



Norwegian University of
Science and Technology

Low Energy Wireless ECG

An exploration of wireless
electrocardiography and the utilization of low
energy sensors for clinical ambulatory
patient monitoring

Rikard Bjørstad Eide

Master of Science in Informatics

Submission date: June 2016

Supervisor: Pieter Jelle Toussaint, IDI

Co-supervisor: Frank Alexander Kraemer, ITEM

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

In this project we have investigated and assessed wireless patient monitoring solutions. Through qualitative interviews we have investigated the infrastructure and practice of wireless ECG monitoring systems at two Norwegian hospitals. Combining what we learned from the existing solutions with different WBAN implementations found in literature, we propose a design for a wireless monitoring system based on available technology and open standards. We validated this design by building prototype that test certain critical aspects of wireless ECG monitoring. The prototype was based on a Bluetooth Smart enabled node, an Android gateway and a modern web-server for real time streaming.

In conclusion we see that the current version of Bluetooth Smart is able to support continuous streaming of clinical grade raw ECG at a 1000 Hz sampling rate with 32 bit samples. A BLE enabled ECG device can support this throughput and still have more than 4 times the battery lifetime compared to the existing solutions we investigated. We have also assessed the end-to-end delay in our proposed design for a monitoring solution, and showed that for one of our use cases, this delay averages around 60ms.

We have evaluated the technology backing a WBAN enabled hospital as ready for clinical usage, but don't believe the current organization of clinical technology will change before outside factors make the manufacturers rethink their current business models.

Problem Statement

Electronic patient monitoring has been practiced at hospitals for almost half a century. Metrics like ECG, arterial blood pressure, SpO₂, respiratory rate and body temperature are measured on every patient in a modern intensive care unit. The consumer industry, the driving force behind many technological advances, have the last couple of years also showed interest in capturing physiological data. With this interest comes new ideas, new solutions and new technology from companies that embed a customer centric focus. The health care has over the years grown into a collaboration of technologically isolated silos, whereas the consumer industry is working towards a ubiquitous reality. In this reality every device and service is connected to each other, and available at any given time. Having the ability to monitor several physiological metrics wirelessly holds potential to improve medical decisions, reduce cables, ease everyday routines, as well as giving ambulatory patients the freedom to move.

Low energy physiological sensors is a promising technology that will have an increased importance for medicine in the future. These sensors cannot be wired, hence there are many practical (ease of use, privacy, security, compatibility, value, price, safety etc.) and technical (interoperability, minimum delay, maximum throughput, maximum network lifetime, minimum energy consumption) challenges that need to be addressed.

Among different physiological metrics, ECG makes for a reasonable study for two reasons:

- There are high technical requirements to medical grade ECG.
- ECG is already monitored wirelessly today through telemetry solutions, which makes for a good case study.

Today's telemetry systems have short battery lifetime, and the systems and infrastructure are specialized for ECG monitoring. Wireless monitoring today is therefore expensive and offer little flexibility. In this master thesis we want to answer the following: *Is it possible to create a monitoring solution for wireless ECG, based on available technology and open standards?*

Preface

This thesis is the result of a collaboration between the Department of Computer and Information Science (IDI) and Department of Telematics (ITEM) at NTNU and was conducted during the fall and spring semester 2015/2016.

For this research project we originally wanted to investigate how feasible it was to introduce today's commercial wireless low-energy technology in a medical environment. What were the shortcomings? Where are the barriers? What are the requirements? As discussed in Section 3, our original approach to answering these questions was through the development of an prototype that solved the practical problem of ambulatory patient monitoring.

Due to the multidisciplinary nature of the topic, the focus has shifted and been changed along the way. In the end we have evaluated the low energy features of Bluetooth Smart on a set of requirements using a mix of available tactics and research methods.

This project was supervised by Prof. Pieter Jelle Toussaint from the Department of Computer and Information Science and Associate Professor Frank Alexander Kraemer from the Department of Telematics at NTNU.

All source code related to this project will be available at <http://rix1.github.io/master/>

Contents

1	Introduction	8
1.1	Background and motivation	8
1.2	Problem Statement	10
1.3	Research Questions	10
1.4	Scope	11
1.5	Contributions	11
1.6	Structure	11
2	Literature Review	14
2.1	Electrocardiogram	14
2.2	Bluetooth and ambulatory ECG	17
2.3	Wireless Body Area Networks	20
2.4	Interoperability	21
3	Research Strategies	24
3.1	Research Methods	24
3.2	Interviews and observations	25
3.3	Evaluation	27
3.4	Document and literature analysis	27
4	Clinical Context	28
4.1	Intended use	28
4.2	Existing Solutions	30
4.3	Use Cases	35
5	Technology Assessment	42
5.1	System architecture	42
5.2	Sensor Node	46
5.3	Gateway	50
5.4	Server	54
5.5	Evaluation	56
6	Discussion	64
6.1	Toward a Multi-Vendor Situation?	64
6.2	Research methods	65

6.3	Implementation	67
6.4	Results	68
6.5	Future work	69
7	Conclusion	71
	Glossary	72
	References	73
	List of Figures	82
	List of Tables	83
	Appendix A SoC Considerations	84
	Appendix B Data formats	86
	Appendix C Battery Lifetime	87
	Appendix D ECG Plot	90
	Appendix E Experiment setup	92
	Appendix F Gateway Screens	94

This page has been intentionally left blank.

1 Introduction

In order to give some context, we will start this chapter with some background and motivation for this master project. This leads up to our problem statement, research questions and scope, where we define the goals for the project. We will end this chapter with an structural overview of the rest of this thesis.

1.1 Background and motivation

During the last couple of years there has been a significant increase in the number of connected devices. Hardware is getting cheaper and smaller, and the demand for information is growing. Economic prosperity, aging population, the growing middle income population and sensitive public policy are key demand drivers for better healthcare and infrastructure. In addition, a population growth of 4 billion is expected within the next 90 years [59]. In order to facilitate these changes, we need infrastructure that is cost effective, sustainable and that scales well. How the healthcare advances in this time of change will have a major impact on how this expected population growth impact our societies.

In recent years we have seen an increased effort made by both big and small actors in the technology industry in trying to commercialize health technology. They offer devices that track and monitor activity, sleep and physical traits like weight and glucose levels to name a few [22, 21]. Some suggest that having this increased access to information about our health and bodies might shift the entire model of medical care; that we are at the brink of a structural change, from a reactive to a proactive healthcare [45, 49, 47].

One of the most radical changes to todays clinical practices is likely to come from the prevalence of new sensory technology, similar to the ones being developed for the consumer market. New measuring tools and techniques, embedded in small, low energy wireless sensors, could enable an affordable monitoring system offering patients as well as physicians good user experiences.

In short, there will soon be a need for a low cost, wearable, wireless system for monitoring physiological health information in a clinical setting (PHI). This system must offer both patients and physicians the same services and level of performance and quality they get from existing monitoring systems, but with an improved user experience and flexibility to match the expectations users have from modern consumer products.

One possible orchestration of these sensors are in a wireless body area network, or WBAN for short. This is a collective term for wireless networks situated

inside, on or around the human body. This means entire systems of devices communicating with each other in immediate proximity of a human body. Researchers have spent the past 20 years investigating applications, optimizations and possibilities of WBANs. The technology however, has until recent years not been small enough nor energy efficient enough for sensor networks like these to be practically feasible for patient monitoring.

Electronic patient monitoring is today an established practice at modern hospitals. It involves *repeated or continuous observations or measurements of patients, their physiological functions, and the function of the life support equipment* [26]. Studying monitoring systems in a multidisciplinary task. A system that affords good management decisions, can only be created by studying and finding the optimal compromise between several design factors; clinical, engineering and economic [42]. On the contrary, personal monitoring devices like fitness trackers, blurs the lines of what monitoring might mean. Because of this ambiguity, we use the following definition of “monitoring” throughout this thesis:

Monitoring: A repeated or continuous measurement of the physiological functions of one or more patients conducted in a clinical environment.

Traditionally, monitoring vital signs has been conducted only on patients in given situations or with special needs for monitoring, and different physiological metrics are captured individually. An electrocardiography (ECG) test produces one example of such physiological metric. First invented by Willem Einthoven in the early 1900’s, it is a technique that captures the electrical activity of the heart over time. Today, ECG is monitored in several different ways, both wired and wirelessly. This thesis will focus on the latter, which consist of two different approaches, depending on the diagnostic need and intention. Telemetry is the process of sampling measurements and transmitting them to receiving equipment for remote real-time monitoring. An ECG telemetry system is sampling and transmitting the heart’s electric activity of patients wirelessly inside a hospital. Another wireless approach to capturing ECG are Holter monitors. These are battery driven devices that capture ECG continuously for 24-48 hours. These are typically worn outside the hospital in every day life. One drawback with a Holter monitor, is that they collect data. Data processing and analysis is done offline, and the devices often have to be sent to the manufacturer, using days or weeks before the results return [80].

Throughout this thesis, ECG will be used as a benchmark for monitoring

physiological metrics because of its technical requirements, and the availability of existing wireless ECG solutions today.

This thesis was motivated by an interest for clinical information systems and the usage of new technology at hospitals. Experience with development of wireless proximity solutions for the consumer market as well as experience with hospitals through prior employment.

1.2 Problem Statement

Today's telemetry systems have a short battery lifetime, and the systems and infrastructure are specialized for ECG monitoring. Wireless monitoring today is therefore expensive and offer little flexibility for both its users and maintainers. We asked ourselves, is it possible possible to create a monitoring solution for wireless ECG, based on available technology and open standards?

What's the value of using available technology? Available technology is cheaper, more developer friendly and more flexible than expensive specialized solutions. Because of the user centric intentions that lies behind much of today's available technology, this technology is often more interoperable and compliant with other solutions. This emphasize the importance of using open standards as it enables different sensor and service providers to co-exist and take part in the same infrastructure. This synergy makes it cheaper and faster to improve and innovate, two factors that are becoming increasingly more important in today's clinical practice.

1.3 Research Questions

Our research questions aims both wide and deep.

1. *What technical components are involved in a ECG monitoring solution?*
 - *What is the practice of ECG monitoring used at hospitals today?*
 - *What factors are most critical when conducting wireless, clinical grade ECG, and what are the minimum requirements for bandwidth, battery capacity and end-to-end latency?*
2. *Is Bluetooth Low Energy able to fulfill the requirements of clinical grade ECG?*

1.4 Scope

Due to the multidisciplinary nature of this project, scoping the research and clearly defining our goals was crucial. For this project, this meant excluding important quality attributes like security, usability and privacy as well as narrowing the research down to just a small set of clearly defined use cases. However, as we had little knowledge about the domain, it was hard to scope the project completely up front. We wanted to learn as much as possible, and we had to adaptively reduce the scope as we grew our understanding of the people, technology and processes involved in wireless ECG monitoring.

Personal health information is considered highly sensitive, and health care is also a safety critical environment. Because of these factors combined with the time frame of this research project, we decided not to do any form of clinical experiments or clinical validation of our prototype.

1.5 Contributions

This research project would not have been possible without the information and we have received at the two Norwegian hospitals where we conducted qualitative interviews on both clinicians, clinical engineers and researchers. The implementation of the prototype and experiment would not have been possible had it not been for all the open source software enthusiasts and maintainers doing all the hard work so that we could pick up the pieces and rearrange them. See [17] for an overview of the different code repositories related to this project and the third party libraries used.

1.6 Structure

This thesis is structured as follows: We start Chapter 2 with background on the domain and previous research, through a literature review. In Chapter 3 we present our research strategies and methods for exploring clinical ECG monitoring in a structured and thought through manner. Based on this approach, we spend the next two chapters presenting our results and findings. In Chapter 4 we present our results from researching the clinical context and practices around both wired and wireless ECG monitoring, as well as investigating the monitoring solutions installed at two Norwegian hospitals today. In Chapter 5 assess the technology involved in this project. We propose a design for a modern monitoring solution, and present a prototype intended to validate this design. The rest of the chapter

explains the implementation details of this prototype, before we end with an evaluation of the technology assessed. This is done by both by experiment, literature review and by combining our experiences and observations from the quantitative clinical research. In Chapter 6 we discuss our results, research strategies and choices taken along the way, before Chapter 7 concludes the research.

This page has been intentionally left blank.

2 Literature Review

This section cover the literature review, as well as required background information on Bluetooth and ECG.

2.1 Electrocardiogram

Most of the reviewed literature discuss or mention ECG in some form or another, because of it's high technical requirements. However, we found many inconsistencies and simplifications about ECG, that in worst case may have invalidated parts of previously conducted research. Because of this, we spent some time understanding both the measurement itself and how it is carried . In the following section we introduce ECG before reviewing other research involving the diagnostic tool.

Electrocardiogram is a measurement of the electrical activity around the heart. The measurements are done by a set of external electrodes attached to the skin at certain places around the chest. These electrodes measure the depolarization of electrical charges that happen in the cell membranes covering the heart at every heart beat. An ECG is conducted by having either 3, 5 or 12 leads. The term lead refers to the voltage difference *between a given pair* of electrodes, parallel to the direction of the lead. Based on the signal strength of each lead, cardiologists can get information about angles and properties of the heart. The leads can either be actual or derived from a vector model¹ based on 2 or more actual leads. Typically 5 electrodes can be used for a 3 and 5-lead ECG, where one of the electrodes are used as a reference point.² It is however important to note that the amount of electrodes don't necessary correspond to the number of leads, and the amount of derived leads does not necessary correspond the the number of actual leads. The terminology here seem to confuse a lot of people, as we have seen and heard in both literature and in practice. Alesanco and García are among the few that describe this correctly in [2], where they state that “(...) *only eight leads should be transmitted since the other four can be derived from them*”. The this confusion is not unfounded, as there exist several techniques for deriving different leads from other leads. As we'll see later, a 12-lead EASI ECG may be derived from 5 electrodes capturing only 3 actual leads (or vectors). Even the traditional 12-lead

¹ The Cabrera circle uses a vector representation to give a good overview of the different leads, and their relationship

² Also referred to as *ground*.

ECG, is derived from 10 electrodes³ which capture 9 actual leads. The differences roles these leads play in practice, are discussed further in Section 4. The consequence of this confusion is that previous research either make incorrect assumptions [55] or report ambiguous results that are hard to verify. Some research [87, 46, 2], only consider 1 or 2 leads, which has been proven insufficient by [16].

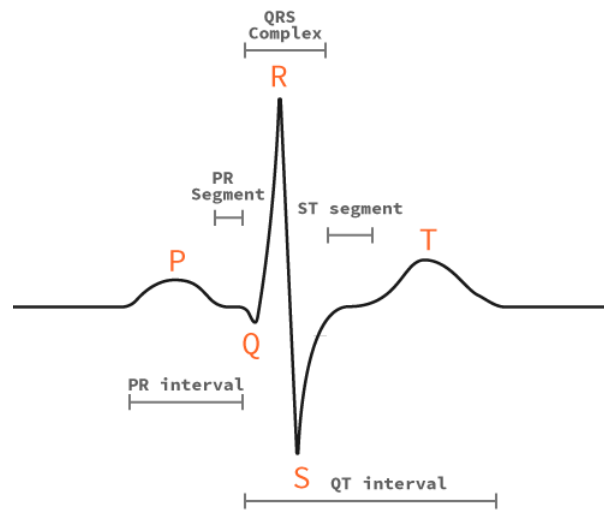


Figure 1: QRS Complex with interval annotations

After the leads are sampled in a wireless ECG solution, they can be streamed in raw, or compressed and optimized for transmission[9, 2]. In order to reduce the scope of this project, we will only focus on the the core aspects of ECG. Different compression and low sampling techniques are therefore omitted completely. After being transmitted, the signals are analyzed at a processing unit, where common angles and intervals in the QRS complex as well as heart rate are calculated. The QRS complex as seen in Figure 1 denotes the plot or graphical representation of the electrical signals. On a per lead basis, these plots are printed on gridded paper or displayed on a monitor. Cardiologists learn to analyze the leads and are able to make a diagnosis based on different intervals in the complex, among other things. An overview of some common intervals and the different cardiac diseases they indicate are presented in Table 1. Based on this analysis, the ECG system may

³ In a 12-lead ECG the electrodes are placed at each extremity (limb) and 6 on the chest.

Table 1: Characteristic ECG intervals from [40]

Interval QRS >120ms	Ventricular hypertrophy, necrosis, BCRRD, BCRI, pacemakers, cardiomyopathies, electrolyte abnormalities.
Interval PQ >200ms	Frist-degree AV block
Interval PR <120ms	Tachycardia, WPW, manners or headphones low rates.
Interval QT >450ms	Antiarrhythmic medicines, ischemic heart deisease, cardiomyopathies, hypocalcemia, mixedema, long QT syndrome
Interval QT <350ms	Hypercalcemia, hyperkalemia, early repolarization, digoxin

automatically trigger alarms if certain thresholds in the intervals are exceeded.

We have reviewed several earlier projects involving wireless ECG. Below we will assess one such project, while others will be discussed in 2.2. In Chou and Hibbs 2006 paper [64] they present “An ultra-wearable, wireless low power ECG monitoring system”. They have developed and tested a contact-less ECG capacitive sensor based on insulated bio-electrodes. They imagine that these sensors can be integrated into clothes, making ECG monitoring ultra-wearable. They recognize that related work based on either standard wireless interfaces or general purpose sensor nodes suffer from large form factor, low battery lifetime or low transmission speed. Because of this, their approach involves using proprietary and custom hardware in order to reach their goals. Aside from the tiny form factor, an interesting observation is their solution’s power consumption, which is less than 10 mA in transmission mode and 22 mA in receiving mode. As we’ll see later, this is 10-20 times the average current drawn from today’s low energy sensors based on wireless standards like Bluetooth Smart. In terms of ECG, their solution supports 1000 Hz sampling rate, and the analog to digital converter (ADC) is configured for 12-bit resolution. They mention their system uses 3 ECG sensors. Assuming they are referring to electrodes, and that one is for *ground*, this means they can only do a 1-lead ECG. Further, they do not address any clinical aspects of their work.

2.2 Bluetooth and ambulatory ECG

In accordance with our research goal of assessing available technology, Bluetooth was an attractive technology. This was because of its widespread presence⁴, as well as the low energy features introduced with version 4.0 of the specification. Bluetooth was standardized as IEEE 802.15.1, but IEEE no longer maintains the standard. The Bluetooth Special Interest Group (SIG) is comprised of more than 25,000 corporate members, which today maintain and oversees the development of the specification. There are two other IEEE 802.15 standards discussed in relevant literature, namely 802.15.4 and 802.15.6. The first one is likely to play a significant role in the development of future IoT applications.⁵ As the second one is still in development, this thesis has not assessed the work on 802.15.6. However, of these standards and their implementations, Bluetooth is the most available today. In this section we will cover the basics of Bluetooth Smart and discuss related research projects using the technology in combination with ECG monitoring.

2.2.1 Bluetooth Smart

Bluetooth originated from Ericsson in 1994, and has its name from a Danish/Norwegian viking king that was characterized as a “good communicator”. Since its origin, the standard has seen 4 major releases, and several minor ones.⁶ The next big release, version 5.0 is currently being being drafted, and is expected to be formally announced later this year.

Version 4.0 introduced new low energy features, which was originally branded with the suffix “LE” or “Low Energy”. Because of confusion around the “low energy” devices and those implementing older versions of the standard, Bluetooth SIG announced their new branding strategy in 2011. The new logo and name read *Bluetooth Smart*. Throughout this thesis we will focus exclusively on the low energy features of the standard and we believe the suffix “low energy” is less ambiguous than “Smart”. Because of this we will be using both the old and new name based on what’s most describing.

⁴ In 2010, Bluetooth SIG estimated an annual sale of 2 billion Bluetooth enabled devices.

⁵ See Google’s Thread protocol and Zigbee for implementations of this standard.

