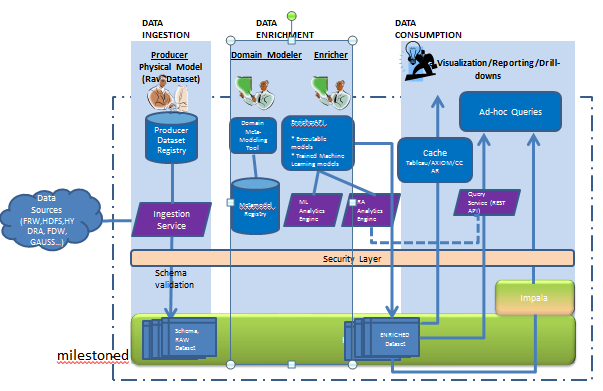
**DATA LAKE PLATFORM**

**Finance ecosystem** providing a secure, resilient platform for:

* Data ingestion
  + from various types of data sources on a scheduled basis
* Data enrichment
  + ability to create domain model (data transformations) and execute it
  + ability to run trained ML models
* Data consumption
  + of enriched data

**ARCHITECTURE**



**DATA INGESTION**

* Producers – have data that is of interest to us
* Dataset – ‘bundle’ or ‘collection’ of related data
* Dataset Registry:
  + “**metadata**” about the data. For example, dataset name, type, producer name, endpoint host/port, persistent schema, ingestion query, ingestion schedule, entitlement model, confidence level (0-1)
* Ingestion service
  + Ingests data from different data source types.
  + Performs Schema Validation and data integrity checks
  + Adds Mile stoning (bi-temporal or otherwise)
  + Persist the raw data and the mile stoned data
  + Resources
  + Hold the data itself in the target namespace as dictated by the dataset.
  + Is very HDFS specific.
  + Supports both flat and nested datasets.
  + Stored in Parquet format with a json/avro schema.

**DATA ENRICHMENT**

* Domain model is a ‘systematic definition’ of Entities in our domain. Some examples of entities: Loan Position, Wholesale risk, MDRM etc..
* Domain Modeler understands and creates domain models for business concepts.
* FRI Platform supports executable domain models.
* Domain Meta Modeling Tool supports creation of Domain Models (via Magic Draw) and Execution rules via DSL.
* Metamodel Registry: Serializes the domain model graph and supports versioning and mile stoning.
* Relational Algebra Analytics Engine
  + Deserializes the metamodel graph and loads the graph on startup.
  + Executes the model
    - Converts the Dependency graph AST to Relational Algebra graph and then to Apache spark operations
    - Supports drill down operations by creating the reverse AST to Relational Algebra graph to Spark operations.
* Machine Learning Analytics Engine
  + Contains a repository of all the trained ML models and makes them available to the enricher.
* Enricher API:
  + Simplistic API to support executing and persisting the defined domain models.
  + Example:
    - Execute domain model schedule L.
    - Run ML Trained Anomaly detection model on Schedule L filings, or run Anomaly detection model on LoanPositions.
* Enricher solves business problems by using RA Engine to execute a model or by writing custom spark code.

**DATA CONSUMPTION**

* Consumers consume only modeled data.
* Adhoc Queries
  + Via Impala by making the Enriched datasets available in the Impala metastore.
  + Query Service
    - Uses RA Analytics Engine to execute any model in the Domain Model.
    - Uses RA Analytics Engine for drill down capabilities
    - Essentially exposes RA Analytics Engine via a Rest API differing with the Enricher API by not supporting persist operation.
* Cache
  + Store enriched datasets outside the HDFS layer.
  + Offload query operations off the cluster, to allow the cluster resources to be effectively used for ingestion, analytics.

**END TO END LIFECYCLE**

* Producer registers Dataset in Registry with end points, physical schema (nested or otherwise), Query, ingestion schedule, confidence level ( 0 -1)
* IngestionService ingests data for datasets at scheduled intervals and persists milestoned data in the Platform.
* Subject Matter experts create the Domain Model and execution rules for the Domain Model.
* Runtime information i.e. mappings between the Domain model and datasets also need to be created.
* Model Registry service persists the model meta data with bi temporal milestoning.
* Business functional team create enrichers that perform analysis against modeled data be it Relational Algebra Analytics or ML Analytics. Platform will support generic RA analyzer that will work off the Entities (Aggregations or otherwise) defined in the Domain Model. ML Models that have been trained for Finance domain will also be provided in the Platform.
* Consumer register interest in modeled data and consume via various options, Query Service, Cache or Direct via Impala.