Table of Contents

[Version History 3](#_Toc197599913)

[Executive Summary 4](#_Toc197599914)

[Scope & Objective 4](#_Toc197599915)

[Assessment Methodology 4](#_Toc197599916)

[Inventory Breakdown 7](#_Toc197599917)

[Inbound Jobs categorization based on ETL layers 8](#_Toc197599918)

[Mapping Complexity Description 9](#_Toc197599919)

[Initialize layers Jobs Analysis Report 10](#_Toc197599920)

[Preland layers Jobs Analysis Report 11](#_Toc197599921)

[Today layers Jobs Analysis Report 12](#_Toc197599922)

[Delta Layer Job Analysis 13](#_Toc197599923)

[Source system wise job count 14](#_Toc197599924)

[Jobs that did not run after Sep 2024 15](#_Toc197599925)

[Last 1 year Incident Analysis – 16](#_Toc197599926)

[High Level Inbound Dataflow 17](#_Toc197599927)

[Metadata driven common ingestion framework 18](#_Toc197599928)

[Modernization Approach 19](#_Toc197599929)

[**Data switch Approach** 19](#_Toc197599930)

[**Custom Framework Approach** 19](#_Toc197599931)

[**Pros:** 19](#_Toc197599932)

[Target State Architecture and Recommendations 20](#_Toc197599933)

[Cloud Modernization Migration Strategy 21](#_Toc197599934)

[Migrating Informatica Jobs to AWS Glue 22](#_Toc197599935)

[Business Impact & Benefits 22](#_Toc197599936)

[Appendix 25](#_Toc197599937)

# Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Name** | **Description** | **Version** | **Revied By** |
| 4/01 | Kuntal | Initial Draft | 0.1 | Das/Nagesh/Arindam |
| 4/7 | Yesudas John | Pradipta review comments |  | Nagesh/Samse/Arindam |
| 5/1 | Kuntal/Nikhil/Pradip/Arindam | Inventory Details | 0.5 | Nagesh/Samse |
| 5/5 | Nikhil/Kuntal/Pradip | Business Impact & Benefits, Data Collection Approach & Redundant Jobs | 0.6 | Nagesh/Arindam |
| 5/7 | Nikhil/Kuntal/Pradip | Job and Incident Analysis | 0.7 |  |
| 5/14 | Kuntal/Pradip | Job Analysis | 0.8 |  |

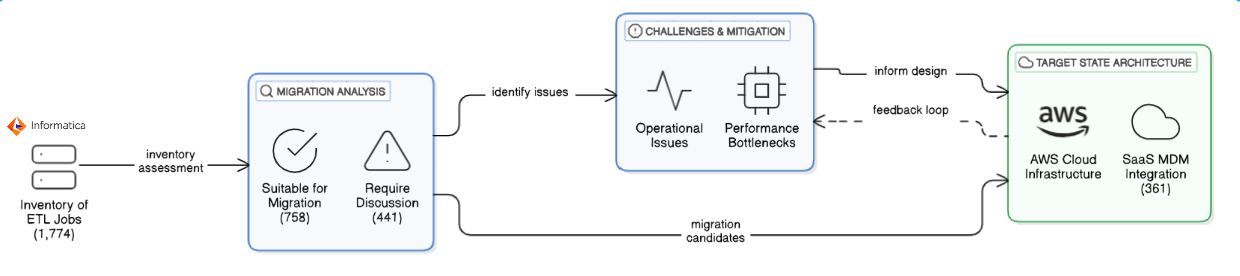
# Executive Summary

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.

# 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

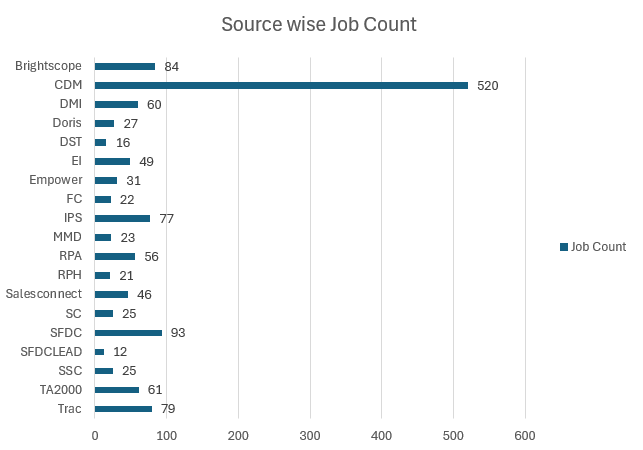
|  |  |
| --- | --- |
|  | **Delta Layer Jobs –** There are 185 jobs pulling the data from source file and dumping into preland tables with same structure as file.  **Complexity –** The majority of the jobs are of medium category as those are using SQL override at source joining multiple tables and calculating 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.

