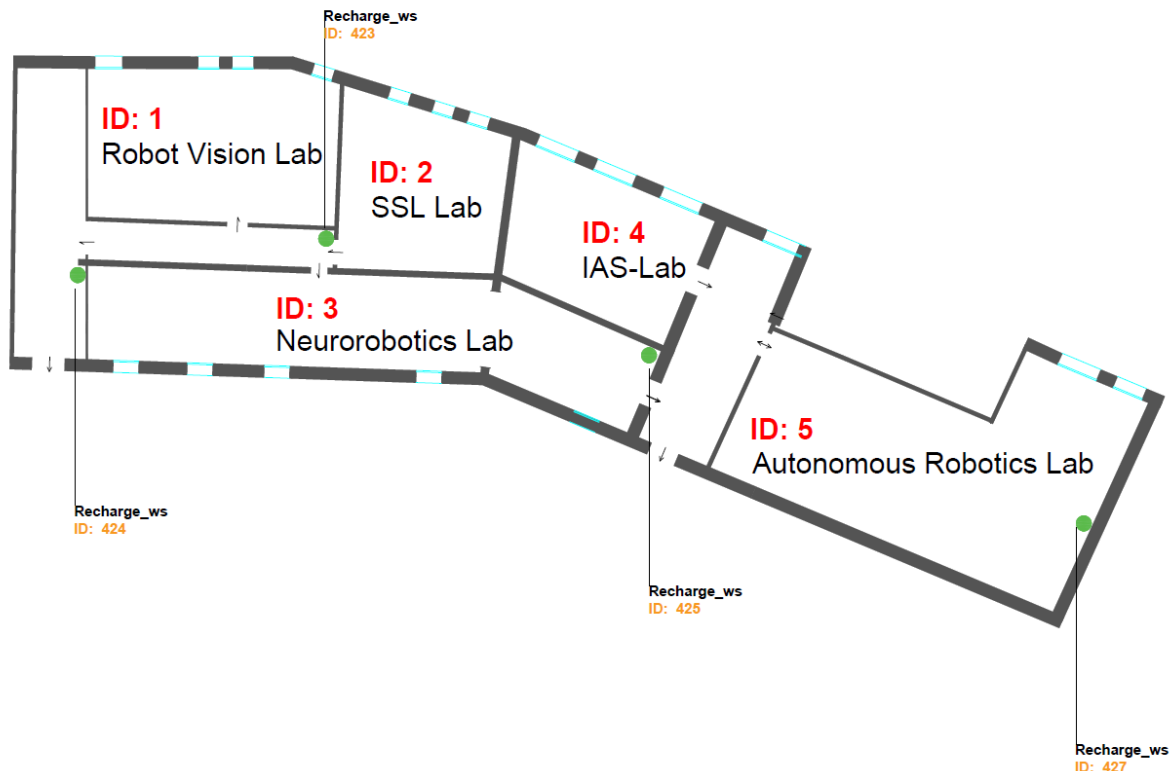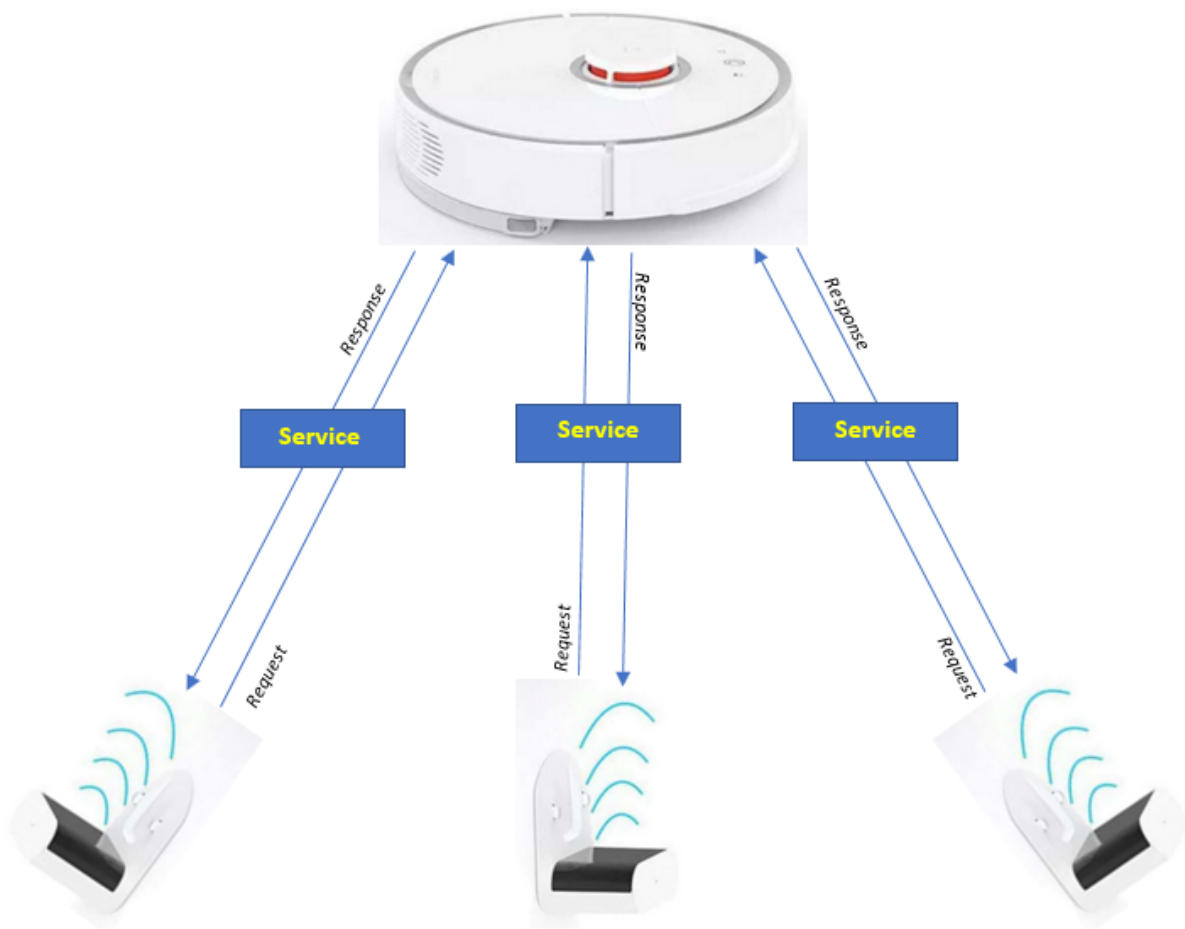# Exercise 2: ROS Services

