



Software Engineering, Department of Computer Engineering
Boğaziçi University
November 2018
By Taner Eşme

PROGRESS REPORT OF THE PROJECT “BLUETOOTH INDOOR POSITIONING USING RASPBERRY PI”

1. Introduction

This report gives you information about the progress of my project for SWE599. As you might remember, my project was an implementation for “Bluetooth indoor positioning using Raspberry Pi”. This report will first present preliminary findings that I have completed so far about my project. The report will consist of three section. You will read the project description provided by the lecturer and the other sections will be the project’s requirements and design.

2. Project Description

Estimating the location of a Bluetooth capable device in an indoor environment has many practical purposes. Either for a smart warehouse with autonomous robots carrying the goods around, or for a classic multi-storey office with employees moving from one place to another, or for a vacation compound where crowded groups of tourists can go to swimming pools, restaurants, and other activity locations with a wearable computer (e.g. a smart bracelet, or a necklace), instantaneous information about where these people are at a certain time period, and where they are moving to, will definitely help organizational planning. Tracking employees on how much time they spend outside their workspace would be an important performance metric especially in production lines. In the case of tracking tourists, the hotels can adjust their security, and other services according to the movement of their guests. For the smart warehouse, location estimation has to be so precise that a robot should never place things in wrong shelves, nor it should look for items that do not exist at the current location of the robot.

Indoor positioning can be performed using a set of stationary Bluetooth access points.

Location estimation inside a building is harder than doing it with GPS. One of the major approaches in indoor positioning is to use a fingerprinting database, trained with multiple signal level measurements at specific locations, covering the entire area, with respect to each available access point. The other major approach is to use triangulation, which is based on coordinated location estimation using at least three access points, and signal arrival times. The arrival time can be calculated at the stationary access points, or the mobile device itself. This method can be roughly considered as an indoor alternative to the GPS solution.

In this project, we are going to implement a hybrid solution of fingerprinting and triangulation.

3. Requirements of the Project

I am going to use .NET technologies in the project to implement needed software that will run on Raspberry Pi. Using .NET technologies will be providing us with easiness in some of the topics and also bringing a little burden of dependencies to the Microsoft technologies. To eliminate some of those dependencies, I will leverage the power of .NET Core, which is an open source framework promoted by Microsoft to support all other platforms such as Linus and Mac OS.

After a little preliminary research, I discovered that we could not run .NET Core on the first generation Raspberry Pis because of the unsupported CPU architecture (ARM v6). I also found something important related to .NET Core, we cannot do any implementations related to Bluetooth without using Windows UWP (Universal Windows Platform) because .NET Core, unfortunately, does not have a cross-platform implementation to communicate the peripheral devices with the capability of Bluetooth.

According to the descriptions above, I can make a list of the requirements like below:

- **Raspberry Pi 3 Model B+:** This is a third generation Raspberry Pi which is able to run .NET Core Runtime and Windows 10 IoT Core. It also has an onboard Bluetooth module to collect signals radiated from agents and the Wi-Fi module to transmit the collected data to the backend server(s).
- **A cheap version of beacon or a smartphone:** As an agent unit. It will be used when constituting the fingerprinting database and during tests.
- **.NET Core 2.1:** As a development framework.
- **Windows 10 IoT Core:** As an operating system to run on Raspberry Pi.
- **A server or a commodity computer:** To transmit the collected data, store fingerprinting database and execute the localization algorithm.
- **Visual Studio 2017:** As a development environment. There are a few other development tools for .NET Core, but when it comes to developing the applications associated with Bluetooth, we will have to use VS2017 because of the compatibility issues.
- **A PC installed Windows 10:** It is a must to have a PC that Windows 10 was installed on to develop applications that can run on the devices such as Raspberry Pi with Windows 10 IoT Core.
- **ML.NET:** A machine learning algorithms library promoted by Microsoft.

4. Design of the Project

In this project, we are going to make use of fingerprinting and triangulation to localize the agents. That's why we will need to have at least three access points as Raspberry Pi implementation deployed around the field. After deployment, it simply will be looking like below.

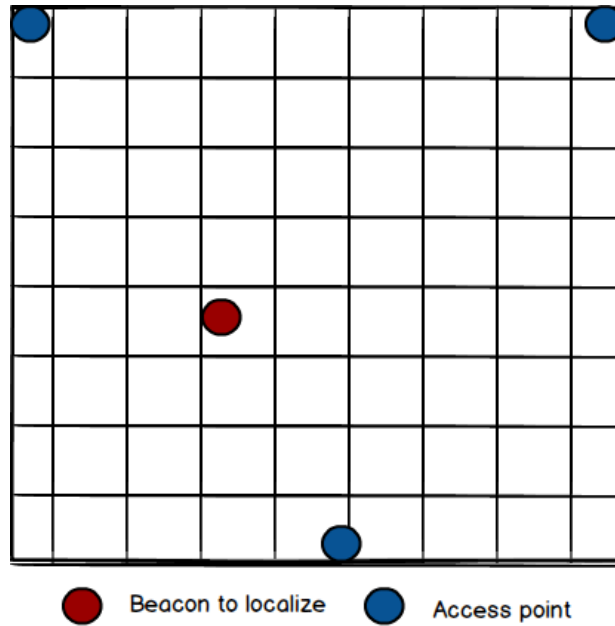


Figure 1 Organization of sensors in the field

The organization of the access points can be changing according to the limitation of the field.

a. Logical Design

The access points receive signals radiated by the agent through Bluetooth protocol and send them to the consumer server(s) that will store and analyze the data. Physical design for the project will look like below:

