Tung Trinh

Positioning and Routing with Bluetooth Low Energy

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

24 February 2016

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Tung Trinh Positioning and routing with Bluetooth low energy 36 pages 24 February 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Development
Instructor(s)	Petri Vesikivi, Principal Lecturer

This thesis report is based on the investigation and application of Bluetooth technology, in specific Bluetooth Low Energy. Bluetooth Low Energy (BLE) refers to Bluetooth 4.0 or above Core Specification, which emphasizes low power consumption, low cost, low bandwidth, and simplified communication protocol compared to classic Bluetooth. It uses proximity, a relative distance between objects of interest as the main measurement. This is the main difference to the majority that use location, an absolute position of one object upon a coordinate.

The main references for the topic came from a commercial project done by the author with Luxus Oy for a shopping mall. The project made use of iBeacon which is a Bluetooth Low Energy protocol, in addition to a web service and beacon recognition framework provided by a third party. The goal of this study was to create an ecosystem that monitors and records the user proximity of the application with BLE devices. These devices were installed at shop entrances and they handed out pre-determined rewards.

Although the project was successful to a point, it was not able to take full benefit from advantages of Bluetooth technology. The problems acknowledged during the project development are discussed in the study. One challenge was caused by BLE and its inconsistency in interactivity. No feasible solution was found to this problem partly because of time limits. The development team had to conclude that the problem lies in the premature nature of BLE. Another problem was the vertical sensitivity in proximity when reading the beacon. This caused false positive results and discouraging customers to visit each shop for rewards. This was solved by reducing the required proximity level of the mobile application and the BLE device.

Finally, this study discusses how iBeacons system to indoor routing or navigation could be further developed.

Keywords	Bluetooth Low Energy, BLE, iBeacon, routing, application

Contents

Abbreviations and Terms

BLE	Bluetooth Low Energy
RSSI	Received Signal Strength Indication
IoT	Internet of Things
MIT	Massachusetts Institute of Technology
Ltd.	Limited Company
LOS	Line of Sight
FOV	Field of View
SIG	Bluetooth Special Interest Group
RF	Radio Frequency
API	Application Programming Interface
SDK	Software Development Kit

1 INTRODUCTION

Communication is a basic human need. Modern technology tries to facilitate not only the communication between human beings but also between different devices. Bluetooth technology aims at creating connections between two physical entities to share data. Although Bluetooth is the most viable option for basic interchange between two devices, it is somewhat unreliability in connection and suffers from high battery consumption. The market desires a faster, less complicated bridge between parties for small size data and specialized services' data. Bluetooth Low Energy (BLE) was introduced as a corporative work between Nokia and several wireless technology owners [1, p. 75]. It aimed at providing fast, low energy transfer among a diversified of available mobile devices, or even any Bluetooth capable breadboard. BLE introduced a refined means of the popular communication protocol.

One of the most significant improvements of BLE is the ability to limit the size and complexity of data. Thus, BLE's most popular applications are positioning-based ones. In this type of application, low energy proximity called RSSI (Received Signal Strength Indication) is exchanged between participants. The developer can implement desired interaction between participants by configured RSSI measurements. The thesis targeted the same area, making use of the interaction between BLE devices and mobile application to produce a minimal service. The author of this study used one of the projects created while working for Luxus Oy as part of the contract with one of the largest shopping malls in Helsinki Metropolitan region.

2 WIRELESS COMMUNICATION TECHNOLOGIES

2.1 Radio technology

Radio technology is the oldest and most popular wireless communication method. The phenomenon known as radio waves was first proved by Heinrich Hertz, inspired by Maxwell's extensive theory on electromagnetic of light in 1888, when he introduced his first recorded results of the experiment. End-loaded dipole antenna was used to make up the transmitter part, with a spark gap. It was then charged and discharged through the gap, creating the oscillating current. The receiver part was constructed from an antenna and spark gap. At the frequency of 50 MHz, radio waves started to appear. Then

Hertz proceeded to reveal the attributes of those waves, from reflection, diffraction, polarization and the wavelength. [2, p. 7]

In 1895, an Italian called Guglielmo Marconi introduced radio waves. He registered for a patent of ‘‘Improvements in transmitting electrical impulses and signals and in apparatus therefor’’ in England in 1896. In 1901, he presented the first time in public his pioneer in radio communication, sending a letter S in Morse from Cornwall to Saint Johns in Newfoundland through his device, with the Atlantic in-between. The significance of this experiment lay in the distance that the waves had travelled, over 3000 km, shattering all expectations at the time. On the other part of the world the Russian scientist named Alexander Popov also carried out the same experiment. Both Popov and Marconi used a better type of detector instead of antennas and spark gaps system as Hertz. This coherer inherited the great light from vacuum tube invention, relying on metal particles reaction of electrode pairs inside glass vacuum tube. [2, p. 7]

In 1904 John Ambrose Fleming invented diode from rectifying vacuum tube; two years later Lee De Forest from USA add another electrode and create the triode. This invention managed the apparatus current and enable amplification. Then in 1912 the American inventor Edwin Armstrong discovered regenerative feedback, which emit the uninterrupted carrier wave and controlled by voice signal. He also created a superheterodyne receiver, making radio broadcasting a reality. AM stations were the first to be on air, starting from 1920, and FM after World War II. The invasion of radio waves didn’t stop there. In 1932, Karl Jansky opened the door to radio astronomy with his detection of oscillation from the Milky Way. In 1940, the birth of microwave tubes, klystron and magnetron created the basic for microwave radar, which has been the abandoned idea from early 1903 of a German engineer Ismeyer until Marconi brought up again after his thorough understanding of radio waves. After that radio technology has developed continuously with the biggest contribution from MIT’s Radiation Laboratory leading scientists, physicists, resulting in more than 27 books relating to the researches. Around the same time, the integration of semiconductor devices into industry practices bloomed, leading by the creation of the first point-contact transistor, opening the new age for electronics. They then evolved and became present time radio technology, with smaller size and more complicated modules, making mobile phone, satellite and GPS possible. [2, pp. 7-8]

2.2 Wi-Fi and Zigbee

2.2.1 Wi-Fi

Wireless Fidelity (Wi-Fi) is one of the most, if not the most popular wireless communication method one can encounter anywhere. Wi-Fi connects mobile and many other devices from office, households, outdoor areas. It was developed based on IEEE 802.11 a/b/g standards in wireless local area networks (WLAN), using 2 main frequency bands of 2.4 and 5 GHz. Its main goal is to provide users with the ability to browse and engage in internet activities, maintaining the speed of broadband connection. Mobile or other devices can connect to Wi-Fi through access point or in ad-hoc mode. The standard registered as IEEE 802.11 covers medium access control (MAC) and physical (PHY) layers. It applies a set up at architectural level with different components fluently communicating with each other to ensure the data mobility till upper layers. [3, pp. 2, 20]

2.2.2 Zigbee

In contrast of Wi-Fi characteristics, Zigbee is a communication method operating in a lower end of the spectrum, results in low-cost, low power wireless networks that emphasize flexibility. It provides much simpler protocols to connected applications than Wi-Fi. Zigbee also supports mesh topologies, meaning messages are relayed between devices through multiple wireless hops. The technology is published and maintained by Zigbee Alliance, who also provide the certified evaluation for it. The name is inspired from the natural phenomenon where bees communicate through dancing around each other, in a multi-hop fashion.

