

Android – ESP32 Data Exchange

Application for data exchange between Android - ESP32
with BLE

Alessio Amatucci

Federica Di Giacomo

Alessandro Paniccià



SAPIENZA
UNIVERSITÀ DI ROMA

Overview of our project.

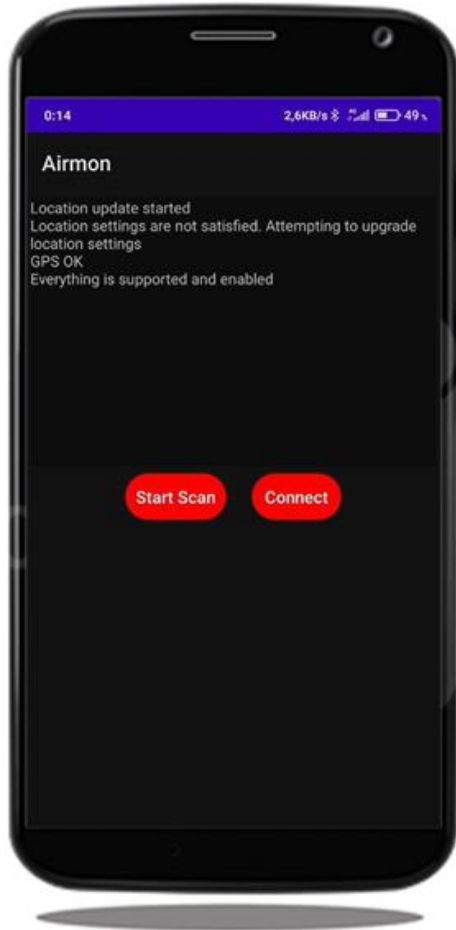
- The purpose of our project is to implement a Point-to-Point communication between an Android mobile phone and an ESP32
- This communication is made using BLE
- In this first phase, our mobile will act as a client and the ESP32 as a server

Startup

At startup, the server advertises, waiting to connect, while the mobile phone scans its surroundings to detect whether or not a server is present.



