Implementation of a Wi-Fi Sensor for the GeneroCity App on Android

Bachelor in Applied Computer Science and Artificial Intelligence

Simone Russolillo 1985206

Advisor
Prof. Emanuele Panizzi

Academic Year 2023/24

SAPIENZA
UNIVERSITÀ DI ROMA

Table Of Contents

The Growing Parking Dilemma

- With vehicle ownership on the rise, the demand for parking availability has become a critical issue.
- In fact, it is estimated that around 30% of urban traffic is generated by drivers merely searching for parking spots.

GeneroCity: An App for Smart Parking

- It aims to ease the exchange of parking spots in real time by leveraging the principle of *generosity*.
- Offers users a comprehensive *Parking Activity Log*:
  - Start date and time
  - End date and time
  - Type of parking
  - Street location

How it Works

- To notify others of an available space, users simply press the **LEAVE PLACE** button.

- Meanwhile, those in search of parking can easily tap the **SEARCH PLACE** option to discover openings nearby.

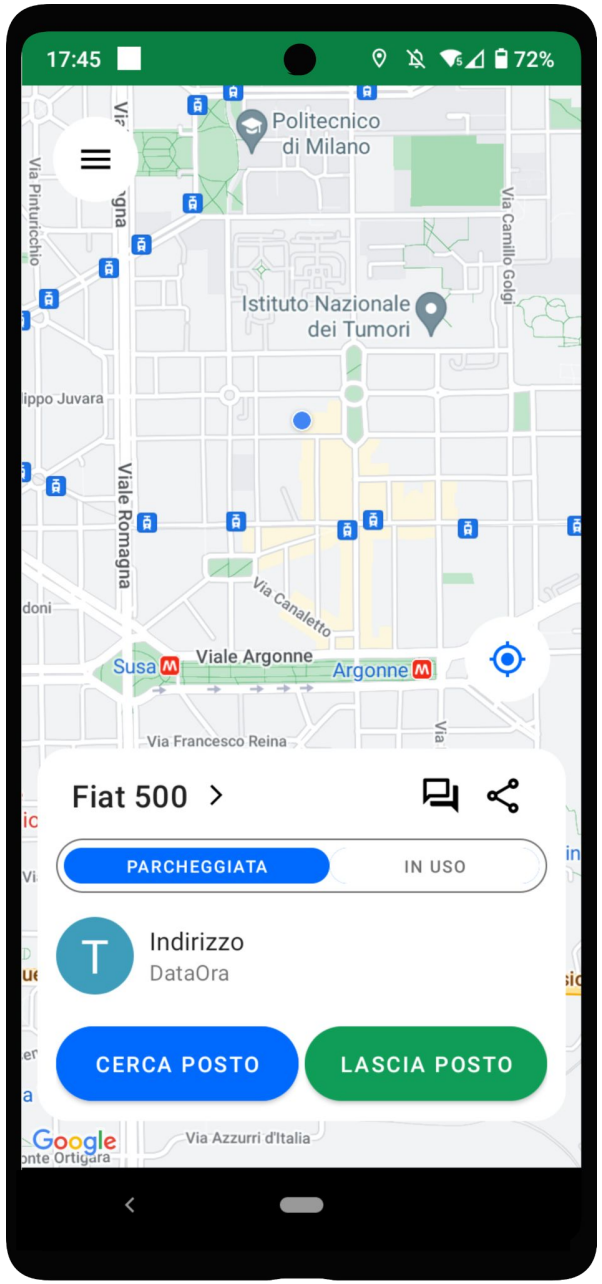
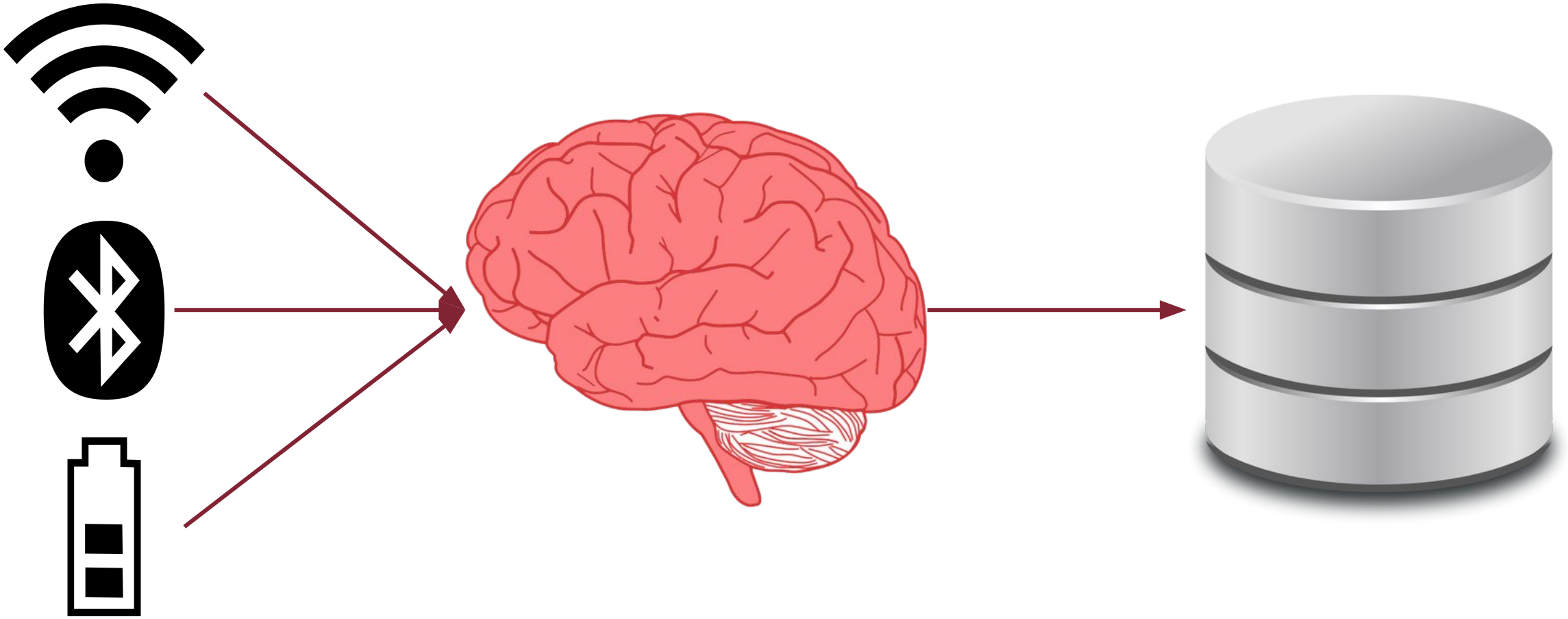


