data

ADVANCING ANALYTICS

Lighthouse

Walmart Luminate Technical Implementation

databricks partner

Microsoft Partner

ai

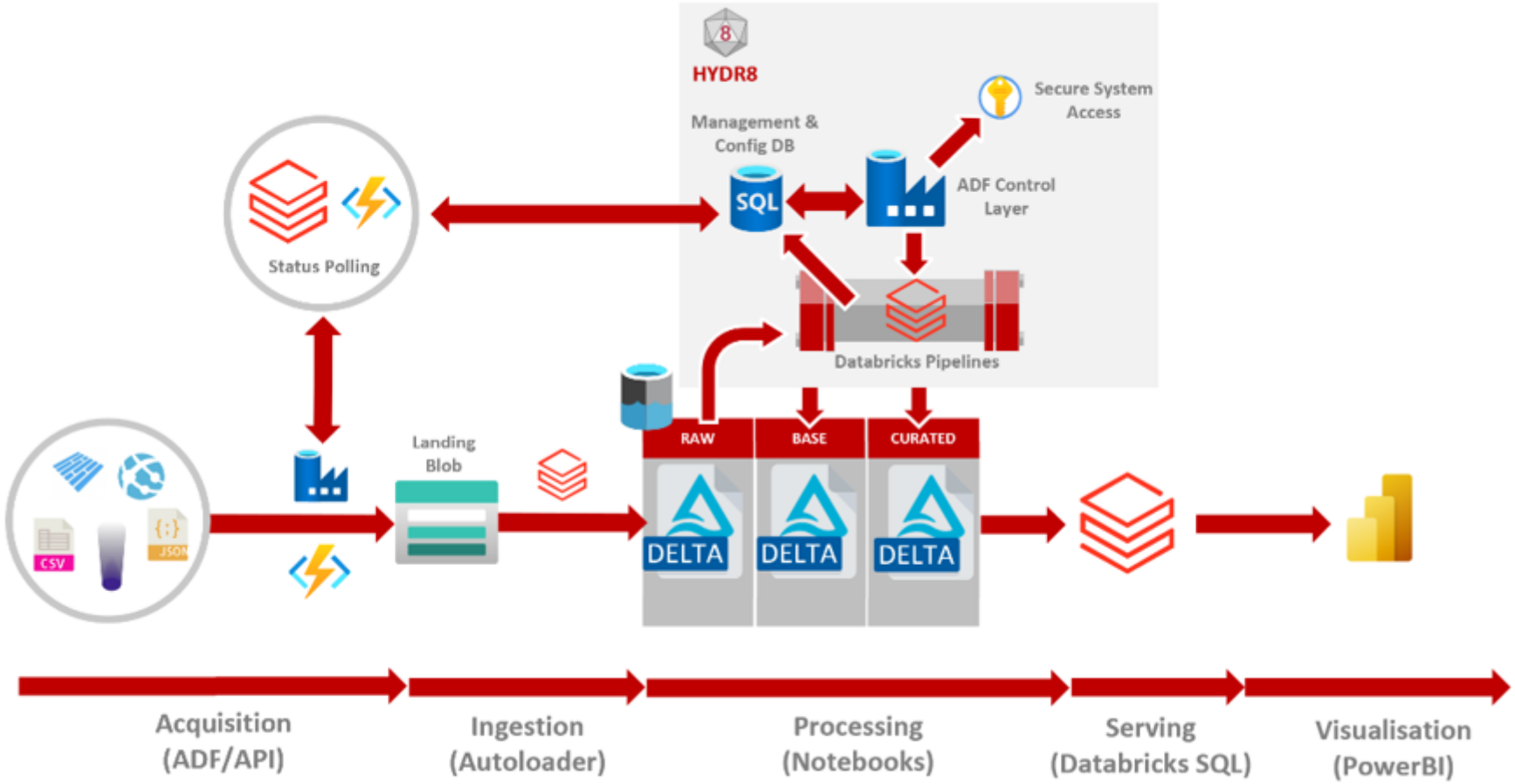
PROJECT LIGHTHOUSE ARCHITECTURE

Description

Lighthouse is a Data Lakehouse in a box, designed specifically to leverage Luminate data. Secure by design, you can be sure your data is in safe hands, and it's scalable to grow as you need.

