Data Lakes

IN A

Modern Data Architecture

BLUEGRANITE

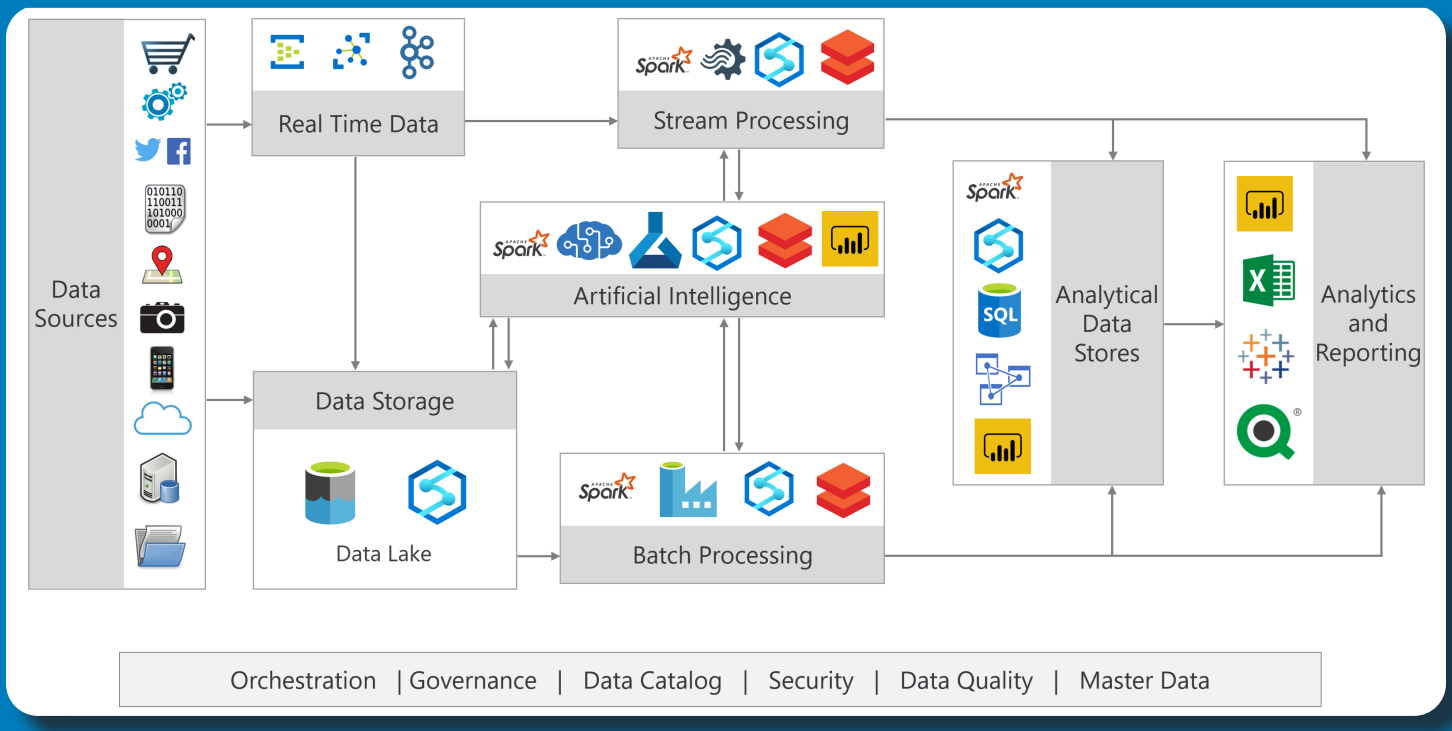
DATA • INSIGHTS • ANALYTICS

TABLE OF CONTENTS

Modern Data Architecture

What does it really mean to implement a modern data architecture? Like many other technology initiatives, it really depends on the implementation objectives. The following characteristics are most commonly associated with a modern data architecture:

- Data originating from internal systems, cloud-based systems, as well as external data provided from partners and third parties
- Diverse set data sources and multi-structured formats
- Streaming real-time data, loading in batches, or some combination of both
- Data volumes that range from moderate to high
- Cloud-based and hybrid delivery modes
- Delivery of analytics to traditional platforms such as data marts and semantic layers, as well as specialized databases like graph, spatial, or NoSQL
- Data virtualization techniques employed, in addition to data integration
- Analytics use cases ranging from operational BI to corporate BI to advanced analytics and data science
- Multi-platform data architecture to suit a variety of use cases
- Agile delivery approach with iterative delivery cycles
- Providing support to a diverse group of users, whether casual data consumers, data analysts, or data scientists
- Automation and DevOps to reduce time-to-value and ensure solution consistency and quality



Business Needs Driving Data Architectures to Evolve & Adapt

Today's business leaders understand that data holds the key to making educated and supportable decisions.

Traditional data warehousing and business intelligence approaches have been challenged as being too slow to respond. **Reducing the time to value** is a fundamental objective of a modern data architecture.

Data warehouses have traditionally excelled in simplifying data access and answering many of the questions required to successfully run the business. However, it's impossible to anticipate every question a business might ask and every report they might need. In a modern data architecture, **acquiring new data should be relatively easy** so that new analysis can be conducted swiftly.

Data volumes have exploded as businesses discover the value contained within social media, documents, comments, sensors, and edge devices. Fifteen years ago, companies never expected to have to keep track of things such as social media "likes." **The ability to capture and analyze practically any type of data** is a critical business capability.