Zigbee is specified through IEEE 802.15.4 standards, consists of MAC and PHY layers creating personal area networks (PAN) in 900 MHz and 2.4 GHz frequency reading. Furthermore, Zigbee specification is the definition for other IEEE 802.15.4 based system in term of network, security and application framework. [3, pp. 2-3]

Zigbee network usually requires low data bandwidth applications with extensive battery life. This results in Zigbee's signal demanding 10 to 100 times lower power consumption than Wi-Fi's. This fact also brings a disadvantage for Zigbee being interfered by Wi-Fi signal and often time overwritten. [3, pp. 9-10]

2.3 Bluetooth

2.3.1 Introduction

The success and immense effort pouring in classic radio communication bore the origin of Bluetooth technology. The concept of Bluetooth was planned to accompany wireless short to moderate range communications between two or more ends. The life of Bluetooth has seen through its days at Ericsson, to later developed by Special Lobby SIG and then employed with IEEE standard referencing to IEEE 802.15.1. Bluetooth production aim specifically at large scale integration, in form of installed circuit or module of minimal energy consumption and competitive at low price brackets. [1, p. 75]

2.3.2 History

Ericsson Mobile Communications kicked off a study focusing on inexpensive (monetary and energy-wised) interface for handhelds devices and their accessories in 1994. Approximately 4 years later the agreement between IBM, Intel, Nokia and Toshiba with Ericsson marked the birth of the SIG. This expanded by the merging with of 3Com, Agere, Microsoft, Motorola during the year 2000 and continuously with several other players from different industrial fields such as cellular, portable computing, automation and digital processing. All members of Bluetooth SIG can use freely their products and services. The Group has gathered more than 2500 names till this day and become the powerful motor leading evolution of Bluetooth competing against other wireless technologies, especially Wi-Fi. [1, pp. 75-76]

The name originated from a Viking king: Harald Blåtand, or Harald blue tooth, who reigned Denmark and Norway while territorialism and religions teared apart Europe. The origin came in line with the formation of SIG, representing manufacturers coming together under one “flag” and pushing boundaries of development by one standard. Bluetooth’s icon also dictates characters H and B, abbreviation for “Harald Blåtand”. [1, p. 76]

2.3.3 Specification

Bluetooth connection requires a network of devices in close proximity working in a master-slave communication priority. This scheme is called piconet as demonstrated in

Figure 2, consists of one master and at most seven slaves' equipment. Several piconets can overlap to create scatter net (Figure 3).

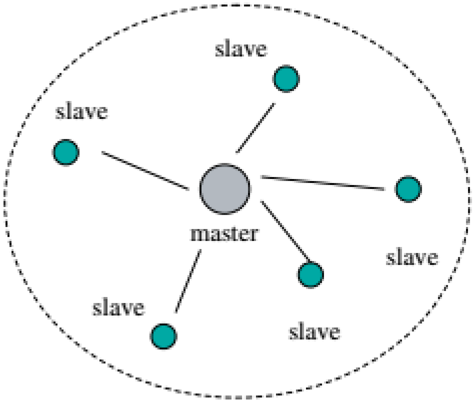

Figure 2: Master/slave configuration [1, p. 77]

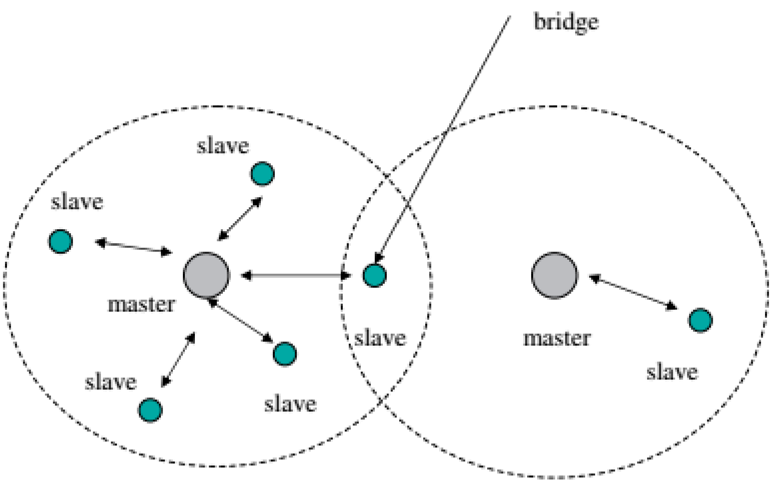

Figure 3: Scatter net of two piconets [1, p. 78]

Conversation within a piconet is managed by master node, controlling frequencies and channels. The discovery phase will detect devices in the neighbourhood while scatter net demands routing between masters and relayed nodes. This extended procedure results in Bluetooth scatter nets configuration to be underdeveloped, though Zigbee standard has contributed in putting up a better-quality channel. [1, p. 77]

Slave devices cannot communicate with each other unless during discovery; all establishments are handled by the master node. There used to be certain limitation on number of simultaneous channels in a piconet; however, newest version of Bluetooth has removed this disadvantage. Additionally, broadcasting is now supported, with the master polling nodes, trafficking connection bandwidths and responsible for synchronizing the piconet's clock for the frequency hopping sequence. [1, p. 77]

The most important advantage of Bluetooth is independency of IP setup. This design emphasizes deployment ease without complicated configuration on upper layers such as address allocation, default router, netmask etc. Bluetooth works around these protocols of:

- Lower layer protocols: Baseband, LMP, L2CAP, service discovery protocol (SDP)
- Interfacing protocols: RFCOMM
- Applicative control specifications: TCS Binary, AT Commands
- Applicative protocols: PPP, TCP/IP, OBEX, WAP, vCard, VCal, WAE

[1, p. 78]

2.4 Infrared

2.4.1 Comparison to radio networks

Infrared while occupies in the shorter communication range possessed multiple advantages over radio. First of all, its bandwidth is not in administered range, in particular it is 200 THz in 700 – 1500 nm range. Infrared radiation additionally inherits similar properties of visible light: impassable through walls or other barriers. This make infrared communication more secured and independent from undesirable sources, significantly advanced in a multiple rooms and levels infrastructures. Finally, infrared is not affected by multipath fading by making use of intensity modulation and direct detection. [4, p. 2]

However, there are certain disadvantages to the method. Infrared, though does not require strict end-to-end visibility, still inoperable under circumstances of concealing such as being inside shirt pocket etc. Another susceptible side of infrared is its limited range of communication due to high amount of noise being ambient light itself, which might cause significant effective path loss. With all the strengths and weaknesses observed, infrared will not overshadow the importance of radio networks but rather complimenting each other's; infrared shines the short distance and high-speed needs, while radio has the omnipresence with lower data rates. The difference between infrared and radio communication can be summarized through table 1. [4, p. 2]

Property	Infrared	Radio
Multipath Fading	No	Yes
Multipath Dispersion	Yes	Yes
Source of Bandwidth Limitation	High photodiode capacitance, Multipath dispersion	Regulatory
Source of Dominant Noise	Ambient background light	Interference from other users
Security	High	Low
Range	Low	High
Pocket Receiver	No	Yes

Table 1: Comparison between Infrared and Radio properties [4, p. 3]

2.4.2 Popular forms

There are numerous forms that one optical link can be arranged, specifically 6 of them as demonstrated in Figure 1.

