Military applications for IoT

Utilizing soldier wearables for enhanced battle space Situational Awareness

Rune Langleite

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2021

Military applications for IoT

Utilizing soldier wearables for enhanced battle space Situational Awareness

Rune Langleite

Military applications for IoT

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Contents

List of Figures

List of Tables

Acknowledgements

I would like to extend my gratitude to my supervisors, Frank T. Johnsen and Carsten Griwodz, for their excellent support and guidance through this thesis. I would also like to direct a special thanks to Ann-Kristin Elstad in relation to use of interviews involving serving military personnel and appropriate research methodologies.

I would also like to thank my friends and family for their continuous support throughout these years towards acquiring my degree. Finally, I would like to thank my partner Aoife for outstanding support throughout this journey while also taking care of our two daughters, a feat which is nothing short of incredible, and I would like to dedicate my work in this thesis to them.

Abstract

Internet of Things, or IoT for short, is a multi-domain, multi-paradigm technology topic rapidly gaining popularity both for commercial use and among hobbyists due to its inherently automated behavior and cheap development costs. With the emergence of available wireless technologies in combination with small-sized hardware, it has become one of the defining technologies of the last decade, where its role towards “smartifying" cities and businesses is nothing short of central. It has therefore also gained the attention of innovators of military technology, where its role could also prove to be central towards gaining information dominance in the battle space through enhanced and augmented situational awareness.

In this thesis, a prototype subsystem of Military IoT taking the form of a soldier wearable was built using commercially available software and hardware, supported by a private network and information-chain built solely on open source, independent of existing infrastructure, so as to showcase the ability to provide military deployments with a self-driven, ad-hoc network of sensors. In order to support the concrete design, serving military personnel have been involved so as to provide important details surrounding their leadership approach in a variety of military mission cases, in addition to provide feedback on the finished prototype so as to conclude whether or not such a system would help increase operational effectiveness.

The prototype developed in this thesis utilized commercial devices to build sensing on the wearers geographical position, biometric readings, and gas detections, which was designed to transmit as often as possible so as keep the update rate as high as possible, while also restraining the time on air to a minimum to lower the risk of electromagnetic detection by hostile Eletronic Warfare units, in addition to limiting battery usage, and thus also operational lifetime of the device.

The findings in this thesis indicate that increased battle space awareness can be made possible through automated data acquisition at soldier level. It is therefore recommended that military organizations partaking in similar mission cases such as the ones presented in this thesis further investigates the usability of such a system.

Chapter 1

Introduction

This thesis explores leveraging civilian consumer electronics for military applications. The purpose is identifying the value of bringing Internet of Things (IoT) into operational use by exploiting its pervasive nature for soldier wearable sensor kits. In this chapter, we define central terms and provide motivation for the work. The research questions pursued and the methodology applied in the research are also discussed.

1.1 About the Internet of Things

In the past couple of decades, we've seen a surge in ground-breaking, disruptive, and innovative paradigms that changed the way we think of machines and interconnected things. One of them, IoT, is quickly gaining foothold in numerous areas. Examples include agriculture, cargo tracking, electricity metering, noise- and air pollution measurements, waste management, smart parking, and smart homes. The definition of IoT, as stated by the Global Standards Initiative (GSI) on Internet of Things and International Telecom Union (ITU) standard (Global Standards Initiative (GSI) 2015; ITU 2012):

> A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

Essentially, IoT encompasses not only a vast array of communication layer technologies, but also hardware and software across multiple domains, from edge to User Interface (UI). IoT devices can be classified into four categories (Russell and Abdelzaher 2018):

1. **Data-carrying device:** Connects the physical object, or thing, to communication networks.

2. **Data-capturing device:** Interacts with the physical thing through a read/write device.

3. **Sensing and actuating device:** Acquires information in its environment, or conducts physical operations, such as measuring temperature (sensing) and solar panel direction adjustments (actuating).

4. **General device:** Refers to embedded devices that may carry both sensing, actuating, and communication capabilities, depending on its application domain.

IoT as a business asset has already proven its potential, with an estimated 21.5 billion IoT-specific devices connected to the Internet (Statista Research Department 2020), and an estimated revenue of $11.1 trillion per year by 2025 (Manyika et al. 2015). Contributing factors to this rise in popularity are, most notably, low development costs and ease of connectivity. Commercial Off-The-Shelf (COTS) equipment such as microcontrollers, Field Programmable Gate Arrays (FPGAs), System-on-Chips (SoCs), sensors, and actuators are the physical building blocks for virtually any IoT implementation, attributed with low purchase costs and relatively simple implementation processes using well-documented and well-supported software and tools. With the emergence and commercialization of wireless technologies such as Radio Frequency Identification (RFID) (ISO 2018), Bluetooth (Bluetooth Special Interest Group (SIG) 1989), ZigBee (ZigBee Alliance 2004), Wi-Fi (IEEE Standards Association 2016b), 4G, and 5G in the future, connectivity is available virtually anywhere, making the second main factor for the huge commercial success IoT has shown. Finally, with the massive increase in data volumes following large scale sensor deployments, the demand for big data storage, processing, and analytics, have also increased drastically. As a consequence, these services are now commercially available with most major cloud service providers. Thus, the three crucial building blocks for IoT systems are currently available, affordable, and relatively low-effort.

1.2 Motivation

Due to the increased commercial usage of IoT-related technologies, it has become a field of interest for military applications due to the importance of information in an increasingly complex and modernized battle space, as stated by the NATO Science & Technology Organization (STO) in its science and technology trends report for the 2020 to 2040 time frame:

> The information domain or info-sphere, is a unique operational environment. This domain is driven by the digitisation and virtualisation of individuals, organisations and societies. [...] 5G and the internet-of-things (IoT) will also increasingly enable the use of the info-sphere.

The report outlines Emerging and Disruptive Technologies (EDT) which is believed to play a crucial role towards increased operational and organisational effectiveness through, among others, knowledge and decision advantage (Reding and Eaton 2020). For this particular reason, this

thesis will investigate the usage of IoT wearables through commercially available technologies, in order to establish its applicability within the military.

1.3 Problem description

Currently, military operations are largely relying on voice communications for effective coordination between units on the ground. In the heat of battle, information conveyed using voice transmissions often includes mistakes or contains information gaps. This extends to administrative tasks, logistics, medical evacuations, standard reporting, and more. Thus, information dissemination can advantageously be automated further in order to lower voice communications usage, which have certain clear benefits. First, it will provide combat units with more availability to coordinate their maneuver, rather than spending a lot of time conducting for instance resupplies or providing information to medical units for evacuation purposes. Second, it offloads personnel for manual tasks which traditionally involves heavy human interaction, such as inventory checks and subsequent status updates. Third, it provides a more timely and precise information dissemination, assuming a low presence of false positives and negatives. This can also be combined with Big Data analysis in order to predict when certain needs arise in the future. For instance, given a pattern in resource usage such as fuel and ammunition consumption, automated alerts and tasking can be conducted on behalf of the commanding elements in order to save precious time for the troops in combat. Ultimately, the goal of the work in this thesis is to investigate the usability of soldier wearables in terms of enhanced or augmented Situational Awareness (SA) by developing a prototype using COTS hardware and open source solutions. A wearable can be defined as follows (Hayes 2020):

> Wearable technology, also known as “wearables", is a category of electronic devices that can be worn as accessories, embedded in clothing, implanted in the user’s body, or even tattooed on the skin. [...] The rapid adoption of such devices has placed wearable technology at the forefront of IoT.

SA can be defined in very simple terms as an appropriate awareness of a situation, i.e. knowing what is going on around us. In the literature, three definitions seem to dominate, of which the following puts emphasis on perception and understanding of the world with some aspect of future projection (Stanton, Chambers and Piggott 2001):

> Situational awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and a projection of their status in the near future.
>
> — M. R. Endsley (1988)

SA is widely considered a crucial foundation for successful decision making in many fields, in particular ones where human safety is of high importance, such as Air Traffic Control (ATC), law enforcement, emergency management, and military operations. In this thesis, we will consider SA in the context of military operations only.

1.4 Scope and limitations

IoT is in general terms a very broad field which overlaps with a large number of specific disciplines, such as data analysis, wireless technologies, antenna- and Radio Frequency (RF) theory, microcontrollers, and so forth. Based on the previously stated definition of IoT, the work in this thesis will focus on wearables, built using COTS equipment and open source resources, supported by an ad-hoc and on-demand deployable backend infrastructure independent of civilian infrastructure for connectivity.

As such, the resulting development work in this thesis will not attempt to adhere to NATO Standardization Agreements (STANAGs) or Military Standards (MIL-STDs) requirements for military communications equipment, in particular with respect to ruggedization and security measures. In addition, Big Data and thorough data analysis, albeit being a crucial component in large-scale data collection systems, will not be part of the work in this thesis due to time limitations. In addition, general security concerns will not be part of the work in this thesis, except for built-in security mechanisms in the technologies that were used in the development process.

Finally, a simple Graphical User Interface (GUI) will be developed for the purpose of showcasing and visualization of the soldier wearable, however it will not attempt to adhere to UI/User Experience (UX) best practices, as battle space information would normally be made available for the end user on existing BMSs, which are not available due to their classification.

1.5 Research methodology

The development process in this thesis will pursue a hybrid methodology using both software engineering principles and a qualitative approach based on semi-structured interviews. The software engineering methodology is described in (Bahga and Vijay 2014), which outlines ten distinct stages for designing and implementing a generic IoT system, including a brief requirement specification.

The qualitative method will mainly utilize a series of interviews for the purpose of acquiring necessary details surrounding current day operational requirements so as to specify functional system requirements, and finally to evaluate the prototype developed in the frame of the software engineering methodology, which can be summarized as follows:

1. **Purpose and Requirements Specification:** Describes the purpose, behavior, and requirements of the system using natural language. This will be largely based on the findings from the first interview based on the given use cases, which will be outlined in Section 4.3, but also defined in the frame of the problem description outlined in Section 1.3 and Section 1.4.

2. **Process Specification:** Formally describes the use cases of the IoT system using process diagrams, which is based on and derived from the purpose and requirements step. As such, this will be conceptually based on the defined cases used for the interviews outlined in Section 1.5.1, and validated through the findings of the first interview outlined in Section 4.3.

3. **Domain Model specification:** Produces a Domain Model which describes the main concepts, entities and objects in the domain of the IoT system, which will be conceptually based on the findings from the literature in Chapter 2 and 3.

4. **Information model specification:** Defines the structure of all the information in the IoT system through Information Models, where the base entity is the Virtual Entity defined in the Domain Model, and defines their attributes and relations, thus establishing a more fine-grained level of detail to the IoT system. As such, these models will be described at a high level in Section 4.5.4, and realized in Chapter 5.

5. **Service specifications:** Defines the services in the IoT system, their types, inputs, outputs, endpoints, schedules, preconditions, and service effects. At a conceptual level, these will be outlined in Chapters 2, 3, and 4, and finally realized in Chapter 5.

6. **IoT Level Specification:** Uses an IoT level descriptor and logically describes where devices, resources, controllers, services, applications, analytics, and database are grouped, locally or in the cloud, and how they are connected and communicate. This system level description will be specified in Section 4.5.6, and realized in Chapter 5.

7. **Functional View Specification:** Defines the functions of the IoT systems grouped into Functional Groups (FGs), namely device, communication, services, management, security, and application. Each FG either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts, and each identified FG maps to components specified in the IoT Level Specification. This specification will be conceptualized in Section 4.5.7, and realized in Chapter 5.

8. **Operational view specification:** Defines various options pertaining to the IoT system deployment and operation by mapping FGs defined from the Functional View Specification to concrete solutions. These solutions will be outlined in Section 4.5.8 and then realized in Chapter 5.

9. **Device and Component Integration:** Integration of devices and components, often while using high-level schematics, where the conceptual description of the device is outlined in Chapter 4, and concrete system level description is outlined in Chapter 5. The findings from the second interview will be used as an operational anchoring towards evaluation of the developed prototype.

10. **Application development:** The development of the IoT system that adheres to the defined specifications, where the findings from the second interview is the sole means of measure for the prototype evaluation. The application as a whole, including the information pipeline, is discussed in Chapter 5.

In relation to the qualitative method, two sets of one-on-one interviews using serving military officers were conducted using a semi-structured interview guide, of which one was used for fact finding and the other was used for evaluation purposes. For the fact finding interview, three specific military mission cases were used in which the informants were placed in the role as Ground Force Commander (GFC) and Operations Officer (OpsOff), before being presented with a high-level design idea for the soldier wearable.

The GFC and OpsOff roles were chosen as they are the leaders organizationally closest to the use cases for the soldier wearable, where the GFC is the direct commanding element holding the best informed and most current perspective about the immediate situation on the ground, whereas the OpsOff is the first line of support from rear elements holding a generally broader view of the situation extending to other actors in the area, including enemies and other friendlies.

The informants were presented with the same case twice, in which the first established an understanding regarding how the informants relate to mission-specific information flows. Following the soldier-wearable presentation, the informants were prompted to hypothetically apply the soldier wearable into the same cases and explain how, and if, they would treat the cases differently, and thus determine whether the informants considered the soldier wearable a positive or negative supplement.

The second interview was used to evaluate the prototype through a controlled simulation, in which the informants were presented with a simple GUI demonstrating the behavior of the physical prototype. The informants were, in the role as GFC and OpsOff, prompted to provide feedback regarding the potential usability for such a system within the military organization today, thus determining whether or not the system design was fit for purpose, and finally acquire suggestions that should be considered for improving soldier wearables in future work.

Informants having a background from combat or reconnaissance units were preferred. The interviews lasted between 90 and 120 minutes, and was recorded for transcription purposes. The transcriptions do not include any personal information pertaining to any of the informants so as to maintain their anonymity, and excludes half-formulated sentences and

Figure 1.1: Social patrol in Kosovo during the KFOR mission (photo: Torgeir Haugaard / Norwegian Armed Forces)

similar that did not produce answers to the presented questions. The recordings were deleted once the transcription was complete.

Excerpts from both interviews containing key aspects surrounding the presented MIoT subsystem are highlighted in the design- and evaluation phase in Chapters 4 and 5, respectively, while the transcribed interviews can be found in Appendix G and H.

1.5.1 Introduction and fact finding interview

In the following, each case that will be used in the fact finding interview is described. Note that the selected cases in this thesis is based on the author's own operational experiences, and the difference in terms of tempo, and strategical and tactical approach. Thus, the system design was put into multiple, differing use cases, thereby narrowing down its potential operational value in the future.

Case 1: Social patrol in urban environments

In, for instance, the International Security Assistance Force (ISAF) mission in Afghanistan and the Kosovo Force (KFOR) mission in Kosovo, social patrols were conducted regularly for the purpose to build trust among the civilian population and to establish a better understanding regarding the situation in their area, as shown in Figure 1.1. These patrols are usually conducted by a dismounted foot patrol and accompanied by an interpreter if necessary. Such a case encompasses standard routines for reporting and coordination with other units in the area and to commanding elements. Thus, it is a well-suited case for establishing a baseline for use cases where there is no immediate hostile activity.

Figure 1.2: Urban warfare training by the Norwegian Army 2nd Battalion (photo: Preben Aursand / Norwegian Armed Forces)

Case 2: Urban assault against a fortified enemy

This case is a high-paced offensive operation in which a ground assault team has been deployed to eliminate an enemy that has established fortified positions within multiple adjacent buildings, as shown in Figure 1.2 depicting infantry about to move into a building with defending hostiles. In general, military doctrines recommend that as an assaulting unit, you should be 3 times the size of the defending team to expect success, adding to the complexity of the case, which involves not only the assault teams, but also support elements such as medical and logistics.

Case 3: Long Range Recon Patrol (LRRP)

This case is in large part an opposite to Case 2, where a 4-man foot patrol is covertly infiltrating a hostile area in order to establish visual surveillance over a given area, as shown in Figure 1.3 depicting a LRRP unit being dropped of by helicopter as part of their infiltration phase. These operations usually last for multiple days, where the patrol remain largely static in an Observation Post (OP) once they have successfully infiltrated the area. Furthermore, it encompasses strict sound and light discipline so as to not get spotted by nearby enemy forces. In modern warfare, the presence of Electronic Warfare (EW) forces the LRRP to exercise Electronic Protection Measures (EPM) best practices, most notably by transmitting as little as possible using as short messages as possible.

Figure 1.3: LRRP during infil (photo: Ole-Sverre Haugli / Norwegian Armed Forces)

1.5.2 Feedback and evaluation interview

The feedback interview is central to the prototype evaluation where a simulation will be used, since the current pandemic situation prevents any live exercises. Synthetic, pre-programmed data will be used to display a foot-mobile infantry patrol moving through the terrain in a given area, initially moving in a single row, before changing formation to a line, and finally indicate that the patrol was engaged in a firefight. This simulation constitutes the discussion foundation for the feedback interviews.

1.6 Contribution

This thesis contributes with an analysis of the applicability of a MIoT subsystem taking the form of a soldier wearable, based on the primary goal of using IoT to improve combat effectiveness through enhanced SA. We will address three problem areas in the thesis, with one specific research question (R) associated with each problem area. The research in this thesis will be limited to answering the questions in context of the specific cases outlined above, not all possible cases that can arise in operations involving the Norwegian Armed Forces.

1. What areas for improvement can be identified from current Modus Operandi (MO) in the Norwegian Armed Forces with respect to information acquisition and dissemination at soldier level?

    - R1: How can an IoT wearable improve the current MO in the Norwegian Armed Forces?

2. In what way can IoT-related technologies enable autonomous information acquisition and dissemination in common military operations, where the technology is to be integrated on a rifleman platform?

- R2: In what way can an IoT wearable enable autonomous information acquisition and dissemination?

3. What existing technologies, be it hardware, software, and paradigms, can be best used to function as a MIoT prototype with respect to documentation, community support and sensor integration, in addition to fit current, if any, technology baselines?

    - R3: What constitutes a viable approach to a wearable prototype, when emphasis is on low cost, ease of availability and using available civilian technologies?

1.7 Thesis outline

The remainder of this thesis is organized as follows:

Chapter 2: Background and Motivation

Provides an insight into military decision making with respect to classic military doctrines, and puts IoT for military use into that context. Furthermore, architectural aspects in regards to IoT will be discussed, in addition to covering standardization, challenges, and enabling wireless technologies.

Chapter 3: State of the Art and Related Work

Covers target scenarios and critical issues surrounding IoT for military applications, which embeds previously conducted surveys, reviews, and field experiments related to MIoT use cases.

Chapter 4: Design

Describes the MIoT system design based on the findings from Chapter 2 and the technical considerations found in Chapter 3, in addition to the findings from the first set of interviews.

Chapter 5: Implementation and Evaluation

Describes the technical implementation and evaluates its performance through a second set of interviews by using a simulated network of nodes behaving in the same manner as the physical prototype.

Chapter 6: Conclusion

Concludes the work.

Chapter 7: Future Work

Outlines a proposition for future work.

Chapter 2

Background and motivation

In this chapter we will first discuss military decision making in the context of Command and Control (C2) process models, which is the core driving factor for enhanced or improved information systems in military organizations. In the context of military doctrine, we will discuss the role of IoT in such process models. Next, we discuss IoT architectural aspects and standardization, before covering enabling communications technologies. Low Power Wide Area Network (LPWAN) technologies are particularly relevant to this thesis, and such approaches are studied in-depth, laying the foundation towards constructing an IoT system.

2.1 Military decision making

Modern warfare has for the last few decades taken huge innovative steps towards streamlining combined-arms operations. Whereas procedures and doctrines with respect to maneuver haven't changed much since World War 2 and the Cold War era, C2 has seen a surge in battlefield intelligence and information acquisition technologies, providing a massive quantity of information across the military domains (land, sea, air, space, and cyberspace) for the decision makers. Due to various studies and innovations, the term C2 has seen many derivatives;

- Command, Control, and Communication (C3)
- Command, Control, Communication, and Intelligence (C3I)
- Command, Control, Communication, Cyber, Intelligence, Surveillance, and Reconnaissance (C4ISR)

In combat operations decision-making processes, the Observe-Orient-Decide-Act (OODA) loop, introduced by military strategist Colonel John Boyd (Boyd 1987), is a concept commonly used at operational levels which identifies the distinct stages a military commander will cognitively iterate from the moment he gains information until he acts on that information. In the face of the modern battlefield, it is imperative that the OODA loop is as short as possible through gaining information dominance over

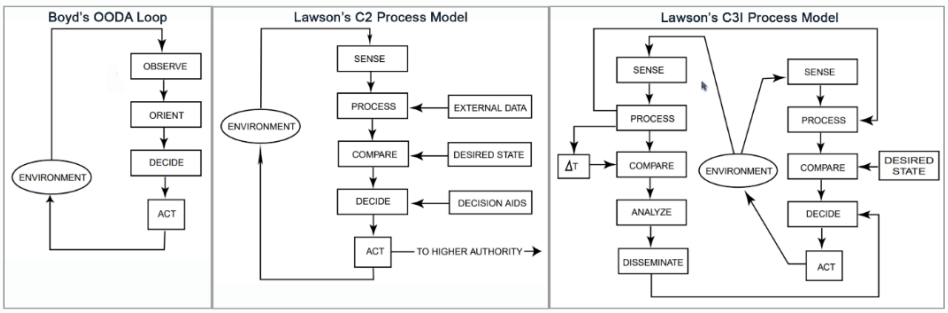


Figure 2.1: The OODA loop and C2 process models (credit: (Russell and Abdelzaher 2018))

an adversary, and ultimately making it into effective military decision-making.

2.1.1 Information and process models for C2

Boyd, originally a fighter pilot during the Korean War, described the OODA loop in the context of a pilots perspective while engaged in aerial combat. However, in general terms, parallels can be drawn from its core principle through theoretical process models aimed towards C2. One such model is Lawsons C2 process model which describes in generic terms the process of sensing and processing information, including an attribute for the desired state to be compared with the outcome after the received information has been processed, which ultimately lays the foundation for the decision making (Brehmer 2005), as seen in the middle of Figure 2.1. The model can be extended to C3I, which involves the additional communication and intelligence elements, where the latter is depicted as the counter-clockwise flow of the right-most diagram in Figure 2.1. The significant element in this particular model is ΔT, which is described as a projection in which from a given time t_0 (the point in time an event was recorded), an appropriate response to the sensed information must be accomplished before a given time t_{p^0} (preempted response to said event). The importance lies with the *Environment*, which includes any adversaries that may sense our activities, thus requiring a timely response to virtually any event in the battle space (Lawson 1981; Russell and Abdelzaher 2018). Intelligence in this case could involve traditional methods such as Signals Intelligence (SIGINT), Imagery Intelligence (IMINT), and Human Intelligence (HUMINT), but also new technologies aimed primarily towards industry and smart city development through intelligent sensing devices deployed at a large scale.

2.1.2 Network-Centric Warfare

Traditionally, so-called Platform-Centric Warfare (PCW) has dominated military doctrines in which the platform (i.e. individual soldiers, tanks, war vessels, fighter aircraft, etc.) is both the sensor, actuator, and often also

the decision maker. With very little SA other than they themselves are able to perceive in the heat of battle, they rely on information provided to them by higher commanding elements. With the emergence of the Information Age[1], it becomes necessary to find ways to leverage information in order to gain the advantage in the face of a modern and capable adversary in the battle space. This was the foundation for the introduction of Network-Centric Warfare (NCW). Based on Admiral William Owens' concept of a system-of-systems (Owens 1996), NCW aims at establishing a shared SA between geographically dispersed entities through the use of communication links, thus gaining information advantage and effectively increasing responsiveness, lower risks, lower costs, and increase combat effectiveness (Alberts, Garstka and Stein 2000). This description for an inherently information-driven doctrine can largely be transferred to both the MIoT concept and on Lawsons C3I model for decision-making.

2.2 Architectures and standardization

A generic IoT pipeline can be viewed as a variant of the generic n-tiered architecture, in which each layer communicates and exchanges data with the one neighboring it. The Cisco IoT reference model (Cisco 2014) can also be used in the same manner, which explains in greater detail how data moves and transforms throughout an IoT implementation. However, standards for generic IoT architectures with detailed specifications are currently an active research topic being conducted by a large group of standardization organizations worldwide (Next-Generation Internet of Things (NGIoT) 2020). One such example is the IEEE Standard for an Architectural Framework for IoT (conforms to the ISO/IEC/IEEE 42010:2011 standard for systems and software engineering with respect to architecture description (ISO 2011)), which describes a reference model that defines relationships among various IoT verticals and common architecture elements (IEEE Standards Association 2016a). In terms of military applications, a framework model nicknamed *IoTNetWar architectural framework* has been proposed which describes a MIoT system as a four-layered architecture (Ray 2015), including prospected technologies towards realization, as shown in Figure 2.2. Other general architecture frameworks, such as the Department of Defence Architecture Framework (DoDAF) (U.S. Department of Defence 2010) based NATO Architecture Framework Version 4 (NAFv4) (North Atlantic Treaty Organization (NATO) Architecture Capability Team, Consultation, Command & Control Board 2020), can be used to describe full-fledged system architectures in the military domain. However, at the time of writing, there is no one-size-fits-all standardization or reference architecture for a MIoT-specific implementation. We will therefore make use of the *IoTNetWar* reference model in this thesis due to its simplistic nature.

[1]The Information Age is commonly known as the historical period beginning in the mid-20th century in which the modern world shifted from an industrial society to an information society.

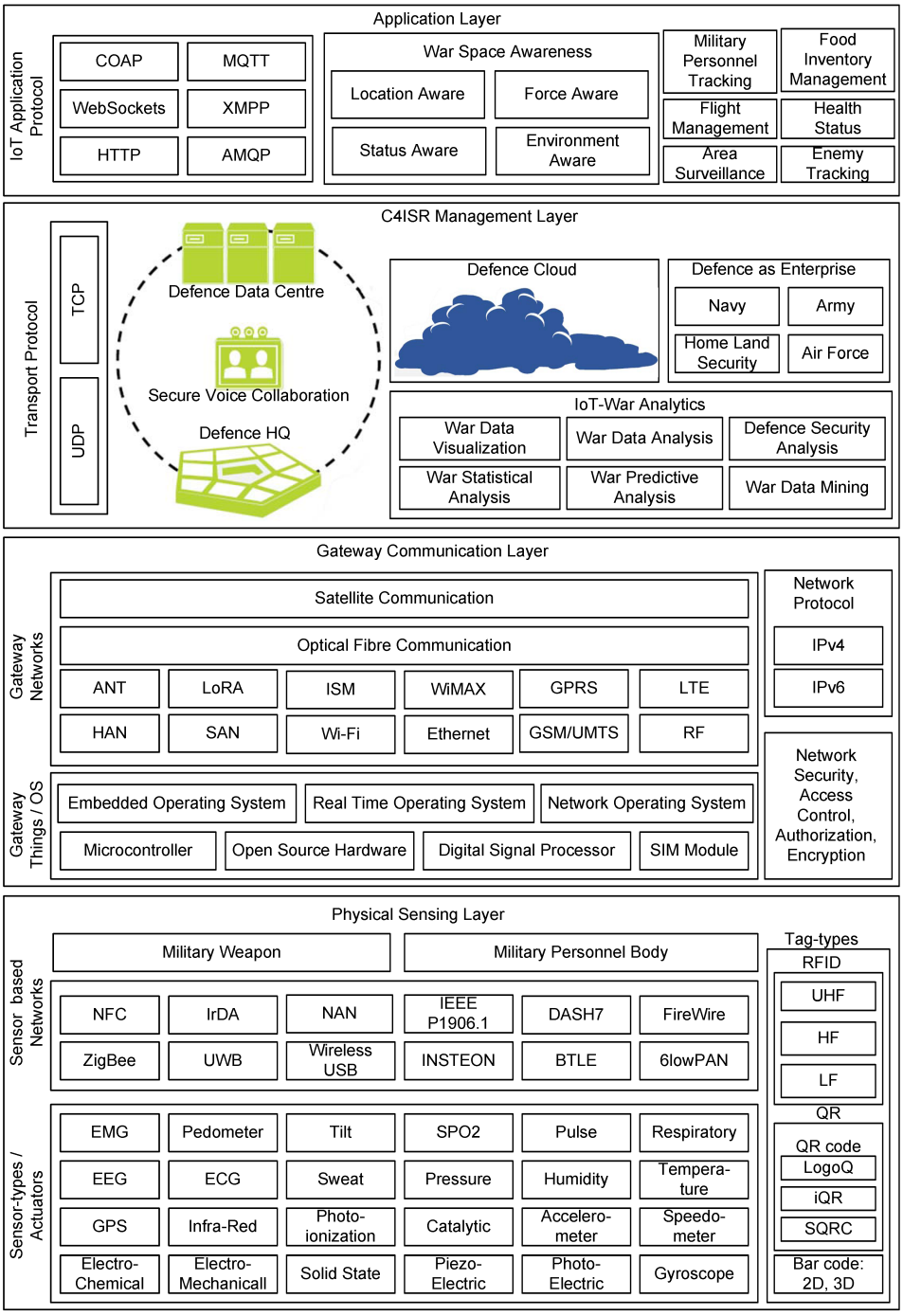


Figure 2.2: IoTNetWar architectural framework (credit: (Ray 2015))

2.3 Challenges

Before we look into the challenges we face for a MIoT implementation, we need to have a clear understanding of what challenges a commercial IoT system may carry, without the added complexity of battlefield environments, tactical communication systems, military procedures and culture, and so forth. Thus, we will initially consider the challenges and issues related to a commercial IoT system with respect to large scale sensor deployments.

2.3.1 Data volume and heterogeneity

IoT systems deployed at a large scale, such as smart cities, tend to generate massive quantities of data, often in a multitude of formats using a variety of devices and vendors. If such information-rich data streams should carry any meaning, depending on the organizational structure of the viewing audience, it must be filtered and aggregated in accordance with the required level of granularity and detail in order to provide grounds for action or decision-making. The emergence of data science principles such as Artificial Intelligence (AI) and Big Data are already being deployed at a large scale for data extraction, filtering, aggregation, and analysis for the purpose of enabling timely and accurate decision making in accordance to the business model it supports. Commercially, targeted advertisement and dynamic marketing in accordance with customer habits are common use cases for increased revenue (Marr 2020).

2.3.2 Communication protocols and operational lifetime

In terms of wireless communications, a number of technologies can be utilized in various use cases involving IoT today, where the IoT-system may simply use the available technology where it resides. However if the idea is to enable location-agnostic sensor deployment, they need to be able to communicate over a wireless protocol that can achieve long distances. Taking well-known protocols into consideration, we see in Figure 2.3 that there is a trade-off between throughput, power consumption, and effective range (Sourmey 2020). Cellular technologies provide both high throughput and range, but consume a lot of energy, while RFID is attributed with low capabilities over the whole baseline. The LPWAN boxes indicate an emerging paradigm within wireless communication protocols which will be central for this thesis, and illustrates a compromise between the above-mentioned attributes for wireless technologies.

Use cases for IoT in battlefield environments, as we will discuss in Chapter 3, include among others asset tracking and remote sensing, all of which more often than not require very long communication distances. This is where commonly known wireless technologies have a significant weakness. Of the ones mentioned in Section 1.1, the longest practical distance that can be achieved is 100 meters with ZigBee (Mukherji and Sadu 2016), if we exclude cellular technologies. There are also options to

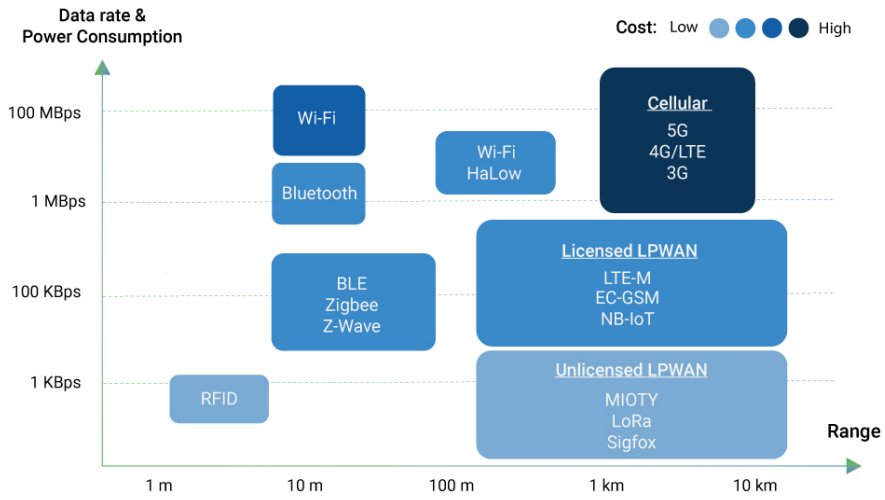


Figure 2.3: Wireless protocols throughput and range (credit: (Sourmey 2020))

achieve much longer distances with Wi-Fi by means of directive antennas, but because of the weak signal penetration in IEEE 802.11 (using 2.4GHz), it becomes prone to severe attenuation with physical objects between the signal source and the end node.