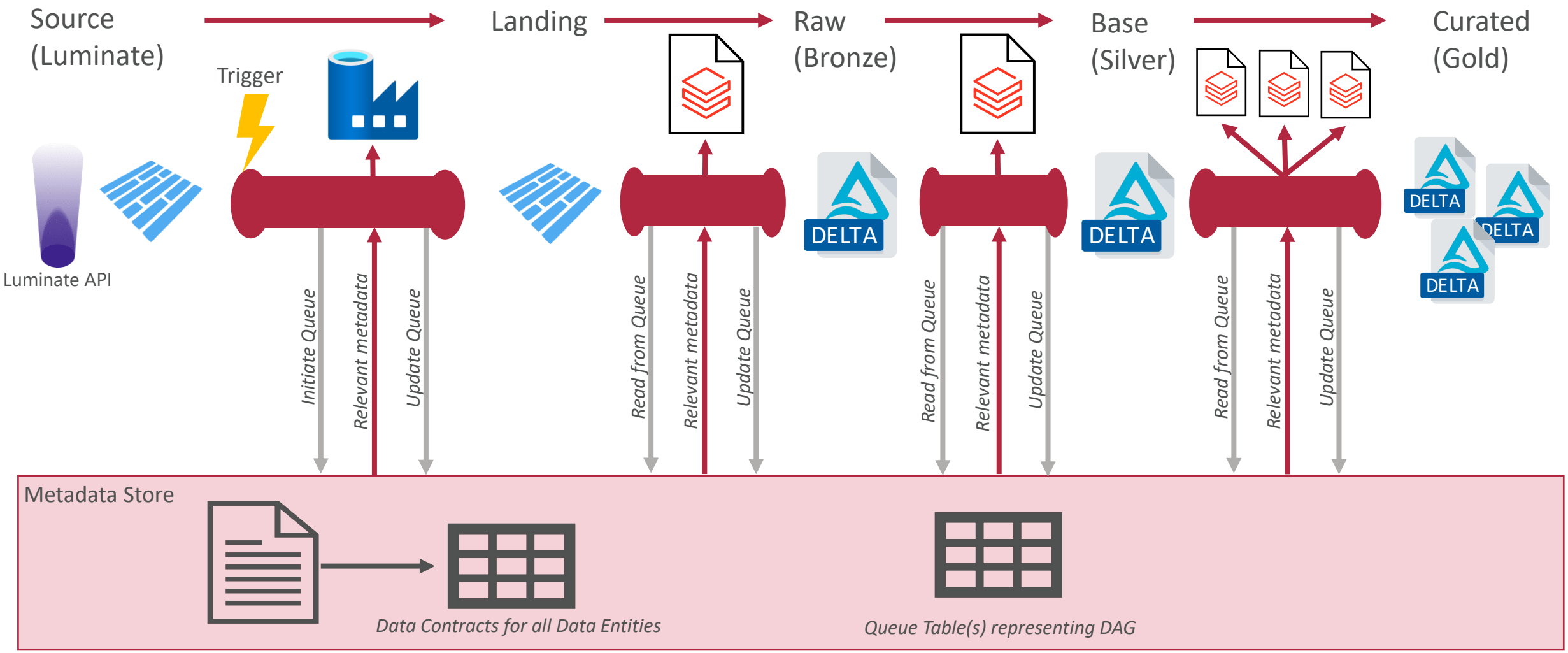
Metadata driven to rapidly respond to new and changing endpoints, Lighthouse enables you to efficiently process restatements, whilst being fully integrated with Unity Catalog.

Lighthouse is deployed into **your** environment, and the data and platform are completely owned and controlled by you.



METADATA-DRIVEN SOLUTION

Repeatable process for each Luminate entity. Build once and reuse.



