Bidirectional Record to Record Sync System Design

Date: 2025-06-07

Version: 1.0

# Table of Contents

1. Introduction

2. Assumptions

3. System Overview

4. Architecture Diagram

5. Data Flow and Mapping

# 1. Introduction

This document provides a detailed design for a bidirectional synchronization system between an internal and external system. It includes assumptions, system overview, architecture diagram, sequence diagram, data flow, conflict resolution strategy, error handling, security considerations, and future enhancements.

# 2. Assumptions

1. Internal System (E.g.: CRM) We have full access.

2. External System (E.g: Marketing) Accessible only from Rest Api.

3. Timestamps are in UTC.

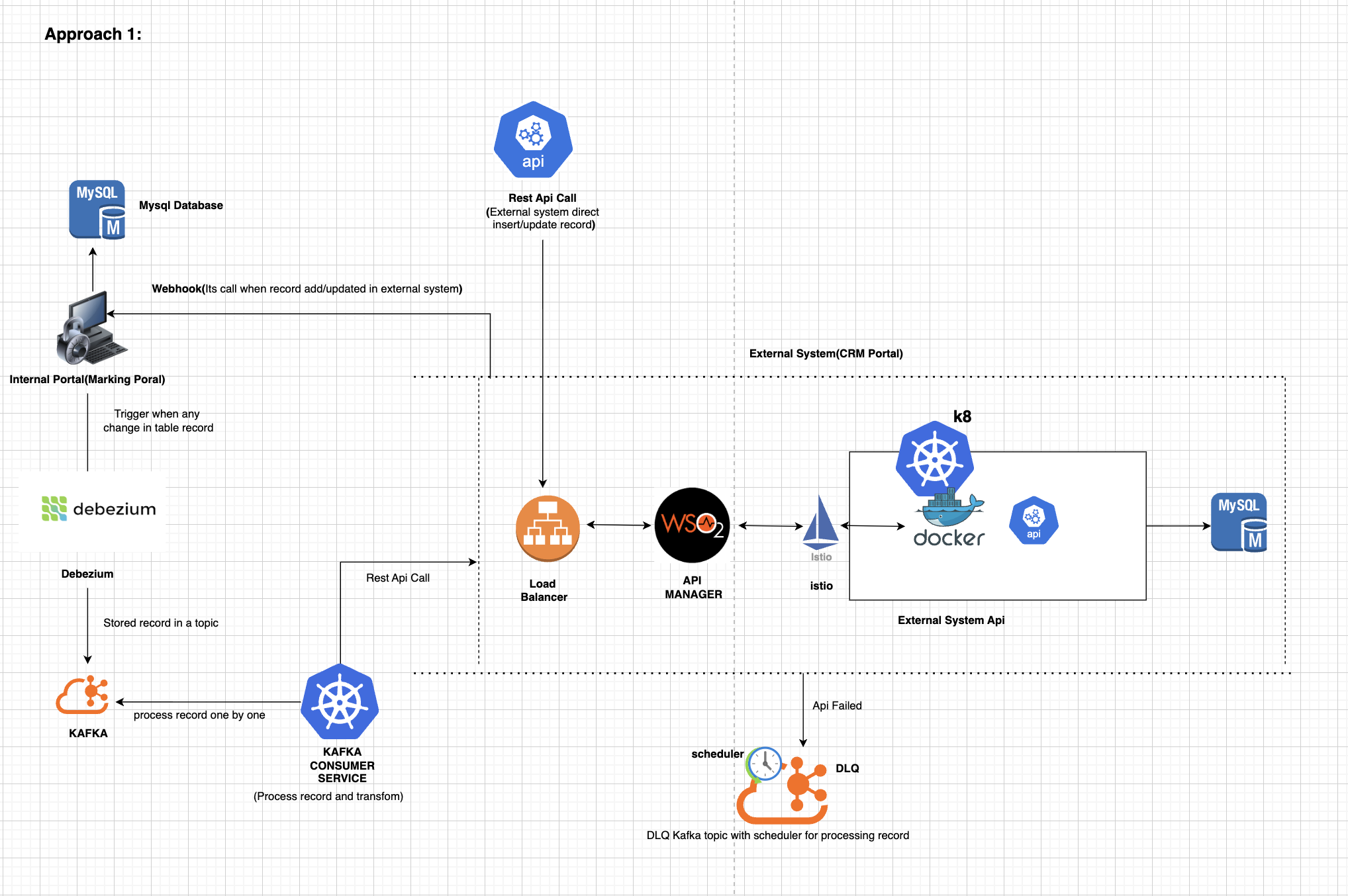
4. External system supports webhook callbacks.

# 3. System Overview

The bidirectional synchronization system ensures that updates made in the internal system are reflected in the external system and vice versa. It handles data transformation, conflict resolution, and logging to maintain data consistency and integrity across both systems.

# 4. Architecture Diagram

The architecture diagram below illustrates the components and data flow of the bidirectional synchronization system.



Component:

Internal System: For now, consider internal system as CRM.

