ETL Architecture

Architecture

Anonymous · 09/09/2026

Architecture

This document explains the architecture and design principles behind Derafu ETL.

ETL Pattern

The Extract-Transform-Load (ETL) pattern is a data integration process used to collect data from various sources, transform it to fit operational needs, and load it into a target database for analysis and storage.

The Three Phases

1. **Extract**: Gathering data from source systems.
2. **Transform**: Converting the extracted data to satisfy operational requirements.
3. **Load**: Writing the transformed data to the target system.

Derafu ETL Implementation

Derafu ETL implements this pattern with a clean, object-oriented approach centered around the Pipeline concept.

Core Components

Extract Phase

- `DataSource`: Encapsulates the data source (spreadsheet, database).
- `DataExtractor`: Handles extraction logic.
- `SchemaSource`: Extracts schema information from the source.

Transform Phase

- `DataRules`: Defines transformation rules.
- `DataTransformer`: Applies transformations to extracted data.

Load Phase

- `DataTarget`: Represents the destination system.
- `DataLoader`: Manages loading data into the target.
- `SchemaTarget`: Applies schema to the target system.

Pipeline Orchestration

The `Pipeline` class orchestrates the entire ETL process, providing a fluent interface:

```
$pipeline
    ->extract($source)    // Configure extraction.
    ->transform($rules)   // Configure transformation.
    ->load($target)       // Configure loading.
    ->execute()           // Run the pipeline.
;
```

When `execute()` is called, the pipeline:

1. Validates the configuration.
2. Extracts data from the source.
3. Transforms the data according to rules.
4. Synchronizes the target schema with the source.
5. Loads the transformed data into the target.
6. Returns a result object with statistics.

Key Abstractions

Database

The `Database` abstraction provides a unified interface for different database types:

- `SpreadsheetDatabase`: Treats spreadsheets as databases using Derafu Spreadsheet.
- `DoctrineDatabase`: Works with any database supported by Doctrine DBAL.

Schema

The `Schema` system represents database structure:

- Tables, columns, indexes, foreign keys.
- Import/export to various formats (Spreadsheet, Doctrine, Markdown, D2, etc.).

Extension Points

Derafu ETL is designed for extensibility:

1. **New Data Sources**: Implement `DataSourceInterface`.
2. **Custom Transformations**: Extend `DataRules`.
3. **New Data Targets**: Implement `DataTargetInterface`.
4. **Schema Visualization**: Implement `SchemaTargetInterface`.

Design Principles

1. **Separation of Concerns**: Each component has a clear responsibility.
2. **Fluent Interface**: Expressive, chainable API.
3. **Flexibility**: Support for various formats and systems.
4. **Extensibility**: Easy to extend with custom components.

Last updated on 09/09/2026