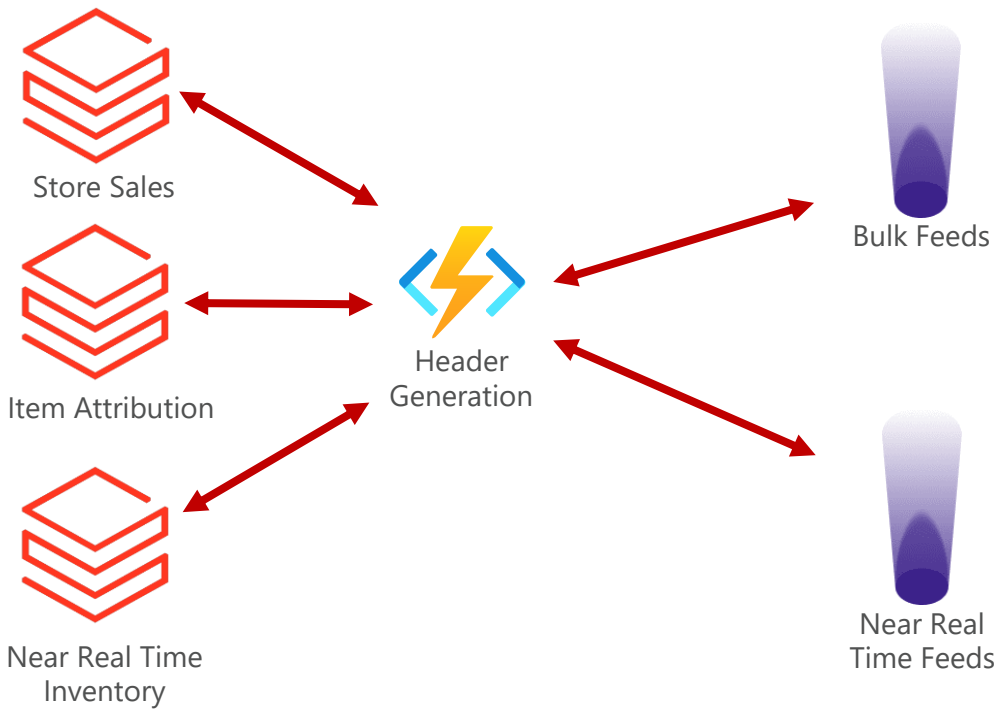
LIGHTHOUSE - SCALABLE AUTHENTICATION

In order to interact with the endpoints in Luminate, requests must be authenticated.

Lighthouse comes ready with a centralised and scalable authentication app, which can be used to authenticate concurrent requests.

This central application is a one stop shop for seamlessly integrating with all of Luminate's data services, including Shopper Behavior, Channel Performance, and Customer Perception.

Can support multiple consumer id's to enable both supplier and category authentication using a single app.

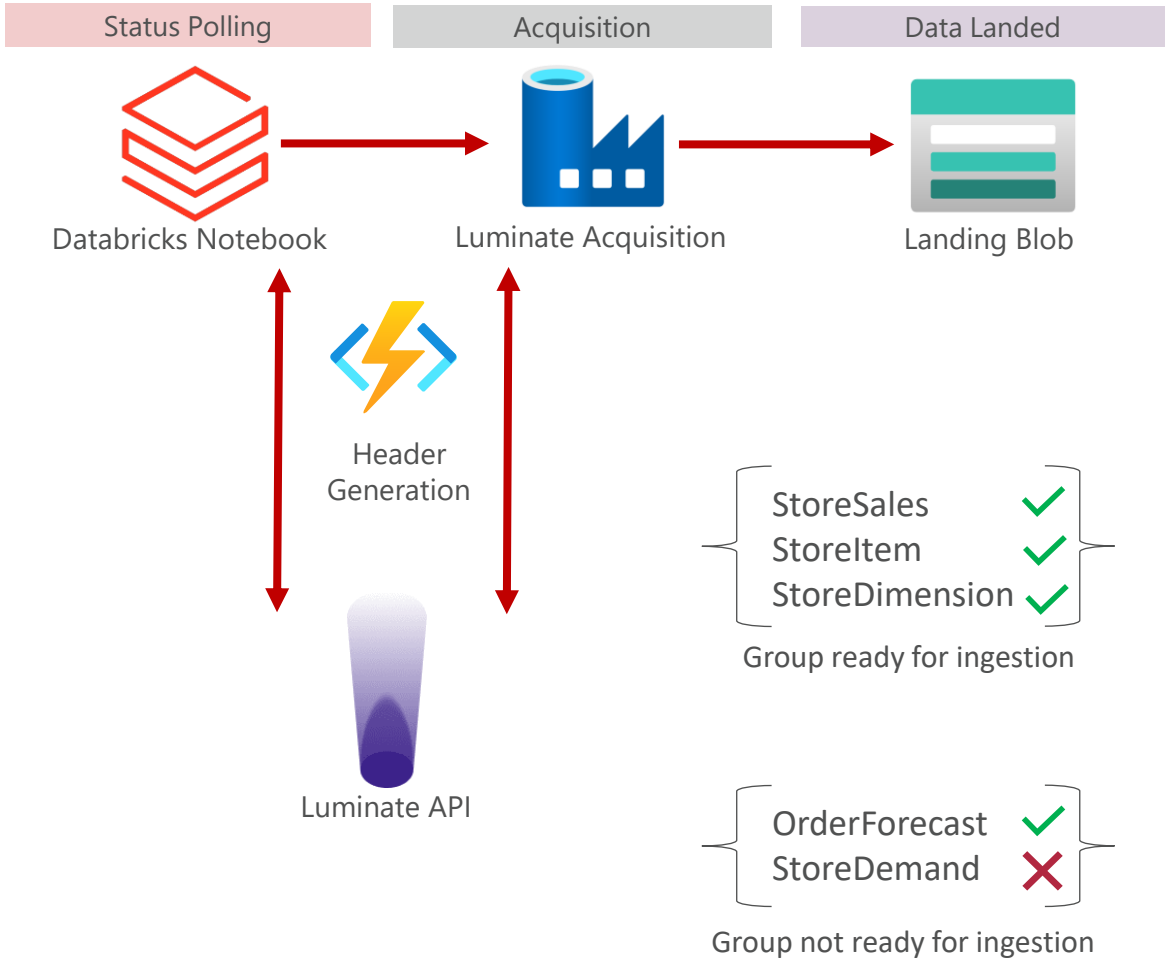


LIGHTHOUSE – AVAILABILITY MARKING

When an endpoint is ready to be ingested, it'll be marked as **available** in the **Status** endpoint. This can be at any time per day prior to the signed SLA.

