Enterprise Data Platform (EDP) – Comprehensive Product Feature Document

📌 Overview

The **Enterprise Data Platform (EDP)** is a robust, configurable, and scalable data ingestion and orchestration platform built to support the full lifecycle of data—from ingestion to reporting—across all business units and data domains within Carlyle.

It allows users to onboard datasets from diverse sources (e.g., CSV, SQL Server, Snowflake, Oracle, APIs) into the AWS-based data lake, transform them, and expose them for analytics via interoperable formats such as **Apache Iceberg**, accessible through **Athena**, **Snowflake**, and other downstream systems.

🏗️ Core Architecture

1. Workflow & Job Orchestration

- **Workflow**: A logical container for related ingestion jobs.
- **Jobs**: Individual units responsible for ingesting, transforming, and validating data.
- **Job Types Supported**:
    - File-based (CSV, ZIP, SFTP, etc.)
    - SQL-based (Tables, Queries, Stored Procedures)

---

2. Three-Layer Data Architecture

Layer	Description	Storage Format

Bronze	Raw ingestion layer storing all file versions (e.g., hourly loads).	Parquet
Silver	Validated, deduplicated data with transformations (e.g., joins, lookups).	Parquet
Gold	Aggregated, reporting-ready datasets with optional direct SQL configuration.	Apache Iceberg

---

⚙️ Key Features

🔄 Data Ingestion

- Supports ingestion from:
  - CSV / Excel files
  - APIs (e.g., MailChimp, Sprinklr, Dispatch, SharePoint)
  - RDBMS: SQL Server, Oracle, Postgres, Snowflake
- **Data source registration** with auto-schema detection from sample files
- Configurable **file format, naming patterns, header/footer skipping**, encoding options

---

📅 Scheduling & Execution

- Manual execution (via play button)
- Scheduled runs using **AWS EventBridge**
- Supports multiple jobs in a single workflow
- **Retry and rerun capabilities** from any failed step

🛠️ Data Transformation

- Column mapping, renaming, and partitioning
- Derived column creation (e.g., uppercase transformation)
- Lookup and joins using reference datasets from Silver
- Custom transformation logic supported via **Python post-processing scripts**

🧠 Direct SQL Mode

- Allows SQL experts to bypass visual config and define Gold layer logic using **custom queries**
- Selecting this mode overrides prior configurations

🔁 Delta/Incr. Load Support

- Watermark column-based delta loads
- Stored Procedure support
- Parameterized SQL queries
- Fallback for systems without natural delta fields (e.g., Investran)

🧪 Reconciliation & Validation

- Config-driven reconciliation engine
- Supports **multi-file consistency checks** (e.g., campaign clicks match across reports)
- Failed validations logged to **CloudWatch**, notifications via **SNS/email**

- Detailed audit logs stored in **S3**

🔐 Security & Access Control

- Scoped visibility via **Active Directory groups**
- Four roles supported: `Admin`, `Read-Write`, `Execute`, `Read-Only`
- Access managed per **business unit/application**

📡 Integration & Output

- **Apache Iceberg** for Gold layer storage
- Accessible from:
  - **AWS Athena**
  - **Snowflake** (via Glue REST catalog integration)
  - Python clients (`boto3`)
- Compatible with downstream systems like Power BI, PDF/Excel reporting pipelines

---

🧩 Developer Productivity Features

- **Lambda layers** with zipped Python dependencies (under 256MB)
- Modular connector structure: `pull_from_source()` as the entrypoint
- Configurations stored in **S3**, secrets in **AWS Secrets Manager**
- CDK deployment enabled (CI/CD pipelines to be implemented)

📊 Monitoring & Observability

- **Workflow Monitoring UI**:
  - View step-by-step logs for each run
  - Trace data flow: Source → Bronze → Silver → Gold
- **Execution Dashboard**:
  - Daily stats on runs, success/failure counts
  - Error logs, DQ failures, and retry options

🧰 Supported Technologies

Component	Technology
Orchestration	AWS Step Functions
Compute	AWS Lambda
Storage	S3, Iceberg Tables
Querying	Athena, Snowflake
Config Mgmt	S3, AWS Secrets Manager
Scheduling	AWS EventBridge
Monitoring	CloudWatch, S3, SNS

Reporting	Power BI, Custom Python, PDF/Excel

🛣️ Roadmap & In Progress

- **Reconstruction module** (rebuilding datasets from audit history)
- **Explorer module** (Excel-style UI for non-technical users)
- **Feed module** (APIs/files to external systems)
- **CI/CD pipeline** for config deployment

Enterprise Data Platform (EDP) – End-to-End Walkthrough

This walkthrough captures the complete flow of the EDP demo as presented by Pushpendra, including live configuration steps, Q&A, and explanations around architecture, usage patterns, and integration capabilities.

---

1. 🔐 Introduction: What is EDP?

Pushpendra introduced the **Enterprise Data Platform (EDP)** as a **centralized platform** to ingest, reconcile, transform, and distribute datasets from any source system into the AWS Data Lake, and beyond.

Core Capabilities:

- Ingest data from **CSV, SQL, Snowflake, Oracle, APIs**
- Perform **validations and transformations**
- Organize data in **Bronze, Silver, and Gold layers**

- Feed data to **APIs, downstream apps, PDFs, and analytics tools**

---

2. 🧩 Workflow Management

🔹 Creating a Workflow

- Navigate to the **Workflow Management** screen.
- Each **workflow** acts as a **container of jobs**.
    - Example: A workflow may have 4 jobs (1 per file), executed as one logical unit.