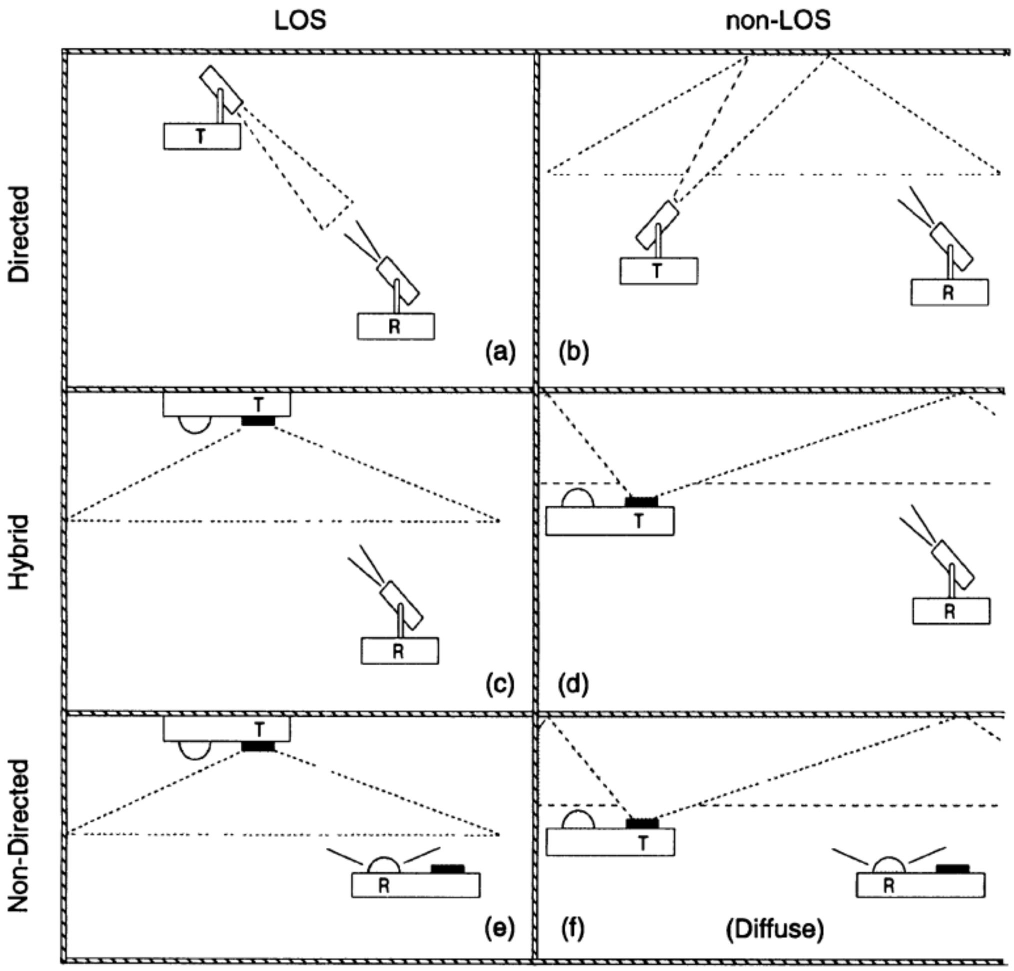

Figure 1: Different configurations of infrared links [4, p. 7]

The first set up of directed-LOS consists of a directed transmitter with narrow beam pattern and a directed receiver with narrow recognition area. This configuration makes the most out of optical transmission power due to narrow outward beam, while the receiver's narrow FOV filters out other noises. Additionally, directed LOS link isn't affected by multipath dispersion because the transmitter and receiver do not interfere with reflectors. There were numerous experiments carried out emphasizing on these advantages of directed-LOS system. In 1985 Yun and Crawford developed a network with core components of directed-LOS links, connection terminals on one side to a central base station. This station then transmitted a 165mW 3-feet-wide vertical narrow beam while receiving from the terminals 5mW 2-feet-wide beam across 50 meters in distance. In order to achieve full duplex transmission, the uplink and downlink used different wavelengths. Chu and Gans also proposed a similar set up with multiple 1mW pencil beam instead of omnidirectional planar pattern from the base station. This helped them achieve final data transmission rate of 50 Mb/s over 30m. Later, there were a number of other competitive products pouring into the market such as the one from BICC Communications with 4 Mb/s through 24 m or one from JOLT Ltd. with 125 Mb/s over 30 m. [4, pp. 7-8]

The directed-LOS system however possesses a big compatibility drawback when it comes to one-to-many or many-to-one mode. This was the main reason for directed non-LOS setup being introduced the first time in 1985 by Photonics Corp. The product is called Photolink, achieving the data rate of 230 kb/s with 22 m range. This configuration concept consists of directing transmitter and receiver towards common reflecting platform in the same room; commonly ideal reflectors are ceiling, painted plaster which can be easily found in most office space. The reflected energy can be calculated proportionally to cosine of 9, the near constant angle from surface normal. This is the result of relatively large surface variations of these surfaces compared to infrared wavelength. The system founded the base for later improved creation from Yun and Kavehrad with multiple directed non-LOS beams against multiple target reflectors, increase disturbance from shadowing. [4, p. 8]

From Photolink the development continued into a hybrid LOS system where active repeater was used instead passive ones, sharing the same wide range of supported configurations. One of the represented products was Minami, achieving the transfer rate of 19.2 kb/s with a base station transmitting 135 mW into beam with 120 degrees wide arc and receivers transmit 75 mW into 60 degrees beam. This system covered approximately 10 m, or half in case of LOS obstruction. With obstructed LOS, the configuration became hybrid non-LOS as depicted in Figure 1-d. Minami project was later on extended by Takashi and Touge, by increasing power and swapping modulation scheme from subcarrier frequency-shift keying to subcarrier phase-shift keying, result in almost 3 times larger data rate. [4, p. 8]