To maintain maximum flexibility, datasets are grouped and ingested in batches, which are polled and then executed when ready. This will ensure that business analytics is not held up by datasets that are not contributing to the dashboard, export or application.

Availability and ingestions are logged to a central location to ensure files are ingested exactly once, missed files are identified and errors can be rapidly recovered from. Files are made available from Luminate for 45 days.



LIGHTHOUSE – IRREGULAR CADENCES

Typical Availability

- Whilst most endpoints are made available daily, there are some exceptions.
- Some endpoints are one-time-ingestion (like the Calendar entity).
- Some other endpoints are available weekly, usually on a Saturday or Sunday.

Bulk Restatements

- Restatements are typically handled within the standard daily or weekly processes, however, for high volume restatements or major schema changes, the data is made available via the history endpoint.
- This enables consumers to more appropriately time their ingestion of this data to prepare in advance, whether that be using larger computes, alternative processing patterns or modifying schemas and table constraints to match the incoming data.

Requested Regeneration

- If a support ticket is raised to regenerate data for a period of time, this will also be made available through the history endpoint.

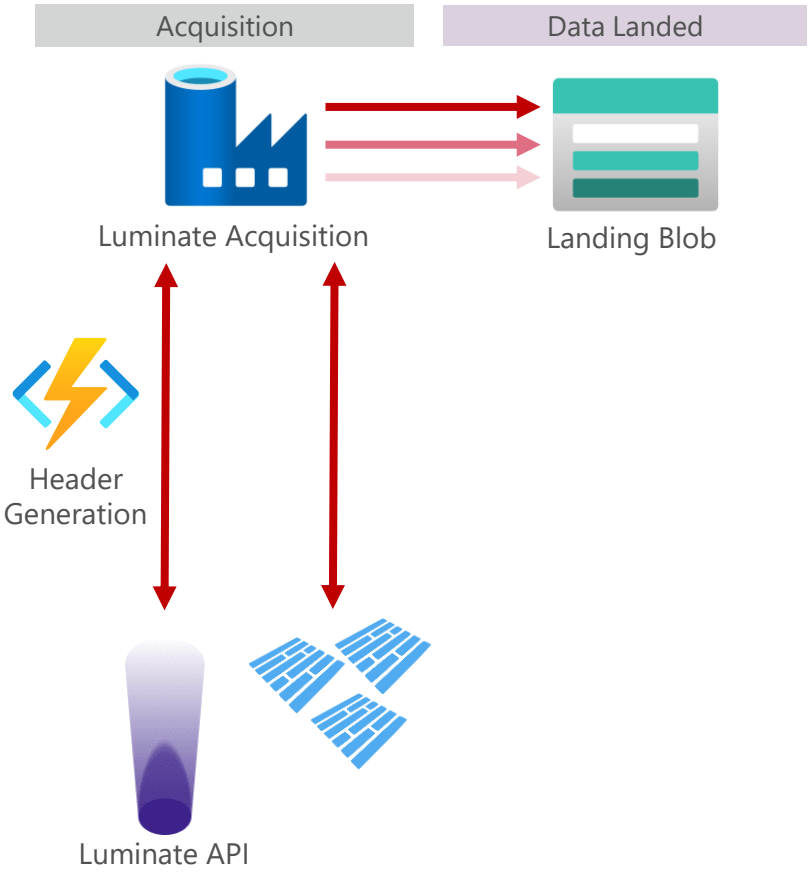
LUMINATE – INGESTION

Upon onboarding, history is ingested for each endpoint to bring vendors up to date with existing datasets.

Following this, datasets are ready for daily, incremental ingestion. Once a batch of endpoints are ready for ingestion, we re-authenticate with Luminate and then send a request to the target endpoint.

The endpoint will return one to many signed URLs to ingest the relevant parquet files.