🔹 Metadata Captured:

- **Workflow Name** and **Description**
- **Application Name** (e.g., ILPA)
- **Business Unit** (e.g., Fund Accounting)
- **Workflow Type**: File vs. Database
- **Notification Email**: For job failures (not for successes)

Manual or Scheduled Execution:

- Manual run via UI "Play" button
- Scheduled via EventBridge

---

3. 📁 File-Based Job Configuration

🔸 Upload & Auto-Schema Detection

- Upload a **sample CSV** to generate schema.
- Infers column names and data types based on first 100 records.

🔸 Source Configuration Options

- File sources: **S3 (internal/external), SFTP**
- Options for:
    - Header/footer row skipping
    - Encoding (UTF-8, etc.)
    - Regex-based file name pattern
    - Handling ZIP files (automated unzip + processing)

---

4. 🏗️ Bronze → Silver → Gold Pipeline

Bronze Layer

- Raw layer with **multiple versions** of the same data (e.g., hourly trade files).
- No transformation; acts as an archive and traceability layer.

Silver Layer

- Deduplicated, cleaned version with only **one version of the truth**.
- Users configure:
    - **Primary keys**, **partition keys**
    - **Column mapping**
    - **Transformations** (e.g., UPPERCASE first name)
    - **Lookups/Joins** (e.g., country code → currency)

Gold Layer

- **Final reporting layer**
- Can use:

    - **Visual config** (joins, group by, filters)
    - **Direct SQL**: Overrides visual configs with custom SQL

Iceberg tables created in this layer can be accessed by **Athena**, **Snowflake**, or other systems.

---

5. 🧪 Data Transformation & Lookups

- Supports creation of **derived columns**
- Joins with reference datasets must point to **SilverDB tables**
- Allows:
    - Conditional logic
    - Concatenation, case transformations
    - Full SQL override if needed

---

6. 🐍 Custom Post-Processing

Supports custom **Python scripts** for advanced operations, such as:

- Generating PDFs from datasets
- Running pre-defined calculations (e.g., ILPA Excel reports)

Executed **after** all jobs complete via post-processing hooks.

---

7. 🧠 Database (SQL) Based Ingestion

Supported Options:

- **Full Table Copy**

- **Delta Load** using watermark column
- **Parameterized SQL Queries**
- **Stored Procedures**

Supported Databases:

- SQL Server
- Oracle
- PostgreSQL
- Snowflake

Secrets are securely stored in **AWS Secrets Manager**.

> 💡 *Supports EC2-hosted databases and replica connections as well.*

---

8. ⚖️ Delta & Incremental Loading

Delta Support:

- Uses `when_created` or `last_modified` column for tracking changes.
- Watermark-based filtering auto-updated per job run.

Investran Edge Case:

- Does **not** have natural delta fields.
- Discussed customizing logic using **SQL Server change tracking** or stored procedures.
- CDC limitations acknowledged (loss of change history, dependency on Microsoft).

---

9. 🧪 Reconciliation & Validation

- Config-driven reconciliation between files (e.g., **MailChimp campaign vs. link report**)
- Example: Compare “unique clicks” between two reports.
- Failed reconciliations:
  - Logged in **CloudWatch**
  - Notifications sent via **SNS/email**
  - Detailed logs also written to **S3 audit path**

---

10. 🧭 Monitoring & Logs

Workflow Monitoring

- View each step:
  - File → Bronze → Silver → Gold
- See:
  - Record counts
  - Transformation times
  - Validation results
  - DQ errors
- Re-run failed jobs from any step

Execution Dashboard

- Daily job summaries
- Total runs, failures, successes
- DQ violations

---

11. 🔐 Access & Security

- Access controlled via **Active Directory groups**
- 4 levels of access:
  - **Admin**
  - **Read-Write**
  - **Read-Only**
  - **Execute**
- Users only see workflows/jobs in their business unit

---

12. 📊 Data Access & Reporting

- **Athena**: SQL access to Iceberg tables
- **Power BI**: Direct connectivity via Snowflake or Athena
- **Python**: Use `boto3` client to query Iceberg tables
- **Snowflake**: Glue REST catalog integration exposes Iceberg

Future additions:

- Excel-like data browser UI for non-technical users
- Data feed module to push to external systems

---

13. 🚧 Additional Notes & Q&A Highlights

🔹 Custom PDF Reporting

- Stivan discussed existing .NET app generating PDFs from SQL Server

- Plan: Replace SQL Server backend with Iceberg or Snowflake queries
- Possible to connect C# app via Python/Lambda to trigger reports

🔹 Stored Procedure Support

- Demo included configuring a stored proc as a job
- Used to simulate data pre-processing from SQL-based sources

🔹 Incremental Data Challenges

- Investran lacks timestamps
- Change tracking used in SQL Server
- Option to mirror logic in EDP's stored proc support

✅ Summary

The **EDP Platform** provides an end-to-end, modular, and fully orchestrated pipeline for:

- Multi-source ingestion
- Schema inference
- Validation and reconciliation
- Transformation and joins
- Monitoring, alerting, and logging
- Flexible consumption (Athena, Snowflake, Power BI, Lambda)

The system was designed with **interoperability**, **modularity**, and **observability** as key pillars, ensuring business users and technical teams can operate confidently at scale.