As adapting as these above-mentioned systems were, none of them are suitable to accompany portable devices due to the strict requirement in set up. Therefore, configurations with non-directed links were developed, LOS and diffuse. The first diffuse (non-LOS), non-directed development took place in 1978 by IBM employees, including

Gfeller; this was also the first paper proposed to apply infrared as wireless network's medium. Broad optical beam emitted from transmitter side to be received by wide FOV. The beam of 800 mW was directed to the ceiling acting as a distributor, resulted in 125 kb/s data rate throughout 10 to 20 m in distance. Another recent accomplishes were Photonics Corp transceiver that supported up to 1 Mb/s, came in small size suitable for portable devices and Spectrix Corp's same non-directed non-LOS configuration that achieved 4 Mb/s rate. This setup obviously benefits user experiences the most due to worry-free LOS arrangement. However, it also brings up obstacles from signal attenuation and multipath dispersion. Therefore, the alternative non-directed LOS configuration was developed simultaneously to make up for the disadvantage of non-directed non-LOS. As long as the path remains unobstructed, this allows significantly high speed and distanced applications. For instance, Poulin proposed a multichannel public telephone system, consists of a base station of multiple narrow directed beams pointing to different directions and a receiver with array of narrow FOV detectors facing different directions. This system produced a 230 kb/s transmission rate, covering 20m in distance. Both LOS and non-LOS co-exist to support different environments or cover certain limitations such as large office space, high ceilings or outdoors. [4, p. 9]

2.5 Bluetooth Low Energy

BLE or Bluetooth Smart, the big feature of Bluetooth 4.0 Core Specification, put its first goal to represent the classic Bluetooth as its optimized, smaller brother. However, it turned out to be taking rather distinct and specific direction. From the beginning, initially named Wibree by Nokia and later went under SIG's wings, BLE was already designed to be the low power consumption, low cost, low bandwidth, low power and simplified radio communication rather than another all-purposed Bluetooth. It was the first concept that successfully adopt the standard that extensive operation period can be fuelled by a mere coin cell, despite previous overpromising claims from many other wireless manufacturers. [5, p. 1]

BLE sharp rise in popularity since its birth was due to being in a right place at a right time, without negating great features it offers. This young standard (introduced in 2010) taken over the market by storm with dominating adoption rate compared to other technologies released at the same time. Rising with the first big wave of smart mobile growth, it has been early adopted by Apple and Samsung, clearing the path for other integrations. Particularly, Apple put significant investment in producing a stable BLE stack and guidelines around it. Apple's approval became winning discussion point for any investment or development attempts by silicon vendors. As the bloom of mobile device shows no sign of exhaustion towards cheaper cost and bigger compact power, it rises the constant encouragement for innovative solutions to consumer needs. The birth of BLE lead way to numerous peripheral vendors investment to micro solution market, creating unimaginable cheap wireless technology provided by an all-in-one system-on-chip for less than $2. With the same feature and significantly better priced than WIFI, GSM, Zigbee BLE opens the whole new orientation for communication product design. [5, pp. 1-2]

The additional key success of BLE is its support directing to exchanging data between extensible frameworks, separates it from the traditional Bluetooth specific use cases. Its core concept revolves around centralized data from one accessory to all available communicators without unnecessary knowledge of underlying technology. Manufactures of smartphone understands this advantage and provide flexibility to application developers through accessible APIs to extend BLE framework to whichever directions imaginable. Another worth mentioning offer that BLE provides is seamless integration without interrupting usual user experience. Smartphone owners who have grown accustomed to their device's visual language don't need to be bothered by another layer of strict handshake communications and protocols. [5, p. 2]

Around summer of 2010, Bluetooth SIG introduced version 4.0 for BLE, include Bluetooth Core Specification. After several years in development this update straightened out many existing complications and got BLE into better shape than ever. The latest update of Bluetooth 4.1 in December 2013 is still the viable tool to present day for developer who wants to get their hand on BLE development process, with APIs and improvements included along with core concepts maintained. [5, p. 3]

2.6 Application of BLE in mobile phone and portable devices

2.6.1 nRF51822-EK (Nordic Semiconductors)

Nordic Semiconductors has been a board member of Bluetooth SIG since the very beginning, taken part in constructing the BLE since it was still a concept. Gained popularity in wireless market by its Radio Frequency solutions, it was also the pioneer in manufacturing and distributing early affordable BLE integration, the nRF8001. The newest instalment from nRF51 line of products brings the complete makeover from their predecessors, introducing a radio and 32-bit ARM microprocessor in one chip. [5, p. 75]

2.6.1.1 Technical Specifications

Nordic Semiconductors' nRF51 is a highly integrated system-on-chip, packing the power of BLE–friendly radio and top of the line 32-bit ARM processor into one microchip:

- ARM Cortex M0 core at16 MHz
- 128 or 256 KB flash memory
- 16 KB RAM

Being dedicated as a flash-based device is the characteristic that stands out from nRF51822 from another products' design. The BLE stacks is coded into customizable flash memory and easily updated along with core spec without new silicon revision. Nordic's choice eventually accumulates the manufacturing cost compared to another provider's ROM-based product. However, considering the fast shifting of Bluetooth Core Specification, it gains profit as a long-term investment due to upgradability, leading to faster support to developers. [5, pp. 75-76]

2.6.1.2 Softdevice Architecture

Softdevice is a black box device that Nordic implements on its chip, directly communicates with flash memory and provided core features as BLE stack and peripheral supports. Developer will write code sitting at a flash memory high address and interact with low level SoftDevice as see fits. The description following will refer to only S110 version SoftDevice, which is a pure peripheral solution, while ignore S120 which supports also central role but less commonly used. [5, p. 76]

There are many advantages of SoftDevice design. Having a closed independent BLE stack helps firmware developers concentrate their effectiveness to high level application code, with APIs like GAP (General Access Profile) or GATT (General Attribute Profile) while trusting SoftDevice to make reliable decisions on low level protocol such as security or message validation. The architecture also takes care of radio configuration instead of programmers, one of the most mistake-prone parts of any RF product development. Compacting low level operations and validations into black box protects application development mistakes and ensure BLE function properly according to Bluetooth Core Specification, thus speed up the product development process in general. Additionally, SoftDevice concept emphasizes one-to-many relationship: one hardware providing support to variety of protocols including custom ones, thus it softens the product cost. Last but not least, the architecture SoftDevice built upon provides user with independence and reliability. The application does not intrude developer's implemented code with side effects from updating or peer dependencies. [5, pp. 76-77]

On the other hand, SoftDevice is resources demanding. The S110 version with 256 KB storage takes up to a third of flash storage and half of SRAM for operation. It also requires developer to take into account latency from architectural code flow. However, the benefit SoftDevice brings outweigh some drawbacks in delays and hardware limitation. [5, p. 77]

2.6.1.3 Operating nRF51822-EK

The best platform to start with nRF51822 is Nordic's nRF51822-EK. The development kit comes with two development board: PCA10001 and PCA10000 as shown in the figure below.

