A blue and black logo

AI-generated content may be incorrect.



**CDM ETL ASSESSMENT REPORT**

This report delivers a comprehensive assessment of Capital Group’s existing ETL environment, currently built on Informatica PowerCenter and orchestrated through Autosys. The evaluation focuses on workflows supporting Customer Data Management (CDM) and Salesforce (SFDC), with a detailed analysis of job inventories, transformation complexities, and performance metrics. This assessment offers critical insights into the structure and usage patterns of the current Informatica workflow inventory and serves as a foundation for informed modernization planning.

The findings support a strategic shift toward cloud modernization. The proposed approach aims to rearchitect the ETL landscape into a modern, scalable, and cloud-native ecosystem. It emphasizes automation, reduction of technical debt, and implementation of adaptable frameworks that support future scalability.

This transformation initiative is designed to equip Capital Group with a resilient, efficient, and forward-looking data integration platform—enhancing operational performance today while laying the groundwork for sustained innovation and growth.

**@Date: 15-05-2025 | @Version: 1.0**

**A black and white logo

AI-generated content may be incorrect.**

**Version History**

* The document evolved from initial draft (v0.1) to v0.8 over multiple iterations.
* Key contributors: Kuntal, Nikhil, Pradip, Yesudas John.
* Each version added more detail, from inventory to incident analysis and modernization strategy.

| **Date**  📅 | **Version** | **Author(s)**  ✍️ | **Description**  📄 | **Reviewed By**  🔍 |
| --- | --- | --- | --- | --- |
| 2024-04-01 | **0.1** | 👤 Kuntal | Initial draft of assessment scope and objectives | * Yesudas * Nagesh * Arindam |
| 2024-04-07 | **0.2** | 👤 Yesudas | Incorporated review comments and methodology updates | * Nagesh * Samse * Arindam |
| 2024-05-01 | **0.5** | 👤 Kuntal, 👤 Pradip,  👤 Nikhil | Added detailed inventory breakdown and job categorization | * Nagesh * Samse |
| 2024-05-05 | **0.6** | 👤 Kuntal, 👤 Pradip,  👤 Nikhil | Updated business impact section and removed redundant jobs | * Nagesh * Arindam |
| 2024-05-07 | **0.7** | 👤 Kuntal, 👤 Pradip,  👤 Nikhil | Added job and incident performance analysis | ---— |
| 2024-05-14 | **0.8** | 👤 Kuntal, 👤 Pradip | Finalized job analysis and added modernization recommendations | ---— |
| 2024-05-15 | **0.9** | 👤 Nikhil | Finalized cloud modernization detail  Recommendations and formatting | * Arindam |

**Executive Foundations**

1. **Executive Summary**
2. **Scope & Objectives**
3. **Assessment Methodology**

**CDM ETL Current State & Workload Analysis**

**1. Workflow Inventory Overview**

* **Inventory Breakdown (High Level)**
* **Source system wise job count**

**2. Functional Categorization**

* **Categorization of Power Center Jobs**
* **Inbound Jobs categorization based on ETL layers**

**3. Transformation Complexity**

* **Mapping Complexity Description**

**4. ETL Layer Job Analysis**

* **Initialize layers Jobs Analysis Report**
* **Preland layers Jobs Analysis Report**
* **Today layers Jobs Analysis Report**
* **Delta Layer Job Analysis**

**5. Job Performance & Utilization**

* **Inbound Jobs Performance Analysis**
* **Jobs that did not run after Sep 2024**

**6. Operational Stability**

* **Last 1 year Incident Analysis –**

**7. Data Flow Optimization**

* **High Level Inbound Dataflow**

**Current State Assessment**

* **📦 Inventory**
* **Total Jobs: 2,334**
  + **1,669 Informatica PowerCenter jobs**
  + **587 Shell/Perl/MDM scripts**
  + **34 File Watchers**
  + **44 uncategorized**
* **Jobs categorized by function: Inbound, MDM, Scrubbing, Publish, Report, Dormant**
* **🏗️ Architecture**
* **On-premise architecture built around:**
  + **Informatica PowerCenter for ETL**
  + **Autosys for job scheduling**
  + **SQL Server and S3 as data sources**
* **ETL layers: Initialize → Preland → Today → Delta → Publish**
* **🔁 Data Flow**
* **Multi-layer ETL progression:**
  + **Initialize: Prepares previous layer data**
  + **Preland: Raw file to table with minimal transformation**
  + **Today: Business logic applied through joins and transformations**
  + **Delta: CDC logic with insert/update/delete tagging**
* **Manual dependency on shell scripts, file watchers, and batch logic**