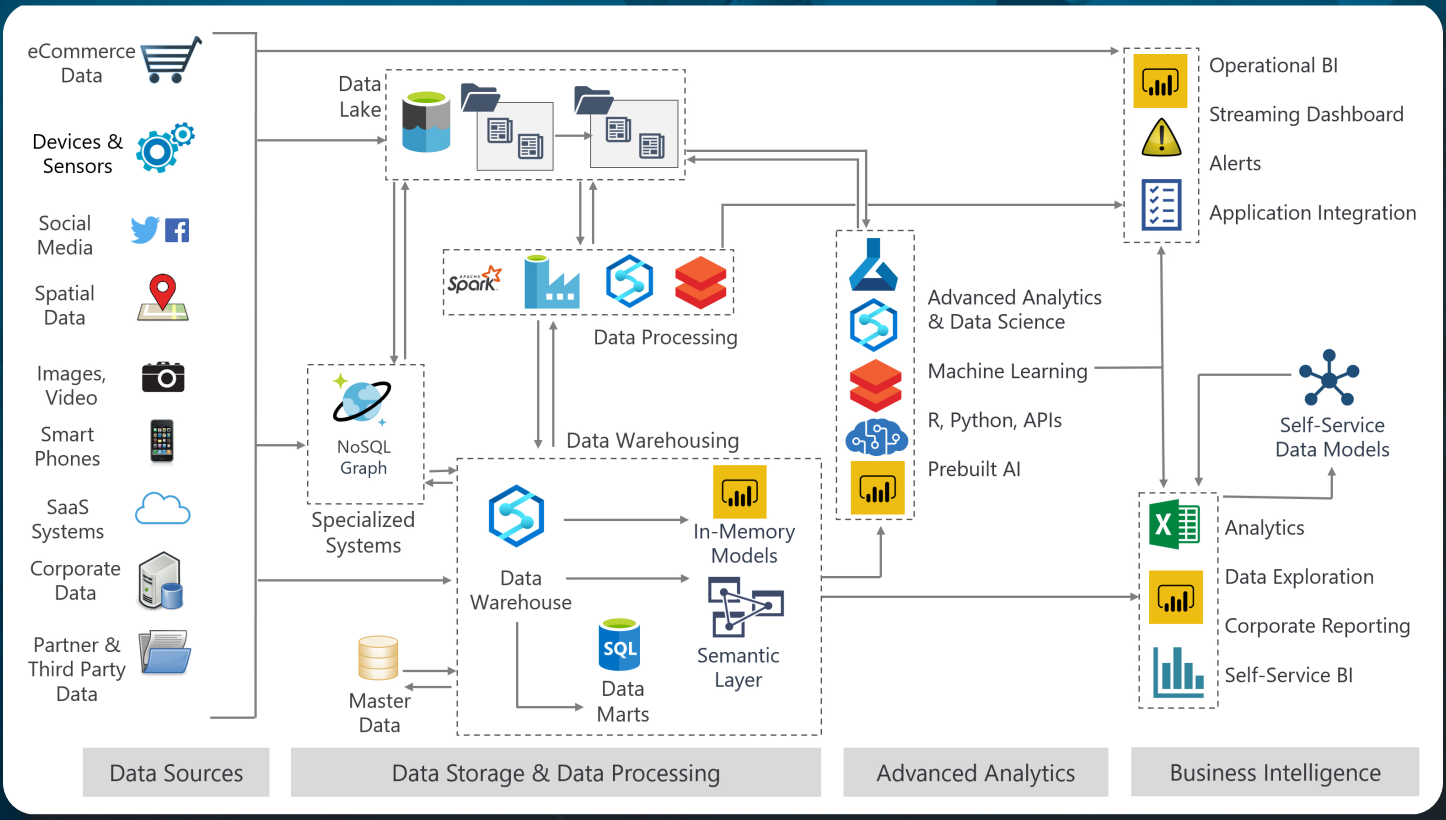
A final thought, users need to know that **data in the data lake is governed, high quality, and not a disorganized unreliable swamp**.

With all the media hype around data lakes and big data, it can be difficult to understand how — and even if — a technology like a data lake makes sense for your analytics needs. Some people believe that implementing a data lake means throwing away their investment in a data warehouse. This perception ends up either sending them down the wrong path or causes them to sideline big data and data lakes as a future project.

The good news? At BlueGranite, we believe that a data lake does not replace a company's existing investment in its data warehouse. In fact, they complement each other very nicely. With a modern data architecture, organizations can continue to leverage their existing investments, begin collecting data they have been ignoring or discarding, ultimately enabling analysts to obtain insights faster.

Principles of a Modern Data Architecture

Big data technologies, such as a data lake, support and enhance modern analytics but they do not typically replace traditional systems.



Multi-platform architectures have become the norm

Within a modern data architecture, any type of data can be acquired and stored. Some implementers elect to accumulate and centralize *all* data within a data lake. Though this "everything in the data lake" approach is architecturally simple and certainly may provide significant value, there are many decisions to make which ultimately impact how the data lake is consumed. Conversely, a multi-platform architecture (depicted above) focuses on best fit engineering, which deems the most effective technology to be based on the data itself. Data remains valuable as changes in technology and enhancements with tools expand. This thought process, also referred to as a polyglot persistence strategy, results in data being distributed across different storage platforms. It is true that no single architecture can be all things to all data, so it is important to find the right balance between architectural simplicity and best fit engineering.

Data integration and data virtualization are both prevalent.

Many IT professionals have become less willing to take on data integration – that is, the requirement to physically move data before it can be used or analyzed. In reality, a lot of data integration still occurs, but it is more thoughtful and purposeful. Data virtualization and logical data warehouse tactics, such as federated queries across multiple data stores, are ways to "query data where it lives." Minimizing data movement is useful in situations such as:

- Large datasets, impractical to move
- Short time window to do data integration
- Data privacy, regulatory, geographic concerns
- Loss of metadata or additional context

Data analysis capabilities are flexible.

A key tenet of the modern data architecture is its agility to rapidly-emerging business needs. Having the ability to access the data very early in the data life cycle, before it has been curated or refined for broad use, offers significant flexibility. Analysis of data in place to determine its value (schema on read) is typically handled by a data analyst or data scientist. Data which is deemed to be valuable to a large audience may be delivered from either the data warehouse or the data lake in a curated data zone – either way, at that point it is considered schema on write.

The architecture is constantly and iteratively changing.

Data lakes support iterative and agile work cycles. By imposing fewer up-front constraints, they allow the delivery team to more quickly conceptualize ideas and solicit feedback from the business. Pushing this collaboration earlier into the project life cycle prevents costly misdirection.

- Raw data becomes progressively more refined as use cases are determined.
- Access to data becomes progressively less restricted as curated, user-friendly data structures are created.
- Sandbox or proof-of-concept solutions can become operationalized for broader consumption.

Other than the raw data, which is immutable, all other areas of the modern data architecture are subject to iterative improvement and change.

The data lake and the data warehouse work in tandem.

