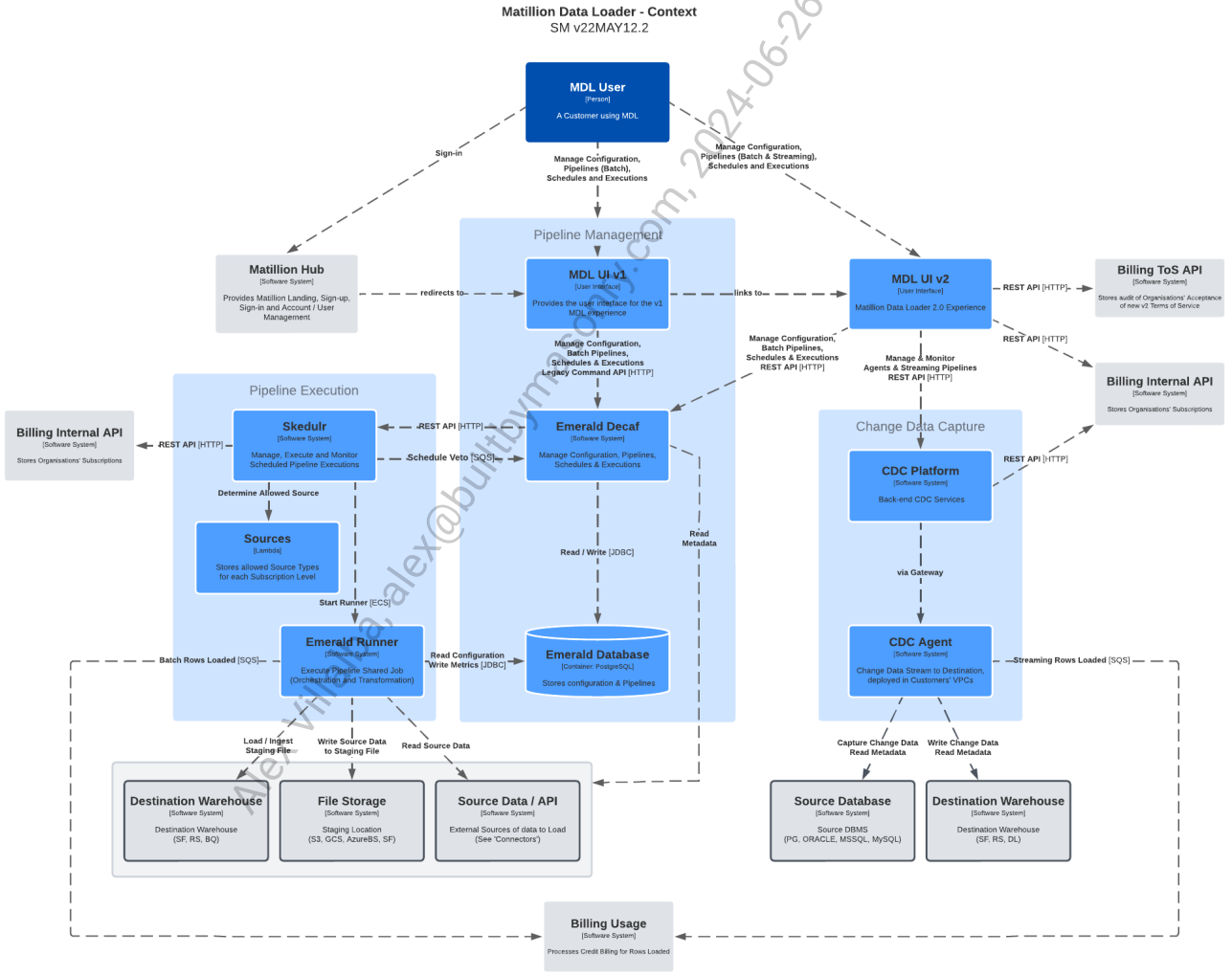
Architecture

Matillion Data Loader

Until recently, MDL as a product was entirely focussed on execution of batch jobs that load data, but in 2022, the term “Data Loading” refers to both Batch (ELT) and Streaming (CDC - change data capture) methods for loading data into a cloud data platform / warehouse. At the same time, MDL’s destiny as a world-class, paid solution for loading data has led to an initiative to create a world-class, valued User eXperience within a new “version 2.0” MDL. To these ends, we’re building a new Matillion Data Loader front-end application that will interface with the underlying Streaming (CDC) and Batch (Emerald Decaf) services to achieve user-facing configuration and management use cases.

Context

The following diagram shows the main applications we build, deploy and integrate as “Matillion Data Loader”. This diagram is intended to be a map of sorts; the highest level view of the MDL Service as a whole, and further detail is available via the links below.



The light blue areas grouping some of the applications (“Pipeline Execution”, “Pipeline Management” and “Change Data Capture”) show boundaries we use to group applications and services into teams.

Further Detail

The following subsections provide the next level of detail for each group of applications;

MDL UI v2 (Batch & Streaming)

Pipeline Management (Batch)

Pipeline Execution (Batch)

Change Data Capture (Streaming)

Sources & Destinations - see Connector Configurations

The connections to `Source Data / API` services have been shown above as a single concern; this is an oversimplification of the truth. MDL (and Matillion ETL, of course) is capable of interacting with many *sources* - as well as other incidental services, such as file storage, and destination Cloud Warehouse / Data Platform. For the purposes of this summary, we aim to describe the key components and interfaces needed to build and run Matillion Data Loader, however, each *source* connector requires a level of configuration and deployment considerations which sit on top of these components.

Alex Villalba, alex@builtbymasonry.com, 2024-06-26T20:05:55.702Z

Search

- MDL UI v2 (Batch & Streaming)
- Pipeline Management (Batch)
- Pipeline Execution (Batch)
- Change Data Capture (Streaming)
- Component Catalogue
  - MDL Frontend
  - Decaf
  - Skedulr
  - Runner
  - Partner Connect
    - Partner connect autostopping
- Connector Configurations
  - HubSpot
  - Netsuite
  - Xero
  - Google (Adwords, BigQuery, Sheets, Analytics)
  - Salesforce
  - Marketo
  - Twitter
  - Dynamics365
  - Quickbooks
  - Facebook
  - OAuth Source Template
- Technical How To...
  - Get access to source code
  - Access Dependencies for Java Applications
  - Access Dependencies for JS / Node Applications
  - Use the Runner Admin Tool
  - Set Up AWS Lambda to connect to RDS & SQS
  - Vault Secrets
  - Use LocalStack for the ScheduleVeto SQS queue

- Using Pact in MDL
- MDL Decaf/Runner Release Tech Deep Dive
- Clean up corrupted pipelines from DB/Scheduler in dev/preprod?
- Running a local scheduler connected to remote db/services
- Setting up Permissions for Databricks Compute Cluster for Cross AWS Account Access
- How to Run ... Locally
  - How to run Decaf locally
  - How to run Runner Locally
  - How to Run Scheduler Locally
- How to Test Runner Code in Isolation (Deployed Environment)
- Setting up a CYOC Pipeline to Run Locally
- Dynamic Parameters in CYOC Pipeline
- Liquibase Walkthrough with Dave Chamberlin-Kidd and Matt Robinson
- Liquibase Refinement

Alex Villalba, alex@builtbymasonry.com, 2024-06-26T20:05:55.702Z