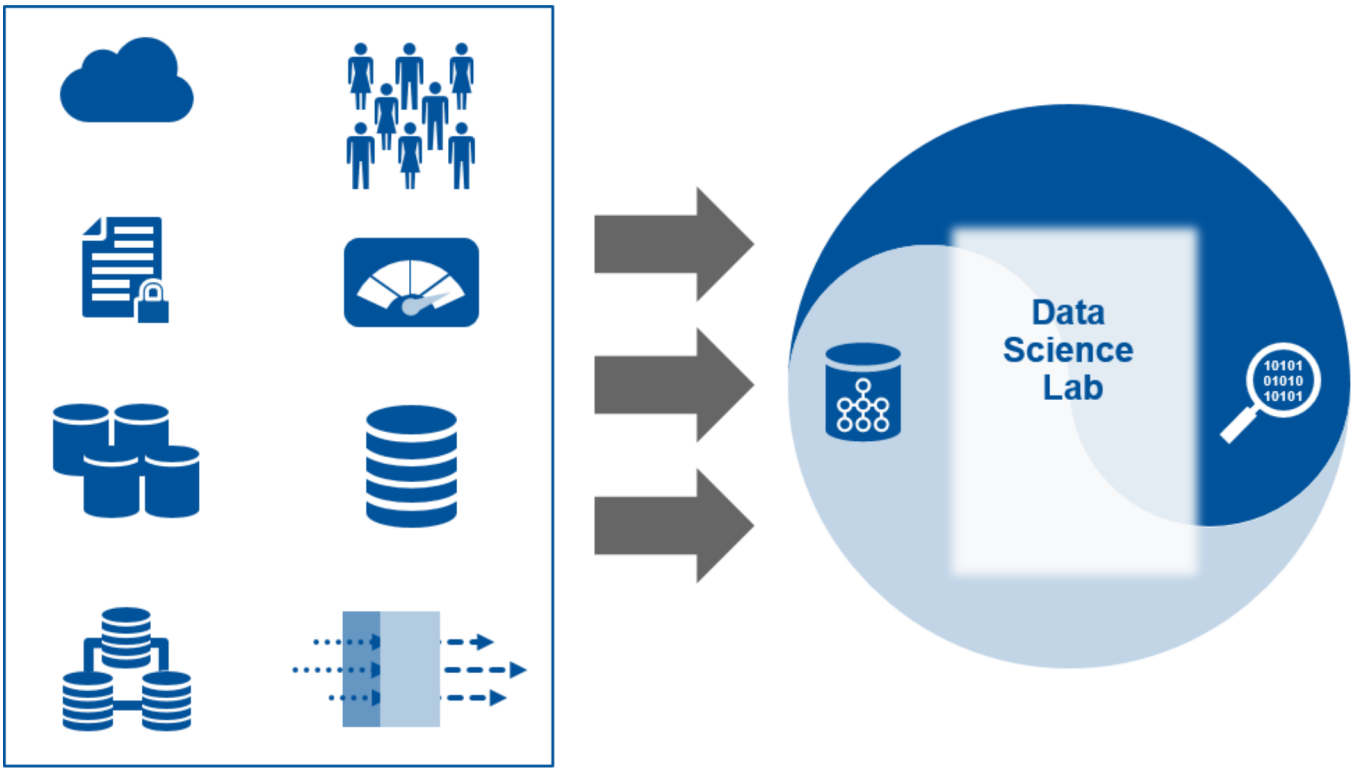


Source: Gartner (July 2016)

Architecture Style No. 3: A Data Science Lab Data Lake

The data science lab style of the data lake architecture is best for enabling innovation in new ways (see Figure 3). It is similar to the architecture styles No. 1 or No. 2 but is used for a more narrow purpose, such as cybersecurity, 360-degree customer view or a jet engine analysis. This type of a lake is most suitable for data discovery and for developing new advanced analytics models — to increase the organization's competitive advantage through new insights or to innovate in the public sector.

Figure 3. The Data Science Lab Data Lake Architecture Style

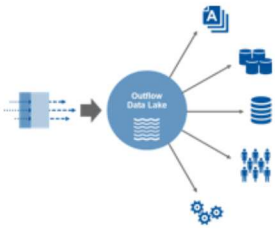


Source: Gartner (July 2016)

Comparison of the Data Lake Architecture Styles

Figure 4 lists only comparison criteria that are significant for selecting a data lake architecture style. The common considerations for all data lakes — underlying technologies, infrastructure options, data modeling, metadata management, data governance and relevant skills — are excluded from the comparison. The Analysis section discusses comparisons and common considerations to make technical professionals successful in implementing data lakes.

Figure 4. Comparing the Data Lakes Architecture Styles by Requirements



Requirements	1. Inward Lake	2. Outward Lake	3. Data Science Lab Lake
Latency			
High latency	●	◔	●
Low latency	◔	●	◔
Big Data Characteristics			
Volume	◕	◑	●
Velocity	◔	●	◔
Variety	●	◔	◕
Information Management			
Data modeling	●	◑	◔
Data governance	●	◑	◔
Fit into the existing EA	●	◑	◔
Scalability			
Optimized for data input	◕	◔	◑
Optimized for data output	◔	◕	◑
User concurrency	●	◔	◔
Consumption Patterns			
BI	●	◔	◑
Advanced analytics	◑	◑	●
Operations	◑	●	◔

Less Applicable ○ ◑ ◕ ● More Applicable

BI = business intelligence; EA = enterprise architecture

Source: Gartner (July 2016)

The criteria important for selecting the right data lake architecture style are divided into the following main groups:

- **Latency** requirements for a dominant delivery and processing of fresh data are key to successfully architecting the data lake. Low latency associated with real-time requirements is getting increasingly popular and is best accommodated by the outflow data lake architecture. The inflow data lake and the data science lab styles are usually used for developing new models, where most work, like machine learning, is done in batch. The first

step to determine if real time is of essence is defining latency expectations for data acquisition, processing and consumption. Potential bottlenecks in the end-to-end systems might make low latency expectations for accessing fresh data unrealistic.

- **Big data characteristics** of high-volume, velocity and variety define what big data aspects underpin data lake.
  - High volume is associated with files that cannot fit a single machine and therefore require a distributed storage.
  - High velocity turns data from static to dynamic. It is frequently associated with performing real-time analytics. Yet, velocity is also about monitoring the rate of change, connecting datasets that are coming at different speeds, and handling bursts of activities, rather than a steady tempo of events.
  - High variety assumes coexistence of differently structured datasets — for instance, images, and social, historical, geolocational and contextual data — contributing observational and transactional information for new analysis.

- **Information management** enables data and analytics professionals to understand, organize and find the right data for the right tasks. In other words, sound information management approaches are vital for preventing data lakes from turning into data swamps.
  - Refer to the Applying Data Modeling and Establishing Data Governance sections in this research for in-depth discussions of these critical factors for data lake success.
  - A data lake fit into the overall enterprise architecture will affect the choices of data, tools and infrastructure for the data lake.

- **Scalability** requirements could be confused for storage and processing scalability, for which all big data technologies are designed. In particular, storage scalability is the foundational characteristic of any data lake.
  - Data acquisition and consumption scalability have very different (and challenging) requirements. For example, Gartner observed a number of data lakes that were designed for data acquisition that should have been organized for data consumption.
  - Most data lakes have few users accessing them directly, but when opened up for wide access, user concurrency could be an obstacle.

- **Data consumption** methods are the way to derive value from the lake. Data lakes should be architected as a means to an end: A good understanding of how the lake will be used is necessary to select the appropriate data lake architecture style from the start.

- Business intelligence (BI) is the predominant method of the data consumption by business users. Expect this category of users to grow the most.
- Advanced analytics — exploratory analysis and building new analytical models — can be mastered by a small number of users, mostly data scientists.
- Data in the lakes can support operations, for example, in the Internet of Things (IoT) or in business processes, like loan approvals and call center interactions.

Analysis

A data lake has different, often conflicting, definitions and interpretations. In some interpretations, a data lake is just synonymous with "big data" and is used in lieu of that tired, overused term. In other interpretations, a data lake function is mistaken as a replacement for an enterprise data warehouse (EDW). However, both are principles of organizing data, not technologies unto themselves. This section clarifies the purposes of data lakes, separates myths from realities, and elucidates success factors to help technical professionals implement the best data lake architecture for their enterprise requirements.