|  |
| --- |
|  |

🧾 **Executive Foundations**

**Executive Summary**

This assessment presents a strategic evaluation of Capital Group’s enterprise ETL ecosystem, focusing on the modernization of data integration workflows supporting Customer Data Management (CDM) and Salesforce (SFDC). The current architecture, powered by Informatica PowerCenter and orchestrated through AutoSys, spans multiple transformation layers, including ingestion, staging, processing, and publishing.

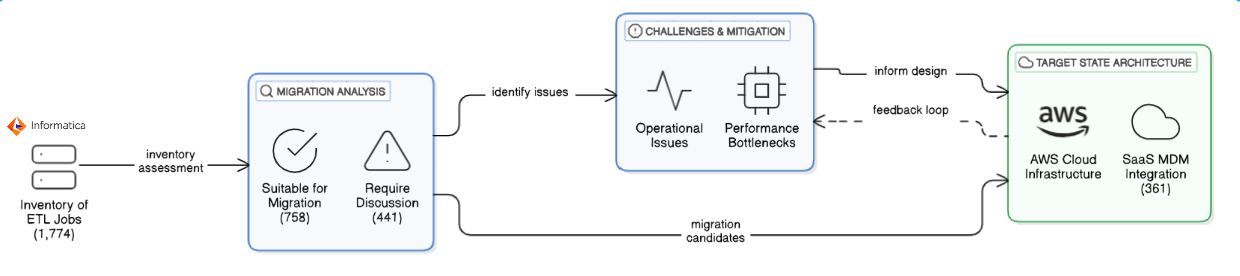
The analysis uncovers significant opportunities to streamline operations, improve system efficiency, and reduce the technical debt associated with legacy platforms. Key challenges include complex dependency chains, rigid transformation logic, limited automation, and growing maintenance overhead—factors that hinder scalability and adaptability in a cloud-first environment.

To address these challenges, a hybrid modernization strategy is recommended. This strategy advocates for transitioning from legacy ETL tools to a cloud-native architecture powered by PySpark, supported by scalable orchestration and automation frameworks. The approach is centred on building reusable, metadata-driven pipelines that promote modularity, transparency, and long-term maintainability.

This transformation represents more than a technical uplift—it is a strategic enabler for Capital Group’s broader goals of agility, innovation, and operational excellence. By embracing modern data engineering practices and scalable infrastructure, the organization will be better equipped to deliver timely insights, reduce risk, and support future business growth.

## Scope & Objective

* **Scope:** The assessment encompasses the entire ETL integration landscape involving Salesforce (SFDC) and Customer Data Management (CDM), specifically targeting workflows and mappings currently executed in the on-premises Informatica PowerCenter Production environment.
* **Objective:**
  1. Perform an in-depth inventory analysis of Informatica PowerCenter ETL components to identify optimal migration candidates to AWS.
  2. Document existing operational and performance challenges, recommending actionable mitigation plans.
  3. Propose a robust and scalable cloud-based architecture using PySpark, AWS Glue, and AWS Airflow (MWAA), integrating Master Data Management (MDM) services delivered as SaaS solutions



**Assessment Methodology**

The assessment methodology presents a structured and comprehensive approach to thoroughly evaluate the current state of CDM ETL workflows. It focuses on identifying inefficiencies, redundancies, and technical debt within the existing environment. This methodology also highlights opportunities for modernization and automation, ensuring alignment with evolving business needs. Furthermore, it assesses the overall ecosystem’s preparedness for transitioning to a scalable, cloud-native architecture that leverages modern data integration tools and frameworks for improved performance, flexibility, and maintainability.

