VERSION 1.2

29/01/2025

Green Score Meter

An IoT Based System For Sustainable Smart Cities

GitHub Repository: https://github.com/sam-a-d/GreenScoreMeter

PRESENTED BY: ABDULKADIR ABUBAKR,

MD ADBUS SAMAD, NABIHAH CHAUDHRY

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

PROF. DAVIDE DI RUSCIO

Table of Contents

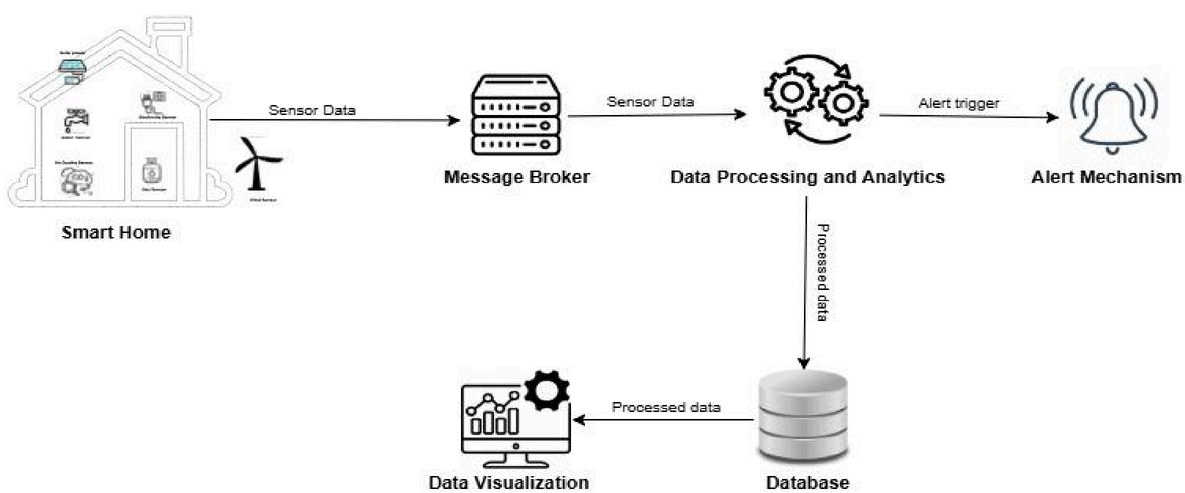
Project Outline

Project Overview

A sustainable Smart City is essential to address the pressing challenges of urbanization, climate change, and resource depletion. By managing resource usage and replenishment in smart homes, cities can reduce greenhouse gas emissions, conserve resources, and enhance resilience against environmental risks. Sustainability in Smart Cities ensures economic growth, social equity, and environmental protection, creating livable spaces that meet the needs of current and future generations.

One of the major challenges in building smart cities today is the ability to measure and monitor resource utilization in households. The reliance of households on fossil fuels and non-renewable sources of energy contributes to greenhouse gas (GHG) emissions. Fossil fuels are still the primary energy source for many household functions, including electricity generation, heating, and cooking, contributing to high levels of carbon dioxide (CO_2) and methane (CH_4) emissions. In urban areas, where a large portion of the world's population resides, homes are significant consumers of energy, and this dependency on fossil fuels drives up greenhouse gas emissions. In fact, urban centres are responsible for nearly 70% of global GHG emissions, much of which stems from inefficient resource use in residential settings. Addressing this problem is critical for reducing cities' carbon footprints and achieving global climate goals.

The Green Score Meter is an IoT-based system designed to assess a household's energy sustainability by monitoring energy consumption, renewable energy production, and pollution levels. It assigns a "green score" using a specialized algorithm, encouraging eco-friendly practices through financial incentives such as tax exemptions. By using IoT and sensor technologies, it monitors energy consumption, production, and pollution levels that reflect a household's energy sustainability. This score is used to offer tax exemptions as financial incentives, motivating households to increase renewable energy usage and maintain eco-friendly practices. This approach not only promotes energy efficiency but also promotes community-wide awareness and participation in reducing greenhouse gas emissions, thus addressing critical sustainability challenges in smart cities.



Informal Description of the System

Project Objectives

The main aim of this project is to develop an IoT-based green score meter that evaluates household energy sustainability by monitoring energy consumption, renewable energy production, and pollution levels through sensors. The project aims to encourage eco-friendly practices through data-driven insights. This work also has the following general objectives:

- Sense resource consumption and resource production data using various sensors from households and send them to the server.
- Calculate the green score for each household.
- Develop a web-based dashboard for authorities to display interactive statistics and visualizations.
- Implement an alarm mechanism to trigger notifications under defined conditions, enhancing real-time awareness and responsiveness.