⁶ The latest of which (4.2) was released in December 2014

2.2.2 GATT and Profiles

The Bluetooth standard specifies a series of communication protocols that together constitute the complete Bluetooth stack. This stack is separated in two; the *Host* and *Controller*. The controller stack is concerned with the lower network layers and is usually implemented in the firmware of silicon devices containing a Bluetooth radio. The host stack are upper layers, and is usually implemented as part of a operating system and runs on a processor together with the user application. With resource constrained devices such as those we'll investigate in this thesis, both the controller and host can be bundled together. They will then run on the same microprocessor, making up what is known as a “hostless” system. See Figure 2.

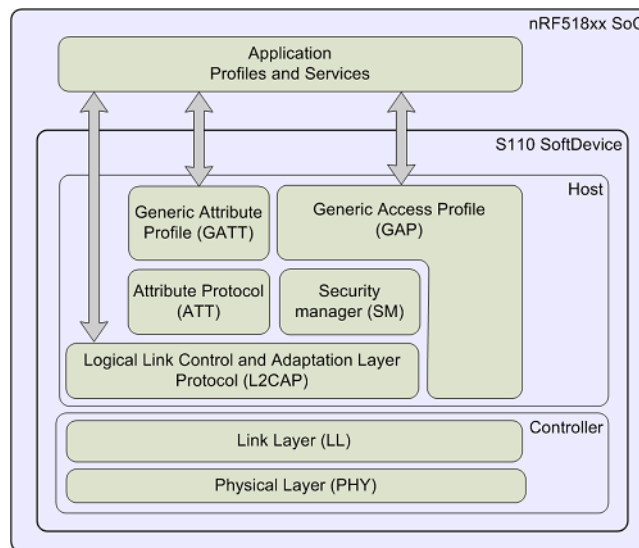


Figure 2: Overview of the Bluetooth protocol stack on a hostess system [72]

In this thesis we are most concerned with the higher layer protocols, more specifically the GATT profiles. This is an acronym for Generic Attribute Profile, and it is responsible for ensuring interoperability in data exchange between two two Bluetooth devices. The profile defines the types of state data that a device exposes and how that data can be used. Profiles are organized in services, which contain one or more characteristics and descriptors. The profiles and its characteristics are specified by working groups within Bluetooth SIG. The process of how these are developed and implemented are discussed in Section 2.4. More information

about how connections are established and data is transmitted, can be found in Section 5.2.2.

2.2.3 Bluetooth enabled ambulatory ECG

We have reviewed multiple previous research projects evaluating Bluetooth Low energy as a wireless technology for streaming ECG data. In this section we will review two of them.

Jara et al. evaluated Bluetooth low energy as a technology for streaming continuous data from a wearable ECG in their 2012 paper. Similarly to our project, they focused on analyzing the capabilities of Bluetooth Low Energy as a medium for streaming raw ECG data. Their initial evaluation concluded that it was necessary to compress the ECG signals before transmission. They built a prototype based on a Bluegiga development kit. This development kit comes with its own Bluetooth API, built on top of the Bluetooth stack, which is programmable with a language called BG-script. They do not mention the amount of leads simulated, and their calculations are hard to follow. For their experiments, they operated with a ECG sampling rate of 300 samples/second. We were unable to recompute their calculated throughput, but they conclude that Bluetooth Low Energy in 2012 is not well suited for raw ECG transmission. However, a lot has happened with the specification since then.

Another paper describing ECG data transmission with Bluetooth Low Energy (BLE) is [65]. They built a three tier of 2 development kits and an Android application. This paper is from 2013, at which time Android did not fully support low energy Bluetooth through their SDK. Because of this, their system is comprised of 3 parts, one low energy board with a Bluetooth interface, one “converter” board, that speaks both BLE and Bluetooth 2.1. This board is responsible for proxying the data from the sensor board through to the Android application via Bluetooth 2.1. At the sensor board, they developed a ECG patch system, which records ECG through by using 4 electrodes. They operate with a maximum data rate of 306 kbit/s over the BLE connection, and they claim ECG has to be sampled with at least 250 samples/s. Finally, of interest to this project, they did some preliminary battery tests, and concluded that the batteries lasted between 9 and 13 hours depending on the surrounding temperature.

Other literature related to ECG and its clinical requirements is discussed further in Section 5.5. One common denominator through the review of these and other research projects, is that no one seems to agree on what’s the required sampling rate for ECG, and the size of each sample should be. Further we re-

cognize that very few, includes the clinical aspects and use cases of doing ECG.

2.3 Wireless Body Area Networks

A lot of work has been done within the field of Wireless Body Area Networks (WBAN). This is a truly multidisciplinary field, that is concerned with everything from antenna design and MAC protocols and smart temperature-based routing schemes [58], to hospital, clinical and medical topics ranging from infrastructure to cardiac arrhythmias. In our case we were not very much interested in the actual networks, as much we were in the problems faced in both the development of WBAN and using low energy sensors for physiological monitoring. Because of this, we have researched a lot of different WBAN problems, in order to find the problems that had already been solved, that might benefit us. In terms of architecture, and the organization of devices outside and around the WBAN, we have investigated several approaches. In the following section some of them will be discussed.

Previous research have proposed categorizing WBAN data in different tiers, based on clinical priority. A real-time ECG stream may be of higher clinical value and have higher QoS requirements than the occasional temperature measurement.⁷

A fundamental question in WBAN how to implement the gateway. Most research uses a gateway of some sort, and different solutions to how to connect a WBAN to external devices and services has been proposed[76], some more controversial than others [88]. Most of the architectures we reviewed involve a personal server, or personal gateway as we refer to throughout this thesis. The role of this gateway and it's capabilities define in many ways how the rest of the infrastructure is going to look like. Shahamabadi et. al. presents a good overview of the most reasonable ways of organizing this in [76]. Here they present a solution to support mobility in . They present 3 different combinations: One where WBAN nodes communicate directly with a boarder router, i.e. without a personal gateway (or mobile router as they call it), another one where all traffic from the WBAN is routed through a personal gateway, and a hybrid of the first two.

The last approach involves using mobile gateway just as a facilitator for “administrative” tasks like coordinating access point handoff and roaming on behalf of the WBAN. This solution require that the WBAN sensors send physiological

⁷ Note that deep Vein Thrombosis, or blood clots, can be detected by sudden temperature changes in extremities.

data directly to a smart access point. This proposition solves the problem of the mobile gateway becoming a bottleneck because data does not have to be routed through the gateway. A smart access point would also have more resources for doing intermediate data-processing [68]. However, there is no silver bullet, as this has approach involve a higher energy consumption: With this organization the node antennas would have to consume more energy in order to send data directly to an access point or border router which would typically be located further away than the mobile gateway.

However there are tradeoffs with all solutions. Making a single gateway the sink of several sensors organized in a WBAN creates a single point of failure. With a single point of failure, ensuring the personal gateway is working properly will become of outmost importance, as a faulty gateway will not only take down the low priority sensor measurements, but also the critical ones. As we'll see in later sections, today's telemetry solutions operate with zero downtime.

2.4 Interoperability

Rahmani et. al. gives a clear overview of different forms of interoperability in their 2015 paper about Smart e-Health Gateways [68]. They separate the concern of interoperability and reconfigurability into the following categories: Device interoperability, protocol interoperability, data interoperability, reconfigurability and device discovery and mobility support. In the following sections we will give an overview of data interoperability related to Bluetooth.

2.4.1 Data Interoperability

Document and message standardization in health care is a huge topic and certain actors like HL7 [36] have been working actively with standardization development for the least 30 years. In this section we will cover WBAN related papers discussing interoperability as well as giving a brief overview of the latest standardization efforts made by actors like HL7, Continua and the Bluetooth SIG.

The Continua Alliance develops interoperability guidelines and offer device manufacturers a way of certifying their Bluetooth Smart devices for Personal Connected Health. Their guidelines are globally recognized as the only standard for Personal Connected Health [3]. The certification program is organized as follows: Continua selects and approves certain GATT Profiles (aka. Smart Profiles). Devices that support these profiles can be certified, meaning that they will be compliant and that the data they deliver can be “mapped into the Continua HL7

Record-set and shared, where it can become available as needed to social media, care providers, hospitals, clinicians, etc.” [3].

How are Bluetooth Smart profiles implemented? Efforts are being made by Bluetooth SIG to standardize data profiles in order to ensure interoperability between devices supporting the protocol. A profile for a type of device defines the GATT Services which a device of this type must or may implement. The GATT profiles are specified by the Bluetooth SIG through Device Data Specifications. This work is organized in different domains and there is a specific sub-group for Health Device Profiles (HDP) (current version 1.1), some of whom are approved by Continua. HDP defines devices as either sinks or sources. In this thesis these roles are referred to as nodes and personal mobile gateways.

The HDP specification does not specify the format nor content transmitted, but confine to ISO/IEEE 11073-20601 Personal Health Data Exchange Protocol [38]. Bluetooth SIG require devices implementing the HDP to follow the ISO standard for exchanging data between HDP devices and IEE 11073-104xx Device Specification. However, from ISO/IEEE 11073-20601 Data Exchange Specification, ISO/IEEE draws a clear line between consumer electronics and medical devices:

“Provides strong application level interoperability by operating with the ISO/IEEE 11073-20601 Personal Health Data Exchange Protocol [7], which defines a transport-agnostic Data Exchange Protocol and representation of device application data based on international standards. This standard defines a common core of communication functionality for personal telehealth basic ECG (1- to 3-lead ECG) devices. Monitoring ECG devices are distinguished from diagnostic ECG equipment with respect to including support for wearable ECG devices, limiting the number of leads supported by the equipment to three, and not requiring the capability of annotating or analyzing the detected electrical activity to determine known cardiac phenomena. This standard is consistent with the base framework and allows multi-function implementations by following multiple device specializations (e.g., ECG and respiration rate).” [38]

This page has been intentionally left blank.

3 Research Strategies

The most critical success factors in every research project lies in the researcher's ability to plan, execute and monitor the study. A clear research strategy needs to be clearly defined, revised and followed. This chapter will lay out the research strategy applied in this project, and discuss how we approached the different research stages.

3.1 Research Methods

Wireless monitoring of physiological metrics is a wide topic covering many similar or related instances of the same problem. In order to reduce the scope we decided to look at only one instance, namely wireless ECG monitoring. Because of this, the project in it self can be considered a case study in ECG. It was important to us that we investigated ECG in depth, as well as studying the practice in it's natural, clinical setting. This is reflected in our data collection methods, which include multiple sources and methods and has a focus on relationships and processes.

Like in any project involving people and technology there is an inherent complexity that the researchers have to unravel in order to understand the forces that guide the socio-technical systems in front of them. We want to avoid the deterministic approach of attributing properties to technology that guarantee the effects and outcome it will have on health care. For this research project it was clear that we needed to approach the practical clinical aspects in addition to the technical aspects of ECG monitoring. Different stakeholders may have different views and knowledge about various aspects of the challenges with today's monitoring systems. In order to achieve a deeper understanding of the context and difficulties, we decided to approach both medical professionals and technical engineers. Because of this, we differentiate our data collection methods in those applied when researching the technical side of these systems, and those applied when exploring the medical domain and current practices.

Clinical: In order to understand how and why ECG monitoring is practiced, we applied mainly qualitative techniques. This was done through observations and interviews with medical professionals, and will be discussed further in Section 3.2 and Chapter 4.

Technical: Our strategy for researching the technical requirements to clinical

ECG monitoring systems involved understanding both today's existing technical solutions and those proposed in literature. Because this a wide field spanning different domains, several methods for data collection have been utilized. These include interviews, document and literature analysis, and experiments, as discussed further in Section 3.3 and Chapter 5.

When studying these systems it is important to understand why and how an prototype works in it's environment, not only that it works. Our overall approach have been influenced by both the Design Science methodology, the research team's personal experiences and IBM's framework for Design Thinking [28]. The latter embrace a iterative strategy for understanding peoples contextual needs and deliver outcomes. This neatly ties together our study of clinical practices with the technical research:



Figure 3: From IBMs Design Thinking framework

3.2 Interviews and observations

This section will describe interviews, how they were conducted and the inherent strengths and weaknesses of doing interviews.

In order assess the validity of our observations and get multiple sources, we decided to conduct interviews at two different Norwegian hospitals. This way we could either record the differences in ECG monitoring practices or confirm that they were the same. Norwegian hospitals are usually organized with an in-house department for technical engineers, which made it easier for us to organize the interviews with different stakeholders.

A lot of knowledge is embedded in the people and practices of their every day routines. An introductory conversation at the first hospital revealed that nurses at both the intensive care units and the cardiac observation unit had most practical experience with the technology. In these units they handle ECG and telemetry solutions on a daily basis. After this quick survey we constructed both unstruc-

tured and semi-structured interviews for both hospitals with a focus on open questions. This way we were able to develop a understanding of how people viewed their work and how they interpreted the technology that was part of their everyday routines. This also made the research more flexible, as we were able to do interviews with nurses in their natural working environment on a short notice.

We also arranged for two observation sessions at the second hospital, in order to see for ourself how different aspects of patient monitoring were carried out. In the first session we observed a nurse preparing and conducting a full 12-lead ECG test on a colleague, and in the second session we observed another nurse monitoring 10 remote telemetry patients, and 6 bedside patients in a intensive care unit. During both observations, we got the chance to ask follow-up questions, which further increased our understanding.

Although nurses have most practical experience with the monitoring practices, they do not assess the ECG plots diagnostically. Because of this we also conducted two semi-structured interviews, with a cardiovascular surgeon and an anesthesiologist. These interviews were conducted later in the project and provided clarifications, useful diagnostic insight as well as domain knowledge.

Because this is a highly specialized field, most knowledge is embedded in the workplace. This also holds for the technical knowledge. In order to learn about the technical solutions backing the ECG monitoring we reached out to the clinical engineers at both hospitals. Conducting structured interviews was inexpedient in the beginning, and we began with unstructured interviews at the first hospital. After we had established a basic understanding of the components and solutions, we moved on to conduct semi-structured interviews at the second hospital. This enabled comparison of the technical solutions between the two hospital.

As mentioned in Chapter 2, wireless body area networks and sensor technology are active research areas. In order to gain valuable insight in this development we conducted another unstructured interview with a researcher working with short-range wireless monitoring in critical care at The Intervention Centre at Oslo University Hospital.

All in all we conducted 8 different interview sessions, talking to a total of 10 people. This included 4 clinical engineers, 3 nurses, 2 physicians as well as the clinical researcher from The Intervention Centre. During all interviews we took note of people's experience with, interpretations of, and reactions to the technical monitoring systems and daily routines. We observed what people said and did, and afterwards reflected over what they did not say or do. This way we were able to search for latent needs in the existing practices. The knowledge gained from these interviews are treated in Section 4.

3.3 Evaluation

In order to answer part of our problem statement and main research questions, we wanted to build a working prototype based on available technology and open standards. We intended to use this process strategically in order to investigate the technical requirements to some critical aspects of wireless ECG monitoring. Our findings from this process has been evaluated and is discussed in Section 5.5. This evaluation consist of both a technical experiment conducted with the prototype, and surveys of technical documents and previous literature.

3.4 Document and literature analysis

A lot of work has been done within the field of WBANs and telemetry, and a systematic approach to gathering, analyzing and extracting knowledge from previous work was an absolute necessity. This section will describe how we collected literature and documents from previous research and existing solutions to investigating the topic of WBANs.

Searching for information was a maturing process. Finding the right search terms did take some time, but we found the most relevant results on Google Scholar using a combination of keywords such as “*WBAN, WSN, patient monitoring, BLE, Bluetooth Smart, ambulatory patient monitoring, ECG, Ubiquitous Healthcare*”.

Documents were collected from two sources: The Norwegian Directorate for e-Health supplied us with standardization documents and guidelines from Continua Health Alliance. These are documents describing interoperability aspects such as how data should be exchanged between sensors, gateways and end services. We received in total 8 documents in the H.810-813 series, dated 2014. One of the clinical engineers also supplied us with installation and service manuals to the telemetry system they had installed. This were 2 documents and one presentation dated 2007, describing the Philips IntelliVue telemetry system in detail.

4 Clinical Context

This section describes our findings from the interview sessions. These findings form the basis of our use cases as well as some functional requirements. We will discuss both similarities and differences found in clinical practices as well as describing the patient monitoring solutions deployed today. Throughout this section we will refer to the three most important stakeholders, which is listed below:

Patients The patients themselves are the main source of physiological data. The patients are the ones in direct contact with the sensory equipment, and will act as the wearer of the WBANs. It is important to remember that these are people admitted to the hospital with a broad span of medical conditions. They might be in pain, unconscious, or mentally exhausted. One must take these considerations into the design and requirement specification.

Practitioner We define health practitioners and clinicians are the medical staff that are responsible for care and treatment of patients, as well as operating their respective wards, departments or units. They are the ones that will install the sensors on the patient, and monitor the signal as they arrive at the central.

Engineer Clinical engineer is defined as the medical and biomedical maintenance engineer responsible for installing and maintaining the technical equipment used for medical purposes at hospitals. Among other things, they are responsible for doing technical maintenance on the central heart monitoring system, which is as close as we come to an established way of monitoring vital signs wirelessly of multiple patients in hospitals today.

The rest of this chapter will be organized as follows: First we will discuss cardiac patients, the practice of patient monitoring and the intended use for our prototype. Then we will be taking a look at the existing solutions deployed at two Norwegian hospitals today, before ending this chapter with a detailed overview of different use cases.

4.1 Intended use

The researcher from The Intervention Centre made it clear that we had to consider the intended use for our prototype. Monitoring ambulatory patients is merely an action following an intention, not the goal in itself. Therefore we must ask, for

what patients are we developing this solution? What use-cases? What is the intended use?

Before we answer these questions, a little introduction to cardiac patients and patient monitoring in practice is needed. Most patients in a hospital today are not electronically monitored on a daily basis. There are however certain units in a hospital that monitor patients continuously. In the Intensive Care Unit (ICU) as well as the Postoperative Unit (PO) all patients are monitored (ECG, SpO₂ etc.) around the clock regardless of their diagnosis. The same accounts for Heart failure Units (HFU) and Intermediate Care Units (IMC) which are wards for patients with cardiac conditions and intravenous medications that require continuous cardiac monitoring and treatment. Depending on the hospital it may have one, two or all four units. All these units usually holds patients temporary before and after a surgical procedure or intervention - here meaning catheter based non-surgical procedures. When asking the cardiovascular surgeon about common heart disease diagnoses at these units, he said the following:

“Acute Coronary Syndrome (ACS) is a group of conditions related to a decreased blood flow in the arteries to the heart, also known as the coronary arteries, resulting a “myocardial ischemia”. This cause a shortage og oxygen and glucose needed for cellular metabolism and makes the heart muscle unable to contract properly and in worst case it will die. This is called a myocardial infarction, better known as a “heart attack”. These coronary artery diseases are the most common cause of death globally. Typically for all ACSs is that they are often diagnosed by ECG. Patient having an ACS needs a continuous monitoring before, during and after therapy, sometimes for days and sometimes even for weeks. Patients at the IMC, or those transferred to normal wards and to rehabilitation may still suffer from different consequences after a myocardial infarction like cardiac arrhythmias and even new cardiac ischemia.”

This is the reason why patients at risk are fitted with a telemetry in order to be monitored. The clinician(s) at a monitoring central observes the heart rhythm and looks for signs of unusual elevations or interval variations in the QRST complex. See Table 1. Abnormalities above or below a given threshold often generate alarm triggering events. The thresholds comes with a default setting and is customizable per patient in the monitoring central.

There is typically one such monitoring central at a given hospital. But where it is located varies. At the first hospital we visited, the central monitoring capability

was located in a dedicated HFU. The second hospital didn't have a dedicated HFU, and the central monitoring station was located at the intensive care unit.s

Based on the observations above we define the intended use for this system as follows:

Intended use: To remotely monitor not critically ill patients at risk for complications regarding an acute coronary syndrome (susceptible for heart defects), located at an IMC or in another ward.

In other words, the intended use is on patients that would have been equipped with a telemetry device today. The existing systems are used for monitoring patients that are expected to experience, or are vulnerable to, heart defects either pre or post surgery, or during treatment. See table Table 1 for a description of the most common irregularities and what identifies them. Because irregularities can happen at any time of the day and night, continuous monitoring is a fundamental requirement to the prototype.

