SOFTWARE ARCHITECTURE DESIGN			
COMPLETED BY:	Zoha Bin Khaliq	DOCUMENT VERSION:	2.0
CODE:	[It will be included by PMO]	UPDATE DATE:	22/02/2023

Software Architecture Design

[Accutraxx]

SOFTWARE ARCHITECTURE DESIGN

COMPLETED BY:	Zoha Bin Khaliq	DOCUMENT VERSION:	2.0
CODE:	[It will be included by PMO]	UPDATE DATE:	22/02/2023

CONTENT

Revision history

VERSION	COMMENTS	RESPONSIBLE	DATE
1.0	Document creation	Zoha Bin Khaliq	13/02/2023
2.0	Revision 1	Zoha Bin Khaliq	22/02/2023

SOFTWARE ARCHITECTURE DESIGN

COMPLETED BY:	Zoha Bin Khaliq	DOCUMENT VERSION:	2.0
CODE:	[It will be included by PMO]	UPDATE DATE:	22/02/2023

1. Introduction

1.1. Objective

This architecture document aims to illustrate the design and architectural structure of Acccutraxx for a clear understanding of developers, Project managers, Technical Users, and stakeholders.

1.2. Scope

The scope of this document is to show a detailed architectural design of Accutraxx while along with describing the connection and use of different components with Accutraxx.

1.3. Audience of document

Role	*Responsibility*	*Expectation*
Software Architect	Maintenance	Plan and maintain information for the development of the current platform.
System Administration	Information	Know and understand the different components, connections, and capabilities for troubleshooting and business continuity.
Product Development	Information	Understand base components, connections, and capabilities to plan for new features and functionalities.
Tech Support	Information	Understand base components for troubleshooting purposes.

1.4. Overview of the service platform

Accutraxx is an issuing system. It processes the transaction of users.

Accutraxx is a business application used mainly for online banking transactions.

1.5. Scope

The scope of this document

1.6. Assumptions and dependencies

Accutraxx is an issuing system where a user can perform transactions. Since WA and Accutraxx are directly related, a user must have an active WA account. WA is a product implemented over Accutraxx.

2. Service Platform

2.1. Functional Requirements

Following are the few functional requirements Accutraxx fulfills:

- It processes the transaction for the IFARHU-CA scholarship programs.
- To safely secure the database on Level 5.
- Two databases: SQL and MongoDB.
- The user can only sign up with a CedulaD.
- A user's transactions are only processed if they have an active bank account.

2.2. Non-Functional Requirements

The following are the non-functional requirements:

- Improve the usability of the system.
- Enhance the reliability and security of the system.

2.3. Constraints

Architectural Constraints
• Accutraxx is an issuing system that processes transactions done by the user. • The current transactions are done view Accutraxx is of IFARHU-CA. • A user must have a valid bank account before processing a transaction via Accutraxx. • The WA (Mobile app) and Solidarity are integrated with Accutraxx. • To process a transaction via Accutraxx, a user must have an active WA account. • Accutraxx only processes transactions done through a bank account, virtual or physical card.

