Claim Management

Documentation

Content

[1. Workflow Overview 2](#__RefHeading___Toc6520_1764905418)

[2. Application Overview 2](#__RefHeading___Toc6522_1764905418)

[3. Installation 3](#__RefHeading___Toc6524_1764905418)

[4. Execution 3](#__RefHeading___Toc6526_1764905418)

[5. Directory and file structure 3](#__RefHeading___Toc6528_1764905418)

[6. Application architecture and design 4](#__RefHeading___Toc2392_3424403694)

[6.1 Architecture Overview 4](#__RefHeading___Toc6530_1764905418)

[6.3 The Archiver service 5](#__RefHeading___Toc6532_1764905418)

[6.4 The Dispatcher service 7](#__RefHeading___Toc6534_1764905418)

[This service streamlines the management of email communications, ensuring that messages are properly organized based on their processing status, thereby enhancing operational efficiency. 7](#__RefHeading___Toc6536_1764905418)

[1. Configuration and initialization: 7](#__RefHeading___Toc6538_1764905418)

[2. Connecting to mailbox and database: 7](#__RefHeading___Toc6540_1764905418)

[3. Rule compilation: 7](#__RefHeading___Toc6542_1764905418)

[4. Dispatching Process: 7](#__RefHeading___Toc6544_1764905418)

[5. Processing Emails: 7](#__RefHeading___Toc6546_1764905418)

[6. Email Movement: 7](#__RefHeading___Toc6548_1764905418)

[7. Database Update: 7](#__RefHeading___Toc6550_1764905418)

[8. Finalization: 7](#__RefHeading___Toc6552_1764905418)

# **1. Workflow** **Overview**

The process of claim handling is established at the Accounts Receivables and Customer Service departments. The debit and credit notes are received from customers in a form of PDF documents to a shared mailbox, or as paper documents that need to be scanned to PDF before further processing.

1. The debit and credit notes are received from customers in a form of PDF documents to a shared mailbox, or as paper documents per post, which are are first scanned to PDF before further processing.
2. Those documents where the nature of the claim cannot be determined from the document data must be per-categorized by the CS team.
3. The PDFs are attached to outlook messages (1 PDF per email) and then distributed by the AR team to dedicated customer folders located in the EMEA-GSS-AR shared mailbox.
4. The automation then scans the email folders for new messages and downloads the pdf documents into a network storage. The PDFs are subjected to the extraction of relevant accounting data. If the document is a debit note, which has no corresponding record in SAP then a new notification/case is created. If the document is a credit note, and a corresponding case record in SAP exists, then this is modified according to the data in the credit note.
5. The processed documents are then dispatched into the corresponding subfolders in the

# **2. Application** **Overview**

The Claim Management application was developed to streamline the process of claim handling as described in the previous section. The teams receive customer debits or credits in paper form by post or as PDFs by email.

1. Documents received by post are scanned as PDF, while those received by email are already textual pdf files that do not require scanning. Those documents where the nature of the claim cannot be determined from the document data must be pre-categorized by the CS team.
2. The PDFs are attached to outlook messages (1 PDF per email) and then distributed by the AR team to dedicated customer folders located in the EMEA-GSS-AR shared mailbox.
3. The automation then scans the email folders for new messages and downloads the pdf documents into a network storage. The PDFs are subjected to the extraction of relevant accounting data. If the document is a debit note, which has no corresponding record in SAP then a new notification/case is created. If the document is a credit note, and a corresponding case record in SAP exists, then this is modified according to the data in the credit note.
4. The processed documents are then dispatched into the corresponding subfolders in the

# **3.** **Installation**

To install the application, run the “install.bat” file located in the “app” directory. Follow the on-screen instructions provided by the setup program to complete the installation process.

# **4.** **Execution**

Each service of the application is started by running the dedicated batch file: svc\_archiver.bat for the Archiver service, scv\_downloader.bat for the Downloader service, svc\_creator.bat for the Creator service and the svc\_dispatcher.bat for the Dispatcher service. Each batch file requires the Task manager’s Order ID to be passed to the %order\_id% parameter.

# **5.** **Directory and file structure**

The Claim Management app folder contains the following directories:

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |
|  |  |
|  |  |

# **6****. Application architecture and design**

The application was developed using Python 3.9. While older versions (3.7–3.8) may also work, they have not been tested.

## **6.1 Architecture Overview**

The application is built using a service-based architecture, which consists of four independent, highly specialized services. Each service operates autonomously, minimizing interdependence and potential disruptions between services.

* **Service Interaction:**

Each service interacts with the client through a user interface, which also serves as the entry point for that service. These user interfaces are tailored to the specific functions of each service, ensuring that users can engage directly with the service they need.