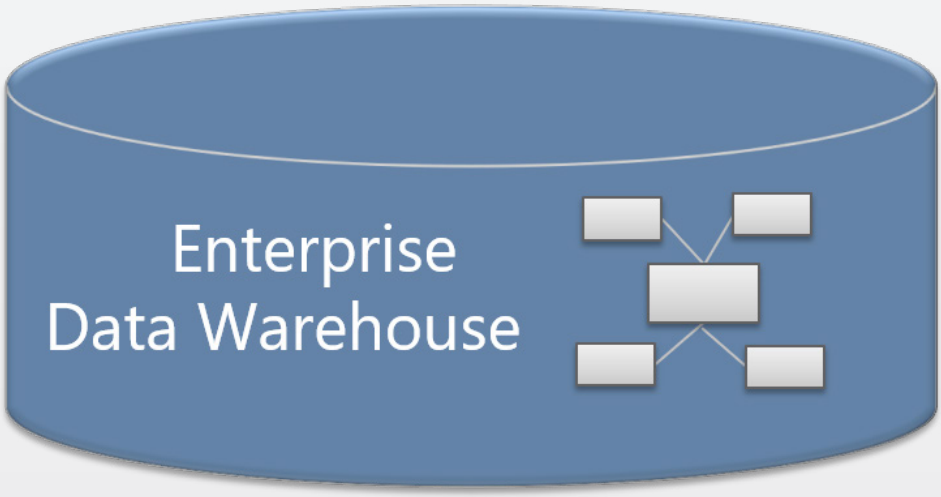
As shown in the diagram at the beginning of this section, both the data lake and the data warehouse are central players in the data storage area. Each is equally important, with complementary roles to play. Azure Synapse is an example of a service that unifies the analytical experience by integrating the data warehouse, data lake, advanced analytics, and business intelligence services together.

Data Lake + Data Warehouse: Complementary Solutions

A traditional data warehouse is a centralized repository containing information which has been integrated from multiple source systems and structured in a user-friendly way that facilitates analytical queries.

A data warehouse has the following properties:

- It represents an abstract picture of the business organized by subject area
- Data is highly transformed, cleansed, and structured
- Data is not added to the data warehouse until the use for it has been defined
- It generally follows a methodology such as those defined by data warehousing pioneers Ralph Kimball and Bill Inmon



Data Warehouse

Data warehousing is characterized by requiring a significant amount of discovery, planning, data modeling, and development work before the data becomes available for analysis by the business users.

This up-front effort to prepare data for user consumption is referred to as ***"schema on write"*** because the schema has to be defined before the data can be loaded. A data warehouse focuses on providing:

- Cleansed, user-friendly, structured data
- Reliable, accurate data ("one version of the truth")
- Standardization of processes
- Pre-defined data structures

Data Lake

Data lakes are broadly accepting of new data regardless of the format. This is a marked departure from the rule-laden, highly-structured storage within traditional relational databases. While this helps relational databases maintain high standards for data quality, this heavy-handed enforcement of dataset schemas can impede the rapid and iterative development.

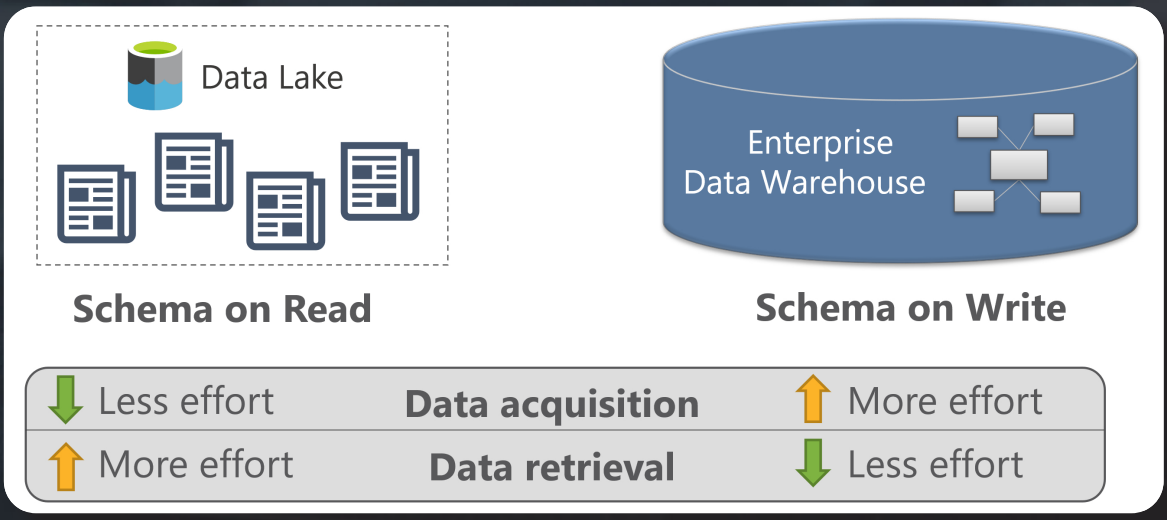
The philosophy of data lakes is to accept new data instantly and with few restrictions, but then apply the rigors of business logic, type-checking, and data governance when it comes time to use (or "read") the data. This is widely termed "schema on read", in contrast to the relational database approach of "schema on write".

This flexibility allows for new value propositions that are more difficult or time-consuming to achieve with a traditional data warehouse. A data lake focuses on providing:

- One architectural platform to house any type of data: machine-generated data, human-generated data, as well as traditional operational data
- Fewer obstacles to data acquisition
- Access to low-latency and near real-time data
- Reduced cost of ownership, permitting long-term retention of data in its raw, granular, form
- Deferral of work to schematize data until value is known and requirements are established

The **tradeoff** to a data lake's agility is the **additional effort** required to analyze the data via **"schema on read"** techniques, during which a data structure is defined at query time to analyze the data.

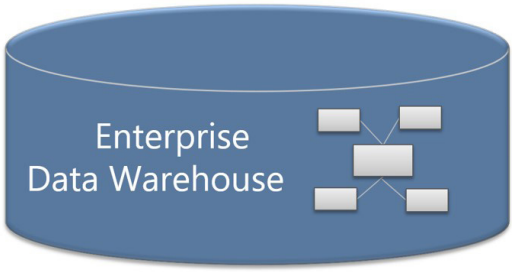
The different characteristics lead to an inverse relationship between a data lake and a data warehouse:



This inverse relationship is the precise reason why a data lake and a data warehouse are complementary solutions.

To summarize:

a data warehouse is a highly structured store of the data that the business has deemed important, while a data lake is a more organic store of data without regard for the perceived value or structure of the data.