2.4. Acceptance Criteria

Figure 1 The acceptance criteria of Accutraxx processing a transaction

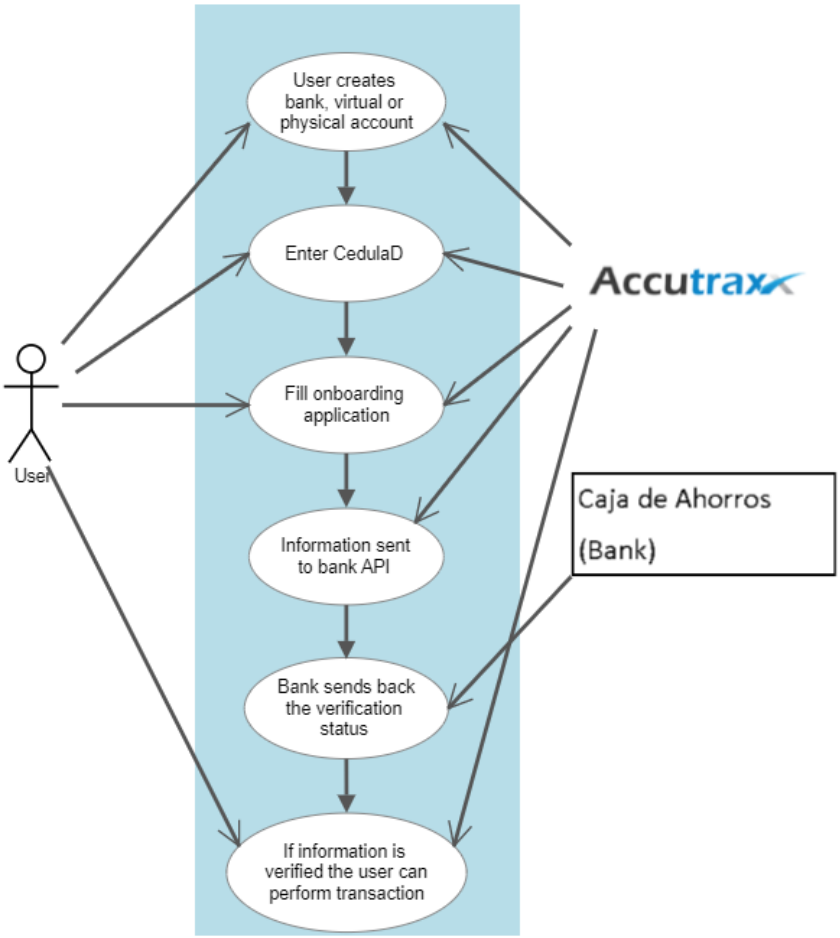
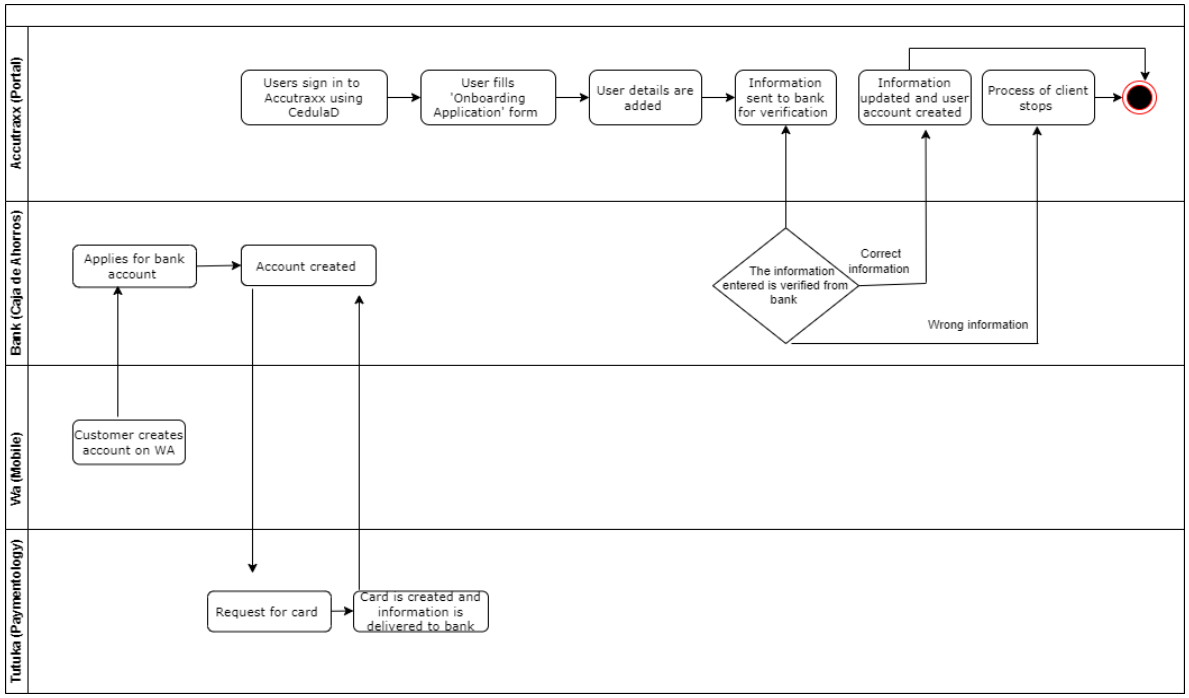


