

**Customer Analytics and Activation**

**Pilot**

**CAA DsU Technical Documentation Phase – I**

**V 0.1**

**Prepared By: Data Solutions Team**

Table of Contents

[Table of Contents 2](#_Toc26793459)

[1. Revision Chart 3](#_Toc26793460)

[2. Approvers 3](#_Toc26793461)

[Business Approval 3](#_Toc26793462)

[Project Team Approval 3](#_Toc26793463)

[3. Introduction 4](#_Toc26793464)

[4. In scope 4](#_Toc26793465)

[5. Out of scope 5](#_Toc26793466)

[6. Assumptions 5](#_Toc26793467)

[7. Objective and Success criteria 5](#_Toc26793468)

[8. Risk 6](#_Toc26793469)

[10. Timelines 7](#_Toc26793470)

[11. Data Feeds 7](#_Toc26793471)

[13. Appendix 9](#_Toc26793472)

# Purpose

This document covers all the technical aspects for CAA related to Data Solutions, which includes the initial architecture framework, prototype design, data model, data feeds. It is a single document to capture all the technical details. It’s focused on Phase 1, and covers details about Pilot and Historical Data Loads.

# In scope

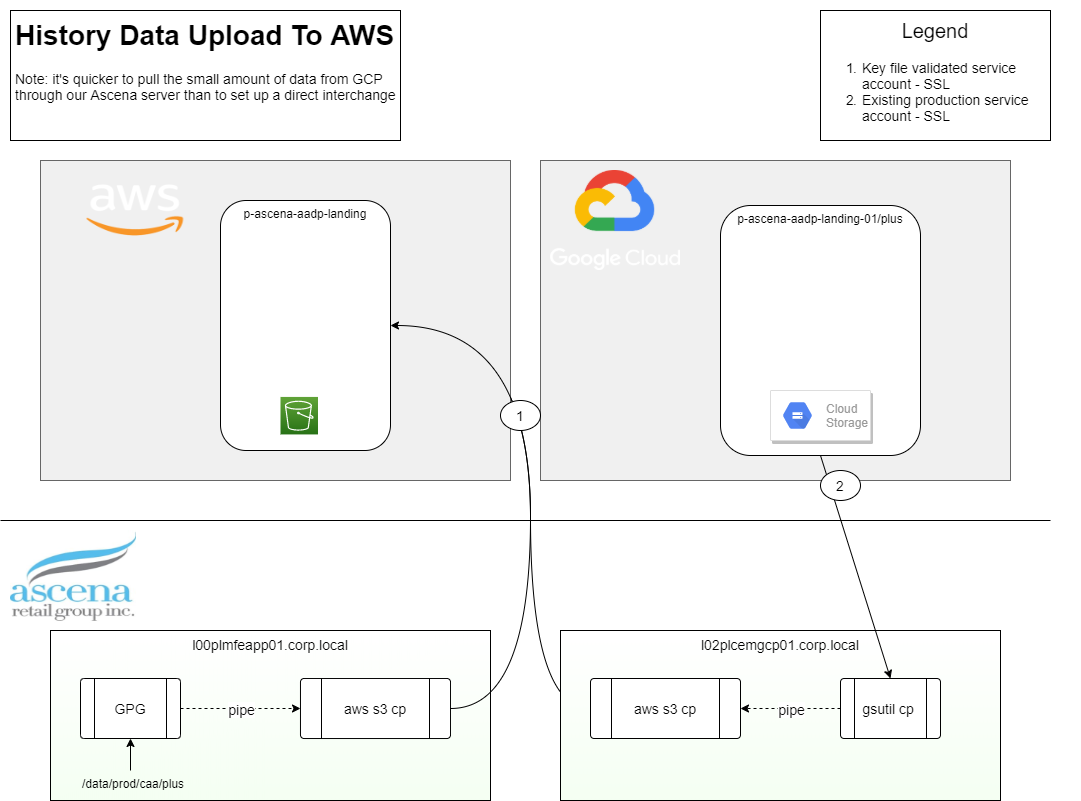
* 1. Historical Data Transfer
  2. Design Framework
  3. Prototype – Design
  4. Coding Standards
  5. Data Model for Foundation Layer
  6. Acceptance Criteria-Baseline

# Out of scope

* 1. Requirement gathering
  2. Incremental Data load
  3. Phase – II or Implementation Design

1. **Architecture Diagram**

This explains the process to load one time historical data from our ascena internal Server and GCP to AWS S3.