		
Type of data stored:	Any type of data structure, any format	Structured data (most often in columns & rows in a relational database)
Best way to ingest data:	Streaming, micro-batch, or batch processes	Batch processes
Load pattern:	ELT (Extract, Load, and Transform at the time the data is needed)	ETL (Extract, Transform, then Load)
Analysis pattern:	Acquire data, analyze it, then iterate to determine its final structured form	Determine structure, acquire data, then analyze it; iterate back to change structure as needed
When structure is applied:	At query time: Schema-on-read or when “curated” data is generated	At data load time: Schema on write

1 Data Lakes
Retain Data Perpetually

During the development of a traditional data warehouse, a considerable amount of time is spent analyzing data sources, understanding business processes, profiling data, and modeling data. The result is a highly structured data model designed for reporting. Generally, if data isn't used to answer specific known questions, or required for a defined report, it may be excluded from the data warehouse. This is usually done to simplify the data model and to wisely utilize data storage.

In contrast, a data lake can acquire all types of data and retain that data perpetually – just in case. Whereas the default mode of a data warehouse is to justify the need for data before it is included, the default expectation for a data lake is to **acquire all of the data** and **retain all of the data**. Unless a firm archival policy is required, long-term data retention is justified because **future requirements and needs are unknown**.

This approach of retaining large amounts of data becomes possible because the hardware for a data lake usually differs greatly from that used for a data warehouse. Inexpensive storage allows scaling of a data lake to terabytes and petabytes fairly economically.

A data lake also can act as an **"active archive"** area in which older data which is rarely needed is transmitted from the data warehouse to the data lake. This is often referred to as hot data in the DW, and cold data in the data lake.

Bridging theory & the real world
The rule of thumb is for the data lake to ingest and retain "all" of the data. However, there are valid exceptions. For instance, perhaps you are acquiring employee data which includes personally identifiable (PII) attributes such as birth date and home address. Standard data governance and security provisions should all still apply to data stored in the data lake. GDPR Compliance may also impact the data retention and deletion policies in the data lake.

2 Data Lakes Support All Types of Data

Traditional relational data warehouses most commonly contain data extracted from transactional systems. These systems, such as sales and inventory, typically consist of quantitative metrics and the textual attributes that describe them.

Nontraditional data sources include items such as web server logs, sensor data, social network activity, text, and images. New use cases for these data types continue to be found. Storing and consuming multi-structured data in a relational database can be very challenging.

The data lake approach embraces these nontraditional data types. A data lake can store all data, regardless of source, regardless of structure, and (usually) regardless of size.

Best practices dictate that raw data be retained in its original native format. No changes should be allowed to the raw data, as it is considered immutable. It is particularly important to retain, and securely back up, raw data in its native format to ensure:

- All data which is transformed downstream from the raw data can be regenerated
- Access to the raw data is possible in select circumstances – for instance, data scientists frequently request the raw data because there has been no context applied to it
- Transformations or algorithms which adapt over time can be reprocessed, thus improving accuracy of the historical data
- Point-in-time analysis can be accomplished if data has been stored to support historical reporting

Bridging theory & the real world

Conceptually, a data lake is like the hard drive on your laptop. It stores a file, but is unconcerned with the file format. While it is true that you can drop in a file of any type, you may need to do some experimentation to find the best file format that performs well for reads & writes, gracefully handles schema changes over time, and supports the data types you need.

3 Data Lakes
Encourage Early Data Exploration

One of the chief complaints about data warehouses is how long modifications can take. A good data warehouse design can adapt to change, but because of the complexity of the data load transformation processes and the work done to make analysis and reporting easy, introducing changes will necessarily consume some DW/BI team resources.

Some business questions don't have the luxury of waiting for the data warehouse team to adapt the system. This **ever-increasing need for faster answers** is one of the main drivers for self-service business intelligence initiatives.

On the other hand, in the data lake, since all data is stored in its raw form, it could be made accessible to someone who needs access to the data quickly. **Although direct access to the raw data should be highly restricted, select users could be empowered to conduct early analysis.**

If the result of an exploration is shown to be useful and there is a desire to repeat it, then a more formal schema can be applied to the data. Data cleansing, transformations, standardization, reusability, and automation can be incorporated to extend the results to a broader audience via the data warehouse or via a curated data area of the data lake. Conversely, if the initial exploration results were not useful, they can be discarded with no additional time and effort.

Bridging theory & the real world

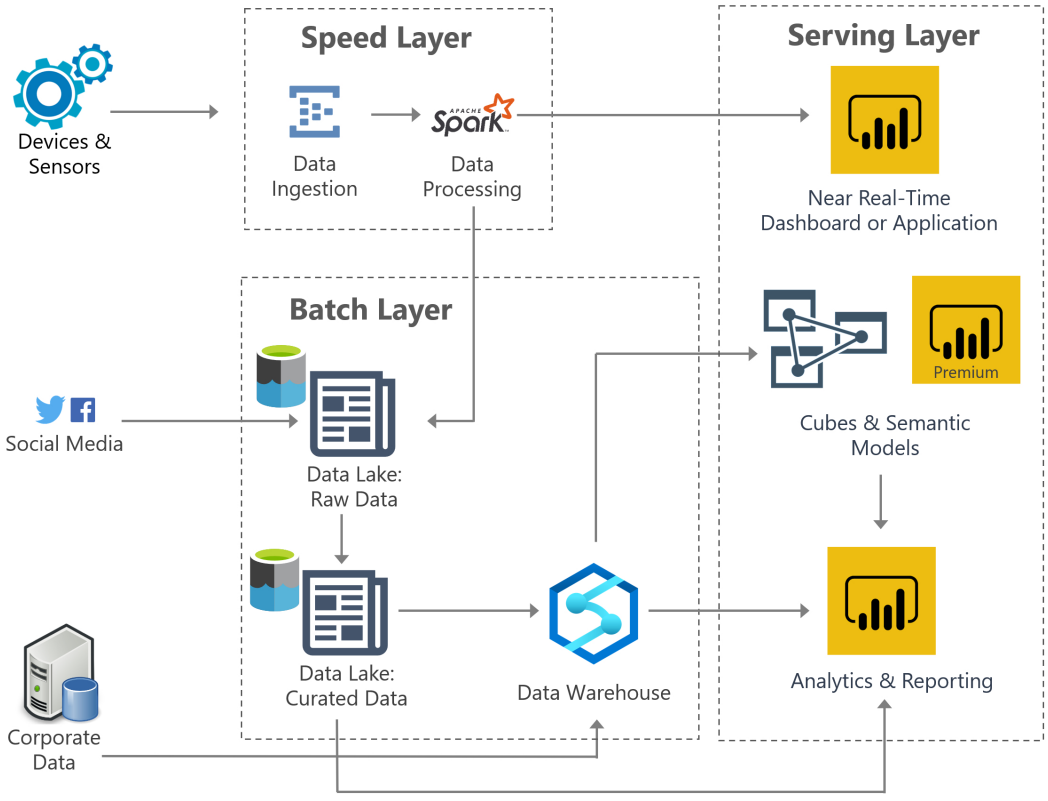
Early access to the data comes at a price. There are many challenges to working with raw "un-curated" data. Only highly adept data analysts and data scientists will want to tackle data wrangling of raw data. A compromise for the non-hard-core data analysts is to introduce a curated data area of the data lake which certain analysts are permitted to access. The curated data area contains structured data views, similar to what users are accustomed to when querying the data warehouse, with somewhat less effort to create. The curated data may be physically populated on a schedule, or may be exposed via a semantic layer.