4.2 Existing Solutions

Interviews with medical engineers at both hospitals revealed that both had the Philips IntelliVue telemetry system installed. In both cases the network infrastructure and equipment was produced by Philips, but was delivered and maintained by different local partners. This section will describe the IntelliVue telemetry system from Philips, which as we found out, is deployed at several Norwegian hospitals today. The studying these systems were conducted in order to gain insight in the practice of wireless patient monitoring today, as well as giving us an indication of how prepared hospitals are for WBANs from a infrastructural point of view.

4.2.1 System Architecture

The Philips IntelliVue telemetry system is a complete end-to-end system enabling both wired and wireless monitoring of ECG. For the medical practitioners the system is composed of two components: the TRx4841/51A wireless transceivers and the IntelliVue Information Centre. The TRx transceivers are devices for capturing ECG and SpO₂ on adult and pediatric patients, and are worn by the patients. Powered by two AA batteries they capture and stream a 3 or 5 lead ECG to a custom 802.11 type access point (AP) operating in the 2,4 GHz spectrum. These APs are installed in addition to, and independent of existing public or private

WiFi access points. Because Philips provide all hardware in this closed system, the telemetry system does not have to confine to the interoperability requirements it would need if the system were to allow for proprietary sensors and measuring devices. The wireless interface is based on the 802.11 specification, and the system offers a “Smart Hopping” technology in order to reduce collisions on the already crowded 2,4 GHz network band. This is made possible by synchronizing all access points through a dedicated sync unit. Another noteworthy network feature is the “make before break” method that improves the roaming abilities. The TRx transceivers will look for new, stronger access points when they detect a decay in signal strength with the current AP. The system will make sure to connect transceivers to the new (stronger) AP before breaking with the previous one, enabling a more reliable handoff between access points. The network coverage at the second hospital was almost 100%, while the first, larger hospital offered 100% network coverage in one of its buildings.

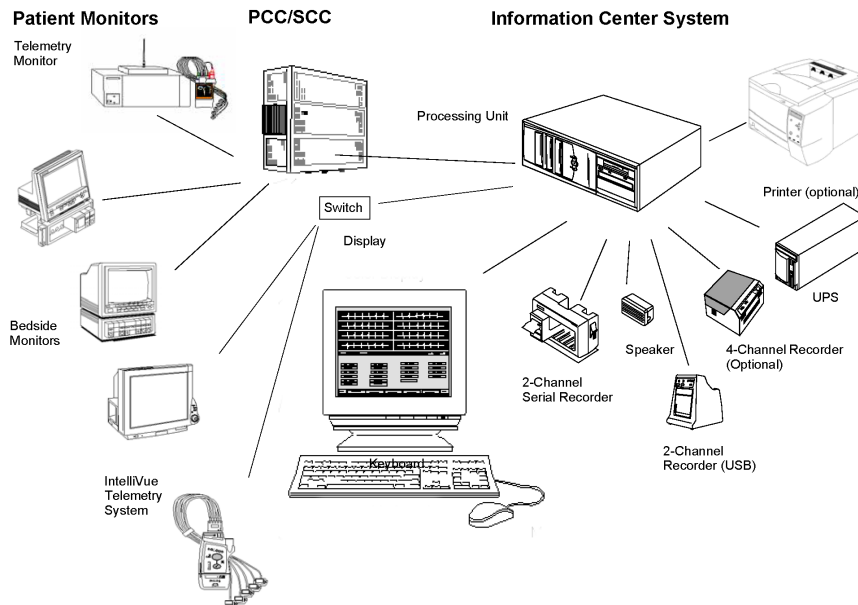


Figure 4: IntelliVue Monitoring System Components [62]

As mentioned, this being an end-to-end system the IntelliVue patient monitoring suite includes everything from transceivers, bedside monitors, access points, synchronization units, gateways, network switches, uninterruptible power supplies (UPS), processing units and databases in addition to the optional printer Figure 4.

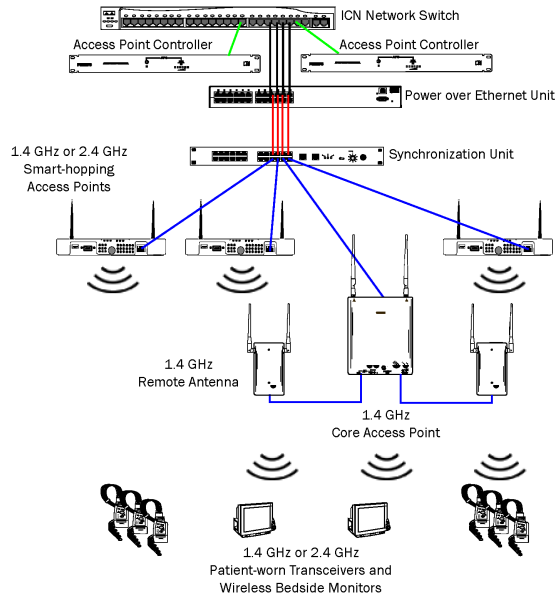


Figure 5: IntelliVue Network Topology [62]

Offering every piece in this architecture enables Philips and its local distributors and providers to guarantee good QoS metrics which is essential for wireless patient monitoring applications. However, installing a complete system like this comes with a substantial economic cost, and ties you down to just one provider for an extended period of time. The IntelliVue Telemetry system at the first hospital was installed in 2009, and is expected to have a lifespan of at least 10 years. This reduces the flexibility and ability to adapt new solutions. On the upside, an all-in-one solution like can easier guarantee QoS as well as offering better support and maintenance. It's easy to think that this also might reduce the need for interoperability - because Philips controls the whole stack. Having studied the solution being used in practice, we would argue this is false. An example of this can be found in the installation of the CorPulse ECG device from Alere [6]. This is a device installed in emergency vehicles enabling a short, pre-hospital ECG to be sent over 3G network with a SIM card. This is often critical information that enable the medical staff to make informed decisions before the patient arrives at the hospital. Because this is a different system from a different provider, the hospitals we visited had installed a standalone computer next to the central monitoring information center system from Philips. This standalone computer display the pre-hospital ECG from the Alere device. Although coming from a different

source, the users we talked to experienced this as “remote ECG”, just as the they experienced the central telemetry solution next to it as “remote ECG”. This is suboptimal not only in technical terms and with regards to maintenance, but also from a user experience point of view. The lack of integrations creates yet another technical dependency in the everyday life of medical practitioners. A workaround for dealing with separate systems is sharing login credentials, which render the security measure completely useless. We observed this at one of the hospitals. Based on these observations, we believe interoperable devices and services will be crucial for the success of a ubiquitous, sensor-based health care.

4.2.2 The Transceivers

The TRx transceivers are small and lightweight with a limited user interface consisting of a couple of lights and a audible alarm. In practice we were told they have a battery lifetime a little over 24 hours, which fits well with the specifications found in the user manual (up to 48 hours). As we’ll see in Section 5.2.2, the average throughput is tied to sampling rate. In order to establish a standard for what throughput wireless ECG must support we reviewed both literature and the user manuals for the IntelliVue Telemetry system. As the numbers used in literature varies a lot, we searched the user manuals for this information. However, we were unable to find any information about neither the sampling rate nor average throughput the transceivers delivered. Both Philips and the local providers of the system were consulted, but no one could give us a clear answer. When this failed, attempts were made to find the information based on network traffic, but we never got hold of the telemetry network administrator.

In terms resolution, Philips Medical Systems have developed the “EASI” lead system with the ability to estimate a 12 lead ECG. This is in line with the results from [86] Wehr et. al. which showed that the 5-electrode EASI was equivalent to a 12-lead ECG for the diagnosis of myocardial ischemia. Although the transceivers provide ambulatory and bedside patient monitoring, there’s a catch with increasing the mobility: Usually when doing a detailed 12-lead ECG, medical practitioners place the 4 extremity electrodes at the wrists and ankles. In a 5-lead telemetry these electrodes are often moved to shoulders, hips and chest in order



Figure 6: TRx transceiver from Philips

to increase mobility. However, because of the large muscle mass located in these areas, the recorded ECG signal is prone to more muscle noise. We observed this ourselves while studying a distorted 5-lead telemetry reading at one of the hospitals. We were told this probably was a result of the patient moving. Accurate, unobtrusive and noise resilient sensors like the ones in [64, 27] will be an absolute necessity in persuasive patient monitoring.

If there should be extended periods of noise or the clinicians are suspicious with the readings on the screen, they have the ability to do an independent ECG. Both ICU and the unit with the remote patients had several standalone 12-lead ECG units that could be connected to the patient on request. We observed how a patient could be connected to the standalone ECG device with 10 electrodes in order to get the full 12-lead resolution. This device was mobile (on wheels) and had the form factor of a small laptop computer. It was standalone in the sense that it's only output was a printed time interval of 6 seconds from the examination as shown in Figure 7 from a built in printer. We also visited two other medical units at the same hospital that did some ECG monitoring. Both of which used a 3-lead setup connected directly to a bedside monitor. This stands to show that at the second hospital, 3 and 5-lead setups are most used in continuous monitoring. The full 12-lead was only used in short examinations on demand. This practice was also confirmed at the Clinic of Cardiology at the first hospital.

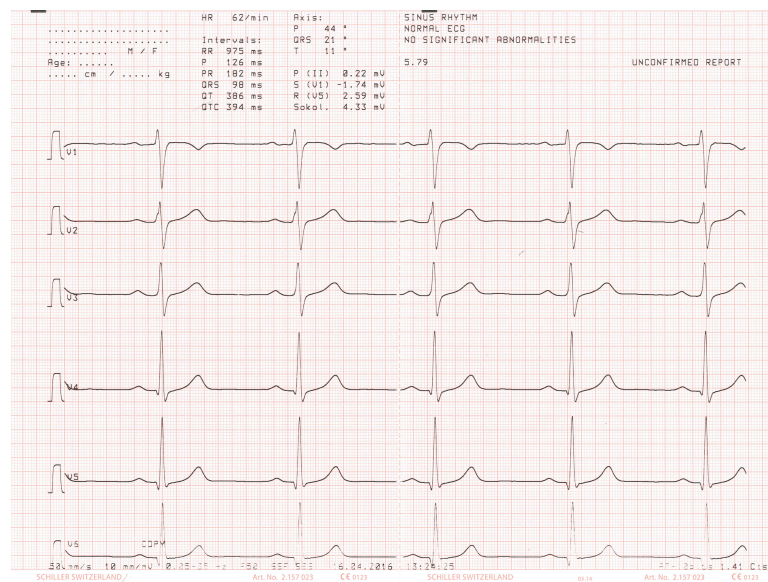


Figure 7: Annotated ECG plot of lead V1-V6. See Appendix D for all 12-leads.

4.2.3 The Information Centre

The Philips IntelliVue Information Center constitute the the central monitoring post. Here clinicians has the ability to view multiple real-time patient leads at once, see detailed view on a given patient, and adjust alarms and thresholds. The IntelliVue Telemetry System supports up to 128 2,4Ghz transceivers.

To this project, the most interesting aspects of the monitoring central was the end-to-end latency between the electrode on the patient to the central. We were unable to find an accurate metric for the end-to-end latency in IntelliVue's documentation. Information on transmission delay is also scarce in the literature we reviewed. Within the field of WBAN very few address this issue, and and we had to look into Telemedicine in order to find any suggestions to end-to-end latency at all. Through clinical trails, Alesanco and Garcia suggest 3-4 seconds as the maximum acceptable delay for a reliable real-time ECG transmission [2]. Their research however is concerned with telemedicine over 2G/3G networks, where the intended use and QoS are different. Because Philips have full control over the whole infrastructure and the distances at a hospital are relatively small, it is reasonable to assume the delay from the measurement to the information center is less than 2 seconds. An alternative approach to getting information about acceptable transmission delay would be to measure the actual network traffic in the systems deployed today. But as mentioned, we were unsuccessful in getting in touch with the network administrator at either hospitals.

As Alesanco and Garcia's requirement was the only exact metric we could relate to, we chose to use this as our baseline for network delay.

4.3 Use Cases

Having established the intended use, we will use this section to describe different use cases. We believe this has to be included in order to get an understanding of the context the solution have to support. Keep in mind that these use cases only assess some parts of the system, and cover more practical features rather than technical.

Patient

U1: **Stationary:** This use case will describe the situation where a patients monitored while lying in bed. Here both gateway and sensor would be stationary.

A patient is being monitored wirelessly while lying in bed.

U2: **Partly ambulatory:** Here patients are monitored while walking around their bed. This means the patient is ambulatory while the gateway is stationary.

U3: **Ambulatory:** This use case describe the situation where patients are monitored while walking around the room or ward, with the personal gateway in their pocket. This means the patient is fully ambulatory.

Medical Professional

U4: Setting up the patients personal gateway

U5: Monitor a patients ECG on a separate device

Table 2: Use case: Stationary

ID	U1
Name	Stationary
Goal	Store ECG data using a stationary sensor and a stationary gateway.
Actors	Patient
Start requirements	1. Mobile gateway is connected to the sensor 2. Mobile gateway is connected to the local WiFi.
End requirements	Patient ECG stream is transmitted to the monitoring station and the data is stored.
Main flow	The gateway is activated and data gets routed from sensor to network.
Alternative flow	None
Parent use case	None
Child use case	U2

Table 3: Use case: Partly Ambulatory

ID	U2
Name	Partly Ambulatory
Goal	Store ECG data while having a moving sensor and a stationary gateway.
Actors	Patient
Start requirements	<ol style="list-style-type: none"> 1. Mobile gateway is connected to the sensor 2. Mobile gateway is connected to the local WiFi. 3. The gateway is activated and data gets routed from sensor to network.
End requirements	Patients are able to move freely around their beds while the gateway is stationary on the nightstand.
Main flow	Patients are able to get out of bed and move walk around undisturbed.
Alternative flow	None
Parent use case	U1
Child use case	U2

Table 4: Use case: Ambulatory

ID	U3
Name	Ambulatory
Goal	Store ECG data having a both a moving sensor and a moving gateway.
Actors	Patient
Start requirements	<ol style="list-style-type: none"> 1. Mobile gateway is connected to the sensor 2. Mobile gateway is connected to the local WiFi. 3. The gateway is activated and data gets routed from sensor to network.
End requirements	<ol style="list-style-type: none"> 1. Patens can move freely outside their room. 2. The gateway automatically connects to new access points.
Main flow	Patients pick up the mobile gateway and walks out of the room.
Alternative flow	None
Parent use case	U2
Child use case	None

Table 5: Use case: Set up gateway

ID	U4
Name	Set up gateway
Goal	The gateway is configured and ready to proxy sensor node data to endpoint
Actors	Clinician
Start requirements	<ol style="list-style-type: none"> 1. Sensor node and gateway have sufficient battery. 2. The clinician is authenticated and authorized to configure the gateway application. 3. Monitoring server is running.
End requirements	The gateway is ready to stream data from the sensor node to the endpoint.
Main flow	<ol style="list-style-type: none"> 1. The clinician opens the application and click “new”. 2. The clinician click scan for sensor and select the correct sensor. 3. The clinician selects an endpoint from the list. 4. Clicks “activate” which test and activates the gateway routing.
Alternative flow	The gateway is not able to connect to either sensor or endpoint. Error message is shown.
Parent use case	None
Child use case	U1,U2,U3

Table 6: Use case: Monitor ECG

ID	U5
Name	Monitor ECG
Goal	The clinician is able to monitor a ECG plot on a separate device.
Actors	Clinician
Start requirements	The clinician is authorized to monitor ECG.
End requirements	The ECG stream is available at the clinicians device
Main flow	<ol style="list-style-type: none">1. A clinician navigates to the correct URL on a web enabled device.2. The clinician selects the correct patient and sees a live ECG plot in her web browser.
Alternative flow	The clinician is not authorized and do not get access to the ECG stream.
Parent use case	U4 and U1 or U2 or U3.
Child use case	None

This page has been intentionally left blank.

5 Technology Assessment

According to the four types of prototypes in design science [41], our artifact was intended to be of the model type. A model can be used for supporting the construction of other prototypes, and hopefully our model can be reused by later projects. We wanted to come up with a design and then create a prototype that let us test different aspects of using low energy sensors for wireless ECG monitoring. Based on what we knew from previous literature we decided the following three aspects constituted the baseline functional tests that would answer our research questions:

- End-to-end latency
- Battery lifetime
- Maximum throughput

We validate our proposed design through a working prototype. Our prototype takes the form of a test bed that can be used to address the items listed above experimentally. This section starts with a discussion on different technical and architectural alternatives, and the decisions that went into it. Later we describe how the base of the testbed was implemented, before we customized it in order to evaluate the baseline functional tests.

5.1 System architecture

As mentioned in Section 2, different architectures have been proposed and discussed in large by previous research. A general solution to how WBANs should be organized in relation to existing infrastructure does not exist, and is an important research topic. Mainly two different architectures have been discussed earlier: One where the WBAN routes all traffic through a personal gateway, and another one where it directs the traffic directly to an access point. See Figure 8. In [76] a third approach is proposed. This merges the two former ones, where WBAN data is sent directly to the access point, while the personal gateway is given the responsibility of coordinating mobility issues like exchanging handover messages on behalf of the WBAN. In the following section, we will compare the first two approaches as seen in Figure 8.

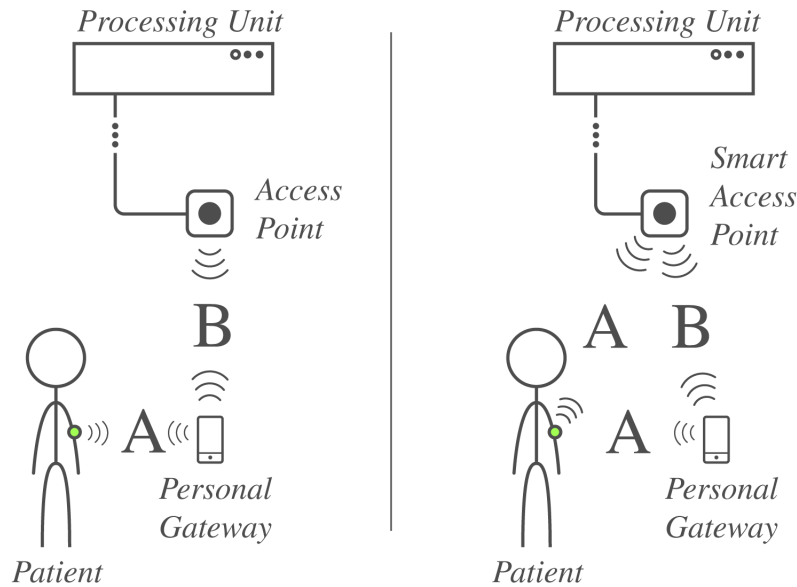


Figure 8: Two alternative architectures. Here A represents Bluetooth Smart while B is 802.11.

Based on what we know about the existing telemetry systems deployed today and the research on WBAN sensor devices, it is improbable that all sensory devices will be (A) developed by the same manufacturer, and (B) that they all communicate over the same physical medium. The question of data interoperability is an interesting topic, because of the technical constraints imposed by the small sensors. At what point in the pipeline do we enforce the standardized data formats for exchanging medical information? In general, we believe this should be done as early as possible in the process, preferably at the node level. However, we want to highlight two arguments to why this might not be optimal: Although there has been invested a lot of effort by standardization organizations like HL7 and OpenEHR the last 30 years, data standardization in health care remains a challenge – not only across institutions, but also between information systems located at the same hospital. Based on this, requiring standardized formats with reduced footprint, optimized for constrained sensory devices, might not be realistic. As mentioned in Sect. 3, organizations like Continua are currently trying to bridge this gap, but with a primary focus on personal health technology, not clinical. This is a topic that needs more research.

Another argument to why it might not be feasible to enforce today's standardized data formats for physiological measurements at node level, is related to the technical constraints of these devices. In order to keep the energy consumption low, overhead is reduced and as little data as possible is sent. This is especially true for continuous, streaming data sources like ECG. The approach Continua and ISO/IEEE has to this problem is to make smaller standardized measurement data formats tailored for constrained devices. Then they specify how these data formats should be transformed or mapped onto richer, data formats with a larger space footprint at a gateway level.