Table Of Contents

Central System and Data Logging



Sensor Interface & Constants

Confidence Values:

between 0 and 1, it reflects the probability that the user is in a certain state.

- [0; 0.5) → Walking
- 0.5 → Uncertain
- (0.5; 1] → Automotive

```
public abstract class GCSensorInterface {

    public abstract double getStatus(Calendar timestamp);
    public void update(Calendar timestamp, SensorData sensorData) {}

}
```

```
public final class GCSensorConstants {

    static void onUpdate(Calendar time) {}
    private static double compute(Calendar time) {}

}
```

Table Of Contents

Setting Up

App Permissions:

- **Android Services**:

allow apps to perform long-running background operations or react to system events without a user interface.

- **Broadcast Receivers**:

listen for specific system broadcasts, such as changes in battery state, network status, or Wi-Fi connection.

Table Of Contents

First Version •

Structure and Data Stream

```java
public class WifiBroadcastReceiver extends BroadcastReceiver {

    public void onReceive(Context context, Intent intent) {}

}
```

```java
public class WifiSensor extends GCSensorInterface {

    private void onData(List<WifiData> listOfWifi) {}

}
```

Wifi List

"Redmi Note 9S"

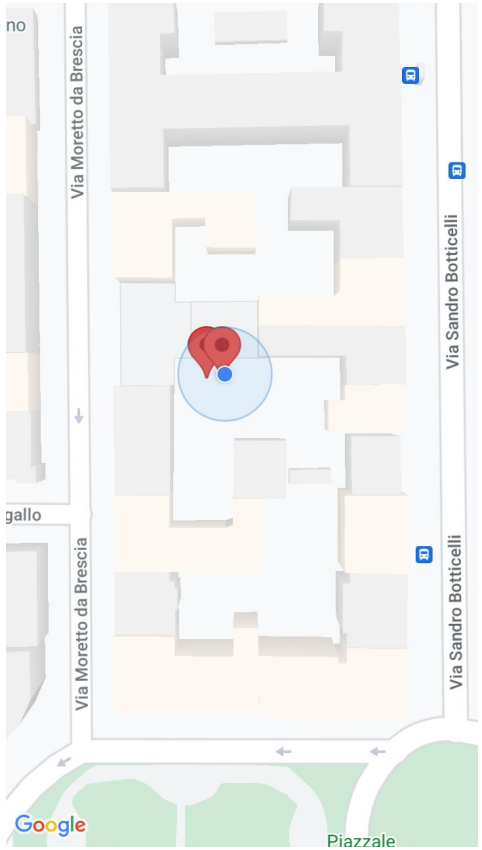
WifiData{
id= ff7d0e27-2f2c-400f-b15f-56cbaa942725
ssid= "Redmi Note 9S"
latitude= 45.4729234
longitude= 9.2286317
isHome= false
intervals=
[From: 22/09/24 19:23:25, To: NOW]
[From: 22/09/24 19:23:16, To: 22/09/24 19:23:19]
[From: 22/09/24 19:23:07, To: 22/09/24 19:23:11]
}

"FurgoneDellaDigos"

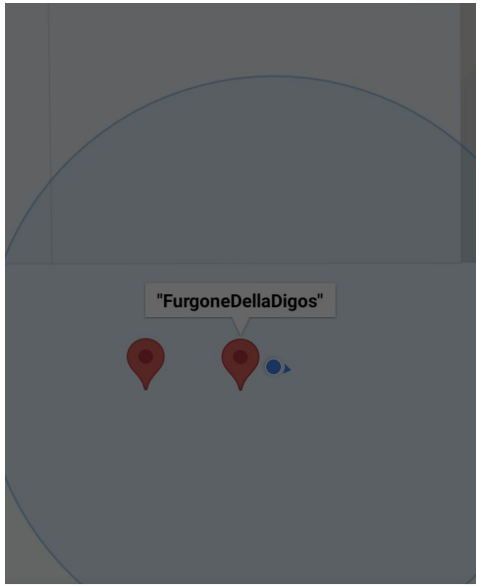
WifiData{
id= 79dac1a2-7522-4621-bb0a-36dc832e332a
ssid= "FurgoneDellaDigos"
latitude= 45.4729575
longitude= 9.2286318
isHome= false
intervals=
[From: 22/09/24 19:22:50, To: 22/09/24 19:23:07]
[From: 22/09/24 19:22:08, To: 22/09/24 19:22:08]
[From: 22/09/24 19:22:03, To: 22/09/24 19:22:03]
[From: 22/09/24 17:44:59, To: 22/09/24 19:21:58]
}

Table Of Contents

Map Interface and Synchronization



Map Markers



"FurgoneDellaDigos"

