**Methodology**

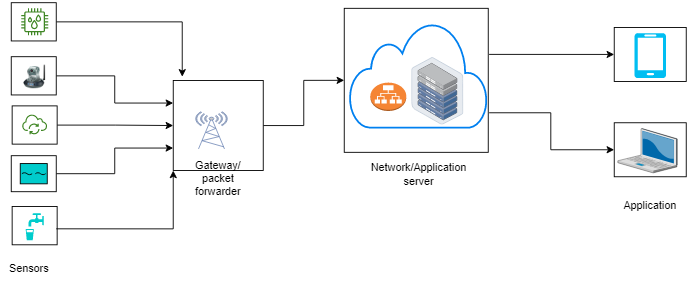
**Needs assessment.**

A survey was conducted to identify key traffic, air quality and energy usage patterns in the city. Existing infrastructure was analysed, and data on traffic flow, air quality and energy consumption were collected from various sources.

**Procedure:**

1. Identify the Network Requirements: The first step is to determine the network requirements. This includes identifying the locations of the devices that will be used for monitoring and management, the data rate and frequency of transmission, and the number of devices that will be deployed.
2. Choose the LoRaWAN Gateway: The next step is to select the LoRaWAN gateway that will be used to connect the devices to the network. The gateway is responsible for receiving data from the end devices and forwarding it to the server. The gateway should be selected based on its coverage area, number of channels, and data rate.
3. Select the End Devices: After selecting the gateway, the next step is to choose the end devices that will be used for monitoring and management. These devices should be selected based on their compatibility with the gateway and the requirements of the use cases.
4. Develop the Software and Analytics: The next step is to develop the software and analytics that will be used to manage and analyse the data collected by the end devices. This includes developing dashboards, alerts, and other tools that will help manage the smart city.
5. Deploy the Network: Once the software and devices are ready, the next step is to deploy the network. This includes installing the gateway and devices at the identified locations, configuring the network settings, and testing the network to ensure that it is functioning correctly.
6. Monitor and Manage the Network: After the network is deployed, the next step is to monitor and manage it to ensure that it is functioning properly. This includes monitoring the network performance, troubleshooting any issues that arise, and making any necessary changes to the network configuration.

**Architectural and Flow Diagram**

****

**Fig 1:** Architectural diagram

Fig 2: Flow chart

**Resource requirements**

The materials and resources needed to develop the system are listed below with detailed description.

**End devices**

The end devices have different set of sensors on them that collect data. The end devices that were used in this project comprise of different STM32 development boards that have different sensors on them.

1. NUCLEO-L073RZ development board.

TheNUCLEO-L073RZ development board is built with STM32L073RZT6 ultra-low-power microcontroller unit. Its memory technology is based on Arm® Cortex®-M+32MHz. It has 192kbyte flash memory and a 20kbyte SRAM. It supports Arduino™ Uno V3 and ST morpho connectors. It has an embedded ST-LINK/V2-1 debugger and programmer. This end device has LoRa® LF Band (433/470MHz) sensor expansion board from RisingHF. Its expansion board module is RisingHF RHF0M003-LF20 low-power long-range LoRaWAN that is based on STM32L071 and Semtech SX1278 transceiver. It has four set of sensors:

* Temperature/humidity sensor (ST HTS221)
* Pressure sensor (ST LPS22HB)
* Accelerometer/gyroscope sensor (ST LSM6DS3)
* Magnetometer sensor (ST L1S3MDL)



Fig 3: An image of the NUCLEO-L073RZ development board.

1. STM32F7691-DISCO

The 32F769IDISCOVERY Discovery kit is a complete demonstration and development platform for STMicroelectronics Arm® Cortex®‑M7 core-based STM32F769NI microcontroller. The Discovery kit enables a wide diversity of applications taking benefit from audio, multi‑sensor support, graphics, security, video, and high‑speed connectivity features. The ARDUINO® connectivity support provides unlimited expansion capabilities with a large choice of specialized add-on boards.



1. Fig 4: An image of the STM32F7691-DISCO

**Gateway devices**

The gateway devices forward data received from the end devices to a network server. The gateway devices used in this project are as follows:

1. NUCLEO-F746ZG development board

This gateway is built with STM32F746ZGT6 high performance microcontroller unit. Its memory technology is based on Arm® Cortex®-M7 217MHz. It has 1Mbyte flash memory and 320kbyte SRAM. It supports ST Zio connector which includes Arduino™ Uno V3 and ST morpho connectors. It also supports Ethernet 10/100Mbps and a USB OTG user connectivity. It has an embedded ST-LINK/V2-1 debugger and programmer. The gateway expansion board is based on LoRa LF band (433/470MHz). It has a Semtech SX1301 LF baseband data concentrator.



**Software**

1. STM32CubeIDE

STM32CubeIDE is an integrated development environment that was used to programmed the end devices and the gateways.

1. Tera Term

Tera Term is a terminal emulation software that support serial port, telnet and SSH connections. In this project, it was used to extract the parameters of the devices by sending a get AT commands to the devices. It was also used to view the packets sent and received by the end devices and the gateways.

**Tools**

* Personal computer
* USB type-A and Micro-B cables
* Ethernet with internet access

Develop a LoRaWAN-based solution:

Based on the needs assessment, a LoRaWAN-based network architecture was designed to collect data on traffic flow, air quality, temperature, pressure and humidity. Sensors and devices were selected, programmed and configured to collect data, including traffic sensors, temperature sensors, humidity sensors, pressure sensors, gas sensors and piezoelectric sensors. Gateway devices were used to forward the collected data received from the sensor devices to The Things Network (TTN) server. Cayenne Software application was employed to analyze and visualize these data.

**Network server setup and device enrolment**

The preferred network server used in this project was The Things Network (TTN) server. A TTN account was created on website at [www.thethingsnetwork.org](http://www.thethingsnetwork.org). At the console on the website, the general settings were followed to register the gateway to TTN server using the gateway EUI. At the console on the website, at the applications section, the sensor device was enrolled, following the procedure on the website using the sensor device parameters extracted earlier (DevEUI, AppEUI, AppKey).

**Setting up Cayenne application server**

On myDevices website, at <https://mydevices.com/>, myDevices Cayenne account was created. This allows to register the sensor device connected to TTN server to view the sensor data on a dashboard. The end device was registered by providing its parameters (DevEUI).

**Testing solution**

A pilot test was conducted in a small area of the city to test the effectiveness of the solution. Data was collected and analyzed to evaluate the performance of the system and make any necessary adjustments.

**Implementation and monitoring of solution**

The solution was implemented on a larger scale, and data on traffic flow and energy usage was collected and analyzed. Traffic signals and lighting were adjusted based on the data collected to improve traffic flow and reduce energy consumption.