**Assessment Data Sources**

|  |  |  |  |
| --- | --- | --- | --- |
| Referenced to extract existing documentation, business logic definitions, and mapping specifications for Informatica workflows. | Used to identify job schedules, command types, execution history, and inactive workflows for ETL readiness analysis. | Reviewed to analyse ETL mappings, transformation complexity, and workflow dependencies across CDM layers. | Accessed to validate source and target schema structures, assess data volumes, and support end-to-end data flow tracing. |

**Data Collection**

**Confluence**

Documentation available on Confluence pageTeam has followed the existing documentation available in confluence

[MSS CLIENT DATA MASTER - MSS CLIENT DATA MASTER - Confluence](https://confluence.capgroup.com/display/CDM/MSS+CLIENT+DATA+MASTER)

**Autosys Jobs portal**

Team has used PRD instance of Autosys portal - [Autorep Browser - PD1](http://autorep-cpz:8000/autorep_pd1.html) to list all the CDM jobs currently configured to run in production.

**Informatica repository for CDM**

Repository – pc105\_repo\_dev2\_3

The folder structure below was followed for analyzing the workflows and mappings

|  |  |
| --- | --- |
| Source | Folder |
| **SFDC** | CDM\_DEV3\_SFDC |
| **SFDC\_LEAD** | CDM\_DEV3\_SFDC |
| **SC** | CDM\_DEV3\_SSC |
| **Salesconnect** | CDM\_DEV3\_SSC |
| **PO** | CDM\_PO |
| **DMI** | CDM\_EACG |
| **DORIS** | CDM\_DEV3\_PRELANDING |
| **FC** | CDM\_DEV3\_FC |
| **TRAC** | CDM\_DST |
| **TA2000** | CDM\_DST |
| **Brightscope** | RPM\_DEV |
| **EI** | CDM\_EI |
| **RPA** | CDM\_RPA |

**SQL Server instance (CSSCDM/ORX)**

SQL Server - w908925\CGSQL

Schema Name

MSSCDM\_PRD/DEV – Preland, Today, Previous layer

CMS\_ORX\_10\_3/DEV3 – Landing layer tables

**Analysis Techniques:** The analysis was done based on below parameters –

**Autosys Jobs**

**Command Type** – Looking at the Autosys JIL file we categorized jobs based on command

types like sh/ksh/pl calling the shell or perl scripts, ctl calling the powercenter workflows

and fw file watcher are the jobs for monitoring the files.

**Not Running Jobs** – Identify the complete list of jobs (X) and jobs that are not running in

last 6 months (Y). The X-Y will give the list of jobs that need to be confirmed by CG on

migration readiness.

**Long Running Jobs –** Extract the duration of jobs for last 6 months. Find the median

values for each job. Order descending based on duration. Get the confirmation from CG

for an acceptable duration. The jobs falling over the acceptance level need to be identified as long-running jobs.

**Informatica Workflows**

**Categorization based on layers and functionality –** The ETL landscape is consist of

multiple layers like preland, today, previous and delta. Additionality functionality wise

we can categorize the workflows like – ingestion, MDM, publish, report etc.

**Categorization of workflows based on complexity –** The entire ingestion workflows can be categorized as Simple/Medium/Complex based number of transformation and type of transformation used.

**CDM ETL Current State & Workload Analysis**

# Inventory Breakdown (High Level)

At a high level all the jobs can be categorized as

Power center – These jobs are calling the power center workflows. There are 1669 jobs falling in this category

Scripts/MDM – There are 587 scripts consist of (.sh, .ksh and .pl) calling file transfer, checking events like file arrival, or service status check. All the MDM jobs are also under this category.

File – There are 34 jobs configured for monitoring the files.

Others – We could not categorize 44 jobs, as did not find any workflows or scripts.

A pie chart with text and numbers

AI-generated content may be incorrect.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total** | **Informatica PC** | **Scripts** | **File Watcher** | **Others** |
| **2334** | **1669** | **587** | **34** | **44** |

# Categorization of Power Center Jobs

The entire list of Power Center jobs can be categorized as below

A graph of blue rectangular objects

AI-generated content may be incorrect.

