DesignMind

Data Lake Based Systems that Work

There are many article and blogs about what works and what does not work when trying to build out a data lake and reporting system. At DesignMind, we have developed a pattern that not only ingests large amounts of data, but:

1. *Makes data available to users at all levels of the system*
2. *Allows data to be accessed by multiple formats*
3. *Allows for simplified schema evolution management*

Components of a Lake

Most references to data lakes refer to them as a total data storage and reporting system. This is categorically untrue, as a data lake is a single part of a complete data system. At DesignMind we think of a Data Lake Based System as a series of expandable components that work together to ingest, process, and make data available to other systems. DesignMind's Lake Based System approach is built out of six pieces:

1. Data Ingestion System
2. The Data Lake
3. Data Processing Systems
4. A Data Access Layer
5. Metadata Management System
6. An Enterprise Scheduler

Each piece is a standalone system that communicates across the system with all other relevant components. This provides an unprecedented amount of flexibility and expandability as each of the systems can not only be built out of different components, but can be updated in the future with no change to any of the other components.

About DesignMind

DesignMind is a Big Data, Business Intelligence, and Data Analytics consulting company headquartered in San Francisco.

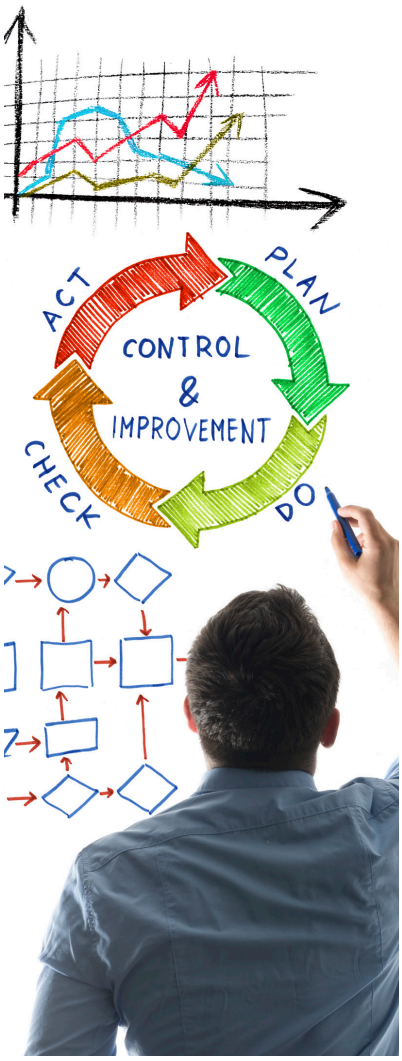
Data Ingestion System

DesignMind's Data Ingestion System is designed to take in a diverse set of data from multiple inputs and insert this raw data into the Data Lake. The system is built out of a set of Ingesters that parse data to a message bus for minimal cleaning and insertion into the Data Lake.

The Ingesters are custom routines that take a single format of data and parse that into a well-defined object. An individual Ingester takes a set of arguments that define the expected data contained within the data feed and format. This allows an ingester to be used for multiple feeds with only changes to the passed arguments, not the code itself.

The message bus then takes the output from the Ingesters and feeds this data to a set of end points. These end points are usually inserts into the Data Lake, but can simultaneously be pushed to other systems for real time monitoring or other data uses. DesignMind is familiar with message systems like Apache Kafka and TIBCO EMS, but a range of others can be used as well.

Finally, within this original ingestion of the data a minimal amount of data sanitization is required. This is to ensure that the data being passed to further systems does not inadvertently corrupt the system. These changes can range from innocuous problems like character escaping, to checking for injection attacks. What needs to be kept in mind is that this data cleaning should be as minimal as possible.



Data Lake

DesignMind's Data Lake is the main data store for the Data Lake Based System. The Data Lake is not required to be specific system of storage, e.g. Hadoop or NoSQL. But should be decided upon by the requirements of the system and familiarity of the group using the system.

There are five different types of storage systems that can be used for a Data Lake. Each of them has its own pros and cons.

DesignMind's preference is to use Hadoop Based systems for the Data Lake component. With the use of query and transformation tools listed in the processing system, it is usually the most cost effective path to data storage.

Data Storage System	Pros	Cons
Hadoop Based System	Easily expandable and cheaper storage	Slower data retrieval times
Non-Hadoop Based Storage + Hadoop / non-Hadoop Compute, e.g. S3 + Hive / Spark	Decouples storage and compute, optimized for cloud platforms	More difficult to implement on-prem
Massively Parallel Processing System (MPP), e.g. H.P. Vertica or IBM Netezza	Fast record retrieval and ease of setup	High cost
NoSQL System (Cassandra, HBase)	Easily expandable and fast	Less familiarity with NoSQL systems with the community
SQL Database (SQL Server, Oracle, MySQL)	Well defined technology	Cannot handle large amounts of data without high cost

The Data Processing System

Once raw data is within the Data Lake, it is normally transformed via ETL (Extract, Transform, and Load) and ELT (Extract, Load, and Transform) processes. These transformations can be run via a native query language or via secondary scripting languages, depending upon the type of Data Lake that is chosen.