4 Data Lakes Enable IoT & Near Real-Time Data

Acquiring and reporting on real-time data has historically been a very big challenge for data warehousing.

Data loads into a data warehouse perform best in batch mode. As data warehousing systems have scaled to larger solutions, the distributed nature of MPP (massively parallel processing) systems makes delivering near real-time data even more problematic.

The ability for a data lake to ingest data with ease brings about many more use cases, especially related to IoT (Internet of Things). A frequent pattern we see with respect to near real-time data is to transmit data to two outputs: once to a streaming dashboard or application, and once to persist the data permanently in the data lake.



Bridging theory & the real world

Although you can ingest data of any size and shape into a data lake, there are practical aspects. Since there is overhead to writing a file and setting its properties (such as security settings), Data lake file systems operate most efficiently with medium to large size files. When acquiring a large quantity of small files (as in, a few KB), consider techniques such as append-only, micro-batches, or file consolidation.

5 Data Lakes
Support Data Science & Advanced Analytics

A data lake offers easier access to multi-structured data, which has historically been difficult in a relational data warehouse, which in turn opens up new value propositions.

As we know, a data warehouse is geared towards business users with its user-friendly structuring of the data. Conversely, a data lake is more appealing to a data scientist because:

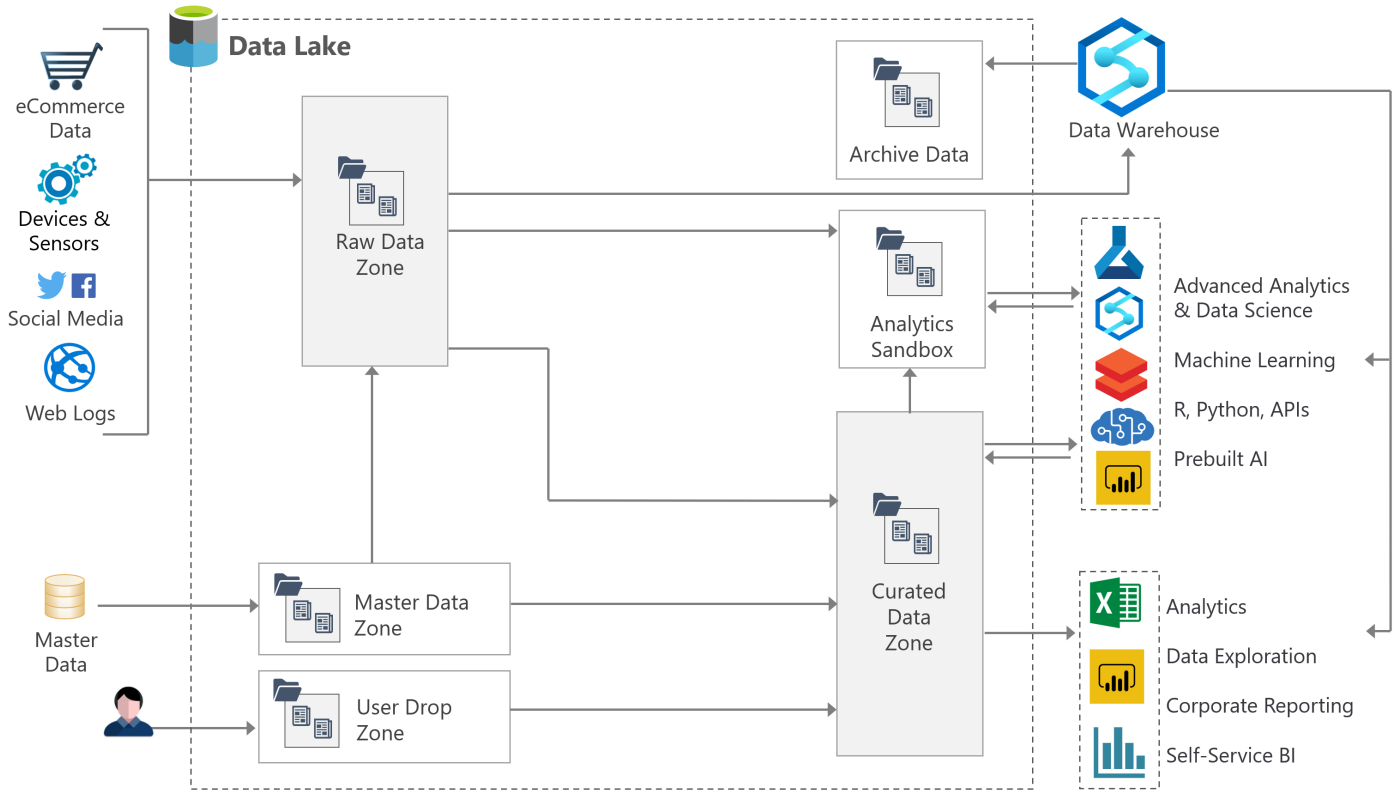
- A data lake offers access to **raw, uncleansed, untransformed data**. Although a BI professional may have spent time transforming data for usability, a data scientist frequently wants data without any context applied whatsoever. The structure can be applied specifically for the individual experiment or analysis.
- Since a data lake can capture any type of data, that makes **data access easy for multi-structured data sources** which have become commonplace.
- Modern analytical tools and languages understand how to connect to many types of data formats.

The objective for data engineers is to facilitate easy access to the data, so that data scientists and analysts spend the majority of their time running experiments and analyzing data, as opposed to acquiring and importing data.