By scattering self-maintained sensors over a large area, we discover another critical issue: power consumption and battery capacities. Our MIoT network cannot be deployed based on the location of the nearest power outlet, and we cannot always rely on civilian infrastructure to achieve connectivity, especially in disaster-ridden areas. Therefore, we need to be able to establish self-maintained and mobile ad-hoc networks with the ability to achieve long range and a satisfactory throughput. To overcome these obstacles, we will consider LPWANs through emerging technologies like Long Range (LoRa), Sigfox and Narrowband IoT (NB-IoT), which will be described in Section 2.4.

2.3.3 Security and privacy

At the time of its introduction, IoT as a paradigm was not concerned with security measures as the overall goal was to connect everything and anything, without any real consideration towards the threat landscape. According to the Open Web Application Security Project (OWASP), the top 10 IoT vulnerabilities that define IoT application attack surfaces, are listed as follows (Messler et al. 2018):

- **Weak, guessable, or hardcoded passwords:** These can be easily brute forced, are publicly available, or cannot be changed. This

vulnerability include backdoors in firmware or client software that grants unauthorized access to deployed systems.

- **Insecure network services:** Unneeded or insecure network services running on the device itself which may compromise confidentiality, integrity, or availability or information.

- **Insecure ecosystem interfaces:** Insecure web, backend Application Programming Interface (API), cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components.

- **Lack of secure update mechanisms:** This includes lack of firmware validation on the device, unencrypted delivery of updates, lack of anti-rollback mechanisms, and lack of notification of security changes due to updates.

- **Lack of insecure or outdated components:** Use of deprecated or insecure software components/libraries that could allow the device to be compromised.

- **Insufficient privacy protection:** Personal information stored on device or in the ecosystem that is used in an insecure manner.

- **Insecure data transfer and storage:** Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.

- **Lack of device management:** Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.

- **Insecure default settings:** Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.

- **Lack of physical hardening:** This could allow potential attackers to acquire sensitive information that can facilitate for future remote attacks or take local control of the device.

Privacy-wise, we have to consider not only the generated sensor data, but also the embedded metadata which may carry significant details about the owner, which could compromise their privacy in the event of information leakage (Covington and Carskadden 2013). Often, this involves physical locations, device or hardware data, and timestamps, usually embedded in the data stream.

2.4 LPWAN alternatives

Primarily, there are three main competitors for large scale IoT deployments, namely Sigfox, NB-IoT, and LoRa, of which the latter is the focus for this thesis due to its flexible transmission implementation and local deployment model, as we will see in Section 2.4.4. We will briefly cover the core aspects surrounding Sigfox and NB-IoT before we continue with a closer look at LoRa and LoRaWAN.

2.4.1 Sigfox

Sigfox was founded in Toulouse, France, in 2010, and is both a technology company and a LPWAN network operator with partnerships with a number of various operators. As shown by the high-level network architecture in Figure 2.4, it is end-to-end focused, utilizing proprietary base stations with cognitive Software Defined Radios (SDRs) operating in the license-free Industrial, Scientific, and Medical (ISM) bands (see also Appendix A) (Barreiro et al. 2018). Over RF, it uses Binary Phase Shift Keying (BPSK) modulation, which is a modulation scheme that encodes the transmission by altering the phase of the sinusoid based on the message bits (θ = 0°for binary 1 and θ = 180°for binary 0), using 100 Hz of total bandwidth. This comes with certain trade-offs: It has very low noise levels, low power consumption, and low-cost antenna design, but has a maximum throughput of only 100 bits per second. To ensure that the base stations receives the messages, the end-nodes will transmit the same message multiple times using a set of channels in the assigned spectrum. The base stations will attempt to listen to all of the specified channels simultaneously, thus ensuring that the message will be received on at least on of them.

2.4.2 NB-IoT

NB-IoT is a technology specified in Release 13 of the 3GPP (3rd Generation Partnership Project (3GPP) 2016), a unification of telecommunications standards, in June 2016. Its main difference from LoRa and Sigfox is the frequency band in which it operates, namely 700 MHz, 800 MHz, and

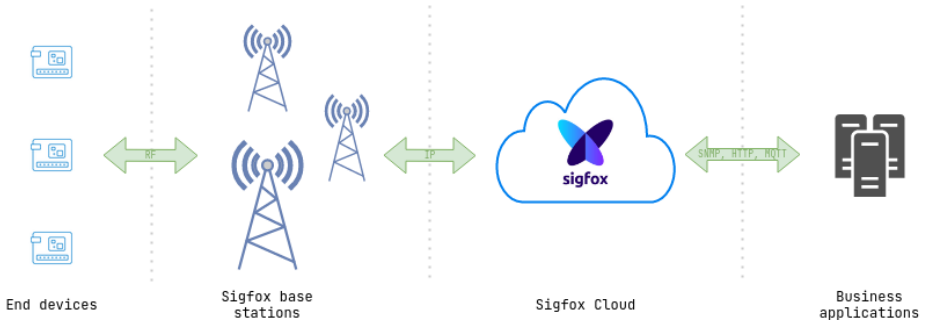


Figure 2.4: Sigfox high-level architecture

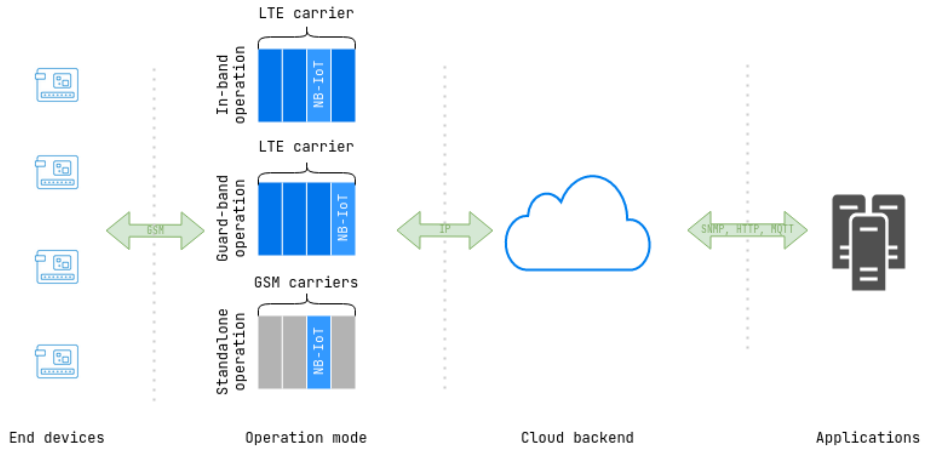


Figure 2.5: NB-IoT high-level architecture

900 MHz, all of which are licensed and in use by GSM and LTE. The protocol itself is also based on LTE, meaning it can operate on LTE-ready devices, but with limitations on down- and uplink speeds (200 kbps and 20 kbps, respectively) due to the reduction of the functionalities already present in the existing LTE protocol down to a minimum. As shown in Figure 2.5, it works by occupying a resource block in LTE transmissions, which corresponds to 180 kHz of bandwidth. This resource block is then assigned to a given channel within the LTE band, either in resource blocks within an LTE carrier (in-band operation), in unused blocks within an LTE carrier's guard-band (guard band operation), or stand alone without any neighboring GSM carriers. The data contained in the packets are then pushed upstream for processing before finally consumed by user applications.

2.4.3 LoRa

LoRa is a proprietary radio transmission technology that first was introduced by Cycleo, based in France, now owned by Semtech Corporation. The transceiver chip is denoted as Semtech SX12xx, and is either sold as an integral part on embedded systems or Microcontroller Units (MCUs)

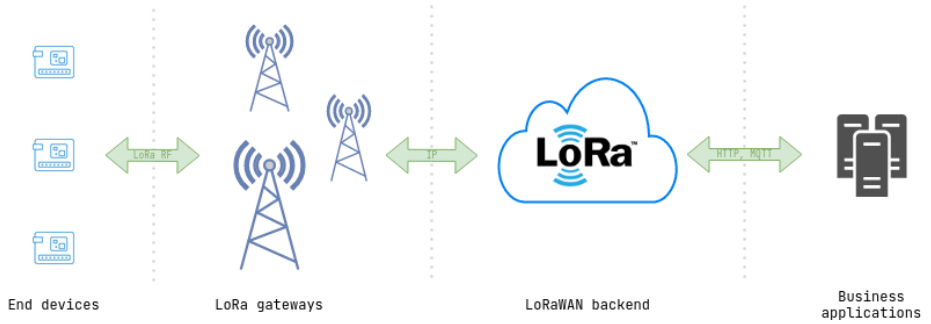


Figure 2.6: LoRa and LoRaWAN high-level architecture

or as a standalone chip breakout. Over RF, LoRa uses Chirping Spread Spectrum (CSS) (Semtech Corporation 2015), a modulation technology that provides high resilience and robustness in noisy or challenging RF environments (Springer et al. 2000), with which the LoRa implementation offers some flexibility for the users by adjusting transmission parameters. For a brief overview of LoRa signal encoding, see Appendix B.

Like Sigfox, LoRa too operates in ISM bands, and thus is also constrained by the same transmission limitations. As shown in Figure 2.6, its high-level architecture is very similar to that of Sigfox, where the data pipeline is end-to-end focused. However, the difference lies in the RF implementation and the deployment model. Infrastructure such as base stations for supporting a LoRa-connected network can be bought commercially from a number of vendors, and the backend component can either be supported through cloud providers, or it can be self-hosted using open source solutions. This leaves the maintenance responsibilities to the network owner, in contrast to a Sigfox or NB-IoT network where maintenance responsibilities lie with the service providers.

2.4.4 Performance comparison

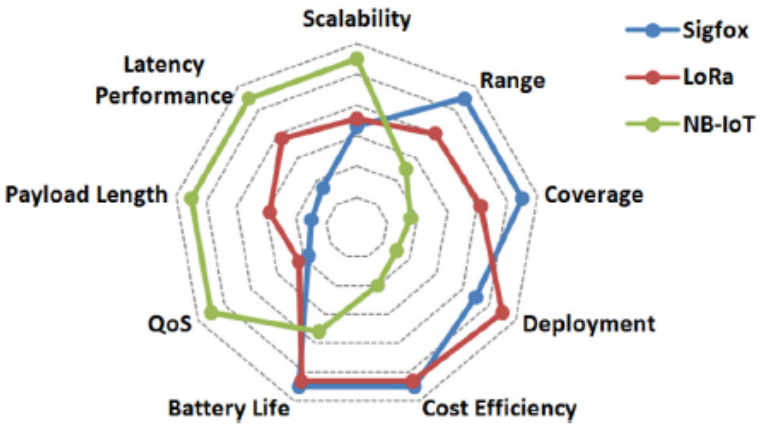


Figure 2.7: Advantage levels between LoRa, Sigfox and NB-IoT (credit: (Mekki et al. 2019))

In order to determine the most suitable technology for a MIoT deployment, we compare Sigfox, NB-IoT, and LoRa based on their capabilities in the face of the challenges described in Section 2.3. As shown in Figure 2.7, we see that LoRa and Sigfox is quite similar in theoretical performance, while NB-IoT would outperform the others in terms of latency and Quality of Service (QoS). However, LoRa has a significant advantage for military applications due to its local deployment model, in addition to relying on a resilient and flexible transmission technique (Mekki et al. 2019).

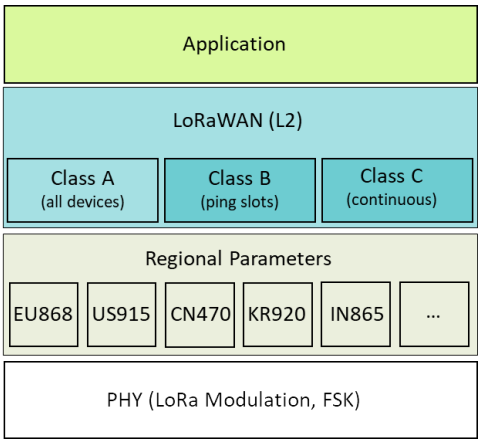


Figure 2.8: The LoRaWAN stack (credit: (LoRa Alliance 2020a))

2.5 LoRaWAN

Whereas LoRa is the protocol which operates on the physical level (commonly denoted LoRa Physical Layer (PHY)), LoRaWAN is the protocol which operates on the application level (commonly denoted LoRa Medium Access Control (MAC)). Essentially, it is a LPWAN manager for LoRa-enabled devices which covers the entirety of the LoRa pipeline. LoRaWAN networks can be logically viewed as a two-part technology stack consisting of LoRa PHY and LoRa MAC, where LoRa PHY handles the radio transmissions by sending the modulated signal over the air, and LoRa MAC specifies the transmission modes in addition to handling input and output to and from the application, as shown in Figure 2.8. In the following sections, we will dive into the LoRaWAN Regional Parameters (LoRa Alliance 2020b) which describes general constraints for LoRa-usage in specific geographic areas, LoRa Link Layer Specification version 1.0.4 (LoRa Alliance 2020a), which at the time of writing is the latest protocol specification in active use, and LoRaWAN Backend Interfaces 1.0 specification (LoRa Alliance 2017) which describes the standard interfaces and message flow among the LoRaWAN backend components.

2.5.1 Topology

In its simplest form, a LoRaWAN network is a star topology, where a number of nodes communicate with a single gateway, which in turn forwards the uplink messages to a single LoRaWAN Network Server (LNS). However, real world deployments would normally deploy multiple gateways in order to maximize area coverage as this won't realistically be achieved with the use of only one gateway. Multiple gateways may receive uplink messages from the same node, where the receiving LNS retains the one with the best Received Signal Strength Indicator (RSSI)[2] level and

[2]RSSI is the received signal level measurement for a device, which indicates how well it is able to "hear" other signals, generally from an access point or a router.

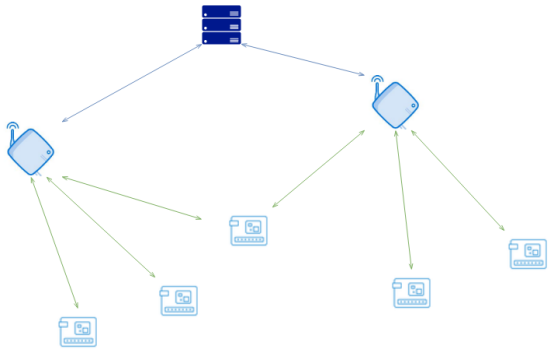


Figure 2.9: Star of stars network topology

deletes the rest. Thus, a LoRaWAN network is a star-of-stars network topology, as shown in Figure 2.9, where one node is able to communicate with two gateways, which in turn forwards to the same LNS.

2.5.2 LoRa radio transmissions

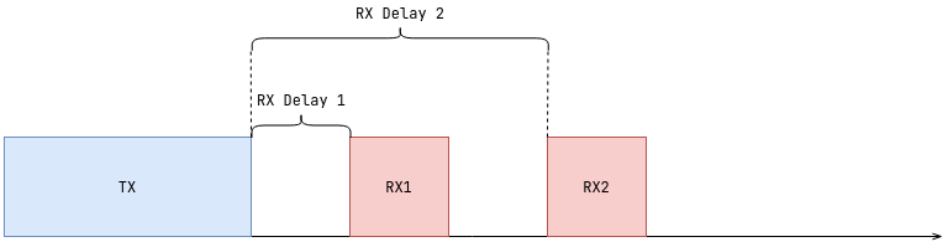
Transmission classes

LoRaWAN specifies three transmission classes which can be implemented based on the intended use cases. Class A, commonly referred to as “Aloha", will first transmit a message (referred to as the uplink, i.e. from end node to the gateway. LoRa messages will be explained in detail in Section 2.5.3) followed by two short receive windows (referred to as the downlink, i.e. from gateway to end nodes), in which for a given time, the device will listen for incoming messages, as shown in Figure 2.10a.

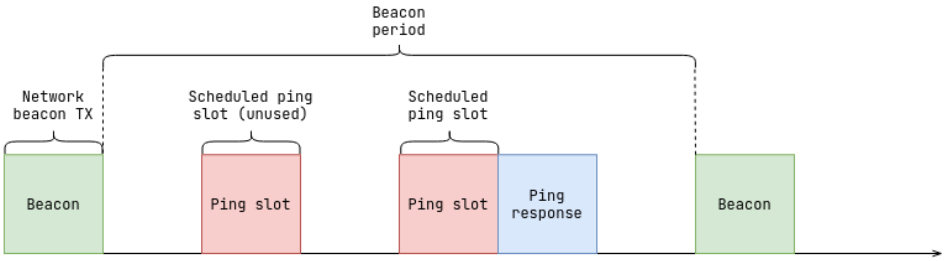
Class B, commonly referred to as “Beacon", extends class A reception rules with additional receive slots (referred to as ping slots), which are scheduled by the gateway by transmitting a synchronization-message (the beacon) to the device, thus enabling the gateway to know when a device will be listening for potential downlink messages, as shown in Figure 2.10b.

Class C further extends the reception slots to a near-continuous listening state, where it will only not do so when transmitting. As it “inherits" the reception slots from class A, it simply adds its own reception slots named `RXC` in places where class A would simply leave the radio module inactive, as shown in Figure 2.10c.

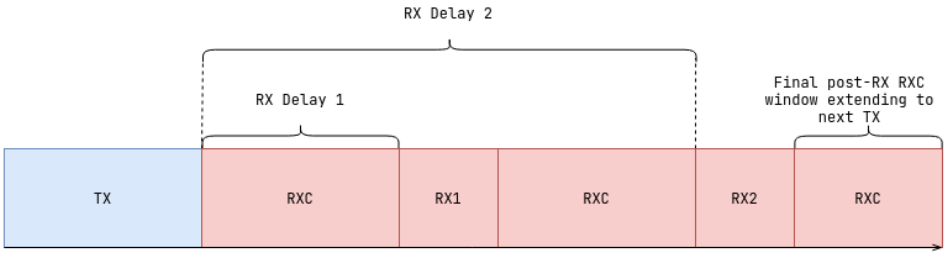
The choice of transmission class naturally depends on the implementation and domain logic of the sensor network, but it is also a matter of energy- and latency tradeoff. Class A have the lowest power consumption, but due to its narrow reception window it also has the highest latency among the classes. Conversely, class C have the lowest latency and highest level of energy consumption due to its always-on listening state. Commonly, the transmissions can be time-based in which sensor data is sent at a given time interval, or alternatively event-based in which triggers or interrupts on the device decides when to transmit sensor data.



(a) Class A



(b) Class B



(c) Class C

Figure 2.10: LoRa transmission classes

2.5.3 Message and frame format

Uplink and downlink

The LoRa terminology distinguishes between two types of messages; uplink and downlink. Uplink messages are broadcasted by the end-nodes using frequency hopping between the assigned channels to be received by one or more gateways, which in turn forwards them to the appropriate network server. The LoRa radio module “wraps" the payload with a preamble consisting of eight up-chirp modulated symbols followed by a synchronization signal consisting of two down-chirp modulated symbols to indicate that a transmission is about to start, in addition to inserting necessary headers and an optional Cyclic Redundancy Check (CRC), a method for detecting transmission errors, as shown at the top in Figure 2.11.

Downlink messages originate from the network server and are transmitted via the gateway to a specific node. One major difference between the uplink and downlink message structure is the lack of payload integrity checks for downlink messages, since the design principle is to keep messages as short as possible.

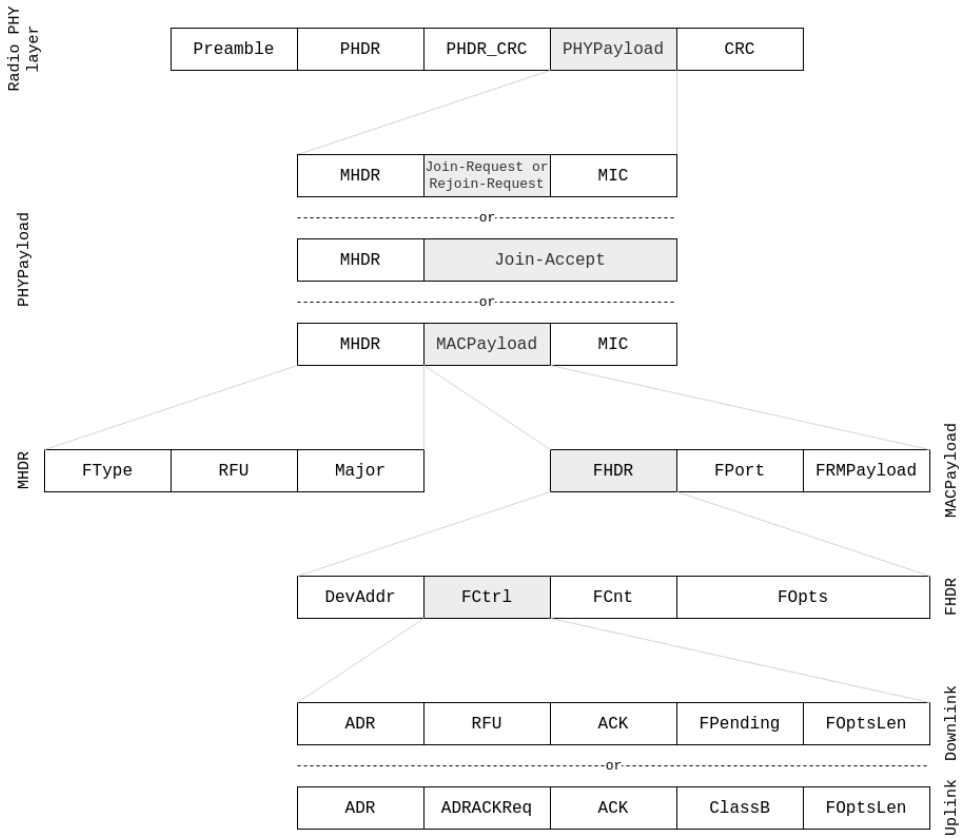


Figure 2.11: LoRa message format

LoRa messages can also carry the `confirmed` or `unconfirmed` property, which means whether the intended recipient should acknowledge successfully received messages or not.

LoRa message format

All LoRa messages carry a PHY payload, whether it is an uplink or a downlink message. The PHY payload consists of a single-octet MAC header (`MHDR`) containing information about what kind of message it is, followed by the variable-sized MAC payload, and ending with a 4-octet Message Integrity Code (MIC). The MAC header specifies the type of message through the `FType` field, which is a set of 8 different MAC messages types, as listed in Table 2.1. The MIC contains a value that is calculated using all the fields in the message, that is, the `MHDR`, `FHDR`, `FPort`, and `FRMPayload` (which must be encrypted prior to the MIC being calculated), which is used by the receiver to verify that the contents of the message haven't been tampered with during transmission.

The MAC payload field can be substituted with a Join-request or Rejoin-request, or a Join-Accept in which the MIC field is encrypted together with the payload and thus does not exist as a separate field in that case. It further consists of a Frame Header (`FHDR`), the frame port, and the frame payload,

Table 2.1: MAC message types

FType	Description
000	Join-request
001	Join-accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	Rejoin-request
111	Proprietary

where the latter two are optional. In other words, a frame with a valid `FHDR`, and no other fields set or equal to zero, is a valid frame. However, if the frame payload is not empty, the port field must be set with a value between 1 and 223. The port values of 0 and 224 is reserved for pure MAC-command messages and LoRaWAN MAC layer test protocol, respectively.

The `FHDR` contains the address of the the end-node, a single octet frame control (`FCtrl`) field, a 2-octet frame counter, and up to 15 octets of frame options used for MAC commands. If the `FOpts` field is present, it should be encrypted using the Network Security Encryption Key (`NwkSEncKey`), which will be discussed further in Section 2.5.6.

The `FCtrl` field embedded in the `FHDR` is structured depending on the direction it is going (e.g. uplink or downlink). It is used to set Adaptive Data Rate (ADR), confirmed message, and whether or not there is more data pending to be transmitted by the network (thereby requesting end-nodes to open a receive window as soon as possible by sending another uplink message).

2.5.4 Throughput and packet sizes

As LoRaWAN is designed as a long range protocol using low powered transmission output, it has limited throughput with a theoretical baud rate

Table 2.2: LoRaWAN datarates for the EU868 band

Data Rate	Configuration	Bits/s	Max payload
0	SF12/125kHz	250	59
1	SF11/125kHz	440	59
2	SF10/125kHz	980	59
3	SF9/125kHz	1760	123
4	SF8/125kHz	3125	230
5	SF7/125kHz	5470	230
6	SF7/250kHz	11000	230
7	FSK	50000	230

ranging from 0.25 kbps to 50 kbps, depending on the radio module in use (see also Appendix E). The throughput is largely decided by the Data Rate (DR), a combined-property configuration between the Spreading Factor (SF) and bandwidth which specifies the maximum potential throughput in bits per second. The SFs range from 7 to 12, and the DR classes range from 0 to 14, of which 0 to 7 is shown in Table 2.2 where DRs for the EU868 band. The remaining DR classes are not shown here as they use different transmission techniques which are outside the scope of this thesis.

The DR is a matter of trade-off, as a higher SF provides better reception rates but at the expense of lower throughput and payload sizes, as indicated by the DR classes. Practical throughput rates and packet sizes will be covered in the literature in Chapter 3.

Note that LoRa radio modules support Frequency Shift Keying (FSK) modulation as well, which can provide a much higher throughput. However, LoRa CSS transmissions are much more resilient to electromagnetic noise and even negative Signal-to-Noise Ratio (SNR)[3], and can be reconstructed much easier at the receiver while also maintaining a similar data rate and link budget[4].

2.5.5 MAC commands

The LoRaWAN stack facilitates certain network administration operations through the use of MAC commands. These can be sent either embedded in the `FOpts` field, or in the `FRMPayload`, provided that the `FPort` field is set to 0.

A MAC command consists of a 1-octet sized Command Identifier (CID) followed by an optional command-specific sequence of octets. A list of available MAC commands is listed in Appendix D.

2.5.6 End-node activation

Personalization and activation

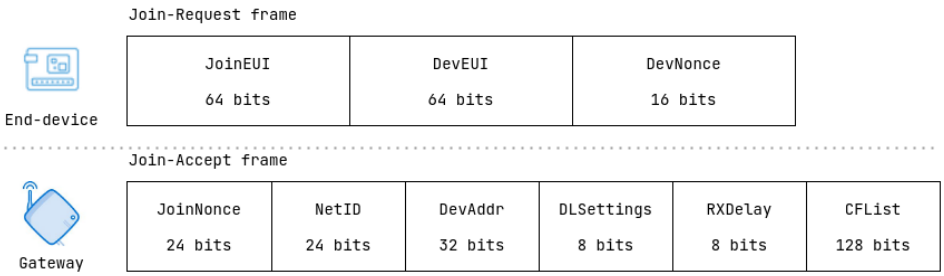


Figure 2.12: Join-Request and Join-Accept frame

[3]SNR is a ratio between the signal- and noise levels, where a positive SNR means the signal level is higher than the noise levels.

[4]Link budgets are an estimation of the received signal levels while taking into account both positive factors (e.g. antenna gain and directivity) and negative factors (e.g. path loss and fading).

Each node that wants to be part of the LoRaWAN network has to be personalized and activated through one of two different methods; Activation By Personalization (ABP) or Over-the-Air Activation (OTAA). Personalization means in this context to uniquely identify each individual node through unique identifiers, namely the Device Identifier (`DevEUI`) and a Join-Server Identifier (`JoinEUI`). Activation means to authenticate the nodes, thereby enabling data exchange between the node itself and the LoRaWAN backend.

When using OTAA, end-nodes should follow a Join-procedure before the activation process. The end-node has to provide the join request with a globally unique `DevEUI`, `JoinEUI`, and a 128-bit Advanced Encryption Standard (AES) key called the `AppKey`, which will be discussed later. The Join-procedure consists of two MAC frames - the `Join-Request` and the `Join-Accept` - which can be seen as part of the PHY payload in Figure 2.11. The `Join-Request` frame consists of the `DevEUI` and `JoinEUI`, followed by a `DevNonce` field, a zero-initialized, 2-octet sized nonce value which should be incremented for each power-cycle or Rejoin-request. If not, the Join Server will discard the Join-request since it keeps track of all the `DevNonce` values for each node, and it expects a consistent increment for each such request. The `Join-Request` frame is outlined at the top in Figure 2.12.

The Network Server responds with a Join-Accept frame if the end-node meets the requirements, which consists of a 3-octet sized Join-Server nonce (`JoinNonce`), a network identifier (`NetID`), an end-node address (`DevAddr`), downlink configuration settings (`DLSettings`), a delay between TX and RX (`RXDelay`), and an optional list of region-specific network parameters (`CFList`) for the Network that the node is joining. The `Join-Accept` frame is outlined at the bottom in Figure 2.12

The `JoinNonce` is a non-repeating value provided by the Join-Server which is used by the end-node to derive the two necessary session keys; the Network Session Key and the Application Session Key, which will be discussed further in the next section.

ABP does not utilize any session keys the same way OTAA does. Instead, the `DevAddr` and the two session keys are stored directly on the end-node, and is thus configured for specific networks. These keys cannot be updated unless when reconfigured manually, and OTAA is therefore the recommended approach for applications with higher security requirements.

Sessions

When an end-node has been personalized and activated in the network, it utilizes two separate sessions for continued communication with the Network and Application servers. These sessions each derive their own keys from the securely stored root key `AppKey`, namely the Network Session Key (`NwkSKey`) and the Application Session Key (`AppSKey`). The `NwkSKey` is used for identification of the device whenever it transmits an uplink message and for data integrity checks by calculating and verifying

the MIC. It is also used to encrypt and decrypt MAC-only data frames. The `AppSKey` is used for payload encryption and decryption. Data encrypted with the `AppSKey` is only integrity-protected over the air and not end-to-end. This is because the Network server can alter the data frames in transit (albeit without the ability to read the contents in plain text).

2.5.7 Backend

Three components make up the LoRaWAN backend; the LNS, the Application Server, and the Join Server, as depicted in Figure 2.13. These communicate over regular TCP/IP and ultimately facilitates for data consumption by system end-users, commonly through some web-based UI.

Network Server

The LoRa gateway is essentially just a stateless packet forwarder for all received LoRaWAN messages to the LNS. This server is a core-component of a full-fledged LoRaWAN deployment which terminates the MAC layer for the end-nodes connected to the network, and is the center of the star topology. Its responsibilities are end-node address checking, frame authentication and frame counter checks, acknowledgements, DR adaption, MAC layer request responses, uplink application payload forwarding to the appropriate Application Servers, queuing downlink payloads coming from any Application Server to any end-nodes connected to the network, Join-request and -accept message forwarding between the end-nodes and the Join Servers. Depending on the configuration of multiple LNSs in a deployment, the LNS can take one of three roles in a roaming setup: Home-, Serving-, and Forwarding Network Server. A brief explanation of roaming can be found in Appendix F.

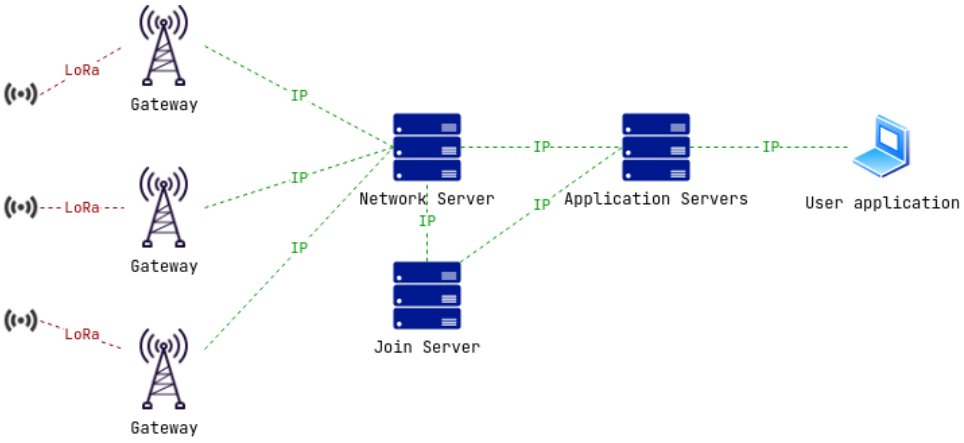


Figure 2.13: LoRaWAN backend components

Join Server

The Join Server manages the OTAA join-process, where the end-node specifies which Join Server it wants to use through the JoinEUI field of the Join-Request message. The LNS then routes this request to the approriate Join Server, where it subsequently takes over the join-procedure. Consequently, the Join Server is responsible for processing the Join-Request frames and generate the Join-Accept frames, in addition to perform the network and application session key derivations, as described in Section 2.5.6, and relays them to the appropriate LNS and Application Server.