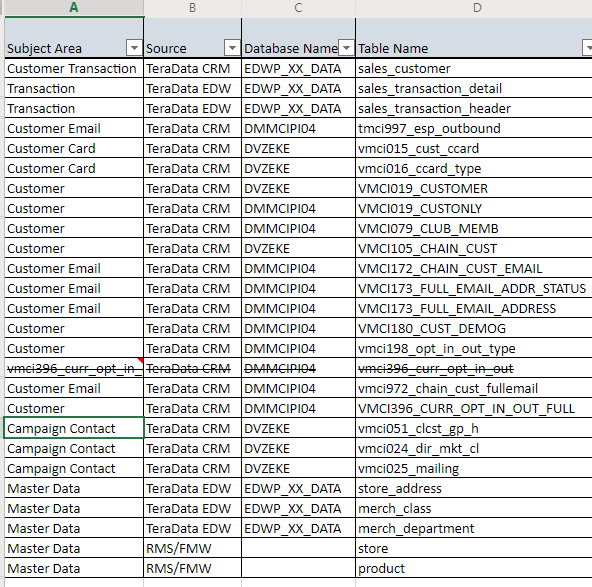


DL – Add teradata

1. **Data Feeds**

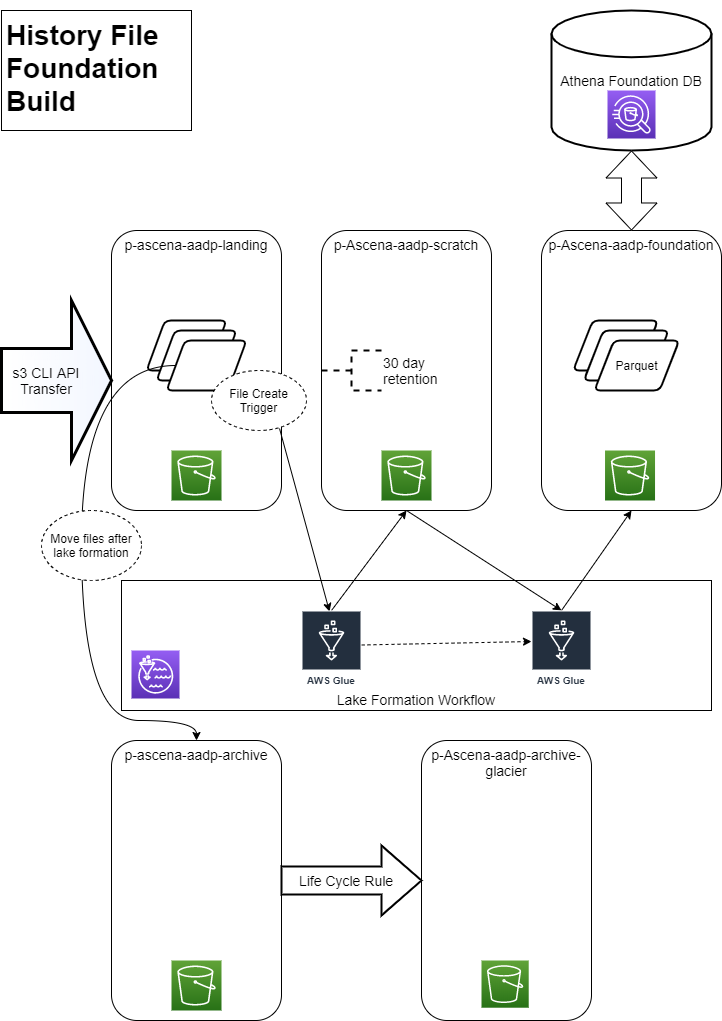
List of extracts -

<https://ascenaretail.sharepoint.com/:x:/s/sp-asc-CustomerAnalyticsActivationCEMReset/EVSZcUg0rAtNvPXyt7xKFBkBBdkCP6BuBVM9EGJOCx5G3g?e=bhDgs3>

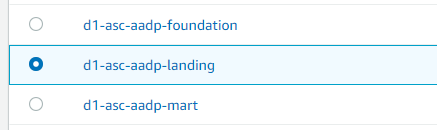


1. **Foundation Design**

This is high-level design for the Prototype, which loads data from S3 landing Layer to S3 foundation Layer using Glue transformations and will be viewed via Athena