Green Score Meter

The IoT-powered Green Score Meter evaluates energy consumption, energy generation, and air pollution caused by a household. It processes this data using the Green Score Algorithm to determine the household's green score.

Green Score Calculation

The green score calculation used in this project is based on the work originally done by *K et al. (2023*). To be aligned with our project's goal we slightly modified the green score calculation process. The system has two main units to collect data, a) Renewable energy production unit and b) Resource utilization unit. The former collects and sends household energy production data; a household can produce energy in various ways such as from solar, wind, and hydrological sources. The resource utilization unit is responsible for collecting energy usage data from each household. All the collected data from both units are then sent to the Green Score Meter for calculation by the Green Score Algorithm.

In the green score algorithm, the economic utilization and required level of production of resources will give positive credits and excess utilization of resources, and the cause of pollution will give negative credits.

The following equation describes the calculation of the monthly green score $GSM(M_i)$ of the month M_i by adding all the resource scores and then subtracting them from the air pollution score.

$$GSM(Mi) = ES(Mi) + WS(Mi) + NGS(Mi) + COS(Mi) + RES(Mi) - APS(Mi)$$

- ES – Electricity Score
- WS - Water Score
- NGS – Natural Gas Score
- COS – Crude Oil Score
- RES - Renewable Energy Score
- APS – Air Pollution Score

By averaging all the twelve-month green scores, the green score of a year (GSY) can be calculated.

Calculation of Resource Utilization, Resource Production, and Air Pollution Score

First of all, the authority sets the per capita resource utilization limit for a particular area. If a household utilized less than or equal to that limit, the score would be set to 1. Suppose the limit for natural gas utilization is set to L and a household H utilizes R amount of natural gas, the NGS score will be 1 if R <= L. This criterion is true for ES, WS, NGS, and COS utilization. For APS, if the total air pollution is greater than the limit set by the authority, it will result in APS = 1, otherwise, APS will be 0.

Secondly, households may produce resources. If a household produces or reuses at least one-fourth of the amount they consume, they will get another 1 point for a particular energy score. For instance, a household H consumes 100 units of natural gas for a month, and it produces 25 units of biogas at the same time, so the NGS score will get another 1 point. This criterion applies to WS, NGS, and COS.

Calculation of Renewable Energy Score (RES)

Renewable energy score is calculated based on how much a household produces in comparison to how much it consumes each month. A household can produce renewable energy by three means, a) Solar source, b) Hydrological source, and c) Wind source. The RES is the sum of the scores of different sources like the following:

RES = Solar Score + Hydrological Score + Wind Score

If a particular source produces more than 25% of how much energy a household consumes, the score for that source will be set to 1. For example, in household H, solar energy production is 30%, hydrological production is 25% and wind production is 15% of the total energy usage, then the RES will be, RES = 1 + 1 + 0 = 2

On the other hand, if a household produces >= 75% of the energy cumulatively from all sources that it consumes, the RES score will be 3. For example, if the production is 50%, 25%, and 10% from solar, hydrological, and wind source respectively, even if the wind source contributes only 10%, however, since the overall energy production is more than 75% the RES score will set to +3.

Calculation of Final (Optimal) Green Score

The green score of a year $GSY(Y_i)$ is the average of the monthly green scores $GSM(M_i)$ over the year; therefore, can be calculated by the following formula:

$$GSY\left(Y_i\right)\frac{\sum GSM\left(M_i\right)+GSM\left(M_{i+1}\right)+\ldots\ldots+GSM\left(M_n\right)}{n}$$

To calculate the final green score of a year (the optimal green score), the amount of yearly groundwater consumption and rainwater harvesting is taken into consideration. If the groundwater consumption is greater than the rainwater harvesting in the same year, then the optimal green score $OGSY(Y_i)$ will be equal to $GSY(Y_i) - 1$, otherwise it will be equal to $GSY(Y_i) + 1$.

Functional and Non-Functional Requirements

Functional Requirements

To satisfy the objectives of the project, the system must satisfy the following functional requirements.