Relating this to the architectures discussed in previous research, we identify certain practical advantages of routing all traffic through the personal gateway: Making a personal gateway compatible with a given sensor and its manufacturer specific or protocol-specific data format is a trivial task through software applications that can be installed and extended on the personal gateway device. The same triviality does not hold for a smart access point or border router wanting to transform the data into a richer standardized data format. In a reality where a standardized data format is not implemented by every sensor node, giving an access point the role as a WBAN sink, as suggested by [68] seems highly impractical: All access points in a roaming environment would have to be compatible with custom data formats from several WBAN sensors carried by a multitude of patients. In this scenario, we only see smart gateways or access points being able to fulfill this data processing and transformation responsibility if standardized data formats for every physiological measurement is implemented at WBAN node level.

Based on these considerations, our proposed architecture consists of the following parts: a single sensor node simulating a patient ECG device, a personal mobile gateway acting as a WBAN sink to be carried by the patient, a monitoring server available on a local or remote network together with a client interface for remote monitoring. In order to reduce the scope and complexity we have decided to only include one node in our prototype. This is a clear limitation with our prototype and study, as we only consider a single-hop star network topology instead of a mesh topology, which is proven to be the most effective topology for WBANs. One concern here might be that Bluetooth does not currently support mesh networks, but as of Feb. 24 2015 the Bluetooth SIG formally announced formation of the Bluetooth Smart Mesh Working Group [24], and we operate under the assumption that this will be supported in future versions. The gateway should have the responsibility of routing the sampled data from the WBAN to an external endpoint. In accordance with Continua's guidelines, we also propose that the gateway enforce interoperability by transforming the data into a standardized medical

format. But how can this be accomplished in a multi-sensor, multi-manufacturer environment with sensors operating with different wireless technologies?

One possible solution might be to use the repository architectural pattern. The gateway application could be extendable with different “profiles” to accommodate for both different manufacturers, different communication protocols, and different data formats. These profiles would be installed from a repository, central to a hospital, region or country depending on the supported data formats. The profiles would contain a mapping between a given sensor peripheral and the preferred output data format compliant with the monitoring system a given hospital was using. This way, the hospital could support a growing amount of sensor peripherals, independent of sensor manufacturer. Below is a formal description of each component of our prototype. See Figure 9 for an overview.

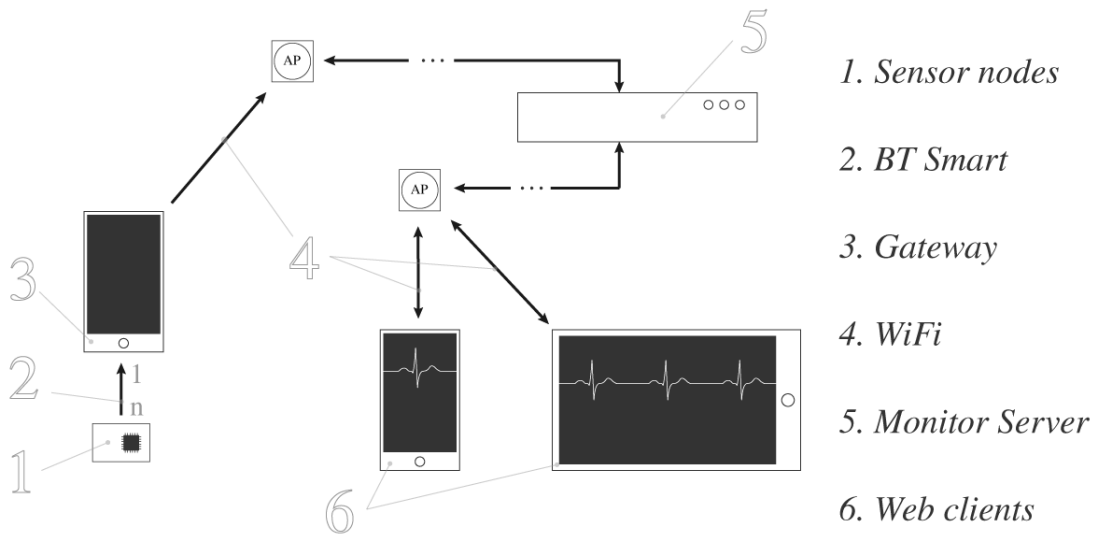


Figure 9: Overview of system and communication protocols.
 AP indicates an access point.

An overview of the different prototype components:

Node: A node is the sensory device that does one or more physiological measurements and communicates it wirelessly to either another node or to a sink in the WBAN. In order to achieve wearability the nodes has to be

as small as possible. Batteries are often the largest part of today's sensory nodes and the biggest energy consumer is typically the antenna [83].

Personal gateway: In previous literature this tier in the patient monitoring stack is also called a Personal Server (PS) or a WBAN sink. This can be a touch device with a graphical user interface, a tele health station, or just a dedicated sink in the WBAN with a larger battery. Because of the low energy requirements in the nodes, there is a need to facilitate communication between the WBAN and a border router with some device that has more battery capacity, a stronger antenna and that is easier to replace on a frequent basis. For the remainder of this thesis we will assume the personal gateway is a feature-rich touch device, with a graphical user interface like the Samsung Galaxy S6.

HIS: Hospital information system. This is a common term for the various types of information systems at hospitals. The way these are implemented and used varies from across borders, regions, and even between different departments within the same hospital. Because of this invariance in systems and practice, interoperability across communication protocols and data formats is important. For the remainder of this thesis we will discuss only one HIS, namely the monitoring central. It is responsible for collecting, processing, storing and serving sensor data to clients.

5.2 Sensor Node

While deciding on what platform to use for the prototype of our proposed design, we looked through a wide array of different System on a Chip (SoC) solutions and boards enabling rapid prototyping and testing with Bluetooth Low Energy. In order to increase reproducibility and ease later projects, an overview of the different development kits we evaluated is included in Appendix A.

Most of the relevant development kits we evaluated were based on one of the Nordic Semiconductor nRF51-series chips [13, 1]. Because hardware design and development was uncharted territory for the researchers, the availability of online communities and documentation was a critical factor when deciding on what development kit to base our prototype on. Nordic Semiconductor has a thriving community on their Stack Overflow inspired developer zone [75], which has at the time of writing has over 13,800 questions asked. Because of this we decided a development kit based on the nRF51822 chip was our best alternative. Because Nordic Semiconductor also offer their own development kits, we decided

to use their boards as the basis of our prototype, reducing the number of external dependencies. In terms of software development on these different boards, we investigated the different IoT operating systems and assessed the feasibility of using these for our purpose. After a quick survey, the small RiotOS [8], running on only 1.5kB RAM and 5kB of ROM, looked like the most promising alternative. A comparison of the major operating systems for IoT-applications can be found at [8]. For our project however, we concluded that the node functionality would be limited and specialized rather than general. Hence, adding a dedicated operating system to our already growing technology stack would impose more risk to our project. It should also be mentioned that this technology and the platforms that are being built around them are not very mature: The seeds for RiotOS were planted less than 10 years ago as it started out as an operating system for wireless sensor networks in 2008 [70]. An example of this lack of maturity, is that we have yet to find a RiotOS library for accessing the Bluetooth stack on nRF51822 through SoftDevices.

Due to the increasing risk and complexity more dependencies add to the development, we dropped support for 6LoWPAN, although this is an important advance for network interoperability in low energy devices. There was an interest for this when we started the project, but a concurrent research project investigating the practical limitations of using IPv6 enabled Bluetooth Smart (6LoWPAN) technology, was already in progress at the Department of Telematics.

5.2.1 Implementation

The first prototype iteration used a Nordic nRF51 evaluation kit from 2013, which supported Bluetooth Smart version 4.0 through a software stack Nordic calls SoftDevices. A SoftDevice is a precompiled and linked binary that implements both the controller and host part of the Bluetooth protocol stack for NRF51 boards. Because of the board's relative old age, the latest Nordic Semiconductor SDK⁸ support for this board was version 6, while the most recent SDK is at version 10. This turned out to cause a lot of compatibility problems, and the development process was tedious: We had to flash both SoftDevice and our own compiled software onto the board manually using Segger's J-Link software. In addition to this, the J-Link debugging options were not supported on our development system.

Because of the compatibility and legacy issues we had with the node in our initial prototype, we decided to replace the 2013 evaluation kit for a nRF51 Devel-

⁸ The nRF5 SDK forms the foundation of what you need of drivers, libraries SoftDevices and code examples to develop your own low-energy Bluetooth, ANT product

opment Kit originally released late 2014 [73]. This board supports a newer SDK (version 8), two new SoftDevices and ARM's mBED development platform [4]. The latter was crucial as it drastically eased the process of compiling and installing software on the development board. mBED is an online development platform for embedded devices developed and maintained by Arm [5] along with partners and contributors. It solves the problem of setting up an correct environment for compiling C++ code specific to a given development board, by building and compiling programs in the cloud. This means they maintain and keep track of all dependencies and environment configurations needed for correctly building and compiling code specific to a given board before making a binary file available for download. An mBED enabled board connected to your computer via USB will be available as a mass storage device. In order to install the software, you simply drag the bundled binaries over to the mass storage device, and the compiled software will auto-install.

While the development process was eased, programming ARM Cortex M is still complicated, and so is Bluetooth Low Energy. Simulating ECG data is also a non-trivial task, and based on these three factors we decided to use as much available code for the board programming as possible.

In order to simulate the ECG data stream on the board, we reviewed several different existing ECG simulators written in C++ [39, 48, 66]. We consider ECGSYN, a realistic ECG waveform generator created at Oxford and MIT [66], to be the best of them. This is a flexible program that let us customize sampling frequency and mean heart rate among other things. ECGSYN generates a synthesized ECG signal based on algorithms described in [51]. See Listing 1 for an overview of the different adjustable parameters.

```
1 >> ecgsyn $
2
3 ECGSYN: A program for generating a realistic synthetic ECG
4 Copyright (c) 2003 by Patrick McSharry & Gari Clifford . All rights reserved.
5
6 O Name of output data file "ecgsyn.dat"
7 n Approximate number of heart beats 256
8 s ECG sampling frequency [Hz] 256
9 S Internal Sampling frequency [Hz] 256
10 a Amplitude of additive uniform noise [mV] 0
11 h Heart rate mean [bpm] 60
12 H Heart rate standard deviation [bpm] 1
13 f Low frequency [Hz] 0.1
14 F High frequency [Hz] 0.25
15 v Low frequency standard deviation [Hz] 0.01
16 V High frequency standard deviation [Hz] 0.01
17 q LF/HF ratio 0.5
18 R Seed 1
19 (Type ? for Help)
20 ->
```

Listing 1: ECGSYN Commando Line Interface (CLI)

5.2.2 Maximum Throughput

Because experience with embedded software development was limited, we did some preliminary research on max throughput on Bluetooth Smart, before starting the customization and implementation of the ECGYN software on the development board. As mentioned in Section 2, Bluetooth Smart specifies the theoretical max throughput to be 1 megabit/s. However there are always factors limiting this theoretical capacity. In the following section will elaborate on these limiting factors, and comparing our findings with the minimum requirements of clinical ECG.

Once a Bluetooth Smart connection is established between two devices, the connection parameters [78] control the frequency at which data can be sent between the two devices.⁹ The two connection parameters of interest to us are, minimum and maximum connection interval, which specify the minimum and maximum rate at which the client will ask for data from the server. By the Bluetooth SIG specification, both fields have an allowed minimum and maximum value of 6 and 3200. The connection interval is specified to increment in 1.25 ms intervals, so by multiplying these values by 1.25 we get the highest and the lowest frequency at which a central may ask for data: 7,5 ms and 4 seconds (4000 ms).

The GATT protocol specifies different commands for the client to get information about the server. Among these, GATT offers notifications and indications. In high-throughput applications a client can register a notification on a given characteristic. This means that the server will send a notification containing maximum 20 bytes of user data to the client whenever data becomes available. These notifications are buffered by the SoftDevice and ACK-ed in the link layer. The Bluetooth specification allows for up to 6 packets to be sent per connection interval, but this number may vary between different implementations. This gives the following formula for calculating max throughput:

$$\frac{1000 \text{ ms/s}}{CI} \times PPI \times BPP \times 8 \text{ bits/byte} = \textit{Throughput}$$

Here, CI is connection interval, PPI is packets per interval, and BPP represents the number of bytes sent per packet. Because packets per interval and the bytes per packet may vary between devices and operating systems, the maximum throughput over a Bluetooth Low Energy link is relative. However, maximizing

⁹ In this example the mobile gateway acts the role of the client, and the wireless sensor acts the role of the server - the one with data.

Table 7: Throughput on different devices.

Device	Interval (i)	Packets/i	Bytes in packet	Throughput
Optimal	7,5 ms	6	20 Byte	125 kbit/s
iPhone	30,0 ms	4	20 Byte	21 kbit/s
iPhone HID	11,25 ms	4	20 Byte	56 kbit/s
Macbook pro	11,25 ms	4	20 Byte	56 kbit/s
Android (Nexus 4)	7,5 ms	6	20 Byte	125 kbit/s

the allowed values above, we get the following result:

$$\frac{1000 \text{ ms/s}}{7,5 \text{ ms}} \times 6 \text{ packets} \times 20 \text{ bytes/packet} \times 8 \text{ bits/byte} = 128 \text{ Kbit/s}$$

This is substantially lower than the advertised 1 Mbit/s. In Table 7 we present calculated throughputs for different devices based on their implementation of the Bluetooth specification.

To confirm this discovery we installed some firmware [74] to see if we indeed could achieve the 128 kbit/s throughput. However, our LG G4 test device did not support this optimized Bluetooth configuration, and dropped the connection as soon as the throughput test started. This proves that there are even differences between Android devices, as the operating system *should* support the optimal configuration. Based on the discoveries made above we decided not to implement ECGSYN for simulating streaming ECG from the node. The reason for this is addressed in Section 5.5.1. For the end-to-end latency experiment, we based our code on the BLE_HeartRate repository [50] published by the Bluetooth Low Energy team at the mBed platform. The details of this implementation will be addressed in Section 5.5.3.

5.3 Gateway

Although creating a user-friendly application was not part of the scope of this project, we believe some thoughts on usability and context should be put into the creation of any artifact. We did not want to hard-code the UUID address of Nordic chip into the gateway and connecting the data stream with a static endpoint address. Therefore we have created what we consider to be the bare minimum of a personal WBAN gateway user interface. The functional requirements of this user interface can be found in Table 8.

Table 8: Personal Gateway Functional Requirements

FR#	Description	Priority
1	Add new sensor	High
1.1	Scan for nearby BLE devices	High
1.2	Select endpoint	High
1.3	Select BLE Characteristic	Medium
1.4	Add new endpoints	Low
1.5	Give each sensor a name	Low
2	Ability to test sensor connection	Medium
3	Ability to test endpoint connection	Medium
4	List all sensor nodes	Low

With an architecture dependent on a personal mobile gateway we chose a technology that was in accordance with our mission statement: available and open. The Android platform is not only more open than its counterparts, but also supports higher throughput as seen in Table 7. The gateway would ideally be a multi-purpose touch device with a rich user interface, enabling an easy and practical way of setting up and configuring the wireless sensors a patient would be equipped with. Although outside the scope of this project, some thoughts have gone into the practicalities of setting up and managing a network of wireless sensor nodes. For further research we recommend looking into using Near Field Communication (NFC) technology for physically conducting events such as (A) pairing sensors, (B) connecting the gateway to the WBAN and (C) authenticating medical professionals and patients on the personal gateway. We believe these are all use cases where an active NFC chip could drastically improve the user experience, but doing this in a clinical context need more research. See [19] for information on security in NFC and using it as a interface for pairing Bluetooth devices.

In the following section we will describe how we approached the implementation of the gateway application. Because we only intended to make a prototype, some details will be kept out for brevity.

5.3.1 User interface

The functional requirements were prioritized and we developed the user interface in a iterative fashion as more functionality gradually were added. The resulting user interface is not user tested as this was outside the scope of this project. The “front-end” of the application consist of two simple views, which can be found

in Appendix F. In Android, a “view” is called an *Activity*. As the application is opened the user is presented with a list of sensors connected to the gateway. Clicking either a sensor or the plus sign in the downright corner brings the user to the next screen, where you can either modify the existing device or add a new one.

The application has prioritized functionality over reliability, but still some basic error handling and connection tests are implemented. When connecting to both a Bluetooth Smart sensor and endpoint, the application tests the connection. How this is implemented is described in Section 5.3.2 and 5.3.3. When both sensor and endpoint is selected, the user is able to activate the routing of the data by clicking on activate. The user should then be navigated back to the overview of previously added devices.

5.3.2 Bluetooth Module

We have split the Bluetooth functionality out in a separate *Android Service*. These are background running threads with no GUI. Having a loosely coupled application makes it easier to extend. Android services run in a separate background thread, which is important as makes the gateway continue to receive data after the application is minimized.

Having structured the code this way, a future version of the application could replace the Bluetooth service with, say a Zigbee service, if the device supported that interface.

We set up the scanning code to only look for low energy devices, and (as of now) the gateway application will only subscribe to the Heart-Rate characteristic. A previous version of the application made it possible to select what Bluetooth characteristic you wanted to receive notifications on¹⁰ when adding the device. We removed this for the benefit of automatically selecting the heart rate characteristic. This was done for usability¹¹. This is also backed up by our proposition to how a multi-vendor environment could be organized, where sensor specific details could be available in a remote repository, as mentioned in 5.1. For a Bluetooth enabled sensor, this sensor specific configuration file, or “profile”, could include meta data, what characteristics to subscribe to, and configuration settings like sample rate. This will be further discussed in Chapter 6.

¹⁰ This is Bluetooth jargon for subscribing to a certain data value

¹¹ We had already set up the sensor node code to only write data to the heart rate characteristic, it was unnecessary to select the characteristic whenever we connected to it.

The Android SDK supplies interfaces and example code for creating a BLE applications. In our case, the `BluetoothLeService` class sets up the necessary methods to initialize, connect and disconnect to a BLE device. The class can list supported characteristics and read a given characteristic as well as handling the service specific functionality, such as broadcasting data from Bluetooth notifications to other services and activities. Because our use cases only involved a one way data flow between the sensor node and gateway, we did not implement functionality such as writing data back to a given characteristic. In addition to this, the class also handles callbacks that Android's Bluetooth library execute when it is finished with a given action, such as scanning for devices, or on connection changes. The latter is how we test the connection in the `NewDevice` class. When the callback for `onConnectionStateChange` is executed, the Bluetooth service will broadcast an update, which the `NewDevice` activity listens for. These updates will contain one of the following identifying strings:

```
"iot_gateway.ACTION_GATT_CONNECTED";  
"iot_gateway.ACTION_GATT_DISCONNECTED";  
"iot_gateway.ACTION_GATT_SERVICES_DISCOVERED";  
"iot_gateway.ACTION_DATA_AVAILABLE";  
"iot_gateway.EXTRA_DATA";
```

Based on these messages the `NewDevice` activity handles the event. Every callback we have implemented broadcasts an update, and the `EXTRA_DATA` identification is used when the callback `onCharacteristicChanged` is executed. In our case this means that the `HEART_RATE_MEASUREMENT` characteristic is changed, since we set up the sensor node to change data on that characteristic in a given interval.

5.3.3 Endpoint module

The endpoint module extends the Android-DDP[15] library released under Apache License, Version 2.0. This library makes it easy to interact with a Meteor server, which is described further in 5.4. Because of how we did the server implementation, we used the library to connect to a WebSocket endpoint created by the Meteor server. Although the endpoint module also was planned as an individual module, we never finished refactoring all endpoint functionality out to the `MeteorHandler` class which ideally would extend as an Android Service. This had some implications our experiments, which is addressed in Section 5.5.3.