Figure 4. Nordic Semiconductor's nRF51822-EK [5, p. 77]

The PCA10001 is the main development board, with all pins available for connection of I2C, SPI sensors or other peripherals. It also includes J-Link to program and debug MCU as well as gain additional control over power intake and usage. The UART gate communicates output message to other accessories or receive simple commands. The board also equips with CR2032 battery slot for extra power supply. [5, pp. 77-78]

The smaller PCA10000 is an USB dongle used mainly in debugging through Nordic's Master Control Panel or from data pushed out to Wireshark. It also features a development board on its own, with Segger J-Link for external programming and debugging. [5, pp. 77-78]

2.6.1.4 Example and toolchains

After received the nRF51822 board, developer can register for an account on Nordic's website with purchase serial number. This will open access to several examples for the development kit to familiarize when one first starts coding. The demo makes use of Keil's uVision or IAR toolchain that freely available for non-commercial projects with size less than 32 KB. Other samples also include instruction for GNU-based tools, despite being inferior to what Keil uVision has to offer. However, this support extends application development beyond Windows limit to Linux and OS X. [5, p. 78]

2.6.2 CC2451DK-MINI (Texas Instruments)

Texas Instruments (TI) CC2541 is the first representation from the American manufacturer to enter peripheral BLE market. It features:

- 8051 core running 2.4 GHz
- 128/256 KB available flash storage
- 8 KB SRAM

The advanced factor about TI kit is full features support, especially up-to-date with the latest Bluetooth Core Specification version 4.0 compared with another vendors decision to make many of them optional. The CC2541 includes USB support to SoC and pin-compatible with other members of this chip family, which reduce the effort in additional design by allowing moving designs to peripherals connected via USB. Moreover, via relentless testing and design polish, TI leaves a significant design lineage of RF chips coupled by well-tinkered radios and additional resources for both hardware and software usages. This brings considerable ease for smaller companies that does not possess many in-house RF or firmware design expertise. [5, p. 79]

The biggest disadvantage of choosing CC2541 comes from its outdated 8051 core, which also requires costly compiler and IDE to operate (IAR Embedded Workbench). With the rising popularity of powerful modern ARM Cortex-M cores, TI no doubt will put investment into upgrading their slow and expensive core to stay competitive. [5, p. 79]

The CC2541DK-MINI represents the CC254x family from TI as the entry development kit for peripheral designers wanting to getting familiar with the platform and SoC.

Figure 5: TI's CC2541DK-MINI development kit [5, p. 80]

This kit includes a hardware debugger, USB dongle connected to PC and play the role of master device, and a key fob development board for programmer's code. The whole kit resembles a real product, includes a plastic cover with two physical buttons to provide real feedback during development or debugging. While not particularly useful during early development stages, it provides generic real-world experimental opportunities of BLE device operation such as attaching on other objects or simulating obstructed locations. This advantage adds up valuable benchmarks before mass production. [5, p. 80]

2.6.3 Other Hardware Platforms and Modules

If vendor's RF peripherals does not suit one's needs, there is the approach of using modules alternatively. The best part about taking advantage of modules is that most of them come with certification from FCC or CE/ETSI, enable them with desirable stability through any test cases by different protocol bodies like Bluetooth SIG. The certification alone often times cost more than ten of thousands of dollars per product, promoting modules as the most economical option for production of lower volume. Furthermore, by purchasing module developers do not have to worry about knowledge or tool demands when designing RF hardware. Poorly designed product affects directly to efficiency and covering range, thus relying on modules manufacturer to tune the RF front-end, antenna or connectors independent of impedance from desired external antenna is a wise choice. Besides the ease from the hardware level, many modular products also layer with high level scripting language that significantly minimize development time and difficulties dealing with low level APIs such as Keil's uVIsion from Nordic's nRF51822-EK or IAR from TI's CC2541MINI-DK; only text editor program is required. [5, pp. 80-81]

As good as any system can get, modules come with consideration in mind. The per unit price costs a lot more than integrated circuit such as ones from Nordic Semiconductors or Texas Instruments. Module maker strength emphasizes on dividing costs across design, certification and testing in low volume manufacture but lose its effectiveness as the number grows upward 10000. [5, p. 81]

There are 3 modules currently gaining popularity that will get following brief introductions:

2.6.3.1 Laird's BL600

Laird's BL600 is inspired by Nordic Semiconductor's nRF51820's line of products. Besides most capabilities featured in nRF51822, it includes a higher-level programming language of smartBasic; developer could use this event-driven language to create all types of basic applications without spending time and money in commercial IDEs or C, C++ training, although standard Nordic's toolkit is still viable option. The module comes with regulatory certifications from CE/ETSI, FCC, Industry Canada, NCC and Bluetooth SIG Qualification, available at various different online component distributors [5, p. 81]

2.6.3.2 Bluegiga's BLE112/BLE113

Bluegiga's BLE112 and BLE113 modules use the base of CC254x family from Texas Instruments. They come installed with BGScript, an XML protocol for application development. A C-language API is also included to communicate with modules over UART. BLE112 and BLE113 only significant difference is that 113 module utilizes input power better and provides a hardware I2C port which is useful in connecting different low-cost sensors such as temperature, accelerometers or pressure sensor. Bluegiga's modules are certified by CE/ETSI, FCC, Industry Canada, Japan and South Korea and can be purchased easily from major resellers like Digikey, Farnell. [5, pp. 81-82]

2.6.3.3 RFDuino

RFDuino is a small BLE module enabled through Arduino IDE to create BLE devices. It provides great starting point in fidgeting with BLE, equip developer with basics and methods that improves and polish prototypes. CE/ETSI, FCC and Industry Canada certifies RFDuino. [5, p. 82]

2.7 iBeacons

2.7.1 Introduction

Bluetooth proximity beacon, or iBeacon, is the first technology that combine virtual platform, handheld devices and location-based actions. The main principle of iBeacons is based on its ability to broadcast identifiers which allow devices operating within a small radius to detect and react upon; thus, applications and services can be constructed to support and accompanies such behaviours. iBeacon also refers to all BLE proximity peripheral submitted under Apple licensing program for testing; Apple therefore documents owns development API with the same name. Due to the its flexibility and simplicity, developer has created various applications for indoor pathfinding, queue management or facility’s guide featuring beacon device. [6, p. 2]

Among many perks that make iBeacons become obsession of developers and manufacturers, there are two deciding factors leading to its popularity. First and foremost is contribution to mapping significantly more precise point of interest’s location around mobile devices. This simply outperform GPS, especially with indoor environment where centimetres difference can be distinguished by iBeacons. Moreover, all location data is available for grab, making development of applications taking advantage of such information straightforward and limitless. [6, p. 2]