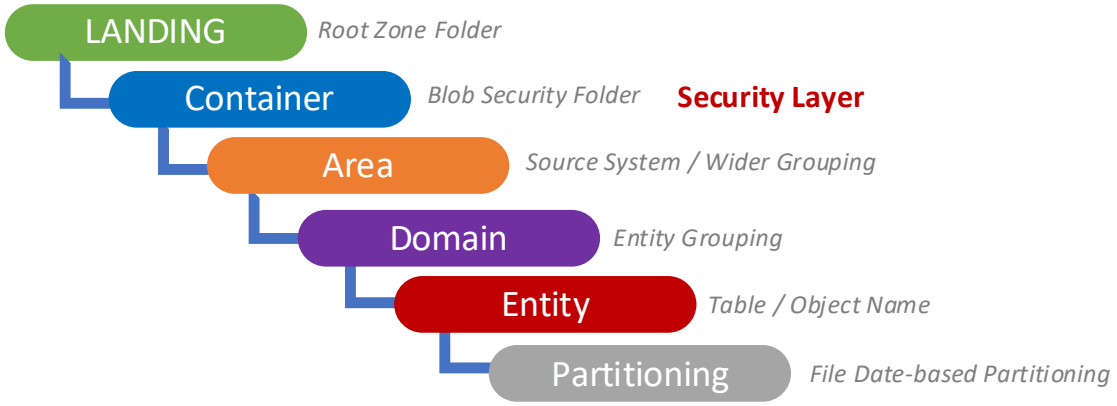
Ingestions occur in parallel per group of endpoints and persist the datasets into **LANDING**. Upon ingestion, the events are logged for auditability.

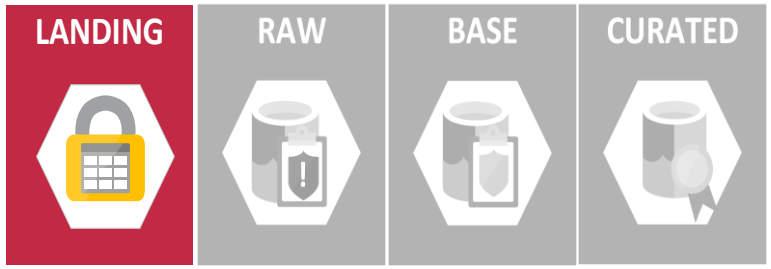


LIGHTHOUSE – INGESTION – TECHNICAL SUMMARY

This pattern of ingestion is driven by the value it provides to the business:

- Re-usable authentication patterns to rapidly onboard new, interesting datasets made available by Luminate
- Logging daily availability times and ingestion processes enables the business to monitor daily runs, analyze average times of availability and identify missing files, encouraging transparency and enabling the business to reliably set user expectations
- Ingest exactly-once optimizes the storage layer to reduce duplicate data and ensure that data is not missed, reducing storage costs
- Metadata driven configuration of entities allows the business to prioritize critical daily datasets and quickly respond to changes in endpoint cadence
- Making use of repeatable patterns also massively speeds up the development time required to introduce new datasets into the platform, enabling Engineers, Analysts and Data Scientists to access new data without delay





Landing Layer: Staging point for landing new datasets before ingestion

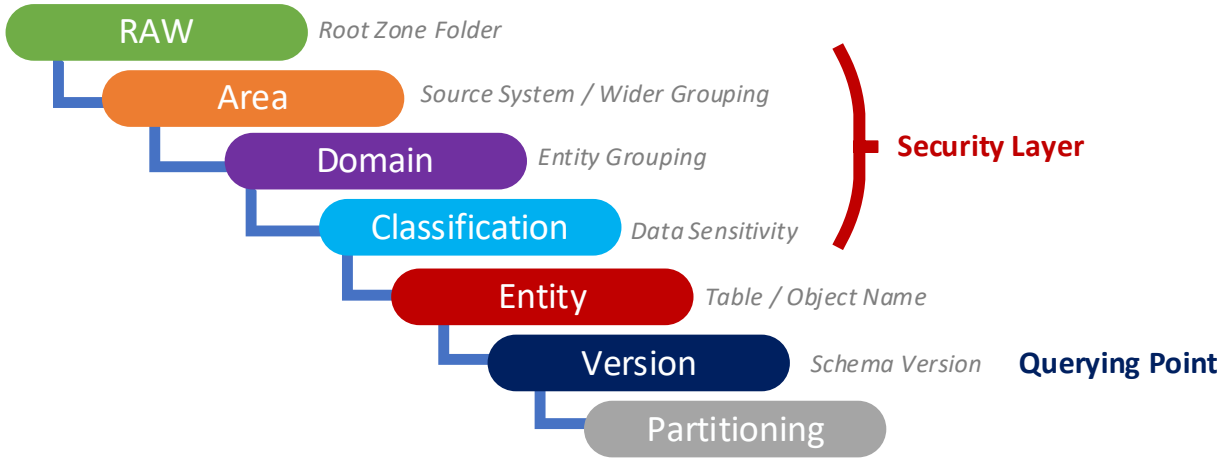
Core Usage: This area provides one or more blob store accounts where third parties (internal or external) can persist new datasets for ingestion into the data platform. Data will be retained in an archive state after loading for disaster recovery purposes