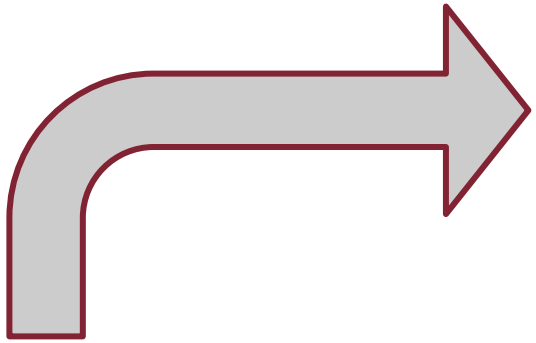
Connection Intervals:

From: 22/09/24 19:24:02 To: NOW
From: 22/09/24 19:22:50 To: 22/09/24 19:23:07
From: 22/09/24 19:22:08 To: 22/09/24 19:22:08
From: 22/09/24 19:22:03 To: 22/09/24 19:22:03
From: 22/09/24 17:44:59 To: 22/09/24 19:21:58

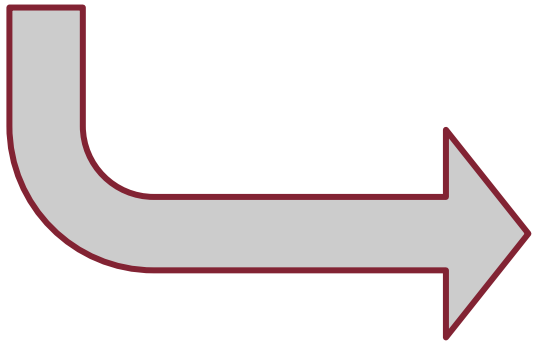
Bottom Sheets

By adding more activities that required the same data, I encountered the issue of **duplicate signals** caused by multiple instantiations of ‘WifiManager‘ and ‘WifiBroadcastReceiver‘ for the same service.

Breakpoint



Two essential features are left:



- **User State Calculation**:

this feature would calculate the user's status, providing a value between 0 and 1 to determine the likelihood of the user being in motion or stationary.

- **Fixed Vs. Mobile Access Point Detection**:

this feature would distinguish between fixed networks (e.g., home or oce) and mobile networks (e.g., temporary hotspots).

Table Of Contents

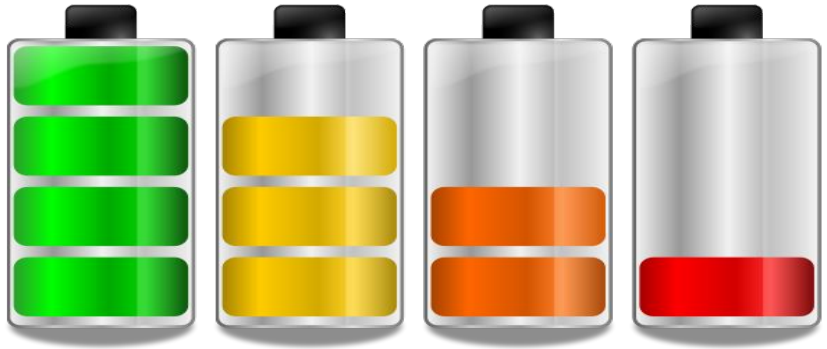
Definition and Efficiency

Attributes:

1. id
2. ssid
3. Latitude & Longitude
4. isHome (False)
5. Connection Intervals

The Battery Issue:

```
LocationRequest locationRequest = LocationRequest.create()
        .setInterval(10000)  // 10 seconds interval
        .setSmallestDisplacement(25);  // 25 meters
```



Fixed Vs. Mobile

Check If Home:

```
public void checkIfHome() {
    this.isHome = this.connectionIntervals.size() > 10;
}
```

Reset Wifi Data:

```
if (displacement > 100) {
    wifiToBeUpdated.resetWifiData(latitude, longitude);
}
```