Application Server

The Application Servers handles the received payload messages generated from the connected end-nodes and generates all application-layer downlink payloads. A LoRaWAN deployment supports multiple Application Servers connected to the same LNS, as well as an Application Server being connected to multiple LNSs, where the routing is handled by the LNS based on the DevEUI.

2.5.8 Security

As outlined in Section 2.5.7, the LoRaWAN protocol utilizes security in two levels; mutual authentication between node and gateway known as the join-procedure, and end-to-end application-level encryption. Both levels make use of the AES (National Institute of Standards and Technology - Federal Information Processing Standard (NIST-FIPS) 2001).

Each package in the network is secured using Advanced Encryption Standard-Counter Mode Encryption (AES-CTR), a mechanism that uses a monotonous counter to encrypt data streams, and a frame counter to avoid packet replay, as described in Section 2.5.3. The MIC field is computed using Advanced Encryption Standard-Cipher-based Message Authentication Code (AES-CMAC) to prevent package tampering (LoRa Alliance 2019; Cerrado, Fayo and Sequeira 2020). A logical representation of a LoRaWAN package with respect to security-wrapping can be seen in Figure 2.14.

Additionally, there exists some other measurements to further improve security. For instance, the AppKey may be hidden from the end user

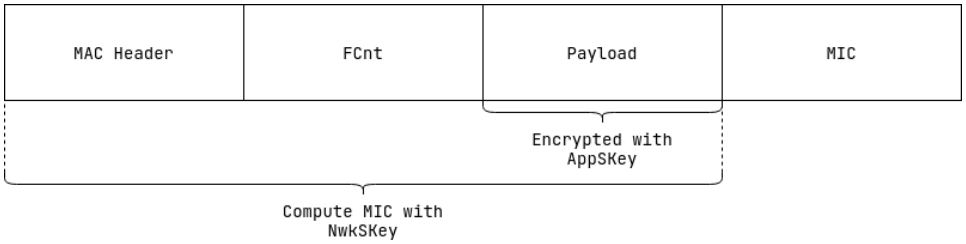


Figure 2.14: LoRaWAN package security

to avoid human error, and key storage on the devices themselves can be implemented using specialized hardware such as Secure Element (SE) (Global Platform 2018) or Trusted Platform Module (TPM) (ISO 2015).

2.5.9 Further reading

For convenience, the following list will point the reader to appendices which goes into further detail regarding the following LoRaWAN topics:

- LoRaWAN signal encoding: Appendix B.
- LoRaWAN ADR: Appendix C.
- LoRaWAN MAC commands: Appendix D.
- LoRaWAN hardware and MCU requirements: Appendix E.
- LoRaWAN roaming: Appendix F.

2.6 Summary

In this chapter, aspects of military decision making were introduced. Notably, the OODA loop and its importance in decision making was presented. The aim of the thesis is to leverage new technologies to improve situational awareness, and hence ultimately also decision making.

Next, technology challenges were presented, where security issues related to IoT and deployment of resource-constrained devices as part of sensor networks were the primary focus, which will be a core concern for future MIoT deployments. In this context, several commercial IoT technologies were discussed and compared in the face of a number of challenges for IoT deployments, which arguably is a subset of MIoT deployment challenges, and must therefore be considered when designing such a system. Concrete challenges and issues surrounding IoT for military use was not discussed here, as this will be the initial focus in Chapter 3.

Out of the prospected IoT communication protocols, LoRaWAN was deemed most suitable for pursuing in this thesis, due to the fact that it offers both long range communication possibilities and does not rely on existing infrastructure. In the next chapter, we will look into possible defence-related target scenarios, and specifically target how LoRaWAN can support such scenarios by investigating related work.

Chapter 3

State of the art and related work

In this chapter we will cover various surveys and reviews that identify military use cases for IoT. This is done to give the reader an overview of anticipated uses of IoT, of which arguably the topic of this thesis, wearables, falls within the outlined uses of MIoT, specifically as a sub-category of personal sensing. Then, operational requirements and critical issues are discussed, to establish constraints that are in effect in an operational setting. Also, recall from the previous chapter that LoRaWAN specifically was identified as a very promising protocol for long-range low power communications. Hence the last parts of this chapter provides the reader with an in-depth study of practical experiments in relation to LPWANs through the LoRaWAN protocol.

3.1 Target use cases for MIoT

Several reviews and surveys have been conducted which have identified several mission-critical use cases for IoT, i.e. use cases which are considered of utmost importance towards achieving success in the military domain, namely collaborative sensing, logistics and supply chain management, personal sensing, crowdsensing, fire-control systems, C4ISR, and exploitation of smart cities (Fraga-Lamas et al. 2016; Suri et al. 2016; W. A. Carter 2015), as depicted in Figure 3.1. In the following sections, these will be discussed further in terms of prospected use cases for future military operations.

3.1.1 Logistics and supply chain management

Currently, logistics and supply chain management is generally a time-demanding and manual task which largely involves direct human involvement, where inventory checks and resupply tasks are largely coordinated using voice communications over tactical radios. Only fairly recently have various Instant Messaging (IM) software and mail systems embedded in BMSs been utilized to conduct such tasks, which effectively just eliminates the use of pen and paper. By using highly pervasive sensors, inventory and

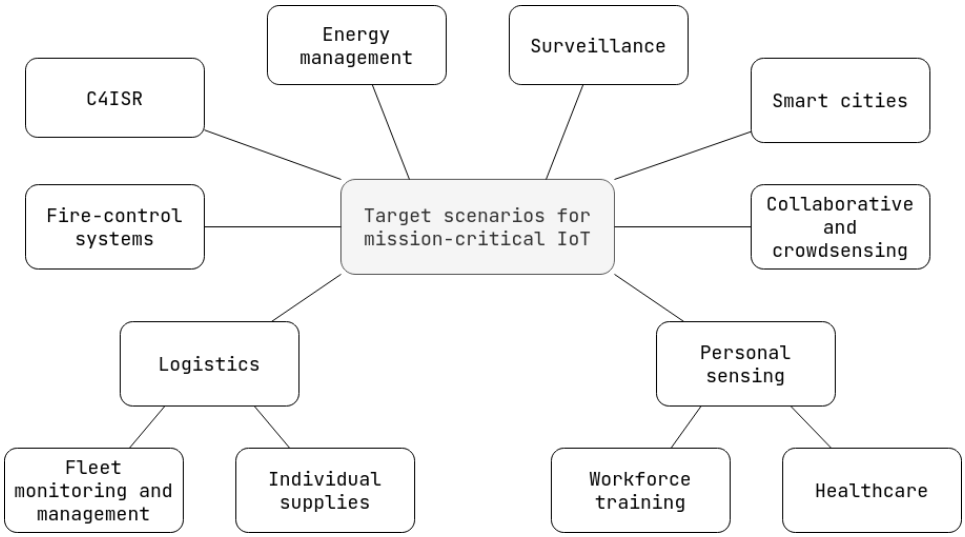


Figure 3.1: Target scenarios for MIoT applicability (credit: (Fraga-Lamas et al. 2016))

fleet monitoring can be conducted automatically (Suri et al. 2016; Fraga-Lamas et al. 2016; W. A. Carter 2015), which combined with data analytics and possibly even machine learning could be leveraged to predict the need for resupplies, and subsequently task the nearest logistics unit without involvement from commanding elements, possible even without the need for the ground force commander to file such requests.

3.1.2 C4ISR

C4ISR is a major factor for decision making by providing mission-specific data collection such as radar, video, infrared, and passive RF detection from a wide variety of platforms, such as UxVs (Unmanned Aerial/Ground/Surface Vehicles/Vessels), ground stations, and soldiers in the field. By integrating all of these data streams to a central operation centre, it builds a Common Operational Picture (COP) (Fraga-Lamas et al. 2016; W. A. Carter 2015). Following the principles of NCW as described in Section 2.1.2 and the importance of short OODA loops as outlined in Section 2.1.1, the same COP could then be disseminated to the troops on the ground, thereby increasing their local SA as well as building a common situational understanding across the whole military force.

3.1.3 Fire-control systems

High-precision munitions integrated with end-to-end sensor deployments enables fully automated responses to real-time threats, by enabling for mobile target tracking and in-flight redirection of missiles (Fraga-Lamas et al. 2016). This could also be expanded to ensure no friendly fire occurrences, by keeping track of the whereabouts of own forces and their movements.

3.1.4 Smart cities

Existing smart infrastructure, such as CCTV and traffic monitoring systems, could help augment SA for military deployments simply by interfacing their information and intelligence collection systems to existing infrastructure (Johnsen, Zieliński et al. 2018).

3.1.5 Energy management

Commercially, IoT has already been used to implement smart waste- and energy management, which could be used to cut day-to-day operational costs for the military organization. Apart from dimming lights and automatically adjusting room temperature, a pilot project conducted at Great Lakes Naval Station, USA, utilized machine learning to reduce energy consumption by combining weather data, energy consumption, comfort thresholds, and data collection from buildings, which showed a reduction of 20 to 30% in energy consumption (Mariani, Williams and Loubert 2015).

3.1.6 Surveillance

Surveillance in terms of ensuring military facility security is a topic where IoT may prove valuable. With the emergence of new, technological threats such as ballistic and hypersonic missiles, as well as drone swarms and autonomous platforms, it is believed that IoT may provide enhanced base defence and aerial surveillance through the use of advanced sensors (Rjaanes et al. 2020).

3.1.7 Crowdsensing

In segregated networks involving multiple actors, collaborative sensing may prove helpful to fill information gaps on lower levels (i.e. between collaborating nations, or otherwise disconnected units due to hierarchical structures or technical barriers) (Fraga-Lamas et al. 2016). Soldier wearables can be expanded to fit this use case by collecting data related to their immediate surroundings. Done on a large scale, this is referred to as crowdsensing, which is generally conducted in one of two ways; opportunistic or participatory. These can be distinguished as passive and active data collection, respectively, in which passive - or opportunistic - data collection is conducted much like the same way data is generated through soldier-worn sensors, whereas active - or participatory - data collection involves active user interaction. In Norway, crowdsensing experiments have been conducted as part of a Home Guard (HV) field training exercise, where the soldiers were equipped with Android phones installed with specialized software in order to provide increased SA (Pradhan et al. 2019).

3.1.8 Personal sensing

Individual sensing have been identified as a potential element for C2 improvement by drawing parallels to commercial wearable devices, such as the FitBit (FitBit Inc. 2021). It is suggested that soldiers can utilize biometric wearables to monitor for instance heart-rate, pulse-oxygen saturation, and respiration for monitoring of their physiological and physical state, in addition to a variety of inventory sensors in order to monitor their resources such as water levels, ammunition, and battery levels (Suri et al. 2016; Tortonesi et al. 2016; Fraga-Lamas et al. 2016; Johnsen, Zieliński et al. 2018). Put into an integrated system, this is effectively a soldier wearable system, a sub-category of personal sensing, which could provide both squad members in the field and commanding elements a near real-time evaluation of their status, thus effectively enabling for a remote patient- and inventory monitoring system, further decreasing the need for largely voice-based radio traffic.

3.2 Operational requirements and critical issues

3.2.1 Federated networks, interoperability and security

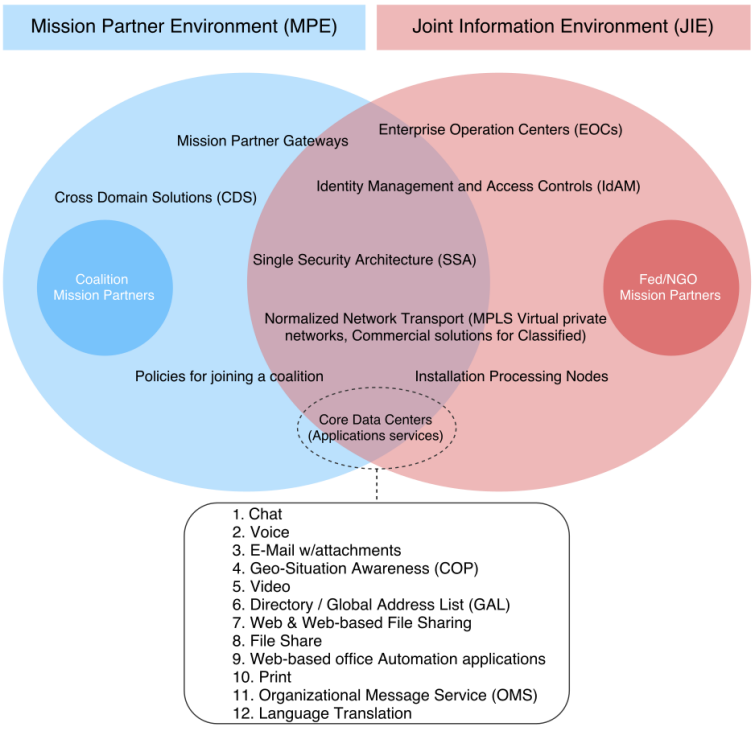


Figure 3.2: Conceptual design of the JIE infrastructure (credit: (Fraga-Lamas et al. 2016))

Military operations involving allied- and partner nations, in addition to multi-discipline military branches, problems concerning interoperability,

trust, and security arises. System heterogeneity and differing security clearance levels are formidable obstacles not only towards accomplishing interoperability between collaborating forces, but also in terms of data and intelligence sharing. Normally, collected intelligence needs to be manually released to requesting parties by the “owning" authorities, which usually resides at a higher level in the military organization. Thus, the request may have to pass several levels in the organization before it is finally released. Utilization of Service-Oriented Architecture (SOA) (W3C 2008) design using predetermined actor attributes for efficient data sharing has been proposed as a possible solution by exploiting well-defined interfaces and common messaging protocols (Suri et al. 2016; Fraga-Lamas et al. 2016), which addresses C4ISR-specific interoperability challenges.

To secure an environment involving such a complex structure, however, necessary security mechanisms needs to be in place so as to autonomously allow intended actors the access they need while preventing unauthorized access to malicious counter-actors. The U.S. DoD have proposed the Joint Information Environment (JIE) that comprises a shared IT infrastructure, enterprise services, and a Single Security Architecture Single Security Architecture (SSA). SSA combined with Identity Management and Access Control (IdAM) could provide the foundation for securely enabling information access and sharing among warfighters, including interfaces for a Mission Partner Environment (MPE) so as to be able to share data and intelligence across domains in a secure manner (Fraga-Lamas et al. 2016), as depicted in Figure 3.2.

3.2.2 Sensor platforms constraints

Sensing and actuating devices need to integrate well with existing platforms without adding additional complexity to the military mission. Whether the carrier is a soldier, vehicle, aircraft, or an unmanned asset, it should not change the behavior of the platform on which it is mounted. Rather, it should provide the platform with non-intrusive means for augmenting war-fighting capabilities. Thus, it is required to be physically small in size, energy-effective, and robust against both harsh usage and environments, which raises implementation challenges in terms of battery life and processing capabilities. At the time of writing, there already exists a number of military standards which describes various requirements for radio systems, such as power cell sizes, transmission capabilities, and ruggedness (MIL-STD 810G, MIL-STD 461F, MIL-STD-1275) (Fraga-Lamas et al. 2016) that a future, fully-integrated MIoT system should comply with.

3.2.3 Network capabilities

Following the principles of NCW in the context of an inter-connected military organization with possibly exponentially more connected devices, bandwidth must be considered a constrained and limited resource. As such, it must be utilized as efficiently as possible. Tactical radio networks often experience outage and low throughput reliability due to surrounding

electromagnetic phenomenons, cluttered spectrums, and even presence of hostile elements such as EW. Additionally, military units are constantly mobile, thus forever changing their local reception levels. Preparations for decentralization or partitioned network segments must therefore be in place, while limiting the usage of communication links down to a minimum, as low-capability networks may become overwhelmed with the vast amount of data being produced and transmitted (Fraga-Lamas et al. 2016).

3.2.4 Security, robustness, and reliability

In modern-day military scenarios, one must consider the EW capabilities for any given adversary, as EW is a major direct and indirect threat to any radio system. Generally, the EW threat can be classified either as Electronic Counter Measures (ECM) or Electronic Support Measures (ESM), where ECM is active measures such as jamming, replay, and deception, and ESM is passive measures such as interception and emitter location. Depending on the exact capabilities, a hostile EW unit would be able to incapacitate sensor systems, extract information from the intercepted transmissions, or geolocate transmitters for targeting purposes. Realistically, complete protective means against such threats are virtually non-existent, however they can be reduced by implementing so-called EPM. Such disciplines describe methods that generally lowers both the ECM and ESM threats by using Low-Probability of Interception (LPI)/Low-Probability of Detection (LPD) techniques; minimum transmission time, secure encryption, minimum transmission power, and spread spectrum modulation techniques.

These techniques must comply with the processing capabilities and battery capacities to that of the devices as well, considering the fact that cryptographic algorithms are generally computationally expensive. In addition, the node increase in a network is to be considered a proportional increase in attack vectors, thus requiring new measures to prove node integrity while taking node processing limitations into account. One proposed solution is integrity attestation functioning as a supplement to subject authentication (Fongen and Mancini 2015). Furthermore, the fragments of the electromagnetic spectrum in which tactical radios and enterprise network services communicate are often cluttered and congested, which affects the reliability and robustness of the communication links. This requires significant preparatory work to coordinate and allocate frequency specifications for all collaborating stakeholders. To make networks less reliant on pre-mission planning and more flexible mid-mission, cognitive radios and dynamic spectrum management have been proposed to automatically reconfigure devices to overcome bad conditions in the communications environment (Fraga-Lamas et al. 2016).

3.3 LoRaWAN range and coverage testing

To date, a number of field experiments focusing on finding maximum practical distances with which the LoRaWAN protocol can effectively work has been conducted. In most cases, this was measured by reporting position data from mobile sensors to a single, stationary LoRaWAN gateway (Johnsen and P. Ø. Puente 2018; Michaelis et al. 2019; Søndrol, Jalaian and Suri 2018; Jalaian et al. 2018; Wixted et al. 2016).

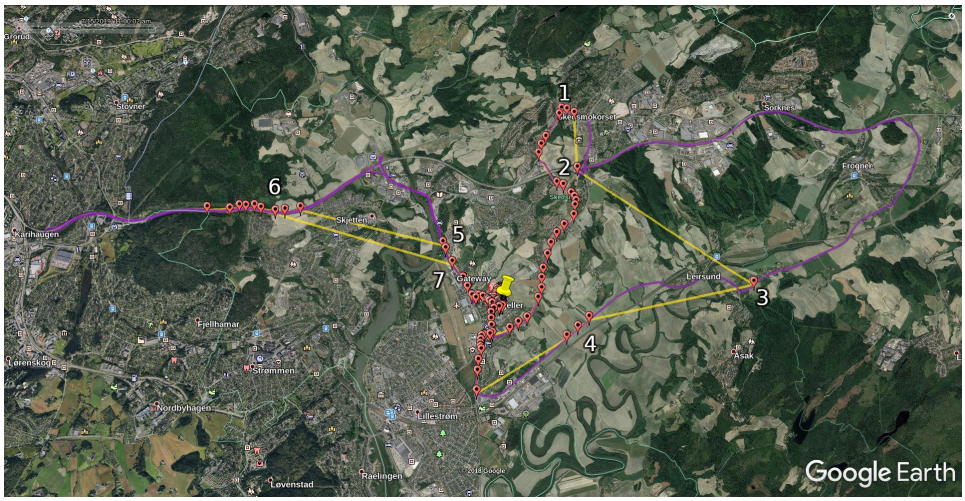
(a) LoRa range trace using DR1 (b) LoRa range trace using DR2

Figure 3.3: LoRa range traces from the experiement in Montreal, Canada (credits: (Michaelis et al. 2019))

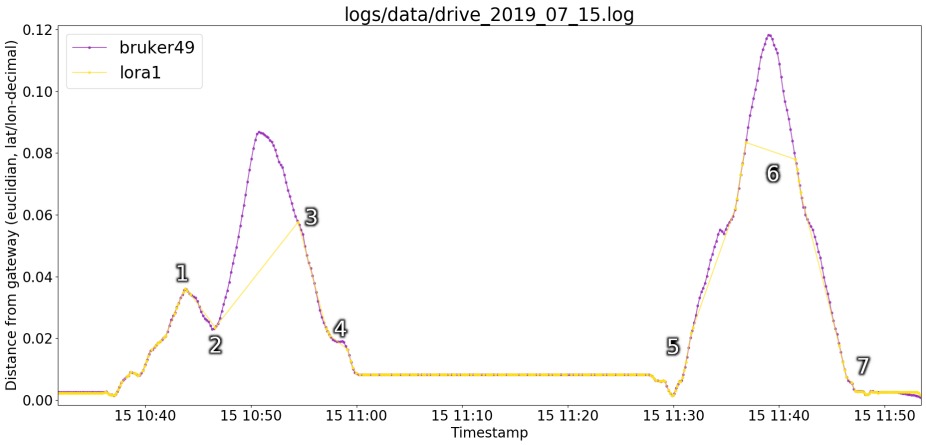
One of the experiments conducted in Montreal, Canada, achieved up to 5.5 km of distance between the device and the gateway. However, the package loss increased significantly once the distance reached 1 km and above. Additionally, the DR configuration played a significant role, where DR1 outperformed DR2 at all ranges (Michaelis et al. 2019), as seen by the green plots in Figure 3.3a showing reported locations back to the gateway, compared to the trace seen by the red plots in Figure 3.3b.

A range testing experiment conducted in Maryland, USA, achieved a range of up to 9.8 km, when the device had unobstructed Line-of-Sight (LoS) back to the gateway in a large, open area (Jalaian et al. 2018). When the device was behind solid structures, such as mountain ranges, the connectivity was lost despite a much shorter distance at 4.1 km. This experiment also shed light on an important issue, namely LoRaWAN packet size limits. Using DR0 in the US, the payload size could not exceed 11 bytes. Thus, the software for this particular experiment had to implement alternations between sending their on-board sensor data, namely position-, temperature-, and humidity readings.

In Kjeller, Norway, similar testing using multiple device setups achieved an effective range of 6 km using SF7 and 125 kHz bandwidth. Like with the range testing experiment in Maryland, this was made possible due to unobstructed LoS back to the gateway, as the connectivity was lost for some period when they headed back to Norwegian Defence Research Agency (FFI) facilities (Johnsen and P. Ø. Puente 2018), as shown in Figure 3.4a, where reported positions from the LoRaWAN devices are



(a)



(b)

Figure 3.4: FFI asset tracking experiment (credits: (Johnsen and P. Ø. Puente 2018))

indicated by the red markers. The connection losses can be seen as the straight yellow lines that corresponds with the ones seen in the graph in Figure 3.4b.

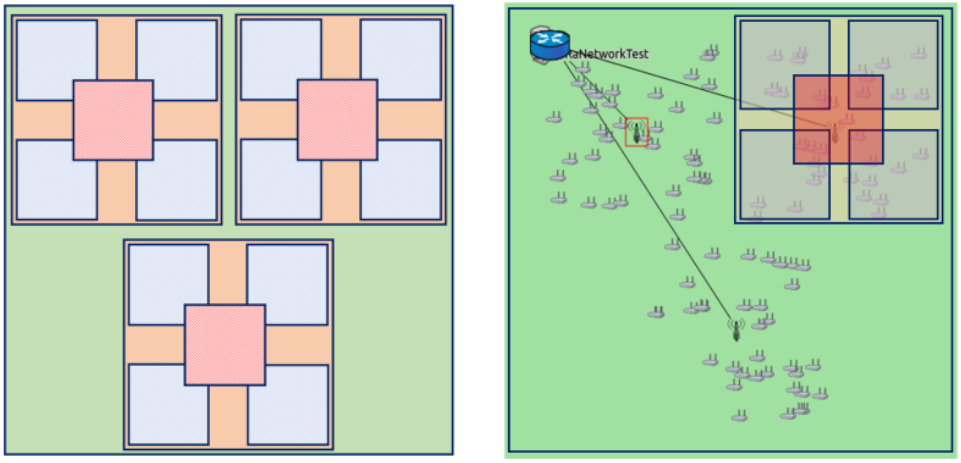
In Glasgow, Scotland, a sensor carried through a dense part of the city found the maximum effective range to be between 1.6 km to 2.2 km. The experiment found that topology played a significant role as the package reception rate rose from 42% to 70% when installing a second gateway for greater coverage (Wixted et al. 2016). The experiment also found the cellular backhaul link to be somewhat unreliable, as it was found that large blocks in time showed no connection for an hour or more, despite no mobile network outages had been advised. Although not confirmed, it is strongly believed that this was due to the unreliable UDP link between the gateway and the LNS, as continuous latency monitoring between the components using Internet Control Message Protocol (ICMP) pings

improved the connection rate to 95.5% as soon as the pings started.

3.4 Military applications for IoT in BMS

A study conducted in Brussels investigated the applicability for IoT in BMSs where the use cases involved perimeter patrols as part of facility security, where two soldiers were equipped with LoRaWAN tracking devices for the purpose of enabling a live location feed to check point personnel.

The experiment found that the check point personnel experienced enhanced SA as the exact location for the foot-mobile patrol was always known, where the tracking devices achieved ranges up to 1.5 km (Baeyens 2017). The same work found that the effective range for LoRaWAN devices could match voice-based radio systems currently in use, at about 6-7 km in urban environments, while using SF12 and half-wave dipole antennas.



(a) Logical hierarchical view of a military company

(b) Node placements in accordance with the military company

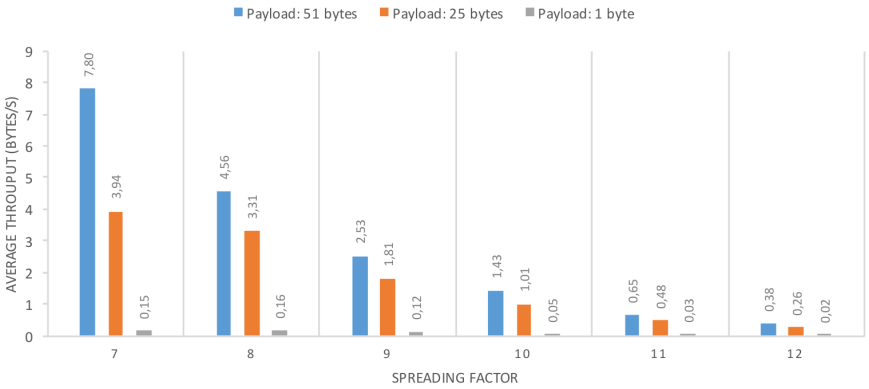
Figure 3.5: On-scale LoRaWAN deployment for a company-sized unit (credits: (Baeyens 2017))

The same study conducted a simulation to measure collision rates for a large scale deployment by deploying 105 sensors in accordance to a military hierarchy logically equivalent to a company-sized unit, as shown in Figure 3.5a. The deployment used a square-shaped area with a diagonal range of 30 km, in which the sensors remained static throughout the tests while applying variable gateway setups, as shown in Figure 3.5b. The results showed that one gateway per section (i.e. 15 gateways) had a reception rate of 97%, compared to a single company-wide gateway with a reception rate of just 20%, which would only be improved to 35% if variable SFs were used for each section. This is most likely due to the high collision rates between messages being transmitted at the same time with the same SF, in combination with the large distances between the gateway and the devices.

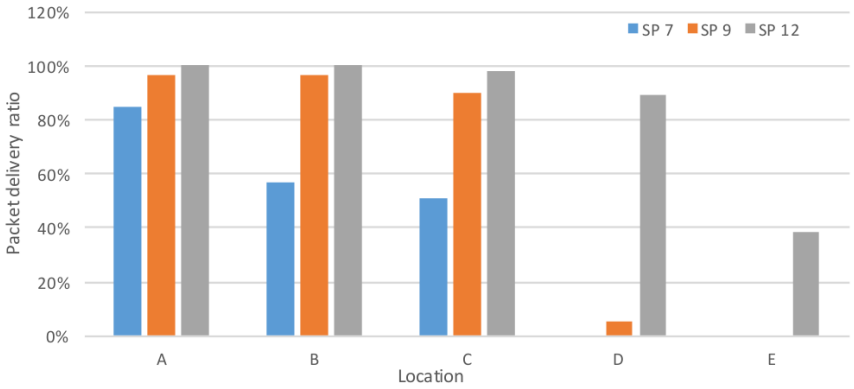
3.5 Protocol testing

In the following, throughout testing of the LoRaWAN protocol found from the literature will be covered. In addition, Message Queueing Telemetry Transport (MQTT) as a protocol for supporting data dissemination will be discussed.

3.5.1 Single device maximal throughput



(a) Single device maximum throughput



(b) Single device packet delivery rate

Figure 3.6: LoRaWAN single device throughput and delivery rate (credits: (Augustin et al. 2016))

An experiment conducted in Paris, France, mapped out the maximal data throughput of a single LoRaWAN device using SFs 7 to 12, variable payload sizes of 51, 25, and 1 bytes, and a fixed bandwidth at 125 kHz. As shown in Figure 3.6a, higher SFs drastically decreases the average throughput. However, higher SFs carry a higher chance of packet delivery, as shown in Figure 3.6b, where node placements were, in order from A through E, 650, 1400, 2300, 2800, and 3400 meters. In addition, the experiment found the mandatory receive windows following a transmission to carry the biggest impact on overall latency, and not the

duty cycle limitations (Augustin et al. 2016).

3.5.2 Throughput testing on simulated radio networks

In relation to effectively connect allied and partner nations in a federated network, standardization of communication protocols is currently a core subject in NATO Federated Mission Networking (FMN) (North Atlantic Treaty Organization (NATO) 2015). Standardization and profiling work conducted through FMN has so far been conducted using static and deployed networks, rather than Mobile Ad-Hoc Networks (MANETs) with limited edge capacities. Currently, NATO uses a Web Services Notification (WS-N) (OASIS 2006) standard for use in publish/subscribe services, which was concluded to have too large an overhead for such low capacity networks as it involved an additional SOAP (W3C 2007) message layer which significantly impacted network delay (Bloebaum and Johnsen 2015). For this reason, robust and lightweight protocols with low throughput costs are necessary to increase network reliability.

MQTT is a publish/subscribe Machine-to-Machine (M2M) connectivity protocol which has shown promising results over simulated radio models (Johnsen, Bloebaum, Jansen et al. 2019), which compared to WS-N has a lower delay and data volume throughput. A third option tested in the same study, Message Queueing Telemetry Transport for Sensor Networks (MQTT-SN) (A. Stanford-Clark 2013), proved to have an even lower data volume, but at the expense of lower reliability due to its UDP usage rather than TCP. It is however worth noting that the conclusions from these protocol tests are based on radio models using Wi-Fi, which do not reproduce the latency and throughput limitations to that of real tactical radio networks.

3.5.3 MQTT as dissemination protocol

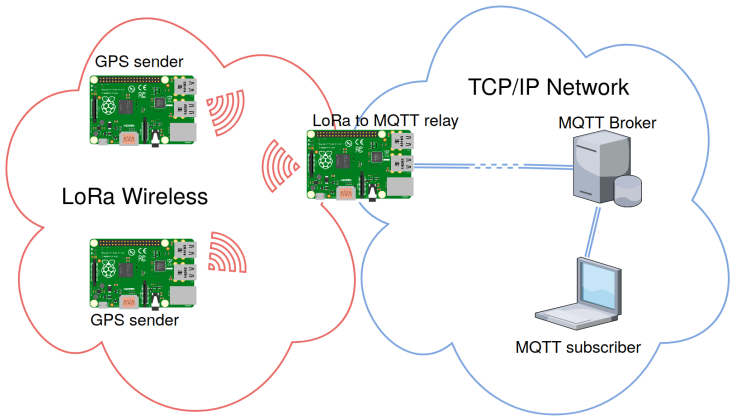


Figure 3.7: Test infrastructure with MQTT over LoRaWAN (credit: (Johnsen, Bloebaum and P. Puente 2019))

As part of investigating its potential for use with MIoT systems, MQTT has been tested as the dissemination protocol for data published by LoRa-devices. The experiment, conducted by FFI, used a simple LoRaWAN network with a single gateway forwarding packages from two other devices to a machine running a MQTT broker, as shown in Figure 3.7, using MQTT publish. The data were subsequently consumed by a client running on a separate machine through MQTT subscribe. The time-effectiveness of the protocol could then be determined by measuring the Round Trip Time (RTT) from the time a message was sent over LoRa to the time it was received by the subscriber, albeit in very good signal conditions and with very few devices producing data.

As expected, the experiment showed that the SF had the biggest impact on transmission time, followed by bandwidth, where larger available bandwidth yielded a shorter transmission time (Johnsen, Bloebaum and P. Puente 2019). The RTT timings relative to the number of transmissions is shown in Figure 3.8. The cause for the visible drop in RTT for the first ten transmissions are still unknown.