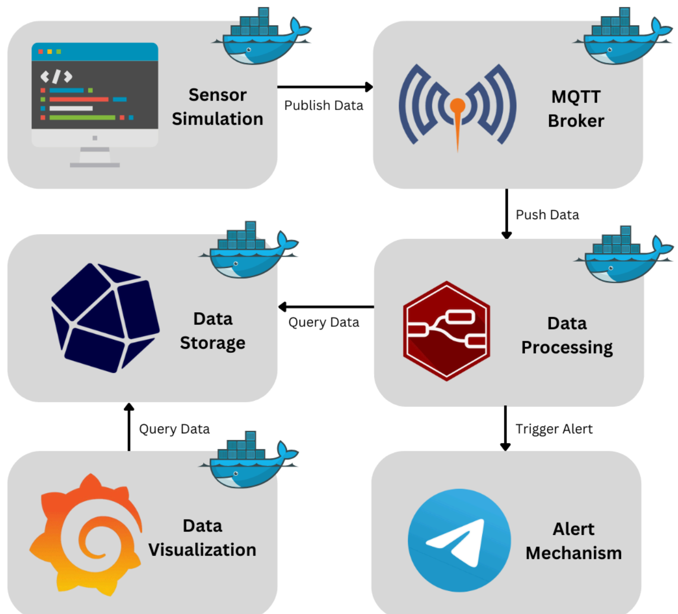
Requirement	Description
Data Collection	The system should support sensors deployed in multiple homes to monitor the usage and replenishment levels of electricity, water, natural gas, crude oil, and other renewable energy sources.
Green Score Calculation	The system shall calculate the green score of a household at Admin-defined intervals, such as weekly, monthly, or yearly.
Data Visualization	The system shall include a dashboard presenting aggregated values, trends, and patterns of resource usage and replenishment.
Reporting	The system shall send green score reports to households at user-defined intervals.
Notifications	The system shall be configurable to send notification to households when an anomalous spike in resource usage value is detected for the household.

Non-Functional Requirements

Requirement	Description
Usability	The system shall provide an intuitive user-friendly interface that allows relevant authorities to access data, configure intervals, and generate reports with ease.
Interoperability	The system shall integrate seamlessly with various third-party sensors and monitoring devices used for electricity, water, and renewable energy tracking.
Scalability	The system shall allow for the addition of new resource types (e.g., solar, wind energy) or metrics without requiring significant architectural changes.
Reliability	The system shall perform regular data backups to prevent loss of critical household and environmental data.
Performance	The system shall process and update resource usage data within 5 seconds of receiving input from sensors. The dashboard shall load and display aggregated data, trends, and patterns within 2 seconds for datasets up to 1 GB in size. The system shall send green score reports to households within 1 minute of a scheduled interval.

System Architecture

The system architecture of this project follows a microservices-based design, where each component operates independently within Docker containers, ensuring scalability and modularity. The data-producer microservice generates and publishes simulated sensor data using MQTT, which is managed by the Eclipse Mosquitto broker. This data is then stored in InfluxDB, a time-series database optimized for handling continuous streams of sensor readings. Grafana retrieves and visualizes this data, providing real-time monitoring through interactive dashboards. Node-RED facilitates data processing and automation by integrating with various services. Docker is used to containerize and orchestrate these components, ensuring seamless deployment and communication across the system. A high-level system architecture is presented in the diagram below.



This system employed microservices using Docker. Every single component is dockerized and deployed independently. The microservices are as follows:

1. Data Producer
2. MQTT Broker
3. Node-RED
4. InfluxDB
5. Grafana

The description of each container is discussed in the following sections.

Data Producer Microservice

The data-producer microservice generates realistic readings using two Python scripts. The Sensor class defined in simulator.py is responsible for realistic data generation. It includes functions for both resource consumption (e.g., Electricity, Water, Natural gas) and renewable energy production (e.g., Solar production, Hydrological production). The data-producer.py uses this data, organizes it into MQTT topics and publishes it to an MQTT broker. Furthermore, the data producer also implements threading to handle the simultaneous publication of data from multiple simulated homes and areas, ensuring efficiency and scalability. The paho-mqtt library enables data producers to publish simulated sensor data to the MQTT broker. It facilitates the creation of MQTT clients that establish a connection with the broker and publish sensor readings in real time.

The sensors include:

- Electricity
- Water
- Natural gas
- Air pollution
- Crude oil
- Solar production
- Hydrological production
- Wind production
- Biogas production

The MQTT topic is created in this format:

sensor/{sensor_name}/area_{area_id}/home_{home_id}

Extensibility and Usability of Data Producer Microservice

Configurations of the data producer for the MQTT broker are separated and stored in simulatorConf.ini file to manage configuration that makes the system easy to use and easy to set up.

The homes.csv file (utilized by multiple microservices) contains necessary information related to the areas, home, and contact information. The data producer microservice uses this data to dynamically simulate sensor values. This implementation makes the system easily adaptable and improves the usability and flexibility for the end users.