The department's robotics group bought an autonomous vacuum cleaner to automate the cleaning of its laboratories. The robotics group has 5 rooms and the robot has to vacuum all rooms.

The department also bought n-charging stations and put them in strategic positions in the robotics building.

All the charging stations need to know the position of the robot and the remaining battery level. To do that, all the nodes (i.e., charging stations) send a request to the robot using the ROS Service communication every X seconds, X is a variable that you should set to a different value for each charging station. E.g., Node-1 sends request every 5 seconds, Node-2 every 1 seconds, …, Node-n every m seconds. The robot's node receives the request and sends a response. You have to use the same message developed for Exercise 1 as response (ID and name of the room, level of the battery).

Code you have to implement:

1. Create a custom ROS service that contains the following structure:
    a. **Request** with a std_msgs/Header and the ID of the charging station.
    b. **Response** with a std_msgs/Header and the message that you created in the Exercise 1 (ROS Message).

(N.B. Don't copy and paste lines from the file .msg in your service file. You must include the msg file as showed in the class).

2. Create the ROS node of the robot that handles the service's requests and sends the responses to all charging stations.
3. Create *n* nodes, one for each charging station, that send the requests using the service described above. In addition, the nodes of the charging stations must print the requests and the responses received in a terminal window.

N.B.: Do not create a different file for each charging station. You have to spawn the n charging stations from the same file. (Look out for errors!!! Suggestion to solve the errors: use ROS launch file!)