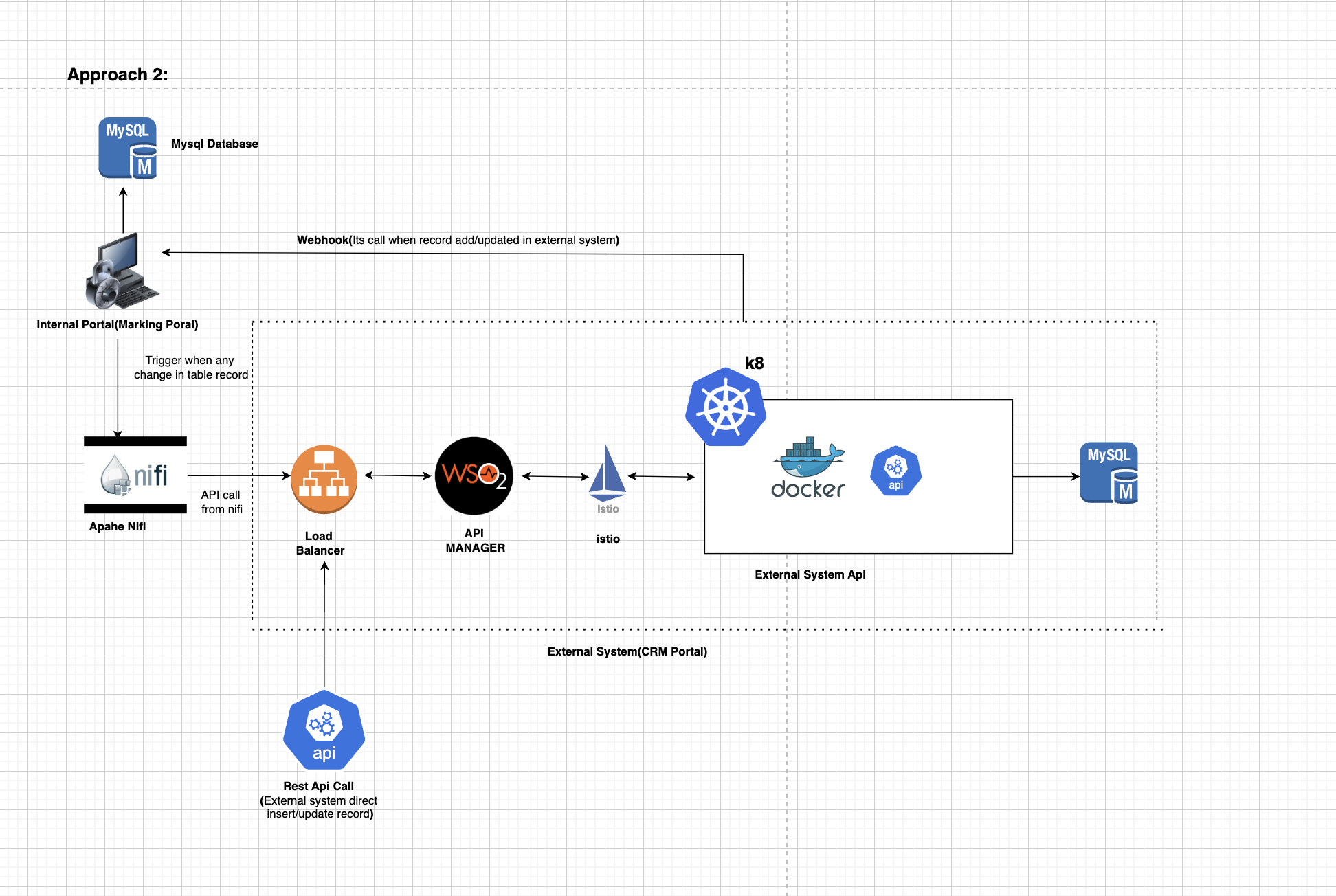
Debezium(Open source) tool: Convert binlog of MySQL to Kafka event message.

Kafka: Messaging layer between internal and external system.

External System: For now, consider external system as Marketing Tool (completely made in microservice architecture)

DLQ: Any failed message stored, and process based on scheduler.

Webhook: if data updated directly to external system then webhook send data to internal system



Component:

Internal System: For now, consider internal system as CRM.

Apache NIFI: Sync Data of MySQL to External System.

External System: For now, consider external system as Marketing Tool (completely made in microservice architecture)

Webhook: if data updated directly to external system, then webhook send data to internal system

# 6. Data Flow and Mapping

This section describes how data is transformed and mapped between the internal and external systems. It includes field mappings and data transformation rules.

Internal System (CRM): Contact table for structure:

CREATE TABLE crm\_customers (

crm\_customer\_id VARCHAR(20) PRIMARY KEY,

first\_name VARCHAR(100),

last\_name VARCHAR(100),

email VARCHAR(255),

phone\_number VARCHAR(20),

last\_updated\_at TIMESTAMP

);

External System (Marketing): Contact table for structure:

CREATE TABLE marketing\_contacts (

    mkt\_contact\_id VARCHAR(20) PRIMARY KEY,

    full\_name VARCHAR(200),

    email\_address VARCHAR(255),

    mobile VARCHAR(20),

    last\_modified TIMESTAMP

);

Central System: Sync table for sync:

CREATE TABLE synced\_records (

syn\_id VARCHAR(50) PRIMARY KEY,

crm\_record\_id VARCHAR(255),

mkt\_record\_id VARCHAR(255),

source\_system VARCHAR(50),

sync\_direction VARCHAR(20),

last\_synced\_at TIMESTAMP,

last\_updated\_at TIMESTAMP,

retry\_count INT,

sync\_status VARCHAR(20),

error\_message TEXT

);

ER Diagram:

A diagram of a customer relationship

AI-generated content may be incorrect.