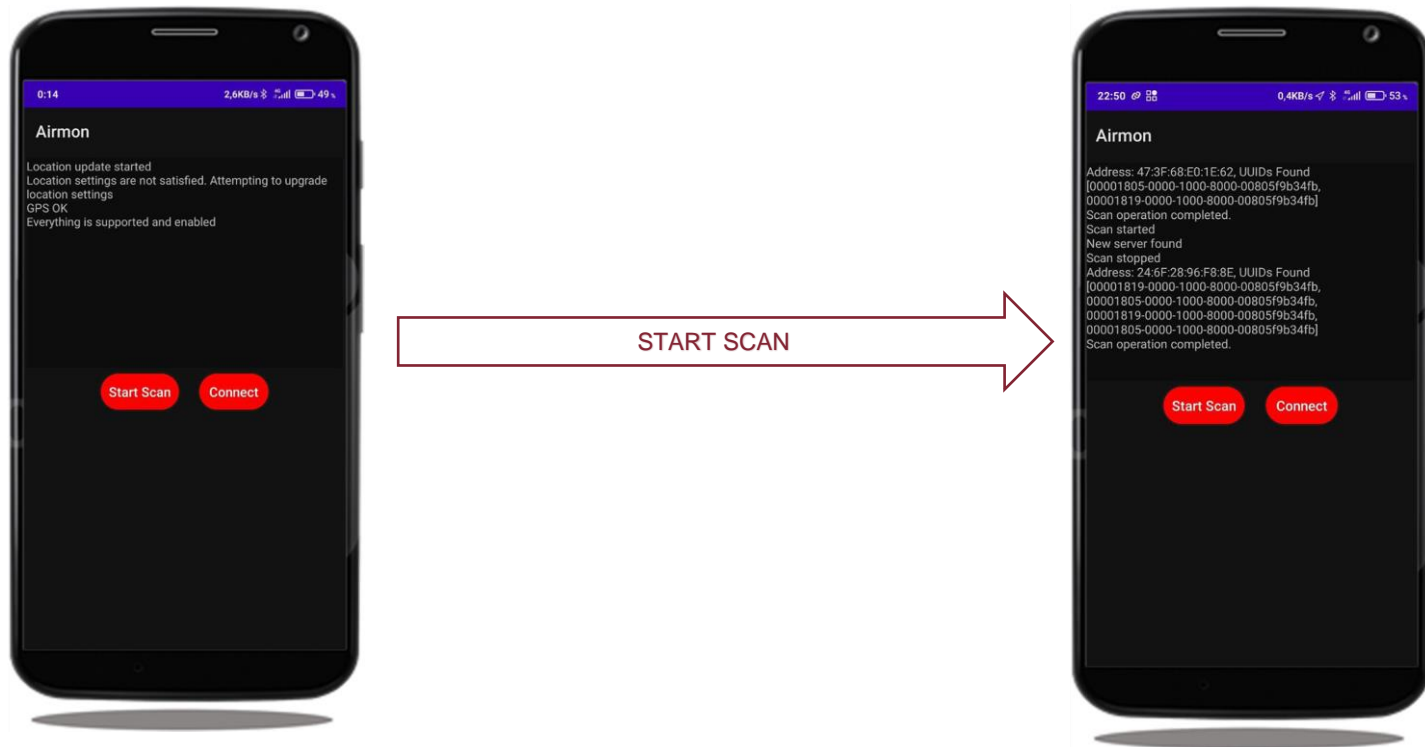
Android Client



- Listeners and Callbacks have been developed to allow the asynchronous exchange of messages;
- Send message;
- Subscribe to the Characteristic of the Service in Server;
- Is not pairing.