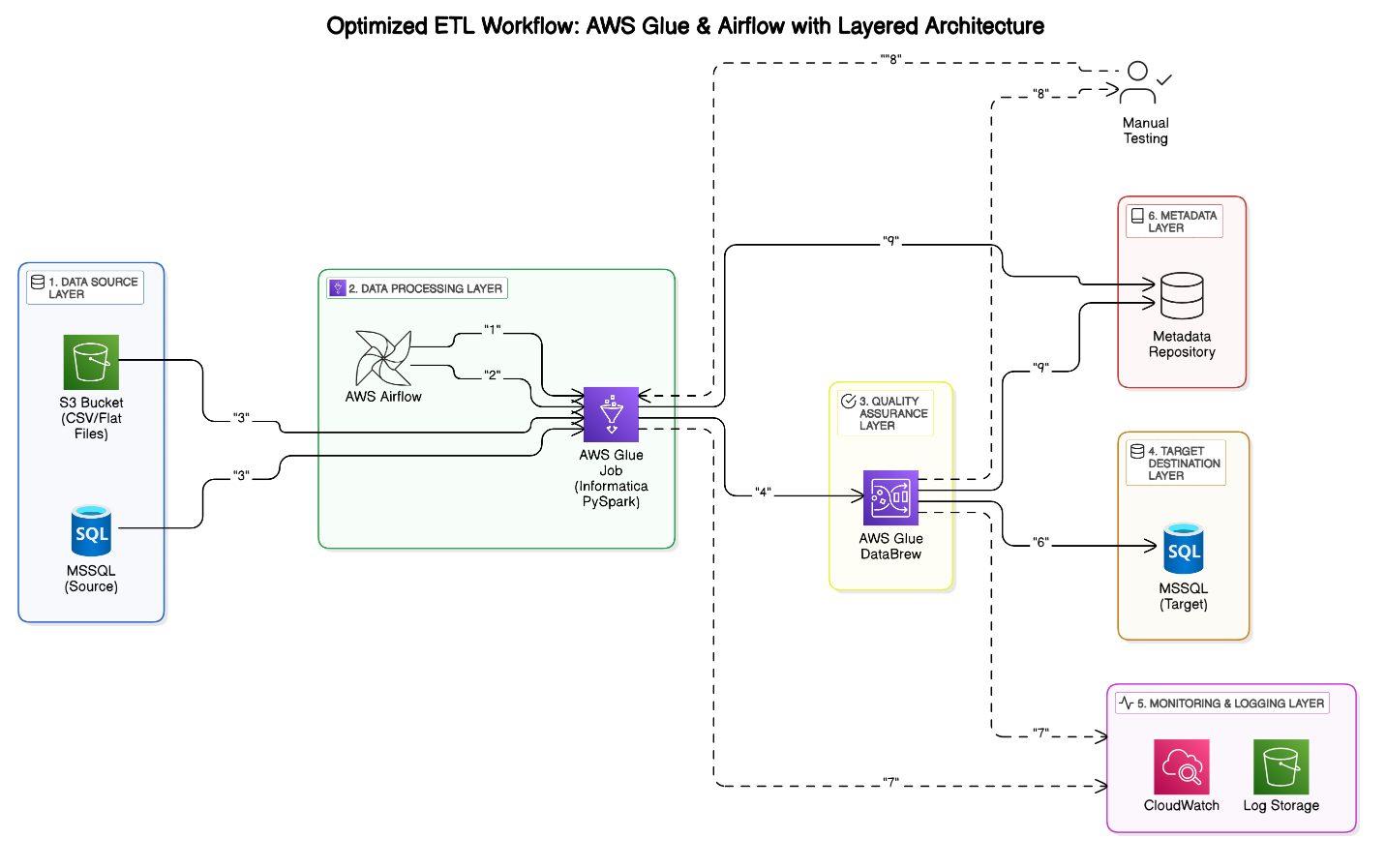
**Key Highlights**

* **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 Approach****Pros:**  * 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



**1. Data Source Layer**

* **Sources**: Collects raw data from various sources, including S3 (flat files) and MSSQL (source database).
* **Purpose**: Acts as the initial point of data entry into the ETL pipeline.