Within an SQL, NoSQL, or MPP based Data Lake most data transformations will take place within the native query language. It is possible to also write custom code with connectors to the database of choice, but this can be brittle and difficult to maintain.

While within a Hadoop based Data Lake, there are multiple ways that the contained data can be processed. DesignMind has expertise in multiple approaches and suggests (in descending order):

1. APACHE HIVE:
A data warehouse for Hadoop that allows anyone with SQL experience to query data stored within the Hadoop system.

2. APACHE PIG OR OTHER HIGH LEVEL SCRIPTING LANGUAGE:
Pig and other scripting languages allow querying of Hadoop based data with more functionality, but have a higher barrier to entry.

3. CUSTOM SPARK OR MAPREDUCE CODE:
Custom code to run on a Hadoop cluster can have the highest efficiency, but can be brittle and difficult to maintain.

Beyond the native query language and higher level scripting languages, there are systems useable for easing the transformation of data in specific cases. One of these is Profisee Maestro, a tool that eases the normalization and mastering of data within your system.

Finally, the transformations being made are usually to take the data from a raw format to one more informative to end users. These transformations usually consist of data reduction, coding of business logic/rules, and data normalization. There are no strict standards on how these processes happen as each data set contains its own nuances, but there are various patterns to get the most understanding from your data in the shortest time.

"This system extensibility makes for faster and easier changes to data importing, processing, and availability."

Data Access Layer

Once the raw data has been processed, it can be made available to users. Usually this takes the form of an ODBC connector (e.g. for Microsoft Power BI or Tableau) or data export. All four of the options in the Data Lake section can expose the data in these two ways.

Depending upon which of the four Data Lake options chosen, and the amount of data currently ingested, queries against your data lake might be too slow for the average user. In this case DesignMind suggests a Data Warehouse sitting between the Data Lake and the Access Layer. This creates a smaller partition of hot data in front of the warm data lake, for faster query times.

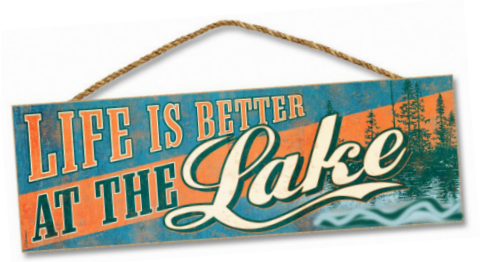
Due to the size of the Data Warehouse being a few percent of the Data Lake, this can usually be created in a standard SQL Database. This can also be accomplished by using low-latency SQL query engines for Hadoop, like Cloudera Impala, Apache Drill on a Hadoop Cluster, or creating a new warehouse only schema if using an MPP or NoSQL approach.

Metadata Management System

The key differentiator of DesignMind's Data Lake Based System is its Metadata Management System. DesignMind's approach is to keep all processing and schema definitions within a separate database. This allows for the creation and modification of any schema by just changing the definition with the Metadata System. Beyond that, all Ingesters, data flows, and any other processes can be defined by system settings stored within the metadata tables.

This allows for simplification of schema migration and data lineage:

- When the definition of a table changes, one no longer needs to worry about lineage. Both the new table schema, and all previous table schemas, are kept within the Metadata Management System.
- When trying to define the conditions that data was processed with on a certain run, the data can be found by querying the active conditions at that time.
- For a new data source with an existing ingester, all that is needed to import the data is an addition to the Metadata Management System.



This system extensibility makes for faster and easier changes to data importing, processing, and availability.

Scheduling

The first five pieces of DesignMind's Data Lake Based System define a framework where data can be ingested, processed, and made available to other systems in a repeatable and easily expandable manner. Still missing is an Enterprise Level Scheduler that will manage running of the Data Lake Based System.

There are many open and closed source options for the scheduler. DesignMind has had success implementing the following schedulers:

- Airbnb Airflow
- Apache Oozie
- Cisco Tidal
- LinkedIn Azkaban

As noted in previous sections, there are many packages that can be used to build out this piece of the system. If your organization already uses another Enterprise Scheduler, it can be easily incorporated into the Data Lake Based System.

Andrew Eichenbaum is VP, Data Science and Mark Kidwell is VP, Big Data at DesignMind in San Francisco.

If you would like to learn more about DesignMind's Data Lake Based System, or provide us with feedback, please reach out to us at info@designmind.com or at 415-538-8484. We would be more than happy to discuss your needs and how we can help.