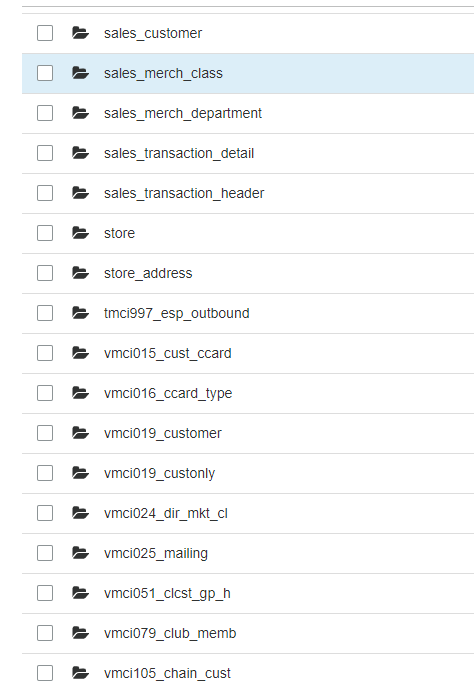


1. **Data Pipeline Layout**
2. **S3 – Simple Storage Service**



1. Landing – This has all the raw data coming from Source systems.

All the data feeds mentioned above are stored in this layer within the appropriate folder structure in BZIP format.



1. Foundation – This layer has few transformation w.r.t hashing of keys and partition. Foundation layer holds the same datatypes and table structure as source data. This layer is available to view in Athena.

All the Objects are stored in parquet format.

Refer annexure for Data model for Foundation

1. Mart – Data Mart is on top of foundation layer to avoid all the data anomalies from source systems. Data Mart follows the best practices of Data Architecture with required naming standards, table structure with more transformations, and give us one common model for all the three different sources.

Refer annexure for Foundation-Mart mapping document and Data Model for Mart Layer.

1. **Glue – ETL**



**PLUS\_Landing**

### Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

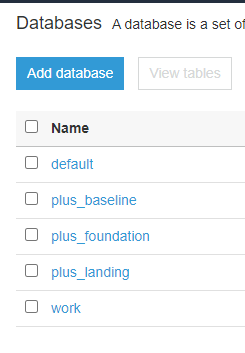
Once the data in loaded in landing table bucket, we use crawler to get the schema definition of the raw files.

e.g. Below crawler is for sales\_customer



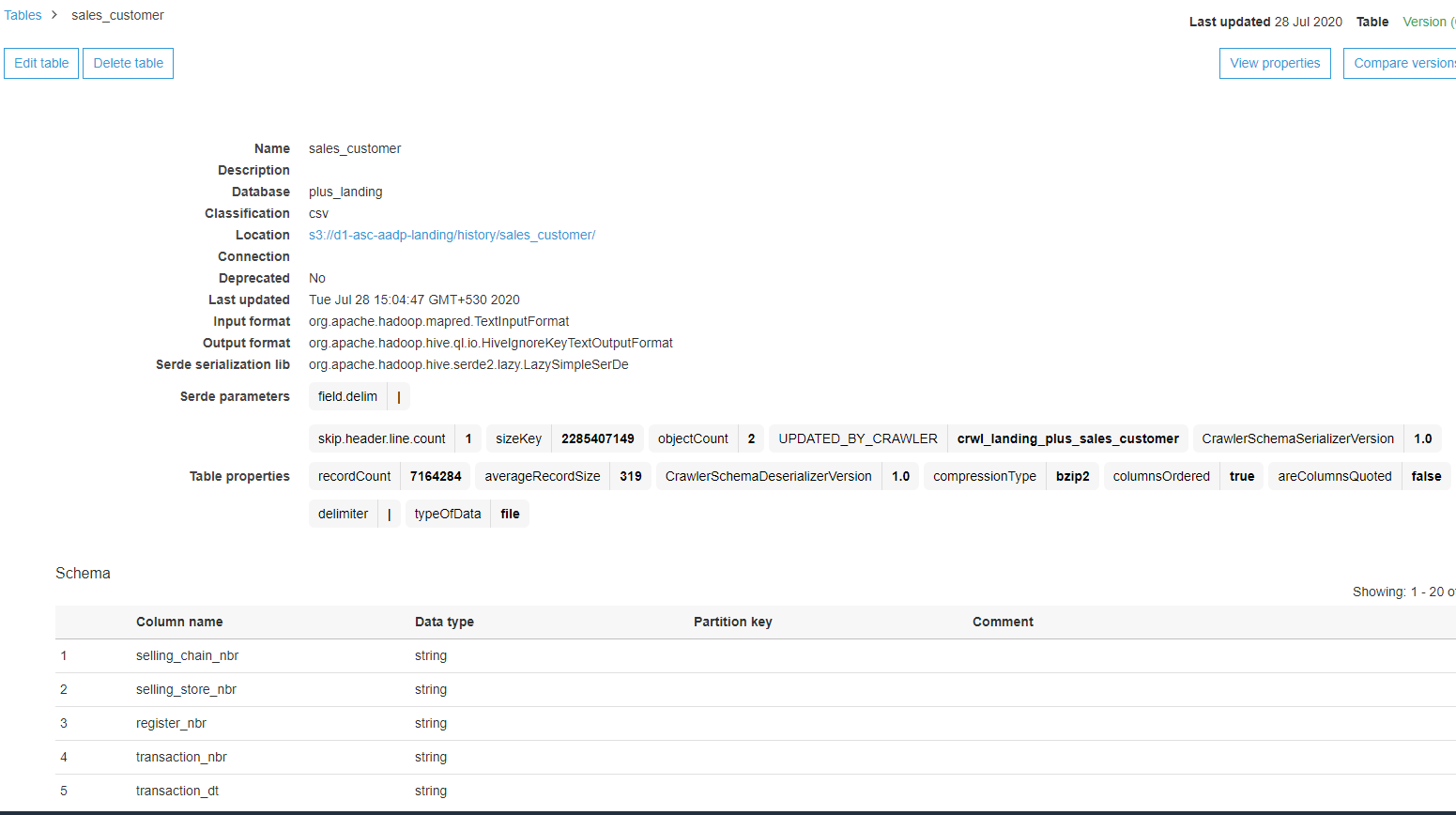
### Database – A database is a set of associated table definitions, organized into a logical group.