Table Of Contents

Confidence

Initial Decisions: **range adjustment**

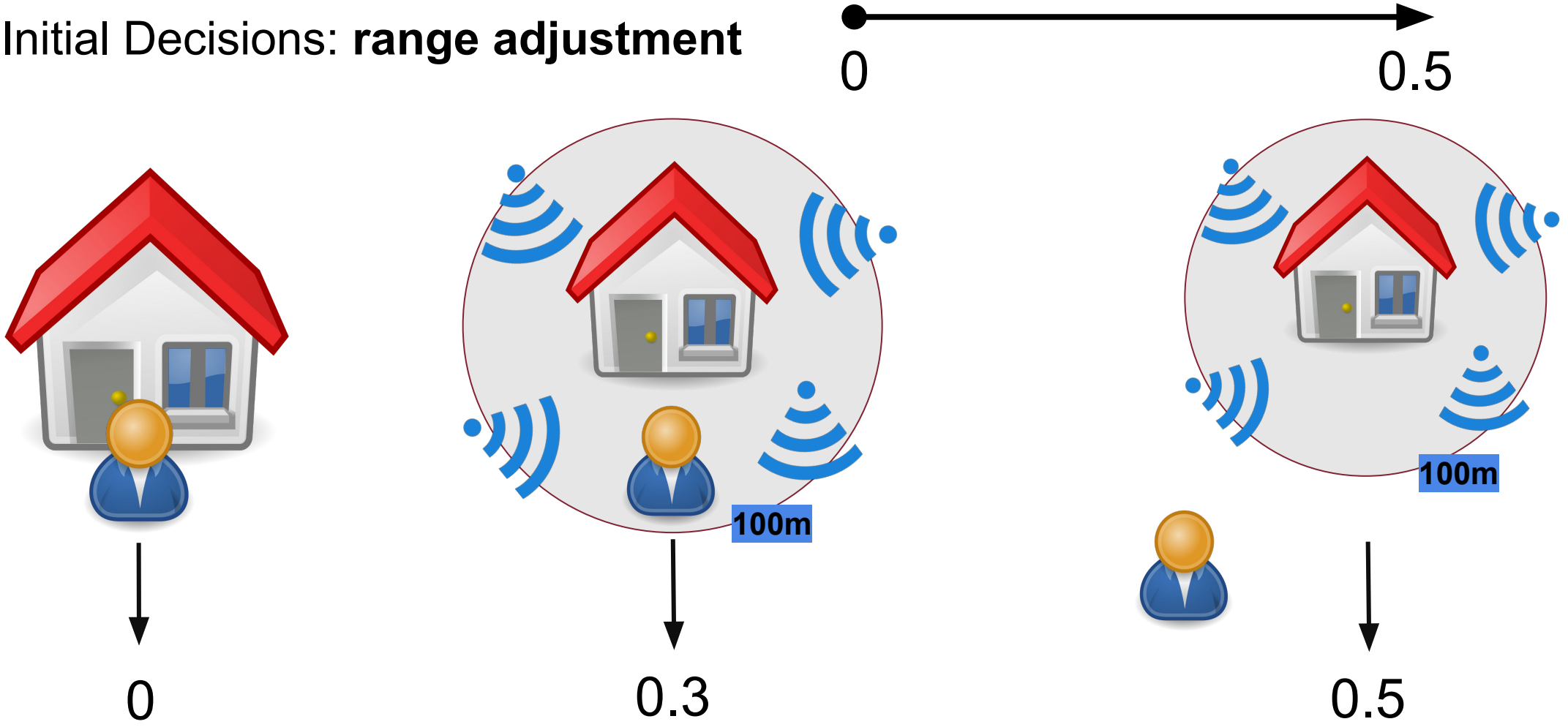


Table Of Contents

Conclusions

- In conclusion, the GeneroCity project presents a **promising solution** to the persistent problem of trac congestion, particularly in urban areas where the daily search for parking can be a frustrating experience for many.

- In future developments GeneroCity may evolve into a more sophisticated tool through the implementation of a **machine learning model**.

SAPIENZA
UNIVERSITÀ DI ROMA

Implementation of a Wi-Fi Sensor for the GeneroCity App on Android

Thank you for listening!

Simone Russolillo
Bachelor in Applied Computer Science and Artificial Intelligence