Bridging theory & the real world
Access to the raw data brings with it additional workload and additional responsibility. When different techniques are used to structure data, varying results are sure to be found. These are lessons we have learned many times in the world of self-service BI, some of which are also relevant for data science activities.

TIPS FOR DESIGNING A DATA LAKE

The importance of purposefully organizing the data lake cannot be overstated. The concept of zones in a data lake is important. Note that zones can be physical, or can just be conceptual. The following high-level zones are commonly used:



Raw Data Zone	As the name implies, the raw data zone is storage for new data acquired, in its native format. This is an exact copy from the source, often in a normalized format, and is immutable to change. History is retained indefinitely in the raw data zone to not only satisfy future business needs, but also to regenerate downstream data whenever needed. Access by users to raw data is highly restricted. A temporary transient zone can be included and selectively utilized when data quality validations are required before the data may land in the raw data zone. It is also helpful when you temporarily need to segregate a "new data zone" separate from the "raw data zone". Alternatively, some people think of this as the speed layer, separate from the batch layer, in a Lambda architecture.
Master Data Zone	The master data zone contains master data and reference data which augments and aids analytical activities.
User Drop Zone	The user drop zone is an area where users can place data which is manually maintained.
Archive Data	The archive data zone contains aged data which has been offloaded from the data warehouse or other systems. As an active archive, the data is available for querying when needed.
Analytics Sandbox	The analytics sandbox is the workspace for data science and exploratory activities. Valuable efforts should be operationalized into the curated data zone, to ensure the analytics sandbox is not delivering production solutions.
Curated Data Zone	The curated data zone is the serving layer location for data which has been cleansed, transformed as necessary, and structured for optimal delivery. Data structures here can be large, wide, flat files, or the structure could mimic a star schema/denormalized format. Nearly all self-service data access should come from the curated data zone. Standard governance, security, and change management principles all apply.

When designing the raw data zone, focus on
optimal write performance

When designing the curated data zone, focus on
ease of data discovery and optimal data retrieval

Organizing the data

When organizing the data within the raw and curated zones, focus on considerations such as:

- Security boundaries
- Time partitioning
- Subject area
- Confidentiality level
- Probability of data access (i.e. hot vs. cold data)
- Data retention policy
- Owner/steward/subject matter expert

Utilizing data catalogs

Utilize a data catalog to provide the discoverability and exploration of data sources and reports across the enterprise with:

- Metadata
- Data tagging
- Data classification
- Data lineage

One tip is to include data lineage and relevant metadata within the actual data itself whenever possible. For instance: columns indicating the source system where the data originated.

Utilizing data catalogs

Deciding on data formats is a choice and considerations include:

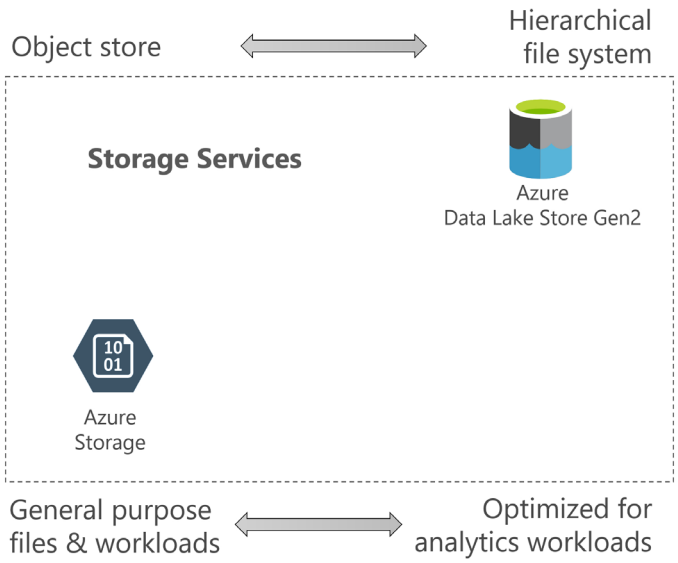
- Compatibility with upstream or downstream systems
- File sizes and compression levels
- Data types supported
- Handling of schema changes over time
- If a self-describing schema is desired
- Convenience, ease of use, human readability

Azure Technologies for Implementing a Data Lake

Data lakes are very well-suited to cloud services, such as the Microsoft Azure cloud platform. In this section we will focus primarily on the data storage and data processing layers of the architecture.