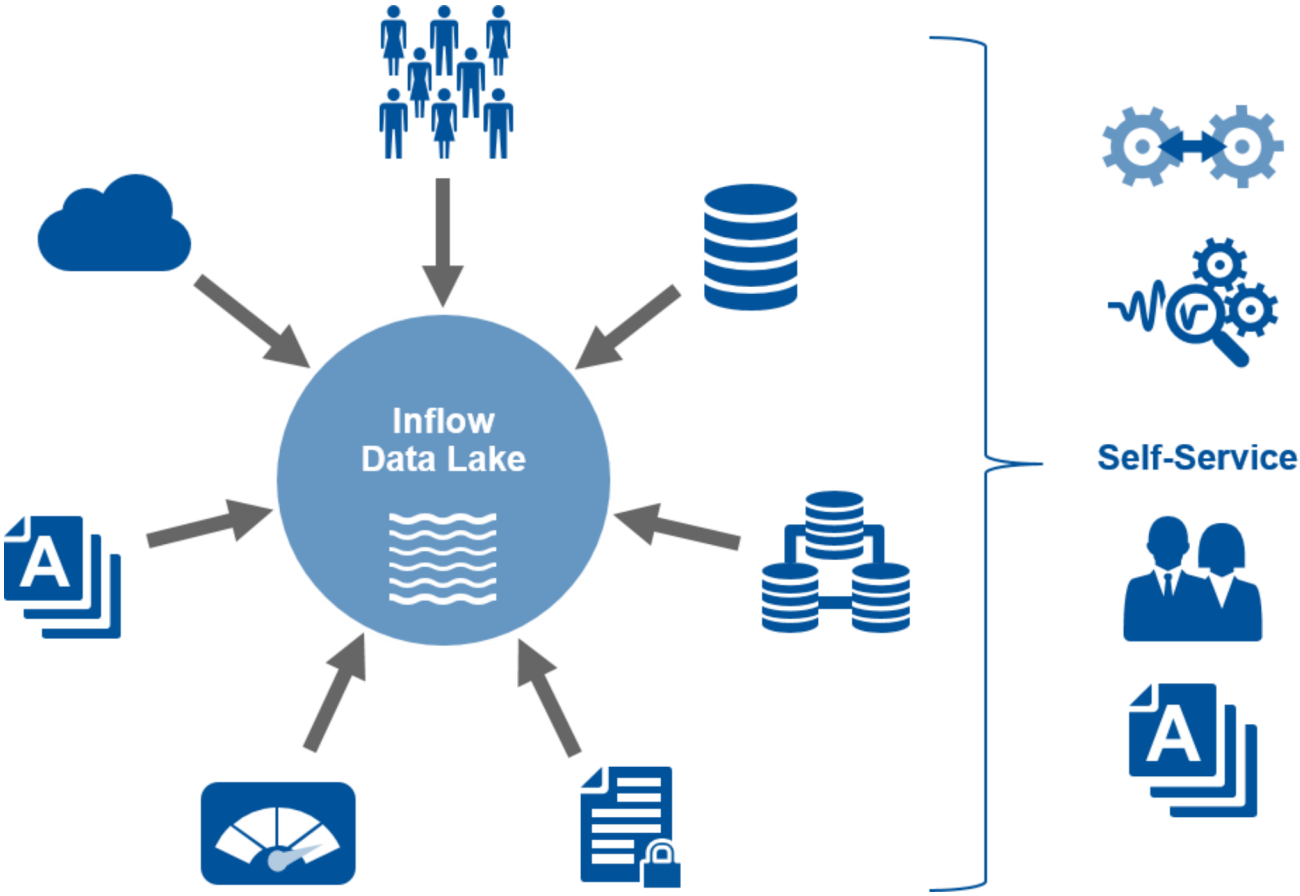
Differentiating Data Lake Architecture Styles

This section summarizes purposes, advantages and cautions specific to each data lake architecture style.

Architecture Style No. 1: An Inflow Data Lake

The inflow data lake resembles the sun, because it is the center of the new insights universe. The value of the inflow data lake is in bridging information silos to find new insights. Often, IT doesn't understand what the data means, and some companies do not allow IT to know what it means. IT should architect the data lake with the focus on self-service capabilities, so that the business can derive value from this data (see Figure 5).

Figure 5. Consumption From the Inflow Data Lake



Source: Gartner (July 2016)

The inflow style is often referred to as "a data hub." It is good for self-service BI. Advanced analytics users can find unprecedented levels of information availability for new business opportunities in the inflow data lake. For example, a manufacturer can benefit by storing all data about its products in a data lake, including such diverse content as design specifications, product production data, warehousing and shipment information, customer orders, returns data, warranty history and social data about product consumption. Being able to analyze all facets of this data may uncover some previously hidden patterns, which can lead to successful product enhancements and to new ways of selling manufactured goods.

The inflow architecture style is the closest to the concept of the EDW, but the difference is in the capability to store unrefined and extraneous data, which would be impractical to keep in the EDW from cost or processing perspectives. The Implement the Data Lake for Its New Capabilities section gives a detailed comparison between a data lake and a data warehouse.

The underlying tools to instantiate the data lake should be able to satisfy a requirement of user scalability, which is challenging for many big data technologies. This data lake requires strong data governance to ensure the appropriate data visibility, interpretation and trustworthiness. A shared data access layer is important for ease of use and less effort for integration (see "Comparing Three Self-Service Integration Architectures" (https://www.gartner.com/document/code/297311?ref=grbody&refval=3380017) for details).

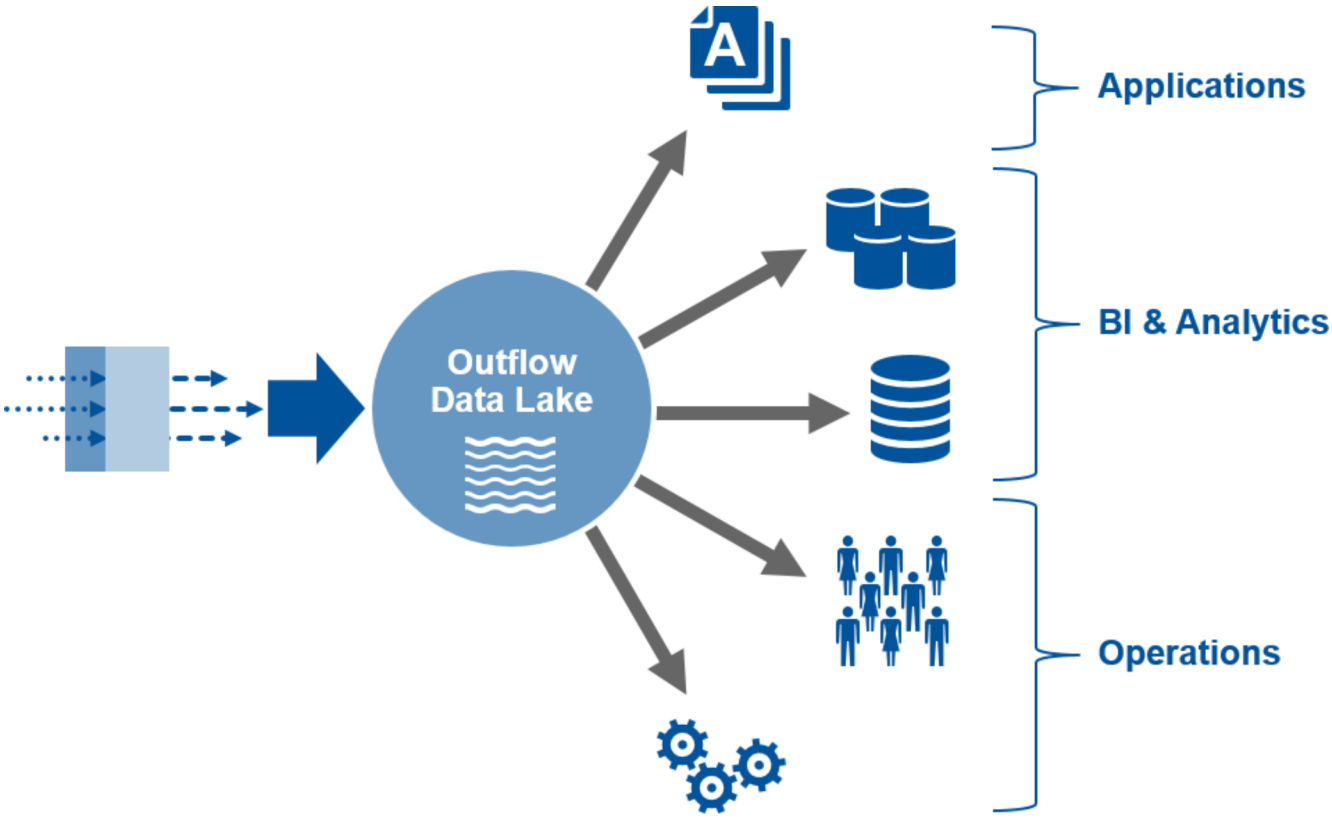
The dominant big data characteristic of the inflow data lake is variety of unstructured and semistructured data, as well as historical, seldom-used or highly granular data, which is extraneous in the EDW. Structured data sources in the EDW are inherently more understandable because they mostly come from operational systems, and those need to be accurate for finance, supply chain and CRM to work. But the data lake sources of different variety are inherently "messier": They add observations of the real world, which are not so easily categorized and organized.

To bridge information silos of a low-level data collation and no specific data commonality in the lake, a substantial data modeling effort is necessary. Pfizer Data Lake (http://bigdataeverywhere.com/files/sandiego-2016/BDE-Beyond the Data Lake-Pfizer-VKAPOOR.pdf) is the example of the inflow data lake style — complementing the EDW and implemented for self-service with metadata and bimodal governance.

Architecture Style No. 2: An Outflow Data Lake

The outflow style of the data lake architecture is best for getting to the data faster. Figure 6 resembles a fan, because it pumps the data at its consumers. To do this successfully, the outflow data lake style requires agile data governance that can accommodate the needs of specific consumers, as well as leave enough room for the localized interpretation of the schema on read.