* **Inbound Jobs**: There are 803 inbound integration jobs loading data to MDM. Loading data from file to preland to today and MDM landing area from various sources.
* **MDM Jobs**: They are basically power center workflows called from MDM layer. There are 94 jobs falling in this category.
* **Data Scrubbing:** This category of jobs updating the data before or after loading to MDM, following survivorship calculation, setting some flags etc.
* **Dormant:** These jobs were not run for a longer period.
* **Publish Jobs**: These jobs make processed data available to downstream systems and users, representing the final stage in many data pipelines.
* **Report Jobs**: There are jobs generating intermediate data for next level of processing or reporting or sending files through emails.
* **Moca Jobs**: These jobs are as of considered Out of Scope.

# Inbound Jobs categorization based on ETL layers

A green and blue rectangles with white text

AI-generated content may be incorrect.

* **Initialize Jobs** – These are precursor to load any source file. It removes the existing data from the previous layer and copy data from today’s layer to the previous layer.
* **Preland Jobs** – These jobs load data from file to preland tables. There is no transformation except changing the date to PST time zone. So, there are number of preland tables equal to the number of source files. Preland load is delete and loads, it deletes the existing data in table and then loads the current data.
* **Today Jobs** – Today layer data model is like MDM data model. In this layer data is maintained at business entity level like Party, Address, Roles etc. While loading the data in this layer multiple tables are joined together, and results are passed through multiple transformation as per business rules and finally populated into today layer tables. The loading is deleted and load, all the existing data is deleted and then loaded into this table.
* **Delta Jobs** – This job calculates the change between previous day snap and today’s snap and mark the record as Insert Update or Delete. The resultant data is loaded into MDM C\_LDG tables. Here also, before loading all the data existing data is deleted and loaded into C\_LDG tables.

# Mapping Complexity Description

Measuring the complexity of a mapping can be approached by evaluating several factors related to transformations. Based on assessment, here are some key aspects that have been considered.

1. **Number of Transformations**: The more transformations a mapping contains, the more complex it is. Each transformation adds to the processing time and resource usage.
2. **Type of Transformations**: Different transformations have varying levels of complexity. For example, an Aggregator transformation, which performs calculations on groups of data, is generally more complex than a simple Filter transformation.
3. **Transformation Logic**: The complexity of the logic within each transformation also matters. Complex expressions, multiple conditions, and extensive use of functions can increase the complexity.
4. **Dependencies and Links**: The number of links between transformations and the dependencies among them can also contribute to the complexity. More links and dependencies can make the mapping harder to manage and optimize.

|  |  |
| --- | --- |
| Simple | Source to Target load with few Transformations. |
| Middle | Multiple Source/Target, Filter, Lookup, Router, Joiner |
| Complex | Multiple Source/Target, Filter, Lookup, Router, Joiner, Aggregator, SP Call, UPDT |

# Initialize layers Jobs Analysis Report

|  |  |
| --- | --- |
|  | **Initialize Layer Jobs –** There are 19 jobs falling under this category. The functionality of this job is to copy the original data in today’s layer to previous layer and delete the data from the previous layer.  **Complexity – 10**0% of the total jobs in this category falling under simple. |
| **Observation –**   * Copying the data from one table to another. * Deleting huge data volume each time will create free space fragmented. Also, the DB stats will be stale if stats are not gathered immediately.   **Recommendation –**   * Implement partitioning strategy at table level using application key or source system. * Execute data gather stats scripts immediately after removing data. * Rebuild the index if any. | |

# Preland layers Jobs Analysis Report

|  |  |
| --- | --- |
|  | **Preland Layer Jobs –** There are 120 jobs pulling the data from source file and dumping into preland tables with same structure as file.  **Complexity –** 80% of the total jobs are simple as it’s dumping the data from source file to preland tables. There are a few sources, like TA2000 and TRAC where delta data is loaded and before loading lookup is done to determine the delta. This category of jobs is falling under medium complexity. |
| **Observation –**   * Copying the data from file to table. * No transformation logic implemented as is copy of data. * Only converting the date from source to Timestamp with PST time zone. * Every time, source file format changes, new code needs to be developed. * For some sources, only incremental data is being loaded.   **Recommendation –**   * Without changing the architecture, date conversion can be included while reading the data from file to target data frame. * If common ingestion framework will be used, it will reduce individual mappings for each file and each source. | |