As an intermediate step, we first implemented the DDP-library directly in the `NewDevice` activity. We set it up as a instance-wide private object, that after first being set up, is available in every method. This was done in order to speed up development, and intended this to be refactored out of the Activity at a later stage. When initializing connection with `mMeteor.connect()` one of two callbacks defined in the `MeteorCallback` interface are executed, `onConnect()` or `onDisconnect()`. When the connection is tested and active we pass data directly to the Meteor endpoint by calling `mMeteor.call("addData")` in the `BroadcastReceiver` method. As mentioned, the `BroadcastReceiver` listens for data from the Bluetooth service.

5.4 Server

Inspired by [81] and informed of the development within the standardization communities like HL7 FHIR [37], we decided to mock up a monitoring central using web technologies. In addition to vastly simplify product development for a manufacturer, using web technologies also enables rapid prototyping for small research teams with little resources. The functional requirements of the server prototype was limited. Based on experience with similar technology we decided to write the server software in Javascript. Contrary to it's young age, this is a programming language that has matured a lot over the last couple of years, and is today the most popular technology for full-stack developers [63].

Because of the simple requirements and limited timespan available building the monitoring server, it was decided to utilize a development platform called Meteor. Meteor [53] is an open source platform for developing web and mobile applications based on NodeJS. It comes with MongoDB database support out of the box, and its own data managing layer. This layer is based on a protocol they call Distributed Data Protocol (DDP), which uses WebSockets as a lower layer message transport. DDP implements the publish-subscribe messaging pattern, which lets clients publish and subscribe to data collections over WebSockets [52]. This functionality was ideal for this project as it made us set up an endpoint where the gateway could route data from the sensor to the server in short time. When this data arrives at the server and is inserted in the database, Meteor will automatically synchronize the updates with all of it's clients that are subscribing to that data source. This new way of structuring web applications, is often called *connected-client* or *cloud client*, and it differs from the classic stateless request-response client-server architecture by synchronizing all clients immedi-

ately without the need for refreshing.¹²

It should be noted that WebSockets make use of TCP at the transport layer. Alensanco and Garcia concluded TCP was not suitable for real-time ECG transmission in a wide area network [2]. They suggested UDP as a more suitable transport layer protocol, but they were transmitting data over a 3G network. Our prototype is connected to local area network over a WiFi link, offering substantial network speeds and reliability over telecommunication networks.

In accordance with use case UC2.2, we designed and implemented a client interface for medical professionals to access the ECG data streams. Technically speaking, we wrote the client view using Blaze, Meteor's frontend rendering system. Blaze uses a variant of handlebars templating language[43] called Spacebars. This allows for organizing the HTML code in templates, which can be reused and enable flow control logic like `if-statements` and `for-loops` in the HTML files. Frontend Javascript code handle logic for subscribing and displaying to the data, and is organized in Template-specific files for a clean structure. These can be viewed as template specific controllers. The initial HTML/CSS/Javascript files are fetched by a regular GET request. After this a WebSocket connection is established. This allows for *reactive data sources*, which is another way of saying what we did earlier: After the initial download the data flow back and forth between clients and the monitoring server reactively.

Subscribing to data sources is done in the client by calling `Meteor.subscribe('deviceData');` in the `onCreated()` function of the template controller. After this, a reactive data source is available in the controller which can be accessed in the DOM-template through helper methods. An example of such helper method can be seen in Listing 2:

```
1 Template.dataListComponent.helpers({
2   dataSet: function () {
3     return DeviceData.find({}, {sort: {reachedServer: -1}});
4   }
5 });
```

Listing 2: Template helper exposing a reactive data source

In this example, `DeviceData` is fetched by using a MongoDB-like syntax.

¹² This is all part of a larger paradigm shift, where *data* is sent the wire instead of documents (such as files).

When the data source is updated at the server, the `dataSet` helper method (available from the DOM) will automatically be re-run, updating the data in the HTML DOM.

Because of Meteor's feature rich platform, our prototype has support for user accounts. These can be used to control access to who is allowed to subscribe to different data sources, and what documents in the MongoDB they are allowed to access. However, these features are not relevant for this project, and their implementation details will be omitted.¹³

In the client we have been looking at various ways of rendering the stream of values into an ECG plot. In the planning phase we landed on using a `jke-d3-ecg` [44], an open source chart component for the popular `D3.js` library drawing charts using Javascript.¹⁴ When we implemented this, the bandwidth restrictions in the Bluetooth protocol was already established. As we had decided to not implement the ECG simulator, we did not spend time implementing any plotting features.

5.5 Evaluation

Inconsistencies and missing information in previous research, motivated us to evaluate the evaluate the performance of the technology in our baseline functional tests. To do this we set up one experiment and based the evaluation of the two other requirements on information from a combination of sources. An overall goal with this evaluation was to get indications on how the technology performed in practice. Because of this we link the requirements to our cases:

- **Maximum throughput:** In order for telemetry solution based on low energy sensors to be used in a clinical context, it should support the minimum throughput requirements for clinical ECG. This requirement is therefore critical for all use cases to be considered in the clinical context.
- **Battery lifetime:** The prototype would have to perform at least as good as the existing solutions. This requirement is applicable in both U1,U2 and U3.
- **End-to-end latency:** Would give us an indication of the overall overhead between the physical entities in our proposed design and the interfaces that

¹³ See attached source code.

¹⁴ Note that this would not produce a clinical grade ECG plot, and was mainly considered for demonstrating purposes.

connect them. This applies to use case U5, as this delay must not exceed 3 seconds.

5.5.1 Throughput

In terms of throughput and data generated from ECG there has been a lot of different metrics used in previous research. Part of our motivation for finding the required throughput was clearing up in this confusion, establishing a realistic minimum for throughput required in order to do wireless, clinical grade ECG, within the context of our intended use (as shown in Section 4.1). In Section 5.2.2 we saw the maximum practical throughput over a Bluetooth Low Energy link. In order to evaluate this capacity with the required throughput for clinical grade ECG, will list the different configurations of a clinical ECG, and calculate the throughput required for each configuration, comparing it to the known Bluetooth performance.

Hypothesis: A Bluetooth Low Energy enabled node is able to transmit a continuous data stream equivalent to that of a 5 lead clinical grade ECG.

Method: We base the minimum requirement for throughput in order to stream raw ECG on the size of the data generated per second in clinical ECG within our intended use. This lead to the following equation:

$$\text{sample frequency} \times \text{sample size} \times \text{ECG leads} = \text{data generated}$$

Our approach for finding the different parameters for this equation was to gather as much information on the matter from previous research and existing solutions. While surveying literature, we searched the papers for keywords like “ECG, lead, leads, kb, b/s and Hz”, taking note of the number of leads they based their experiments on, the sampling rate, and the size of each sample. Looking at exported data from available databases or realistic ECG simulators, we tried to establish a metric for the number of bits needed used to represent each sample.

Results: After surveying literature [69, 84, 82, 10, 55, 83, 65, 64, 27, 87, 2, 86, 11, 58], and data samples from test data [66, 67], we were unable to establish *one* clear combination of sample size, sample frequency or number of leads. As mentioned, many base their their calculations on only 1-2 leads, which has been show to be limited [16]. Because of this, we have decided to combine results from different sources in order to establish a basis for further research to investigate:

In terms of sample frequency, we decided to pick [69] as a standard, based on their research strategy and clear results. They recommend a minimum frequency bandwidth of 250 Hz to record ECG, and make it clear that because of Shannon’s theorem sampling rates should be 2-3 times the theoretical minimum.¹⁵

In literature, the size of each sample vary from 10 bit to 48 bit. Looking at data from existing data sets, a sample size of less than 24 bit seemed inadequate. Because the voltage recorded in a ECG are both positive and negative float numbers¹⁶, a sample size able to support an accurate representation of this is necessary.

The number of leads is based on our intended use. As mentioned in Section 4.2.2, today’s telemetry transceivers use 5 electrodes, but the number of leads is not given from the number of electrodes used. According to Wehr, the EASI solution derive most leads from “*a linear transformation of 3 vectors using optimized fixed coefficients.*” [86]. These 3 vectors are in essence the difference in voltage between 3 pairs among 4 of the electrodes, and the last electrode is just a reference (ground). Hence, only 3 actual leads are sampled in the 5 electrode EASI telemetry solution.

In conclusion, we present a set of different combinations of sample size and sample frequency in Table 9, based on our results. To avoid the ambiguity in terms of bandwidth and data sizes, here is an example of how we calculated this matrix:

$$1500 \text{ Hz} \times 48 \text{ bit} \times 3 \text{ leads} = \frac{216000 \text{ b/s}}{1000} = 216 \text{ kbit/s}$$

Together with our findings on maximum Bluetooth throughput presented in Table 7, we see that streaming clinical raw ECG over a low energy Bluetooth link might be possible if the proper combination of device and ECG configuration is used.

5.5.2 Battery

Conducting an experiment to test battery lifetime made little sense as we decided to not implement any form of ECG simulation. Instead we base our battery evaluation on two excel spreadsheets created by Texas Instruments and Nordic Semiconductor, as well as results from earlier research. Texas Instruments have also

¹⁵Note that in signal processing, bandwidth refers to a measure of the width of a range of frequencies. This is measured in Hertz (Hz). This thesis uses the definition from computing, where bandwidth refers to the rate of data transferred. This is measured in bit/s.

¹⁶Varying with the leads and how they are positioned relative to each other.

Table 9: Different ECG configurations when measuring 3 actual leads. Emphasized calculations exceed the optimal throughput of BLE as seen in Table 7.

	500 Hz	1000 Hz	1500 Hz
24 bit/sample	36 kbit/s	72 kbit/s	108 kbit/s
32 bit/sample	48 kbit/s	96 kbit/s	<i>144 kbit/s</i>
48 bit/sample	72 kbit/s	<i>144 kbit/s</i>	<i>216 kbit/s</i>

released a technical document [35] describing the different measurements and techniques for measuring and calculating Bluetooth Low Energy power consumption.

Hypothesis: Streaming continuous data over a Bluetooth Low Energy link with maximum throughput is more energy efficient than the existing telemetry solutions, lasting 24-48 hours [61].

Method: Based on our throughput evaluation, we knew the Bluetooth connection in a 5-lead streaming ECG scenario would be required to be set up with at least optimal connection parameters, as discussed in Section 5.2.2. Because of this, we were able to input these configuration parameters into the Excel sheets provided by Texas Instruments and Nordic, and get a theoretical metric for battery lifetime.

Results: Not surprisingly the two numbers we got from the Excel sheets are different; these are after all just approximations. The results from Texas Instruments' calculations suggest the ECG board would have a life time of ~ 628 hours¹⁷ using a 1000 mAh¹⁸ coin cell battery, while Nordic's calculations resulted in ~ 899 ¹⁹ using the same 1000 mAh battery. Philips' telemetry transceivers are powered by two AA batteries with a capacity of 2500 mAh each. Comparing these results we see that the estimates are way above the the established minimum requirement we set of 48 hours. Even if a real life implementation's battery last only 25% of these estimates, the implementation would still operate $\times 4, 6$ times longer than today's transceivers from Philips. However these are clearly approximations, and there may be many internal and external factors affecting the battery lifetime.

¹⁷ Texas instruments estimation: 26 days.

¹⁸ The coin cell battery with highest available capacity.

¹⁹ Nordic semiconductor estimation: 37 days

An example is the excess power going to indication lights, buttons and potential audible alarms that may be installed on in a commercial sensor node. Philips' transceivers also support for a SpO₂ sensor that is not accounted for these estimations. Comparing the hourly consumption we see that the low energy solution would only consume 1,11 mAh/h while Philips' land on 104,17 mAh/h. Including a 0-75% error margin²⁰, the Bluetooth enabled sensor would reduce hourly power consumption with between 20-98% compared to the existing solution.

5.5.3 End-to-end latency

In order to get a baseline indication on the overall network performance of our prototype we set up an experiment measuring the end-to-end latency. We were interested in the Δt between the node and the client (the two endpoints) and not the intermediate steps. Because of this we set up an experiment where we measured the time it took for a message from the patient's node to the clinician's web client. The goal for this experiment was to compare our delay with the maximum comfortable delay for a ECG transmission found in [2].

Hypothesis: The end-to-end transmission delay in the system is within the comfortable limits of clinical ECG monitoring.

Equipment:

- nRF51 Development Kit
- LG G4 with gateway software
- Monitoring server
- Clinician's web client
- Time synchronization server
- Bluetooth event logging server

Method: Our basic setup here was the same as the solution would have been in practice: A sensor node streams data to a mobile gateway. This gateway routes the traffic to a monitoring server, which stores the data and updates it's connected clients. See Figure 8 for an overview. But in order to accurately measure the time a

²⁰ Because of the great uncertainty of these estimates.

message was sent and received in the client, we had to add two more components, as well as making some modifications to the Nordic development board and web client.

First off we connected the development board to a computer using a USB cable. We changed the original BLE_HeartRate example code that generated and sent a “heart rate measurement”. We modified it to increment a `uint8` integer and send it over the Bluetooth interface²¹ in a 1 second interval. The characteristic can either hold a `uint8` or `uint16`.²² We did not synchronize the clock on the Nordic board with the monitoring server, so we gave this synchronization responsibility to a dedicated time synchronization server, which is described below.

The moment the timestamp was sent to the gateway over BLE, we also logged the value to the connected computer using mBed’s SerialPC interface. This enables the micro controller to communicate with a computer through a USB Virtual Serial Port. On the computer we created a NodeJS application²³ that listened to the serial port and registered the value along with a creating a synchronized timestamp from the time synchronization server. This setup assumes the delay over the USB cable is negligible.

The BLE-emit-logger then sent the incremented value and the timestamp to the monitoring server, where it was registered in a data collection as an event. See Appendix B for an example of the data format used. This was strictly not necessary, as we could have just stored the result locally, but this method eased the post-experiment work as all events from different sources would be located in the same data collection when the experiment finished. We also modified the web client to register the same type of event when it received a new measurement from the monitoring server, i.e. when a document was inserted in the data source. By doing this we achieved a way of measuring the time used, from one end to the other.

In order to get accurate, synchronized timestamps across different machines, we synchronized the web client and BLE-emit-logger with a remote Time Synchronization server we set up on the same machine as the monitoring server was running. This way, both the BLE-emit-logger, the monitoring server and the web client would produce synchronized timestamps. We based the time synchronization server on enmasse.io’s TimeSync library[18] which is a NodeJS implementation of Zachary Booth Simpson’s algorithm for synchronizing time in networked

²¹ To avoid changing the gateway software, we still wrote to the Heart Rate Measurement Characteristic.

²² Integer represented by exactly 8 or 16 bits.

²³ From this point addressed as BLE-emit-logger.

Table 10: End-to-end delay of messages *between* given intervals.

Count	12927	4	3	3	2	2
%o of total	998,2 %o	0,3 %o	0,2 %o	0,2 %o	0,2 %o	0,2 %o
Interval	0 ms	500 ms	1500 ms	2500 ms	3500 ms	4500 ms
Count	2	2	2	2	1	0
%o of total	0,2 %o	0,2 %o	0,2 %o	0,2 %o	0,1 %o	0,0 %o
Interval	5500 ms	6500 ms	7500 ms	8500 ms	9500 ms	10500 ms

based computer games[79]. The experiment ran on NTNU’s network infrastructure, which is available for students and employees. In order to create a realistic routing scenario, we set the monitoring server and time synchronization server up on a machine located in a different building on campus, making sure the traffic wouldn’t only do a single hop through the nearest access point. During the experiment, web client and mobile gateway was located in proximity to each other, as the Nordic board was connected to the same machine as the web client was running. However, this presented no disadvantages as all the information had to travel to back and forth between the Nordic board, monitoring server and web client. A complete graphical overview of the setup can be found in Appendix E. After setting everything up, the experiment ran for 4 hours during night time, sending a total of 12 950 messages from the sensor node to the web client, and registering 25 900 events.

Results: Because the incremented value from the Nordic board (called `msg_id`) was present in both `send` and `receive` events, we matched up pairs of events with the same `msg_id`. The value on the board was initialized with as an 8 bit unsigned integer, and so the counter returned to zero after reaching 255. Because of this we and other reasons²⁴ we sorted the data set first on the `msg_id` field, and then the `timestamp` field, in order to get paired events after each other. After this we calculated the Δt between the events and did further analysis using Excel and MatLab. See Table 10 and a generated a plot in Figure 10.

²⁴ In MongoDB documents are not necessarily stored in the consecutive order they were inserted.

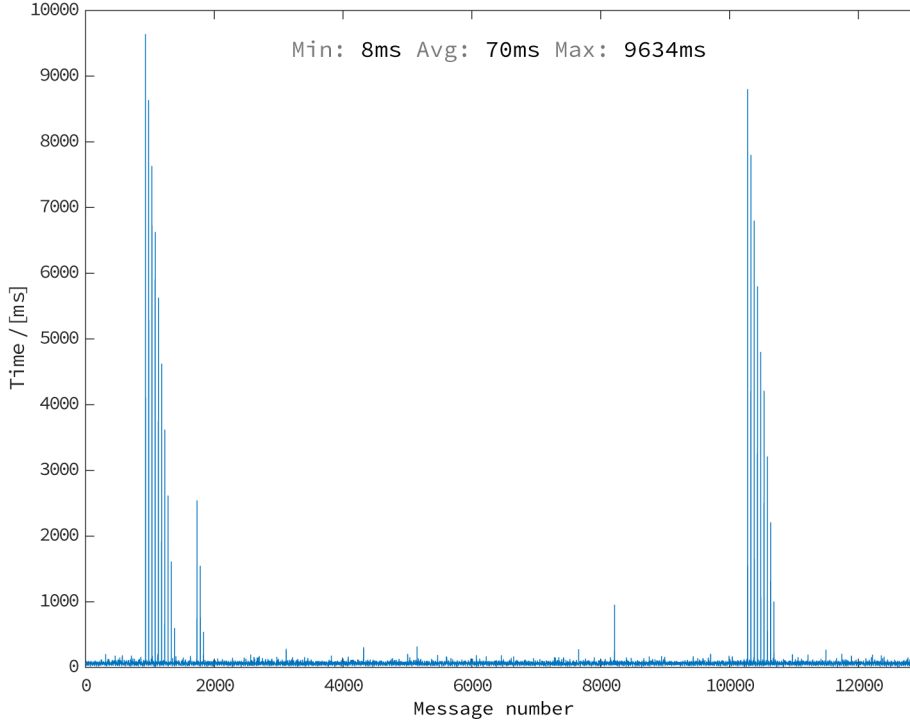


Figure 10: Distribution of transmission delay over the course of 4 hours.

The data has a calculated standard deviation of $\sigma_{delay} \approx 220 \text{ ms}$ which implies positive skewness towards the right. As we see, the general latency is well below what was reported in [2] as a upper limit for clinically analyzing near real-time ECG. However, we did record some transmissions that were severely delayed: The two observed spikes in delay time of the messages portrays a periodic behavior. Within each spike, every 50th message has a “way above average” delay time, every subsequent one with a reduced delay compared to the previous. This implies a systemic cause for these spikes, instead of a random, probabilistic one. We believe these spikes to be an artifact from some component (or combination of such) in our specific technology stack. In Table 10, we see the distribution of these abnormalities when grouped in 1000 ms intervals. The messages are counted between the intervals listed on the bottom row. As such, only 1 message had a delay of more than 9,5 seconds, and 99,82% of the messages arrived less than 500 ms after they were sent. Because the abnormalities are so rare, we don’t consider them important for our research result, as the general trend of 70 ms is well within the comfortable boundaries.

6 Discussion

In this chapter we will pull together and discuss our findings from Chapter 4 and 5. Further we will discuss alternative approaches as well as strengths and weaknesses with our research project.

We believe the usage of smart devices in health care will increase in the coming years, and that the amount of cables will decrease. A ubiquitous future where both medical staff and patients use their private or hospital provided touch devices as a tools in treatment and everyday medical related tasks, seems probable in a not too distant future. Some forms of this have already been deployed, like AFGA Healthcare's ORBISme! tablet sized EMR-device, being used in Germany today [25].