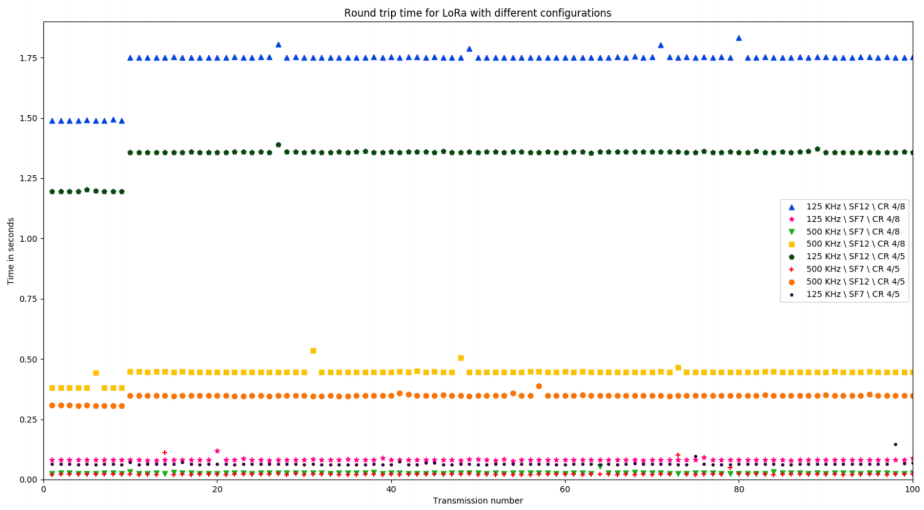
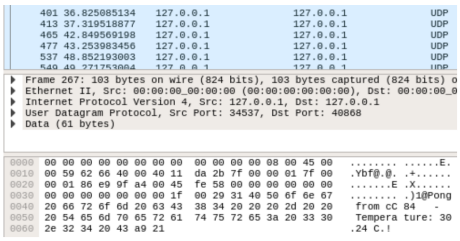


Figure 3.8: RTT from LoRa transmitter to MQTT subscriber (credit: (Johnsen, Bloebaum and P. Puente 2019))

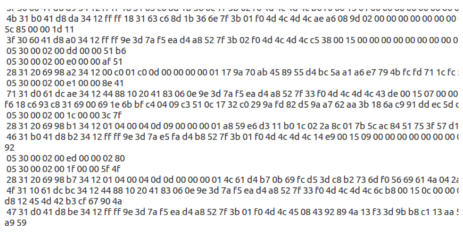
3.6 Security

During a LoRa-focused experiment conducted in Maryland, USA, a core research contribution was to investigate the possibilities of intercepting LoRa communication and injecting LoRa packets using inexpensive COTS SDR equipment, where it was concluded that in particular M2M mode[1] is potentially vulnerable against interception as the packages are transmitted unaddressed and unencrypted, while the regular LoRaWAN mode kept

[1]M2M mode is communications between two devices using LoRa PHY only, meaning no encryption as provided by the LoRaWAN stack is involved.



(a) LoRa packet capture



(b) LoRa traffic capture

Figure 3.9: LoRa package interception and traffic capture (credits: (Søndrol, Jalaian and Suri 2018))

their data integrity intact due to its use of encryption (Søndrol, Jalaian and Suri 2018). The data captured from the air is shown in the Wireshark capture in Figure 3.9a. While still being interceptable, it was impossible to extract payload data, as shown in the traffic capture in Figure 3.9b.

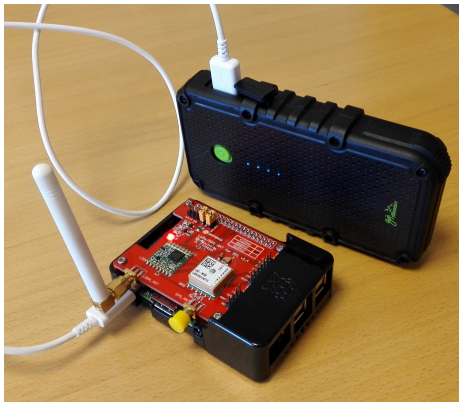
The experiment also attempted LoRa packet injection using a Hak5 SDR (Great Scott Gadgets 2014) as the transmitter and a RTL-SDR (RTL-SDR 2021) as the receiver, by using a modified GNU Radio script developed by Matthew Knight (Knight 2017), which found that it was possible to inject LoRa packages by using non-LoRa devices, albeit at very low ranges at 1 to 30 meters.

3.7 Hardware comparison

3.7.1 End-nodes

For prototyping and testing, the Raspberry Pi nanocomputer is a popular platform which supports a wide range of peripheral devices through Hardware-Attached-on-Top (HAT) which connect to its General Purpose Input/Output (GPIO) pins. To create a LoRa node from a Raspberry Pi, a LoRa shield from Dragino (Dragino Technology Co. 2016) with an embedded GPS module can be mounted onto the Pi directly, as shown in Figure 3.10a. This particular shield is based on the Semtech SX127x chip and can operate on three different ISM bands, namely EU433, EU868, and US915, and has served as a prototype in range-testing (Johnsen and P. Ø. Puente 2018).

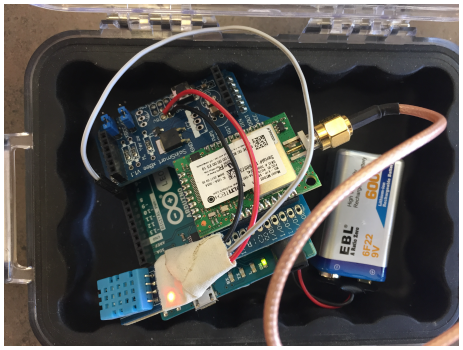
The Pytrack and LoPy4 microcontroller development boards from Pycom (Pycom Ltd. 2017) has, like the Dragino LoRa/GPS HATs, also served as a prototype in range-testing. The Pytrack consists of a GPS, GLONASS, and an accelerometer module, while the LoPy4 consists of the wireless modules, including the LoRa radio module. Based on the Semtech SX1276 chip, it runs a flavour of Python called MicroPython and supports both LoRa and Sigfox, in addition to Bluetooth Low Energy (BLE) and WiFi. Unlike the Raspberry Pi, it does not run any operating systems and requires less effort to setup. The ISM band on which it operates is immutably hard-coded, meaning that LoPy4 devices used in the US cannot legally be used in Europe, and vice versa.



(a) Raspberry Pi with Dragino LoRa/GPS HAT (credit: (Johnsen, Bloebaum, Jansen et al. 2019))

(b) LoPy4 (credit: (Michaelis et al. 2019))

(c) Freescale KRDM-KL25Z development board with mounted SX1276 Mbed shield (credit: (Augustin et al. 2016))



(d) mDot/Leonardo LoRaWAN node (credit: (Jalaian et al. 2018))

Figure 3.10: LoRaWAN end-node prototypes

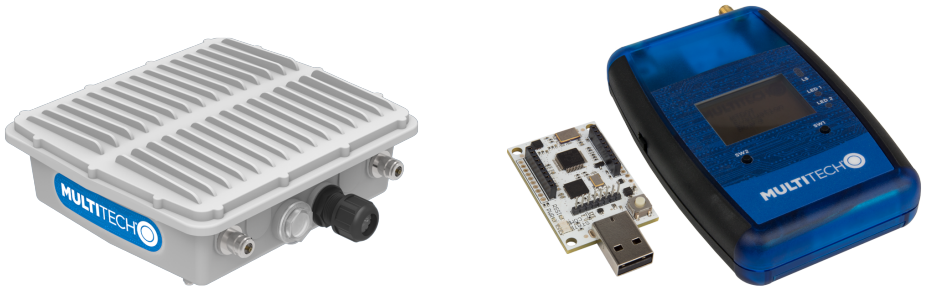
An additional prototype consisting of an mDot LoRaWAN board from MultiTech, mounted via an Xbee shield on top of a Leonardo Arduino microcontroller, is shown in Figure 3.10d. Both setups are from the field experiments conducted by ARL in Maryland.

The final, notable prototype setup is the Freescale KRDM-KL25Z development board (Arm Mbed 2015) mounted with a Semtech SX1276 Mbed shield (Arm Mbed 2014), used in the experiments conducted in Paris, France. The KRDM-KL25Z development board served as the integration platform, while the SX1276 shield served as as expansion component providing the LoRa radio module.

This field experiment found the Pytrack/LoPy4 setup to be a lot easier in terms of assembly, modification and implementation. In addition, the test results showed that the accomplished range using this setup was better than the other prototypes (Jalaian et al. 2018). The field testing conducted in Norway also made use of the LoPy4 setup, but experienced

some problems when attempting to change the SF and bandwidth, possibly due to “flakey" firmware differences between devices made for different regions. Consequently, the reliability of the Raspberry Pi setup was found to be the best hardware platform in this particular case.

3.7.2 Gateways



(a) MultiTech Conduit IP67 gateway (credit: (MultiTech Systems Inc. 2018))

(b) MultiTech mDot Box gateway (credit: (MultiTech Systems Inc. 2021b))

Figure 3.11: LoRaWAN gateways

A number of gateways have been tested in relation with the field experiments outlined above. For the range-testing conducted at FFI, the MultiTech Conduit IP67 as shown in Figure 3.11a was used. This particular gateway support up to 64 simultaneous channels, and comes integrated with an LTE module for Internet reachback.

The MultiTech mDot-Box shown in Figure 3.11b is a handheld alternative to its rugged counterpart, however this particular device is meant to work as a measuring tool to test coverage and provide means for proof-of-concepts rather than operational use.

3.8 Summary

In this chapter, we discussed high-level IoT use cases for military applications, such as personal sensing, logistics, fleet management, crowdsensing, C4ISR, and facility management, in addition to leveraging existing IoT infrastructure for augmented SA. In this thesis, we pursue wearables for military use, an aspect of the personal sensing application category.

Next, critical issues surrounding the implementing of civilian IoT technologies into existing military organizations were discussed. In particular, network coverage, secure interoperability, and data analysis of massive volumes of data in partitioned and disconnected networks are a major challenge which to this day remain as large research topics.

From the field experiments focusing on finding practical usage of LoRaWAN, distances of up to about 10 km have been achieved. However, the experiments have shown that the RF links are highly prone to

connection losses at the lack of unobstructed LoS to the gateway, a problem particularly present in dense urban areas. Apart from practical ranges, other noteworthy observations are the importance of topology, where multiple gateways significantly increased coverage. Another discovery made from the experiments was the unreliability of the backhaul link, which is most likely due to its use of UDP to the receiving LNS.

For information dissemination, MQTT have shown promising results for use in constrained and low-throughput networks, thus enabling for a potentially effective substitute for existing WS-N protocols currently in use.

Finally, a brief hardware comparison was made, where previous experiments have favoured the Pytrack and LoPy4 or Raspberry Pi with a mounted LoRa/GPS HAT, both of which were programmed using Python.

Chapter 4

Design

In this chapter, we will use interview findings and lessons learned or identified from previous work related to LPWAN usage in the military domain to specify a MIoT subsystem design for a soldier wearable. In order to specify functional requirements for the system, interviews of serving military officers will be conducted as part of a fact finding process, as described in Section 1.5. In combination with LoRaWAN-specific implementation constraints, the findings in these interviews work as the foundation for said system design.

4.1 IoT baseline

The range of enabling technologies spans across a wide selection of paradigms, hardware platforms, protocols, software libraries and frameworks, some of which were mentioned in Chapter 2 and 3. Hence, narrowing it down to a specific baseline facilitates for a more manageable and interoperable system for both integrations with existing solutions and future work. A feasibility study conducted in 2017, which focused on consuming smart city sensor information for military use, laid the foundation for further proof-of-concept efforts by specifying the following IoT baseline (Johnsen, Zieliński et al. 2018):

- **Information exchange:** JavaScript Object Notation (JSON) (ECMA 2017), a human-readable, easy to parse and generate, lightweight data-interchange format.
- **Dissemination protocol:** MQTT
- **Waveforms:** Wi-Fi and LoRa/LoRaWAN

Based on the findings from the literature in Chapter 2 and 3, the same baseline is used for the design- and implementation process in this thesis.

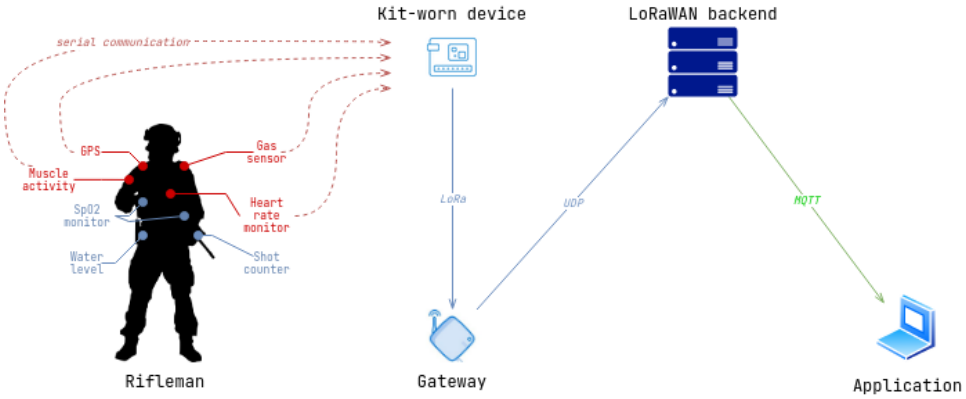


Figure 4.1: Soldier wearable high-level architecture

4.2 MIoT subsystem proposal: Soldier wearable

Due to the vast selection of possible use cases that can be put into practical testing, we have to limit the scope of a MIoT system by restricting the work effort to one in particular, of which the soldier wearable subsystem was chosen, as it is generally easier to integrate sensors on existing soldier platforms than on vehicles, vessels, or aircraft, thus making a rifleman-worn MIoT system an obvious candidate.

The high-level architecture of the proposed solution can be seen in Figure 4.1, where at the far left, the sensors we want to integrate and test are highlighted in red color. Other alternatives that were considered is highlighted in blue color, which will not be part of the MIoT subsystem in this thesis. Inspired by suggested solutions outlined in related work (Fraga-Lamas et al. 2016; Suri et al. 2016; Johnsen, Zieliński et al. 2018) and the proposed pool of input at the physical layer outlined in the *IoTNetWar architectural framework* reference model described in Section 2.2, the sensor kit will include a GPS, biometric sensors for Electrocardiography (ECG)[1] and Electromyography (EMG)[2], as well as a sensor for gas detection. As the prototype was developed using commercial equipment, the gas detector is not designed to detect military-grade weaponized gases. Rather, it is able to detect the presence of gases aimed towards industrial- or work environments for health and safety purposes, such as carbon monoxide and ethanol.

[1]ECG is a technique for evaluating heart activity through the electrical activity of the heart muscles.

[2]EMG is a technique for evaluating muscle activity through the electrical activity in skeletal muscles.

4.3 Subject-based considerations

As part of establishing the system design, three semi-structured interviews were conducted following the principles described in Section 1.5, which involved military personnel of different yet relevant operational backgrounds. The interview guide presented in Table 4.1 was developed and used to establish a common foundation in the discussions involving the three informants. Case 1 through 3 represent the same cases discussed in Chapter 1. That is, case 1 is social patrol, case 2 is urban assault against a fortified enemy, and finally case 3 is LRRP. The statements given below are quoted and attributed to the informants (denoted INF1..3), to advance the knowledge of the problem space and aid the overall design process. Note that the interview consists of two parts; part one establishes an understanding regarding how the informants relate to information flow in the given cases, part two hypothetically applies the soldier wearable to the same cases to see if and how the informants would treat the cases differently.

Table 4.1: Interview guide

Case	Question
General	What kind of information would you generally need from the unit you are commanding (including down to each individual) throughout a mission with or without hostile activity?
	In general, would you prefer more or less radio equipment in your load-out? In either case, why?
Case 1	If you were part of the patrol as a squad leader, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates, and how often? This can be either from each individual member of the unit, or from the unit as a whole.
	If you were the OpsOff stationed in Headquarters (OPS), how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the patrol, and how often? This can be either from each individual member of the unit, or from the unit as a whole.
	As patrol leader, what information would you normally receive or request from HQ during the mission, other than what you've already mentioned?
	Given the assets standing by at your disposal (Unmanned Aerial Vehicle (UAV) and Quick Reaction Force (QRF)), how would you normally activate these, and what criteria do you feel would need to be fulfilled for you to do so? What information would you normally have to provide with the request?

Continued on next page

Table 4.1 – continued from previous page

Case	Question
Case 2	If you were part of the deployed unit as the GFC (i.e. platoon commander), how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the deployed unit, and how often? This can be either from each individual member of the unit, or from the unit as a whole.
	If you were the OpsOff stationed in OPS, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the deployed unit, and how often? This can be either from each individual member of the unit, or from the unit as a whole.
	As GFC, what information would you normally receive or request from HQ during the mission?
	In the event one member of the unit is wounded during the operation, how would this particular event affect the information flow? How would you as GFC handle this?
Case 3	If you were part of the deployed unit as the patrol leader, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the patrol, and how often? This can be either from each individual member of the unit, or from the unit as a whole.
	If you were the OpsOff stationed in OPS, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the deployed unit, and how often? This can be either from each individual member of the unit, or from the unit as a whole.
	As patrol leader, what information would you normally receive or request from HQ during the mission?
	In the event of Troops in Contact (TIC) during infil or exfil, how would the information flow internally in the squad and between you and HQ?
Part two of the interview following the soldier wearable proposal	
Case 1	Hypothetically, to which level do you think using such a system would benefit you in the role as OpsOff in OPS or patrol leader in this case?
	In the role as a OpsOff in OPS or patrol leader, do you have a positive or negative view regarding using such a system in this case? Please explain why.
Case 2	Hypothetically, to which level do you think using such a system would benefit you in the role as OpsOff in OPS or GFC in this case?
	In the role as a OpsOff in OPS or GFC, do you have a positive or negative view regarding using such a system in this case? Please explain why.

Continued on next page

Table 4.1 – continued from previous page

Case	Question
Case 3	Hypothetically, to which level do you think using such a system would benefit you in the role as OpsOff in OPS or patrol leader in this case?
	In the role as a OpsOff in OPS or patrol leader, do you have a positive or negative view regarding using such a system in this case? Please explain why.
Other	Other comments or thoughts regarding usage of such a system, independent of the described cases?

In the following, we will analyse the responses from the three informants in order to identify key points towards describing functional requirements for a soldier wearable. These will be described in the context of mission equipment with emphasis on communication systems, mission-specific information requirements from the point of view of GFC and OpsOff, the potential information detail augmentation made possible by the soldier wearable, and what challenges the informants think might arise if such a system would be put to operational use in the existing military organization.

4.3.1 Mission equipment

This topic relates to mission-specific equipment with emphasis on communication systems, and how the informants relate to the level of management required by them to bring such equipment along on missions.

One informant outlined a number of inconveniences for bringing communication systems along on missions, especially if it was extra equipment, as stated by the following:

> [...] all radio equipment takes up space, weighs a lot, requires power. Power and batteries requires further resources in addition to transporting all that radio equipment. If I'm to bring more, then it'd be if we're working with multiple types of communication systems that provides different types of information.
>
> — INF2

Another informant outlined the level of management required to handle multiple communications systems, where he stated the following:

> [...] a lot gets lost because of the large variety of platforms in active use. Because when a lot happens on the voice systems, and we start to receive a lot on the text systems from other actors that are not directly involved with what happens over voice, then it takes a lot of time until we are able to process that.
>
> — INF3

Based on the statements given above, it might seem to be ideal to only bring what is necessary to solve the mission, as more equipment would mean more maintenance, added physical weight to vehicle or personnel, and added strains to battery or energy sources.

4.3.2 Information detail requirement: Ground Force Commander

This topic relates to mission-specific information which the informants consider to be important in the role as GFC, and what information they would require from the unit they are commanding, and the information exchange between them and OPS. The informants seem to agree that their understanding of the enemy is one of the most important factors for their ability to solve the mission. In addition, information related to how combat-effective they are, and changes in the situation, seemed to play a crucial role as well, as derived from the following statements:

> [...] the information you need is really the mission, where is the enemy, what can the enemy do against you, and what can you do to the enemy. What I as patrol leader need is therefore all changes in the situation that carry a meaning for you and your team.
>
> — INF1

> I am interested in as accurate descriptions of the enemy as possible, and their current location [...] and what kind of unit is it, an infantry platoon on foot, or a tank, in which case what kind of tank is it, and of course where they are going. [...] At individual level, I rely on information regarding how each individual is doing in order to take care of them. Then its mostly general status updates, are they OK, are they capable of doing the mission, what resources do they need.
>
> — INF2

> As maneuver platoon commander I would need personnel status, if there's been contacts with the enemy, any casualties, illnesses and such, just to know exactly how combat effective we currently are. Daily log reports of all kinds of logistics,

> water, ammo, fuel, etc. and status on the material, like primarily our vehicles. Then in addition we do a lot of reports regarding systems, in particular comms systems. The most important information however, is information during the mission, our units own position, contact reports, target acquisitions for those contacts, their assessments, their status after the contacts, how the enemy reacted, etc.
>
> — INF3

Based on the answers from the first case, the information requirement for the GFC regarding the status of each individual soldier's status and their perception of the current situation can be interpreted to be necessary, as derived from the following statements, where all informants have similar answers:

> [...] you don't need continuous updates, you need updates of changes. If something happens, you need to know immediately. You're not interested in the meaningless chatter.
>
> — INF1

> If someone sees something suspicious, then that would be reported back to OPS. Then they might provide us with a recommendation, and I'd take a stand based on that. I would also probably ask for a personnel status within the unit every hour or half hour, if they're struggling with something or anything at all.
>
> — INF2

> On the internal comms I would have more or less continuous chatter regarding situation updates from my own patrol, or at least quite often. Things like sectors, observations, and such so that I can have a good SA. Including status on each individual.
>
> — INF3

One informant states the importance of keeping track of his unit, before essentially keeping the OpsOff informed, where he indicated that OpsOff will be informed after a certain amount of time have passed since the event took place:

> Normally the platoon commander wants to primarily keep control over his platoon, then secondarily to keep the updates flowing to the OpsOff.
>
> — INF1

For the urban assault case, one informant outlined the necessity of leaving the net open for the troops engaged in combat:

> During the fire and maneuver, then it is important that there is as little chatter on the comms as possible so as to give the squad leaders space to conduct the mission, and not be blocked by unnecessary chatter.
>
> — INF2

Two of the informants also described the sequential nature of the tasks at hand, where the tasks involving the actual combat takes precedence, followed by re-organization:

> Coordination between the squads would take a lot of space on the comms, usually controlled by me. Like who is the breaching team, who is covering what sector. Then afterwards, we would do a complete re-organization, a situation report from all the squads, and an assessment for further action.
>
> — INF3

> When we have taken control of the building and the enemy is neutralized, then we have to re-organize, how many enemies are neutralized or taken captive, what's the status on the rooms, do we need external engineer support. Here, the chatter on the comms may start again, then I as platoon leader will need as much information as possible that I can push upwards.
>
> — INF2

Regarding medical evacuation tasks in relation to the urban assault case, one informant outlines his perception of the distinct steps they would make to organize the use of such resources, which includes a standard reporting format and some coordination between the GFC, the medic treating the wounded, and the medical evacuation unit, as derived from the following:

> [...] my task is just to get the medical resources we need, such as reporting the 9-liner, and to task the MEDEVAC capacities we need, be it in the air or on the ground, and to evacuate the wounded. The medic would provide me with the minimum amount of information I would need to get the medical resources deployed to us, and when it arrives then that medic would conduct HOTO to the MEDEVAC.
>
> — INF2

According to the following statement, the information chain between the medic treating the wounded and the external medical resources therefore consists of multiple links, including GFC. However, another informant instead states that he expects the medic to conduct the coordination with the external resources himself:

> [...] it is not up to you as platoon commander to coordinate with MEDEVAC or the likes, that is the medic's job. You do not wish to spend resources on wounded soldiers, because this means loss of combat forces.
>
> — INF1

Based on the statements provided above, it might seem that information provided over voice to a certain extent follows a somewhat standard format, such as status updates. During offensive operations, it was indicated that fire and maneuver related transmissions would take precedence on the radio net until the threat is eliminated, thus causing other reports or otherwise mission-specific information to be delayed until GFC receives this information. In medical evacuation cases, the information chain could in some cases be made of multiple links before the necessary information arrives at the intended recipient.

4.3.3 Information detail requirement: Operations Officer

This topic relates to mission-specific information which the informants consider to be important in the role as OpsOff, and what information they would require from the deployed troops so as to provide necessary support from OPS. For the social patrol case, one informant indicated that a more continuous form of updates regarding the situation on the ground was more desirable when he was placed in the role as OpsOff:

> Then it is suddenly different, then you want updates all the time. [...] Then you are preoccupied with receiving updates from the patrols as often as possible, that all is OK. Then radio checks are very nice, just to check that they are still there. Which is really annoying as patrol leader on the ground.
>
> — INF1

Another informant states that he would actively prompt the patrol for more information if something out of the ordinary occurs or if the provided information from the patrol is lacking:

> [...] if something have happened then I'd be reaching out a bit more, depending if the information I receive from the patrols are good enough and detailed enough, or if the reporting procedures are up to standard.
>
> — INF3

One informant outlined the rather manual nature of information dissemination between himself in the role as OpsOff to the GFC in situations where only voice communications is available:

> Depending on what communications system we have, if we only have voice and no data comms, then I'd receive these updates over voice, which would have to be written down and converted into a visual presentation on a map, regarding where the enemy is located, where our patrol is, whether there is enemy activity, civilians, where is the UAV, etc. If we cannot provide the patrol with a map update over data comms, then that update will have to be provided over voice to everyone in a simple fashion, so that everyone knows where all friendly forces are located, what they are doing, do they have any enemy activity, and what the UAV is doing.
>
> — INF1

Another informant outlines the slow process of acquiring and disseminating intelligence updates to the GFC in relation to the urban assault case, as derived from the following:

> You receive requests on intel from the platoon commander, which you provide if you have access to it. Intelligence updates during the attack regarding the target takes a long time to reach its intended recipients [...] Everything that can have some form of impact on the mission should be conveyed to the platoon commander.
>
> — INF1

Based on the statements above, the effectiveness of keeping track of, and to support, deployed units could rely in part on the communication systems in use, where situations involving only voice-based communications may inflict a large amount of manual tasks for the OpsOff.

4.3.4 Mission-specific information augmentation

This topic relates to potential augmentation of the information provided from the deployed troops which would be using the soldier wearable, in addition to hypothetically increasing its operational value by providing said information faster, thus enabling for a shorter OODA-loop.

Based on the overall response, the informants seem in some degree positive towards utilization of a soldier wearable, as it could potentially close information gaps and improve operational tempo, as derived from the following statements:

> You really get full control over your squad, mainly with respect to healthcare and logistics. You automatically know the losses,

> if the ammunition drops drastically then we are suddenly facing something that isn't quite right. As squad leader, you will always know where people are [...] The less time spent on your third arm, the better. Then you can focus on the mission.
>
> — INF1

> Especially as OpsOff it would be extremely useful, because it would be incredibly easy to get detailed status updates from the unit. [...] it would be incredibly fast as well to react on the provided data, this and that unit needs resupplies, this and that unit has MEDEVAC needs, which would save an awful lot of time spent on the radio. If I as a patrol leader had, say, a tablet or an Android device with the processed data, then that too would have been very nice [...] if the same user interface shows detailed information about neighboring patrols as well, then I will get a lot better understanding of the situation, which would make me able to assess the situation and save a lot of chatter time, since the core of voice chatter is assessment.
>
> — INF3

One of the informants expressed a more positive stance for utilizing such a system in a less general manner, where the mission case is of low-intensity by nature, and with a low risk of EW threats. The same informant also indicated that such a system could positively affect logistics operations:

> What I'm a bit more skeptical towards is the use of biometric sensors in this specific case. [...] I think it would be too much information for a ground force commander to monitor heart rates, and other biometric conditions. [...] For social patrols, so-called low-intensity operations, where the enemy pressure isn't necessarily that high, not as high EW threat, then a think such a thing would be very good. In particular on the logistics side, especially water, so that we in the rear can be ready with resupplies, but also in terms of biometrics, since personnel status in such situations is very important. [...] I think everything that exclusively handles logistics, especially water and ammo, I am exclusively positive towards that elements in the rear get this information. [...] I have also worked with combat support, and it is great to receive this information quickly in order to provide necessary logistical support to the front.
>
> — INF2

One informant outlined the possibility for faster information dissemination through such a system, as stated by the following:

> I think it is in this scenario that this technology would be of most value, since it is a local high-intensity scenario, complex environment and mission, lot of internal chatter, challenging to lead externally and in relation to higher elements. So this would have been an extremely good thing as platoon commander, since HQ would receive all this information a lot earlier than if I've had to provide it for them. So it would help us be one step ahead, rather than waiting for me to identify the same needs that the technology would pick up. In addition it would save a lot of time spent on the radio systems.
>
> — INF3

Based on the statements above, it is likely that such a system could improve operational tempo and save time, thus enabling more time for commanders to focus on the mission at hand. In particular, automated logistics and resupply management received exclusively positive responses. Furthermore, biometric data could potentially help commanders re-organize faster with respect to solving the mission. However, it was also suggested by one informant that biometric data monitoring may not improve the situation on the ground, since this could lead to information overload for the GFC.

4.3.5 Challenges

This topic relates to challenges identified by the informants if the soldier wearable were to be implemented into the Norwegian Armed Forces today. In particular, it was found that the leadership culture could show resistance against the use of a soldier wearable, in addition to using the information to micromanage leaders on lower levels, as stated by the following:

> [...] what you're presenting here is a disruptive technology that will alter some of the cultural procedures in the armed forces. So I don't necessarily see all the possibilities that such a system provides. Because during my education and my experience I've been affected by existing procedures and culture [...] I also think it will be certain resistance within the organization simply because the armed forces, in particular the army, is very conservative, where leaders are educated in how the Germans conducted infantry- and maneuver combat during WW2. Generally, there is a lot of resistance against new things, especially against things that there might be some insecurities surrounding its usage [...] So I think the project needs to discuss how this system fits into a assignment-based leadership, and in armed forces leadership philosophy, with regards to a very conservative officer corps. The more information the

> commanding officer has in regards to me and my mission then he can fall into the trap of micromanaging me, which I've experienced a lot before when using BMS.
>
> — INF2

The concern for the EW threat was present with all informants, as given by the following statements:

> If you can lead within a bubble, and all internal communications happens within that bubble, then it isn't a problem at all using such a system. But then again, when you're in OP, the problem is being detected.
>
> — INF1

> With existing knowledge, then I think such a system would be very vulnerable for detection by enemy EW, which would give away the position of the patrol, especially if they are behind enemy lines.
>
> — INF2

> [...] except for the EW threat of course. That would be the only big challenge I think. But apart from that, this would have been very valuable.
>
> — INF3

One informant outlined the importance of filtered and aggregated information in accordance to the military level of the recipient, as derived from the following statement:

> [...] assuming that the information is aggregated correctly at all levels so that unnecessary information does not get displayed. It is easy to get stuck on details.
>
> — INF1

This concern overlaps to some degree with the concern for micromanagement by higher commanding elements, as stated by the following:

> I am a bit skeptical with regards to [...] micromanagement by commanding elements in the rear [...] We see this already with BMS usage, it provides higher elements a lot more information, and it makes it possible to micromanage the units a lot more. The more streams of data from the ground, the less mission-based leadership you get, and the more micromanagement you get.
>
> — INF2

4.3.6 Key take-away points

To summarize, the findings from these interviews range from both technical aspects to cultural concerns. In line with the scope and limitations of the goals for this thesis, a priority list can be extracted from the findings as follows:

1. The level of detail presented by the high-level design idea gained positive responses from the informants. Thus, the same level of detail should be implemented in the prototype so as to determine whether or not it is of operational value. However, raw biometric data may not provide commanders with improved SA, as this could lead to information overload, where the data would need to be interpreted in context of the situation to that of the wearer.

2. The provided data must be filtered and aggregated at an appropriate level in accordance to the viewing audience. For this thesis, we are considering officers at GFC or OpsOff positions as these were used in the interview cases.

3. Hostile EW has been identified as the prime counter-argument against implementing autonomous sensing across the whole military organization. At the level on which the soldier wearable resides, local Emission Control (EMCON) (i.e. radio silence) should be in place in situations where it is necessary to attempt to avoid detection by RF emissions.

These items are, as far as possible, considered to be core requirements for the prototype developed as part of this thesis. Note that the quotes represent the opinions of the informants, whereas the synopsis and emphasis for the design is based on the author's understanding of the problem space, the discussions conducted with the informants, and the author's own, previous operational experiences. This is leveraged in the further design of the prototype soldier wearable.

4.4 Technical considerations

In the following, we will shift focus to technical aspects in relation to implementation of the soldier wearable. Specifically, we will consider LoRaWAN messaging best practices in terms of sizes and formats, and thus also time on air. Furthermore, hardware and open source solutions fitting the use case of the prototype will be discussed.