2.7.2 Main principle

iBeacon strength revolves arounds introducing “proximity” as the base for all operations, versus the familiar idea of using “location” as the pure identifier for determining “where”. While occasionally interchangeable in meaning, there are distinctions that separate the two. Location refers to exact geographical point defined through a coordinate system, a named address defined through street, city or country. The most popular way to determine one user’s location is through GPS or cell tower mapping, results in location info accuracy up to street level. This is useful enough for navigating between blocks and buildings, finding event area. On the other hand, proximity only aware about surrounding area and other point of interest within. The data brought from this concept is relative positions of all covered entities in defined radius against a central point, or each other. However, the collective of this info can be effectively applied to locate indoor targets, where GPS coverage is handicapped. The usual method is triangulation - that is inferring to a point in space through a proximity collection from a system of three or more facilitated proximity points, each at known location. This method brings significantly more accurate description of one's location within a small or confined space, i.e. buildings or basement. [6, pp. 3-4]

iBeacons take this principle by heart and conceptualize the proximity in most practical ways one can think of. The best example is indoor navigation - in museum, big fair or conference venue - by polling for one's proximity against certain item of interest. The collective clear info of what is near you and what is not, how near it is provided from beacons measured up to the accuracy of position. This relieves the user from dependence of error-prone location system, as proximity only emphasizes on interest item or event within vicinity. The main reason is that proximity is provided through different technologies than location, scaling down significantly from big outdoor navigation network of satellites or cellular to short range Wi-Fi or BLE. [6, p. 4]

2.7.3 Applications

2.7.3.1 Indoor location and guiding

Navigation inside big facilities often poses challenges, even with implementations with overwhelmingly reoccurring maps. With the help of an iBeacon network placed reasonably over the floors, visitors' accurate location within the building or direction from one point to another is clearly mapped. This map replacement method does not only provide more accurate solution but also instantaneously update to reflect real time changes in user's position. [6, p. 6]

In a similar field, there are big airports or station hubs around the world whose minutely and daily flow of information might deem overwhelming to infrequent travelers. Accompanied by a ticket management functionality, applications can assist mobile holders with gates information as well as directional guiding through beacons. [6, p. 6]

In many built-up cities, subway or underground pathway usually confuses commuters, or instructions provided are too vague for an optimal path finding. By using proximity technology, applications could be developed to incorporate it, analyzing and coming up with a correct and quick direction suggestion for riders or walkers. [6, p. 6]

One of the most practical use cases for indoor direction finding is parking assistance. Relying on GPS to find one's car in a huge parking place is often unheard of, or plainly too technological impaired to be applied; furthermore, the car alarm integrated into most car keys nowadays is not useful until being in close proximity to the car itself. With beacon attached to owns car, the guiding will be independent to the garage parking spot itself and still effective inside. [6, p. 6]

Lastly, mention should be made for the current most popular usage of proximity peripheral: enhanced experience for commercial's or tourist's purpose, specifically in museums, attractions or retail shops. [6, p. 7]

2.7.3.2 Proximity trigger action

Besides using beacons and proximity as a virtual map, more actions can be invoked through defined relative location between stationed broadcasting peripheral and user's mobile device. An example one could find in some of trade fair or technology gathering is mobile advertisement. Devices that moved close enough to a display, banner or a point of interest could get a push notification containing more information about products, sale or offered coupon from the store. [6, p. 8]

There are integrations with transportation industry too, due to the heavy shift from physical to digital microservice making electronic tickets more and more common. While the convenience is already enhanced, using proximity to signal train conductors with valid traveler's ticket to be displayed create even more seamless experience. [6, p. 8]

One comfort beacon proximity triggered mechanism could bring to the doctors and nurse are treatment and medical records by beacons could be attached to patient's wristband. All health history and status can be collected real time and conveniently by patrolled nurses or doctors. [6, p. 8]

2.7.3.3 Queue management

The advantage of BLE technology can relieve considerable amount of anxiety in queuing for services. By installing beacons along the queue lines businesses can measure transaction times and average queue time that update live for waiting customer. While it does not improve the actual queue time, customer being communicated with information considerably improve their experience. The same contribution could be brought to restaurant business and the frequent impatience taking place while ordering and begin waited. [6, p. 9]

3 EXPERIMENTAL AND PRACTICAL APPROACH

3.1 Problems

The research problem arose as part of the project between Luxus Oy and the contractor, owner of one of the largest malls in Finland's Metropolitan region. They proposed the need to have a system, hardware and software included, to serve as an important part of one promotional campaign, happening during a period of two to four weeks. The implementation and architectural design was left to decide freely by Luxus, as long as it could be shaped to fulfil one ultimate goal: encouraging people to spend time and engage with shops located inside the mall and itself, in a format of challenges and rewards. There were not any long, back-and-forth discussions on methods to achieve this challenge; due to one previous project working successfully with beacon technology and BLE, the development team chose it to be the lodestar.

3.2 Existing library and resources

There are a few acknowledged beacon producers providing APIs and framework to make use of their products. They usually stick to the business model of $30 average per beacon with the attached documents and instructions, making a complete development kit.

3.2.1 Estimote

Estimote is arguably the biggest and most well-known beacon manufacturer, being one of the frontiers back to when IoT reached its boiling point. Besides physical beacon devices, Estimote offers a variety of SDKs ready for developers, covering from most popular to even more advanced usage of proximity and indoor location technology.

3.2.2 Kontakt.io

The European manufacturer brings nice, simple and easily customized beacons to developers everywhere. Additionally, the supplier provides an extensive back-end and CMS.

3.2.3 Gimbal

This is the only product originated from a big manufacturer, Qualcomm. Known widely for their majority in mobile phone processor production, Qualcomm decided to incubate Gimbal to lead their way to Proximity Data Platform.

3.3 Required tools

The final goal of the project is creating the gamification mobile application in which visiting different shops inside the shopping mall will gain users with immediate or accumulate point system for bigger awards. With clear expectation in mind, it presents the classic case of proximity-triggering-actions application. Thus, the ingredients for Luxus' project should at least consist of:

- iBeacons broadcasting BLE: at least one setup for each shop in the mall, addition to more at all entrances or other desired monitored locations.

- Mobile application: receiving proximity information streaming from different recognized beacons, incorporating with a cloud service to respond with correct actions as well as registering interested statistics.

- Cloud service: handling all data sent through applications and reacting to configured conditions

3.4 Technical solution

3.4.1 Prerequisites

There was native support for BLE or beacons from both Android and iOS platform, with official document from Google and Apple that was straightforward and easy to implement. However, since the project demanded a simple yet coherent ecosystem from application to backend service sides and the decision has been made to rely on MobileBridge for their CMS, it was convenient to also make use of their mobile SDK for the applications. Due to the dated timeframe that this project took place, the following demonstration will be in Objective C and xCode for iOS, Java and Eclipse IDE for Android, reflecting the original state of then application.

3.4.2 Project configuration

3.4.2.1 iOS

The setup for iOS project is handled totally through xCode, mostly by adding dependencies and the dedicated framework provided by MobileBridge.