Figure 6. Consumption From the Outflow Data Lake



Source: Gartner (July 2016)

The outflow lake is a revamped traditional operational data store (ODS) and/or a staging area, with more granular and more historical data. Organizations may find many uses for the high volume of data, both analytically and operationally, but they also don't want to duplicate the data preparation in every downstream consumer process. Commonality of data helps the different consumers to agree on what they are looking at. This is very similar to the need for an ODS to supply structured data, with low latency and commonality of meaning to multiple operational and analytical processes. "Upgrade the Enterprise Data Warehouse Architecture With Hadoop" (https://www.gartner.com/document/code/275430?ref=grbody&refval=3380017) discusses the architecture evolution of the EDW, ODS and data marts.

The outflow data lake often hosts the data, which will end up in the EDW after it is transformed in the extract, transform, load (ETL) or extract, load, transform (ELT) process, but meanwhile it provides flexibility for the immediate analytics or operational use. This is the prevalent style for capturing the IoT data.

The dominant big data characteristic of the outflow data lake is velocity, while variety is usually pretty low — its rapid ingestion is intended for specific data, such as log files or data for certain applications.

The chief data officer of TD Ameritrade calls the outflow data lake "a data marshaling yard" because it is upstream of other data stores and it is "focused on pulling in chat information and emails, a lot of textual stuff, to try and understand client behavior and so we can optimize the client experience in terms of scenarios." [1] (#dv_1_td_ameritrades)

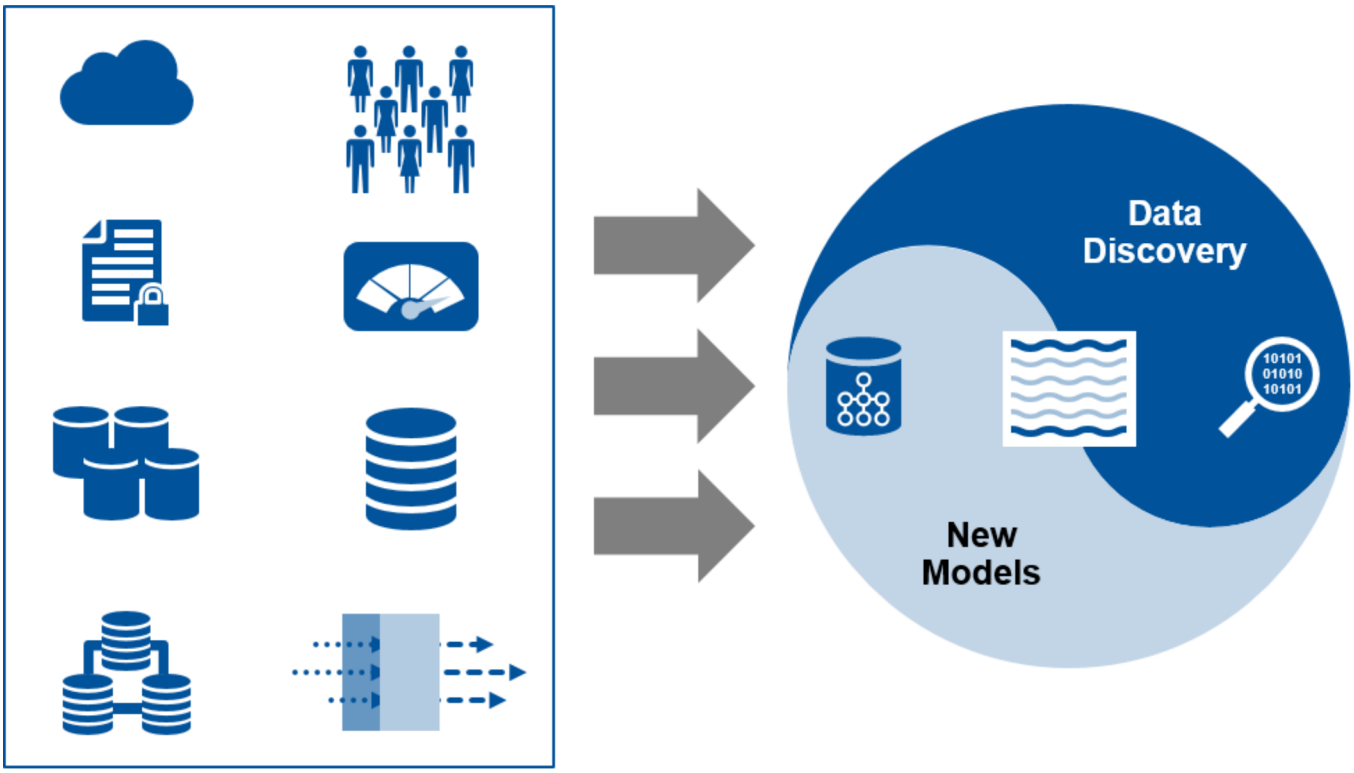
The outflow data lake is usually part of a greater architecture solution, like a staging area for the EDW or part of lambda architecture described in the Guidance section. The outflow architecture style is good for near-real-time streaming analytics, which involves a well-defined, but limited, set of advanced analytics models. It is the style for IoT operations and reporting.

An example of implementing an outflow data lake architecture is described in the Pinterest engineering blog Monitoring A/B Experiments in Real-Time (https://engineering.pinterest.com/blog/monitoring-ab-experiments-real-time) .

Architecture Style No. 3: A Data Science Lab Data Lake

The data science lab style of the data lake architecture is best for enabling innovation. Figure 7 resembles yin and yang: It contains elements of the inflow and outflow styles, but compared to them, it requires little or no governance, just guardrails to ensure that data scientists are not violating basic security, ethics and regulations, such as privacy, laws for data locality or industry compliance. Data science lab also has a more narrow scope of use cases compared to other architecture styles, for example, a data lake for medical research.

Figure 7. Consumption From the Data Science Lab Data Lake



Source: Gartner (July 2016)

The data science lab is the closest to the data mart concept: It makes available a specific scope of data, often curated, to particular users. Like a data mart, which is aimed at a specific purpose or a set of purposes, this data lake does not attempt to solve all the problems of the enterprise. Compared to a traditional data mart, the data science lab can store new, not previously available, data, for data discovery and for developing new advanced analytics models, which are often conducted with new tools. Due to the advanced nature of analytics, the data science lab lake is available only to a small, highly skilled user population.

The dominant big data characteristic of the data science lab is volume, because more data often reveals new insights or contains a proverbial needle in the haystack. For example, machine learning on more data can yield high-quality results without resorting to sophisticated algorithms. The volume of data is high in the lake, but eventually just a small portion of it will be in actual use. The data in this lake is often curated to include relevant sources. In this type of the lake, business acumen is very important, and data scientists usually work side-by-side with specialists who can ask the right questions and interpret the outcomes.

Below are the examples of the data in industry-specific data lakes:

- **Banking:** Accounts, households, savings, credit, payments, ATM, risk, CRM, clickstream, mobile app, social media, call center notes.
- **Public sector:** Usually data about citizens, social security, demographics, labor statistics and public policy combined with specialized information for a particular purpose, such as public

health or regulatory supervision.

- **Telecommunications:** Call detail records (CDR), geolocation, CRM, data from mobile devices, data about network operations.
- **Manufacturing:** Materials, factory equipment (both real-time and historical), documentation, warranties, test results, ERP, bills of material, shipping, process monitoring logs and alarms, regulations.
- **Healthcare:** Data from medical devices, insurance, electronic medical records, treatment protocols, data on experience of using drugs or devices, medical libraries like PubMed (http://www.ncbi.nlm.nih.gov/pubmed) .
- **Higher education:** Data for scientific research.

The data science lab architecture style is usually a stand-alone data lake. Many data lake implementations start with this style.

Summary of the Data Lake Architecture Styles

Table 1 contains the summary of the Analysis section applied to the data lake architecture styles.

Table 1: Summary of Data Lake Architecture Styles

Characteristics ↓	1. Inflow Lake ↓	2. Outflow Lake ↓	3. Data Science Lab Lake ↓
Main implementation driver	■ Bridging information silos	■ Getting to the data faster	■ Innovation

Characteristics ↓	1. Inflow Lake ↓	2. Outflow Lake ↓	3. Data Science Lab Lake ↓
Purposes	■ Data hub ■ Data warehouse offloads ■ New insights from bridging silos	■ Staging area for further analytics and operations ■ ETL offloads ■ IoT ■ Operations	■ Experimentation ■ New models ■ Customer 360 ■ Finding a needle in a haystack
The analogical principle	■ Data warehouse	■ Operational data store	■ Data mart
Consumption model	■ Self-service	■ Services and APIs	■ Analytics sandbox
Dominant consumption method	■ BI	■ Operations	■ Advanced analytics
Dominant data characteristics	■ Variety ■ More metadata and some MDM	■ Velocity ■ Schema-on-read	■ Volume ■ Data of unknown value

Characteristics ↓	1. Inflow Lake ↓	2. Outflow Lake ↓	3. Data Science Lab Lake ↓
Governance	■ Mode 1 and Mode 2 ■ Standards ■ Policies ■ Best practices	■ Mode 2 ■ Standards ■ Data visibility to downstream consumers ■ Agile practices	■ Little to no governance ■ Guardrails ■ Freedom ■ Next practices
Place in the enterprise architecture	■ Part of a greater enterprise architecture	■ Part of a greater end-to-end engineering solution	■ Stand-alone
Challenges	■ Metadata ■ Mapping data across the sources ■ Cost of centralized service ■ Providing data at the right level of expertise	■ Metadata ■ Low latency ■ End-to-end architecture ■ Real-time analytics	■ Metadata ■ Subject-matter data curation ■ Data integration ■ High skills requirement

Source: Gartner (July 2016)

Future of the Data Lake Architectures: Separation of Storage and Compute

Data lakes represent storage exploited by a plethora of compute options, from programming languages to operational applications and data visualization tools. With maturing of data lakes in the enterprise (e.g., successfully exploiting more than one data lake architecture style), the separation of storage and compute will increase.