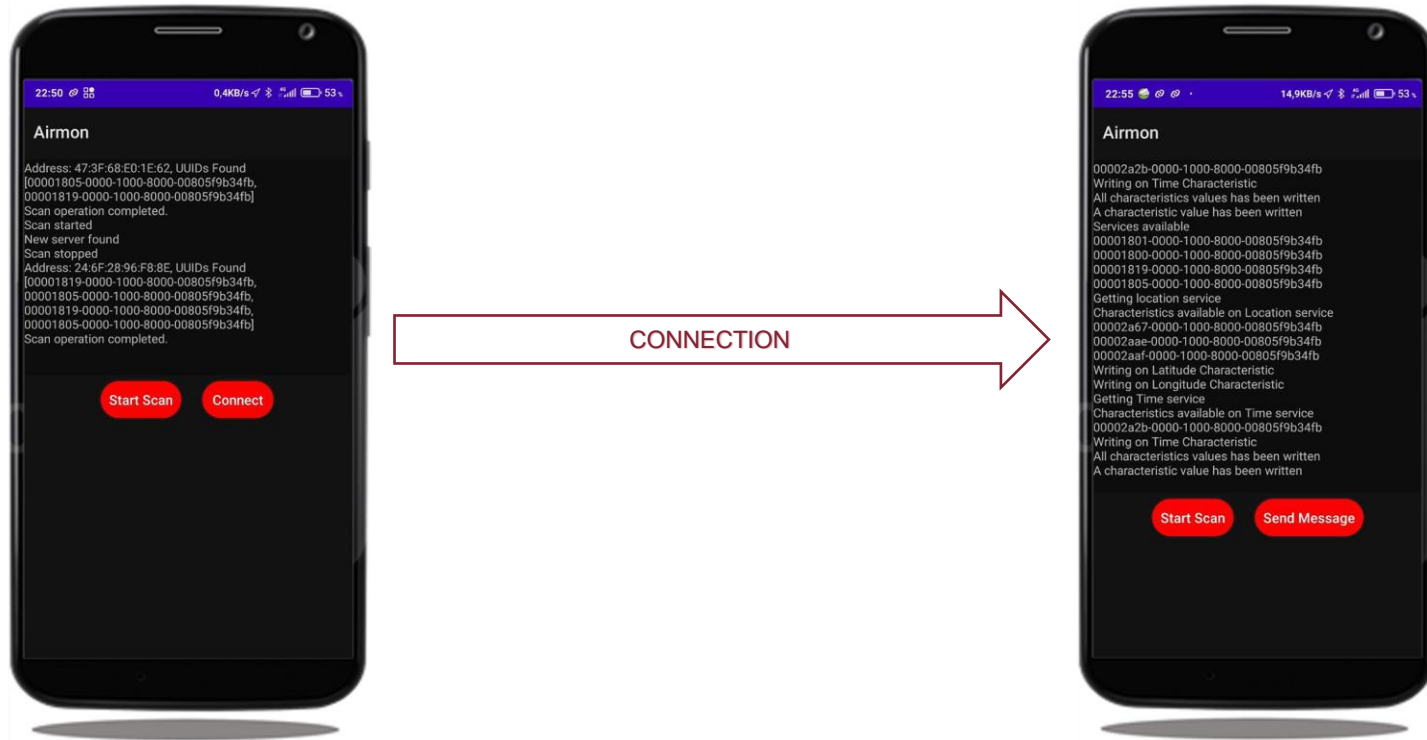
Airmon: Start scan

Start scanning to find server



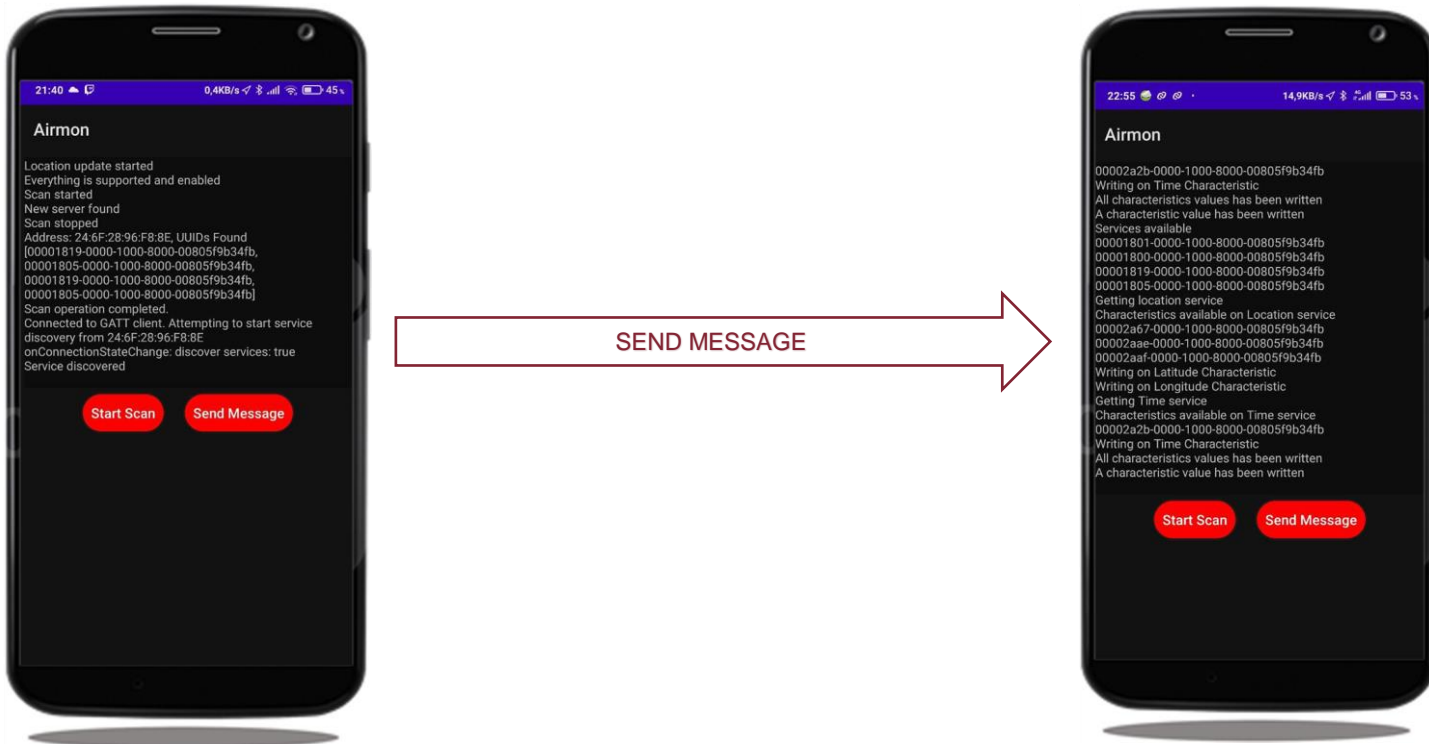
Airmon: Connection

Connection to the server



Airmon: Send message

Write characteristics values



Communication between device and ESP32

What does the development of our project consist of?

Our project provides for an air monitoring that through an air pollution chip takes the data of the latter and through BLE takes the data of the position. Through the intersection of these two data we can verify the quality of the air with the relative position.

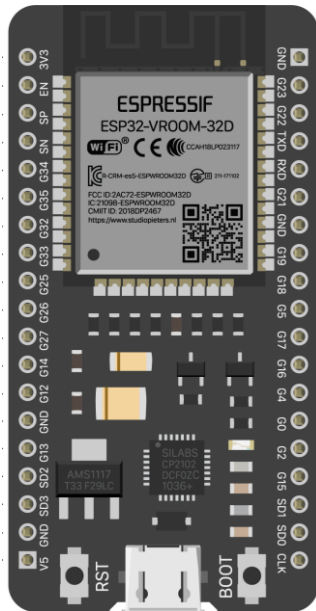
In our project there are 3 operations available through the characteristics:

1. Read
2. Write
3. Notify

Let's see some examples of use cases in the following slides

Communication between device and ESP32

Write values to the ESP and be notified after being subscribed

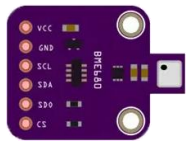


Project architecture overview

The modules and devices

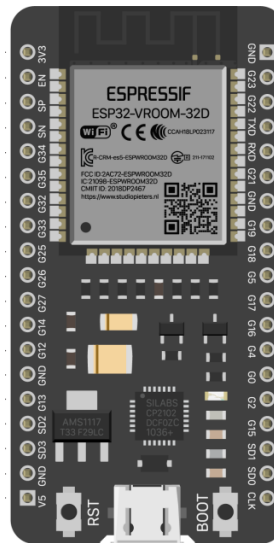


Coordinates + Time + Air Data



BME680

Air pollution data



Coordinates + Time

Warning



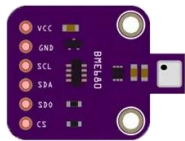
Communication between device and ESP32

Case 1/3: everything good

Data is received from the sensor and the phone, then latitude, longitude, time and air pollution are sent to LoRaWAN