Figure 2 Primary general flow of clients creating an account on Accutraxx



3. Architecture Overview

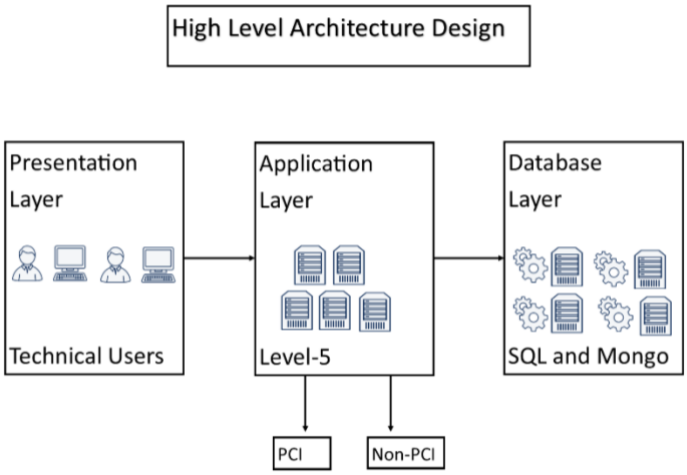
The software Architecture design of Accutraxx aids in understanding the architectural flow of the product. It shows how different products are connected with Accutraxx and the type of payment Accutraxx processes.

3.1. High-level architecture

The high-level architecture design of Accutraxx is shown in 3 tier architecture designs. It is based upon three layers:

- Presentation Layer: Client/customer
- Application Layer: Level-5, which securely contains all the data.
  - PCI: It is secured and holds all the production data.
  - Non-PCI: It is less secure and has all the test cases.
- Database Layer: SQL and Mongo server.

Figure 3: The 3-tier architecture



Explanation

The high-level architecture design represents the 3-tier architecture design of Accutraxx. The presentation layer includes all the Technical users who might code, Test, deploy or review the product.

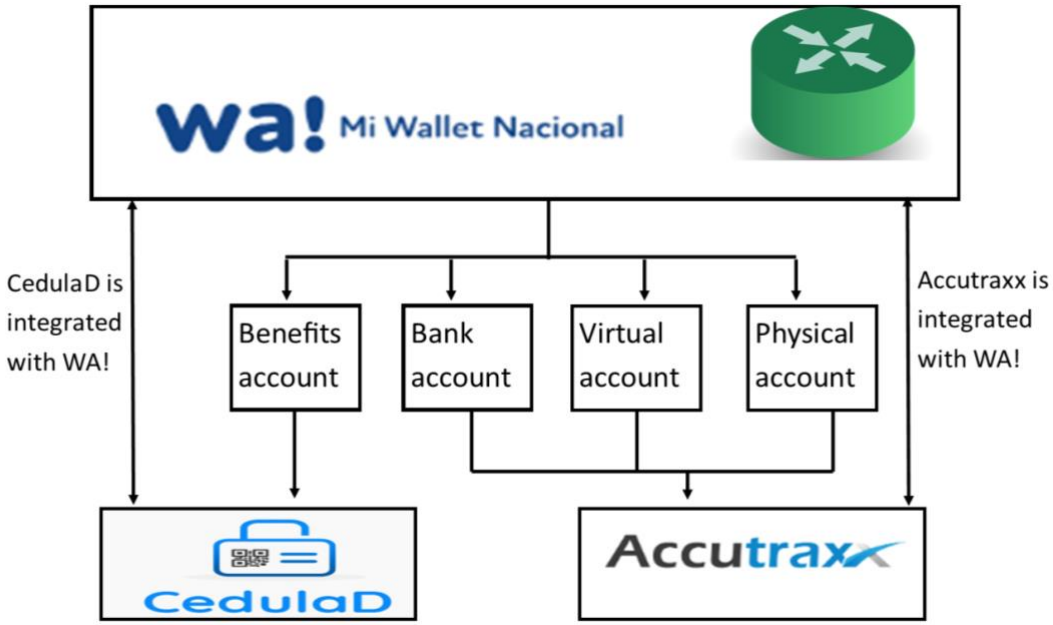
The application layer holds the Level-5 company, where all the data is secured. PCI is for secure data where production is done. Non-PCI is for testing phases and is not secure.