There is however an important difference between consumer technology and medical devices. With the growing interest for personal health monitoring within the consumer market, this distinction easily becomes blurry. Although not having any diagnostic value, it is not improbable that consumer devices soon will see it's way into a medical setting one way or the other.²⁵

In the Section 5 we presented a solution where patient's touch devices was used as a WBAN hub, a mobile gateway connecting WBAN sensors to local area network and external services. On the same side, the consumer market has recently showed an increased interest in "smart access points" or "IoT hubs". These are WiFi access points with built in support for Bluetooth Smart and 802.15.4 in addition to the standard 802.11.X interface.²⁶ Deciding on whether the sensor devices should communicate directly with a smart access point, or through a personal gateway, is an important discussion as pointed out in the paper "The Internet of Things Has a Gateway Problem" by Zachariah et. al. [88]. in the following section we will discuss the possibilities and restrictions with our chosen architecture.

6.1 Toward a Multi-Vendor Situation?

In Section 4 we saw how single-manufacturer, specialized monitoring solutions are deployed at two Norwegian hospitals today.

For our proposed design we chose a two-tier architecture based on a per-

²⁵ Just recently was the first reported case of medical professionals using historic data form a personal fitness tracker to make an informed, life saving medical decision [71].

²⁶ For more information, see OnHub WiFi router from Google [23] and Cassia Hub Bluetooth Router [60].

sonal²⁷, mobile gateway. To enable a multi-manufacturer sensor environment and ensure interoperability in a scalable manner, we proposed using a repository pattern where a mobile gateway could install the “profiles” required for mapping sensor data to the correct formats based on the *context*.

We would also suggest this reduces the flexibility and interoperability of the monitoring solution. Should the backend monitoring system not be developed by the same manufacturer as the WBAN nodes, data need to be transformed into compliant application layer data formats for the information exchange.²⁸ From a practical standpoint, we believe this transformation and similar pre-processing of data scales better at a personal gateway tier. Based on the assumption that the environment will be crowded by similar sensors from different manufacturers, as proposed by [68], we believe it is easier for personal gateways to maintain this responsibility independently, rather than all access points supporting all sensors present in the clinical setting.

However, this is not an easy assumption to be made, as there are both economic, security, maintenance and performance concerns that drive the implementation that we see at today's hospitals. Unless larger partnerships between different companies are established (as we saw with Apple in 2014 [29]), it is unlikely that any manufacturer will come forward and develop a platform or infrastructure for supporting WBANs. Taking Philips' IntelliVue telemetry solution as an example, it just isn't profitable for them to open up for other suppliers to make use of the infrastructure they have installed at hospitals around the globe. That being said, this situation is similar to the one we are experiencing everyday through our internet connections – so to rule out that a change like this might happen, is unwise. However, we believe that as long as wireless sensors and the idea of a “connected self”, is limited to wristbands and bathroom scales - we won't see a change in the hospital infrastructure in regards to supporting multi-manufacturer WBANs.

6.2 Research methods

Working with this many different components and across several domains have also proven to be a challenge. As mentioned in the introduction, scoping this project had to be done along the way. Because the goals and intentions also shifted,

²⁷ Personal in the sense that each patient has its own gateway. Whether or not these are private devices or hospital provided raises several security and privacy questions which is the topic for another discussion entirely.

²⁸ This should indeed be standardized between manufacturers and wireless protocols, but from the experience of HL7 and OpenEHR we know this to be a hard problem.

we ended up making decisions that had to be revisited and in some cases undone at a later point.

Initially we were interested in the practical organization of a future, ubiquitous and distributed monitoring solution based on personal touch devices. We asked questions like “How could alarm triggering events be distributed to different clinicians not located at a central monitoring station?”, “How can clinicians be assured that patient sensors have enough battery for registering an important event? How does this scale?”. However, these questions were in themselves based on assumptions about systems that does not yet exist. Therefore we saw the need for taking a step back and investigate today's existing practices, and the more fundamental aspects of ECG and wireless technologies. We recognize that the practical organization and structure of a wireless and ubiquitous monitoring solution might be a relevant problem statement in the future.

After taking a step back, we realized two things: A multi-manufacturer environment is not well supported by today's hospital infrastructure. This is a fundamental requirement in a future where patients in a larger degree capture sensing information about their bodies outside the hospital.

And from previous literature very few define a clear link between the research and the intended use. As we learned, establishing this intended use is necessary in order to say anything about minimum requirements like throughput, and end-to-end latency. And as we discovered, even when having established a intended use, deciding on these metrics are still no easy task.

After having set the path we did, we are satisfied with the decision of comparing two different hospitals. As we visited the second hospital, several misconceptions were cleared and we got a broadened view, especially on the organization of patients and how the the practice of on-demand monitoring were conducted in relation to the telemetry monitoring. The challenge with qualitative research methods however, is to convert increased personal insight and experience, into knowledge for everyone else. We took an explanatory approach to this in Chapter 4, where we focused on explaining the systems and practices in a logical manner. We did not structure the chapter according to the source of the information, and rather focused on mixing sources. Hopefully this created a more compound impression of today's systems and practices.

We spent a lot of time finding the minimum requirements for ECG, without ever finding any gold standards. This is why we present our own guidelines to this in Chapter 5. Although we searched literature, the reason we didn't find this might be related to our approach and research strategy. For the technical aspects of our prototype, the heavy focus on practice and use cases did not play as big

of a role as we first imagined it to do. An alternative approach could have been to target departments and instances at a higher organizational level for our interviews. Here access to standardization documents and practices might be more accessible, as compliance is usually the concern of middle, and higher level management. However, this approach however could easily suffered from a mismatch between the general guidelines and the technology deployed in practice. This would also increase turnover times, meaning a lot of time would have gone by waiting for answers from the respective higher level organizational departments.

In terms of our data collection, one thing is clear: Getting information from the manufacturers and solution providers is not worth pursuing unless a collaboration or sponsorship is established before the research begins. The time spent running down dead ends in order to find ECG related technical specifications on commercial solutions, was in the end deemed unproductive.

In retrospect we acknowledge that ISO Standards probably would have given us a good insight on the matter, especially ISO 11073-91064:2009 and ISO/IEEE 11073-10406:2012.

6.3 Implementation

In retrospect, we see that aiming for making a flexible solution took much more time to develop, and proved to be generate more edge cases, making the experimentation more prone to errors. Because of the complex setup of our end-to-end latency experiment, more time could have gone into planning this making it more robust. In terms of the gateway, we could have focused less on the user interface. One alternative approach could have been to hard-code the devices, characteristics and endpoints into the gateway, which could have eased the implementation, debugging and testing.

By the time were conducting the first trails of the experiment, we saw that the code (especially on the gateway) did not perform well over time, which was critical for our experiment. We had to do modifications on the fly during the experimentation phase in order to make everything work. We unit tested the individual components manually as we developed, but we left little time for integration tests before conducting the experiment. Had this been planned better we would have reserved time for integration tests. All in all, we see that testing and debugging the time synchronizing on both clients, as well as working with the control flow and user interface of the gateway proved to be most time consuming. It should be mentioned that we intentionally spent as little time as possible on writing custom code for the sensor board. This was because we had little experience with

hardware and software development for low energy development kits.

Implementation wise we admit more time should have been put into prioritizing reliability over functionality.

6.4 Results

Throughput: We decided to exclude signal and data compression from our scope. Investigating this would have made an impact on both the required throughput and the battery usage. Other than this, we hope to have established a good understanding of the different considerations that go into wireless ECG. It is worth noting that we compared our solution to Philips' EASI telemetry lead setup, which uses 3 leads to derive the rest. As such our results illustrates the minimum required throughput (without considering protocol overhead) for a ECG setup measuring 3 actual leads. In another use case, where the intended use is different, the number of actual leads would have to be multiplied accordingly.

End-to-end latency: As the results from our experiment showed, the end-to-end latency is within the acceptable limits. These limits were however set based on only one related study. They clinically tested latency in a ECG monitoring system, but their intended use was in a *telemedicine setting*. It should therefore be noted that the requirements in a in-hospital setting might be different.

A weakness in our experiment was that it was only conducted in accordance to U1, our first use case. We therefore have little insight in how our prototype would perform in a semi-ambulatory or ambulatory setting. Conducting the experiment with the other use cases would have required another experiment setup. Not only because of the USB cable used to decide when values was sent from the sensor node, but also because the increased variables *movement* introduces. This variability will require more details about the communication in-between the different components in the end-to-end test.

Our approach of testing the delay in this “integration testing” type of manner, proved to be both time-consuming, error prone and gave few answers to exactly *why* the delay was as measured. An example of the lack of details can be illustrated with the spikes we experienced. With our setup we had no way of investigating these spikes without doing larger changes in the experiment setup or the code. However, because our desired goal was measured

in the scale of *seconds*, we expected our results to only give us an indication of the feasibility of our prototype. Because results was well within the established limits and the experiment measured delay at the application layer of the network stack, we know the results can be optimized further in the future. Based on this we were satisfied with the experiment and it's results.

Battery: In our evaluation of the battery capacity we excluded investigating the battery consumed when sampling and processing data. This decision was based on the assumption that the radio would be the largest consumer of energy in our setup. However, this assumption makes our already general approximates even more uncertain, and the battery consumption ECG streaming devices should be investigated further. In Section 6.5 we recommend an alternative approach that may reduce the radio usage further. Another thing we did not assess in our evaluation was the personal gateway's battery. Although one fundamental requirement of the gateway will be that it affords recharging, the actual battery consumption of continually (and periodically depending on the situation) streaming is another factor that threaten the feasibility of our proposed design.

6.5 Future work

From a priori knowledge and studying Bluetooth Smart we know that the biggest energy consumer is the radio during the actual wireless transmission. In essence, version 4.0 of Bluetooth were able to reduce the energy consumption drastically compared to previous versions by powering off the radio in between events, instead of keeping a persistent connection. Based on this and our insight in today's practices we realize that there might be worth investigating radically different monitoring practices - like self-monitoring sensors. There is a trade off between doing onboard analysis on the sensor chip versus streaming the data continuously to a WBAN gateway which have to be investigated further. It might be more energy efficient to implement algorithms for analyzing sampled data on the sensors themselves and only transmit data on demand from clinicians or when deviations from given thresholds occur.

Exploring the boundaries of classical monitoring is exiting, and might be proven to yield results: During one interview with one of the medical professionals it was mentioned that in his experience, the prevalence of cardiac abnormalities of medical interest might happen during physically straining activities like walking stairs etc. However, today they had no practical way of seeing this correlation.

The only place where you can monitor the real-time ECG stream is at the central monitoring station. A great deal of planning and synchronization would have to happen in order to conduct a simple ECG-test on a patient under physical strain: One would have to record the time and place while watching the patient doing the activity, then return to the monitoring central and look at the history in order to get a clear view of how the patient's heart reacted to the activity. A better solution would be ability to watch the ECG on a tablet device while observing the patient doing the activity. An improved version of this could record indoor location, number of steps and altitude together with the ECG. This way, the medical professional wouldn't have to consult with neither the patient nor the monitoring central. They could just look at the historic ECG data with activity data overlaid.

Should small, inexpensive and available patient monitoring solutions enable monitoring of every patient, regardless of medical condition, problems regarding data storage and compression will have to be addressed. Our calculations from Section 5.5.1 show that collecting raw ECG data from a single patient amount to between 388,8 megabyte²⁹ and 3110,4 megabyte³⁰ on a daily basis. This is only raw data, and packaged in a standard compliant data format, this data would occupy a lot of storage. Yearly, only the raw data amounts to a Terabyte of data per patient on a single physiological metric.

One key area that is in need for inexpensive solutions based on available technology and open standards are within "Information and Communication Technologies for Development" or ICT4D for short. This denotes using information and communication technology in the fields of socioeconomic development, international development, and human rights. Enabling inexpensive ECG monitoring in a rural area is definitively a possible direction for further development of this technology.

²⁹3 leads, 24 bit sample size and 500 Hz sampling rate

³⁰3 leads, 64 bit sample size and 1500 Hz sampling rate

This page has been intentionally left blank.

7 Conclusion

In this thesis we have explored clinical wireless ECG monitoring. Through qualitative research and the creation of a prototype based on a low energy wireless node, we have evaluated the technology against some aspects of wireless ECG. In this master thesis we want to answer the following: *Is it possible to create a monitoring solution for wireless ECG, based on available technology and open standards?*

As we have discussed in previous sections, part of our underlying motivation for finding an answer to this question was to learn more about the future of sensor technology in a clinical setting. Within the context of our intended use, and on the attributes we evaluated, we have showed that it is indeed feasible to create such artifact today, using available and open technology such as Bluetooth Smart.

An inexpensive, flexible and scalable low energy monitoring solution might enable more widespread collection of physiological data among patients outside the intended use presented in this thesis. Using available technology and open standards, this form of data collection used for patient monitoring can be achieved today. By using open web standards and standardized data formats like HL7 FHIR, one might allow for increased data flow between hospital information systems and personal sensory equipment.

In conclusion, this thesis has established some guidelines to certain technical requirements of conducting wireless ECG. This was done through the creation of a prototype that performed well within the acceptable limits of end-to-end latency. We have highlighted different clinical considerations that are an integral part of doing research within this field. We've also managed to give some insight in the different practices of patient monitoring today, that hopefully can give some indications on what directions the future might take. Although using open and standardized technology could enable a multi-manufacturer organization of WBANs in hospitals today, we don't believe these solutions will see the light of day before both patients and clinicians alike place such a demand.

Glossary

ACS Acute Coronary Syndrome.

AES Advanced Encryption Standard.

DICOM A file format and network communications protocol that is used to transfer different medical information online.

ECG Electrocardiogram: a record or display of a person's heartbeat produced by electrocardiography.

EMR Electronic Medical Record.

HFU Heart Failure Unit.

HIS Hospital Information Systems.

HL7 Health Level 7. International standardization organization.

Holter Monitor A Holter monitor is a machine that continuously records ECG. Is normally worn 24-48 hours during normal activity..

ICU Intensive Care Unit.

IMC Intermediate Care Unit.

NodeJS Event-driven I/O server-side JavaScript environment based on Chrome's V8 engine..

OUS Oslo University Hospital.

PAS Patient Administration System.

SoC System on a Chip.

References

- [1] Adafruit. *Search results for nrf51822*. 2016. URL: <https://www.adafruit.com/?q=nrf51822&> (visited on 18th May 2016).
- [2] Á Alesanco and J Garcia. “Clinical Assessment of Wireless ECG Transmission in Real-Time Cardiac Telemonitoring”. In: *IEEE Transactions on Information Technology in Biomedicine* 14.5 (Aug. 2010), pp. 1144–1152.
- [3] Continua Alliance. *Bluetooth Smart to Continua*. 2016. URL: <http://www.continuaalliance.org/find-products-certification-process/bluetooth-smart-to-continua-ready> (visited on 18th May 2016).
- [4] ARM. *ARM mBED Developer Site*. 2016. URL: <https://developer.mbed.org/> (visited on 18th May 2016).
- [5] ARM. *ARM The Architecture for the Digital World*. 2016. URL: <https://www.arm.com/> (visited on 18th May 2016).
- [6] Alere AS. *Alere Corpuls 3*. 2016. URL: <http://www.alere.com/se/sv/brands/akut/Corpul3.html> (visited on 18th May 2016).
- [7] Dag Torstein Ausen, Eirik Næss-Ulseth and Karl Øyri. *Biomedical Wireless Sensor Network - phase II*. Tech. rep. /0/774/1555/1566/INovelda AS/Oslo University Hospital/, 2010.
- [8] E. Baccelli et al. “RIOT OS: Towards an OS for the Internet of Things”. In: *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. Apr. 2013, pp. 79–80. DOI: 10.1109/INFOCOMW.2013.6970748.
- [9] Mohammadreza Balouchestani, Kaamran Raahemifar and Sridhar Krishnan. “Low Sampling-rate Approach for ECG Signals with Compressed Sensing Theory”. In: *Procedia - Procedia Computer Science* 19 (2013), pp. 281–288.
- [10] M. Barua et al. “Secure and quality of service assurance scheduling scheme for WBAN with application to eHealth”. In: *2011 IEEE Wireless Communications and Networking Conference*. Mar. 2011, pp. 1102–1106. DOI: 10.1109/WCNC.2011.5779285.

- [11] Hadda Ben Elhadj et al. “Multi-Attribute Decision Making Handover Algorithm for Wireless Body Area Networks”. In: *Comput. Commun.* 81.C (May 2016), pp. 97–108. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2016.01.007. URL: <http://dx.doi.org/10.1016/j.comcom.2016.01.007>.
- [12] Plux Wireless Biosignals. *Bitalino Freestyle Kit*. 2016. URL: <http://bitalino.com/index.php/freestyle-kit> (visited on 18th May 2016).
- [13] *BLE Nano — RedBear*. 2016. (Visited on 18th May 2016).
- [14] Chris Otto, Aleksandar Milenković, Corey Sanders, Emil Jovanov. “System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring”. In: *Journal of Mobile Multimedia* (Jan. 2006), pp. 307–326.
- [15] Delight Development. *Android-DDP*. 2014. URL: <https://github.com/delight-im/Android-DDP> (visited on 18th May 2016).
- [16] B J Drew et al. “12-lead ST-segment monitoring vs single-lead maximum ST-segment monitoring for detecting ongoing ischemia in patients with unstable coronary syndromes.” In: *American journal of critical care : an official publication, American Association of Critical-Care Nurses* 7.5 (Sept. 1998), pp. 355–363.
- [17] Rikard Eide. *Wireless ECG: Repository overview*. URL: <http://rix1.github.io/master/> (visited on 2nd June 2016).
- [18] Enmasse.io. *TimeSync on Github - Time synchronization between peers*. 2015. URL: <https://github.com/enmasseio/timesync/> (visited on 25th May 2016).
- [19] Ernst Haselsteiner and Klemens Breitfuß, Philips Semiconductors. *Security in Near Field Communication (NFC)*. Paper presented at RFID Security 2006. Apr. 2006.
- [20] TIK WSN Research Group at ETH Zurich. *The Sensor Network Museumtm - Tmote Sky*. 2016. URL: <http://www.snm.ethz.ch/Projects/TmoteSky> (visited on 18th May 2016).
- [21] Inc. Fitbit. *Fitbit Aria™ Wi-Fi Smart Scale*. 2016. URL: <https://www.fitbit.com/no/aria> (visited on 18th May 2016).

- [22] Inc. Fitbit. *Fitbit Official Site for Activity Trackers More*. 2016. URL: <https://www.fitbit.com/> (visited on 18th May 2016).
- [23] Google. *OnHub from Google. A better way to Wi-Fi*. 2016. URL: <https://on.google.com/hub/> (visited on 18th May 2016).
- [24] Bluetooth Special Interest Group. *Bluetooth® technology adding mesh networking to spur new wave of innovation*. Feb. 2015. URL: <https://www.bluetooth.com/news/pressreleases/2015/02/24/bluetoothtechnology-adding-mesh-networking-to-spur-new-wave-of-innovation> (visited on 19th May 2016).
- [25] AGFA Healthcare. *ORBIS ME!* 2016. URL: http://www.agfahealthcare.com/germany/de/main/products_services/Krankenhaus-Informationssystem/ORBIS_ME/index.jsp (visited on 18th May 2016).
- [26] LD Hudson. “Design of the intensive care unit from a monitoring point of view”. In: *Respiratory care* 30.7 (July 1985), pp. 549–559. ISSN: 0020-1324. URL: <http://europepmc.org/abstract/MED/10315668>.
- [27] Hsien Che Hung and D. Wei. “Muscle noise cancellation in four-lead ECG system for ubiquitous healthcare”. In: *Pervasive Computing (JCPC), 2009 Joint Conferences on*. Dec. 2009, pp. 115–118. DOI: 10.1109/JCPC.2009.5420203.
- [28] IBM. *IBM Design Thinking: Human-centered outcomes at speed and scale*. 2015. URL: <http://www.ibm.com/design/thinking/> (visited on 19th May 2016).
- [29] MD Iltifat Husain. *Why Apple’s partnership with Epic is a game changer for patients*. URL: <http://www.imedicalapps.com/2014/06/apple-partnership-epic-game-changer/> (visited on 18th May 2016).
- [30] Microchip Technology Inc. *RN42 Class 2 Bluetooth radio*. 2016. URL: <http://www.microchip.com/wwwproducts/en/RN42> (visited on 18th May 2016).
- [31] Zolertia™ Inc. *IoT hardware*. 2016. URL: <http://zolertia.io/store> (visited on 18th May 2016).
- [32] Texas Instruments Incorporated. *Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com*. 1970. URL: <http://www.ti.com/> (visited on 18th May 2016).