**2. Data Processing Layer**

* **Tools**: Utilizes AWS Airflow for orchestration and AWS Glue (PySpark jobs) for data transformation.
* **Functions**: Manages the scheduling, execution, and transformation of data to ensure it is in the correct format for downstream processes.

**3. Quality Assurance Layer**

* **Tools**: Employs AWS Glue DataBrew for data quality assurance.
* **Functions**: Validates the processed data to ensure accuracy, consistency, and reliability before it moves to the target destination**.**

**4. Target Destination Layer**

* **Storage**: Stores the transformed data into MSSQL (target database).
* **Purpose**: Makes the data available for downstream consumption, such as reporting, analytics, and business intelligence.

**5. Monitoring & Logging Layer**

* **Tools**: Uses AWS CloudWatch and S3 log storage.
* **Functions**: Tracks job executions, monitors errors, and captures system metrics to ensure the ETL processes are running smoothly and efficiently.

**6. Metadata Layer**

* **Functions**: Captures job metadata and transformation details.
* **Support**: Facilitates manual data validation workflows and provides insights into the ETL process for better management and troubleshooting.

**Additional Considerations**

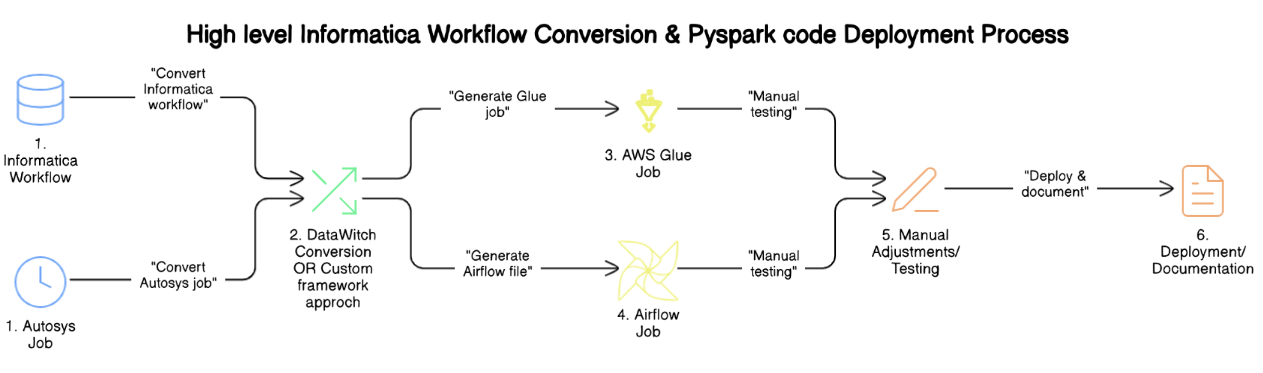
1. **Scalability**: The architecture is designed to scale with increasing data volumes and complexity, ensuring robust performance.
2. **Security**: Implements security best practices, including data encryption, access controls, and regular audits to protect sensitive information.
3. **Extensibility**: The modular design allows for easy integration of new data sources and processing tools as requirements evolve.
4. **Automation**: Emphasizes automation to reduce manual intervention, improve efficiency, and minimize the risk of errors.

A diagram of a computer

AI-generated content may be incorrect.

# 

# Cloud Modernization Migration Strategy



**Additional Notes**

* **Data Quality Check:** Although not explicitly mentioned in the flowchart, data quality checks are an integral part of the process to ensure the accuracy and integrity of the data.
* **Process Health Check:** Similarly, process health checks are conducted to monitor and maintain the overall health and performance of the workflows, ensuring they run smoothly and efficiently.

# Migrating Informatica Jobs to AWS Glue

AWS Glue is a fully managed serverless ETL (Extract, Transform, Load) service that enables easy migration and modernization of traditional Informatica PowerCenter jobs.

