Distributed Systems

July 2023



Smart water/air pollution tracking System

**Xiangnan Liu, x22190562**

Contents

[1 Introduction 4](#_Toc139747457)

[2 Service 1: PollutionSensorService 4](#_Toc139747458)

[2.1 Methods 4](#_Toc139747459)

[2.1.1 RPC Method 1: GetPollutionLevel 4](#_Toc139747460)

[2.1.2 RPC Method 2: SubscribeToPollutionUpdates 4](#_Toc139747461)

[2.1.3 RPC Method 3: GetPollutionHistory 4](#_Toc139747462)

[3 Service 2: WeatherService 4](#_Toc139747463)

[3.1 Methods 4](#_Toc139747464)

[3.1.1 RPC Method 1: GetTemperature 4](#_Toc139747465)

[3.1.2 RPC Method 2: GetHumidity 4](#_Toc139747466)

[3.1.3 RPC Method 3: GetWeatherForecast 5](#_Toc139747467)

[4 Service 3: ControlPanelService 5](#_Toc139747468)

[4.1 RPC Methods 5](#_Toc139747469)

[4.1.1 RPC Method 1: SetDeviceStatus 5](#_Toc139747470)

[4.1.2 RPC Method 2: GetDeviceStatus 5](#_Toc139747471)

[4.1.3 RPC Method 3: StreamDeviceLogs 5](#_Toc139747472)

# Introduction

This proposal outlines the development of an intelligent distributed system for tracking water and air pollution. The objective is to create a comprehensive solution that includes protocols, messages, and a reference implementation for simulating intelligent automated environmental operations. The system will consist of multiple services and devices that communicate with each other using gRPC, and the implementation will be done in Java.

The primary goal of this system is to provide real-time monitoring and analysis of water and air pollution levels. By leveraging intelligent services and devices, we aim to create an efficient and reliable system that can track pollution sources, identify trends, and facilitate proactive measures to mitigate pollution.

# Service 1: PollutionSensorService

The PollutionSensorService is responsible for monitoring pollution levels in the environment. It will provide functionalities such as retrieving current pollution levels, subscribing to real-time updates, and accessing historical pollution data.

## Methods

### RPC Method 1: GetPollutionLevel

This method allows clients to query the current pollution level in a specific location. Clients can specify the latitude and longitude coordinates, and the service will return the pollution level measured at that location.

### RPC Method 2: SubscribeToPollutionUpdates

Clients can subscribe to receive real-time updates on pollution levels in a specific area. The service will periodically send notifications to subscribed clients, providing them with the latest pollution readings.

### RPC Method 3: GetPollutionHistory

This method enables clients to retrieve the historical pollution data for a specific location within a given time range. Clients can specify the start and end dates, and the service will return a collection of pollution readings for that location during the specified period.

# Service 2: WeatherService

The WeatherService provides weather-related information for environmental analysis. It assists in understanding the impact of weather conditions on pollution levels.

## Methods

### RPC Method 1: GetTemperature

This method retrieves the current temperature in a specific location. Clients can specify the latitude and longitude coordinates, and the service will return the temperature at that location.

### RPC Method 2: GetHumidity

Clients can use this method to obtain the current humidity level in a specific location. By specifying the latitude and longitude coordinates, the service will provide the humidity reading for that location.

### RPC Method 3: GetWeatherForecast

This method allows clients to access the weather forecast for a specific location within a given time range. Clients can specify the desired time range, and the service will return the forecasted weather conditions, including temperature, humidity, wind speed, and precipitation.

# Service 3: ControlPanelService

The ControlPanelService is responsible for controlling and managing the environmental devices in the system. It provides functionalities for device status management and real-time monitoring.

## RPC Methods

### RPC Method 1: SetDeviceStatus

Clients can use this method to set the status (on/off) of a specific device in the environment. Clients will provide the device identifier and the desired status, and the service will update the device accordingly.

### RPC Method 2: GetDeviceStatus

This is a **bidirectional streaming** RPC. This method retrieves the current status of a specific device in the environment. Clients will provide the device identifier, and the service will return the current status of that device.

### RPC Method 3: StreamDeviceLogs

This is a bidirectional streaming RPC that allows clients to receive real-time logs and updates from the devices in the environment. Clients can establish a persistent connection to the service and receive continuous updates on device status changes, sensor readings, and other relevant information.

To demonstrate the implementation and provide a user-friendly interface for interacting with the system, a client GUI will be developed. The GUI will serve as the main controller, enabling users to discover and utilize the services and devices in the system. It will include features for discovering pollution levels, subscribing to updates, accessing historical data, monitoring weather conditions, controlling devices, and receiving real-time device logs.

In conclusion, this proposal presents a detailed plan for the development of an intelligent distributed system for water and air pollution tracking. By implementing the proposed services and functionalities, we aim to create a robust solution that contributes to environmental monitoring and pollution control efforts.