Technical Document

The following is the technical documentation outlining how the **CDC Case Notification** system operates and functions end-to-end.

Input

- The pipeline begins with a **Debezium source connector** configured on the `ODSE..CNTransportQOut` table.
    - Any new data inserted into this table is detected by the connector, which publishes the data to a Kafka topic. This topic is then consumed by the **Data Extraction** service.
- The **Case Notification** process depends on predefined configurations stored in the `MSOUTE..Case_Notification_Config` table
    - This table contains all essential settings required for the pipeline to operate correctly, including **PHINMS properties** that are critical for the downstream PHINMS processing.

Output

- Processed data is persisted in either the `MSGOUTE..TransportQOut` or `MSGOUTE..NetssTransportQOut` tables, depending on the event type.
- Faulty data is routed to the `MSGOUTE..case_notification_dlt` (Dead Letter) table for further investigation by analysts..

For the entire pipeline to function properly, all three services—**Data Extraction**, **Case Notification**, and **HL7 Parser**—must be up and running. Each plays a critical role in the end-to-end processing and transformation of case notification data.

- Data Extraction continuously extracts and categorizes data, then sends it downstream using a Debezium source connector and Kafka topic.
    - This service requires the `Source connector` to be set up prior to launch.
    - The source connector on the `CNTransportQOut` table is actively maintained by the RTR Data Team.
    - Additionally, the service must use the same Kafka cluster as RTR.
- Case Notification consist of both microservice and APIs
    - The microservice actively listens for events from the Data Extraction module and processes any incoming events.

        - The service also relies on configuration settings defined in the `Case_Notification_Config` table.
        - There are two valid event types: **MMG (NON_STD)** and **STD**. These event types are processed further and delivered to either the `TransportQOut` or `NetssTransportQOut` tables.
        - Invalid events will be ignored
        - Faulty MMG or STD events are pushed to the `Case_Notification_DLT` table for further investigation.
    - Case Notification requires a constant connection to the HL7 server to transform NND XML into HL7 format.
    - Case Notification's APIs allow clients to explicitly control key features without requiring direct access to the database. This includes actions such as updating configurations, checking data status, investigating issues, and reprocessing records from the Dead Letter Table (DLT).
        - Find out more here at Swagger
    - he service also includes built-in **Liquibase** integration, enabling it to automatically manage database schema changes—such as creating or updating relevant tables—during startup.
- HL7 Service
    - This helper service is actively used by the Case Notification system. It receives NND XML data, attempts to convert it into HL7 2.5.1 format, and performs validation to ensure the transformed data meets HL7 standards before returning it to the Case Notification service
    - Additionally, the service provides API endpoints that allow users to test the HL7 transformation functionality directly, given they have the NND XML payload on hand.
        - Find out more here at Swagger

Installation checklist

1. Helmchart
2. Debezium source connector
3. Case Notification Services
    a. HL7 Parser Service
    b. Data Extraction Service
    c. Case Notification Service

Liquibase info

- NEDSS-NNDSS-Case-Notifications/case-notification-service/src/main/resources/db at main · CDCgov/NEDSS-NNDSS-Case-Notifications

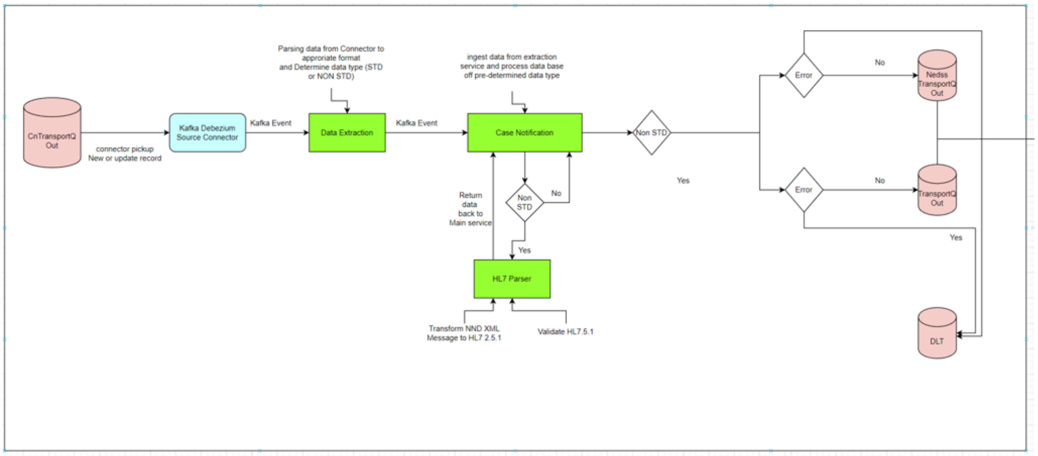
Required environment varaible

- LINK

Project Repos

- GitHub - CDCgov/NEDSS-NNDSS-Case-Notifications

Technical Diagram



CaseNotificatio....drawio
02 Jun 2025, 10:05 pm