Dispute Resolution Engine

Installation Manual

Introduction

Project Context

Alternate Dispute Resolution (ADR) provides a way for parties to negotiate disputes without involving the judicial system. Conventional litigation processes are often costly, time-consuming, and can vary significantly across different companies and domains. By automating the processes involved, using custom workflow generation and NLP enhancements, the Dispute Resolution Engine aims to drastically increase the speed and cost-effectiveness of ADR.

How It Works

The Dispute Resolution Engine (referred to as DRE henceforth) is built and deployed as a series of individual containers managed by Docker. These containers communicate with each other and the internet via an automatically configured container network interface. By default, data inside the containers, unless part of the image itself, is ephemeral. To achieve data persistence, internal directories within the containers are bound ad hoc to the host system's file system—these are known as volumes. This containerized model, managed by Docker Compose, enables flexible configurations and straightforward deployment.

Prerequisites

Required Software

Before proceeding with the installation, ensure that the following software is installed on your system:

- Docker: Required for managing the containerized deployment of the application.
- Loki Plugin for Docker: This plugin is used for logging and monitoring.

Install the Loki Plugin with the following command:

- docker plugin install grafana/loki-docker-driver:2.9.2 --alias loki --grant-all-permissions

Technical Installation

Building And Running

Follow these steps to build and run the Dispute Resolution Engine (DRE):

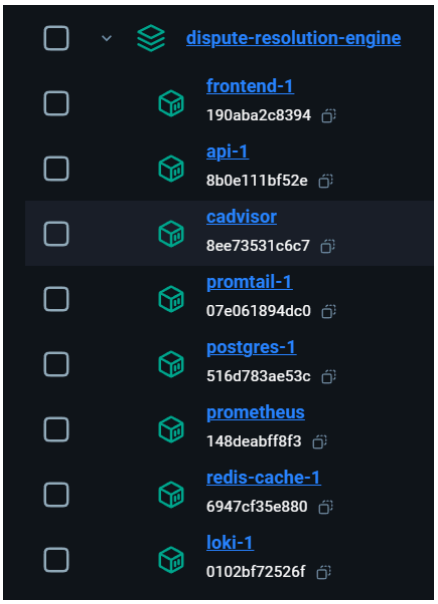
1. Clone the Repository
    a. Clone the project repository from GitHub and navigate to the project directory
- git clone https://github.com/COS301-SE-2024/Dispute-Resolution-Engine
- cd Dispute-Resolution-Engine
2. Build and Run the Project
    a. Use Docker to build and run the necessary containers
- docker compose up -d

This command will automatically download, install, and run the relevant containers needed for the system to operate.

WARNING: The application requires specific environment variable files to function correctly. These files are confidential and must be obtained by contacting the development team.

Verifying Install

If the installation is running correctly, then all the containers defined in the *compose.yaml* file should be shown to be running on the Docker dashboard with all container icons green.



Once that has been checked, one should be able to access the interface from http://localhost:3000 which is where the frontend container is exposed to.