4.4.1 LoRa messaging

Given the constraints both in terms of battery lifetime, ISM duty cycle restrictions, and EW threats, it is in our best interest to keep messages as short as possible. Both message size and Spreading Factor heavily affects the airtime and energy consumption, while the data volume on the air

Table 4.2: Payload formats and sizes (credit: Semtech Corporation 2019c)

Format	Message	Size
JSON string	{ "Temperature" : 20.635 }	40 bytes
Compressed JSON	{"t":20.635}	11 bytes
Using float instead of string	20.63	5 bytes
Signed 16 bit integer	0x080F	2 bytes

heavily affects the reliability of the network since packets may be lost due to busy gateways or a clogged RF spectrum.

It is considered best practice to always use bit packing to transmit sensor data over LoRa (Semtech Corporation 2019b), meaning to transmit sensor values in the smallest number of bits possible. Consider the format and size comparison as shown in Table 4.2: By removing unnecessary characters and using appropriate numerical representation, we reduced the payload size from 40 bytes to only 2 bytes.

In particular, textual information and meta-characters are generally a wasteful use of available message space. Keep in mind that for the EU868 band, we are limited to payload sizes to as little as 51 bytes when using DR0-2, assuming an empty `FOpt` field. If not, the total payload size will decrease even more.

To achieve effective bit packing, it is important to utilize appropriate scale and precision (Semtech Corporation 2019b). If we consider geographical positions represented by Latitude and Longitude formats, we see from Table 4.3 that there is a trade-off between accuracy and size requirement. Using six decimals, we achieve a resolution that suits our use case well, while conveniently being able to fit each parallel value within a 32-bit unsigned integer variable. A precise geographical location would then require 64-bits, rather than 110 bytes if we used a plain JSON string.

Finally, redundant data should be avoided so as to not transmit unnecessarily, which would deplete the battery quicker, clutter the RF spectrum and thus increase the risk for collisions, and increase the risk of EW detection and/or attacks. Hence, appropriate transmission policies

Table 4.3: Precision of decimal degrees (credit: Semtech Corporation 2019b)

Decimal places	Decimal degrees	Recognizable objects
0	1	Country or large region
1	0.1	Large city or district
2	0.01	Town or village
3	0.001	Neighborhood, street
4	0.0001	Individual street, land parcel
5	0.00001	Individual trees, door entrance
6	0.000001	Individual humans
7	0.0000001	Practical limit of commercial surveying

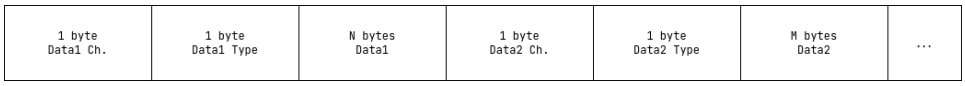


Figure 4.2: Cayenne LPP message structure (credit: (myDevices Inc. 2018))

and encoding schemes needs to be well-defined.

The naive approach is to transmit whatever data the sensor captures, regardless if it changes or not or zero-readings are made, and without regard for payload sizes. This would mean a lot of redundant and meaningless data would clutter the network, both in the LoRaWAN backend as well as the air interface.

Another, possibly better approach is to utilize the `FPort` field, where any port in the range of 1 to 223 can be used freely to link a specific port to specific data (Semtech Corporation 2019e). Thus, the LNS would only need to evaluate the `FPort` field of the LoRa message and subsequently launch an appropriate decoder in order to extract meaningful data. However, this approach would most likely cause time-separated sensor data. This is most undesirable if we want to tie geographical locations with critical sensor readings, such as toxic gas measurements and bad biometric readings. In addition, it will most likely lower the volume of uplink messages, as a larger fraction of the total data consists of message overheads due to separate messages for different data.

A third option could be prioritization of data, i.e. sensor readouts would each have an assigned priority, and in the event there is simply not enough space to put all sensor readings in one single message, the embedded software could prioritize for instance biometric readings over gas level measurements. This approach could however lead to data never being transmitted if continuous readings from a higher-priority sensor keep taking precedence.

A fourth option is a shared, dynamic schema on both ends which uniquely identifies data values using a pre-defined identifier. This particular solution, provided by Cayenne Low-Power Payload (LPP) (myDevices Inc. 2018), organizes a payload message such that the first two bytes functions as a channel and an identifier for the sensor in the device and the type of data it holds, respectively, followed by the actual sensor data, as shown in Figure 4.2. This approach allows for simple message packing and handling from multiple sensors at the expense of a slightly larger overhead, while also facilitating for easy decoding at the backend. The data types conforms to the OMA Lightweight M2M Object and Resource Registry (OMA (Open Mobile Alliance) SpecWorks 2018) (formerly known as the IPSO Alliance), which assigns a unique Object ID for known data, thereby creating a universal standard towards identifying sensor readings in dynamic environments such as IoT deployments. For example, GPS locations are designated with the Object ID 3336, and specifies 8 composite resources, most notably latitude and longitude, which should be represented in string formats.

4.4.2 Hardware

End-nodes

Semtech developed a BSD-licensed implementation to be used as the software component controlling end nodes in LoRaWAN networks, called LoRaMAC, which at the time of writing serves as a reference implementation supporting LoRaWAN Specification versions 1.0.3, 1.0.4, 1.1.1 and LoRaWAN Regional Parameters versions 1.0.3 and 2-1.0.2. To utilize this software, it must either match one of the supported platforms as listed in the code repository (Semtech Corporation 2014), or it must be ported to a platform which matches the MCU requirements (list of requirements can be found in Appendix E).

For the purpose of ease of development, a few open source LoRaWAN stacks for different targets is available; Arm Mbed OS LoRaWAN stack (Arm Mbed 2021b), ST Microelectronics LoRaWAN stack (STMicroelectronics n.d.), and LoRa Basics MAC (Semtech Corporation 2019d). The Mbed OS LoRaWAN stack is built to support development in C and C++ and is supported by a large number of MCU manufacturers. The ST Microeletronics LoRaWAN stack is design to be used on STM32Lxx MCUs, a family of 32-bit MCUs built around the Arm Cortex M-series CPU. The LoRa Basics MAC is a separate Semtech-developed LoRaWAN stack built to be a portable implementation.

To enable ease of development while maintaining the possibility to utilize as many implementation options as possible, it is therefore to our advantage to build a prototype based on a platform that can be used across all of the frameworks described above.

Gateway

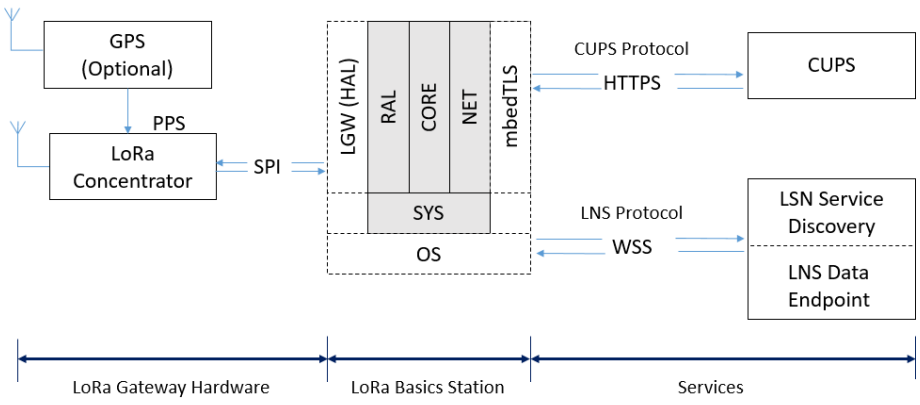


Figure 4.3: LoRa Basics Station system overview (credit: (Beitler and Singh 2019)

Most commercially available LoRaWAN gateways, such as the ones offered by Cisco (Cisco 2021) and MultiTech (MultiTech Systems Inc.

2021a), usually ship as an integrated platform in a rugged casing for outdoor usage, which are not usually easily extensible nor cost-friendly. For development and prototyping purposes, so-called concentrators[3] which should integrate well with Linux-based host platforms, such as the Raspberry Pi or Beagle Bone, should be used.

To setup a gateway, it must be installed and configured with a packet forwarder appropriate for the radio module it embeds. Currently, the one most widely used is Semtech UDP Packet Forwarder (Miermont and Coracin 2013). However, at the time of writing, LoRa Basics Station (Beitler and Singh 2019) seems to be the most viable solution for certain gateway setups, such as iC880A concentrator boards (IMST 2021) mounted on a Raspberry Pi host. This particular software package is featured with two distinct protocols, namely LNS and Configuration and Update Service (CUPS). The LNS protocol makes use of the WebSockets, a protocol widely used for two-way communications without opening multiple HTTP connections (IETF 2011), for information exchange between the gateway and the LNS, while the CUPS protocol enables remote firmware updates and connection credentials via HTTPS, as shown in Figure 4.3.

4.4.3 LoRaWAN backend

As stated in Section 3.2, military deployments requires the ability to maintain and operate a private infrastructure that do not rely on commercial connectivity, power, or data processing. LoRaWAN is however designed and licensed to be deployed on-demand using private networks, as discussed in Section 2.4.3, including its backend component (i.e. the Network-, Application-, and Join Server), which only require a common computation platform to run on. For this reason, we want to investigate the use of a self-deployed and self-maintained LoRaWAN backend to support the network.

At the time of writing, there exists a couple of open-source LoRaWAN backend solutions that can be used freely. Most notably, ChirpStack (Brocaar 2019) and The Things Stack (The Things Industries 2017). ChirpStack, released with a MIT license, seems at the time of writing to be the most popular solution for private LoRaWAN networks, and offers support for a wide range of application integrations, both external and custom, through ready-to-use APIs. The Things Stack, released with an Apache-license, is very similar in terms of features, but carries some difference regarding architectural design.

[3]Concentrators are a term taken from the telecommunications industry where a device multiplexes multiple low-speed channels into a single transmission medium.

4.4.4 Key take-away points

Based on the discussed options and solutions pertaining to LoRaWAN deployments, we can summarize the key findings as follows, which includes prospected technology candidates selected on the premises of open source, cost, and availability:

1. Cayenne LPP is the simplest and most flexible option for message packing for end-nodes that are implemented in such a way that the message content may vary, however at the expense of an additional 2 bytes per data field. If however a fixed message structure is in place, then manual bit packing will yield the lowest bit size, and thus also time on air.

2. The end-node hardware platform should be selected based on its compatibility across the available LoRaWAN stacks, so as to remain flexible in the implementation phase. As such, Mbed OS and ST cross-compatible MCUs with on-board LoRa radio modules seems to be the most viable solution.

3. Gateway hardware should be selected based on their compatibility with LoRa Basics Station, as this seems to be the best approach for newer LoRaWAN deployments for low-cost private networks.

4. Based on the features provided by the two LoRaWAN backends presented here, it makes no practical difference which one we use for the prototype development.

4.5 Specification

The specification constitutes the author's interpretation and understanding of the technology approaches viable to develop a prototype MIoT wearable supporting the previously identified operational cases.

4.5.1 Purpose and requirements

Purpose

A soldier wearable sensor kit which in part automates mission-specific data dissemination, lowering the need for voice-based transmissions for situation and status updates to higher military echelons.

Behavior

The wearable sensor kit should transmit as often as possible, provided that new sensor readings demands it, so as to provide up-to-date information, while using compact data formats due to the presence of EW threats and a cluttered electromagnetic spectrum. Specifically, the integration platform should as a minimum remain in sleep mode until its mandatory off-period in accordance with ISM band duty cycle restrictions have passed.

Once this time threshold have been met, the device should wake up to execute new measurements, with which it should compare the new and previous readings and decide whether or not the values meet the deviation requirements for transmission.

System management requirement

The sensor kit should require as little interaction from the wearer as possible, and only be managed remotely using C2 application integrations.

Data analysis requirement

The backend component should not perform data analysis as this is outside the scope of this thesis. Instead, it should filter the uplink data such that the output contains only the following data:

- `DevAddr`
- Callsign associated with the wearer
- GPS location data in Latitude and Longitude form
- Biometric qualitative descriptor, namely HEALTHY, EXHAUSTED, or UNHEALTHY, derived from the given heart rate and muscle activity sensor readings
- Gas detections

Application deployment requirement

The application should be deployed locally on the device, but should be remote-controlled using MAC commands (i.e. Network Server commands) and custom-made application commands (i.e. Application Server commands).

Security requirement

The end-nodes should use OTAA only for device activation. With this exception, security is not a concern in this thesis.

4.5.2 Process specification

From the requirements, we can derive four primary behavioral requirements that the end-node should meet:

1. The node should transmit as often as possible, but keep the transmissions as short as possible on the air, with respect to the EW threat. In between transmissions, i.e. the duty-cycle determined wait-period, the device should simply go to sleep.

2. New sensor measurements should meet a certain threshold in order to overwrite current sensor measurements, and then transmitted. Otherwise, the data should be ignored.

3. Emission control mechanisms needs to be in place which would override the default transmission logic.

4. The end-node should be capable of receiving custom commands from a simulated C2 component (i.e. the UI).

An activity diagram illustrating the workflow of these four elements are shown in Figure 4.4. Note that for Class A mode, which will be the only one mode used in this thesis, downlink message handling must succeed a transmission event, as described in Section 2.5.2.

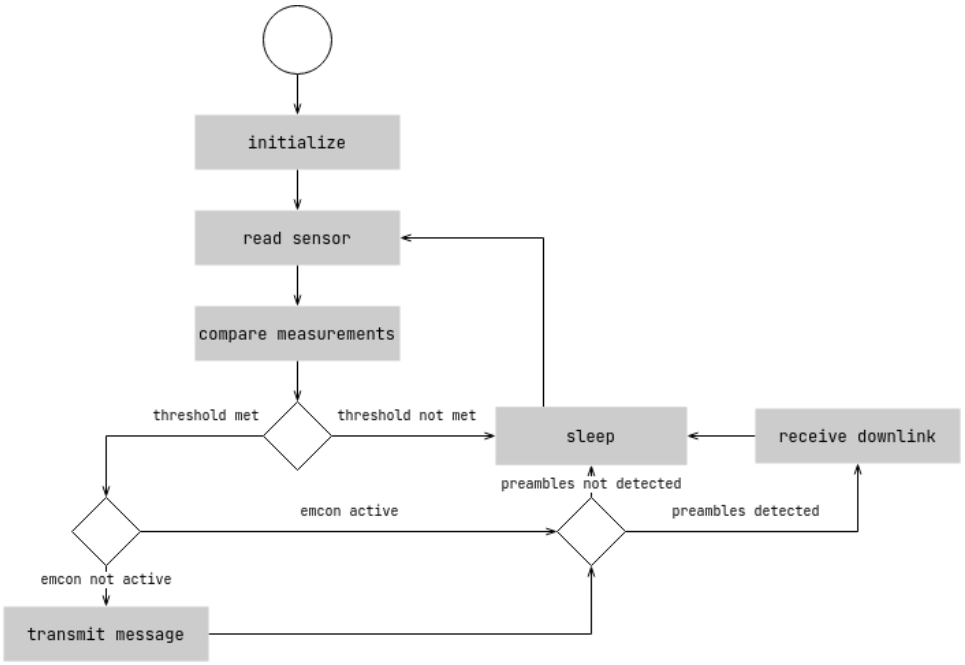


Figure 4.4: Process specification

4.5.3 Domain model

The domain model shown in Figure 4.5 illustrates how entities, objects, and concepts relate to each other in an IoT system, using the following descriptors:

- **Physical Entity:** A discrete and identifiable entity in the physical environment. In this case, we distinguish between the categories *Environment* (geographic location and gas levels) and *Personal* (biometric data).

- **Virtual Entity:** A representation of the Physical Entity in the digital domain, where for each Physical Entity, there is a corresponding Virtual Entity.

- **Device:** Refers to the platform which are used to gather information about Physical Entities using its attached sensors.
- **Resource:** Software components which either exists on-device (hosted on the device) or as network-resources (i.e. hosted on the network and must therefore be remotely accessed).
- **Service:** Provides an interface for interacting with the Physical Entities by accessing the resources hosted on the device or through the network. In this subsystem, the services will only be concerned with on-device resources, namely the APIs which enables LoRa transmission and reception.

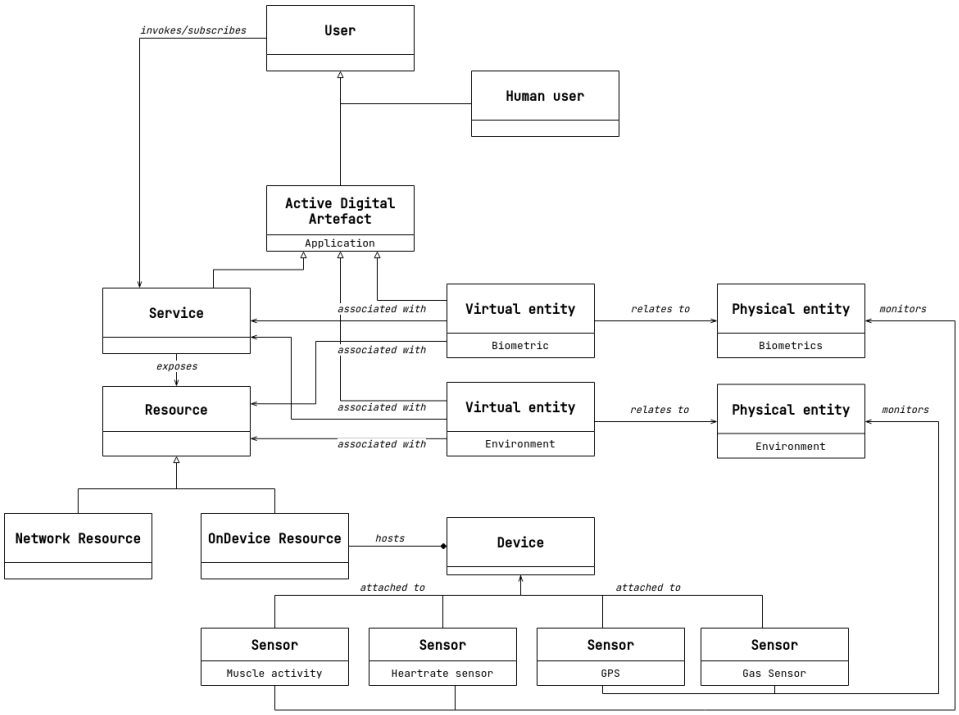
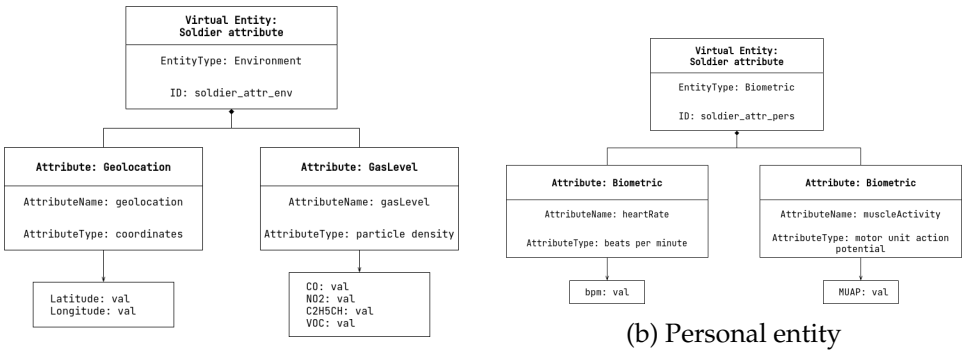


Figure 4.5: Domain Model

4.5.4 Information model



(a) Environment entity

(b) Personal entity

Figure 4.6: Information Models

The information model defines the structure of all the information in the IoT system derived from the Virtual Entities identified in the domain model. As we found two Virtual Entities, we use the information model to specify in detail how their attributes map to measurable units. The Virtual Entity *Environment* is described in Figure 4.6a, and lists the sensor values for geographical coordinates and gas level measurements.

The second Virtual Entity, *Personal*, maps to the biometric values of the wearer. The attributes for muscle activity and heart rate were chosen to represent a status class which reflects the wearers health status and exhaustion, as shown in Figure 4.6b.

4.5.5 Service specification

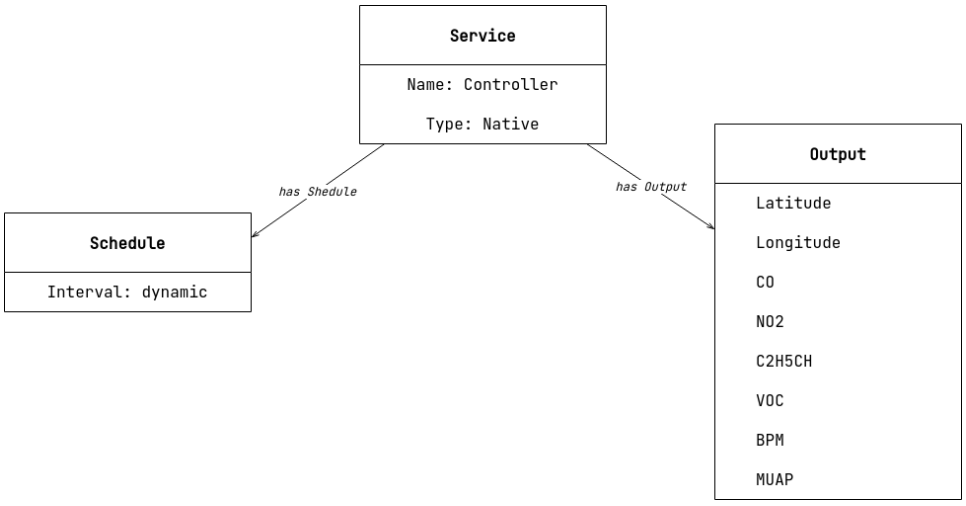


Figure 4.7: Service specification

For each state and attribute identified from the process specification and information models we specify a corresponding service, which either

change or retrieve the current values. As seen in Figure 4.7, there is one native controller running on the device which operates with a dynamic schedule, i.e. duty cycle restrictions in addition to threshold requirements for data transmission, and outputs a list of attributes as specified in the information models in Section 4.5.4.

4.5.6 IoT level specification

An IoT level specification logically organizes the components as either a local or remote (i.e. cloud) deployment, where the level denotes a certain level of complexity. The components are as follows:

- **Device:** The physical IoT device.
- **Resource:** Software components residing on the device for accessing, storing, processing, or controlling sensors or actuators, including network access.
- **Controller Service:** A native service that runs on the device and sends data to the web service, as well as receiving commands from the application. In this case, the service is the LoRa radio driver on the device which enables any kind of communication with the LoRaWAN backend.
- **Database:** Stores the data generated by the device(s).
- **Web Service:** Serve as a link between the device, application, database, and analysis components, and can be implemented using a suitable protocol, for example WebSocket.
- **Analysis Component:** Analyzes the generated data and present the results through a user application.
- **Application:** Provides an interface for the users to control and monitor the IoT system.

The deployment design resembles an IoT Level-2 deployment design where we use local machines serving as our backend platform hosting the cloud services we need, i.e. the LoRaWAN backend. As shown in Figure 4.8, the focus is developing a proof-of-concept system using one device and one gateway which constitutes the local segment. The cloud segment hosts both backend services and data storage, while the UI is locally available for device management and monitoring tools. The Analysis Component links the LoRaWAN backend with the UI, however without proper data analysis capabilities, as stated in Section 4.5.1.

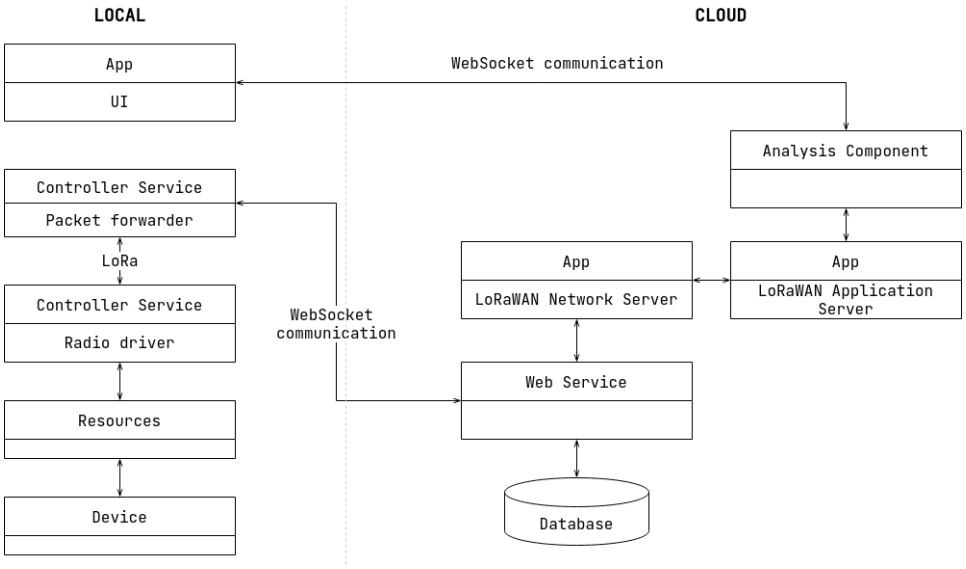


Figure 4.8: Deployment design

4.5.7 Functional view specification

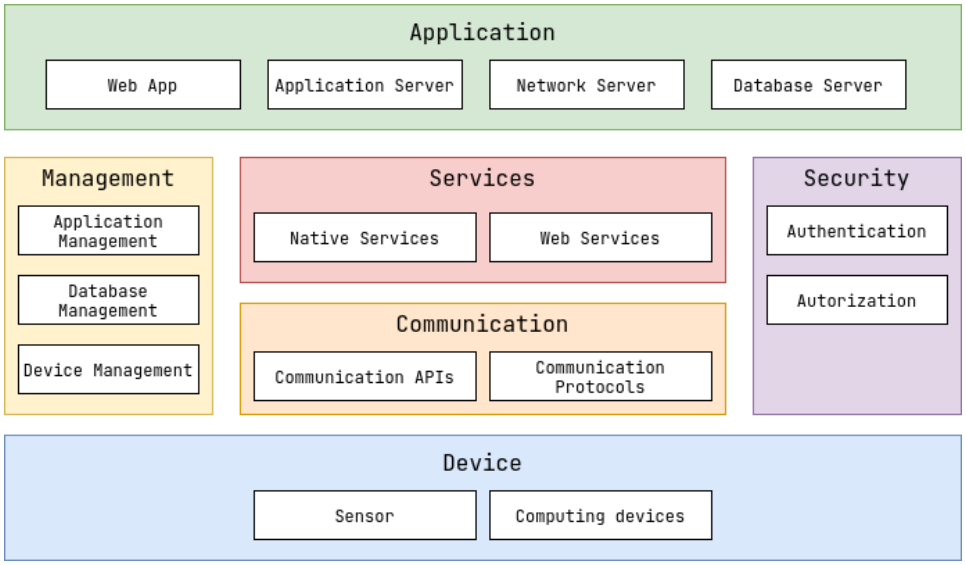


Figure 4.9: Functional Groups

The functional view serves as a way to logically group functionalities with instances of concepts defined in the domain model, or to provide information related to these concepts. The FG, as defined in (Bahga and Vijay 2014), with added mappings for the MIoT subsystem can be logically organized as shown in Figure 4.9, and can be described are as follows:

- **Device:** Contains devices for monitoring and control. For the MIoT subsystem, this is the end-nodes with mounted sensors.

- **Communication:** Handles the communication for the IoT system, including the protocols that form the backbone of the IoT system and enable network connectivity. In the MIoT subsystem, this is, in order starting with the end-node, the LoRa radio link, WebSockets over WiFi for the backhaul link, and MQTT for the remaining components.
- **Services:** Includes various services involved in the IoT system such as services for device monitoring, device control services, data publishing services, and services for device discovery. The LoRa link requires a service interface on the device which handles the message preparation and transmission, and the gateway will need a similar interface which forwards these received messages to an intended LNS.
- **Management:** Includes all functionalities that are needed to configure and manage the IoT system. In particular, the device and gateway will need to be able to be remotely controlled and reconfigured on-demand.
- **Security:** Includes security mechanisms for the IoT system. In this case, the two join-procedures for activation and personalization (see Section 2.5.6).
- **Application:** Includes applications that provide an interface to the users to control and monitor various aspects of the IoT system.

We then use the IoT level specification to map the implementation-specific entities to an appropriate FG, as shown in Figure 4.10, which will later be used to specify the operational view as described in Section 4.5.8.

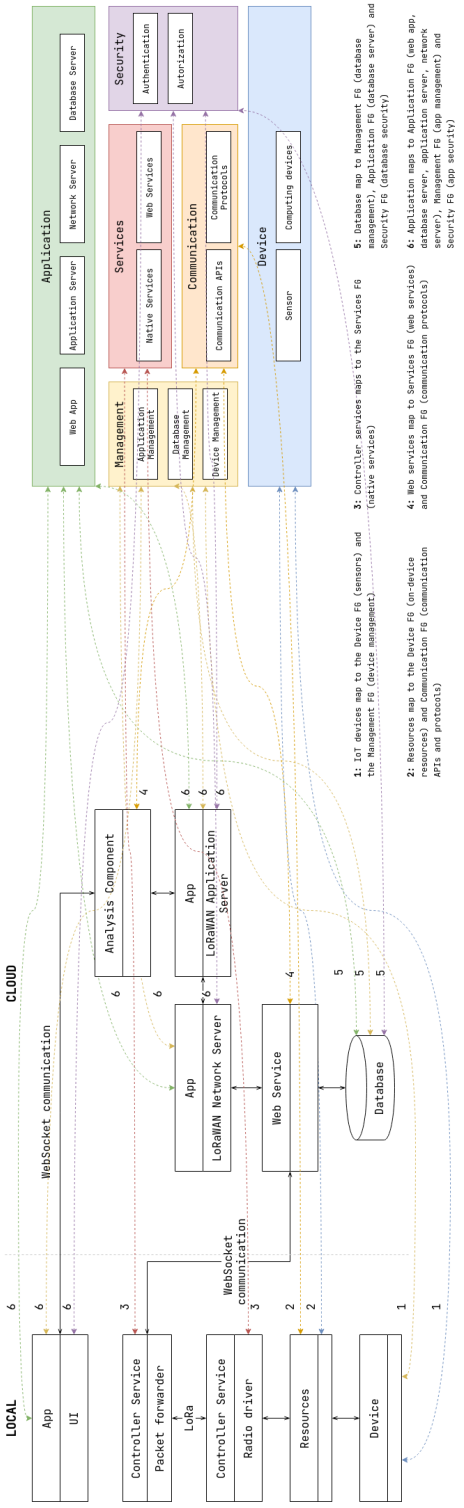


Figure 4.10: Mapping deployment level to functional groups

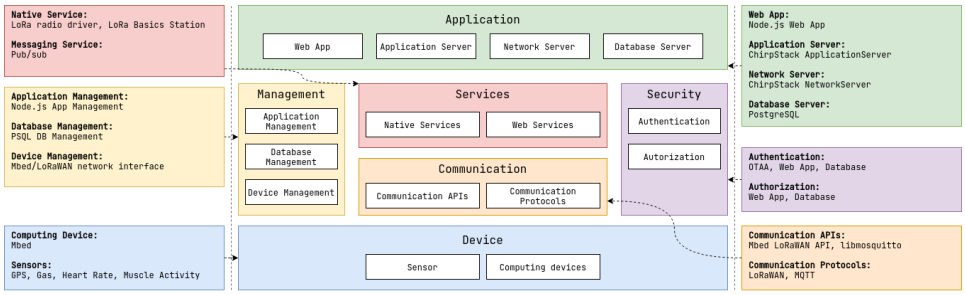


Figure 4.11: Mapping functional groups to operational view

4.5.8 Operational view specification

The operational view uses the identified Function Group mappings to describe concrete options for an operational implementation, such as service hosting, storage, devices, applications and so forth, as shown in Figure 4.11. At the edge, we chose Arm Mbed and its LoRaWAN API (Arm Mbed 2020) for reasons outlined in Section 4.4.2.

At the backend, we chose ChirpStack due to the lenient license associated with it and its large community, which means that it can be freely used and support is easy to obtain. Furthermore, MQTT can be realized using several different broker implementations, which all adhere to the standard. However, different implementations, while standard compliant, exhibit differences when it comes to efficiency of message dissemination. Since research has shown that Mosquitto is the most efficient broker implementation (Biswajeeban and Biswaranjan 2020), we chose that for as the broker supporting the data integration component. The data integration component's responsibility is to filter the contents of the LoRaWAN frames to contain only the data fields described in Section 4.5.1, thereby leaving only a minimal and simple data set for the UI to handle. Thus, the data integration component will effectively act as middleware, as depicted in Figure 4.12.