* **Internal Components:**

Each service comprises at least one internal component, responsible for carrying out the core functions of that service.

#### ****6.2 Data Flow and Storage Management****

A central data controller orchestrates the flow of data between the services and the storage components. This controller ensures that data is routed correctly without services needing to directly interact with each other, thus promoting modularity and reducing the risk of errors. The application utilizes multiple storage solutions, each serving a specific purpose:

* **SAP Internal Database:**

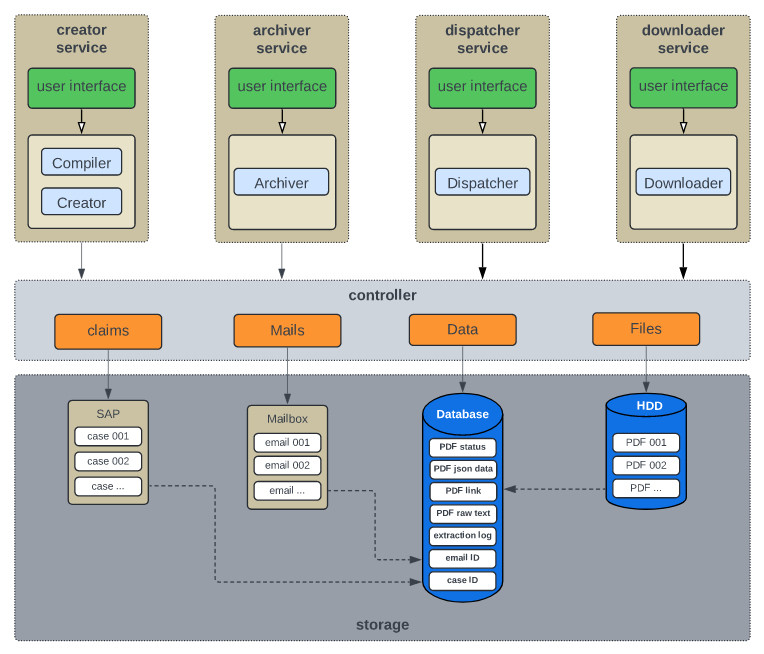
Stores disputed cases that are created by the application. This database is part of the SAP ecosystem and is integral to managing and tracking cases.

* **PostgreSQL Database:**

Used for storing parameters related to PDF processing and the data extracted from PDFs. This database serves as the primary storage for structured data outside the SAP environment.

* **Mailbox:**

Serves as a repository for emails received from customers, particularly those with PDF attachments. The mailbox acts as an input channel for documents that need to be processed by the application.

**Fig. 01: The service-based architecture of the application**

# **6.3 The Archiver service**

# This service is responsible for automating the process of archiving PDF files from specified source directories to a designated archiving folder, based on certain conditions and configurations:

1. ****Mailbox and Database Connection:****

The service begins by connecting to a specified mailbox and a database. The mailbox likely contains emails with PDF attachments or relevant metadata, while the database stores records that help identify and manage the PDF files.

1. **Configuration Retrieval:**

It retrieves necessary configurations from a predefined configuration file or database, including the paths for the source directories (where PDFs are initially stored) and the destination directories (where the PDFs will be archived).

1. **Retention Time Calculation:**

The service calculates the retention time for certain documents, such as credit notes, based on the conditions specified in the configuration. This retention time could determine whether a file should be archived immediately or retained in the source directory for a certain period.

1. **Iterating Over Source Directories:**

The service scans the specified source directories to identify PDF files that meet the archiving criteria. For each file found, the service extracts relevant information, such as a record ID embedded in the file name or metadata.

