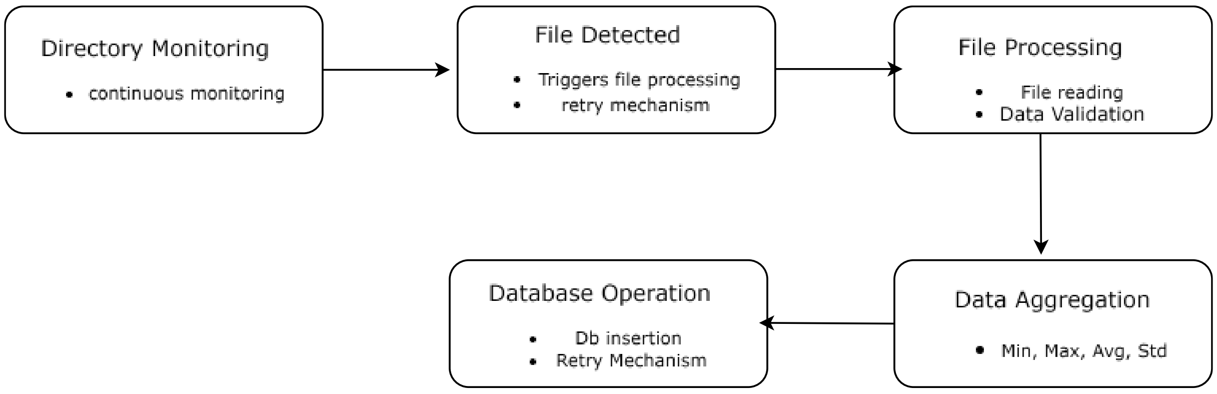
Process Workflow

- Watchdog monitors the directory for new files.
- On new file detection, the pipeline reads the file and attempts to process it. File is validated to ensure data integrity and correctness.
- If valid, the data is aggregated (e.g., compute min, max, avg, std) and inserted into the database.
- Any failed data rows (due to validation issues) are saved into a quarantine directory.
- Error handling ensures retries for failed operations (both file and database operations).
- Logs are generated at each stage for execution.
- Error and application logs are maintained under logs directory.



Details

1. **File Monitoring (Using Watchdog)**

   The pipeline begins with monitoring a specified directory for new CSV files using a Watchdog observer. This process is continuous.

   Components

   - **monitor.py** uses the watchdog library to observe file changes and trigger processing when a new CSV file is added.
   - **data_handler.py** is triggered when a new file is detected.

2. **File Processing Pipeline (Data Handler)**

   Upon detecting a new file, it is processed in several stages: validation, transformation, aggregation, and finally, storage into a database.

   Components

- **data_handler.py**: Validates the data, processes it, and logs errors if necessary.

- **Retry Mechanism**: If an error occurs while reading the file, it retries the operation up to 3 times with a delay.

3. **Data Validation**

   The data is checked for missing values, data type consistency and out-of-range values.

   Components:

   - **validation.py**: Contains functions for checking missing values, validating numeric columns, and checking timestamp formats.

4. **Data Transformation**

   After validation, data is aggregated to compute statistics (e.g., min, max, average, and standard deviation for temperature, humidity, and pressure).

   Components:

   - **aggregation.py**: Aggregates metrics like min, max, avg, std_deviation etc., by grouping data by station name and source file.

5. **Database Operations**

   Both the raw data and aggregated data are inserted into the PostgreSQL database for storage.

   Components:

   - **db_utils.py**: Handles database connections, table creation, and inserting raw/aggregated data.
   - **retry_utils.py**: Provides retry functionality for database operations in case of failures.

6. **Error Handling and Logging**

   Ensure errors are captured, logged, and failed data is saved for review.

   Components:

   - **file_utils.py**: Handles saving failed rows to quarantine and logging errors.