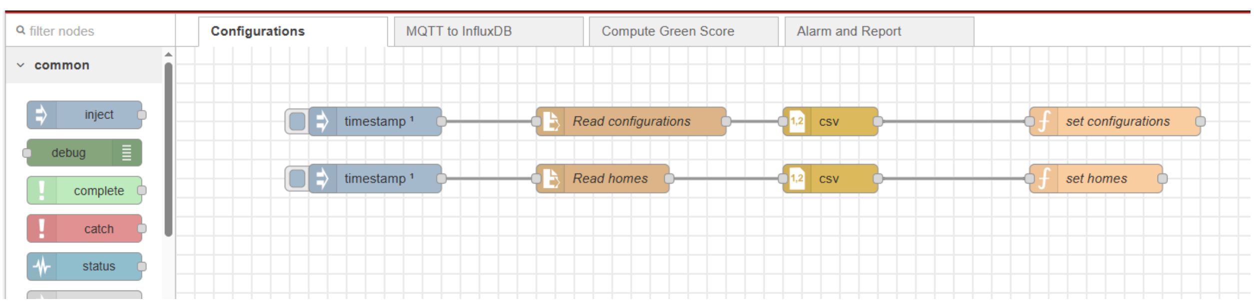
Message Broker Microservice

The MQTT broker, Eclipse Mosquitto, serves as the central hub for message communication between the data producer microservice and other system components. It receives sensor data published by the data producer under structured topics and ensures its reliable delivery using MQTT protocol to the subscribed clients (Node-RED). Mosquitto's lightweight and efficient design makes it ideal for handling the real-time flow of simulated sensor data across multiple homes and areas, enabling seamless data distribution throughout the IoT system.

Data Processing Microservice (Node-Red)

Node-RED serves as the central processing middleware for the Green Score Meter. It plays a pivotal role in managing the overall functionality of the system and is responsible for several key tasks. These include handling system configurations, processing and storing data, computing the Green Score, and triggering alarms based on predefined conditions. Through its robust integration capabilities, Node-RED ensures seamless data flow and efficient operation across all components of the Green Score Meter.

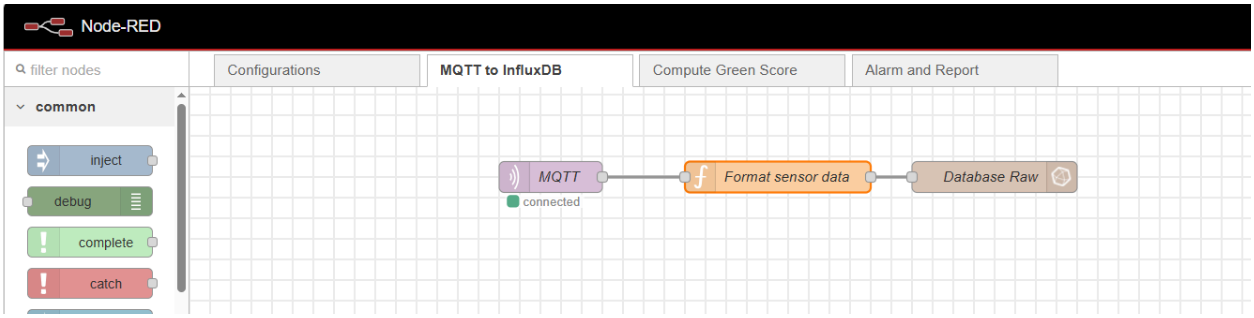
Managing Configuration



To support flexible threshold configurations and facilitate the onboarding of new households, the Green Score Meter utilizes configuration files to manage system-wide settings. Node-RED reads these configurations and the associated household data using a file node then processes the data using a CSV node and finally stores them as globally accessible variables that are used throughout data processing, calculations, and storage. This approach ensures that key parameters such as usage limits, report frequency, and Green Score computation intervals are not hardcoded into the system. By leveraging dynamic configuration, the system is easily adaptable and scalable, ensuring easy accommodation of new sensors, locations, and households without the need for complex reconfiguration.

