Client: Capital One
FLOGA: Data migration from on premises databases to AWS cloud.

FLOGA is the enterprise-wide Data Transformation program tasked with creating a brand new data ecosystem that will simplify the producer and consumer experience. It is responsible for creating new data ecosystem for Financial Services by rebuilding data ecosystem from the ground up, in the cloud, using modern tools and technologies. Capital One partnered with Wipro to help them design, develop and deploy for this program.

Capital One has a complex and fragmented data ecosystem due to multiple acquisitions, growth and change in technologies over many years. The Data Transformation program is responsible for the delivery of this cloud-based ecosystem and the tools within it. FLOGA aims to transition away from current data warehouses and into the cloud. The databases/Tools we are leaving are Ab initio, Teradata/SQL databases, SAS, BOBJ, on premises Hadoop lake.

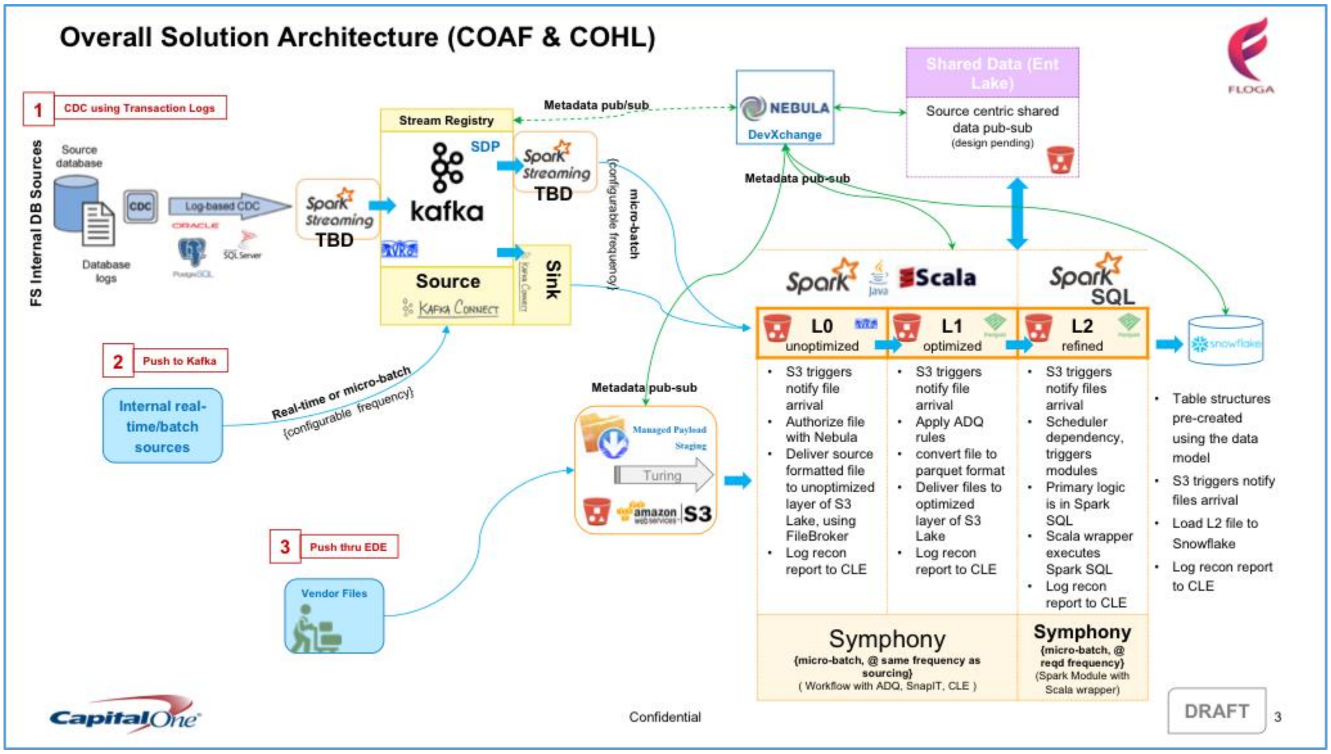
Components of the new ecosystem:

1. A **real-time streaming data** platform that will allow us to capture all the data in real time, and check for completeness and quality as it is brought into the ecosystem.
2. The **cloud-based data lake** is where we will keep all the data in its raw and complete form.
3. New **data access layer** that will make it simple for data users to discover and access data, with state-of-the-art analysis, modeling, visualization and business intelligence tools.

Benefits:

1. 100% Data in the Cloud
   - Maintain continuity and simplified data flows with source systems in the cloud
   - Rapid Scalability and Computational Flexibility
2. Enablement of Real-Time Data
   - Enable real-time monitoring and machine intelligence potential
3. Separation of Data and Computing
   - Elasticity of options to leverage rapidly evolving data computing technologies
4. Self-Service Data Products
   - Improved scale and means within data analytics via automated data collation
5. Accurate and Complete Metadata
   - Increased efficiency by allowing analysts to spend more time analyzing data instead of searching for or trying to understand data

Architecture:



Overall Solution Architecture (COAF & COHL)

L0 unoptimized

- S3 triggers notify file arrival
- Authorize file with Nebula
- Deliver source formatted file to unoptimized layer of S3 Lake, using FileBroker
- Log recon report to CLE

L1 optimized

- S3 triggers notify file arrival
- Apply ADQ rules
- convert file to parquet format
- Deliver files to optimized layer of S3 Lake
- Log recon report to CLE

L2 refined

- S3 triggers notify files arrival
- Scheduler dependency, triggers modules
- Primary logic is in Spark SQL
- Scala wrapper executes Spark SQL
- Log recon report to CLE

snowflake

- Table structures pre-created using the data model
- S3 triggers notify files arrival
- Load L2 file to Snowflake
- Log recon report to CLE

Symphony
{micro-batch, @ same frequency as sourcing}
(Workflow with ADQ, SnapIT, CLE)

Symphony
{micro-batch, @ reqd frequency}
(Spark Module with Scala wrapper)

The new ecosystem is designed in the form of a trigger based pipeline.

1. Data sourcing: Data from different sources is streamed using Kafka. Source and sink jobs are scheduled to run continuously. The raw data from source is brought into un-optimized bucket in AWS S3 in Avro format. External data is landed in another bucket where it goes through sanity check before copying to un-optimized bucket.
2. Data Quality Check: The raw data undergoes data quality check process to filter out bad records. Data is also checked against business rules. This data is stored in optimized bucket in AWS S3 in the parquet format. Data quality check is a Java based application executed on AWS EMR.
3. The data is remodeled into dimension-fact model before loading into Snowflake MPP database.
4. Different views are created based on data classification to which users with specific roles are given access.
5. The consumption layer is created from dimension fact tables for each subject area to be consumed by reports.
6. Metadata for datasets at each stage in flow is registered onto an internal portal called Nebula and reviewed by DRM team.
7. Different Apache spark based frameworks are created and deployed to execute one after another for this data movement.