Architecture and Code Analysis: Social Notifier System

This document provides a structured analysis of the Social Notifier project developed during Hackathon 2017. The goal of this PDF is to explain the architecture, technologies used, database structure, and core logic of the system in a clear and understandable manner. This document can be used as input for an AI-based PDF chatbot.

1. Architecture Overview

The project follows an event-driven integration architecture. System-generated monitoring events from IBM Domino are automatically captured and forwarded to Watson Workspace using REST APIs. The architecture is designed for enterprise alerting and notification.
Architecture Flow: IBM Domino DDM → Social Notifier Plugin → Watson Workspace → Slack or other platforms.

2. Technologies Used

The system is implemented using Java within the IBM Domino environment. It uses OSGi for modular plugin deployment and REST APIs for communication with Watson Workspace. JSON is used as the message format, and HTTPS ensures secure transmission.

3. Database Structure

This project does not use a relational database. Instead, it relies on IBM Domino's document-based database system. Data is stored in the form of documents and forms rather than tables.
Stored information includes monitoring event details, timestamps, alert messages, and configuration data such as Watson Workspace App ID, Secret, and Space ID.

4. Code Logic

The system continuously monitors Domino events using DDM. When an event is detected, the Social Notifier plugin captures the event, converts it into a structured JSON message, authenticates with Watson Workspace, and sends the notification via an HTTP POST request.

5. Conclusion

The Social Notifier project demonstrates an effective enterprise notification solution using event-driven architecture. It integrates IBM Domino with Watson Workspace to deliver automated alerts. This documentation is structured to support AI-based question answering through a PDF chatbot.