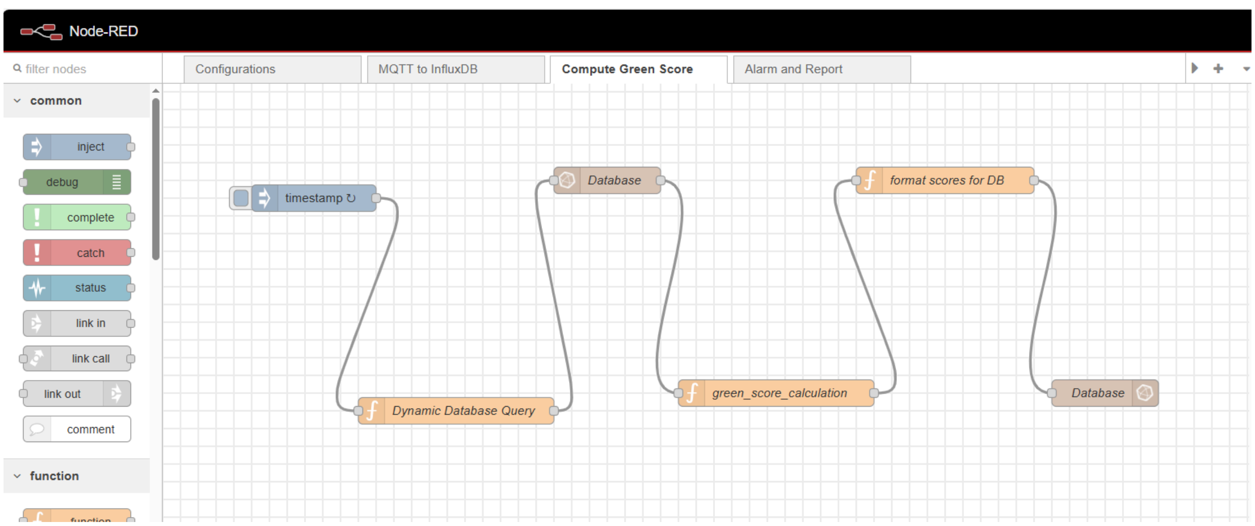
Processing and Storing Sensor Data

In this workflow, Node-RED is utilized as the primary processing and storage mechanism. It subscribes to the Mosquitto MQTT broker, receiving real-time data from various topics. Node-RED then processes this incoming data, performing necessary transformations or computations such as topic parsing and data tagging, before storing the results in the InfluxDB database.



Computing Green Score

A flow is also implemented in Node-RED to calculate rolling consumptions and green scores dynamically, based on configurations set beforehand. It starts with a timestamp that acts as a trigger every second. This trigger runs a function that creates a custom database query using Flux. The query is tailored to needs — like pulling data for just the last two days if weekly reporting is required. This ensures the system is efficient and only works with the data it actually needs.

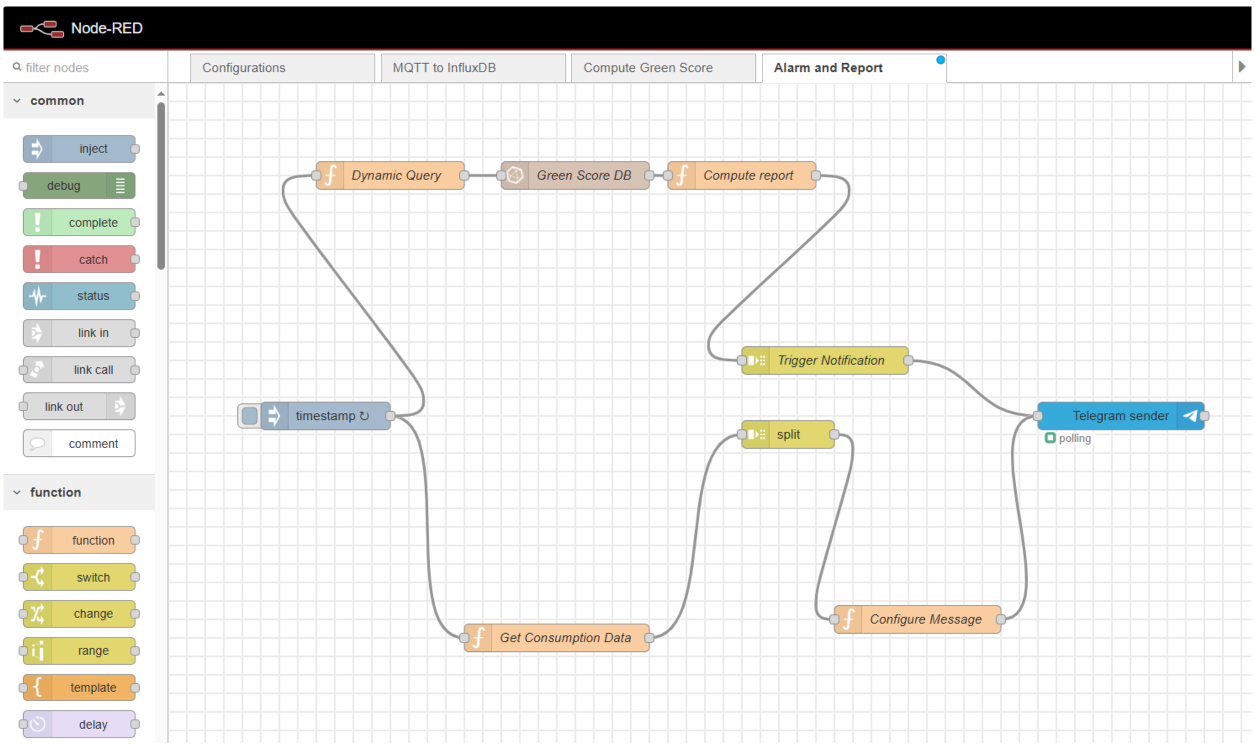


Another function takes over to calculate green scores. First, it looks up the houses in the system and creates a list of the resources being tracked, like water, electricity, air pollution, and renewable energy production (solar, wind, etc.). It then pulls the sensor readings and adds up the values for each resource consumed or produced by every house.