Database Layer contains the databases on which Accutraxx performs. Currently SQL and Mongo is used.

3.2. Key Architecture Decisions

Figure 4: The client-server architecture

Figure 5: The bank accounts whose transactions are processed by Accutraxx.



Explanation

Accutraxx API only processes transactions from a Bank account, Virtual or Physical performance. Transaction of Benefits account is done from CedulaD.

Both Accutraxx and CedulaD are integrated with WA (mobile app). WA is a middleware between Accutraxx and CedulaD.

3.3. Architecture Patterns and styles

3.4. Architecture Building Blocks

3.5. Technology Stack

The architecture design will be elaborated based on four different views. Each of these views will present an architectural viewpoint from various stakeholders' perspectives.
The following are the current technologies used in the application:

- **API:**
  - .NET 6
  - AspNetBoilerplate.com
  - EF Core
  - LINQ,
  - Dapper
  - Abp.io (Boilerplate)
- Language: c#
- **Database:**
  - SQL Server
  - Mongo
- **Frontend:**
  - Angular (Typescript)
- **Mobile:**
  - IOS(Xcode)
  - Android (Android studio)

4. Detailed Overview

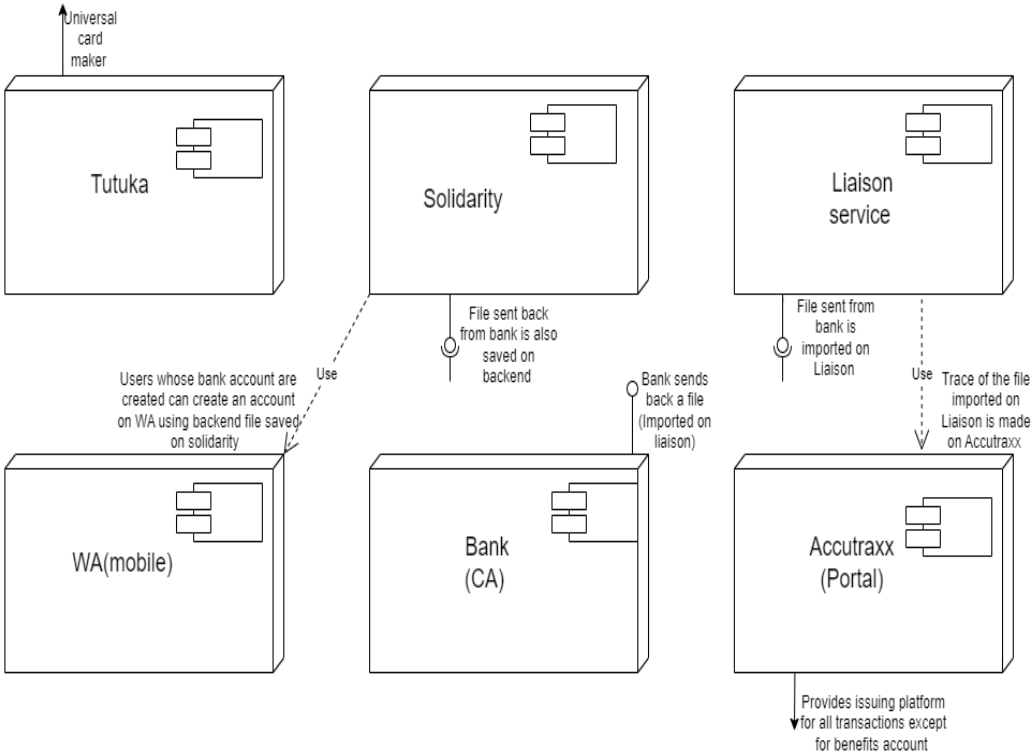
The software Architecture design of Accutraxx aids in understanding the architectural flow of the product. It shows how different products are connected with Accutraxx and the type of payment Accutraxx processes.