# Today layers Jobs Analysis Report

|  |  |
| --- | --- |
|  | **Today Layer Jobs –** There are 480+ jobs populating the data from preland tables into today layer tables. Multiple joins within preland tables are done while generating the dataset. All business rules and logic is implemented in this layer,  **Complexity –** There are 60% of the mappings using medium complex transformation (Salesconnect, SFDC) and complex SQL queries to select the data from multiple sources. 30% are complex as there are multiple sources and number of transformation is more. Some cases SP call is also being used. |
| **Observation –**   * Load the data from preland to Today layer by joining multiple preland layer tables. * Most of the business rules are implemented in this layer. * Removes all the data from the Today layer before loading. * Lookup override is used in most of the unconnected lookups. * Stored Procedure call is being made in this layer.   **Recommendation –**   * Implement partitioning strategy at table level using application key or source system. * Execute data gather stats scripts immediately after removing data. * If the common ingestion framework will be used, it will reduce the number of mappings. * For certain cases, we can merge the transformation logic to minimize creating temporary datasets. | |

# Delta Layer Job Analysis

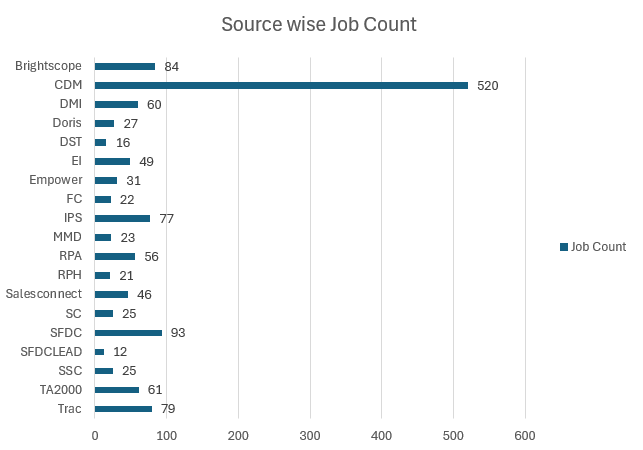
|  |  |
| --- | --- |
|  | **Delta Layer Jobs –** There are 185 jobs pulling the data from source file and dumping into preland tables with same structure as file.  **Complexity –** The majority of the jobs are of medium category as those are using SQL override at source joining multiple tables and calculating the CDC by comparing today & previous layer, followed by Union, Aggregator and Sorter. |
| **Observation –**   * Identify the change data by comparing previous & today table and mark each record as “Insert”, “Update” or “Delete”. * Update Strategy transformation is used but not used optimally as identification is done at SQL query level. * Delete all the records from C\_LDG tables and load full data with change flag.   **Recommendation –**   * Implement partitioning strategy at table level using application key or source system. * Execute data gather stats scripts immediately after removing data. * Remove unused Update Strategy transformation where only Insert is considered. | |

# Inbound Jobs Performance Analysis

|  |  |
| --- | --- |
|  | Preland :- |
| Today :- |
| Delta :- |

# Source system wise job count

The job list was extracted using <http://wlautility-primary-cpz-prd-pd1:8000/art.html>. The categorization below is based on Jobs run between last year October. (10/24) till March 2025. There are other sources, but the in the below picture we are showing the most contributing sources.

****

# Jobs that did not run after Sep 2024

Based on total CDM jobs and the list of Jobs that run in last 6 months, we arrived at the list of jobs that were not run in last 6 months. We have categorized the list of infrequent jobs yearly including the list of jobs that never run as per Autosys report.

A graph with numbers and a bar

AI-generated content may be incorrect.