It increases the score if a house uses less of a resource than the allowed limit (like staying under a water or electricity threshold). It also rewards houses that rely on renewable energy — if solar, wind, or hydropower generates enough of their electricity (e.g., over 70% or 25%, depending on the case), the score goes up.

When all scores are calculated, the results are formatted for storage. If it's the scheduled report day (like the end of the week or month), the system saves both the green scores and the detailed consumption data for each house. On other days, it just saves the consumption data. This information is structured in a way that's easy to visualize, so you can quickly see how each house is performing and how sustainable their resource use is.

Alarm and Reports



This Node-RED flow automates two key functions: sending utilization alerts and generating green score reports.

Utilization alerts are sent daily to households that exceed 80% of their allowed resource consumption before a predefined time. This helps residents stay aware of their usage and take corrective action before they fully deplete their allocated resources.

Green score reports, on the other hand, are sent at the end of a specified reporting period—whether weekly, monthly, or yearly. These reports summarize how well each household is performing in terms of sustainable resource usage.

The top section of the flow is dedicated to generating these reports. It dynamically queries the green score database, computes the report, and sends it via Telegram using the Telegram sender node. The lower section of the flow handles alerts, ensuring that consumption thresholds are monitored in real time.

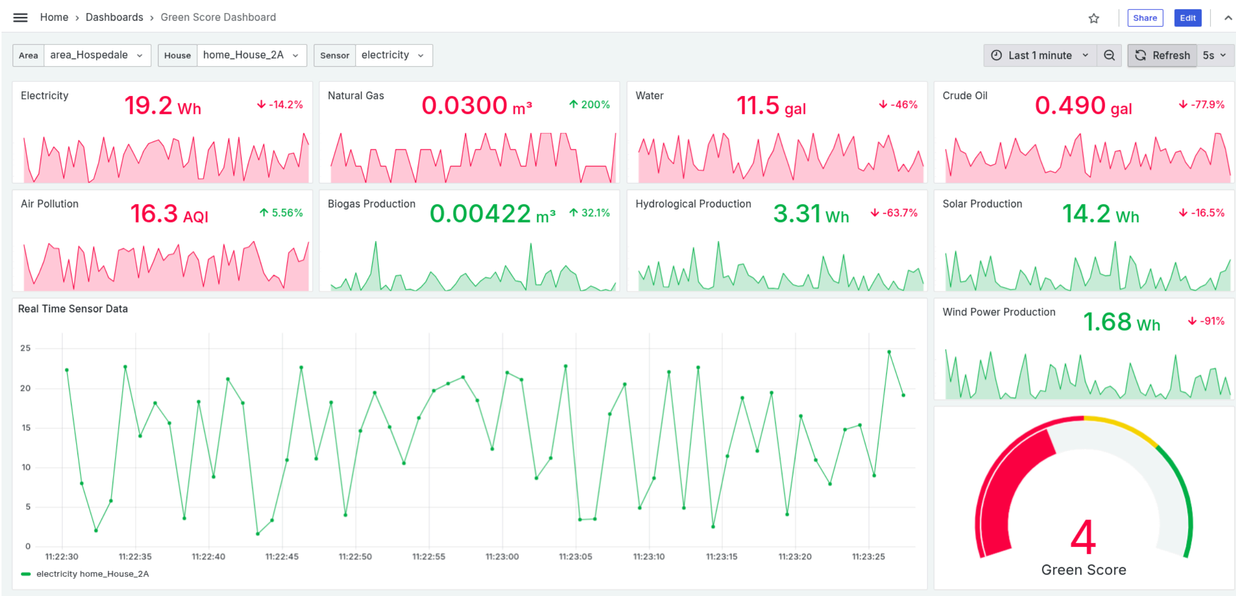
Both sections rely on a split node to process each household separately. For alerts, the system retrieves the latest consumption data, checks if any household has exceeded the threshold, and sends a notification if necessary. The report section simply verifies if it's the scheduled reporting time, fetches the stored green scores from the database, and distributes the results accordingly. Everything is seamlessly integrated using Node-RED, making the process efficient and fully automated.