The two most important items are: MobileBridge.bundle file containing resources and graphics and MobileBridge.framework containing implementation logic. Aside that, there are a few native libraries necessary for linking such as MapKit.framework, CoreLocation.framework, QuartzCore.framework, Social.framework, SystemConfiguration.framework, CoreBluetooth.framework, Foundation.framework, UIKit.framework necessary for the MobileBridge SDK to perform. Lastly, Amazon Web Service (AWS) SDK has to be downloaded and located into the project folder

Additionally, there are some configurations for the project plist file:

- FacebookAppID (only required if Facebook sharing is desired): the App ID obtained registering the app with Facebook.
- NSLocationAlwaysUsageDescription: Describes the reason why the app accesses the user's location information.
- NSLocationWhenInUseUsageDescription: Describes the reason why the app accesses the user's location normally while running in the foreground.

- ibeacon_uuid (only required if ibeacon notifications are desired): A 128-bit value that uniquely identifies one or more beacons.

3.4.2.2 Android

The settings for Android, beside the certain differences from xCode due to nature of IDEs, are quite similar through Eclipse.

The first important step is ensuring that MobileBridge distributed JAR (Java ARchive) file added into the library list and build path of the project. In Android, Google Play Services Library is required, imported by Eclipse native package manager. One important note is app manifest file needs to be included with following tag:

```
<meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
```

Afterwards, the same AWS SDK must be installed.

3.4.3 Initilize and setup

3.4.3.1 iOS

First and foremost, MobileBridge Framework has to be declared in the header file of AppDelegate. AppDelegate is the central point focusing on handling application states, in specific when it starts all configuration and construction or before cleaning up and termination. This includes declaring the framework in the header file AppDelegate.h; then adding support for MobileBridge own delegate functions (MobileBridgeLibraryDelegate) and declare the MobileBridgeLibrary object, set to desired variable name for instance mbLib:

```
#import <UIKit/UIKit.h>
#import <MobileBridge/MobileBridge.h>

@class ViewController;

@interface AppDelegate : UIResponder <UIApplicationDelegate,
MobileBridgeLibraryDelegate>
{
    MobileBridgeLibrary* mbLib;
}
@property (nonatomic, retain) MobileBridgeLibrary* mbLib;

@end
```

Afterwards we have to initialize some required variables and functions in the implementation file AppDelegate.m. Synthesizing mbLib object that we declared in header file is advisable for easier user:

```
#import "AppDelegate.h"

#import "ViewController.h"

@implementation AppDelegate
@synthesize mbLib;
```

Within didFinishLaunchingWithOptions method activationCode received when registered to MobileBridge needs set, delegate needs to be self-assigned to AppDelegate file and mbLib is initialized:

```
NSString* activationCode = @"xxxxxxxxxx";

// Init MobileBridge Library
mbLib = [MobileBridgeLibrary InitLibrary:activationCode];
if(mbLib!=nil)
{
    mbLib.delegate = self;
    NSLog(@"Library initiated properly");
}
else
{
    NSLog(@"%@",@"MobileBridge Library failed initialization.");
}
```

Finally required MobileBridgeLibraryDelegate methods need implementations.

```
- (void)MB_SettingEventReply:(NSDictionary*)response;
- (void)MB_NetworkAlertPrompt:(int)alertId;
- (void)MB_LocationUpdate:(int)position max:(int)max;
```

3.4.3.2 Android

Android's setup process for MobileBridge is also straightforward. In the main class MyApplication or whichever one chose when creating a new Android project, mbLib constant is added:

```
package [package name];

import com.mobilebridge.library.MobileBridgeLibrary;

import android.app.Application;
import android.content.Context;
import android.content.res.Resources;
import android.graphics.Point;
import android.net.ConnectivityManager;
import android.util.DisplayMetrics;
import android.view.Display;
import android.view.WindowManager;


public class MyApplication extends Application
{
        public static MobileBridgeLibrary mbLib = null;
        public static Point gScreenSize = null; // keep screen size

        @Override
        public void onCreate()
        {

        }
}
```

Then the same application class needs to be declared as an attribute of the <application> element in the AndroidManifest.xml file:

```
<application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="@string/app_name"
        android:name=".MyApplication">
```

In the next step there requires initialization of the MobileBridge library in the MainActivity.java file by implementing MobileBridgeLibraryInterface:

```
public class MainActivity extends
        Activity implements MobileBridgeLibraryInterface
```

By click MainActivity and choose Add Unimplemented Methods, the following methods are added:

```
public void MB_LocationUpdate(int position, int max)
public void MB_NetworkAlertPrompt (int alertId)
public void MB_SettingEventReply(JSONObject response)
```

Then the activation code you received from MobileBridge needs to be declared as a variable:

```
String activationCode = "[activation code]";
```

In the onCreate method, MobileBridgeLibrary object initialize function is called:

```
MyApplication.mbLib = MobileBridgeLibrary
         .InitLibrary(context,activationCode,this);
```

There is an important side note at this point is that MobileBridgeLibrary needs to be imported, by clicking MobileBridgeLibrary and choose “Import 'MobileBridgeLibrary' (com.mobilebridge.library)”.

Lastly, AndroidManifest.xml file is modified with addition of all demanded permissions:

```
<!-- Camera-->
<uses-permission android:name="android.permission.CAMERA"/>

<!-- Allows access to the list of accounts in the Accounts Service -->
<uses-permission android:name="android.permission.GET_ACCOUNTS" />

<!--Permissions required by the Android Cloud to Device Messaging Framework -->
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
<permission android:name="[applicationPackage].permission.C2D_MESSAGE"
        android:protectionLevel="signature" />
<uses-permission android:name="[ApplicationPackage].permission.C2D_MESSAGE" />

<!-- internet connectivity-->
<uses-permission android:name="android.permission.INTERNET"/>
<!-- access information about networks -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<!-- access to phone state -->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- Allows an application to initiate a phone call -->
<uses-permission android:name="android.permission.CALL_PHONE" />
<!-- keep processor from sleeping required  for notification -->
<uses-permission android:name="android.permission.WAKE_LOCK" />
<!-- Allows an application to write to external storage -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<!-- Allows an app to access approximate location cell towers and Wi-Fi-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<!-- Allows an app to access approximate location GPS-->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

<!--  Allows the API to access Google web-based services -->
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
```

If one’s application needs geofence notifications, the following declaration should be pasted as a child of the <application> element:

```
<!-- GeoFence proximity Receiver -->
<service
        android:name="com.mobilebridge.proximityAlert.MB_ProximityReceiver"
        android:label="@string/app_name"
        android:exported="false">
</service>

<!-- proximity update receiver -->
<receiver android:name="com.mobilebridge.proximityAlert.proximityReceiver"
android:enabled="true" />
```

If iBeacon notifications are required, this beacon declaration should also be added as a child of the <application> element:

```
<!-- beacon clear receiver -->
 <receiver android:name="[applicationPackage].beacon.clearbeaconReceiver"
android:enabled="true" />
```

4 RESULTS

The aim of Luxus' project was kept original to the initial proposal idea to customer, though the process was more flexible due to obstacles faced throughout development time. It was the ecosystem including iBeacons installed around the shopping mall, mobile application as a medium to receive proximity info and communicate to service that manage records and rewards.