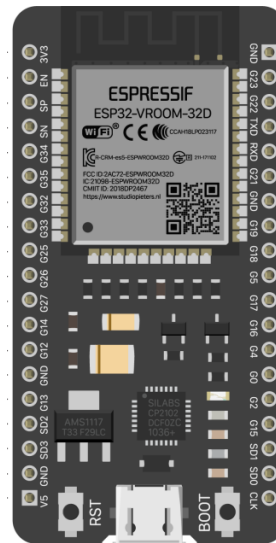


Coordinates + Time + Air Data



BME680

Air pollution data



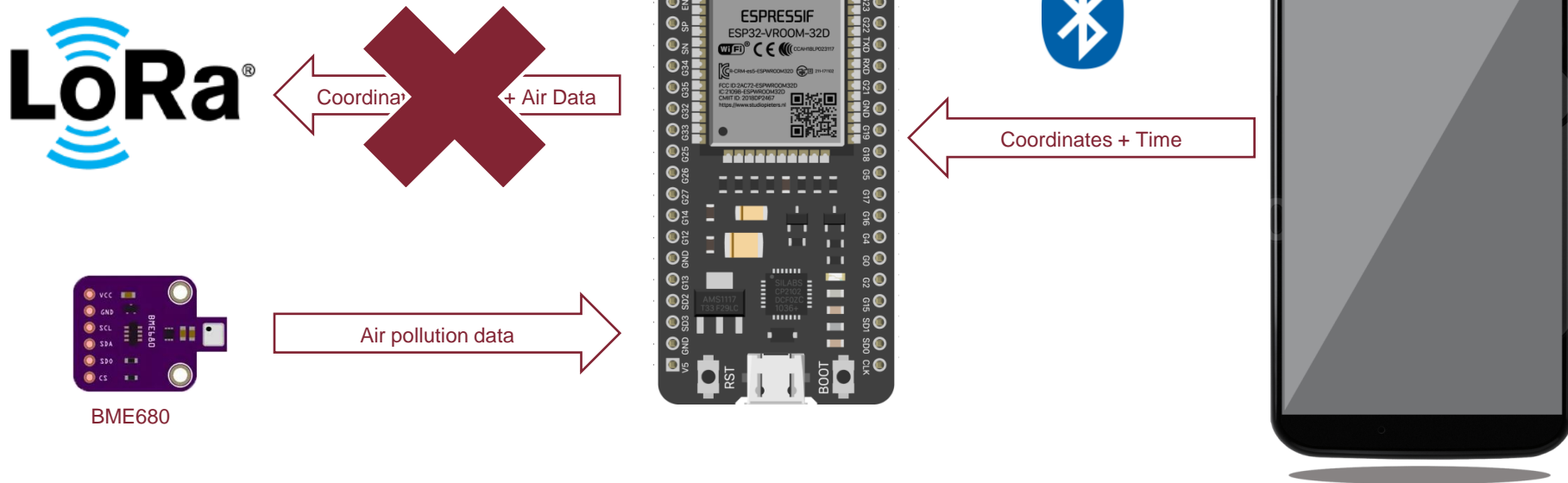
Coordinates + Time



Communication between device and ESP32

Case 2/3: LoRaWAN is not reachable

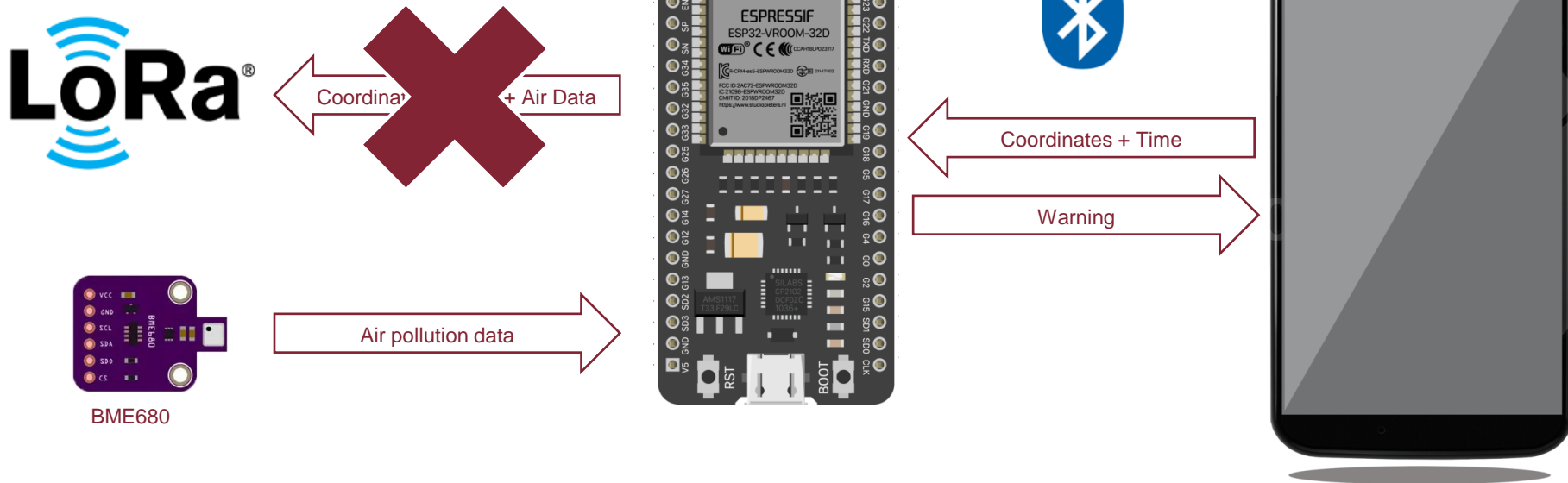