Hadoop paved the road for loosely coupling storage with other components in the stack that provided compute capabilities. The rise of Apache Spark signified a further move toward separation of storage and compute, where Spark represented compute capabilities that can process data persisted in many ways, including data lakes.

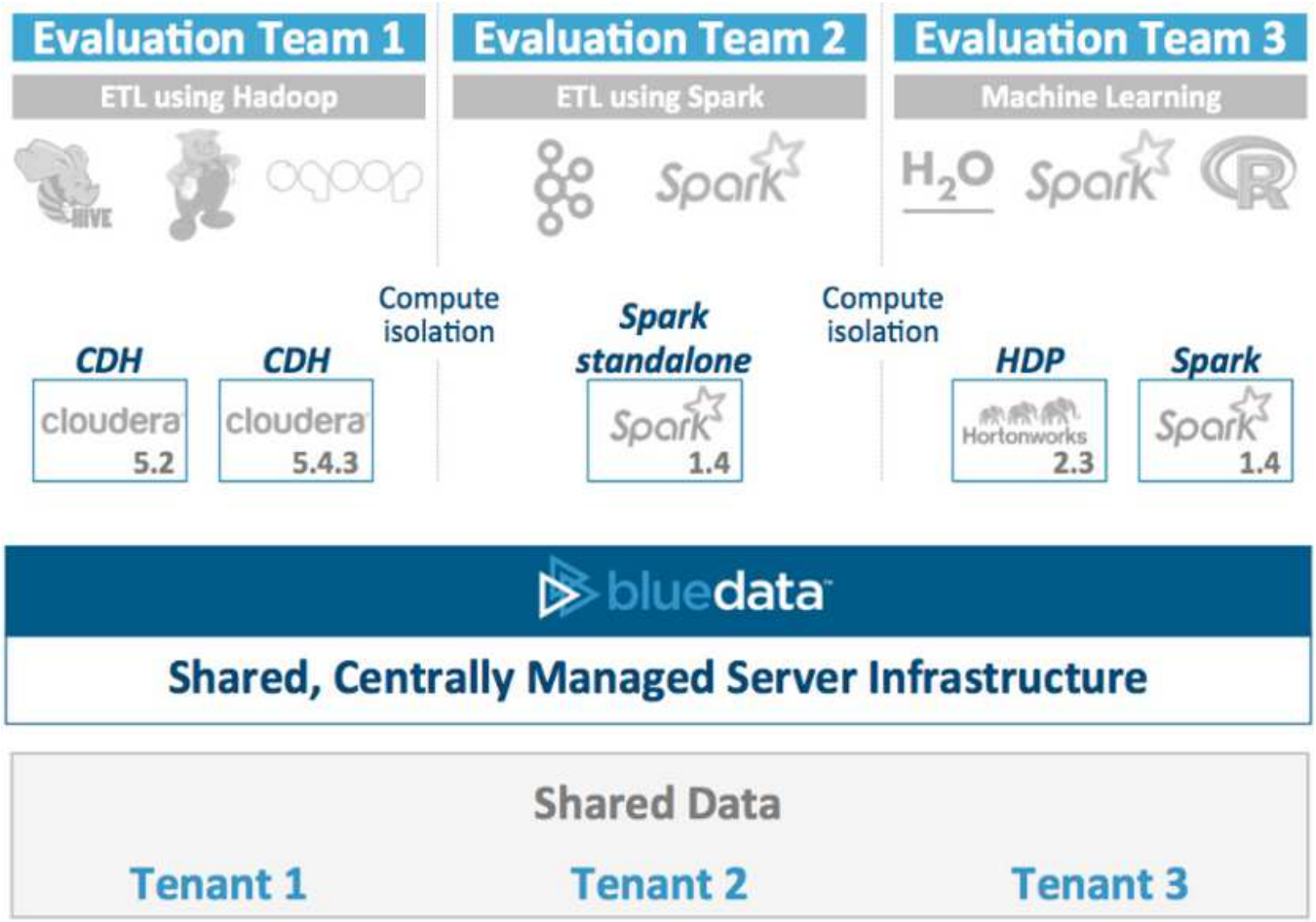
Currently, most cloud providers are eagerly implementing decoupled storage and compute because of the following benefits of this approach to their customers:

- Using the right tool for the right task (and it could be a one-time task)
- No need for capacity planning
- Lowest cost of keeping the data in the cloud when compute is not used
- Independent cost attribution per dataset

Also, the compute choices in the cloud are very wide, from native options offered by cloud providers to third-party tools offered as services in cloud marketplaces.

From the infrastructure perspective, containers and server virtualization draw attention to separation of storage and compute too. Figure 8 illustrates how server virtualization allows simultaneous evaluation of use cases on the same storage.

Figure 8. Example of Separation of Storage and Compute on the Shared Infrastructure



Source: BlueData

Ins and Outs of a Data Lake

A data lake is a collection of storage instances that preserve the data in its exact or near-exact format. The architecture of the data lake mostly depends on the two factors outside the lake:

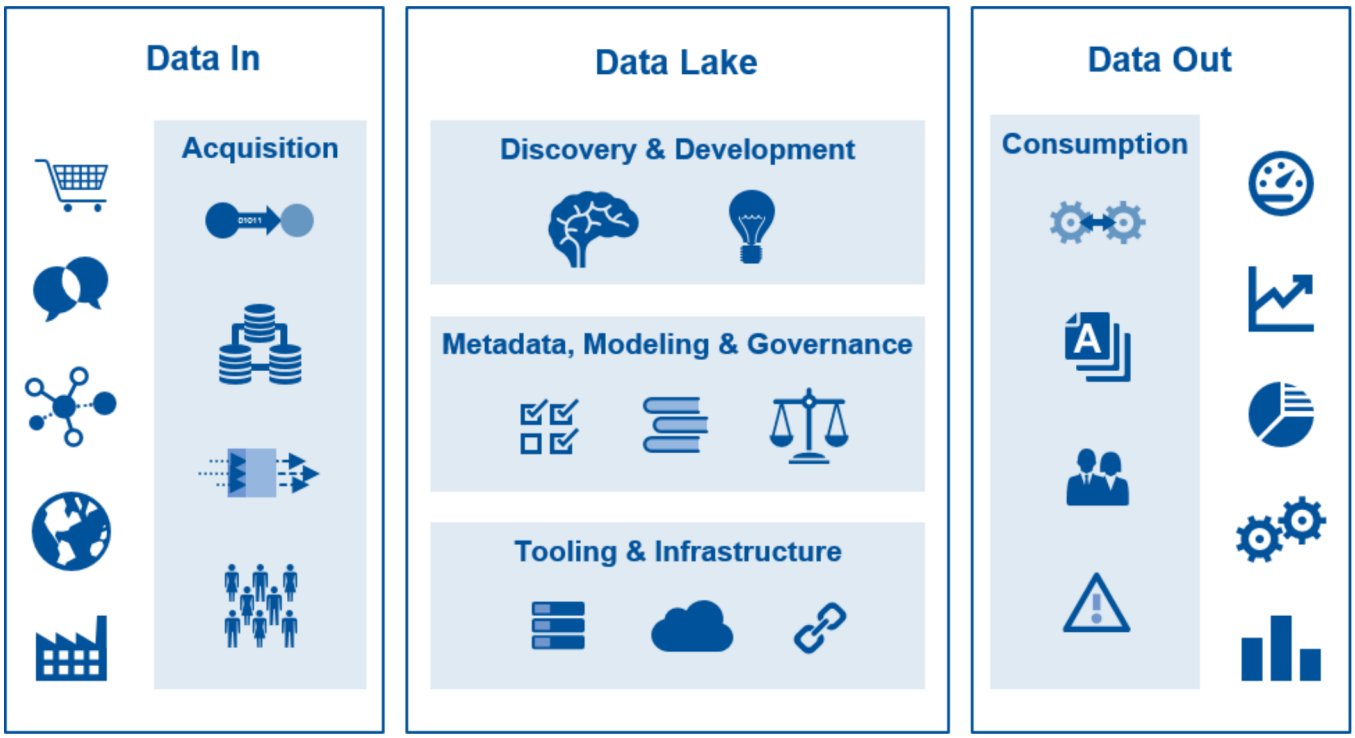
- How the data gets in the lake from the sources.
- How the data gets out of the lake for consumption.