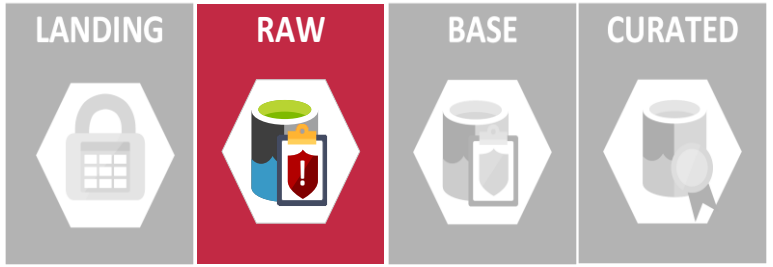
Data Standards: No fixed file format – data will land in many different forms and standards. Data is partitioned by received date.

Key Processes: History retention, initial data collection

Outbound Processes: AutoLoader process watches the landing area for new datasets, these files are automatically ingested into the RAW zone at the next loading interval

Maintenance: As is it not held in optimal, compressed state, and rarely accessed once processed through to RAW, the data in landing can, optionally, be regularly archived to a separate blob store using the cold access tier.





Raw Layer: Initial Staging of Data for Validation & Inspection

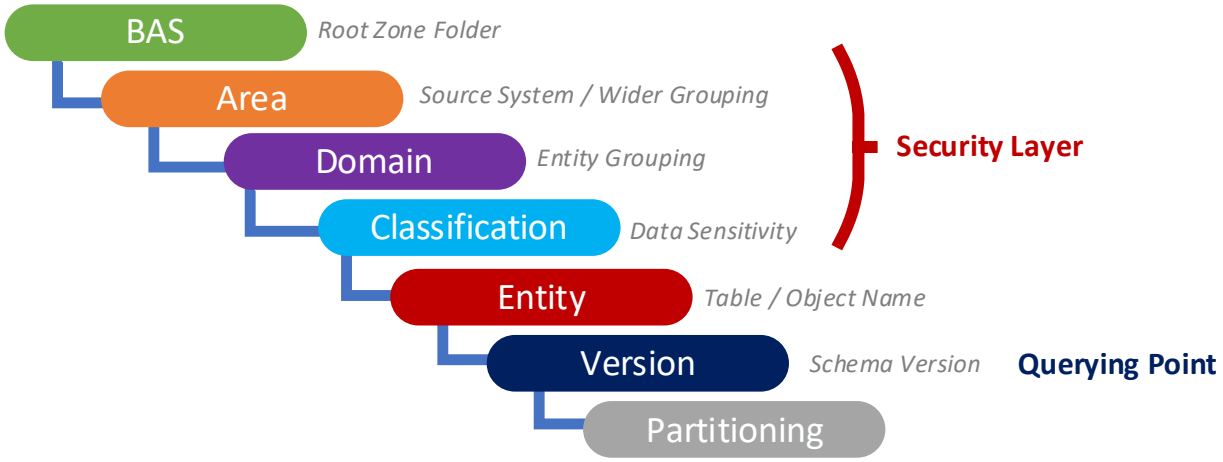
Core Usage: This area acts as the original copy of all data received, used for troubleshooting, investigation & reprocessing. There is deliberately no validation on incoming data to ensure data is always collected – we do not want to go back to source systems to re-query data where possible.

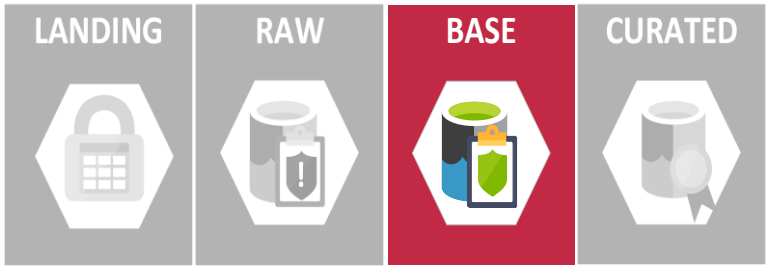
Data Standards: Data is persisted as a Databricks Delta table and makes of the schema management capabilities offered by Autoloader. Schemas can be set to auto-evolve or remain static, capturing unexpected or erroneous columns inside the rescue column for validation and reparation.

Key Processes: Historical archiving, initial data collection, sensitive data obfuscation, troubleshooting & application of auditing columns

Outbound Processes: Data from RAW is validated and cleaned, applying various rules such as standardizing dates, trimming strings, removing special characters. This data is landed in the BASE zone once clean.

Maintenance: As is it not held in optimal, compressed state, and rarely accessed once processed through to BAS, the data in RAW can be regularly archived to a cold tier, then eventually physically deleted. This can be determined on a dataset basis.