Data Storage Microwave (InfluxDB)

In this project, InfluxDB is implemented as the dedicated time-series database for storing and managing the continuous flow of simulated sensor data generated by the data producer. Each data point, tagged with precise timestamps, is stored in InfluxDB to capture trends over time for parameters such as electricity consumption, water usage, air pollution, and renewable energy production. Its high performance and optimized storage structure for time-series data enable efficient querying and retrieval of large volumes of data. This integration ensures seamless data persistence, providing a reliable foundation for real-time monitoring and visualization through tools like Grafana.

Data Visualization Microservice (Grafana)

The Green Score Dashboard in Grafana is implemented with multiple panels and configurations to effectively monitor and visualize IoT-based environmental and energy-related metrics. The dashboard uses InfluxDB as its data source, with queries dynamically filtering data based on user inputs for areas, houses, and sensors. These inputs are configured as templating variables, allowing users to seamlessly filter and customize the data displayed in the dashboard.



Variables and Filtering Options

The dashboard supports three main templating variables:

1. Area: Filters data based on the geographical region. The schema.tagValues query retrieves the list of available areas from the gs_sensors bucket in InfluxDB.
2. House: Dynamically filters data for specific houses within the selected area. This is achieved using a predicate that ties the house to the selected area.
3. Sensor: Enables users to select the type of sensor data to display, such as electricity, natural gas, water, or renewable energy metrics. The schema.fieldKeys query retrieves available sensor types.

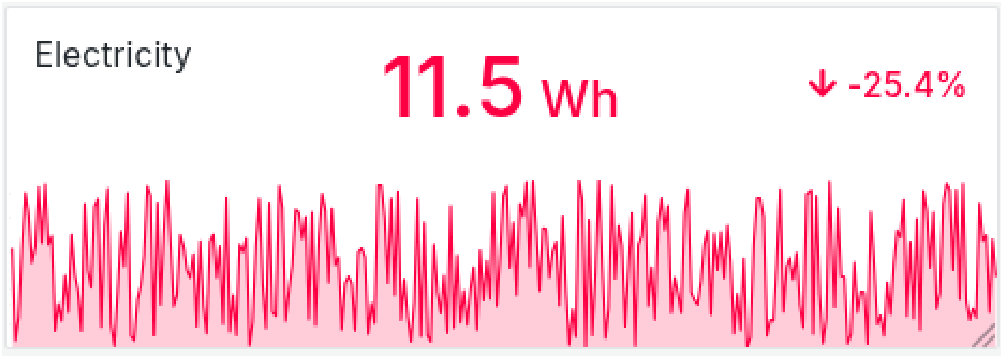
These variables make the dashboard interactive and adaptable, allowing users to drill down into specific datasets

Data Visualization

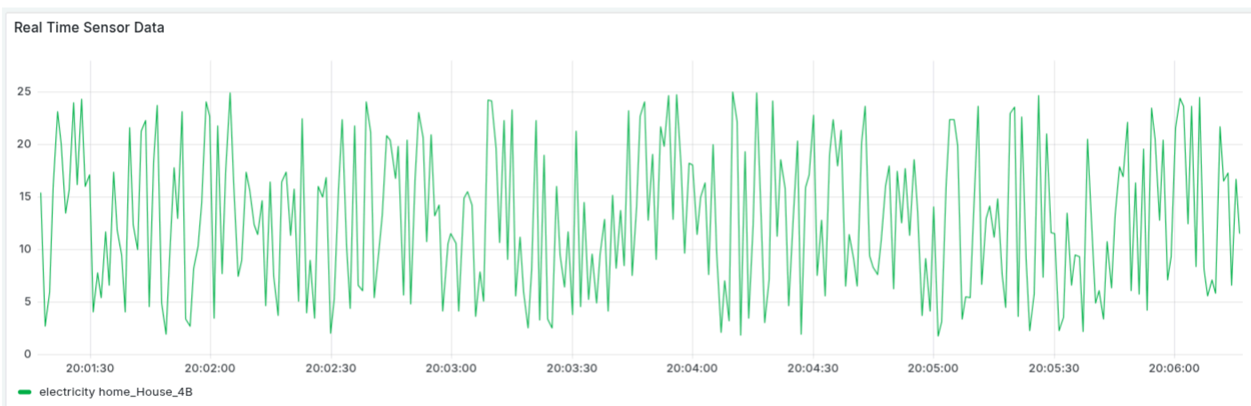
The dashboard includes multiple stat panels, a time series panel, and a gauge panel, each are described below.