- [33] Texas Instruments Incorporated. *Texas instruments Wireless Heart Rate Monitor Reference Design*. Tech. rep. Texas Instruments, Sept. 2014, pp. 1–40.
- [34] Texas Instruments Incorporated. *TI Designs*. 2016. URL: <http://www.ti.com/general/docs/refdesignsearchresults.tsp> (visited on 18th May 2016).
- [35] Texas Instruments. *CC2540, CC2541 AN092 – Measuring Bluetooth Low Energy Power Consumption*. Tech. rep. Texas Instruments, 2012.
- [36] Health Level Seven® International. *Health Level Seven International Homepage*. URL: <http://www.hl7.org/> (visited on 18th May 2016).
- [37] Health Level Seven® International. *Index - FHIR v1.0.2*. URL: <http://www.hl7.org/fhir/?ref=learnmore> (visited on 18th May 2016).
- [38] *Basic electrocardiograph (ECG) (1- to 3-lead ECG)*. Norm. 2012-12-01.
- [39] Peter Isza. *MobileECG II at Github*. 2016. URL: <https://github.com/peterisza/mobileecg> (visited on 18th May 2016).
- [40] Antonio J Jara et al. “Evaluation of Bluetooth Low Energy Capabilities for Continuous Data Transmission from a Wearable Electrocardiogram”. In: *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE, May 2012, pp. 912–917.
- [41] Paul Johannesson and Erik Perjons. “Introduction”. In: *An Introduction to Design Science*. Cham: Springer International Publishing, 2014, pp. 1–19.
- [42] J.S.S.STEWART. “The aim and philosophy of patient monitoring”. In: *Postgraduate Medical Journal* 46 (June 1970), pp. 339–343.
- [43] Yehuda Katz. *Handlebars.js*. 2015. URL: <http://handlebarsjs.com/> (visited on 26th May 2016).
- [44] Joakim Kemeny. *jke-d3-ecg at Github.com*. Jan. 2014. URL: <https://github.com/joakimkemeny/jke.d3.ecg> (visited on 19th May 2016).
- [45] Kari Kvaerner. *Ror IT og helse i samme retning?* <http://www.ntnu.no/documents/21469517/1266455774/kari-kvaerner-ror-it-og-helse-i-samme-retning-2015.pdf/7358456c-fe9b-426d-b68a-c37dc4cd0f6d>. Sept. 2015.

- [46] Young-Dong Lee and Wan-Young Chung. “Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring”. In: *Sensors and Actuators B: Chemical* 140.2 (July 2009), pp. 390–395.
- [47] Deloitte Southeast Asia Ltd. *Healthcare 3.0 Healthcare for the new normal*. <http://www2.deloitte.com/content/dam/Deloitte/sg/Documents/risk/risk-sea-healthcare-3.0-healthcare-for-the-new-normal.pdf>. 2013.
- [48] James P. Lynch. *Arduino Menta ECG Simulator project*. 2013. URL: https://github.com/lynchzilla/ecg_simulator (visited on 18th May 2016).
- [49] John Maeda. *Better Health By Design: Making Healthcare Tech More Usable, Understandable And Profitable*. 2015. URL: <http://techcrunch.com/2015/12/08/better-health-by-design-making-healthcare-tech-more-usable-understandable-and-profitable/> (visited on 18th May 2016).
- [50] Bluetooth Low Energy Team at mBed. *BLE HeartRate Repository*. 2014. URL: https://developer.mbed.org/teams/Bluetooth-Low-Energy/code/BLE_HeartRate/# (visited on 18th May 2016).
- [51] P. E. McSharry et al. “A dynamical model for generating synthetic electrocardiogram signals”. In: *IEEE Transactions on Biomedical Engineering* 50.3 (Mar. 2003), pp. 289–294. ISSN: 0018-9294. DOI: 10.1109/TBME.2003.808805.
- [52] Meteor Development Group (MDG). *DDP Specification*. Oct. 2014. URL: <https://github.com/meteor/meteor/blob/devel/packages/ddp/DDP.md> (visited on 19th May 2016).
- [53] Meteor. *Meteor App Development Platform*. URL: <https://www.meteor.com/> (visited on 19th May 2016).
- [54] Aleksandar Milenković, Chris Otto and Emil Jovanov. “Wireless sensor networks for personal health monitoring: Issues and an implementation”. In: *Computer Communications* 29.13-14 (Aug. 2006), pp. 2521–2533.
- [55] J. Misić and V. B. Misić. “Bridge performance in a multitier wireless network for healthcare monitoring”. In: *IEEE Wireless Communications* 17.1 (Feb. 2010), pp. 90–95. ISSN: 1536-1284. DOI: 10.1109/MWC.2010.5416355.

- [56] BioRadio: Wireless Wearable Human Monitoring. *BioRadio SDK Download*. 2016. URL: <https://glneurotech.com/bioradio/support-page/sdk-download/> (visited on 18th May 2016).
- [57] BioRadio: Wireless Wearable Human Monitoring. *The BioRadio Wireless Physiology Monitor*. 2016. URL: <https://glneurotech.com/bioradio/bioradio-wireless-physiological-monitor/> (visited on 18th May 2016).
- [58] Samaneh Movassaghi et al. “Wireless Body Area Networks: A Survey”. In: *IEEE Communications Surveys & Tutorials* 16.3 (July 2014), pp. 1658–1686.
- [59] United Nations. *World Population Prospects - 2015 Revision*. Tech. rep. United Nations, 2015.
- [60] Cassia Networks. *Cassia Hub Bluetooth Router*. 2016. URL: <http://en.cassianetworks.com/> (visited on 18th May 2016).
- [61] Koninklijke Philips N.V. *Training guide: TRx4841A and TRx4851A IntelliVue Transceivers for the IntelliVue Telemetry System*. Tech. rep. Philips Healthcare, Feb. 2007.
- [62] Koninklijke Philips Electronics N.V. *IntelliVue Telemetry System Infrastructure Installation and Service Guide*. 2007. URL: <http://www.manualslib.com/manual/815492/Philips-Intellivue-Telemetry-System-Infrastructure.html?page=16> (visited on 18th May 2016).
- [63] Stack Overflow. *Developer Survey Results 2016*. Feb. 2016. URL: <http://stackoverflow.com/research/developer-survey-2016> (visited on 19th May 2016).
- [64] C. Park et al. “An ultra-wearable, wireless, low power ECG monitoring system”. In: (Nov. 2006), pp. 241–244. ISSN: 2163-4025. DOI: 10.1109/BIOCAS.2006.4600353.
- [65] Young-jin Park and Hui-sup Cho. “Transmission of ECG data with the patch-type ECG sensor system using Bluetooth Low Energy”. In: *2013 International Conference on ICT Convergence (ICTC)*. Oct. 2013, pp. 289–294. DOI: 10.1109/ICTC.2013.6675359.
- [66] PhysioNet. *ECGSYN: A realistic ECG waveform generator*. 2003. URL: <https://physionet.org/physiotools/ecgsyn> (visited on 18th May 2016).

- [67] PhysioNet. *PhysioBank Archive Index*. URL: <https://physionet.org/physiobank/database/#ecg> (visited on 18th May 2016).
- [68] A. M. Rahmani et al. “Smart e-Health Gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems”. In: *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. Jan. 2015, pp. 826–834. DOI: 10.1109/CCNC.2015.7158084.
- [69] P R Rijnbeek, J A Kors and M Witsenburg. “Minimum bandwidth requirements for recording of pediatric electrocardiograms.” In: *Circulation* 104.25 (Dec. 2001), pp. 3087–3090.
- [70] RIOTos. *RIOT - The friendly Operating System for the Internet of Things*. URL: <https://www.riot-os.org/#about> (visited on 19th May 2016).
- [71] Joshua Rudner et al. “Interrogation of Patient Smartphone Activity Tracker to Assist Arrhythmia Management.” In: *Annals of emergency medicine* (Mar. 2016).
- [72] Nordic Semiconductor. *Nordic Semiconductor Developer Portal*. 2016. URL: <https://developer.nordicsemi.com/> (visited on 18th May 2016).
- [73] Nordic Semiconductor. *nRF51 DK*. 2014. URL: <https://www.nordicsemi.com/eng/Products/nRF51-DK> (visited on 18th May 2016).
- [74] Nordic Semiconductor. *nrf51-ble-app-lbs - Nordic Throughput Test on Github*. 2014. URL: <https://github.com/NordicSemiconductor/nrf51-ble-app-lbs/tree/throughput-test-5.2.0> (visited on 18th May 2016).
- [75] Nordic Semiconductor. *Questions - Nordic Developer Zone*. URL: <https://devzone.nordicsemi.com/questions/> (visited on 18th May 2016).
- [76] Mohammadreza Sahebi Shahamabadi et al. “A Network Mobility Solution Based on 6LoWPAN Hospital Wireless Sensor Network (NEMO-HWSN)”. In: *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE, May 2013, pp. 433–438.
- [77] Shimmer. *Consensys ECG Development Kits*. 2016. URL: <http://www.shimmersensing.com/shop/ecg-development-kit> (visited on 18th May 2016).

- [78] Bluetooth SIG. *Bluetooth Core Specification 4.2*. Tech. rep. Bluetooth SIG, 2014.
- [79] Zachary Booth Simpson. *A Stream-based Time Synchronization Technique For Networked Computer Games*. 2000. URL: <http://www.mine-control.com/zack/timesync/timesync.html> (visited on 22nd May 2016).
- [80] Inc. iRhythm Technologies. *Zio patch Listen to Your Heart Arrhythmias*. 2016. URL: <https://www.zioreports.com/Application.html#HOME> (visited on 18th May 2016).
- [81] Sebastian Thelen et al. “Using off-the-Shelf Medical Devices for Biomedical Signal Monitoring in a Telemedicine System for Emergency Medical Services”. In: *IEEE Journal of Biomedical and Health Informatics* 19.1 (Dec. 2014), pp. 117–123.
- [82] Farid Touati et al. “Feasibility and performance evaluation of a 6LoWPAN-enabled platform for ubiquitous healthcare monitoring”. In: *Wireless Communications and Mobile Computing* (2015), n/a–n/a. ISSN: 1530-8677. DOI: 10.1002/wcm.2601. URL: <http://dx.doi.org/10.1002/wcm.2601>.
- [83] Sana Ullah et al. “A Comprehensive Survey of Wireless Body Area Networks”. In: *Journal of Medical Systems* 36.3 (Aug. 2010), pp. 1065–1094.
- [84] D. J. Vergados, D. D. Vergados and I. Maglogiannis. “NGL03-6: Applying Wireless DiffServ for QoS Provisioning in Mobile Emergency Telemedicine”. In: *IEEE Globecom 2006*. Nov. 2006, pp. 1–5. DOI: 10.1109/GLOCOM.2006.260.
- [85] S. Warren et al. “Interoperability and Security in Wireless Body Area Network Infrastructures”. In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2005, pp. 3837–3840. DOI: 10.1109/IEMBS.2005.1615297.
- [86] Gabriele Wehr et al. “A vector-based, 5-electrode, 12-lead monitoring ECG (EASI) is equivalent to conventional 12-lead ECG for diagnosis of acute coronary syndromes.” In: *Journal of electrocardiology* 39.1 (Jan. 2006), pp. 22–28.

-
- [87] B. Yu, L. Xu and Y. Li. “Bluetooth Low Energy (BLE) based mobile electrocardiogram monitoring system”. In: *Information and Automation (ICIA), 2012 International Conference on*. June 2012, pp. 763–767. DOI: 10.1109/ICInfA.2012.6246921.
- [88] Thomas Zachariah et al. “The Internet of Things Has a Gateway Problem”. In: *the 16th International Workshop*. New York, New York, USA: ACM Press, 2015, pp. 27–32.

List of Figures

1	QRS Complex with interval annotations	15
2	Overview of the Bluetooth protocol stack on a hostess system [72]	18
3	From IBMs Design Thinking framework	25
4	IntelliVue Monitoring System Components [62]	31
5	IntelliVue Network Topology [62]	32
6	TRx transceiver from Philips	33
7	Annotated ECG plot of lead V1-V6. See Appendix D for all 12-leads.	34
8	Two alternative architectures. Here A represents Bluetooth Smart while B is 802.11.	43
9	Overview of system and communication protocols. AP indicates an access point.	45
10	Distribution of transmission delay over the course of 4 hours. . . .	63

List of Tables

1	Characteristic ECG intervals from [40]	16
2	Use case: Stationary	36
3	Use case: Partly Ambulatory	37
4	Use case: Ambulatory	38
5	Use case: Set up gateway	39
6	Use case: Monitor ECG	40
7	Throughput on different devices.	50
8	Personal Gateway Functional Requirements	51
9	Different ECG configurations when measuring 3 actual leads. Emphasized calculations exceed the optimal throughput of BLE as seen in Table 7.	59
10	End-to-end delay of messages <i>between</i> given intervals.	62

A SoC Considerations

This section contain an overview over the different systems on a chip (SoC) and boards we considered.

RedBears Nano [13] development boards were of particularly interesting because of their form factor. They are based on the Nordic NRF51822 chip and measure only 18.5mmx21.0mm. However, we because our goal was not related to mobility the actual size of the chip was not important to us. Another board of interest was Bitalino’s development board for capturing body signals [12]. Among other things, it comes with interfaces and built in analog to digital converter for ECG leads. This could have been a good starting point for our prototype, but the development kit only supports Bluetooth 2.0 which was of little interest to us.

We looked at Zolertia [31] development boards, and even ran some preliminary tests using simulated Zolertia Z1 motes and the Contiki Cooja Simulator. However, these are based on the 802.15.4 specification, and was therefore outside the scope of this project. The same accounts for the Tmote Sky [20], which have been used extensively in earlier research [54, 14, 64, 85, 64, 7]

Among more medical targeted development kits we looked at Shimmer Tools [77] which offer professional, medical grade wireless sensors suited for “academic, applied and clinical researchers”. These were of particular interest because of the configurable ECG options and that it shipped with libraries [77] for streaming and parsing data on the Android platform. However, the Bluetooth module used for wireless transmission is based on the RN-42 chip [30], which only supports the 2.1 version of the Bluetooth protocol. Another, similar development kit targeting clinical trials, research and teaching labs, is the BioRadio from Great Lakes NeuroTechnologies [57]. This is a highly configurable device with both adjustable sampling rate (250Hz-16kHz) and sample resolution (12-24 bit), as well as a SDK for pc [56]. To our knowledge the BioRadio has a newer Bluetooth chip supporting version 4.0. However, because of the high sample rate and resolution, it is highly unlikely that they utilize the low energy features of the specification, due to the bandwidth constraints of the low energy specification. A further discussion on Bluetooth bandwidth versus the minimum requirements for ECG can be found in Sect. 3. It is also unlikely that the firmware is customizable in commercial products like the ShimmerTools and BioRadio, and was these two devices was therefore not interesting to us.

After investigating these different development platforms, we decided to avoid development kits capturing real physiological data (Bitalino, ShimmerTools and

BioRadio) for the sake of testability and reproducibility. This way it would be easier to conduct consistent experiments. However, these devices offer a great deal of functionality and possibilities both in terms of hardware capabilities, technical specifications and ease of development, and should be considered in later research.

Texas Instruments another major actor in the embedded community. They design and develop a wide array of semiconductors and hardware for industries ranging from audio and hifi to medical devices [32]. In fact they have a own sensory product line for biological signals. TI also has an active research community publishing reference designs for various applications branded under the name TIDesigns [34]. In the later stages of this project, we became aware of an for a 5-lead ECG monitor based on Bluetooth Low Energy [33]. The existence of this reference design have not influenced neither our design choices nor research questions, as it was discovered late in the project. An overview of the TI's proposed design and the implications an earlier discovery of this reference design could have had are discussed in Sect. 6.

B Data formats

Data format for events logged in experiment 5.5.3:

```
1
2 // Event registered when a message was
3 // sent from the Nordic BLE development kit
4
5 {
6     "_id" : "MBSbcKFQKErd6WFFA",
7     "msg_id" : "16",
8     "timestamp" : 1464302487111,
9     "eventtype" : "send",
10    "client_id" : "ble_node"
11 }
12
13 // Event registered when message arrived
14 // in web client
15 {
16     "_id" : "4P5r3pR6GG4qdawdq",
17     "msg_id" : "16",
18     "timestamp" : 14643024891232,
19     "eventtype" : "receive",
20     "client_id" : "LJxMcrPSnpP5TpFoo" // SessionID of the web client
21 }
```

Listing 3: Event data format

C Battery Lifetime

The following two graphics are exported from the Excel spreadsheets supplied by Texas Instruments and Nordic Semiconductor. In the first graphic, the resulting calculations are marked in yellow, and in the second graphic they are placed within the black box.

