* **OpenRec** is a reconciliation matching engine written in the Rust language. It can be used to group and match data presented to it in CSV format using easy-to-configure YAML and Lua rules.
* OpenRec comprises a small number of modules which are intended to be used as libraries and services within your own enterprise-wide solution.

**YAML (Yet Another Markup Language)**

YAML is used for configuration.

What it does in OpenRec?

* Tells OpenRec **which files to read**
* Describes **what columns to compare**
* Defines **rules** about matching and logic

**Lua (Lightweight Scripting Language)**

Lua is used for custom logic—like adding small pieces of code when simple matching isn't enough.

What it does in OpenRec?

* Adds custom formulas or logic to define how two fields are considered “equal” or “similar”
* Used when basic column matching isn't enough (for example, ignoring minor differences in date format or rounding errors in amount)

**OpenRec – High-Level Features**

**Schema-less**

* No need to define data types upfront.
* It automatically understands column types at runtime.
* You only need to specify the type if it's used in a matching rule.

**Fast**

* Built in **Rust**, so it starts instantly.
* Can match **1–2 million CSV transactions per minute** (depends on system specs).

**Lightweight**

* Uses **very little memory** (less than 100MB even for huge files!).
* It handles sorting using **disk files** instead of RAM.

**Easy to Configure**

* Uses simple, easy-to-read **YAML configuration files** (called charters).
* If you make mistakes, the tool gives **helpful error messages**.

**Flexible & Extendible**

* Supports **Lua scripting** to add custom rules or logic.
* Comes with **ready-to-use helper functions** and examples.

**Database-Free**

* Fully **file-based** — no need for a database.
* Input data goes to an **inbox** folder, and unmatched records come out in an **outbox** folder.
* Optionally supports **Prometheus monitoring** (for advanced users).

**Prometheus monitoring?**

OpenRec can be connected to Prometheus using a **Pushgateway**, which is just a helper that lets OpenRec send its metrics to Prometheus.

You can monitor things like:

* How many files it processed
* How many matches were found
* How long it took
* If any errors occurred

**Core Modules of OpenRec**

**1. Steward – *The Orchestrator***

* **Function**: Acts as the central controller, monitoring designated folders (inboxes) for new data. Upon detecting new files, it initiates the reconciliation process by coordinating other modules.
* **Responsibilities**:
  + Ensures only one reconciliation job per control runs at a time.
  + Triggers the data cleaning module (**Jetwash**) and the matching engine (**Celerity**).
  + Optionally integrates with Prometheus via Pushgateway to expose metrics.
* **Analogy**: Think of Steward as the conductor of an orchestra, ensuring each component plays its part in harmony.

**2. Jetwash – *The Data Cleaner***

* **Function**: Prepares and standardizes incoming CSV data to ensure consistency and compatibility for matching.
* **Tasks**:
  + Trims unnecessary whitespace.
  + Converts dates to a standardized ISO8601 format.
  + Adds a schema row to the data.
* **Outcome**: Delivers cleaned and well-formatted data to the matching engine (**Celerity**).
* **Analogy**: Jetwash is like a meticulous editor, refining raw data to ensure clarity and uniformity.​

**3. Celerity – *The Matching Engine***

* **Function**: Performs the core reconciliation by matching and grouping data based on predefined rules.
* **Features**:
  + Utilizes an external merge sort algorithm, leveraging disk storage instead of RAM, allowing efficient processing of large datasets with minimal memory usage.
  + Applies matching rules defined in YAML and Lua to identify and group matching records.
  + Releases matched data and retains unmatched data for further review.
* **Analogy**: Celerity is the detective, meticulously comparing records to identify matches and discrepancies.​

**Workflow Overview**

The typical data processing flow in OpenRec is as follows:​

1. **Data Arrival**: New CSV files are placed into the designated 'inbox' folder.
2. **Steward Activation**: Detects new data and initiates the processing sequence.
3. **Jetwash Processing**: Cleans and standardizes the incoming data.
4. **Celerity Matching**: Processes the cleaned data, applying matching rules to identify reconciliations.
5. **Output Generation**: Matched records are processed accordingly, while unmatched records are placed in the 'outbox' for further investigation.

https://github.com/GrandmasterTash/OpenRec/blob/main/README.md