Finally, at the application level, the data should be presented to the user in a manner appropriate for their role in the military organization. Like in the interviews, we are considering the first levels of authority in operational settings, namely the GFC and OpsOff. Hence, the UI will resemble a

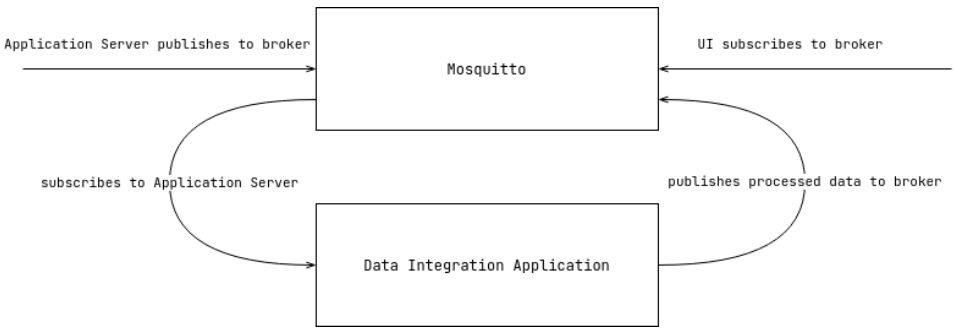


Figure 4.12: Data Integration application

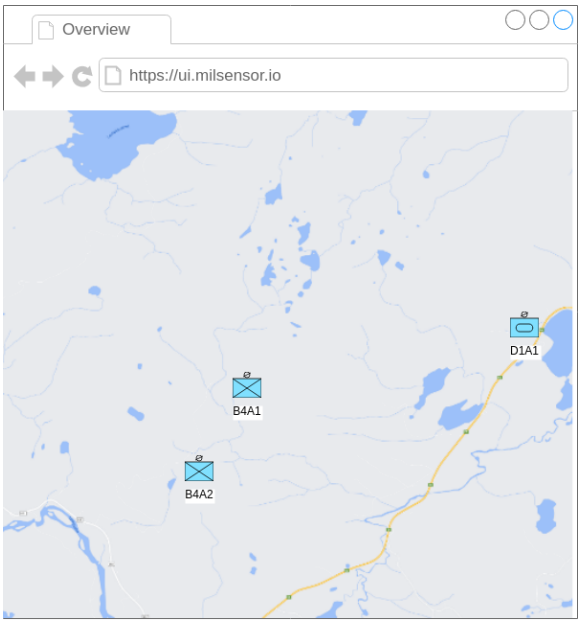


Figure 4.13: UI wireframe

traditional BMS as depicted in Figure 4.13, albeit without considerable regards towards UI or UX best practices, as this is outside the scope of this thesis.

4.6 Summary

In this chapter, we discussed both subject-based and technical considerations towards a MIoT system implementation. Notably, the presence of EW threats and cluttered RF spectrum dictates that best practice towards maintaining tactical advantage while providing augmented SA to target recipients is to keep transmissions as short as possible, both in terms of payload sizes and duration on the air.

These key findings was then a core factor in creating a system design, which followed the methodology for IoT system development steps 1 through 8 as described in Section 1.5. This design methodology aided in developing a logical system design that specifies a complete IoT data pipeline from device to UI, including its connectivity protocols. In the next chapter, the implementation process of the design presented here is presented, along with an evaluation of the prototype wearable.

Chapter 5

Implementation and evaluation

In this chapter, we will cover the implementation process and describe in detail how the technical challenges outlined in Chapter 4 were solved. The chapter is organized in tree main Sections; technical implementation process, technical evaluation, and finally subject evaluation. A second round of semi-structured interviews were conducted using the same informants from the fact finding interview, in support of the subject evaluation.

5.1 Technical implementation process

In accordance with the specification described in Section 4.5.8 which outlines the concrete technologies we want to use to build the MIoT subsystem, we see in Figure 5.1 where these were implemented based on the high-level architecture described in Section 4.2. These technologies will in the following sections be described in the context of the proposed reference architecture described in Section 2.2, namely the *IoTNetWar architectural framework* reference model.

5.1.1 Physical Sensing Layer

The physical layer outlines a number of sensors and actuators, classified under the labels “weapon" or “personnel body", of which only the latter applies in this particular subsystem, namely GPS, ECG, and EMG. The local network (i.e. between the integration component and the devices) use serial communication, namely Universal Asynchronous Receiver-Transmitter (UART) and Inter-Integrated Circuit (I2C), and analog signals (i.e. voltage level readings).

Integration component

For this thesis, two Arm Mbed OS enabled DISCO-L072CZ-LRWAN1 development boards, shown in Figure 5.2, were purchased to function as the integration platform. This board has a wide range of header connectors, of which the inner rows are Arduino Uno Revision 3 compatible. A

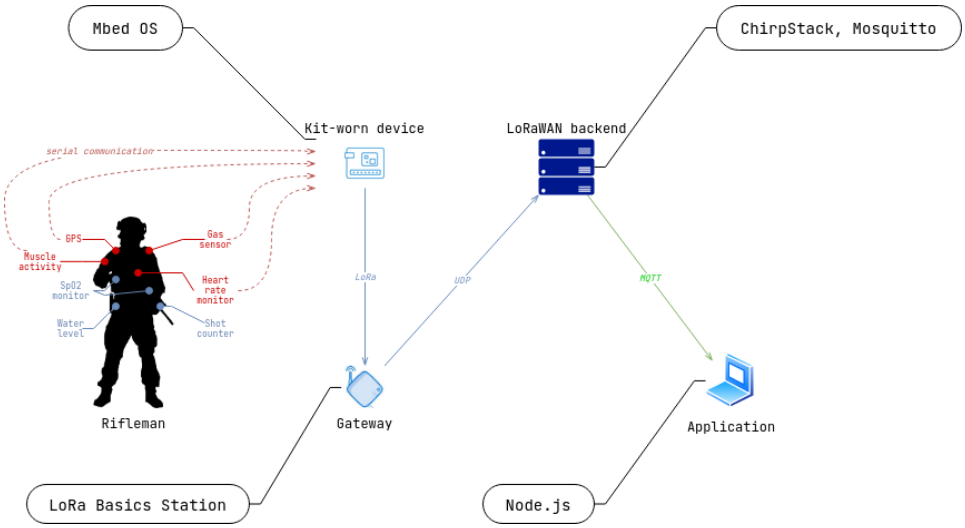


Figure 5.1: Soldier wearable implementation technologies

complete overview of the board can be seen at the Mbed OS board overview (Arm Mbed 2017a).

The main-thread on the device is based on the official Mbed LoRaWAN example implementation (Arm Mbed 2017b), slightly modified to handle message transmission and reception in accordance to our use case. Before it can be successfully initialized, a configuration file which specifies LoRa radio module settings and LoRaWAN settings needs to be specified (Arm Mbed 2021c). In accordance with the hardware for the DISCO L072-LRWAN1 board, it should be specified as shown in Listing 5.1.

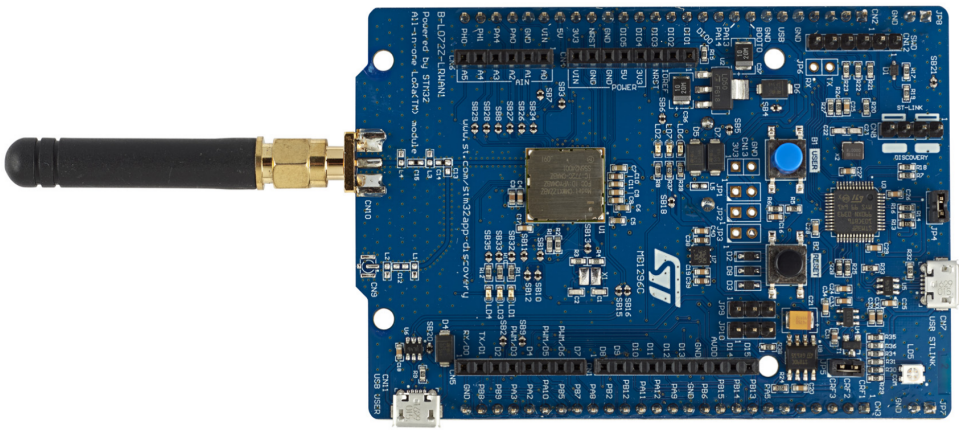


Figure 5.2: DISCO-L072CZ-LRWAN1 development board

Listing 5.1: Mbed configuration file

```
{
    "config": {
        "lora-radio": {
            "value": "SX1276"
        },
        "main_stack_size":     { "value": 2048 },
        "lora-spi-mosi":       { "value": "PA_7" },
        "lora-spi-miso":       { "value": "PA_6" },
        "lora-spi-sclk":       { "value": "PB_3" },
        "lora-cs":             { "value": "PA_15" },
        "lora-reset":          { "value": "PC_0" },
        "lora-dio0":           { "value": "PB_4" },
        "lora-dio1":           { "value": "PB_1" },
        "lora-dio2":           { "value": "PB_0" },
        "lora-dio3":           { "value": "PC_13" },
        "lora-dio4":           { "value": "NC" },
        "lora-dio5":           { "value": "NC" },
        "lora-rf-switch-ctl1": { "value": "NC" },
        "lora-rf-switch-ctl2": { "value": "NC" },
        "lora-txctl":          { "value": "PC_2" },
        "lora-rxctl":          { "value": "PA_1" },
        "lora-ant-switch":     { "value": "NC" },
        "lora-pwr-amp-ctl":    { "value": "PC_1" },
        "lora-tcxo":           { "value": "PA_12" }
    },

    "target_overrides": {
        "*": {
            "target.printf_lib":                    "std",
            "platform.stdio-convert-newlines":      true,
            "platform.stdio-baud-rate":             115200,
            "platform.default-serial-baud-rate":    115200,
            "platform.cpu-stats-enabled":           true,
            "lora.over-the-air-activation":         true,
            "lora.duty-cycle-on":                   true,
            "lora.phy":                             "EU868",
            "target.components_add":                ["SX1276"],
            "lora.device-eui":                      "{ 0x0f, 0xd7, 0xc5, 0
                ↪ xc6, 0x89, 0xbe, 0x7f, 0xc0 }",
            "lora.application-eui":                 "{ 0x00, 0x00, 0x00, 0
                ↪ x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0
                ↪ x00, 0x00, 0x00, 0x00 }",
            "lora.application-key":                 "{ 0x4d, 0xdb, 0xe7, 0
                ↪ x5f, 0xda, 0x27, 0xe1, 0x66, 0x54, 0x93, 0xf6, 0xd3, 0
                ↪ xac, 0xf4, 0xc2, 0xbc }"
        }
    },
    "macros": ["MBEDTLS_USER_CONFIG_FILE=\"mbedtls_lora_config.h\""]
}
```

Note the first line following the attribute `target_overrides`, namely `target.printf_lib`, which specifies that Mbed OS should use the full standard `printf` library. As of version 6.0, this is disabled by default to save ROM usage (Kamba-Mpiana et al. 2019). However, for debugging purposes, it was found useful to keep this enabled.

Finally, to enable CPU statistics (i.e. sleep, deep sleep, and active time metrics), `platform.cpu-stats-enabled` must be set to `true`. The Sleep Manager API will by default put the device to sleep when no threads are active, where one of the sleep modes will be activated based on whether (Arm Mbed 2019):

- Low power tickers are available, which are required to wake up the MCU.
- Tickless mode is enabled, which enables the system to function without a running SysTick, a standard timer on Cortex-M cores which raises an interrupt with a set frequency.
- Any active bus or driver that relies on high-frequency clock is active, such as a Timer, asynchronous I2C, or any class inheriting from `SerialBase`.

If any of the items listed above are present, “regular" sleep mode will be activated in which the core system clock will be disabled. If none of the items are present, deep sleep mode will be activated, in which all high-frequency clocks will be disabled, including SysTick.

Location awareness: GPS

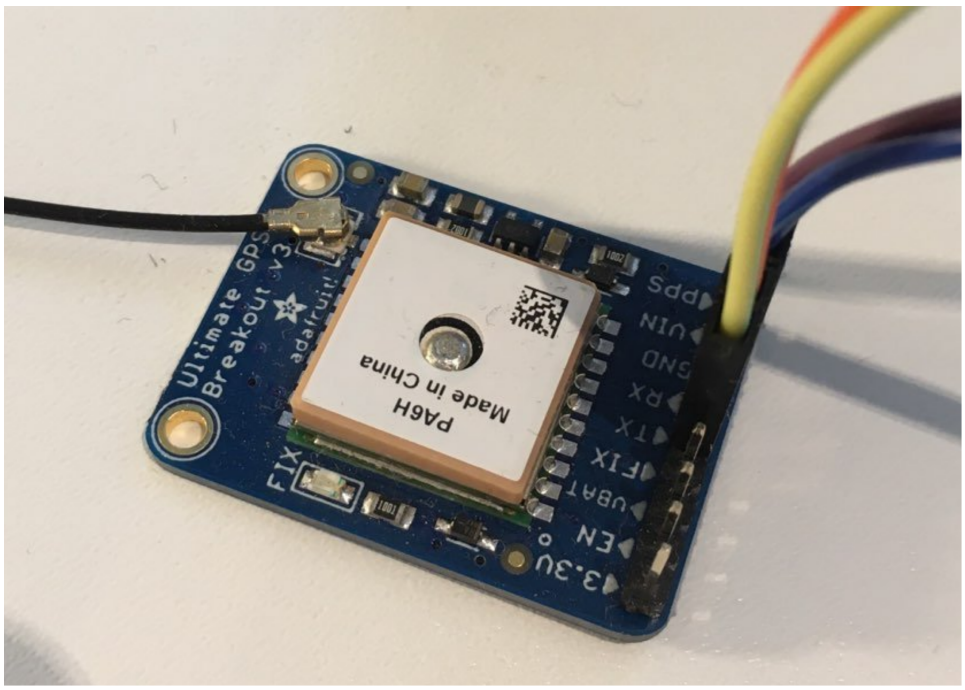


Figure 5.3: Adafruit Ultimate GPS Breakout v3

To enable geographic position reporting, an Adafruit Ultimate GPS Breakout v3 (Adafruit Industries 2012) was purchased due to its platform

support through existing community-developed libraries and its feature-rich capabilities.

This GPS module embeds a built-in 64K logger and features command-receptive functionality so as to tweak its behavior, and outputs standard NMEA 0183 sentences (NMEA Standards Committee 2018) containing location, speed, and altitude data. To communicate with the development board, it requires one UART interface using a fixed baud rate at 9600. The module can be seen in Figure 5.3.

To enable software-controlled information flow, the SerialGPS library (Arm Mbed OS Components Team 2014) was used as a starting point. The library uses the Serial API to communicate over UART, which as of Mbed OS version 6.0.0 is deprecated in favour of the new UnbufferedSerial and BufferedSerial APIs. Thus, some modifications were required. According to the documentation, serial communication should use BufferedSerial for data transfers, which provides UART functionality using software buffers to send and receive bytes.

The raw output from the GPS is a continuous data stream of all NMEA formats. Since these sentences are of varying length, we cannot use a fixed-size buffer to acquire the values. However, each sentence starts with a $ character and terminates with a newline character, which could be used to extract NMEA sentences, as shown in Code Listing 5.2. Note the `_gps` variable which points to the `BufferedSerial` object.

Listing 5.2: Data capture from GPS serial interface

```
void GPSController::getline()
{
    char ch;
    uint8_t idx = 0;

    // Wait for start of sentence
    while(read_char() != '$');

    // Read character until EOL
    while((ch = read_char()) != '\r')
    {
        this->_nmea[idx] = ch;
        idx++;
    }
}
...
inline char read_char()
{
    char ch;
    this->_gps->read(&ch, 1);
    return ch;
}
```

Once this code returns, it will be checked against the GPRMC format which carry a minimal data set for position information. This function would subsequently set the `fix` variable to `true` in the event a fix or lock have been acquired, before converting the values from DMS (Decimal Minutes Seconds) to Decimal Degrees (DD), and finally a position deviation check which, if true, stores the current position data and adds them to the message payload, as shown in Code Listing 5.3.

Listing 5.3: NMEA sentence matching

```
void GPSController::getline()
{
    char ch;
    uint8_t idx = 0;

    // Wait for start of sentence
    while(read_char() != '$');

    // Read character until EOL
    while((ch = read_char()) != '\r')
    {
        this->_nmea[idx] = ch;
        idx++;
    }
}

void GPSController::read(CayenneLPP* clpp)
{
    float lat, lon;

    this->_tmr.start();

    while(this->_tmr.elapsed_time() <= std::chrono::seconds(10))
    {
        getline();
        // GPRMC sentence: UTC time, status, lat, N/S, lon, E/W
        if(sscanf(this->_nmea, "GPRMC,%f,%c,%f,%c,%f,%c", &this->_UTCtime,
            ↪  &this->_status, &lat, &this->_NS, &lon, &this->_EW) > 0)
        {
            if(this->_status == 'A')
            {
                this->_fix = true;

                lat = convert(lat);
                lon = convert(lon);

                if(this->_NS == 'S') { lat *= -1.0; }
                if(this->_EW == 'W') { lon *= -1.0; }

                if(position_deviation(this->_lat, this->_lon, lat, lon))
                {
                    printf("␣Threshold␣met␣--");

                    this->_lat = lat;
                    this->_lon = lon;

                    clpp->addGPS(LPP_GPS, this->_lat, this->_lon, 0);
                }
                break;
            }
        }
        // Clear the NMEA buffer
        memset(this->_nmea, 0, strlen(this->_nmea));
    }
    this->_tmr.stop();
    this->_tmr.reset();

    if(this->_fix)
    {
        printf("␣Lat␣=␣%f,␣Lon␣=␣%f", this->_lat, this->_lon);
    }
    else
    {
        printf("␣No␣fix");
    }
}
```

DMS takes the form `ddmm.mmmm` for latitude values, and `dddmm.mmmm` for longitude, where `dd(d)` is degrees, `mm` is minutes, and `mmmm` is seconds. To calculate the DD based on a DMS value, simply use the following formula (LatLong.net 2019):

$$DecimalDegrees = degrees + (minutes/60) + (seconds/3600)$$

Which can be put into effective code as shown in Code Listing 5.4, where the `modf` C library function call set the fraction component to the variable `sec` and sets the integer component to the variable `degmin`.

Listing 5.4: Lat-Lon conversion from DMS to DD

```
inline float convert(float dms)
{
    float degmin, sec;
    sec = modf(dms, &degmin);

    float deg = (int) degmin / 100;
    float min = (int) degmin % 100;

    float dd = deg + (min / 60.0) + (sec / 3600);

    return dd;
};
```

The position variance was subsequently calculated by using the Haversine formula, which uses current Lat-Long pairs to measure the distance to previous Lat-Long pairs in meters. Note the `EARTH_RADIUS` macro which we use to calculated distance in meters.

Listing 5.5: Position-to-position distance calculation using the Haversine function

```
#define PI 3.14159265358979323846
#define RADIAN_DEGREES PI / 180
#define EARTH_RADIUS 6371000
#define MIN_DISTANCE_TRESHOLD 2

/**
 * Disclaimer: code taken from https://stackoverflow.com/a/63767823
 */
bool GPSController::position_deviation(float lat, float lon, float
    ↪ current_lat, float current_lon)
{
    lat *= RADIAN_DEGREES;
    lon *= RADIAN_DEGREES;
    current_lat *= RADIAN_DEGREES;
    current_lon *= RADIAN_DEGREES;

    float haversine, temp, min, dist;

    haversine = (pow(sin((1.0 / 2) * (current_lat - lat)), 2)) + ((cos(lat
        ↪ )) * (cos(current_lat)) * (pow(sin((1.0 / 2) * (current_lon -
        ↪ lon)), 2)));

    sqrt(haversine) < 1.0 ? min = sqrt(haversine) : 1.0;
    temp = 2 * asin(min);

    dist = RADIAN_TERRESTRIAL * temp;

    return (dist > MIN_DISTANCE_TRESHOLD);
}
```

Status awareness: Heart Rate sensor

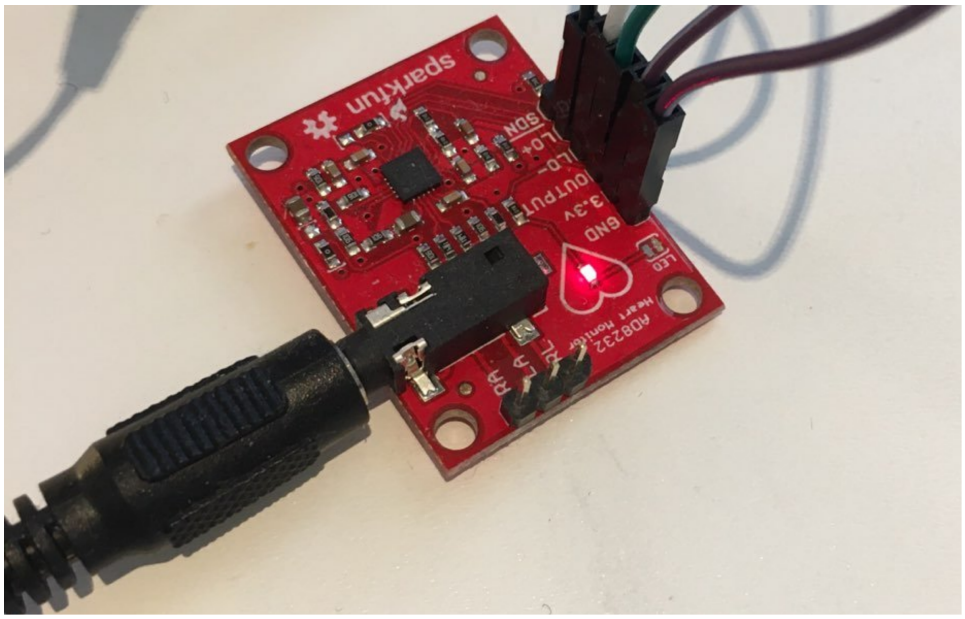


Figure 5.4: AD8233 Heart Rate Sensor

The Sparkfun AD8233 Heart Rate (Sparkfun Electronics 2016b) sensor was found to be the most promising candidate to detect heart beats, largely due to its convenient cable integration for sensor pad placements on the body. As stated in the specification, the HR sensor should be used to calculate a simple BPM value by using ECG, which is not calculated by the sensor itself, as it only returns an analog signal to the connected pin. To set it up with the development board, it would require an `AnalogIn` instance which returns a given voltage acquired from the sensor pads, which would have to be manually interpreted to calculate a BPM. Fortunately, a similar implementation using an optical photo-resistor for BPM calculation (Ionascu 2015) by Mbed user *Mary-Ann Ionascu* fit nicely with this particular sensor, with some minor modifications. This code runs using a `LowPowerTimer` instance which would either run for a maximum of 10 seconds or until a heart rate was found (i.e. minimum 5 beats was successfully detected).

Status awareness: Muscle activity sensor

The MyoWare Muscle Activity sensor (Advancer Technologies 2015) developed by Advancer Technologies was found to be the best option towards recording muscle activity, which outputs an analog signal representing the rectified and integrated signal of the activity of one single muscle (i.e. only the positive voltage values represented by fixed-width time slices). When stacked with the cable shield as shown in Figure 5.5, it enables the use of the same sensor pads as the AD8233 Heart Rate sensor. The placement of the pads should be as shown in Figure 5.6, where the red

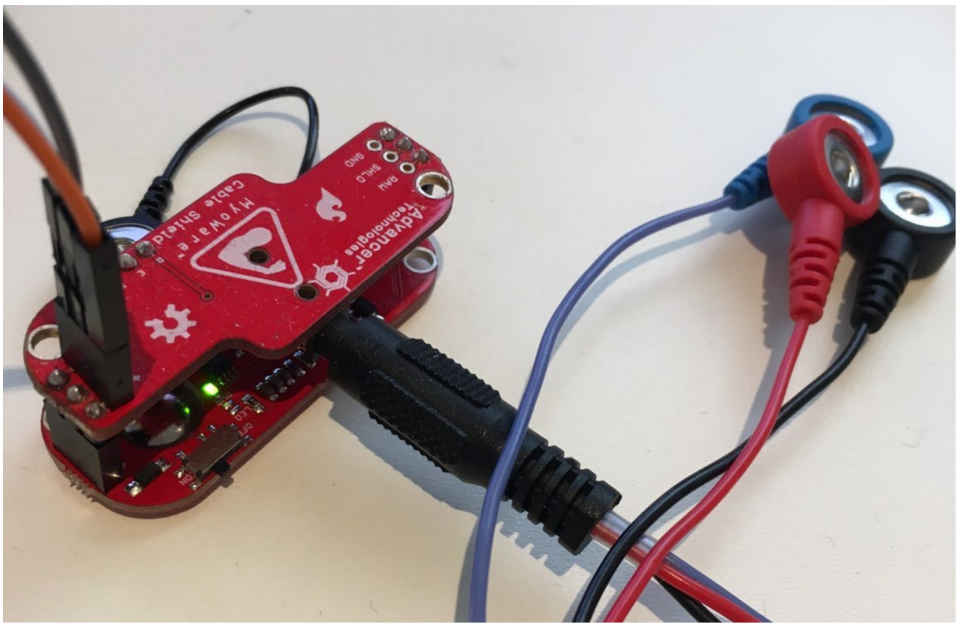


Figure 5.5: MyoWare Muscle Activity sensor stacked with cable shield

and blue pad should be placed on the middle and end of the muscle body, respectively, while the black pad should be placed on a separate section where it should not detect the muscle being measured. This is because this particular pad works as a reference for the two primary pads.

As this sensor also outputs an analog voltage, the code should in the same manner utilize a `LowPowerTimer` instance and average the muscle activity throughout the set time, which is 10 seconds in this case, as shown in Code Listing 5.6. This is however not a clinical MUAP measurement as this would require more sophisticated algorithms, in addition to the fact that we are not continuously measuring the muscle activity. Thus, we implemented a simple averaging measurement to simulate this behavior.

Listing 5.6: Muscle activity calculation

```
1  void MUAPController::read()
2  {
3      float muap      = 0.0;
4      int samples     = 0;
5
6      _tmr.start();
7
8      while(_tmr.elapsed_time() <= std::chrono::seconds(10))
9      {
10         muap += _sig.read() * 100;
11         samples++;
12     }
13
14     _tmr.stop();
15
16     this->_muap = muap / samples;
17
18     clpp.addDigitalOutput(MUAP, this->_muap);
19
20     _tmr.reset();
21 }
```

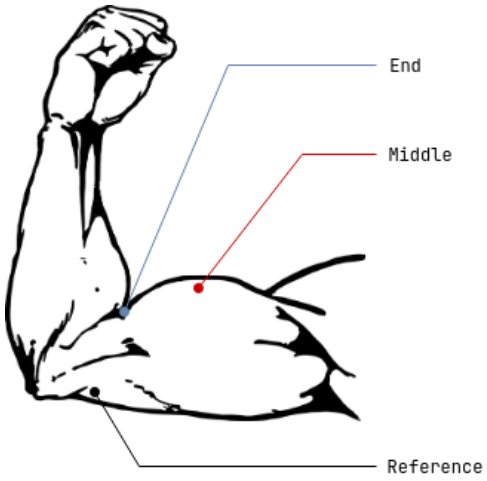


Figure 5.6: Muscle activity sensor pad placements

Environment awareness: Gas sensor

The Grove Multichannel Gas Sensor V2 (Seeed Technology Co. Ltd. 2018) as shown in Figure 5.7 communicates over I2C and require a 3.3V power supply, and qualitatively detects a variety of gases through its four on-board gas detection modules:

- GM102B: NO2 (Nitrogen Dioxide)
- GM302B: C2H5CH (Ethanol)
- GM502B: VOC (Volatile Organic Compounds)
- GM702B: CO (Carbon Monoxide)

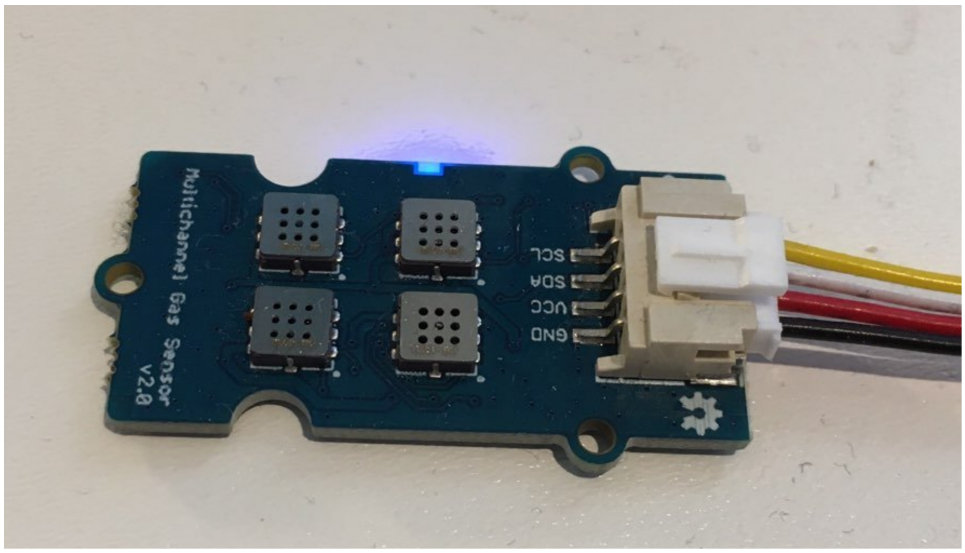


Figure 5.7: Grove Multichannel Gas Sensor V2

The provided library for this particular sensor is built for Arduino (Weng 2019), thus requiring to be ported to Mbed OS for compatibility. This was solved in large part by changing the Arduino-specific `TwoWire` and `SoftwareWire` interface libraries with the Mbed OS I2C API (Arm Mbed 2021a).

5.1.2 Gateway Communication Layer

The gateway communication layer is the link between the physical sensors and the data processing layer, here named the C4ISR Management Layer, which commonly resides in the cloud. The gateway OS is Raspberry Pi OS (previously known as Raspbian), a Debian-based OS built for Raspberry Pi SBCs, which will use LoRa communications between itself and the end-nodes, and Wi-Fi to communicate with local machines running the backend components.

A prototype gateway was built using an IMST iC880A LoRa concentrator (IMST 2021) and a SMA antenna with 2 dBi gain, connected to a Raspberry Pi 3B using a LinkLab LoRa gateway shield (CH2I 2018) as shown in the final assembly in Figure 5.8.