These two factors are out of scope for this research, but they will drive data lake architecture decisions that can be divided in the following groups (see Figure 9):

- How to organize the data in the lake, depending on consumption patterns and on the ease of integrating disparate data sources. The sections Selecting Appropriate Underlying Technologies and Optimizing the Supporting Infrastructure answer this question.
- How to optimize the data consumption for performance and make data usable. The sections Acquiring Metadata, Applying Data Modeling and Establishing Data Governance address these important considerations.

- How to turn the data lake into the place where data science and application development thrive. The answer is in the Obtaining the Right Skills section.

Figure 9. The Data Lake Architecture Depends On How the Data Gets In and Out of the Lake



Source: Gartner (July 2016)

Common Success Factors for Building Data Lakes

All successful data lake implementations have common success factors — appropriate choices of underlying technologies and infrastructure, as well as data modeling, metadata management and data governance. Last, but not least, having a plan for obtaining the right skills to exploit the data lake is of utmost importance.

Selecting Appropriate Underlying Technologies

Myth: A data lake is Hadoop.

In practice, Hadoop is the most popular, but not the only, technology for implementing data lakes. The following technologies are the examples of non-Hadoop choices for data lakes:

- **Relational databases**, for example, Microsoft SQL Server, IBM dashDB, MySQL, MemSQL, Amazon Aurora DB. Use relational database management system (RDBMS) technologies if your data is mostly structured and your requirements include low latency (fast) data

consumption and high user scalability. Most relational databases have in-memory capabilities for data processing and can seamlessly handle tiers of data — from hot (in memory) to cold (on an auxiliary storage).

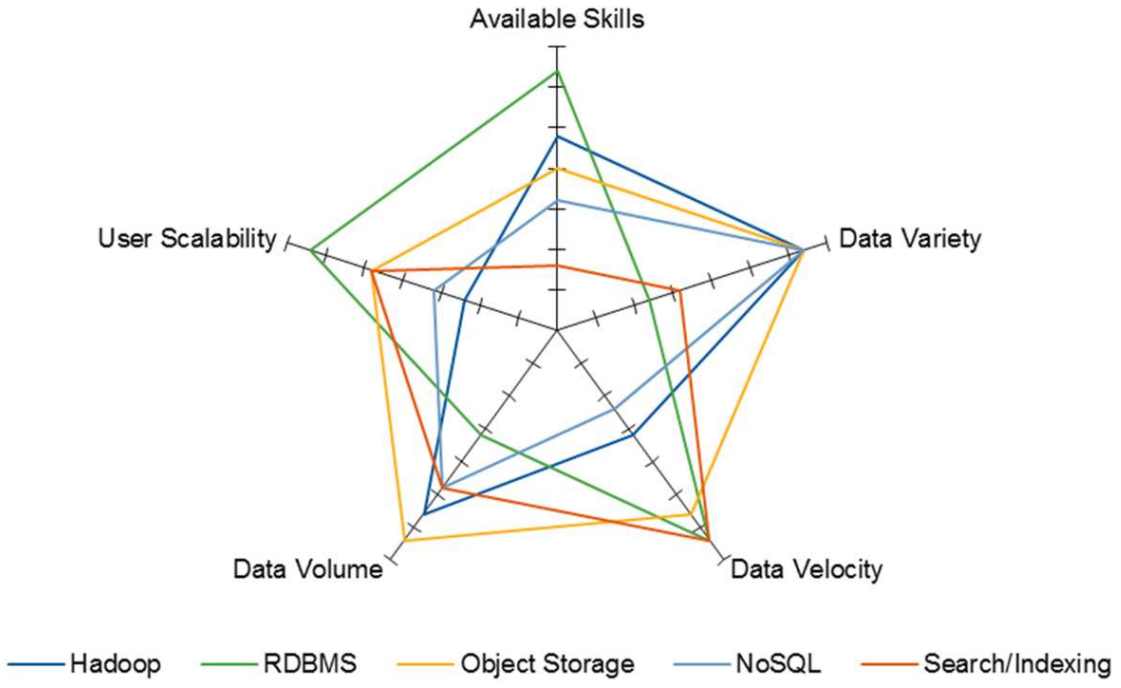
- **Object storage**, like Amazon S3 and Microsoft Azure Blob Storage. These technologies are the most popular choices for the data lakes in the public cloud. They provide virtually unlimited storage for high volume, and are a basis for using the data in the lake with the wide variety of ingest, integration, analytics and development tools available within the same cloud. For more information, refer to "Comparing Object Storage in the Public Cloud: Amazon, Google and Microsoft." (https://www.gartner.com/document/code/277691?ref=grbody&refval=3380017)

- **NoSQL databases**, like Cassandra, HBase and MongoDB. NoSQL technologies can be very different: key-value database management system (DBMS) for rapid storage and retrieval of binary data, a document DBMS for semistructured data and rapid development, a table-style DBMS for event log or clickstream data, or a graph DBMS for relationship analysis. For more information, refer to "Framework for Assessing NoSQL Databases." (https://www.gartner.com/document/code/278868?ref=grbody&refval=3380017)

- **Search/indexing stores**, like Splunk, Apache Solr and Elasticsearch. These technologies are good when the data in the lake is mostly logs, text or geolocation. They can meet requirements of real time or can help less technology-savvy users work with data. How-to: Use Apache Solr to Query Indexed Data for Analytics (http://blog.cloudera.com/blog/2015/10/how-to-use-apache-solr-to-query-indexed-data-for-analytics/) provides an explanation of applicability of Apache Solr.

Figure 10 illustrates high-level strengths and weaknesses of the technologies depending on the available skills on the market (see the Obtaining the Right Skills section) and the parameters in the data lakes architecture comparison that affect technology choices the most — data volume, velocity, variety and user scalability.

Figure 10. Comparison of Data Lakes Technologies by Criteria



Source: Gartner (July 2016)

Figure 10 is intended to give an idea of comparison: For each organization, criteria may vary and technologies will be more specific than generic. For example, Apache Hive can handle very high volume and variety, but it is not good for subsecond response time or user concurrency. In-memory RDBMSs demonstrate extremely low latency and high user scalability, but are limited in volume; most NoSQL databases can be optimized for writing high data volumes but perform poorly on user scalability.

Technical professionals might conclude that many technologies can satisfy their data lakes requirements. In this case, available expertise, use cases and the technology fit into the current business and enterprise architecture are the best parameters for the technology choice.

Finally, the key consideration for selecting technology is the ecosystem to which this technology belongs. Example of ecosystems include Hadoop, Amazon Web Services (AWS), Microsoft Azure and IBM Bluemix. It is the ecosystem that provides storage and compute options. It also contains the tools necessary for getting the data in and out of the lake. For more information on integrating the data into or out of a data lake, refer to "Comparing Four Hadoop Integration Architectures." (https://www.gartner.com/document/code/303737?ref=grbody&refval=3380017) For consuming data out of the lake for analysis, refer to "Architecture Options for Big Data Analytics on Hadoop." (https://www.gartner.com/document/code/275431?ref=grbody&refval=3380017)

Optimizing the Supporting Infrastructure

Myth: Data lakes are inexpensive to implement.

In reality, the goal for the data lakes infrastructure is optimization of cost and efficiency by balancing four parameters — storage, compute, memory and I/O. In many data lake implementations, infrastructure is an afterthought. Infrastructure specialists — cloud, storage, network and security experts — should be involved in the data lake project from its inception. Otherwise, infrastructure costs can sink the entire data lake.

Do not confuse the cost of storage and the absence of licensing of the open source software with the total cost of a data lake implementation. [2] (#dv_2_the_2015) Moore's Law (https://en.wikipedia.org/wiki/Moore's_law) brought data lakes into existence due to the steadily reducing cost of hardware. Because the data in the lakes is vast, imprecise and often of unknown value, it makes sense to store it only at a low cost per terabyte.

A data lake gets expensive when implemented enterprisewide. Production requirements for security, monitoring, performance, high availability and disaster recovery substantially add up to the data lake cost, especially given maturity gaps in the expanding and rapidly changing software solutions for data lakes. Also the cost to get data in, process it and get data out for large volumes is significant.

To build, maintain and exploit the lake, organizations need a multidisciplinary team of specialists whom they pay market price to hire or retain. Optimizing data lakes for analytics is also costly.

Data Lakes On-Premises

Software used for data lakes can run on "commodity" hardware and scale out. Software vendors and end users assume that "commodity" means low cost. In response, hardware vendors cater to the low-cost expectations: Driven by big data, they deliver a relatively inexpensive hardware — in the vicinity of $10,000 per server. [3] (#dv_3_the_2015) However, many enterprises prefer to run their data lakes deployments on high-end hardware to scale up, and then scale out. The larger the data lake, the more nodes are in the cluster, and the network is often a performance bottleneck.

Leading hardware vendors offer reference architecture for data lakes. Examples include Cisco UCS Integrated Infrastructure for Big Data (http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/Cisco_UCS_Integrated_Infrastructure_for_Big_Data_with_SAP_HANA_Vora.html) , EMC Data Lake (http://www.emc.com/en-us/big-data/data-lake/index.htm#accordion3=0) and HP Moonshot for NoSQL Apache Cassandra (http://www8.hp.com/h20195

/V2/GetDocument.aspx?docname=4AA5-7525ENW&cc=us&lc=en) .