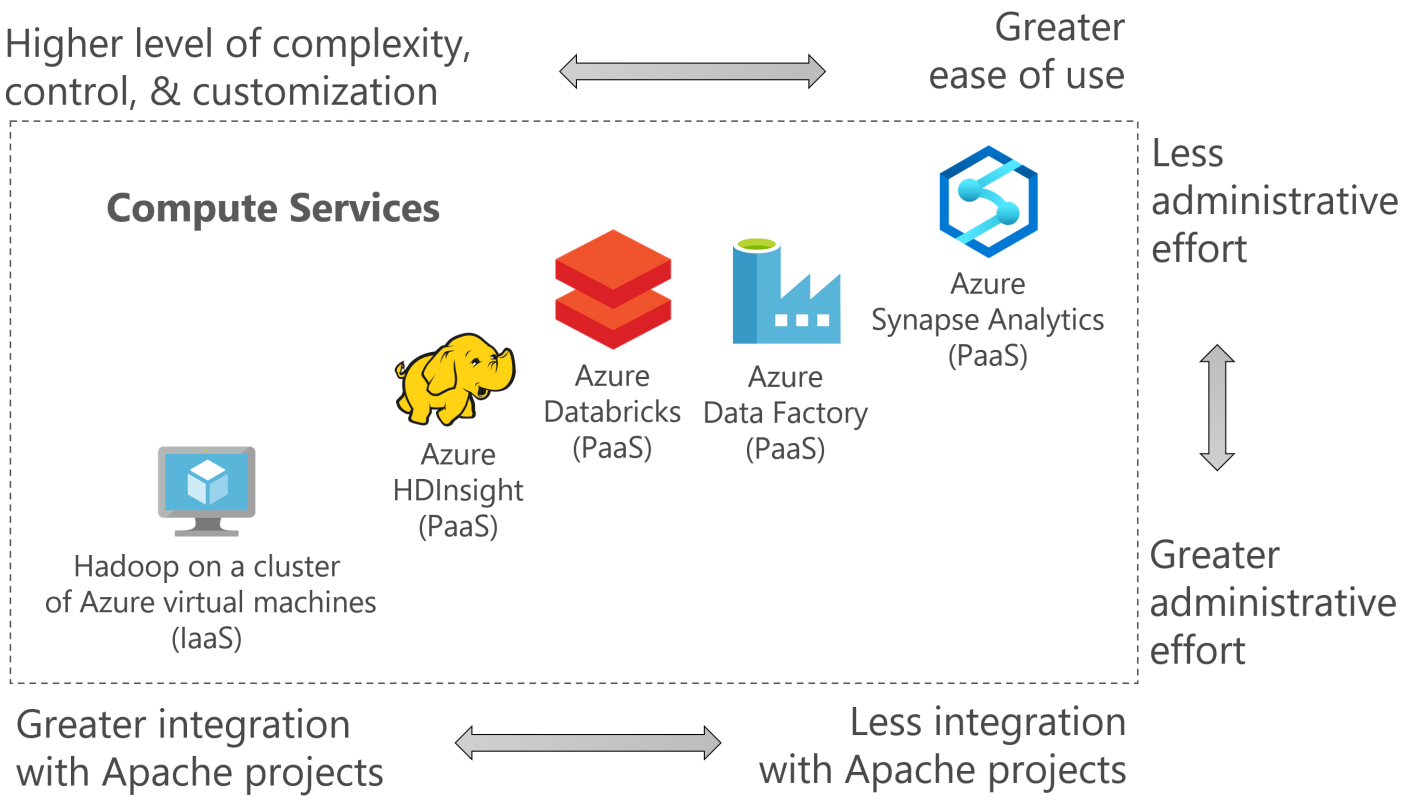
Data Lake Storage Services

When designing a data lake, the place to begin is with data storage. There are quite a few considerations when deciding which service is most appropriate for data storage. It's common to use both Azure Storage and Azure Data Lake Store for different scenarios.



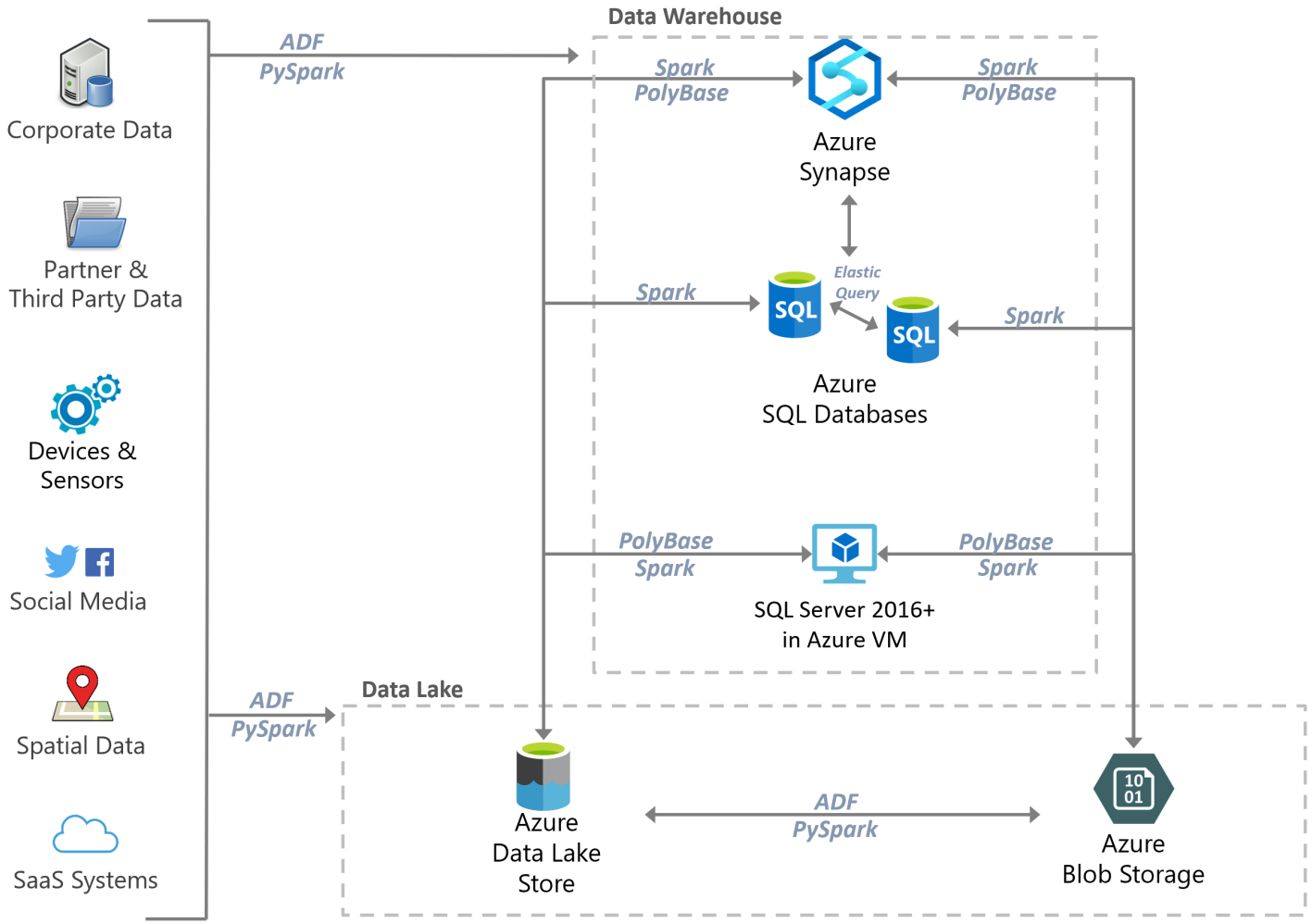
Data Lake Compute Services

Once data is available in storage, a number of compute services are available.



Since data in storage can be reused across various compute services, choosing a compute service for processing is not an either/or proposition. For instance, you may have one cluster running certain hours of the day to handle data processing operations, and another cluster running 24/7 to handle user queries. Typically, the cost for compute services significantly exceed the cost for data storage.

The data in the data lake can also be used in conjunction with your data warehouse. This could be via a full-fledged third-party data virtualization provider. There are also numerous ways to execute remote federated queries between the data lake and relational databases:



Considerations for a Successful Data Lake in the Cloud

Cloud-based services, such as Microsoft Azure, have become the most common choice for new data lake deployments. Cloud service providers allow organizations to avoid the cost and hassle of managing an on-premises data center by moving storage, compute, and networking to hosted solutions.