**Migrating Autosys Workflows to AWS Airflow (MWAA** AWS Managed Workflows for Apache Airflow (MWAA) provides a scalable and cost-effective way to orchestrate complex workflows. It serves as a modern replacement for Autosys job scheduling.

# Business Impact & Benefits

Modernizing our ETL landscape by moving from Informatica PowerCenter to AWS Glue and Airflow isn’t just a technology upgrade—it’s a strategic transformation with significant business implications. This shift touches everything from system performance and cost control to scalability, data accessibility, and future-readiness. Here's how this migration directly benefits the organization:

**1. Performance That Keeps Up With Business Demands**

One of the most immediate benefits of moving to a modern cloud-native architecture is the speed improvement. ETL processes that previously ran for hours due to hardware or legacy limitations can now be executed in a fraction of the time. AWS Glue’s serverless infrastructure automatically allocates the necessary resources, and with distributed processing via PySpark, even large datasets are handled smoothly. For business teams, this means faster access to refreshed data and reports—no more waiting for overnight batch jobs to complete.

**2. Smarter Spending, Not Just Cost Cutting**

The traditional setup required upfront investments in infrastructure, licensing, and maintenance—even when resources were underutilized. With AWS, we adopt a pay-as-you-go model. This means we only pay for what we use, when we use it. Over time, this not only lowers our operational costs but also eliminates the hidden costs of maintaining aging systems and outdated tools. It's not just about reducing spend—it’s about spending more efficiently and gaining flexibility to redirect funds toward innovation.

**3. Ready to Grow When You Are**

As our data volumes grow and new sources come in, our systems need to scale without becoming a bottleneck. The new architecture is designed to be elastic and modular. Need to onboard a new data source? No need to rework the whole pipeline. Need to process more data due to increased business activity? The system automatically scales to handle it. This flexibility allows our data strategy to grow with the business, not hold it back.

**4. More Reliable, Less Manual Work**

A lot of manual effort currently goes into monitoring jobs, troubleshooting failures, and validating outcomes. With AWS services like CloudWatch and Airflow's native tracking, we now get built-in monitoring, real-time alerts, and complete visibility into every job run. Errors are automatically flagged, retries are handled gracefully, and logs are centralized for faster debugging. This means less firefighting for engineering teams and more confidence for business users relying on this data.

**5. Easier to Maintain and Evolve**

Legacy ETL systems tend to become complex and difficult to manage over time. Code is often duplicated, transformations are hard-coded, and onboarding new team members becomes a challenge. With the new setup, jobs are built using standardized, reusable PySpark templates. Configuration is externalized, and transformations are more transparent. This simplifies maintenance, makes changes easier to implement, and improves team productivity. Over time, we also clean up unused or redundant jobs—making the system leaner and more efficient.

**6. Built for the Future**

This isn’t just about today’s needs. The platform we’re building is designed with tomorrow in mind. Whether it's integrating real-time data streams, applying machine learning models to enhance decision-making, or scaling analytics to new business units—this architecture can support it all. We’re not just solving today's problems; we're enabling tomorrow's opportunities. And because it’s built on AWS, we’re always a few clicks away from the latest tools and innovations.

**7. Direct Business Value**

At the end of the day, this isn’t just a tech upgrade—it’s about improving how the business works. With fresher, more accurate data delivered faster and more reliably, teams can make better decisions. Sales and marketing get timely customer insights, operations run more smoothly, and leadership gains visibility into key metrics without delays. It also enhances our ability to meet compliance needs, respond to audits, and build trust in data across the organization.

# Appendix

|  |  |
| --- | --- |
| **Document Name** | **Link** |
| Master Inventory List | https://confluence.capgroup.com/download/attachments/1081144759/ETL\_Assessment\_Master\_Job\_List\_v3.xlsx?api=v2 |
| Preland Assessment | https://confluence.capgroup.com/download/attachments/1081144659/File%20to%20Pre-Land%20Analysis.docx?version=4&modificationDate=1743609904927&api=v2 |
| Today Layer Assessment | https://confluence.capgroup.com/download/attachments/1081144622/Today\_Layer\_analysis\_v1.0%20%281%29.docx?api=v2 |
| Delta Layer Assessment | https://confluence.capgroup.com/download/attachments/1104085558/Delta%20Layer\_Analysis\_v1.4.docx?api=v2 |
| Publish Job Assessment |  |