According to the official documentation, LoRa Basics Station is locally configured using two files, namely `station.conf` and `tc.uri` , used to enable configuration of the radio HAL (Hardware Abstraction Layer) and local station configuration by using pre-determined reference designs

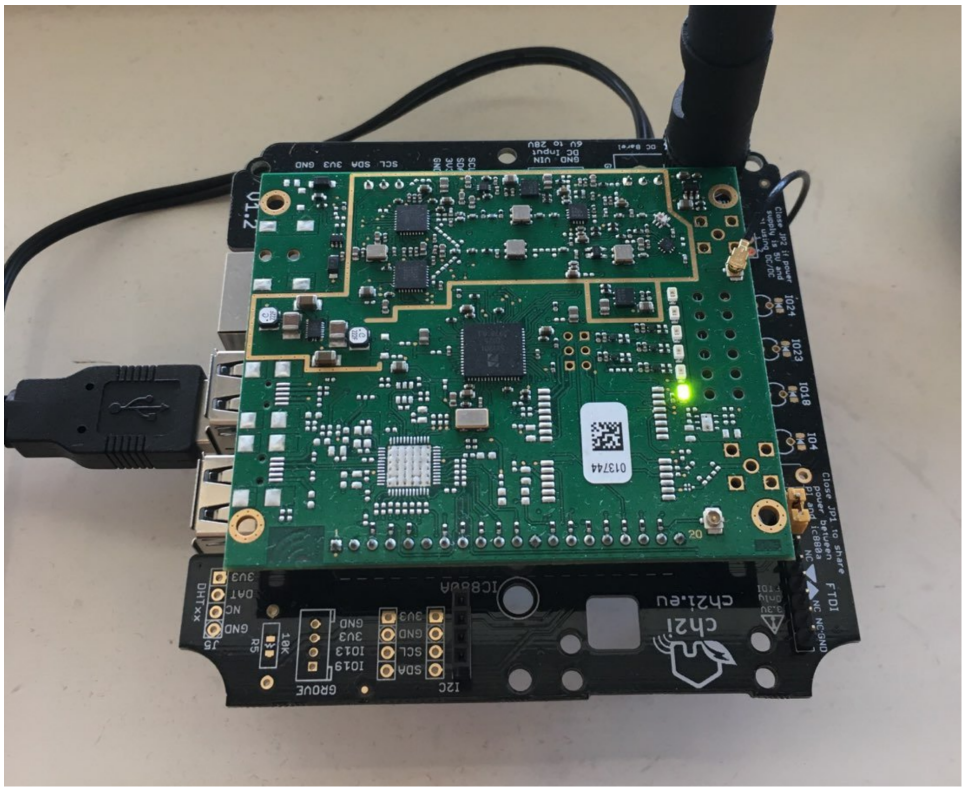


Figure 5.8: Gateway assembly

and to specify the LNS endpoint with which the gateway will attempt to communicate, respectively. At the time of writing, the available station configuration references are as follows:

- Concentrator v1.5: Used for single SX1301 radios over Serial Peripheral Interface (SPI)
- Concentrator v2: Used for multiple SX1301 radios over SPI
- Picocell: Like Concentrator v1.5, but using USB
- Corecell: Used for single SX1302 radios over SPI

The IMST iC880A LoRa concentrator is based on the SX1301 chip. Thus, we are bound to either v1.5 or v2 design. For simplicity, we will use the v1.5 design since it handles a single concentrator only. The configuration file `station.conf` is listed as follows:

Listing 5.7: LoRa Basics Station configuration

```
{
    /* alias for radio_conf */
    "SX1301_conf": {
        "lorawan_public": true,
        "clksrc": 1,
        "device": "/dev/spidev0.0",
        "pps": false,
        /* RFCONF object -- radio_0 on the iC880A concentrator  */
        "radio_0": {
            "type": "SX1257",
            "rssi_offset": -169.0,
            "tx_enable": true,
            "antenna_gain": 0
        },
        /* RFCONF object -- radio_1 on the iC880A concentrator */
        "radio_1": {
            "type": "SX1257",
            "rssi_offset": -169.0,
            "tx_enable": false
        }
    },
    "station_conf": {
        "routerid": "GATEWAY_EUI_PLACEHOLDER",
        /* Runs before the concentrator starts -- ensures that it will be
            ↪ in a clean state  */
        "radio_init": "iC880A-SPI_reset.sh",
        "log_file":  "/var/log/station.log",
        /* XDEBUG,DEBUG,VERBOSE,INFO,NOTICE,WARNING,ERROR,CRITICAL */
        "log_level": "INFO",
        "log_size":  10e6,
        "log_rotate":  3
    }
}
```

The first item, `SX1301_conf`, is an alias for `radio_conf` and is therefore treated as the RAL (Radio Abstraction Layer) by the JSON parser when we execute the gateway binary. Organized under the RAL we find the two radio modules embedded on the concentrator, namely two SX1257 chips, the SPI device with which the concentrator communicates with the host system, and an optional GPS enabler. In the second primary item, `station_conf`, we specify the DevEUI for the concentrator and logging

configuration. In addition, there is support for initialization logic using the `radio_init` field which points to a shell scripts that can be used to reset the concentrator to a clean state prior to executing the software, which is required for certain hardware, as it is a known bug that the LoRa packet forwarder may be unable to start unless the concentrator is in a clean state (MojIoT Lab 2020).

Finally, the gateway is preferably installed as a service which enables features such as automatic program launch on startup and restarts on failure. In this case, the specification shown in Listing 5.8 was used.

Listing 5.8: LoRaWAN gateway service

```
[Unit]
Description=LoRaWAN gateway

[Service]
WorkingDirectory=/opt/basicstation/build-rpi-std/bin
ExecStart=station
SyslogIdentifier=lorawan-gateway
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

Finally, the station executable needs to know specifics regarding the LNS or CUPS protocol. In this thesis, the gateway was configured to use LNS only, which can be enabled by simply adding the WebSocket endpoint and the port to the controlling LNS in the file `tc.uri`.

5.1.3 C4ISR Management Layer

This layer is tasked with specifying the general backend component for a MIoT system, in particular data analysis and visualization. However, as outlined in Section 1.4, Big Data and full-fledged data analysis is outside the scope of this thesis. Thus, a simple application tasked with filtering the data produced by the Application Server was implemented, effectively acting as a middleware between the LoRaWAN backend and the UI.

LoRaWAN backend

The backend infrastructure was set up using a standalone Raspberry Pi 4 installed with the full ChirpStack stack, namely the Network Server, Application Server, and Gateway Bridge. The Gateway Bridge is ChirpStack-specific, which ensures communication with the gateway over UDP, and subsequently publishes gateway traffic through MQTT using an internal broker, which the Network Server subscribes to. Alternatively, the gateway itself can be installed with the Gateway Bridge, which will then publish gateway traffic directly to the internal broker, as shown in the architectural description shown in Figure 5.9.

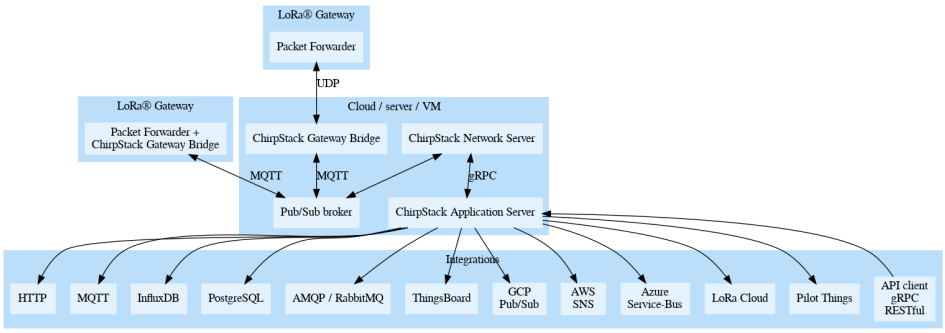


Figure 5.9: ChirpStack architecture (credit: (Brocaar 2019))

According to the official documentation, the importance lies with the configuration files to each of the three components to match the local environment. The excerpt from the `chirpstack-network-server.toml`, shown in Listing 5.9, shows the main fields that must be configured for the Network server to work properly.

Listing 5.9: Network Server configuration excerpt

```
[postgresql]
dsn="postgres://PSQL_CHIRPSTACK_NS_USER:PSQL_CHIRPSTACK_NS_PW@localhost/
    ↪ chirpstack_ns?sslmode=disable"

[redis]
url="redis://localhost:6379

[network_server]
net_id="000000"

[network_server.band]
name="EU868"

[network_server.api]
bind="0.0.0.0:8000"

[join_server]
server="http://localhost:8003"
```

This configuration specifies local database connections, the ISM band, and the join server to be used. At the time of writing, the Application Server acts as the Join server, requiring the JoinEUI on the end-nodes to be set to all 0s. Likewise, in the configuration file for the Application Server, we specify the bind address for the Join Server to be itself using the same port, as shown in Listing 5.10.

Listing 5.10: Application Server configuration excerpt

```
[postgresql]
dsn="postgres://PSQL_CHIRPSTACK_AS_USER:PSQ_CHIRPSTACK_AS_PW@localhost/
    ↪ chirpstack_as?sslmode=disable"

[redis]
url="redis://localhost:6379"

[join_server]
bind="0.0.0.0:8003"
```

The Gateway Bridge must be configured to match the packet forwarder used on the gateway, in this case, LoRa Basics Station. The excerpt from the `chirpstack-gateway-bridge.toml`, shown in Listing 5.11, shows the local configuration.

Listing 5.11: Gateway Bridge configuration excerpt

```
[backend]
type="basic_station"

  [backend.basic_station]
  bind=":3001"

  region="EU868"

  frequency_min=863000000
  frequency_max=867000000

  [[backend.basic_station.concentrators]]

    [backend.basic_station.concentrators.multi_sf]
    frequencies=[
      868100000,
      868300000,
      868500000,
      867300000,
      867500000,
      867700000,
      867900000,
    ]

    [backend.basic_station.concentrators.lora_std]
    frequency=868300000
    bandwidth=250000
    spreading_factor=7

    [backend.basic_station.concentrators.fsk]
    frequency=868800000
```

Once all three components have been properly configured, we can go back to the gateway and specify the desired WebSocket endpoint which points to the Gateway Bridge, which will have to be inserted into the file `tc.uri`, shown in Listing 5.12.

Listing 5.12: LoRa Basics Station WebSocket endpoint

```
ws://192.168.1.203:3001
```

Data Integration

The Data Integration application was built as a simple C++ program utilizing the Mosquitto C-library (Eclipse Foundation 2010) and Nlohmann JSON (Lohmann 2013) in order to filter the contents of the data produced by the Application Server. Essentially, it uses a Mosquitto instance to acquire the uplink data, extracts elements that are of interest for the end users and subsequently creates a minimal JSON array, before finally publishing said JSON array on a given topic which the UI subscribes to. An excerpt of the code of interest is shown in Code Listing 5.13.

Listing 5.13: Data Integration code excerpt

```
template<typename T>
void build_msg(json* j, std::string key, T value)
{
    j->push_back({key, value});
}

void uplink_handler(struct mosquitto* mosq, const struct mosquitto_message
    ↪ * msg)
{
    json* j_msg = new json();
    auto j_in = json::parse((char*) msg->payload);
    build_msg<std::string>(j_msg, "direction", "uplink");

    try
    {
        // devEUI and associated callsign
        auto devEUI = j_in.at("devEUI").get<std::string>();
        std::string cs = get_callsign(devEUI);
        build_msg<std::string>(j_msg, "devEUI", devEUI);
        build_msg<std::string>(j_msg, "callsign", cs);

    }
    catch(const std::exception& e) { std::cout << ", no ID data"; }

    try
    {
        // location
        auto lat = j_in.at("object").at("gpsLocation").at("136").at("
            ↪ latitude").get<float>();
        auto lon = j_in.at("object").at("gpsLocation").at("136").at("
            ↪ longitude").get<float>();
        build_msg<float>(j_msg, "lat", lat);
        build_msg<float>(j_msg, "lon", lon);
    }
    catch(const std::exception& e) { std::cout << ", no NS data"; }

    try
    {
        // BPM and MUAP
        auto bpm = j_in.at("object").at("digitalInput").at("0").get<int>()
            ↪ ;
        auto muap = j_in.at("object").at("digitalOutput").at("1").get<int
            ↪ >();
        std::string health_status = interpret_biometric_data(bpm, muap);
        build_msg<std::string>(j_msg, "health_status", health_status);
    }
    catch(const std::exception& e) { std::cout << ", no BIO data"; }

    std::string fwd_msg = j_msg->dump();
    publish_message(mosq, PUB_TOPIC, fwd_msg.length(), fwd_msg.c_str(), 0)
        ↪ ;
    delete j_msg;
}
```

Ideally, a data structure linking the DevEUI to the callsign of the wearer would be in place so as to correctly associate all incoming data to known devices. In this implementation, the function returning the callsign is just a placeholder, as it is built to only return one callsign for the prototype.

Note that to enable WebSocket connections to the Mosquitto broker, which the UI as a web application used, the broker must be configured as such. Fortunately, Mosquitto allows both plain MQTT (using default port 1883) and WebSocket (using port 9001 in this particular case) connections simultaneously, as shown in the configuration in Listing 5.14. Note that the

broker is configured to allow any connections as well, as specified with the `allow_anonymous` field.

Listing 5.14: Mosquitto broker configuration excerpt

```
listener 1883

listener 9001
protocol websockets

allow_anonymous true
```

5.1.4 Application Layer

At the application layer, we see from the reference architecture that the system is location-, status-, and environment-aware through GPS, biometrics, and gas sensing, where the purpose is military personnel tracking and health status monitoring of the wearer. Throughout the whole pipeline, the communication protocol is largely MQTT, with the exception of the backhaul link between the gateway and the backend, where WebSockets are in use.

The UI was built as a simple web application based on an example Node.js integration by LoRaWAN Academy (Semtech Corporation 2019a), but extended to use the Eclipse Paho MQTT JS library (Eclipse Foundation 2013) to receive the filtered uplink data and to schedule downlink commands, in addition to using Google Maps API (Google 2009) to display a map showing sensor location and status.

5.1.5 Code

The code excerpts shown in the previous section were taken from the release branch of their respective code repositories. Their URLs are as follows, including the repository containing the simulator used for the feedback interview:

- End node: https://gitlab.com/ffi-miot/end-node/-/tags/v1.0
- Gateway setup script: https://gitlab.com/ffi-miot/gateway-ic880a/-/releases/v1.0
- Data integration application: https://gitlab.com/ffi-miot/dataintegration/-/releases/v1.0
- UI: https://gitlab.com/ffi-miot/ui/-/releases/v1.0
- Simulator: https://gitlab.com/ffi-miot/simulator/-/tags/v1.0

5.2 Technical evaluation

This section covers technical assessment by the author of this thesis, and summarizes experiences gained from the development process.

5.2.1 Platform development

Using the official Arm Mbed LoRaWAN example as a starting point, the implementation process was mostly focused on finding proper means to acquire sensor readouts in the context of the LoRaWAN event loop. Most notably, the sensors were initially integrated with the platform through controllers which worked as wrappers for the imported sensor libraries so as to provide common logic for packing sensor data, which used manual bit packing for the various data types for the sensor readouts. One example of doing so was through the function specified in Code Listing 5.15, where a pointer to a custom `lora_msg_t struct` was passed along with a given index for the payload it contained and the sensor data. The `uint8_t` fragments making up the sensor value were copied into the payload field of said `struct` by applying a proper bit mask and shifting the values to the lower 8 bits. Note that in this particular case, this function takes `uint32_t` converted floating point values (i.e. 6 decimals, a convenient precision for GPS data as given by latitude and longitude format).

Listing 5.15: Manual bit packing

```
1 static size_t bit_pack_message(lora_msg_t* msg, int idx, uint32_t data)
2 {
3     msg->payload[idx++] = (data & 0xFF000000) >> 24;
4     msg->payload[idx++] = (data & 0x00FF0000) >> 16;
5     msg->payload[idx++] = (data & 0x0000FF00) >> 8;
6     msg->payload[idx++] = (data & 0x000000FF);
7     return sizeof(msg->payload);
8 }
```

Cayenne LPP in itself worked very well and proved easy to use. However, it was found that the data identifiers provided through the library came a bit short. It offered a ready-to-use function call for packing GPS data, but it lacked identifiers for both biometric readings and gas measurements. As a consequence, generic identifiers were used instead, namely `DigitalInput` for BPM, `DigitalOutput` for MUAP, and `Presence` for gas detections.

5.2.2 Sensor integration

The particular board used in this thesis worked well as an integration platform due to its many peripheral connectivity options. However, it was found that some of the pins conflicted with each other, rendering some interfaces unusable. For example, the Serial2 TX/RX pairs conflicted with the STLink connection, requiring removal of the SB28 and SB29 solder bridges and thus loosing said STLink connection, and the Serial1 TX/RX pairs conflicted with both I2C and on-board LEDs. Fortunately, the sensor setup did not require more than the board could offer despite these

interface conflicts, namely 1 UART for the GPS, 1 I2C for the gas sensor, and 2 analog inputs for the biometric sensors, in addition to the power supply and ground pins.

The GPS required the most attention with respect to implementation and integration, as geographic locations are arguably the most important information the system provides in the context of military applications, in addition to also carrying the highest level of complexity among the raw sensor data in the prototype. Some issues regarding transmitting PMTK commands was experienced, as the module did not seem to respond to any of the given commands. As a consequence, a `LowPowerTimer` instance was added to give the module time to acquire a recognized NMEA sentence from the serial interface. When enabling continuous output from the connected UART interface, the NMEA sentences will be produced in a certain order, as shown in Listing 5.16.

Listing 5.16: NMEA sentences from the Serial interface

```
$GPGGA,105821.000,5938.9031,N,01049.6273,E,2,08,1.09,112.2,M,40.6
    ↪ ,M,0000,0000*62
$GPGSA,A,3,31,29,32,25,03,02,12,06,,,,,2.21,1.09,1.92*0A
$GPGSV,3,1,09,25,80,235,47,12,53,098,26,02,40,089,24,29,34,206,37*77
$GPGSV,3,2,09,31,33,302,46,06,29,047,29,32,24,245,36,49,22,186,33*76
$PS,,,90,9303*62,4,93,4253,92,8,370,006
```

The Arduino library for the gas sensor was found to be relatively easy to port to Mbed OS, where the significance lied with how Mbed OS handled I2C addresses. These addresses are usually 7-bits long, however in Mbed OS, they are 8 bits, requiring to be left-shifted by 1 bit before being written to the device.

However, despite a large number of attempts at detecting alcohol fumes by soaking a cloth sponge with hand sanitizer and keeping it on top of the gas sensor modules, it never successfully detected the presence of such a gas. Acquiring some form of canisters containing any of the other detectable gases (e.g. CO or NO2) as given by the documentation (Seeed Technology Co. Ltd. 2018) proved to be rather difficult. As such, no other gases were tested, and the usability of this particular sensor therefore remains undetermined.

The biometric sensors both proved simple to assemble and integrate, but proved challenging to properly interpret based on the returned analog signal. Using the above-mentioned example integration for BPM calculation, the calculated heart rate seemed to be quite accurate when compared with manual pulse calculation, thus deemed effective for its purpose. However, the sensor itself is very sensitive, as it produced valid BPM readouts even when the pads were not placed on the body. The same applies for the muscle activity sensor, where at times maximum voltage was continuously provided to the software controller despite the pads not being placed on the body. These sensors are also prone to external noise, where other muscle activity in the body would affect the signal and could lead to very misleading calculations. Pad placement on the body was found to be a major factor for calculating the correct BPM and MUAP values as accurately as possible. For the heart rate sensor, it was found that it

produced the most accurate readings when the blue and black pad were placed high up on the chest, close to the brachial artery (i.e. upper arm), and the red pad was placed low on the abdomen, right above the right hip, as described in the AD8232 hookup guide (Sparkfun Electronics 2016a).

TX logic

The `EventQueue`, which largely control application behavior, is implemented in such a way that it will continue to send messages after the mandatory wait-period has passed. This was found to be sufficient for the prototype development, where a simple check on the global `emcon_active` variable prior to acquiring sensor readouts was in place in order to demonstrate the EMCON feature. This handle could be affected both via the GUI and by the on-board user button, which was implemented in such a way that it activated an interrupt which would execute a function call toggling the EMCON feature. As shown in Code Listing 5.17, the controller for the on-board user button is initialized in line 4, where the constructor takes a function pointer argument pointing to the function which toggles the EMCON feature on or off.

Listing 5.17: EMCON feature

```
1  /**
2   * in main.cpp
3   */
4  static PushbuttonController local_emcon_control(PB_2, &toggle_emcon);
5  ...
6  /**
7   * in PushbuttonController.h
8   */
9  class PushbuttonController
10 {
11     public:
12         PushbuttonController(PinName pin, void (*func)(void)) : _pin(pin)
13         {
14             this->_pin.fall(func);
15         }
16     private:
17         mbed::InterruptIn _pin;
18 };
19 ...
20 /**
21  * in emcon.h
22  */
23 static void toggle_emcon()
24 {
25     if(emcon_active)
26     {
27         emcon_active = false;
28         lorawan_event_queue.call(send_message);
29     }
30     else
31     {
32         emcon_active = true;
33     }
34 };
```

5.2.3 Energy conservation through sleep mode

The device will automatically sleep in the mandatory wait-period following a transmission, which is automatically determined by the LoRaWAN stack provided that the `lora.duty-cycle-on` field in `mbed_app.json` is set to `true`, as shown in Listing 5.1.

Following each transmission, the CPU stats were printed, which yielded the amount of time the device has spent active and idle since the previous call to the function. The output in Listing 5.18 shows typical CPU stats following a transmission for this particular implementation.

Listing 5.18: CPU stats following an uplink transmission

```
[GPS]  Threshold met -- Lat = 59.633583, Lon = 10.816844

[Gas]  No gas detected

[HR]  BPM = 66

[MUAP]  MUAP = 100

[MAIN] 17 bytes scheduled for transmission

[MAIN] Message Sent to Network Server

=============== CPU stats ===============
Time(us):                         67792877
Idle:                             22373881
Sleep:                            22373881
DeepSleep:                              0
Idle: 23% Usage: 77%
```

Although all items required to enable deep sleep were checked according to the documentation (Arm Mbed 2019), the device never activated deep sleep, for reasons unknown. In concrete terms, tickless mode was enabled, low-power tickers were available, and no active drivers or buses were active outside sensor readouts which could potentially block deep sleep.

5.2.4 LoRaWAN backend and data integration

ChirpStack was found to be very easy to install, setup, and configure for the platform on which it was running and the system it supported. In particular, the MQTT interface and embedded support for Cayenne LPP made ChirpStack a crucial component for enabling application-wide information flow. The screenshot in Figure 5.10 shows the LoRaWAN frames being transmitted on the air to and from the gateway. Note that this particular view is not that of the Application Server. Thus, the payload data is not visible in plain text.

The frames would subsequently be processed in the Application Server, where the payload format presented in JSON form shows a lot of metadata, as shown in Figure 5.11. Note the fields corresponding to the named identifiers used in Cayenne LPP. GPS location from the device is organized under `gpsLocation`, while biometric data for heart rate

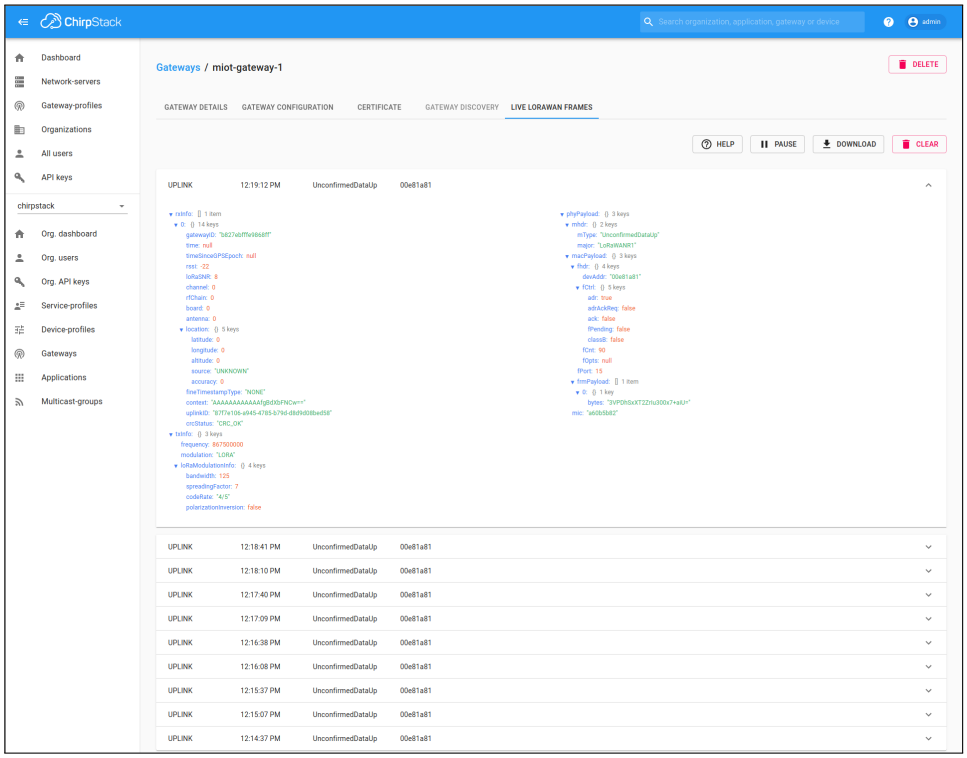


Figure 5.10: ChirpStack GUI showing LoRaWAN frames

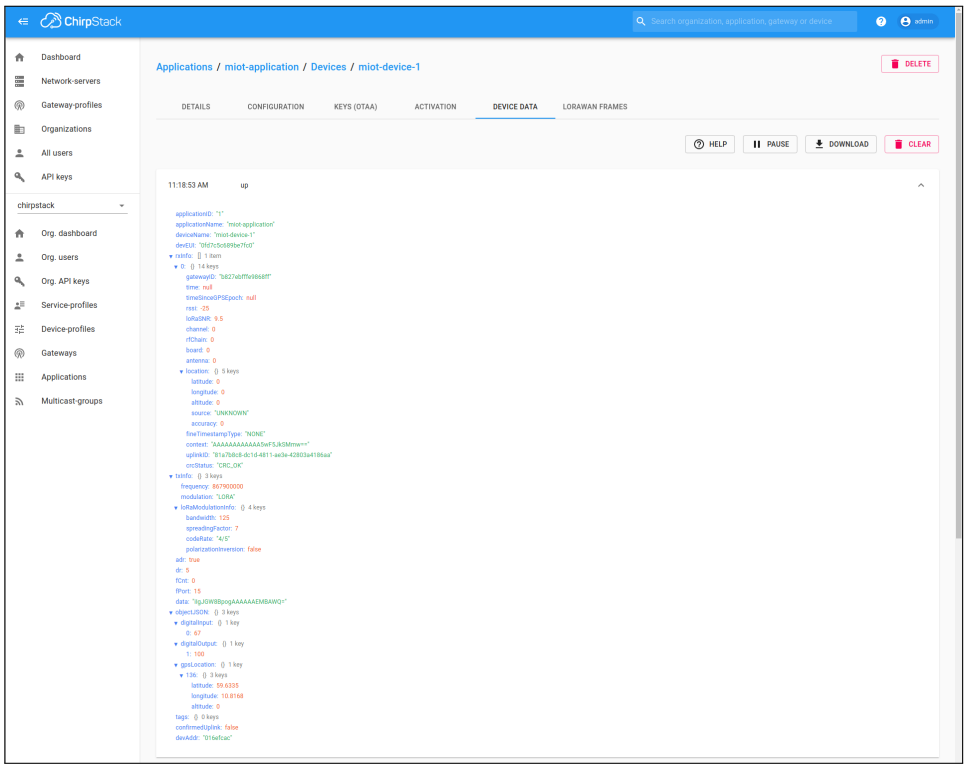


Figure 5.11: ChirpStack Application Server showing device data

BPM and muscle activity can be seen under the fields `DigitalInput` and `DigitalOutput`, respectively, as described in Section 5.2.1.

Listing 5.19: Data Integration application output

```
Apr 19 08:58:30 data-integration miot-data-integration[17108]: Received
    ↪ uplink, ID-data: DevEUI = 0fd7c5c689be7fc0 CS = A-6-A-5, no NS
    ↪ data, BIO: BPM = 66 MUAP = 100
Apr 19 08:58:30 data-integration miot-data-integration[17108]: Published
    ↪ message (107 bytes): "[["direction","uplink"],["devEUI","0
    ↪ fd7c5c689be7fc0"],["callsign","A-6-A-5"],["health_status","
    ↪ UNDEFINED"]]"
Apr 19 08:58:55 data-integration miot-data-integration[17108]: Received
    ↪ uplink, ID-data: DevEUI = 0fd7c5c689be7fc0 CS = A-6-A-5, NS: Lat =
    ↪ 59.6335 Lon = 10.8168, BIO: BPM = 66 MUAP = 100
Apr 19 08:58:55 data-integration miot-data-integration[17108]: Published
    ↪ message (157 bytes): "[["direction","uplink"],["devEUI","0
    ↪ fd7c5c689be7fc0"],["callsign","A-6-A-5"],["lat",59.6334991455078],
    ↪ ["lon",10.8168001174927],["health_status","UNDEFINED"]]"
```

The implementation of the data integration component proved simple due to the good documentation of the libraries in use, in addition to a high level of community support. In addition, no MQTT package losses were observed in either direction (i.e. from device to application, and application to device). However, it should be noted that the devices running the LoRaWAN backend and the data integration component both resides on the same local Wi-Fi. Sample output from the application can be seen in Listing 5.19.

5.2.5 User Interface application

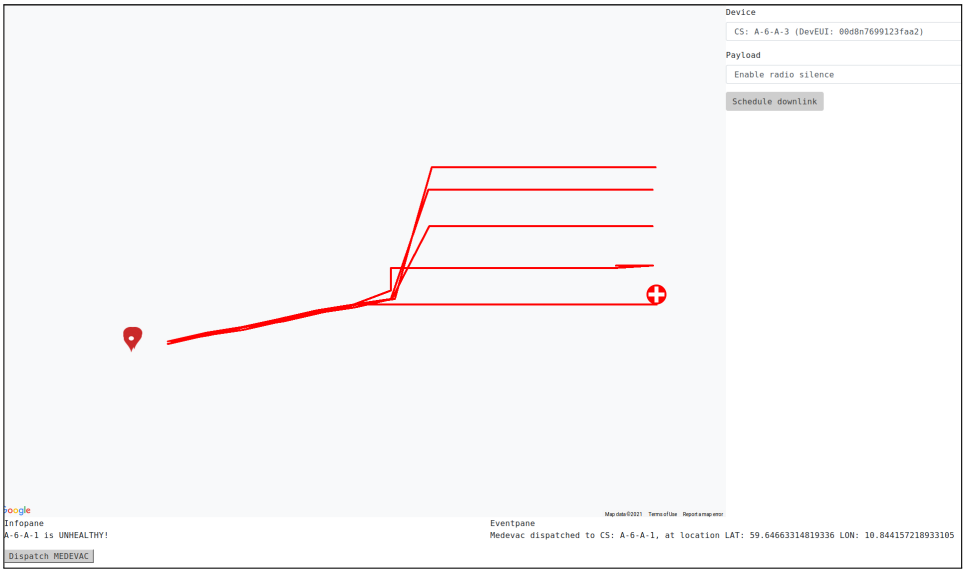


Figure 5.12: User Interface application

The UI is largely an ad-hoc application meant only to serve as a visual means towards showcasing the value of the soldier wearable in a BMS-like fashion. In this thesis, UI/UX was therefore not a concern as the visual representation and management tools it provided would need to

be integrated into existing BMSs in use today, as described in Sections 1.4 and 4.5.1. As shown in Figure 5.12, where simulated devices were deployed, the GUI showcases a simple information feed which provides the user with a recommended course of action for given events (in this case, a recommendation to deploy medical resources to the wounded soldiers' location was advised), and a simple set of commands for device management. In this particular case, it shows a 5-man foot patrol that sustained one injury during enemy contact, where the patrol managed to escape to a holding area.

The interface was implemented with two custom commands for enabling or disabling EMCON on the device. Once activated, the embedded MQTT instance would publish a message to the ChirpStack Application Server targeting the specific design using the topic *application/1/device/<devEUI>/command/down*. The message was built as a JSON string using a Base64-encoded payload, as shown in Code Listing 5.20.

Listing 5.20: Downlink JSON message

```
const commands =
{
    ENABLE_RADIO_SILENCE:
    {
        text: "Enable radio silence",
        value: '{"confirmed":false,"fPort":1,"data":"ZW1jb246dHJ1ZQ=="}',
            ↪    //emcon:true
    },
    DISABLE_RADIO_SILENCE:
    {
        text: "Disable radio silence",
        value: '{"confirmed":false,"fPort":1,"data":"ZW1jb246ZmFsc2U="}',
            ↪   //emcon:false
    },
};
```

Once published, the message will be queued for transmission and sent once it receives an uplink message from the device. The queue can be queried using the following API endpoint:

```
http://<URL>:<port>/api/devices/<devEUI>/queue
```

Note that in order to access it, an authorization token must be acquired through the ChirpStack dashboard. An example output is shown in Listing 5.21, where one message is waiting to be transmitted to the device.

Listing 5.21: Downlink queue API endpoint