# Last 1 year Incident Analysis –

Based on total CDM jobs and the list of Jobs that run in last 6 months, we arrived at the list of jobs that were not run in last 6 months. We have categorized the list of infrequent jobs yearly including the list of jobs that never run as per Autosys report.

|  |  |
| --- | --- |
| **Category** | **Count** |
| **Access** | **89** |
| **Batch** | **38** |
| **Data Quality** | **190** |
| **Error** | **153** |
| **Facilities** | **1** |
| **Failure/Not Responding** | **41** |
| **Performance/Slow** | **16** |
| **Report** | **13** |
| **Training/Informational** | **16** |

A pie chart with numbers and text

AI-generated content may be incorrect.

The above incidents are considered into re-designing certain jobs associated with Data Quality, Performance and Batch.

# High Level Inbound Dataflow

We have analyzed the inbound data flow into CDM. There are multiple hops where data is stored and accessed while moving from source to target.

At high level, PARTY entity is combination of Office, Org and Person, where for each subtype there is a table at preland layer as the data is received from source. The different preland tables are converged into Party entity at Today layer and there onwards granularity remains same till MDM load.

In proposed data flow, we are trying to show that different source files can be directly loaded into Today Party table without changing target table structure and removing the additional layer preland. In doing so we could reduce the technical debt of intermediate mappings covering file to preland and preland to today layer load.

A diagram of a data flow

AI-generated content may be incorrect.

# Metadata driven common ingestion framework

As we analyzed the inbound ingestion process, we came across multiple tables used to store the data before it gets merged into Party table in today layer. Also, there is a drawback in current design with respect to adding new sources or adding new attributes.

So we came up with a metadata (the source to target attribute mapping) based approach, where there will be a configuration file storing all the source to target column mapping for each source file and target entity. The data loading program will read this configuration file and load the data to the target.

A diagram of a computer

AI-generated content may be incorrect.

**Key Highlights**

* **Maintainability** -By-passing pre-landing storage reduces approx. 30% of inbound Jobs till today later data load.
* **Extensible** – Easy to support dynamic attribute loading by changing the configuration file.
* **Performance** – Eliminating multi hops and in-memory processing improves performance from file to stage layer load at least by 2x.
* **Less Regression** – Keeping source and target stage layer same, regression is limited to stage layer.
* **Data Lineage** – Keeping history data will ease to back track and help in creating dashboard for business needs.

# Modernization Approach

|  |  |
| --- | --- |
| Data switch Approach **Pros:**   * Pre-built tool for converting Informatica XML to PySpark * Faster initial migration * Less development effort * Standardized conversion process   **Cons:**   * Less flexibility for customization * Potential limitations in handling complex transformations * Will require additional post-migration adjustments | Custom Framework ApproachPros:  * Tailored solution specific to Capital Group's needs * Opportunity to re-architect MDM inbound and outbound integration * Full control over the migration process * Better optimization opportunities * No tool dependencies   **Cons:**   * Longer development time * Requires specialized AWS and PySpark expertise * Higher initial development cost * May require more extensive testing |

# Target State Architecture and Recommendations

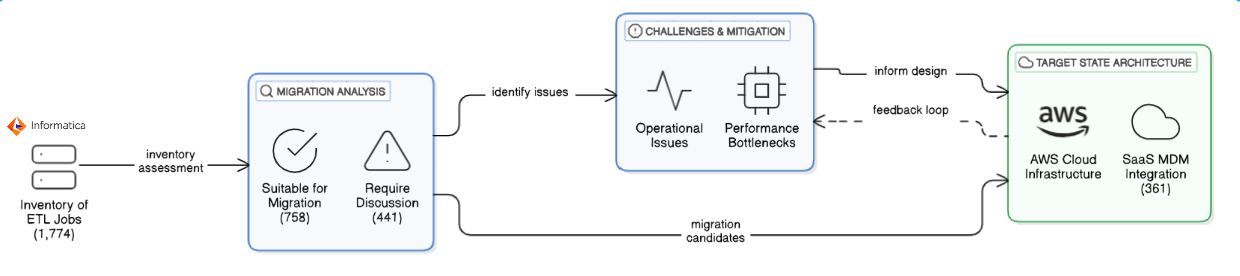
# Executive Summary

This assessment comprehensively analyzes the existing ETL Assets (Informatica PowerCenter workflows and mappings) Autosys Jobs (File Watcher, Event Driven Jobs) span across multiple layers (file-Preland-today-delta-publish around CDM and SFDC. Our detailed analysis highlights significant opportunities to streamline operations, enhance system efficiency, and reduce technical debt.