We have following databases in Glue where we are saving tables/ glue catalogs.



### **Tables**

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition. Once the file is crawled using crawler and table is created in plus\_landing database, we are changing all source data types to String manually in catalog to avoid any data type issues in landing and have consistent data from source to landing.

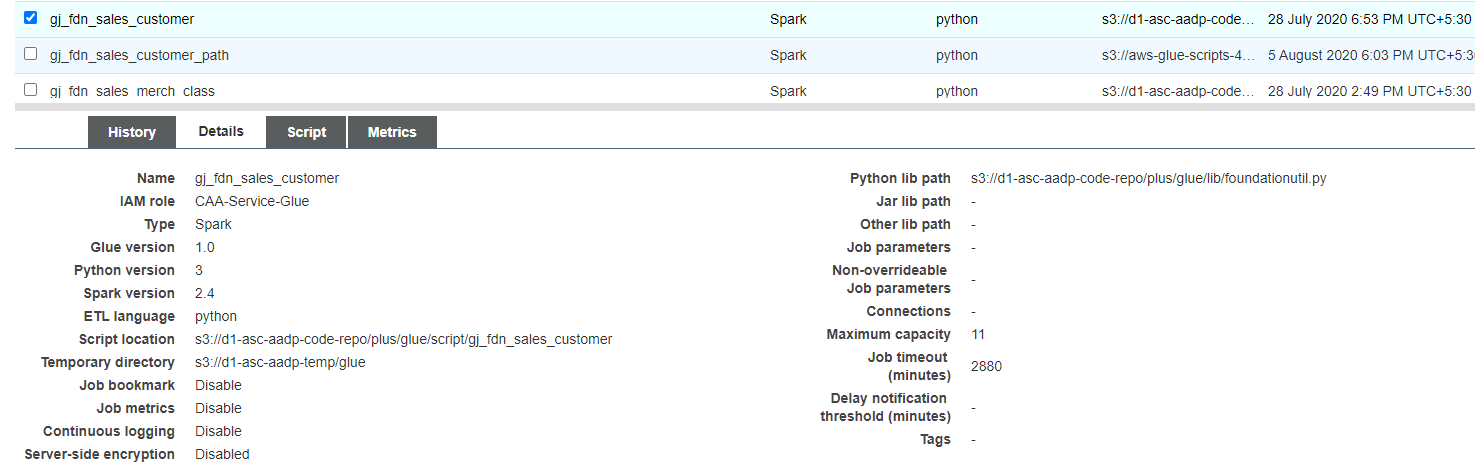


**PLUS Foundation**

Once all the landing data catalogs are created, Glue Script with Pyspark code to achieve the following –

* 1. Hash Key – Create hashing for Primary key and Foreign Key reference
  2. Audit Columns –
     1. Foundation Program Name
     2. Current Timestamp
  3. Partitioning
     1. Month-Year partition
     2. Brand code – LB/CA
  4. Cast the datatypes as per Source system
  5. Write the final table to foundation S3

**Glue Job Script**



1. **Athena – Visualization**

Data from all the three tiers can be viewed via Athena. We will have different databases for foundation and Landing. We will also be using Athena for Data Profiling on raw data in Landing Layer.

Data Model

CRM & Foundation

1. **Technical Details**

Basic Guidelines

* 1. All the resources are created in us-east-1
  2. Focus is to build a server less architecture

# Assumptions

* 1. Adequate help from Amazon SME