Status:	OK		
Battery capacity (mAh):	1000		
Connection Interval (ms):	7,5		
Sleep Current with timer running (mA)	0,001		
	Case 1		
	Time (us)	Current (mA)	Percent of events
			50
State 1 (wake-up)	400	6	2400
State 2 (pre-processing)	315	7,4	2331
State 3 (pre-Rx)	80	11	880
State 4 (Rx)	275	17,9	4922,5
State 5 (Rx-to-Tx)	105	7,4	777
State 6 (Tx)	115	18,2	2093
State 7 (post-processing)	1325	7,4	9805
State 8 (pre-sleep)	160	4,1	656
State 9	0	0	0
State 10	0	0	0
State 11	0	0	0
State 12	0	0	0
State 13	0	0	0
State 14	0	0	0
State 15	0	0	0
State 16	0	0	0
State 17	0	0	0
State 18	0	0	0
State 19	0	0	0
State 20	0	0	0
Total time of connection event	2775		23864,5
Average Current draw during connection event (mA):		8,5998198	
Average current draw accounting for sleep (mA):			3,182563
Average current draw during connection (mA):	1,591781667		
Expected battery life (hours):	628,2268611		
Expected battery life (days):	26,17611921		

Advertising						
Interval					1000	ms
Payload					31	bytes
Channels					3	
Broadcast					Off	
DC/DC					Off 1	
Average current					30,8	uA

Stage	Description	Current Calculation	Time (uS)	#	Time _{total} (uS)	Current	Charge	%
(A)	Pre-processing	$I_{ON} + I_{RTC} + I_{X32k} + I_{CPU_Flash} + I_{START_X16M}$	125	1	125	5 203	650	2,1%
(B)	Standby + XO ramp	$I_{ON} + I_{RTC} + I_{X32k} + I_{START_X16M}$	675	1	675	1 103	745	2,4%
(C)	Standby	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M}$	600	1	600	473	284	0,9%
(D)	Radio start/switch	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + \int I_{START_TX} + I_{CPU_Flash}$	130	1	130	11 573	1 505	4,9%
(E)	Radio start	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + \int I_{START_TX}$	130	2	260	7 473	1 943	6,3%
(F)	Radio TX	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + I_{TX_0dBm}$	376	3	1128	10 973	12 378	40,1%
	Broadcast break	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M}$	300	0	0	473	142	0,5%
(G)	Radio turn-around	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + \int I_{START_RX} + I_{CPU_Flash}$	130	3	390	13 273	5 177	16,8%
(H)	Radio RX	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + I_{RX}$	59	3	177	13 473	2 385	7,7%
(I) ₁	Post-processing BLE	$I_{ON} + I_{RTC} + I_{X32k} + I_{CPU_Flash}$	60	2	120	4 103	492	1,6%
(I) ₂	Post-processing BLE	$I_{ON} + I_{RTC} + I_{X32k} + I_{CPU_Flash}$	500	1	500	4 103	2 052	6,7%
(I)	Post-processing APP	$I_{ON} + I_{RTC} + I_{X32k} + I_{CPU_Flash}$	0	1	0	4 103	0	0,0%
(J)	Idle	$I_{ON} + I_{RTC} + I_{X32k}$	995 895	1	995 895	3,1	3 087	10,0%
	Peripherals	Sleep mode	1 000 000			0,0	0	0,0%
30 838								

Connected						
Connection Interval					7,5	ms
Average current					1 112,1	uA

Stage	Description	Current Calculations	Time (uS)	Current	Charge	%
(A)	Preprocessing	$I_{ON} + I_{RTC} + I_{X32k} + I_{CPU_Flash} + I_{START_X16M}$	125	5 203	650	7,8%
(B)	Standby + XO ramp	$I_{ON} + I_{RTC} + I_{X32k} + I_{START_X16M}$	675	1 103	745	8,9%
(C)	Standby	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M}$	600	473	284	3,4%
(D)	Radio Start	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + \int I_{START_RX} + I_{CPU_Flash}$	90	7 473	673	8,1%
(E)	Radio RX	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + I_{RX} + I_{CRYPTO}$	192	13 473	2 596	31,1%
(F)	Radio turn-around	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + \int I_{START_TX} + I_{CPU_Flash}$	72	13 273	956	11,5%
(F)	Radio turn-around	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + \int I_{START_TX}$	80	9 173	734	8,8%
(G)	Radio TX	$I_{ON} + I_{RTC} + I_{X32k} + I_{X16M} + I_{TX_0dBm} + I_{CRYPTO}$	78	10 973	865	10,4%
(H)	Post-processing	$I_{ON} + I_{RTC} + I_{X32k} + I_{CPU_Flash}$	181	4 103	743	8,9%
(I)	Idle - connected	$I_{ON} + I_{RTC} + I_{X32k}$	5407	3	17	0,2%
	Peripherals	Active	7 500	10,5	79	0,9%
8 341						

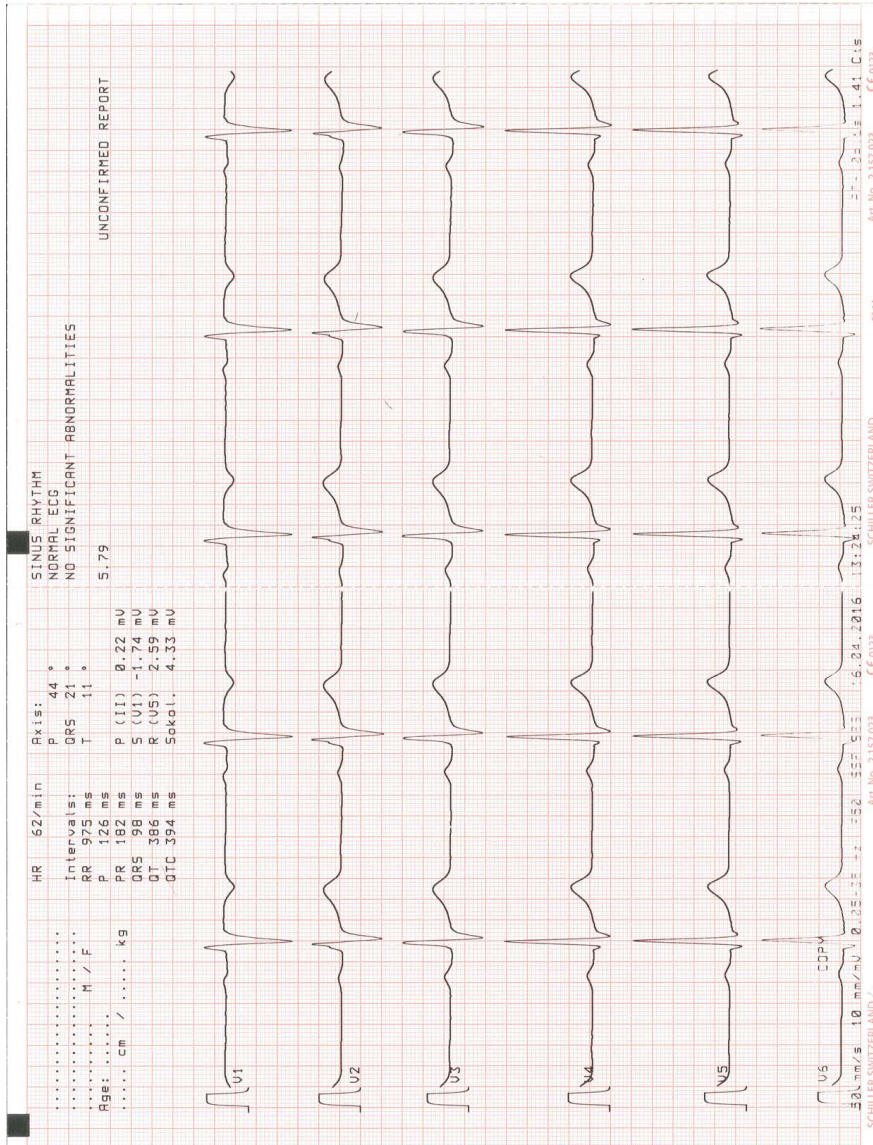
Battery			
Battery	CR2477	1000	mAh
Connected	1440	Minutes per day	100,0%
Connected	9748	mAh/yr	
Advertising	0	mAh/yr	1,1 mAh/h

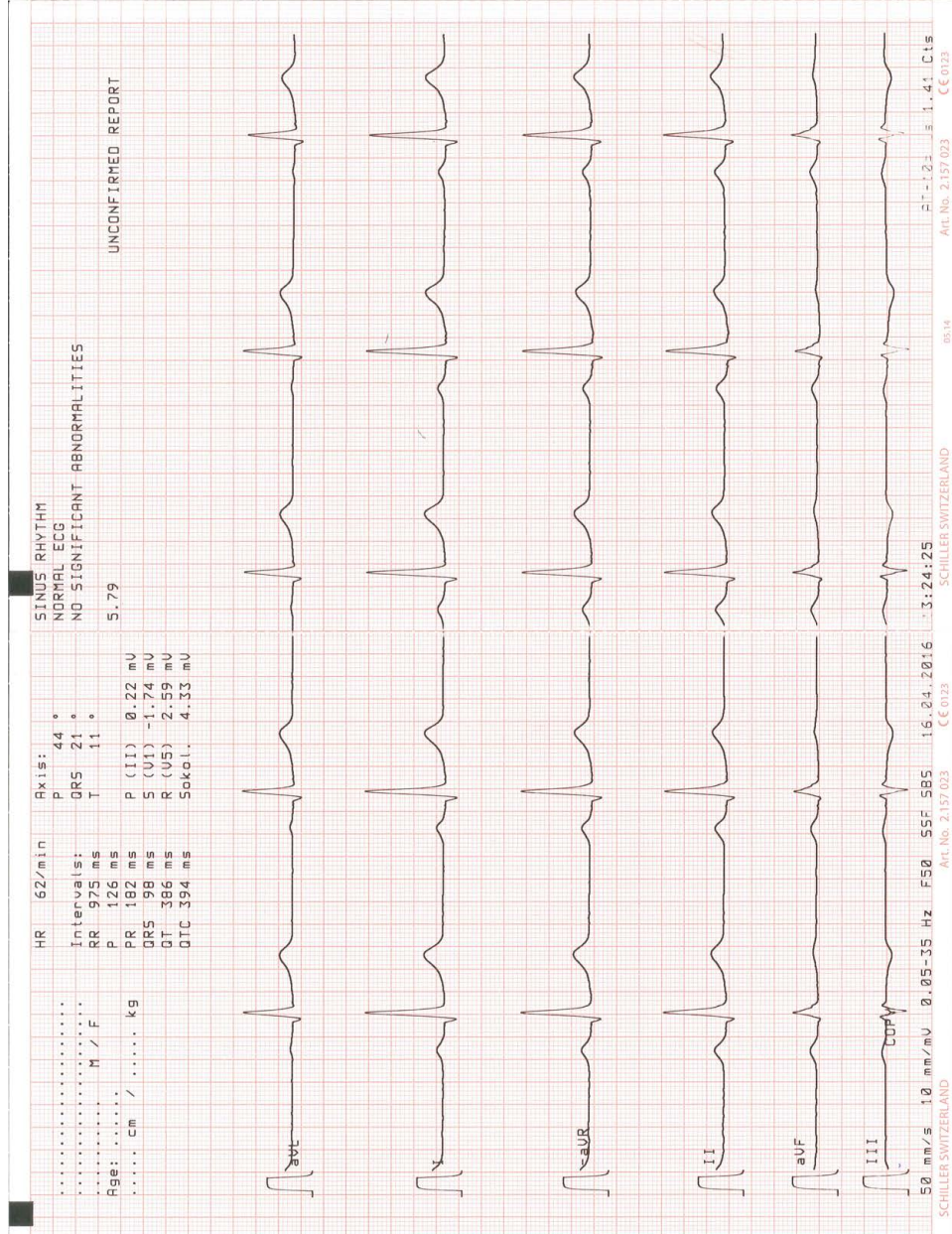
9748	mAh/yr	899	Hours
0,1	Years	37,5	Days

uA			
I_{ON}	#REF!	On 16kB RAM	
I_{RTC}	0,1		
I_{X32k}	0,4		
I_{CPU_Flash}	4100		
I_{START_X16M}	1100		
I_{X16M}	470		
I_{TX}	#REF!	POUT = +0 dBm	LFCLK
I_{START_TX}	7000		LFCLK _{peripheral} 20 ppm nRF51822 32.768kHz clock
			LFCLK _{central} 40 ppm Typical 40ppm for iOS devices
I_{START_RX}	8700		
I_{RX}	13000		
I_{CRYPTO}	550		

D ECG Plot

The following graphic is a full 12-lead ECG taken of a clinician during our observational studies.



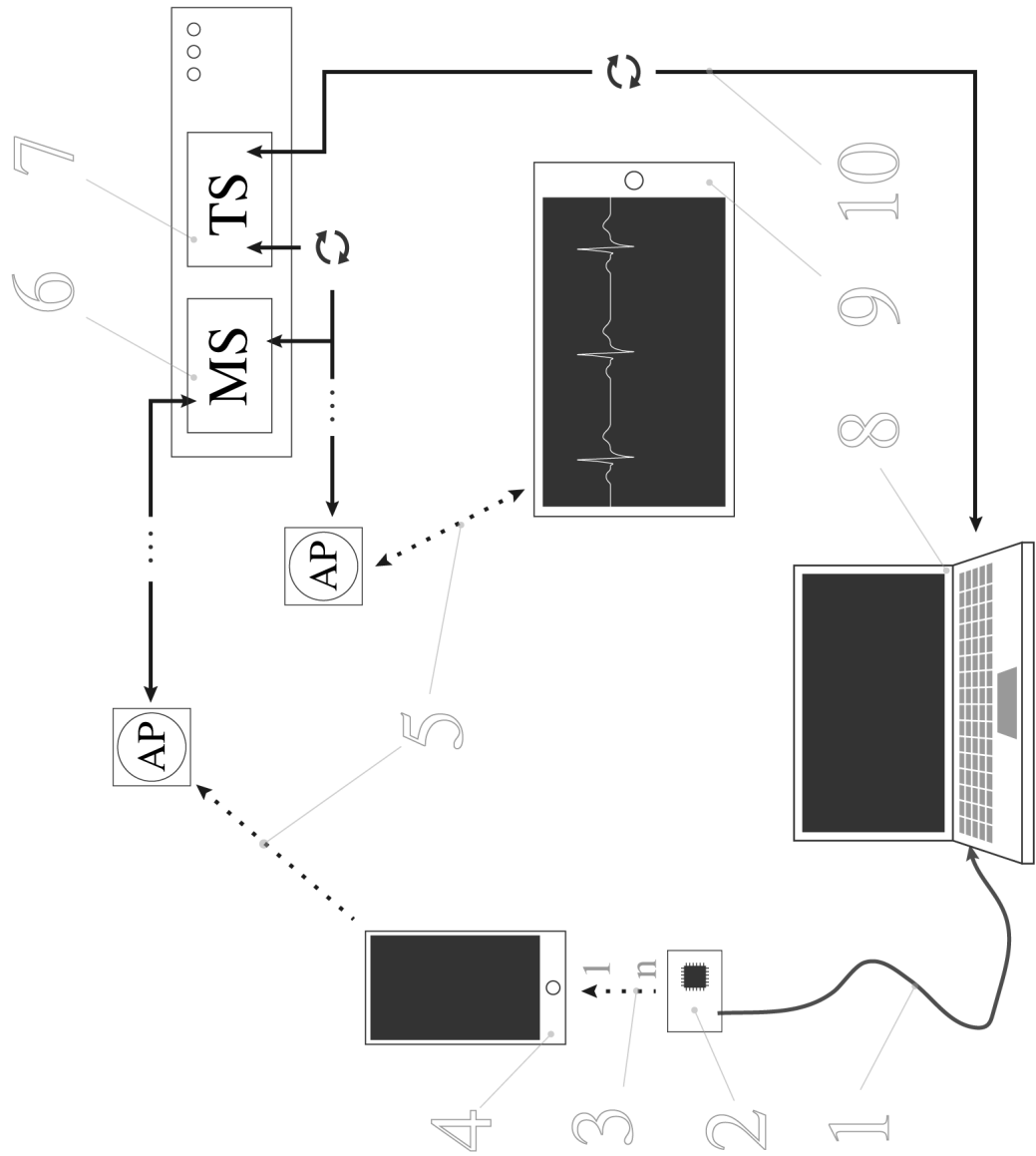


E Experiment setup

The following diagram is a visualization of the setup and different components we used for our end-to-end latency experiment. The different components are as follows:

1. USB cable
2. Sensor node
3. Bluetooth Low Energy Connection
4. Android Gateway Software
5. WiFi
6. Meteor Monitoring server
7. Time Synchronizer server
8. Laptop
9. Web Client
10. Ethernet connection

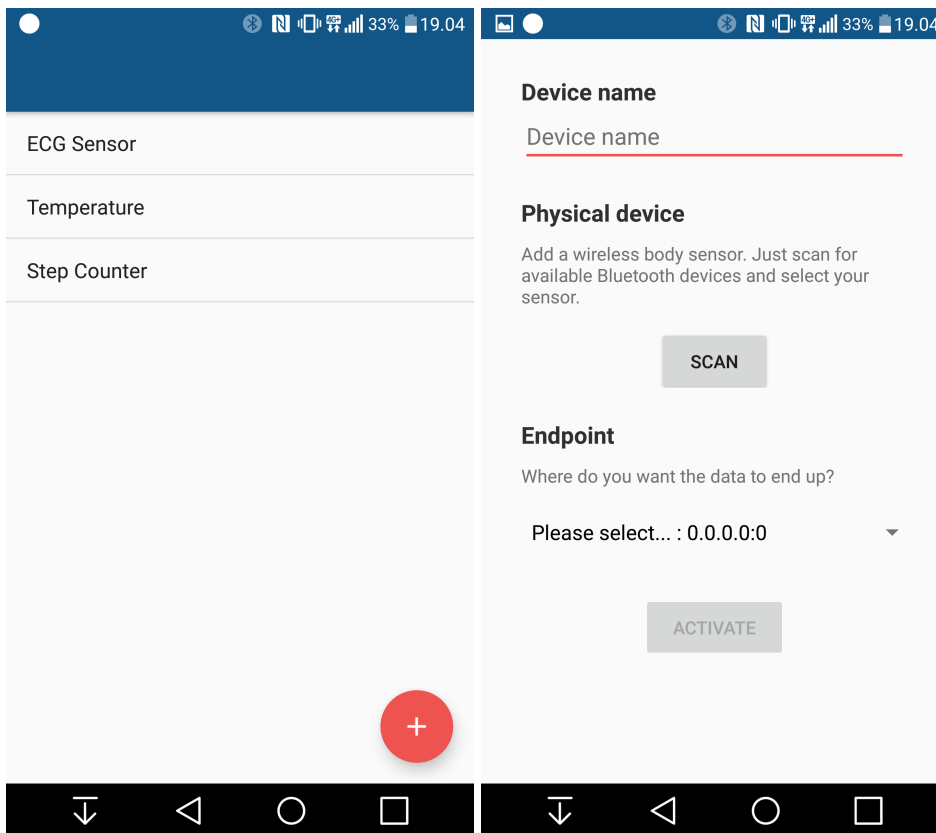
AP denotes an access point.



F Gateway Screens

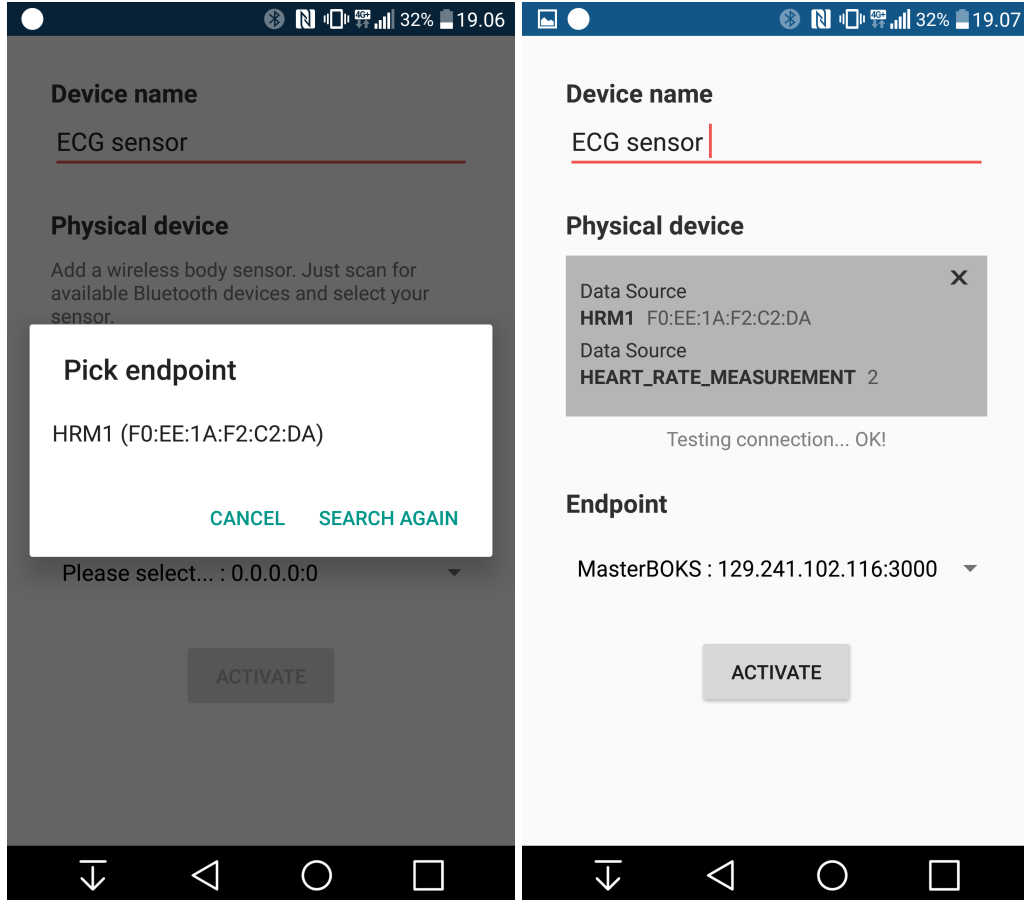
Below is the different user views of the personal gateway application listed in a chronological order. The first view lets a user list all the sensors currently connected to the person. When a user adds a new device with the red button, it can add a new device by:

1. Giving it a name.
2. Search for the wireless Bluetooth enabled sensor.
3. Select an endpoint.
4. Activate the sensor by clicking the “Activate” button.



1: List all sensors

2: Add new sensor



3: Search for Bluetooth Smart device

4: Sensor set up and is streaming