Some organizations are looking at appliances as a shortcut to implementing a data lake. Big data appliances, such as Teradata Appliance for Hadoop and Oracle's Big Data Appliance, are the lower-cost systems, compared to the other appliances by the same vendors.

Data Lakes in the Cloud

Select the cloud as the data lake infrastructure not for its pure cost, but for its unique capabilities of elasticity, security, packaged data lake solutions and the access to a wide variety of tools and external data. In the cloud, an option to scale down is important.

Public cloud, managed services and cloud colocation providers are in play for implementing data lakes.

- **Managed services** are most often used for the style No. 3, a data science lab, where a line of business wants to conduct analysis. For example, a data lake for security or for marketing. Managed services can also facilitate multienterprise data science labs. Managed services are the choice when organizations cannot go to the public cloud because of regulatory constraints, user entitlement provisioning and data security requirements. Altiscale (https://www.altiscale.com/) and Think Big (https://thinkbiganalytics.com /big_data_solutions/managed-services/) are examples of managed services.
- **Colocation** is the cloud choice for style No. 1, an inflow lake. It makes sense when the data is already hosted by a colocation provider. This choice simplifies data loads to the cloud; for example, Equinix (https://blog.equinix.com/blog/2015/10/27/data-lakes-and-clouds-flowing-together/) can host NetApp, which will become storage for Azure and AWS. Colocation also provides the best network option for the cloud infrastructure.
- **Public cloud** is the ideal place for "greenfield" analytics projects; for example, IoT, where organizations can build "on the shoulders of the giants," taking advantage of IoT platforms in the cloud and rapidly innovating. Architecture style No. 2 is the most popular data lake in the public cloud.

To make the best infrastructure choice, refer to "Decision Point for Application Placement: Cloud, Managed, Colocation or Do It Yourself." (https://www.gartner.com/document /code/270922?ref=grbody&refval=3380017)

Gartner observes the following patterns of implementing data lakes in the public cloud:

- **An object storage as a data lake:** To instantiate and then destroy a cluster when compute needs are sporadic, for example, a data lake in Amazon S3 and a cluster instantiation on Amazon Elastic MapReduce (Amazon EMR).

- **A permanent cluster, using Hadoop, NoSQL or RDBMS technologies:** In this case, organizations have a good understanding of permanent workloads: They can pack the cluster as much as they can or scale the cluster up and down. Examples include data lakes on Azure HDInsights, Amazon EC2 Reserved Instances (http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts-on-demand-reserved-instances.html) , DataStax Enterprise (https://docs.datastax.com/en/datastax_enterprise/4.5/datastax_enterprise/install/installCloudTOC.html) , Cloudera Enterprise (http://www.cloudera.com/documentation/other/reference-architecture/PDF/cloudera_ref_arch_aws.pdf) and Cazena (http://www.cazena.com/) .
- **A data lake in the hybrid deployment:** Here the data is both on-premises and in the cloud, or in multiple clouds. Ingestion sources could be different in different locations, for example, when multiple organizations have a common data lake, but the solution should handle it as a single data lake. Hybrid deployment comes into play when a data lake architecture outgrows a single architecture style and matures to accommodate two or three styles. Vendor examples include Microsoft SQL Server Stretch database (https://azure.microsoft.com/en-us/services/sql-server-stretch-database/) , Teradata, MapR (https://www.mapr.com/company/press-releases/mapr-expands-customer-cloud-deployment-options-hadoop-amazon-web-services) and Qubole (https://www.qubole.com/?nabe=5695374637924352:1) . Many integration vendors — such as Informatica's Intelligent Data Lake and IBM BigInsights — facilitate multistorage data lakes.

Acquiring Metadata

Myth: Get all the data you can in the data lake.

In fact, Gartner predicts that through 2018, 90% of deployed data lakes will be useless as they are overwhelmed with information assets captured for uncertain use cases. [4] (#dv_4_metadata_is) Data for uncertain purposes hoarded in many lakes creates a lot of noise that can obstruct hearing the signals in data discovery and developing new models.

Metadata is taking a center stage in the big data world exactly because of the need to find the right information in the vastness of data lakes. Most data lake implementations lack capabilities to capture data lineage and findings by those who have previously discovered value using the same data. Metadata — both machine- and human-generated — is a requirement for a successful data lake.

Metadata is perspective-based and ties into use cases and users. Data lakes implementations should capture the context, lineage and frequency of the incoming data, as well as the data

definitions from a business-use-case perspective. Metadata should include the information about data creation, usage, privacy and trustworthiness, as well as regulatory and encryption business rules that apply to the incoming data.

Consequently, a vendor community has sprung up to offer metadata management solutions that automatically capture the necessary information at the time of acquisition, as well as crawl the lakes to create metadata. The following options are available for data lakes metadata needs.

- **A data catalog** is a rapidly emerging technology accompanying data lakes to bring clarity and ease of use to raw data of high volume, velocity and variety. Examples include Apache Atlas, Cloudera Navigator, Azure Data Catalog, Amazon S3 Metadata Index, Alation, Zaloni and Waterline Data.
- **Enterprisewide integration tools capable of cataloging information assets** in data lakes and beyond. Vendor examples include Informatica, IBM, Cisco, Pentaho and Talend.
- **Tools for keeping track of metadata enterprisewide**, like Attunity, Collibra, Cambridge Semantics, Oracle, SAP, Global IDs, Data Advantage Group and Adaptive.
- **Self-service data preparation** capabilities make users understand and interpret the data for their own needs. Tools include Paxata, Trifacta and Tamr.

More information on metadata management approaches and tools is available in "Comparing Three Self-Service Integration Architectures" (https://www.gartner.com/document/code/297311?ref=grbody&refval=3380017) and "Metadata Is the Fish Finder in Data Lakes." (https://www.gartner.com/document/code/274543?ref=grbody&refval=3380017)

Applying Data Modeling

Myth: When building a data lake, no data modeling is necessary upfront.

In reality, thinking ahead about how to organize the data for efficient analytics and processing for your use cases is time well spent. Data modeling expresses use-case requirements at the conceptual, logical and physical level.

- **Conceptual** models express the business semantics of data, without regard for any technological aspects of how that data is represented or the tools or paradigms that are used to store it. The first decision for data lake design is whether to store the data in native or near-native form; this decision will affect the choice of tools and data acquisition. For

example, what data should we include in the data lake for a 360-degree customer view? Should it be raw or preprocessed (enriched, profiled, deduplicated, etc.)? Business users and data science teams are the main contributors to conceptual models.

- **Logical** models are a set of assertions about how a conceptual model can be rendered in a particular technology, such as JavaScript Object Notation (JSON), Apache Parquet, Apache Avro, Apache Kafka topics or a relational model. For example, relational data loaded in data lakes could be flattened out in columnar format for in-memory processing efficiency. Indexing is a common practice for searching accessibility to data in the lake. Making a decision on data standards is the most important step at this point. Logical data models are devised by technology specialists to make design decisions required by a selected underlying technology for a data lake.
- **Physical** models are a set of assertions about how a logical model is implemented in a specific infrastructure, such as a Hadoop cluster or a cloud object storage. Physical modeling includes data organization for compute efficiency, user access, availability, security, location awareness and SLA requirements. For example, partition large datasets for query efficiency, repackage small files for better storage utilization or design the physical layout for high availability. If not done upfront, reorganization of the infrastructure from optimization for incoming data on acquisition to optimization for user access on consumption could be costly. A physical data model is driven by engineers who are responsible for the implementation, deployment and maintenance of the delivered data lakes.

Establishing Data Governance

Myth: By virtue of keeping all data in one place, you get a single source of the truth.

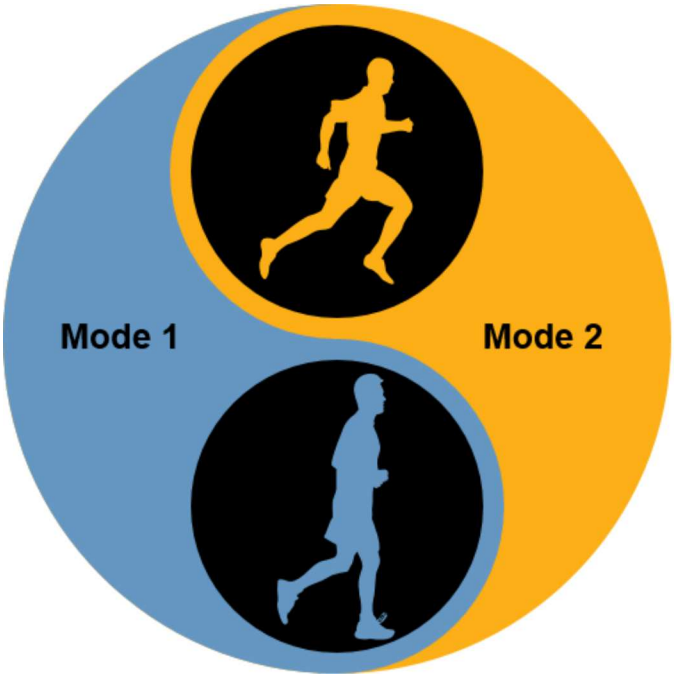
Much of the data collected in the data lake is defined as schema-on-read, where the structure of the data collected is not known upfront but needs to be evaluated through discovery when it is read. The function of data governance is enabling users to best interpret the data in schema-on-read from their unique vantage point.