4.1. Component Description

Following are the six components.

1- Solidarity (Reloads)
2- Liason Service (File Processing Service)
3- Accutraxx (Portal / API)
4- WA (Mobile)
5- Bank (CA API)
6- Tutuka (Paymentology)

Figure 6: Component diagram of Accutraxx architecture



Explanation

Reload file is sent from Solidarity to the bank. The bank validates the file and then sends back a file. This file is also uploaded on FTP. The users whose file is returned from the bank can now apply for a bank account. The final file is imported into the Liaison service. A trace is also saved on Accutraxx. Users who have created a bank account can log in to WA and apply for a virtual/physical card from Tutuka.

4.2. Interaction between components

- File Solidarity (reload file)
- Generate a file and send it to the bank (Solidarity). Uploads ON FTP.
- Imports the bank response
- Now a person can apply for a bank account
- Generate a new file of applied customers and send it to the bank
- Bank send us a new file (final file)
- A user with a valid bank account applies for the card on Tutuka.
- Users with a bank account can sign in to WA
- Liaison service imports that file
- The trace of the final file sent by the bank is also saved on Accutraxx.

4.3. Data Flow

The data flow will focus on showing the static view of Accutraxx.

Following are the five systems that are related to Accutraxx (portal). All these five systems interact before a client can make virtual or physical payments using Accutraxx issuing system.

1. Solidarity (Reloads)
2. Liaison Service (File Processing Service)
3. WA (Mobile)
4. Bank (CA API)
5. Tutuka (Paymentology)