Data is received from the sensor and the phone, but can't send to LoRaWAN then add tuple



Communication between device and ESP32

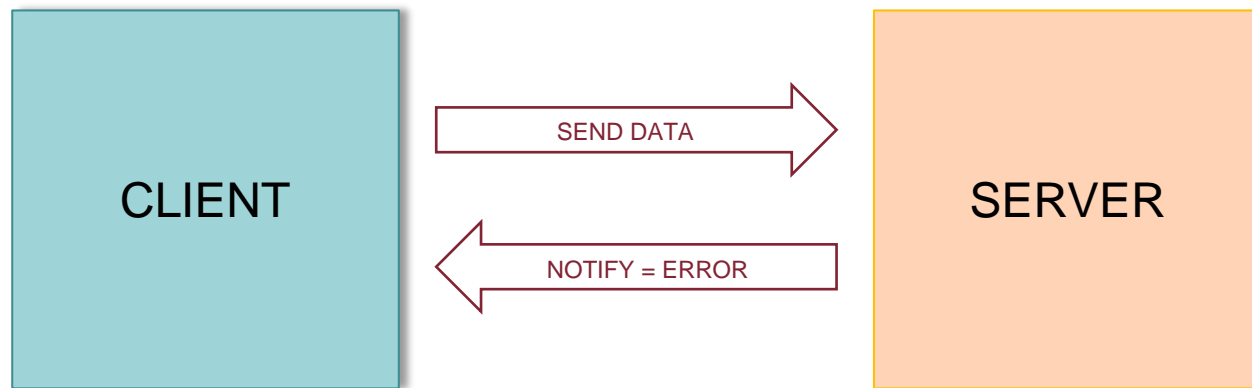
Case 3/3: LoRaWAN is not reachable for a while

Data is received from the sensor and the phone, but can't send to LoRaWAN then add tuple until 256, then send warning