1. Stat Panels - Each stat panel represents a key metric, such as electricity, natural gas, water usage, renewable energy production, or air pollution levels. These panels display the latest values that are coming from the sensor with the corresponding unit value and also a time series trend for quick interpretation. In addition, a percentage change of values compared to the previous record is also displayed with an appropriate arrow. For instance:
    - Electricity consumption is measured in watt-hours (wh).
    - Water usage is displayed in gallons, and air pollution is measured using the Air Quality Index (AQI).

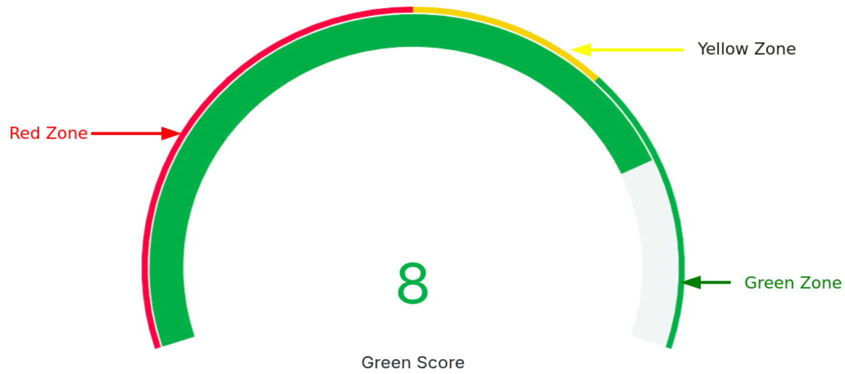
- Renewable energy metrics like solar, wind, and hydrological production are also shown with units such as watt-hours or cubic meters.



2. Time Series Panel - This panel, titled "Real-Time Sensor Data," visualizes trends over time for the selected sensor (changeable from the filter). Users can analyze historical data by defining the time range and aggregating values using mean functions over a window period.



3. Gauge Panel - Green Score is displayed by the gauge that shows a number between 0 and 10. The value ranges are divided into 3 zones, red zone (below 5), yellow zone (5-6), and green zone (7 or above).



Dashboard Layout and Interaction

The panels are laid out in a grid-based structure for clarity and ease of access:

- Resource Consumption metrics like electricity consumption, natural gas consumption, and water consumption are arranged in individual stat panels and they are colored red.
- Resource Production metrics (solar, wind, and hydrological production) are highlighted using the green color.
- Interactive dropdown menus for Area, House, and Sensor enable users to focus on specific data subsets dynamically.

Initial Application Setup

Obtaining a Telegram Bot Token

To enable Telegram notifications in the Green Score Meter project, you need to create a Telegram bot and obtain a bot token. Follow these steps:

1. Open Telegram and Start a Chat with BotFather
    a. Open the Telegram app and search for @BotFather.
    b. Start a chat and type /start to begin.
2. Create a New Bot
    a. Type /newbot and press enter.
    b. BotFather will ask for a name for your bot—enter a suitable name (e.g., GreenScoreNotifier).
    c. Next, you'll be asked to choose a unique username for the bot (must end in bot, e.g., GreenScoreMeterBot).
3. Receive the Bot Token
    a. After successfully creating the bot, BotFather will send you a message containing a token.

Copy this token for further use.

Integrating the Token to Nodered

To use the telegram token with the system, you need to open the environmental file (.env) located in the root folder of the application and replace the text `your_token_here` with the actual token you generated from the [previous step](). Please do not alter the variable name `TELEGRAM_TOKEN`. Finally, the environmental file (.env) might look similar as follows:

```
TELEGRAM_TOKEN=8102941795:AAFhYubAyxxxxxxxxxxl82rZHb7uVccW3q0
```

Running the Project

To run the project, navigate to the `docker-compose.yaml` file and run `docker compose up` or `docker-compose up` command. More detailed instructions are available in the [project's ReadMe]().

Conclusion

The Green Score Meter project presents an innovative approach to addressing the critical challenges of sustainability in urban areas. By harnessing IoT technologies and data-driven algorithms, the system provides an effective mechanism for monitoring, evaluating, and improving energy sustainability at the household level.

Through real-time monitoring, accurate scoring, and interactive visualizations, the project empowers individuals and authorities to make informed decisions that promote sustainable living. The incentivization framework further encourages the widespread adoption of renewable energy and resource-efficient practices, fostering a culture of environmental responsibility.

As cities continue to struggle with the impacts of urbanization and climate change, the Green Score Meter offers a scalable and impactful solution that aligns with global climate goals. Its implementation represents a step forward in building smarter, greener, and more resilient communities for present and future generations.

References

K, M. K., R, M., S, K., & S, A. E. (2023). An IOT based environment conscious green score meter towards Smart Sustainable Cities. Sustainable Computing: Informatics and Systems, 37, 100839. https://doi.org/10.1016/j.suscom.2022.100839