For data lakes, data governance is better described as data advocacy to prevent or minimize mistakes due to misinterpretation of data or inability to find the right information in the vastness of the lake. Data policies and standards to avoid anarchy in the lakes, change management for various versions of data and models, and data lineage to avoid disasters with data derivatives are advocacy tasks. Additionally, data governance expands to governing

analytics of the data in the lakes. [5] (#dv_5_the_gartner)

Data lakes governance approaches should exercise bimodal [6] (#dv_6_bimodal_it) flexibility: Where Mode 1 embodies stability and long-term planning, Mode 2 supports agility and quick results (see Figure 11).

Figure 11. Bimodal Governance



Mode 1	Mode 2
Reliability	Agility
More governance	Less governance
Long projects	Short iterations
Schema-on-write	Schema-on-read
Better data	More data
Data reuse	Data discovery
Data-centric	App-centric
Specialized skills	Versatile skills
High risk if fails	Low risk if fails

Source: Gartner (July 2016)

Governance differs depending on the main goal for building a data lake:

- Bridging information silos in the inflow data lake involves both Mode 1 governance, necessary for the data hub, and Mode 2 governance to maintain agility while adding new data sources and analyzing data in the lake.
- Getting to the data faster in the outflow data lake employs mostly Mode 2 to sustain the schema-on-read flexibility. Schema-on-read results in many subsequent schemas from the same data in the lake compared to schema-on-write. Governance should ensure agility in quickly landing the data and in supporting multiple downstream perspectives.
- Innovation in the data science lab lake thrives on freedom of experimentation. This data lake style needs only minimal governance in the form of guardrails to prevent legal or ethical violations. Note that no governance is a conscious governance decision.

For more information on information governance, refer to "EIM 1.0: Setting Up Enterprise Information Management and Governance." (https://www.gartner.com/document/code/294451?ref=grbody&refval=3380017)

Obtaining the Right Skills

Myth: Everyone can use the data lake.

Technical professionals and their business counterparts underestimate the skill level to conduct data discovery and new models development directly in the data lake. It is usually a more high-touch skill than initially expected — an average employee doesn't know math, statistics and machine learning, so organizations end up in the infinite loop of conceptually understanding the value of a data lake but not being able to take advantage of the data due to the lack of data science skills.

Plan to evolve your in-house data and analytics skills toward data engineering and data science, and sparsely hire new talent to cover skills gaps. Cultivate literacy of data fundamentals among the developers who code directly on the data lakes. Have data scientists work side-by-side with business domain experts — this is an effective method of exploiting a data science lab lake.

Guidance

Follow the guidelines in this section in addition to selecting the right data lake architecture style and implementing key success factors.

Start With the Single Architecture Style

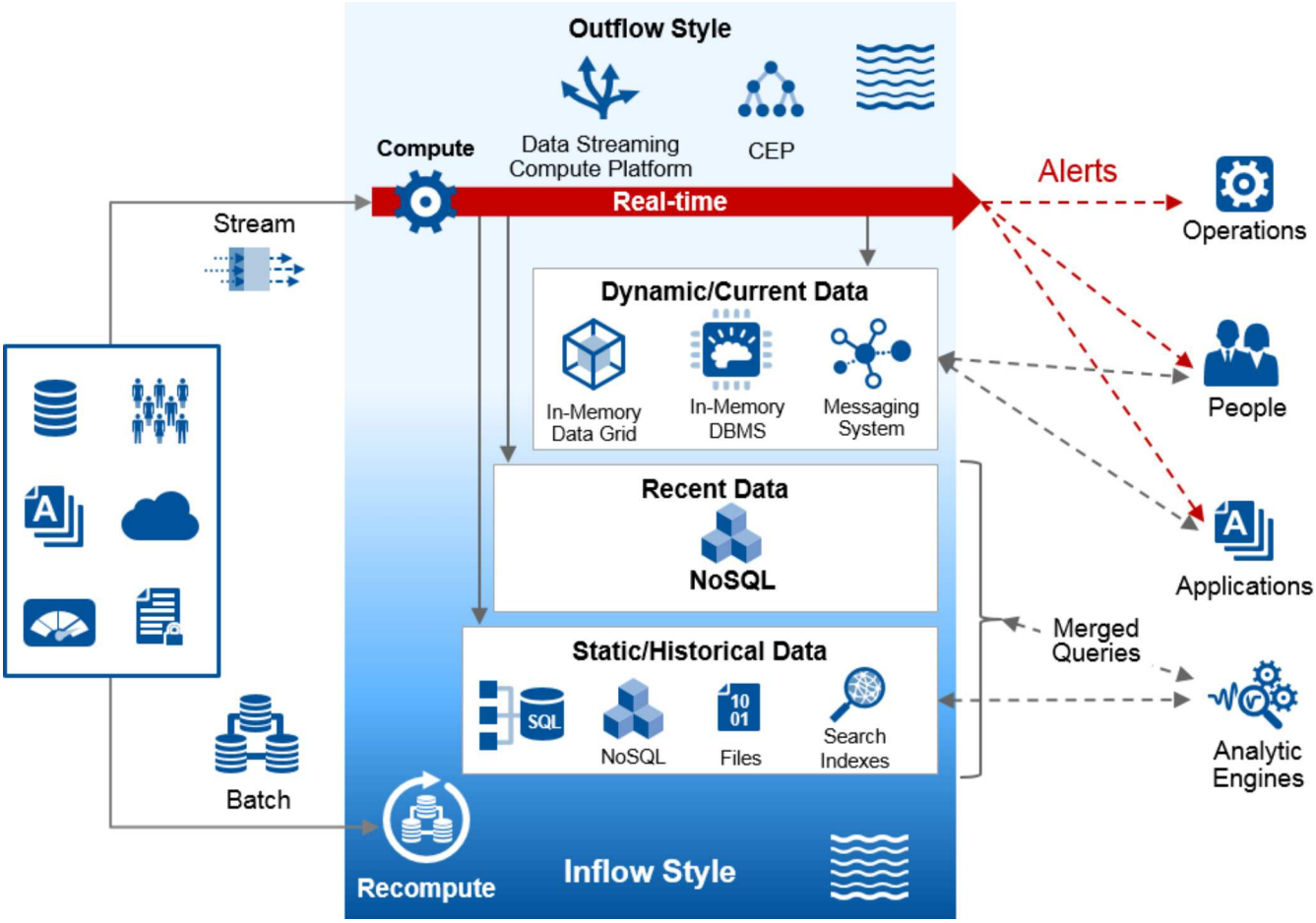
Myth: Data lakes contain petabytes of raw data.

In fact, only advanced data lakes deployments are ready to handle truly raw data at scale. Mature data lakes may contain elements of all three styles. An example of an advanced data lake is Cosmos (https://azure.microsoft.com/en-us/blog/behind-the-scenes-of-azure-data-lake-bringing-microsoft-s-big-data-experience-to-hadoop/) , Microsoft's internal data lake that "manages exabytes of diverse data (ranging from clickstreams and telemetry to documents, multimedia and tabular data) in clusters that each span more than fifty thousand machines."

Successful data lake beginners start with implementing a single architecture style, usually a data science lab lake for narrow purposes. Data lakes continue in a single style while

infrastructure and initial data ingest are stabilizing. Then a data science lab may expand to the inflow or inflow architecture. Figure 12 depicts further combining the outflow and the inflow data lake styles in the Lambda Architecture (http://lambda-architecture.net/) .

Figure 12. Combining the Inflow and Outflow Data Lake Styles in Lambda Architecture



Source: Gartner (July 2016)

Most event analysis does not require fresh data acquired with low latency. It is typically a part of an interactive investigation of an issue in the inflow-style architecture or in a data science lab. Exploratory analysis of historical event data is common for operational technology (such as equipment performance analysis), financial applications (such as stock market analysis), marketing applications (such as studying customer buying behavior), web analytics and other applications.

The inflow data lake accommodates batch or interactive (ad hoc) analysis of large volumes of historical data. Event streams and other data may be directly loaded into an appropriate persistent data store (discussed in the Selecting Appropriate Underlying Technologies section) for processing purposes and made available for analytics.

The outflow data lake adds the "Real-time Layer" on the top. Lambda architecture provides a view of historical and recent data through merged queries in a shared data access layer. Event streams are processed in real-time through complex-event processing (CEP) engines,

distributed-stream computing platforms such as Apache Storm, or a combination of those. They are immediately available through NoSQL databases. The outflow architecture can also include alerts to people, devices or systems. It can store current event data to in-memory data grids or in-memory databases for use by applications or people.