Figure 7: The sequence diagram of Accutraxx

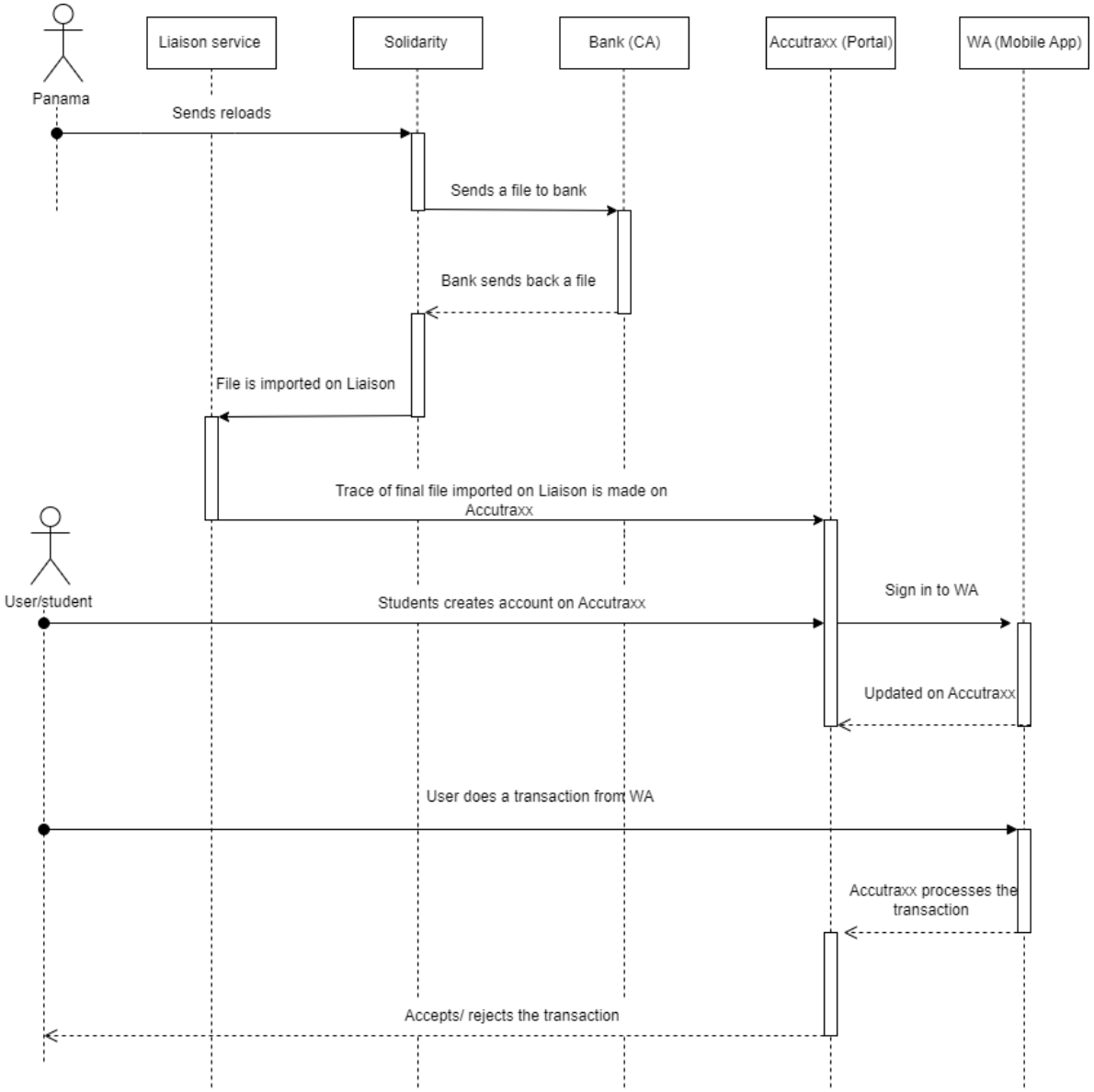