Base Layer: Initial Staging of Data for Validation & Inspection

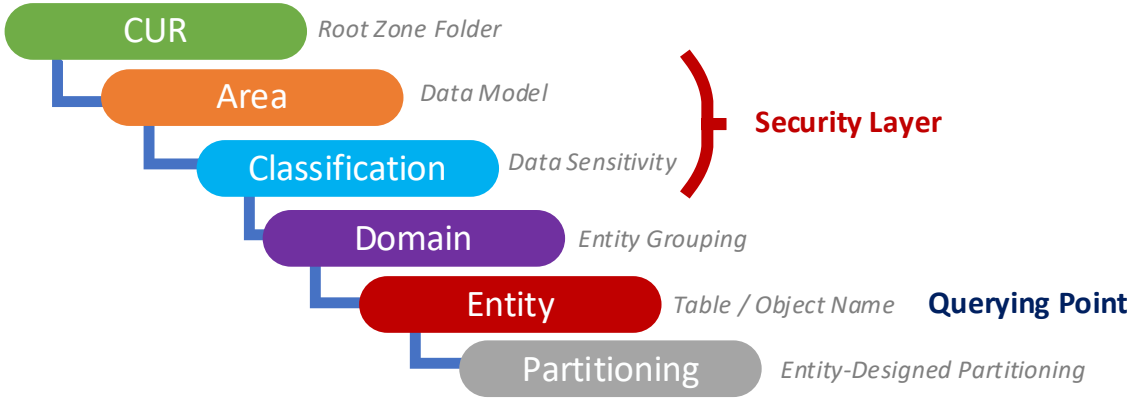
Core Usage: A one-to-one copy of RAW, with validation, lineage and cleansing applied. This layer benefits from configuration-based automation which forms part of Advancing Analytics Hydr8 framework. This greatly reduces the effort in implementing this validation and cleansing layer. Data in this layer is cleaned and standardised

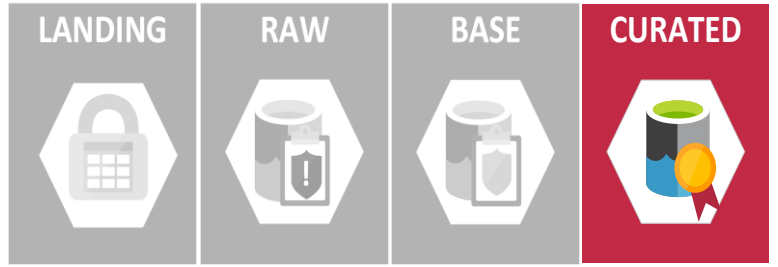
Data Standards: Data is persisted as a Databricks Delta table. This ensures that we can replay files from a single format and take advantage of the Columnar format discussed later. Files can be loaded using a merge statement to handle CDC-style updates.

Key Processes: Clean Data, Validated Data, Trusted Data Source, Supports Data Science Use Cases.

Outbound Processes: Data from BAS will be combined with other data sources and persisted to the CUR layer. This is usually performed either in period recalculation (ie: update entire day/month partitions in CUR) or we can determine a CDC table using our auditing columns

Maintenance: The Delta tables in this layer need regular maintenance – optimizing for file compaction, vacuuming obsolete history and archiving historical data where necessary.





Curated Layer: Staging point for landing new datasets before ingestion

Core Usage: This layer is where we store transformed and enriched data. Data from other external sources may be combined with core data sets in the BAS layer to provide extra dimensions and attributes needed for the final consumers of this data

Data Standards: Delta (unless used for explicit export to other platforms)

Key Processes: Delta calculation, transformation and augmentation. Ad-hoc querying & analysis by the business.

Outbound Processes: Data from CUR will go to be used for a variety of different functions. This could be served via dashboards, ingested into data marts, used directly in machine learning models or simply analysed directly. There may be several nested layers of dependency – where curated objects are derived using other curated objects as a data source

Maintenance: This layer requires the most management from a performance point of view as it will directly affect the data lake end users. Tables must be maintained, optimized and vacuumed.

LUMINATE – DELTA FLAGS AND FEED DATES

Displaying the most up to date datasets is critical in ensuring the data is reliable and accurate

Luminate handles this with two key mechanisms, the Delta Flag and the Feed Date

The Delta flag indicates a new record (I – inserted), a changed record (U – Updated) or a deleted record (D – Deleted). Tracking these changes over time, and processing them in the correct order, is required to produce the latest version of the data.

Changes to records can come through many times, and the Feed Date (The date the file was made available for ingestion) is what indicates the order that these should be processed in.

The latest feed date will always take priority, and if records share the same feed date, deletes should take priority over updates, and updates take priority over inserts.

LUMINATE – PROCESSING - SNAPSHOT

Examples: ItemAttributes
StoreDimensions
Calendar

Snapshot data does not provide history. Daily snapshot files are ingested into landing.