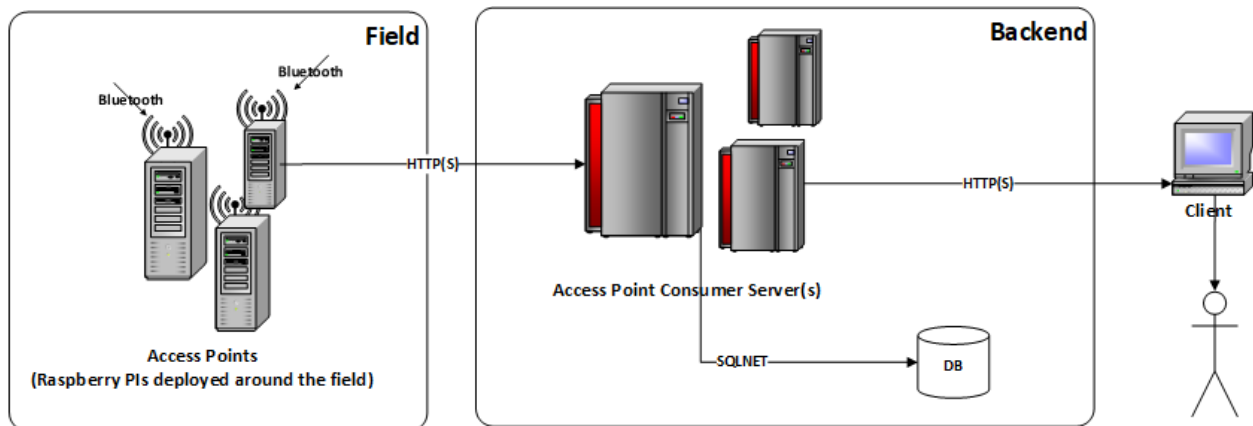


Figure 2 Logical Design

- **Access Points:** The modules receiving the signals through Bluetooth which are Raspberry Pis.
- **Access Point Consumer Server(s):** The servers that access points will be sending the data they collected via HTTP or HTTPS. There will be API for access points to connect. These servers also will have a UI application for users to view the locations of the agents.
- **DB:** The database will store the signal levels and fingerprinting data.

b. Software Modules

The software has two essential parts, first is the application running on Raspberry Pis and the second one to run at the backend.

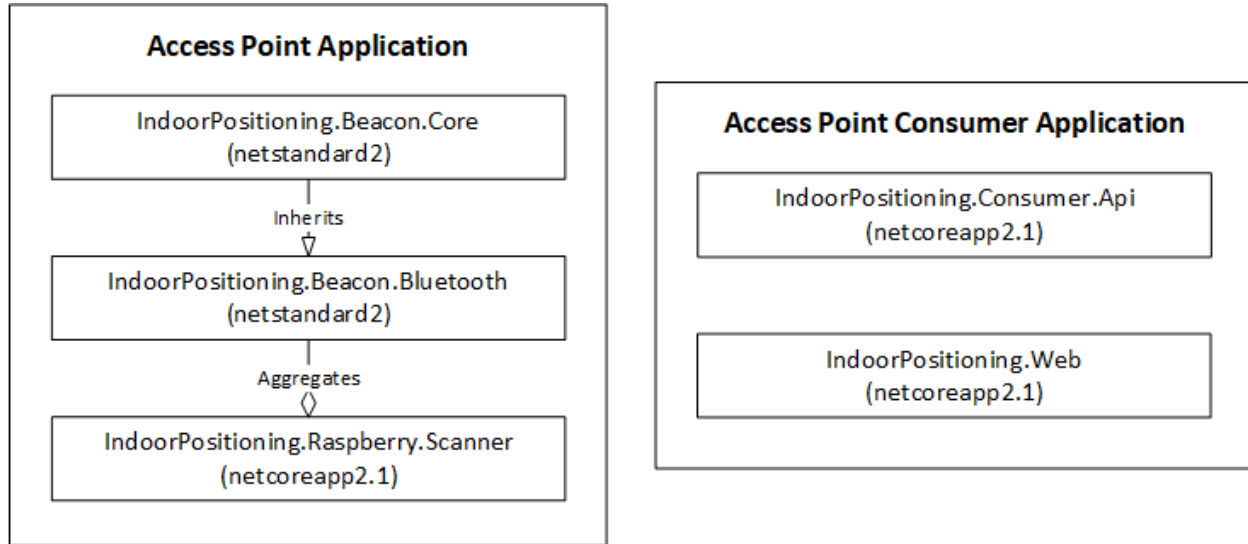


Figure 3 Software Modules

- **IndoorPositioning.Beacon.Core:** It is a .NET Standard 2 library project to constitute a standard for the devices that can be used by agents.
- **IndoorPositioning.Beacon.Bluetooth:** It is a .NET Standard 2 library project to implement the standard that is constituted by **IndoorPositioning.Beacon.Core** for Bluetooth.
- **IndoorPositioning.Raspberry.Scanner:** It is a .NET Core 2.1 console application project using the **IndoorPositioning.Beacon.Bluetooth** library to scan the signals of the Bluetooth capable devices.
- **IndoorPositioning.Consumer.Api:** It is a .NET Core 2.1 WebAPI project for the access points (Raspberry Pis) to send their data collected. This application also is where the localization algorithms are implemented.
- **IndoorPositioning.Web:** It is a .NET Core 2.1 MVC project for users to see the positions of the agents and allow them to view the reports related to the system.

Conclusion

This progress report created to update you on the status of my project on “Bluetooth indoor positioning using Raspberry Pi”. I am on schedule so far and complete the project by the deadline, Dec 24, 2018.