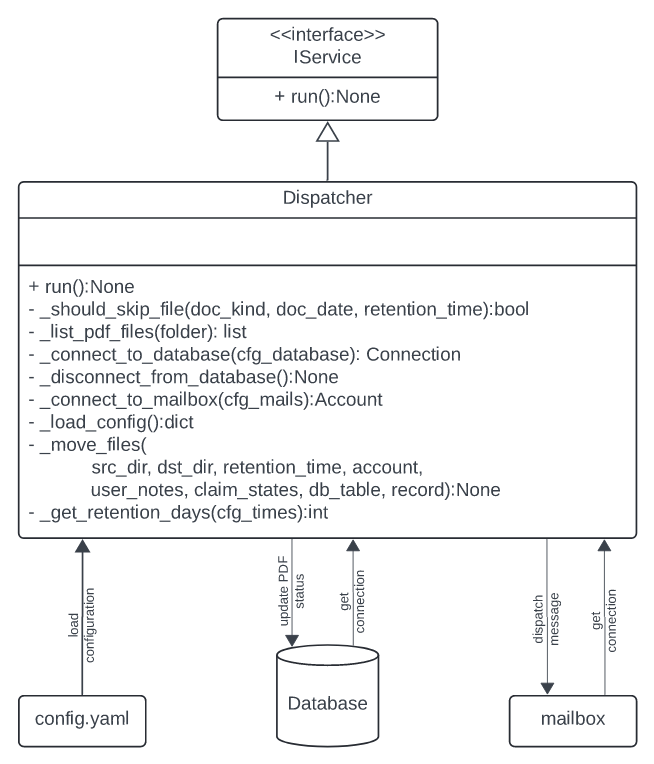
1. **Database Querying:**

Using the extracted record ID, the service queries the database to retrieve additional details or confirm whether the file should be archived. This ensures that only files meeting specific criteria are processed.

1. **File Archiving:**

Once a PDF file is validated, it is moved from the source directory to the designated archiving folder. This action may involve renaming the file according to a standard format or organizing it within a specific directory structure.

1. **Finalizing Operations:**

After processing all relevant files, the service disconnects from the database, ensuring that all connections are properly closed, and any resources are released.

**Fig. 02: The object-oriented design of the Archiver service and its connection to the storage.**

# **6.4 The Dispatcher service**

# This service streamlines the management of email communications, ensuring that messages are properly organized based on their processing status, thereby enhancing operational efficiency.

### ****1. Configuration and initialization:****

The service begins by loading its configuration settings from a YAML file. This configuration includes details necessary for connecting to the mailbox and the database, as well as rules that define how emails should be routed.

### **2. Connecting to mailbox and database:**

Using the configuration parameters, the service establishes connections to both the mailbox (to access the emails) and the database (to retrieve and update records associated with the emails).

### **3. Rule compilation:**

The service compiles routing rules based on the configuration. These rules are crucial as they determine which subfolder an email should be moved to, depending on the status of the related claim or document.

### **4. Dispatching Process:**

The service retrieves records from the database based on specific document statuses. These records include information about the emails (such as message IDs) that need to be processed. If no records match the criteria, a warning is logged, and the service terminates.

### **5. Processing Emails:**

For each record:

* + The service identifies the status of the item (e. g., whether the claim was successful or failed).
  + It checks if the item's status matches any of the predefined dispatch rules.
  + If the status is unrecognized, it logs an error and skips the item.
  + If recognized, it retrieves the email from the mailbox using the message ID.

### **6. Email Movement:**

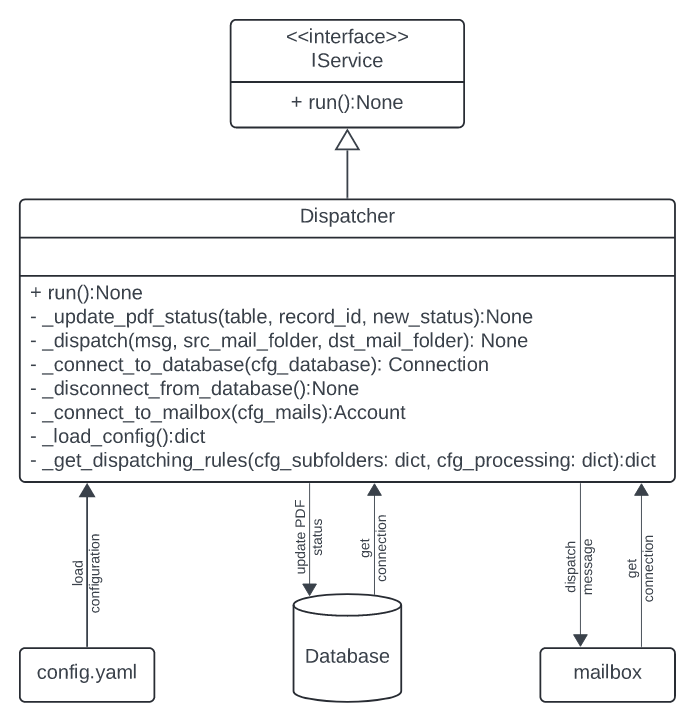
The service moves each email to the appropriate subfolder as defined by the dispatch rules. An informational email is sent to the user, notifying them of the new location of the message.

### **7. Database Update:**

After successfully moving an email, the service updates the status of the corresponding PDF file in the database to reflect the action taken.

### **8. Finalization:**

Once all records are processed, the service disconnects from the database and terminates its execution.

**Fig. 03: The object-oriented design of the Dispatcher service and its connection to the storage.**