A modern data architecture often involves a wide variety of technologies which would require a large up-front investment in an on-premises data center. Employing cloud technologies translates costs to a subscription-based model which requires much less up-front investment of both cost and effort. Cloud services also offer many other advantages such as ease of provisioning, elasticity, scalability, and reduced administration.

Th following are some specific considerations when planning a data lake deployment on a cloud service:

Type of storage

A data lake is a conceptual data architecture, and not a specific technology. The technical implementation can vary, which means different types of storage and storage features can be utilized.

The most common data lake implementations utilize:

- HDFS (Hadoop Distributed File System)
- Propriety distributed file systems with HDFS compatibility (ex: Azure Data Lake Store)
- Object storage (ex: Azure Blob Storage or Amazon S3)

The following options for a data lake are less commonly used due to greatly reduced flexibility:

- Relational databases (ex: SQL Server, Azure SQL Database, Azure SQL Data Warehouse)
- NoSQL databases (ex: Azure CosmosDB)

One data lake vs. multiple lakes

Organizations have always sought “one version of the truth.” One data lake which contains all organizational data is sometimes a goal, particularly if an objective is integration of older, legacy systems with newer types of data. In reality, many organizations end up with multiple data lakes or document stores either unintentionally or intentionally (to segregate business units, for instance). Tasks like security, auditing, and sharing of objects/data are all important considerations when deciding on a multi-lake environment. Some level of data duplication is tolerated in multi-lake environments.

Security Capabilities

Different technology platforms implement security differently. A service such as Azure Data Lake Store implements hierarchical security based on access control lists, whereas Azure Blob Storage implements key-based security.

Data cataloging, metadata, and tagging

A data catalog is a crucial component for facilitating data discovery for authors of self-service solutions. A well-designed data catalog acts not only as a data dictionary, but also assists with data previews, data profiling, and data access. Clear documentation, metadata, tagging, and data classification capabilities are vital for users to be able to effectively use data in a data lake. If users don’t understand the data provided, they won’t use it and/or they’ll create their own siloed datasets, diminishing the value of the data lake architecture.

Elasticity

One of the most powerful features of cloud-based deployments is elasticity, which refers to scaling resources up or down depending on demand. This equates to cost savings when processing power isn't needed. The services in Azure Data Lake are decoupled with respect to compute and storage, which provides independent scalability and flexibility.

Client tool access

One of the key success criteria for an analytics deployment is user adoption. One aspect of driving user adoption is the flexibility of client tool access to curated data. A successful data lake deployment has data access occurring from automated systems (like the data warehouse loads), as well as from tools such as Excel, Power BI, and data science notebooks.

Generic compute service integration

The data lake should be able to serve as a robust storage for many compute services such as Databricks, Azure Data Factory, and Synapse through native integration or an API. Although you may not use every compute service, having multiple services that are compatible will provide the flexibility to select the best option and grow with future demands.

Data warehouse integration

For most companies, a data lake vs. data warehouse is not an either/or decision. Rather, a blended approach is typically most effective. Most commonly we see two types of data lake integration with the data warehouse:

- Processes which pick up and move the data from the data lake to the data warehouse (or vice versa)
- The ability to issue federated queries which returns data from both your data lake and your data warehouse via a single query

Disaster recovery

The most critical data from a disaster recovery standpoint is your raw data (because, in theory, your curated data views should be reproducible at any time from the raw data). Data center outages are certainly not a common occurrence, but you do need to pre-plan and test how to handle a situation when acquiring raw data is business critical. The ability to recover your data after a damaging weather event, system error, or human error is crucial.

Getting Started with a Data Lake

Confirm a data lake really is the best choice

Make sure that your data and use cases truly are well-suited to a data lake. Ignore the rampant marketing hype. Continue using your relational database technologies for what they are good at, while introducing a data lake to your multi-platform architecture when it becomes appropriate. The existence of many types of data sources, in varying formats, is usually a great indicator of a data lake use case. Conversely, if all of your data sources are relational, extracting that data into a data lake may not be the best option unless you are looking to house point-in-time history.

Start with a small, practical project

A few of the easier ways to get started with a data lake include:

- **Begin storing new "unusual" datasets.** This affords time to do longer-term planning and learning while beginning to accumulate data.
- **Data warehouse staging area.** By relocating the data warehouse staging area to the data lake, you can reduce storage needs in the relational platform and/or take advantage of big data processing tools (such as Spark, for instance). This works particularly well for low-latency datasets.
- **Data warehouse archival area.** Data which ages past a date threshold can be offloaded to the data lake for historical retention. It can still be available for querying, typically via data virtualization techniques.

Address 'readiness' considerations

Be prepared to handle the trade-offs of schema on read vs. schema on write solutions. Remember, structure needs to be applied at some point so all decisions become a "pay me now or pay me later" proposition - which requires the right balance for your use cases. Additionally, you can expect to introduce new technologies and new development patterns.

Do a POC whenever possible

Conducting a technical proof of concept (POC) is something you should become accustomed to doing routinely when introducing new features, new technologies, or new processes. POCs can also be referred to as sandbox solutions in a data lake. Features and functionality change rapidly in cloud services, so a POC can help you be agile and learn what needs to be done for the full-fledged solution. There is always a risk that a POC is pushed into being production-ready too quickly; therefore, factor in the effort for operationalizing POC/prototype/sandbox efforts.

Don't shortchange planning

Although agility and less up-front work are touted as advantages of a data lake, that does not mean there is no up-front planning. You absolutely must manage, plan, organize, secure, and catalog a data lake so that it doesn't turn into the dreaded "data swamp." Techniques for how to ingest raw data quickly, and how to retrieve curated data effectively, are iterative in nature.

Implement the right level discipline

Finding the right balance of agility (to get things done today) versus process and rigor (to implement a sound, maintainable, extensible architecture) is a challenge. The answer isn't the same for every company. To the extent that more schema on read approaches become prevalent, the need for governance and data cataloging are just as important as they always been, perhaps even more so.

BLUEGRANITE

DATA • INSIGHTS • ANALYTICS

We hope you have found this eBook useful. If you are looking to bring in new approaches, combined with proven techniques, to support decision making at all levels of your organization, let BlueGranite help.

Contact us today for a free consultation!