The recommended approach is a hybrid migration strategy, leveraging PySpark, AWS Glue, and AWS Managed Workflows for Apache Airflow (MWAA). This strategy prioritizes high-frequency and high-impact jobs, ensuring rapid performance improvements, better scalability, and easier maintainability.

## Scope & Objective

* **Scope:** The assessment encompasses the entire ETL integration landscape involving Salesforce (SFDC) and Customer Data Management (CDM), specifically targeting workflows and mappings currently executed in the on-premises Informatica PowerCenter Production environment.
* **Objective:**
  1. Perform an in-depth inventory analysis of Informatica PowerCenter ETL components to identify optimal migration candidates to AWS.
  2. Document existing operational and performance challenges, recommending actionable mitigation plans.
  3. Propose a robust and scalable cloud-based architecture using PySpark, AWS Glue, and AWS Airflow (MWAA), integrating Master Data Management (MDM) services delivered as SaaS solutions



# Assessment Methodology

* **Data Collection**: Data was collected through
  + Documentation available on Confluence page

Team has followed the existing documentation available in confluence [MSS CLIENT DATA MASTER - MSS CLIENT DATA MASTER - Confluence](https://confluence.capgroup.com/display/CDM/MSS+CLIENT+DATA+MASTER)

* + Autosys Jobs portal

Team has used PRD instance of Autosys portal - [Autorep Browser - PD1](http://autorep-cpz:8000/autorep_pd1.html) to list all the CDM jobs currently configured to run in production.

* + Informatica repository for CDM

Repository – pc105\_repo\_dev2\_3

The folder structure below was followed for analyzing the workflows and mappings

|  |  |
| --- | --- |
| Source | Folder |
| SFDC | CDM\_DEV3\_SFDC |
| SFDC\_LEAD | CDM\_DEV3\_SFDC |
| SC | CDM\_DEV3\_SSC |
| Salesconnect | CDM\_DEV3\_SSC |
| PO | CDM\_PO |
| DMI | CDM\_EACG |
| DORIS | CDM\_DEV3\_PRELANDING |
| FC | CDM\_DEV3\_FC |
| TRAC | CDM\_DST |
| TA2000 | CDM\_DST |
| Brightscope | RPM\_DEV |
| EI | CDM\_EI |
| RPA | CDM\_RPA |

* + SQL Server instance (CSSCDM/ORX)

SQL Server - w908925\CGSQL

Schema Name

MSSCDM\_PRD/DEV – Preland, Today, Previous layer

CMS\_ORX\_10\_3/DEV3 – Landing layer tables

* **Analysis Techniques:** The analysis was done based on below parameters –

**Jobs**

**Command Type** – Looking at the Autosys JIL file we categorized jobs based on command

types like sh/ksh/pl calling the shell or perl scripts, ctl calling the powercenter workflows

and fw file watcher are the jobs for monitoring the files.

**Not Running Jobs** – Identify the complete list of jobs (X) and jobs that are not running in

last 6 months (Y). The X-Y will give the list of jobs that need to be confirmed by CG on

migration readiness.

**Long Running Jobs –** Extract the duration of jobs for last 6 months. Find the median

values for each job. Order descending based on duration. Get the confirmation from CG

for an acceptable duration. The jobs falling over the acceptance level need to be

identified as long-running jobs.

**Informatica Workflows**

**Categorization based on layers and functionality –** The ETL landscape is consist of

multiple layers like preland, today, previous and delta. Additionality functionality wise

we can categorize the workflows like – ingestion, MDM, publish, report etc.

**Categorization of workflows based on complexity –** The entire ingestion workflows can

be categorized as Simple/Medium/Complex based number of transformation and type

of transformation used.