Functional Specification Document

Project Members

Altin Kelmendi, Benjamin Jukic, Julian Hoffmann | **Group 6, B1/B2**

Project Title

Transaction Service for Aggregation and Management of Account Data

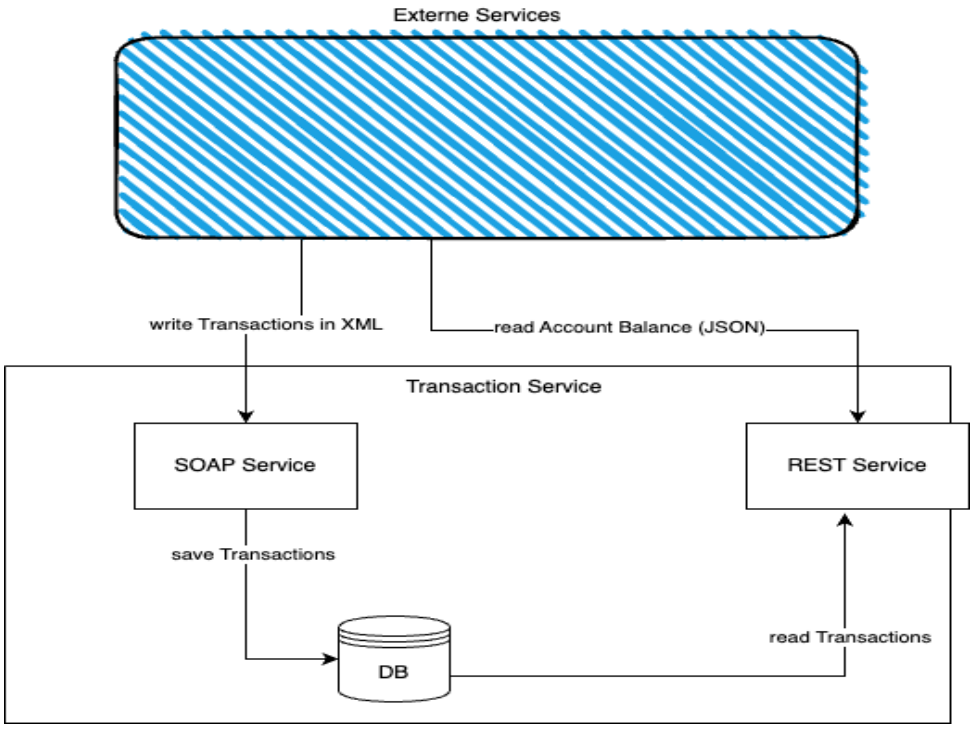
Objective

A transaction service that receives external transactions, stores them, and aggregates or queries the account balance at a specific timestamp.

System Overview

The system consists of two interfaces (SOAP and REST), a database (PostgreSQL), and external services that interact with the system.

Component Diagram



Component Description

1. SOAP Service

Function: Receiving and storing transactions.

Data format: XML (XSD-based)

Received data (attributes):
- ID (Integer)
- Name (String)
- Timestamp (ISO8601 format)
- Amount (Decimal, positive or negative)

Behavior:
- Transactions are stored in the database without filtering.
- No business logic validation or balance calculation in the SOAP service.

2. REST Service

Function: Querying the account balance (aggregated) at a specific timestamp.

Data format: JSON (schema-based)

Input parameters:
- ID (Integer)
- Timestamp (ISO8601 format)

Response:
- Current account balance at the given timestamp (aggregated from transactions)
- User (ID, Name)

Behavior:
- Aggregation: All transactions with timestamp <= given timestamp are summed.
- Result: Total account balance

3. Database (PostgreSQL)

Function: Persistence layer for transaction data.

Table structure (Transactions):
- ID (Primary Key)
- Name
- Timestamp
- Amount

Usage:
- SOAP writes data
- REST reads and aggregates data

4. External Services

Write transactions via SOAP (XML)
Read account balance via REST (JSON)

Use of Standards

- SOAP Service: XSD for validating the transaction structure
- REST Service: JSON Schema for validating requests

Use Cases

+++ UC1: Store Transaction

An external service sends a list of transactions via SOAP.
Each transaction contains ID, Name, Timestamp, and Amount.
The transactions are persisted.

Example XML:

```
<transaction>
  <id>123</id>
  <name>Max Mustermann</name>
  <timestamp>2025-05-10T12:00:00Z</timestamp>
  <amount>+100.00</amount>
</transaction>
```

--- UC2: Query Account Balance

A client sends a GET/POST request to the REST API with ID and Timestamp.
The account balance at the given time is calculated.

Example JSON request:

```
{
  "id": "123",
  "timestamp": "2025-05-10T13:00:00Z"
}
```

Response:

```
{
  "id": "123",
  "balance": 250.00
}
```

Summary:

This service acts as a central integration component for transactions, offering both a write interface (SOAP/XML) and a read interface (REST/JSON). The data is stored in a PostgreSQL database, which serves as the source for aggregated balance information.

Technology Stack:

- SOAP + XML + XSD
- REST + JSON + Schema
- PostgreSQL