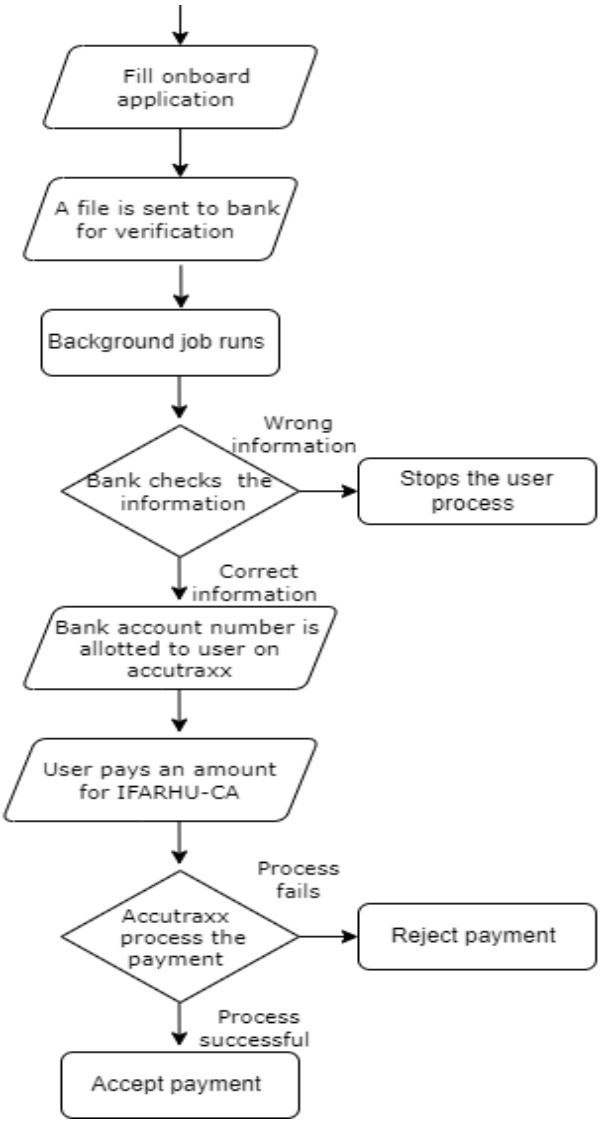
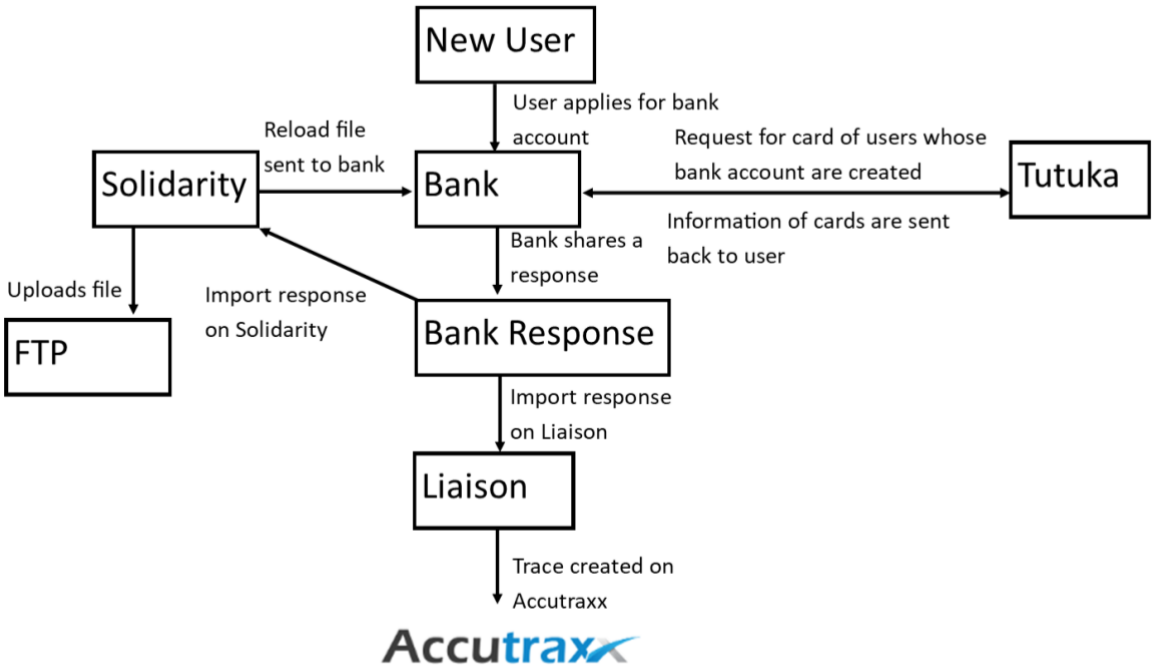