Examples of implementing Lambda architecture can be found in the CERN's presentation Monitoring Update (https://indico.cern.ch/event/452559/contributions/1120070/attachments/1174541/1697280/monitoring-data-lake.pdf) and in the Weather Company's talk Lambda at Weather Scale (https://www.youtube.com/watch?v=G2MDcm_wF1s&index=7&list=PL-x35fyliRwjE2DhodDQARUP0NVKaA_h5) .

Implement the Data Lake for Its New Capabilities

Myth: A data lake is the new EDW.

While the EDW is a proven and well-understood system of record, the data lake is the new and highly debated system of insight. Table 2 outlines the main differences between a data lake and a data warehouse.

Table 2: The Differences Between a Data Lake and a Data Warehouse

Data Lake ↓	Data Warehouse ↓
System of insight	System of record
Convenience	Productivity
Physical collection of uncollated data	Data of common meaning
Schema-on-read	Schema-on-write
Data of unknown or unconfirmed value	Well-understood, high-value data
More-detailed data	More-refined data
Stores any data type	Stores limited data types
Data immutability to preserve history	Time invariant to preserve reusability

Data Lake ↓	Data Warehouse ↓
Limited skills availability	Skills are mostly available
Low cost per terabyte	High cost per terabyte
Optimized for cost-effective storage	Optimized for I/O and CPU
Data discovery	Data and analytics reusability
Low user concurrency	High user concurrency
Varying SQL query performance	Predictable SQL query performance

Source: Gartner (July 2016)

Carefully Plan How the Data Flows In and Out of the Lake

The critical tasks in implementing robust data lakes solutions are data ingest and data consumption.

Data Ingest

Myth: A data lake is a data integration method.

In reality, replacing some of the steps of a data integration process with a data lake doesn't eliminate data integration. Instead, it creates a new design and adds data engineering, such as building data pipelines, to an existing architecture.

A decision on whether to load data in the native or near-native format dramatically affects complexity of data acquisition into a data lake. A near-native format makes data consumption more convenient, but it requires data governance, and more data modeling and data engineering compared to relatively straightforward loads of data in its native format.

An initial load and ongoing loads are the two distinct ingestion needs that require totally different approaches.

- **The initial bulk load** is always a batch activity where a key performance indicator (KPI) is throughput for a faster time to load.

- **The ongoing loads** should satisfy a single criterion in the consistency, availability and partition tolerance (CAP) theorem — consistency or availability, because the third criterion — partitioning — is always present in the distributed systems. In addition, ongoing KPIs will depend on SLAs for selected use cases. Programmatic solutions can be used to create services for coping with many different sources, constantly adding new information or removing old information.

For information about data ingestion, refer to the following Gartner research:

- "Use Data Integration Patterns to Build Optimal Architecture" (https://www.gartner.com/document/code/270543?ref=grbody&refval=3380017)
- "Legacy Data Migration Is a High-Risk Project — Be Prepared!" (https://www.gartner.com/document/code/263940?ref=grbody&refval=3380017)
- "Eventual Consistency and Its Implications: Can You Trust Your DBMS?" (https://www.gartner.com/document/code/276734?ref=grbody&refval=3380017)

Data Consumption

Myth: A data lake can scale to thousands of users.

"Big Data Analytics Failures and How to Prevent Them" (https://www.gartner.com/document/code/272497?ref=grbody&refval=3380017) describes a failed attempt to build a data lake without proper planning for data consumption. Specifically, the failed data lake was designed for fast data acquisition, and it totally missed the point of serving this data to the concurrent users (customers!) who were frustrated with extremely slow response time.

Data lake implementers seldom consider how applications or processes will consume data. In the inflow data lake, data frequently lacks sufficient metadata or is simply persisted on the wrong technology layer. The outflow data lake often starts with the assumption of a great benefit of real-time analytics or operations on the streaming data. Low latency expectations soon break because of a business process that is not able to act in real time, or because of some factor external to the data lake, for example, a slow public internet connection.

Data lakes implementation should start with confirming a business value and technology in a proof of concept, and most importantly — with a business champion committed to implement the planned use cases within the next nine to 12 months.

Beware of Overcommitting and Underdelivering

Myth: If we build a data lake, people will use it.

Select the right use cases for the data lake. Use cases support focusing on the right data in the lake and on selecting the right data lake architecture style. Often, technical professionals execute flawlessly on a single element — for example, automated ingest, without considering how the ingested data will be consumed and even stored in the lake. As a result, data keeps coming in, with no clear idea of what to do with it, no metadata to look at the data and growing costs for the lake's maintenance. This situation puts more and more pressure on people who often bet their careers on the data lakes, and as a result, these people defend data lakes by all means: They make up data lake benefits and put huge efforts to attracting users for the wrong reasons.

One Gartner client called defending a data lake with growing data volumes and no value "an escalated commitment." Both technical professionals and their organizations should be aware of escalated commitments and prevent them by careful planning and implementation of the data lake.

Evidence

[1] "TD Ameritrade's Big Data Push One Year Later: Benefits Coming From All Corners," (http://www.networkworld.com/article/3026263/big-data-business-intelligence/td-ameritrade-s-big-data-push-1-yr-later-benefits-coming-from-all-corners.html) Network World, 25 January 2016.

[2] The 2015 average annual storage cost per raw TB of capacity was $2,009 (see "IT Key Metrics Data 2016: Key Infrastructure Measures: Storage Analysis: Multiyear" (https://www.gartner.com/document/code/291390?ref=grbody&refval=3380017)).

[3] The 2015 average annual Linux x86 server cost per OS instance is $8,454 (see "IT Key Metrics Data 2016: Key Infrastructure Measures: Linux x86 Server Analysis: Current Year" (https://www.gartner.com/document/code/291385?ref=grbody&refval=3380017)).

[4]"Metadata Is the Fish Finder in Data Lakes" (https://www.gartner.com/document/code/274543?ref=grbody&refval=3380017)

[5] The Gartner blog post "Balancing Data and Analytics Governance" (http://blogs.gartner.com/svetlana-sicular/balancing-data-and-analytics-governance/) gives a brief overview of analytics

governance.

[6] Bimodal IT is an approach to IT operations that is intended to allow applications, system architecture and data management to move fluidly between critical, focused support of business processes or "stability" (Mode 1) and the purposeful innovation that is often referred to as "agility" (Mode 2). Mode 2 also, however, provides critical business support — agility does not mean precariousness. Agility in IT means the ability to move rapidly from one level of stability to another without incurring any appreciable loss of critical support to the business.

Document Revision History

Use Design Patterns to Increase the Value of Your Data Lake - 29 May 2018 (https://www.gartner.com/document/code/342255?ref=ddrec)

Recommended by the Author

Solution Path: Implementing Big Data for Analytics (https://www.gartner.com/document/code/294453?ref=ggrec&refval=3380017)

Upgrade the Enterprise Data Warehouse Architecture With Hadoop (https://www.gartner.com/document/code/275430?ref=ggrec&refval=3380017)

Architecture Options for Big Data Analytics on Hadoop (https://www.gartner.com/document/code/275431?ref=ggrec&refval=3380017)

Defining the Data Lake (https://www.gartner.com/document/code/276838?ref=ggrec&refval=3380017)

Initiate and Sustain a Business Glossary to Improve the Value of Business Analytics and the Logical Data Warehouse (https://www.gartner.com/document/code/263937?ref=ggrec&refval=3380017)

How Chief Data Officers Can Use an Information Catalog to Maximize Business Value From Information Assets (https://www.gartner.com/document/code/292843?ref=ggrec&refval=3380017)

Are DBMS Appliances in Your Future? Don't Bet on It! (https://www.gartner.com/document/code/274952?ref=ggrec&refval=3380017)

Calculating and Comparing Data Center and Public Cloud IaaS Costs (https://www.gartner.com/document/code/297328?ref=ggrec&refval=3380017)

Combine Pace Layering and Bimodal IT to Modernize Your Information Infrastructure (https://www.gartner.com/document/code/276946?ref=ggrec&refval=3380017)

Recommended For You

Toolkit: Best of ... Data and Analytics Strategies (https://www.gartner.com/document/3868210?ref=ddrec&refval=3380017)

Solution Path for Implementing a Comprehensive Architecture for Data and Analytics Strategies (https://www.gartner.com/document/3880568?ref=ddrec&refval=3380017)

Toolkit: Map Your Data Management Landscape With the Data and Analytics Infrastructure Model (https://www.gartner.com/document/3874478?ref=ddrec&refval=3380017)

Securing the Big Data and Advanced Analytics Pipeline (https://www.gartner.com/document/3871496?ref=ddrec&refval=3380017)

Use Design Patterns to Increase the Value of Your Data Lake (https://www.gartner.com/document/3876783?ref=ddrec&refval=3380017)

JOIN THE CONVERSATION

Svetlana Sicular
Research VP
Gartner Peer Connect

"Webservices, APIs Or RPA"

by tor_christian_ulvang1

EXPLORE KEY INITIATIVES

Analytics and BI Strategies

The analytics and BI strategies initiative focuses on the strategies, practices, technologies and products needed to support a variety of users across different types of business problems.

+ Track

RECOMMENDED BY THE AUTHOR

Solution Path: Implementing Big Data for Analytics

View More

RECOMMENDED FOR YOU

Toolkit: Best of ... Data and Analytics Strategies

View More