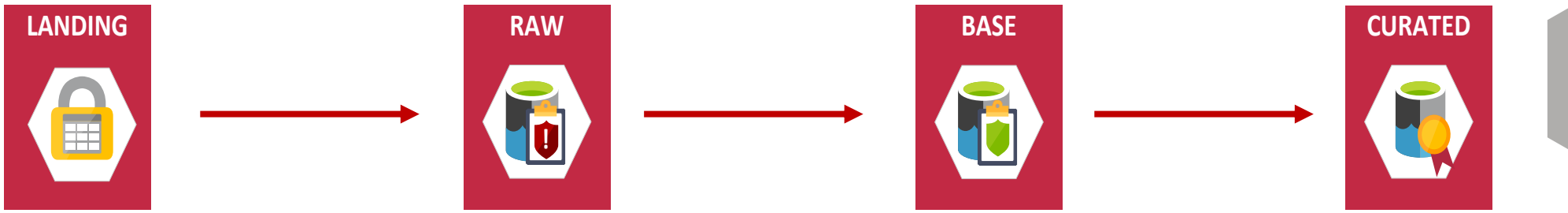
Snapshot data is appended into Raw for full traceability.

Snapshot data is either overwritten, merged into, or maintained as slowly changing dimensions in Base, depending on the analytical and historical requirements. A single record per primary key will exist in Base.

Curated forms the warehouse layer. Snapshot data typically form Dimensions and can either be overwritten or maintained as slowly changing dimensions.

LUMINATE – PROCESSING - INCREMENTAL

Examples: StoreSales
StoreItem
StoreFulfilment



History data is ingested into landing during onboarding.
Daily incremental files are ingested into landing.

Incremental data is appended into Raw for full traceability.

Incremental data is merged into Base using the delta flag and feed date, ensuring Base reflects the most up to date version of the data. This can be handled as a single stream, processing new data and restatements together, or as two streams, with a rapid stream for daily data and a slower moving batch stream for restatements.

Curated forms the warehouse layer. Incremental data typically form Facts and are incrementally updated using similar mechanisms to Base.

Large restatements can benefit significantly from Photon accelerated merges

LUMINATE – PROCESSING – TECHNICAL SUMMARY

These processes have been designed with some key considerations in mind:

- Rapidly processing high-value, new, daily data
- Persisting data for complete traceability of changes over time
- Improving the performance of processing large restatements
- Supporting business users in onboarding and exploring new, exciting data feeds and innovations, like Near-Real-Time
- Rapidly accommodating to minor schema changes
- Enabling controlled responses to major schema changes
- Optimizing the persisted data to increase query speeds

LUMINATE – SCHEMA CHANGE

Luminate datasets are constantly evolving to provide consumers with richer data.

Raw and Base can be configured to expect a particular schema, and automatically add new columns to the Rescue column for future promotion or can be set to automatically attempt to merge the schemas, appending new columns and merging data types where natively compatible.

Allowing schemas to evolve where possible reduces the overhead schema management but also reduces the amount of control the platform places on the schema. As a trusted data source, Luminate can be considered for automatic schema evolution, but this is not recommended for non-validated sources.

Curated should have a clearly defined and controlled schema applied.

Minor schema evolution, such as additional columns, can be promoted with minimal effort

Data type changes of existing columns should be evaluated in advance to ensure compatibility

Grain changes will always require careful evolution and implementation

CURATED AND REPORTING

Partitioned and Z-Ordered to optimize partition pruning, file skipping and read speed

Walmart Luminate provides data in a structured and standard way, resulting in a basic out of the box data model.

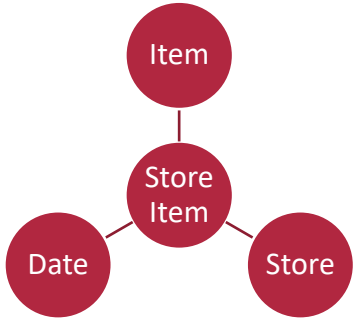
Our enhanced data model is easy to customize to your business, enabling you to augment items, stores, and more with your own wider business data.

Using the data from individual tables enables analysis of a particular area, such as sales, or stock levels. However, blending these models together enables a far richer view across your business with Walmart.

Advancing Analytics have developed common dashboards to provide instant value once data has been ingested. These can be used as a springboard to designing and customizing your view of the data.

Seasonal Analytics

Channel Performance



Stock Levels

Stock Movement

Executive Summary

Weeks of Supply

ADVANCING ANALYTICS

EXPORTS AND APPLICATIONS

Data from Luminate can feed into many downstream analytical, application and AI use cases. For this reason, it's crucial that the orchestration implementation is flexible enough to support these.