Figure 8: The flow chart diagram of Accutraxs presenting the



4.4. Integration with External Systems

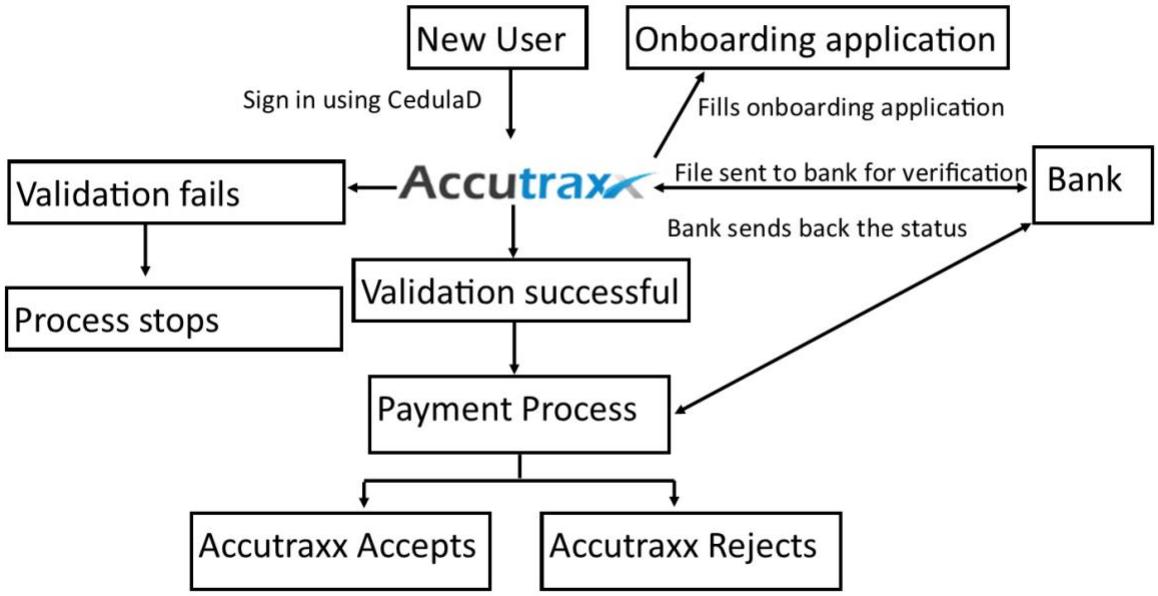
Figure 9: Integration of Accutraxx with other external systems



Explanation

The above image initiates where a reload file is sent to the bank from Solidarity. The bank shares back a file with Solidarity. The bank's response is also imported on Liaison, and the trace is saved on Accutraxx.

Figure 10: Transaction process via Accutraxx



Explanation

Once a trace is saved on Accutraxx, the Accutraxx API is called for other processes. A new user can sign in to Accutraxx by entering the CedulaD. The user will then fill in Accounts->Onboarding Application. Once the application is filed, the information entered by the user is sent to the bank for validation. The bank already has the user's knowledge saved while creating a bank. The data is validated, and the bank sends the file back. The user can proceed with the payment process from Accutraxx API if the information is correct. If the information does not match, the user cannot further process on Accutraxx.

Accutraxx accepts or rejects the payment process when the user performs a transaction via Accutraxx.

4.5. Security and Access control

According to Aspnetboilerplate security features, details can be found here

- https://aspnetboilerplate.com/Pages/Documents/Zero/User-Management
- https://aspnetboilerplate.com/Pages/Documents/Zero/Role-Management
- https://aspnetboilerplate.com/Pages/Documents/Zero/Permission-Management

5. Network architecture

6. Glossary

Glossary	
Accutraxx	Issuing system which processes the transaction
CA	Caja de Ahorros (Bank)
CedulaD	The number through which the user signs in to
FTP	Accutraax.
IFARHU-CA	The protocol server on which the file is transferred
Liaison Service	The scholarship program
Reloads	The service on which the file is uploaded.
Tutuka	Reloads are the file.
	Global paymentology

7. Abbreviations

Abbreviations	
CA	Caja de Ahorros (Bank)