By using MobileBridge service for registering beacons, user profiles and bonuses, a simple mobile application written in web languages wrapped in Cordova and beacons purchased from Kontakt, the project has been carried out relatively fluently. The service recorded a few hundred users, around a third were active ones collecting rewards and interacting with beacons. The application ran smoothly without any major bugs. Though the project's uptime was short, it panned out nicely enough for the amount of content it offered.

Unfortunately, there were almost no further extensions to the project's core features. It was a pity because this development took place at the high rising trend of BLE and IoT; thus, it could gain a lot more attractions and become something substantially more advanced.

To minimize problems during project development, the application side was made up from a simple html visual front and a BLE native layer supported natively from two chosen platforms. The service was single handedly managed by a third party, reducing significantly development resources. Expectedly, almost every unexpected trouble arose from the little experience of working with BLE devices and communication mechanics. The two main problems were:

- Vertical detection: since main idea of the project was to get people visiting physically to certain shops, it posed as a big disadvantage for Luxus' client and their participants when the frequent "successful visit" from the different floor's storefront was detected. While there was no solution to the root of the problem itself, which was the rather expected spherical proximity radius of beacons, reacting distance was being tweaked and tested multiple times to limit its occurrence chance.

- Inconsistent reactivity: while almost every possible human error from both application side and user's behavior had been eliminated, it recorded unexplained inconsistencies in proximity monitoring and reactions. Even sampling by having one tester going around with two phones running simultaneously the application resulted in different "successful visits". There were many suspects investigated: operating systems (with their own settings of BLE support), internet connection (to contact and input visited shop into user's data), application states (foreground or background). However, tests carried out with each of the criteria as constants or variables did not produce the reasonable results. Due to the short operation time of the project the most probable guess would be the inconsistent nature of Bluetooth or Bluetooth LE along with occasional deterioration of cellular connection.

5 DISCUSSION

5.1 Possible expansion

As mentioned earlier, there was no further development added to the finished project; a months-long research and effort into BLE features ended up being one short sprint. Hence, the team agreed that although the technology enables better results, due to the limited nature of contract and customer's vision, the project fell short of expectations.

One simple configuration that could make the experience better is registering iBeacons proximity-visited flags to device local storage and having the background job polling

Mobile Bridge service to register it to user's database. That way the user would be welcomed immediately with an expected reaction in every shop visit instead of variety in input time to remote server and awaiting response. On the other hand, while the standard at the time only just reached Bluetooth 4.0, there were possibilities for more compact message and data to be communicated between the beacons and mobile's application than just proximity readings. This would also help make the experience more fluent and reduce the unreliability of cellular data in-house or public Wi-Fi network.

Furthermore, if the project had been developed with a wider vision in mind there would have been more possibilities. For example, instead of having the server handling and react to individual user being in the beacon's vicinity there could be a group reward or event. The routing implementation for facilities could also be a big upgrade; with a reasonable network of beacons installed among shops, indoor navigation would be a nice addition.

5.2 Indoor routing

Indoor routing or pathfinding is not necessarily an advertised BLE feature. However, by providing reliable proximity measurements which work well inside, where GPS and other form of location-based technique fail, beacons can be helpful in navigation. For example, Steerpath Ltd framework, a Finnish company working in this area could be consulted to demonstrate this feature.

Steerpath Ltd provide service packages and easy installation of indoor positioning, including venue map creation, preconfigured plug-and-play beacons and supporting on SDK implementation. There is also an extended service package for indoor navigation and separate map management service hosted on their server. Though provided to developer as a black box module, at its core their solution is still based on Bluetooth and triangulation, combining with map asset to pinpoint the exact location of a user to the premise.

6 CONCLUSION

During my very first months in Luxus Worldwide I have had the chance to participate in a project for a shopping mall, with the demand on creating a gamification platform for visitors and their shops to further promote themselves and attract buyers. The project took place during a boom of BLE and iBeacons, with the goals, needs and technical capabilities lined up. It made use of the core features from BLE – indoor location, proximity and interaction with proximity changes – through mobile application built on Android and iOS, backing up with a service for data management, creating extensive communication channels with client sides. iBeacons were registered to backend service and positioned to reasonably cover most of shops' front area or interested gateway; the mobile app would then react based on proximity of customers versus shops to reward them with points which later on could be exchanged to different prizes.

This project was partly successful due to the straightforward technical requirements and the simplicity of the features required. However, there were a few problems recorded during testing and even when user experience was examined. Apart from some client – backend communication problems during the first days, the most critical problems are related to vertical proximity.

This final year project was mostly successful apart from some minor technical issues. It was unfortunate that the client desired no further investment on the technology and infrastructural setup for the event to become a permanent interactive platform for the mall. Therefore, discussions on extending beacons' capability to indoor routing, advertisement, or remote checkout were brought to a halt. Because this study was carried out at the beginning of the fast rise in BLE and beacons usage for both commercial, industrial or individual purposes, the technology was not as developed as it is today. Nonetheless, the author was among a pioneer group of developers to create one of the first commercial ecosystems for BLE. It has been interesting to observe the popularity this technology has achieved and the way it has improved.

References

[1] H. Abloid, H. Afifi and C. De Santis, Wi-Fi, Bluetooth, Zigbee and WiMAX, Dordrecht: Springer, 2007.

[2] A. Lehto, Radio Engineering for Wireless Communication and Sensor Applications, Artech House, 2003.

[3] K. L. Gaotao S, Signal Interference in WiFi and ZigBee Networks, Springer International Publishing, 2017.

[4] J. R. Barry, Wireless Infrared Communications, New York: Springer US, 1994.

[5] K. Townsend, K. Cufí, A. and R. Davidson, Getting Started with Bluetooth Low Energy, California: O'Reilly Media, Inc., 2014.

[6] M. S. Gast, Building Applications with iBeacon, Sebastopol: O'Reilly Media, 2015.

[7] N. Gupta, Inside Bluetooth Low Energy, London: Artech House, 2013.

[8] H. Lehpamer, RFID Design Principles, London: Artech House Publishers, 2012.

[9] "About - Gimbal," [Online]. Available: https://www.gimbal.com/about/. [Accessed 25 7 2016].

[10] "About," [Online]. Available: https://www.gimbal.com/about/#team. [Accessed 25 7 2016].

[11] MobileBridge, "MobileBridge Documentation," [Online]. [Accessed 2014].

[12] Steerpath, "Bitbucket," 2017. [Online]. Available: https://bitbucket.org/nimbledevices/steerpath-ios-examples/src. [Accessed 2018].