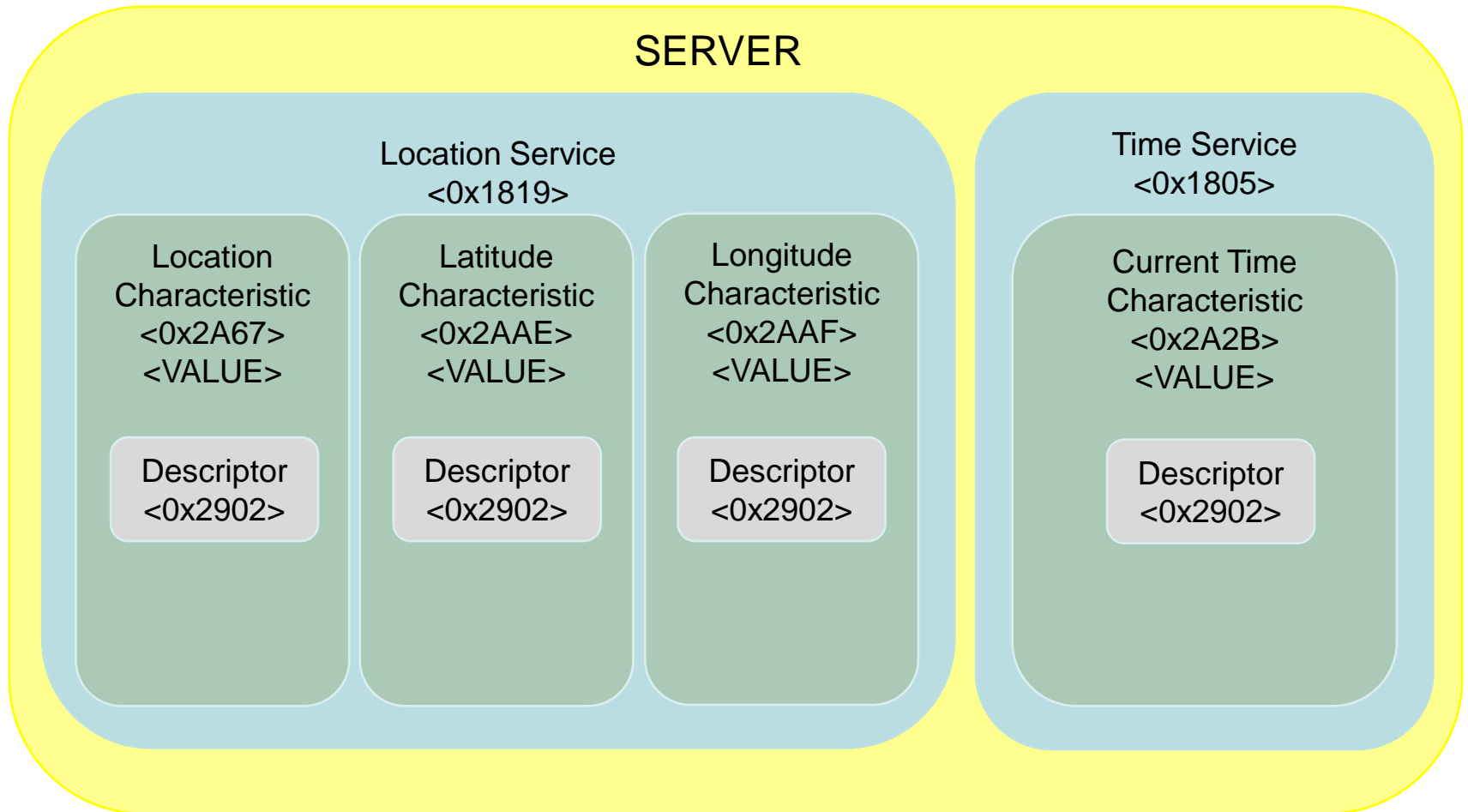
Establish BLE communication

To send and receive data with BLE we had to create a BLE server and establish a bi-directional connection. But how does our BLE server works? How have we implemented this kind of connection?



Connection between layers

BLE Server: How is our BLE server composed?



How we implemented BLE communication

- The server is able to communicate just by writing in one of it's own characteristic
- Why not implement a server on android too?
- Complexity of:
 - increase in system complexity
 - waste of resources
 - waste of energy

Future developments

- Addition of the air pollution module with its implementation
- Addition of the LoRa module to send the data tuples received from the mobile device and from the BME680
- Implementation of the server on the Android side for any additional features

Questions?

Thank you for your attention!