```
~$ curl -H "Authorization: <token>" http://192.168.1.113:8080/api/devices
    ↪ /0fd7c5c689be7fc0/queue
{"deviceQueueItems":[{"devEUI":"0fd7c5c689be7fc0","confirmed":false,"fCnt"
    ↪ :5,"fPort":1,"data":"ZW1jb246ZmFsc2U=","jsonObject":""}],"
    ↪ totalCount":1}
```

5.3 Subject evaluation

In technical terms, integration of a proof-of-concept MIoT system is a success. However, to further evaluate the operational value of such a system, a new round of interviews was conducted using the same informants from the fact finding interview. The interview was designed to acquire concrete feedback on the prototype, with emphasis on the information it provided. The interview started with a simulation using node behavior that replicates the prototype, where the simulation shows a foot patrol that eventually indicates that they are caught in enemy contact, sustaining one injury. The questions outlined in Table 5.1 followed the simulation, and the transcribed responses can be found in Appendix H.

Table 5.1: Interview guide

Number	Question
1	Hypothetically, how do you think such a system fits with existing operational patterns and doctrines?
2	Hypothetically, do you think a full-scale deployment of such a system would enhance SA, and why? If so, at what level (i.e. in the military organization) would it be the most applicable?
3	Would you consider the level of information in the system to be too little, or too much?
4	Do you consider the level of control provided to you in the role as OpsOff, team leader, or platoon commander over the devices as sufficient? Why, or why not?
5	Other comments or thoughts regarding the implementation and deployment of such a system?

In the following, we will analyse the responses from the three informants in order to get an indicator of whether or not the system design fulfills its purpose, namely increased SA. To achieve this, the topics discussed here are divided into concrete prototype feedback based on the informants' experience from the simulation, what operational use the informants think such a system could be best suited for, and what challenges they think such a system could face towards large-scale deployment.

5.3.1 Prototype feedback

This topic relates to the informants' experiences while using or observing the GUI during the simulation, and how and if they consider the presented system a positive supplement for the military organization.

One informant outlined the similarities between the prototype and existing systems today, where vehicle tracking is in place, in addition to

extended sensor data for specific armored vehicles. Overall, the feedback was largely based on enhanced battle space information resolution, lowered need for voice communications, and increased responsiveness, as derived from the following statements:

> I can see their position, their formation. The patrol leader can spend more time leading what's happening on the ground rather than keep a report with the rear, because they receive most of the information through this instead. If I as OpsOff are wondering about something, [...] I can instead look at the screen, where are they, what are they doing, they are doing OK. [...] If something unforeseen happens, then I can prepare resources immediately when something happens, like a QRF or MEDEVAC, [...]. So when I then get voice comms with the patrol leader saying he's in this or that position, then I can press that dispatch button [...] So it's really about increased operational tempo, in addition to increased SA, thus improving the decision basis for the commanders. [...] It greatly improves the tempo on the battlegrounds, so you don't drop artillery on your own forces, you know where not to drive if they are firing in certain directions, and so on. I also think it is useful to be able to zoom in and out to see the units formation and such, since this tells me a lot about their threat assessment. That it updates real-time is also something I appreciate. I also think it is good to be provided with information regarding their state, such as if they are physically exhausted, unhealthy, or healthy, as long as you know what those terms means.
>
> — INF2

> [...] here you get the information directly. Such as MEDEVAC can be executed, you know where they are, if you need fire support then you can add that directly, because you know where they are, direction, distance, everything you need. And then you can plot that in.
>
> — INF1

Another informant emphasized the importance of compatibility, where the system should integrate with existing systems currently in use today.

> Provided that the sensors can handle both harsh weather and potentially a bit of a beating, then I think it is very compatible with the doctrines, the tactical and strategical operational methods and procedures we use today. As long as the hardware can handle said treatment and is compatible with existing software we already have, and that it doesn't require a lot of extra stuff, then I think it is very compatible.
>
> — INF3

Regarding the `Dispatch`-button offered in the UI, two of the informants seemed reluctant to press it once it showed up in the UI. One of them stated that he would coordinate with the patrol leader on the ground before hitting dispatch, as given by the following statement:

> [...] when I then get voice comms with the patrol leader saying he's in this or that position, then I can press that dispatch button, then it's just to drive out and get them.
>
> — INF2

The other informant described a scenario in which the patrol leader would be the executive authority to activate such resources, as derived from the following statement:

> [...] it depends on the information you get, you get one UNHEALTHY here, then the patrol leader himself can also press that same button, in a smart phone fashion. The platoon commander is only concerned with if the wounded soldier is still usable or not. Then the patrol leader could for instance just press a button regarding his state [...] all information regarding injuries and such that is not your concern, that is the medical units concern. If they should bring a lot of bandages or just a large black bag [...] So you should disseminate the information to the right people.
>
> — INF1

The third informant seemed rather positive regarding such a feature, where he suggested that it could be useful if voice-based communication systems are jammed down (i.e. not available for him to use), thus only requiring him to press said button. However, this informant agrees with the others that such buttons shouldn't be activated without considering the full picture of the situation, as derived from the following:

> Action buttons as shown here is potentially a very useful feature, because you might have to execute for example MEDEVAC using a completely jammed down radio net, so to have the possibility to just push the button is just awesome, as long as the one pushing it still making the decision based on the situation.
>
> — INF3

Based on the above statements, the prototype received positive responses in terms of improving SA on lower levels, thus improving the decision basis for commanders. It was also suggested that it could potentially improve operational tempo, exemplified by fire missions as stated by one informant, where friendly forces locations are known automatically, thus ensuring that artillery strikes do not accidentally hit own forces.

It would seem that the informants found the dispatch-feature somewhat disruptive, as it wasn't clear how they would want to use such a functionality. From the responses, it would seem that some voice-based communications would be required regardless if a button-press would, at its core, solve the same task, which in this case is narrowed down to a potential location for evacuation.

5.3.2 Suggested operational use

This topic relates to how the informants think the system presented to them could be best utilized in military missions or use cases.

When prompted for comments regarding large-scale deployment (i.e. across the whole Norwegian Army), one informant outlined some concerns regarding its usability in traditional steel versus steel warfare due to the sheer intensity and volume. Instead, the informant suggested that it could be put to better use in low-intensity missions, such as mentoring or peacekeeping missions, as stated by the following:

> There is a lot that indicates that such a system may produce information overload during high-intensity, steel versus steel, warfare. Where it is a matter of minutes or hours until a unit has either been eliminated or eliminated the enemy. So I think in that case then this might just be an added complexity to the scenario, and not help the SA in any remarkable way. [...] For units conducting stabilization missions or mentoring in for instance Iraq then I think such a system have a completely different role, majorly due to the very low acceptance for loss of life during such international missions compared to the previously mentioned large-scale warfare. So I think it is more in the low-intensity operations that such a system would truly shine, mainly at platoon and company levels.
>
> — INF2

Another informant described the potential time saving effect of using such a system for tracking purposes, as stated by the following:

> Something I've really missed as a platoon commander is a live feed of the foot-mobile infantry whenever they were out, where I've had to receive a GPS position from the foot-mobile team leader and plot that manually. So if I as platoon commander have had access to this data in a live feed, then it would have built an incredible SA at platoon, company, and battalion level. It would have been insane amounts of time saved. [...] for contact situations, I think we would have saved, my guess, half an hour.
>
> — INF3

Based on the responses given above, it is likely that such a system could help close information gaps on lower levels in terms of individual soldiers' whereabouts. It was also suggested that the information provided through such a system might not be as useful in all scenarios, as one informant described traditional full-scale warfare as too intense and too vast in volume for such a system to provide meaningful data in a timely manner. However, the same informant described low-paced missions, such as mentoring or peacekeeping, as more appropriate due to the lower acceptance for loss of life in such scenarios.

5.3.3 Challenges

This topic relates to what challenges the informants think might arise from utilizing such a system across the whole Norwegian Army. From the fact finding interview, it became clear that leadership culture was a concern, as it became evident that micromanaging the units on the ground may occur due to the improved level of information. The same concern was brought up again during this interview, as given by the following statements:

> [...] you can have that at squad, platoon, company, battalion, at all levels, no problem, but it have to be aggregated. So as a brigade commander, then you see the battalion as a box, and then downwards to the patrol leader that can see all the members of the patrol as individuals. That is absolutely the biggest problem, that leaders get stuck on details they are not really supposed to have. When the brigade commander is interested in what rifleman 1 is doing then he doesn't know his own job.
>
> — INF1

> I personally know about officers and NCOs that would use this to micromanage them, "go a bit more to the left", "don't go that way", "don't do that", which is a pitfall in itself. But that's more about leadership culture, and not the technology.
>
> — INF2

One informant suggested that in order to successfully implement the system in a manner where units would consider it a positive supplement, it should be tried and tested in increments, starting with units small in size, and compare said unit's performance with others who do not utilize the same system, as stated as follows:

> If we consider end-state where each individual combat soldier have such a system and uses it, then I think we need to test the system on smaller sub-parts of the organization for it to gain success among the users. [...] How is it for the soldiers to wear the system, how is it for the commanders to receive the

> information, and when you start to understand the experience to this squad alone, then you can start scaling up to larger parts of the organization. [...] If you start with the whole unit, and the test results are initially bad, then you will most likely have lost the possibility for positive reception among the larger parts of the organization, then you will most likely meet resistance from the users. [...] So small-scale, incremental testing and fine-tuning will most likely spread the word about such a system in a positive manner, for instance, platoons using the system perform better during exercises, they make decisions faster, keep a faster tempo, and so forth.
>
> — INF2

5.3.4 Subject evaluation summary

The following key findings should be considered for any future work wanting to pursue soldier wearable systems:

1. High-resolution in terms of information detail is always good, but must be put to proper use at the appropriate leadership audience.
2. To integrate well, the system must integrate with existing systems, most notably BMS, and not exist as a parallel solution.
3. Soldier wearables such as the one presented here could potentially work better in scenarios where the combat intensity and volume of troops are rather low compared to conventional steel versus steel warfare.
4. To successfully integrate such a system at large scale, it should be tested and evaluated starting with small units where their performance should be evaluated in comparison with others who do not utilize such a system, thereby measuring its effect on combat effectiveness.

5.4 Summary

In this chapter, we covered the implementation process using specific hardware for the integration platform, in addition to the process of connecting software components along the whole IoT pipeline up to and including the UI where users could evaluate the sensor platform. It was found that Mbed OS proved to be a relatively simple software platform to develop the prototype on due to its large community and well-documented APIs. The COTS sensors were also found to be relatively simple to connect to the development board, but required some attention towards successfully outputting meaningful data.

The finished sensor prototype was shown to establish a solid proof-of-concept through the use of position data and biometric sensors, as stated

by the feedback acquired through a second interview designed to give the end users a firm understanding of the system, and to acquire their opinions regarding its potential operational use. As such, it was found that low-paced mentoring or stabilization missions could potentially make better use of such a system rather than traditional large-scale warfare due to the sheer intensity and short periods of time where massive losses would occur.

Chapter 6

Conclusion

In this thesis, the goal was to investigate the usability of soldier wearables in terms of enhanced or augmented situational awareness by developing a prototype using COTS hardware and open source solutions. The research contribution was conducted in the frame of the following research questions (denoted R1..R3 below), as outlined in Section 1.6:

1. R1: How can an IoT wearable improve the current MO in the Norwegian Armed Forces?

2. R2: In what way can an IoT wearable enable autonomous information acquisition and dissemination?

3. R3: What constitutes a viable approach to a wearable prototype, when emphasis is on low cost, ease of availability and using available civilian technologies?

6.1 R1: Improving the current MO

This research question was conceptualized in Section 1.5.1 through the interview guide, and finally realized in Section 4.3 through interview findings, related work found in Chapter 3, and own operational experiences, where it was found that the soldier wearable could help improve combat effectiveness through increased SA and information resolution by visualizing each individual's health status and position. Specifically, the acquired feedback from the subject evaluation in Section 5.3 highlighted a potentially lowered usage of voice-based communication, which would relieve leaders on the ground to focus more on the task at hand. Additionally, the increased battle space information resolution through the provided position data for individual riflemen proved to be a useful feature for officers in the rear as well, in particular with respect to coordinating external resources such as medical evacuation or fire support, which relies heavily on own units' location. It also proved to be a useful feature towards closing information gaps in the battle space due to the lack of an automated data feed as opposed to BMSs currently in use on armored vehicle platforms.

6.2 R2: Autonomous information acquisition and dissemination

This research question was conceptualized in Section 3.2 through previously identified mission-critical use cases for IoT in the military domain, and finally realized in Section 4.5 and 5.1 through system design and -implementation of a wearable prototype in the frame the identified use cases in Section 1.5.1. Using the rifleman platform for information acquisition was found to be useful for tree particular information categories, namely:

- Geographical position data
- Biometrics
- Logistics

The former two were investigated in this thesis. The position data was shown to be most crucial towards extending existing tracking systems to the rifleman as well as armored vehicles, whereas biometric sensor data was shown to be valuable for the end-user using both quantitative or qualitative data, depending on their position in the military organization. Recall that information needs to reach both vertical and horizontal elements, e.g. the commanding officer needs to know what his units are doing, and support units needs to know where the requesting units are located.

6.3 R3: Viable prototype

This research question was conceptualized through Chapters 2 and 3, in particular in Sections 3.5 and 3.7 through previously conducted experiments involving practical hardware and protocol testing using open standards, open source solutions, and COTS equipment. The prototype was designed in Chapter 4, and later realized in Chapter 5 through both the implementation and evaluation phase of the development process. It was found that Mbed OS and its LoRaWAN API stack was fairly simple to use for our custom sensor build. The integration challenge was however found in properly integrating the external sensors, and in particular parsing the returned data. Furthermore, Cayenne LPP was found to be the easiest and most flexible way to pack data into a LoRa-message, rather than using manual bit packing, in particular if the transmitted messages did not adhere to a fixed structure. It was however slightly lacking available data identifiers, which ideally would have offered specific identifiers for every possible biometric attribute.

Furthermore, the gateway setup using Raspberry Pi 3B and iC880A LoRa concentrator, linked together using a LinkLab LoRa gateway shield, proved to be a stable and flexible solution when configured to run LoRa Basics Station.

The evaluation also showed that the specified IoT baseline outlined in Section 4.1 worked well for the prototype developed in this thesis, with emphasis on LoRa and LoRaWAN as the carrier for the outer elements in the MIoT network. Furthermore, MQTT was found to work very well for this particular system design, where its low overhead and ease of use made implementation of custom solutions a relatively simple task.

Finally, ChirpStack running on a Raspberry Pi 4 proved to be an effective and stable LoRaWAN backend solution, due to its ease of setup and configuration. The built-in support for Cayenne LPP also proved to be a crucial part towards enabling dynamic data transmissions from the end-node to user application.

6.4 Summary

In this thesis, we conducted the ten steps for IoT system design methodology (see Section 1.5), combined with interviews to further strengthen the design and evaluation steps with input from serving military officers. Following the conclusion outlined above, we were able to successfully answer the research questions pertaining to this thesis. Hence, the purpose of the work was fulfilled and the goal we set out to investigate was reached. During the work, however, several ideas arose that could not be pursued in the scope and time frame of this thesis. These ideas for future work are presented in Chapter 7.

Chapter 7

Future work

In this chapter, we outline recommended paths towards improving a wearable soldier system. Some of which were outside the scope of this thesis due to time limitations, but are considered crucial components in an operational setting, while others were identified during the development process, and thus should be considered for future work. These recommendations will be put into context of a next-generation soldier wearable, BMS integration, and finally Big Data and machine learning.

A proposed high-level architecture for future soldier wearable systems is depicted in Figure 7.1, where the level of information for the rifleman platform is grouped into the three information categories outlined in Section 6.1. At the centre, we've outlined the infrastructure supporting the MIoT subsystem, where data analysis capabilities and the LoRaWAN backend resides. This infrastructure is depicted without a topology as it remains to be determined how this could best be solved using existing tactical radio systems as communication links. At the application layer, we see two distinct roles which should be able to view the processed data at a certain granularity and level of detail in accordance to their responsibilities in the military organization. Additional roles and their required level of detail should therefore also be investigated in future work.

In the following, we will further discuss the three information categories from Section 6.2 pertaining to a soldier wearable, namely 1) geographical position, 2) biometrics, and 3) logistics.

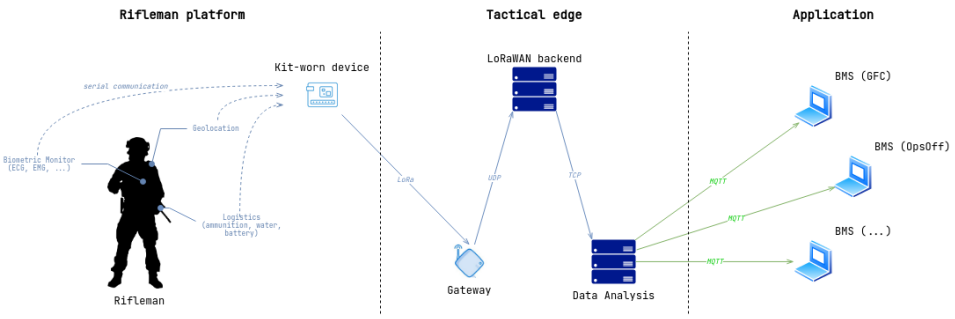


Figure 7.1: Proposed future high-level architecture for the soldier wearable

7.1 Geographical position

Geographical position is arguably the most important piece of information such a system could provide to the end users. However, it would require some attention towards making the system more robust. Most notably, the EW threat could involve potential GPS signal jamming on all bands, thus potentially reporting false position data or no position data whatsoever. It is therefore suggested that software logic is in place so as to determine whether or not the position data makes sense in comparison with previous readouts.

7.2 Biometrics

Health status could effectively act as a live patient monitoring system should the individual wearer sustain injuries of any kind. However, this would require monitoring of additional biometric features about the wearer, such as respiration, pulse-oxygen saturation, blood pressure, and so forth. Such sensors also need to be integrated in a less intrusive manner than the sensor pads presented in this thesis. Ideally, biometric sensors should be an integral part of inner layers of clothing.

Additionally, the work in this thesis only tested the usability of muscle activity sensors on one single muscle. To provide an accurate reading regarding the state of the wearer, a much wider range of muscle groups should be measured as well, where the level of detail required for soldiers remain to be discussed in the frame of such a system. If we consider for instance 13 main muscle groups (which effectively means monitoring 26 individual muscles), it will require 78 bytes total to provide readings for all groups if we're using Cayenne LPP. Evidently, this will not work for the lower data rates. Therefore, it is recommended to investigate means to determine the physical and physiological state of the wearer on the device rather than at the backend.

7.3 Logistics

Logistics is a property that was not part of this thesis. However, it was initially presented as a prospected feature, which gained the attention of all informants in the interviews. Generally, the two most critical items of inventory for any soldier are water and ammunition, which should be monitored by the integration platform through water level sensors and a shot counter. The shot counter could also be extended to act as a target designator and alarm raiser. Firstly, a laser range finder which activates at the same time as a shot was being fired could provide simple means for target designation by calculating the distance and account for bullet drop. This information could provide crucial information in the UI for the end users towards providing assistance, possibly through deploying aerial assets or request indirect fire to enemy positions. Lastly, if a shot was fired, it should also raise an alarm in the UI so as to alert the end user that either

one of two events have occurred: a weapon was discharged accidentally, or enemy contact was made.

7.4 Additional features

In the literature, significant work efforts have been put into finding practical ranges and throughput rates using LoRaWAN. However, most of these findings were based on somewhat static sensors while manually altering the transmission parameters. For this reason, blind ADR (see also Appendix C) should be investigated to determine reception rates for mobile units, including mobile gateways which most likely would be mounted on armored vehicles.

Furthermore, as federated networks are a likely topology for large-scale military deployments, it is also likely that units within said deployment will manage their own LoRaWAN infrastructure. In order to enable connectivity between allied and partner nations, it is therefore of interest to investigate roaming (see also Appendix F) between LNSs while relying on tactical radios as communication links.

Security-wise, additional features should be investigated to secure end-nodes and the information pipeline in an end-to-end fashion, in particular with regards to data integrity if an additional component such as data analysis resides between the Application Server and the end users.

Finally, regarding use of Cayenne LPP, it is recommended to extend existing libraries with known identifiers for certain data. At the time of writing, there is no identifier available for heart rate BPM or muscle activity, neither for gas types. Thus, when a complete information set for a wearable has been identified, the items embedded within it need to have added support for use with Cayenne LPP.

7.5 BMS integration

Even though UI and UX was not part of this thesis, a lot of feedback was given regarding how the information was presented to the informants. In particular, following a trace of single sensors could provide useful information regarding formation, and by extension their alertness or threat assessment. However, a trace should only be visible for a certain amount of time, or for the most recent positions only, as it would clutter the map view for the user if it was deployed over time with a lot more soldiers.

For a complete end-state to be reached, the information provided by the system should be integrated into existing BMSs currently in use. This is a matter of decreasing the amount of equipment a military unit would need to bring on missions, and so eliminate the need for further management, control, and maintenance for equipment, in addition to keeping the soldier load-out as low as possible. In addition, both iterations of interviews (see transcripts in Appendix G and H) raised concerns regarding the level of detail presented to the user, which most likely requires customized views in accordance to the viewing audience. In concrete terms, the lower a military

leader resides in the hierarchy, the higher information resolution he should have, and vice versa.

7.6 Big Data and machine learning

As mentioned both in the introduction and system specification, Big Data or data analysis was not part of this thesis. However, as described in Section 3.2, sensor data is not particularly useful without proper analysis. As experienced through the prototype demonstration, manually interpreting heart rates and muscle activity is not something a military commander would spend time doing. Rather, qualitative descriptors regarding their states should be in place, which should be determined through proper data analysis software.

Furthermore, machine learning and AI could prove useful for this particular scenario, so as to provide information regarding the situation on the ground much faster to the decision makers. It is therefore suggested that this is investigated in order to measure the time significance for utilizing such a component in the context of the OODA-loop and/or Lawsons C3I model.

7.7 Summary

In this chapter, a brief overview of future work propositions were presented. It was suggested that the level of detail regarding the rifleman platform be discussed further so as to properly provide the required information to any stakeholders further up the chain of command. This level of detail should then be used in conjunction with a data analysis component, which was not investigated in this thesis due to time limitations. Then, a number of information sets should be defined in accordance with the viewing audience, where these information sets should be made available on existing BMSs.

Appendix A

ISM and transmission restrictions

A.1 Regions

Sigfox and LoRa both operate in the sub-gigahertz spectrum of the license-free ISM bands. The exact parameters and usage policies of these RF bands depends on geographical location. These regions can be seen in Table A.1 along with the allocated frequency range for ISM bands in said region. These frequencies are allocated under the authority of the ITU Radio Regulations Article 5.

To enable fair usage and availability among a large user mass, channel occupation and transmission power limits are in place to ensure no permanent channel occupation. The exact restrictions varies depending on the geographical region. A list of ISM bands and a non-exhaustive list of countries or regions pertaining to it is outlined in Table A.2.

However, not all devices transmitting on ISM are ISM bands only. In fact, quite a lot of devices operate in or overlap with ISM bands, such as car keys, garage door openers, wireless headphones, baby alarms, RFID, and others.

ISM band	Region	Frequency band
EU 868	Europe	863 - 870 MHz
US 915	North and South-America	902 - 928 MHz
CN 779	China	779 - 787 MHz
EU 433	Europe, Africa, Russia	433 - 434 MHz
AU 915	Australia and Oceania	915 - 928 MHz
CN 470	China	470 - 510 MHz
AS 923	Australia	923 - 924 MHz
KR 920	Republic of Korea	920 - 923 MHz
IN 865	Indian sub-continent	865 - 867 MHz
RU 864	Russian Federation	840 - 870 MHz

Table A.1: ISM band overview

ISM band	Max uplink EIRP	TX restriction
EU 868	14 dBm	LBT AFA or duty cycle <1%
US 915	30 dBm	Max dwell time 400ms on uplinks
CN 779	12.15 dBm	Duty cycle <1%
EU 433	12.15 dBm	Duty cycle <10%
AU 915	30 dBm	Depends on UplinkDwellTime parameter in TxParamSetupReq[1]
CN 470	14 dBm	LBT AFA and max TX time 1 second
AS 923	14 dBm	LBT (Japan only), otherwise duty cycle <1%
KR 920	14 dBm	LBT AFA
IN 865	20 dBm	No dwell time or duty cycle limitations
RU 864	14 dBm	No dwell time limitation

Table A.2: ISM band restrictions

A.2 Duty cycle

Time spent per transmission is usually measured using duty cycle, which is the fraction of one period in which a signal is active, expressed as a percentage or ratio. A period in this sense is defined as the time it takes for a signal to complete an on- and off-cycle. Put simply, if a signal is active for 40% and off for the remaining 60% of the time, where time can mean any time unit desirable (second, minute, hour, day, etc.), the signal's duty cycle is 40%.

In the EU868 band however, the European Telecommunications Standards Institute (ETSI) allows the Listen-Before-Talk Adaptive Frequency Agility (LBT-AFA) transmission management as an alternative over the duty cycle restrictions. This form of spectrum access requires the device to "listen" on specific channels to determine the average signal level over some period of time. If it is below a given threshold, it proceeds with the transmission. Conversely, it will have to either wait until the threshold reaches an acceptable level or alternatively switch to a different channel.

A.3 Output power

Effective Isotropically Radiated Power (EIRP) is essentially the radiated output power from the antenna with respect to a half-wave dipole. What this practically means is that the radiation from any antenna cannot exceed a certain field strength. If this were not regulated, even low wattage would be enough to "drown" out other actors attempting to use the band. A brief overview of these restrictions can be seen in Table A.2.

[1] See also Appendix D

Appendix B

LoRa signal encoding

LoRa uses CSS modulation for RF transmissions, a Direct-Sequence Spread Spectrum (DSSS) [1] technique which linearly increases or decreases the carrier frequency over a given bandwidth, both of which can be altered in LoRaWAN. This provides some level of flexibility in terms of reception levels and throughput, while also reducing the complexity of receiver design. This is because both the transmitter and receiver timing and frequency offsets are equivalent compared to traditional DSSS, in which accurate and expensive reference clock sources are usually required to keep both ends synchronized. In addition, compared to CSS, DSSS requires the receiver to spend more time decoding and synchronizing with the received signal, thus demanding more processing capacities (Semtech Corporation 2015).

Each “sweep" of the available bandwidth is commonly referred to as a “chirp", where frequency-increasing chirps are referred to as up-chirps. Conversely, frequency-decreasing chirps are referred to as down-chirps. Using a waterfall viewer, we can see the chirps of a LoRa transmission as a relationship between time and frequency as shown in Figure B.1.

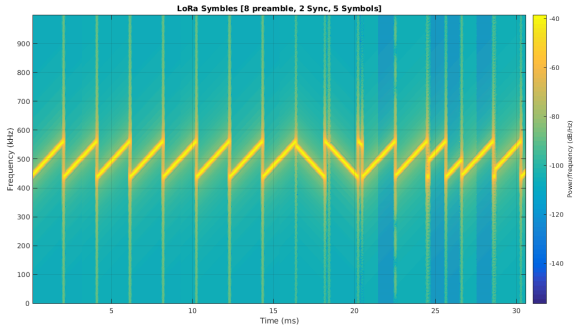


Figure B.1: LoRa transmission as seen in a waterfall viewer (credit: (Ghoslya 2017))

[1] DSSS is a spread-spectrum modulation technique used to increase signal resilience, and is widely used in data communication applications.

Appendix C

Adaptive Data Rate

LoRaWAN implements ADR (Semtech Corporation 2016b), which is a mechanism used to determine optimal data rates for the given signal conditions surrounding the end-node. When enabled, the node will automatically set the DR based on certain metrics from the last 20 packages transmitted from the moment the ADR was enabled, thus enabling a somewhat smart link sensing based on fairly simple conditionals.

This works well for dynamic signal conditions, but not for mobile sensors since the surrounding signal conditions for the device most likely changes with the position. As a consequence, the link measurements needed to determine optimal DR is too volatile to provide an accurate decision. For mobile sensors it is therefore recommended to use blind ADR, a variant of ADR that uses a fixed selection of three different DRs transmitting at periodic intervals, as shown in Figure C.1, where we transmit at SF12 once every hour, SF10 twice every hour, and SF7 thrice every hour. This way, the battery life can still be economized while achieving some level of throughput guarantee in dynamic environments (Semtech Corporation 2016a).

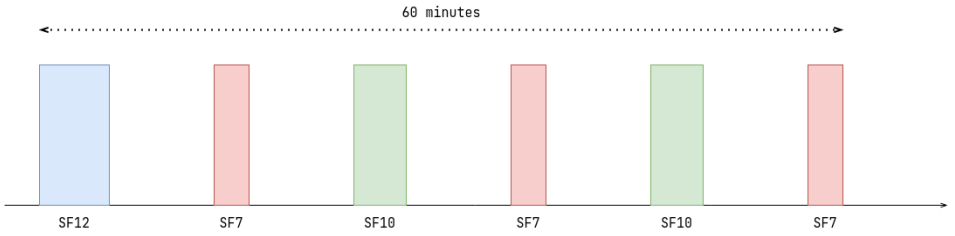


Figure C.1: Blind ADR (credit: (Semtech Corporation 2016a))

Appendix D

MAC commands

Table D.1: LoRaWAN MAC commands

CID	Command	TX	Description
0x02	LinkCheckReq	Device	Used to valide network connectivity
0x02	LinkCheckAns	LNS	Answers a LinkCheckReq containing signal metadata
0x03	LinkADRReq	LNS	Requests device to change DR, TX power, redundancy, or channel mask
0x03	LinkADRAns	Device	LinkADRReq ACK
0x04	DutyCycleReq	LNS	Sets max aggregated transmit duty cycle of and end device
0x04	DutyCycleAns	Device	DutyCycleReq ACK
0x05	RXParamSetupReq	LNS	Sets RX slot parameters
0x05	RXParamSetupAns	Device	RXParamSetupReq ACK
0x06	DevStatusReq	LNS	Status request of a device
0x06	DevStatusAns	Device	Returns status, namely battery level and radio status
0x07	NewChannelReq	LNS	Creates or modifies the definition of a radio channel
0x07	NewChannelAns	Device	NewChannelReq ACK
0x08	RXTimingSetupReq	LNS	Sets timing of RX slots
0x08	RXTimingSetupAns	Device	RXTimingSetupReq ACK
0x09	TXParamSetupReq	LNS	Sets the max allowed dwell time and MaxEIRP of a device, based on local regulations
0x09	TXParamSetupAns	Device	TXParamSetupReq ACK
0x0A	DIChannelReq	LNS	Creates an asymmetric channel by shifting the downlink frequency from the uplink frequencies
0x0A	DIChannelAns	Device	DIChannelReq ACK
0x0D	DeviceTimeReq	Device	Requests the current GPS time
0x0D	DeviceTimeAns	LNS	Answers DeviceTimeReq

Continued on next page

Table D.1 – continued from previous page

CID	Command	TX	Description
0x10	PingSlotInfoReq	Device	Periodically communicates the unicast ping-slot to LNS (Class B)
0x10	PingSlotInfoAns	LNS	PingSlotInfoReq ACK (Class B)
0x11	PingSlotChannelReq	LNS	Sets the unicast ping channel frequency and DR of device (Class B)
0x13	BeaconFreqReq	LNS	Modifies the frequency at which the device expects to receive a beacon broadcast (Class B)
0x13	BeaconFreqAns	Device	BeaconFreqReq ACK (Class B)
0x20..0x2F	NIL	NIL	Reserved for Class C commands
0x80..0xFF	Proprietary	Both	Reserved for proprietary network command extensions

Appendix E

LoRaWAN hardware

Hardware selection is important for any IoT system. In the following, we will briefly cover available LoRa radio modules and their features, and what requirements that should be met for said radio modules to operate as an embedded part on MCUs.

E.1 Radio modules

At the time of writing, several LoRa radio modules are available for commercial use (Semtech Corporation 2021b), both for end-nodes and gateways. Many of which come integrated on development boards, offering developers ready-to-use hardware communication interfaces such as I2C, UART, and SPI through on-board GPIO pins. A non-exhaustive list of currently available LoRa modules for end-nodes, called the transceivers, is listed in Table E.1. Note that most radio transceivers may be configured to match a variety of use cases, thus, the same module is listed multiple times using different features. The third column indicates whether or not the module is able to detect preambles, i.e. if a LoRa transmission is about the begin, an important feature for Listen-Before-Talk (LBT) bands.

The LoRa gateways, also denoted as LoRa concentrators (Semtech Corporation 2021a), are radio chips that establish the link between the end-nodes and the LoRaWAN backend by forwarding received LoRa messages to the appropriate LNS. The chip, denoted as SX130x, come in many variants as they are region-specific, much like the LoRa transceivers.

Table E.1: LoRa radio modules

Name	Modulation	Pre. det.	Freq. range
SX1261/62/68	LoRa, FSK	-	150 - 960
SX1261/62/68	LoRa, G/F/MSK	-	150 - 960
SX1272/73	LoRa, G/F/MSK, OOK	✓	860 - 1000
SX1276/77/78/79	LoRa, G/F/MSK, OOK	✓	137 - 1020
SX1276/77/78/79	LoRa, G/F/MSK, OOK	✓	137 - 525

E.2 MCU requirements for transceivers

Depending on the LoRa chip in use, the MCU is required to meet certain minimum criteria for it to successfully integrate with the radio module (Semtech Corporation 2017). Common for both the SX126x and SX127x radio modules is that they both require at least 8KB MCU RAM and 128KB MCU Flash memory space, in addition to mandatory support for AES decryption, SPI support, and provide a Real Time Clock (RTC) for accurate time keeping. However, the Digital Input/Output (DIO) usage differ in terms of how they are connected to MCU Interrupt Request (IRQ) inputs, where SX1276x require at minimum BUSY, and DIO1, while SX127x require at minimum DIO0, DIO1, and DIO2.

Appendix F

LoRaWAN roaming

LoRaWAN deployments can also support connectivity for end-nodes where the uplink messages are intended for a different LNS than the one that the gateway is forwarding to. This feature is called roaming, and can take one of two forms, namely passive and active, where both will be briefly explained in the following.

F.1 Passive roaming

In passive roaming, the LoRa Session and the MAC-layer control of the end-node is maintained by an LNS called the Serving Network Server (SNS), and the frame forwarding to and from the air interface is handled by the LNS that manages the gateway, called the Forwarding Network Server (FNS). When the SNS and FNS are separated, they are said to be in roaming agreement, where there may be one or more FNSs serving the end-nodes, but there may only be one SNS for a given LoRa Session. The LNS where the end-nodes profile information and DevEUI is stored is called the Home Network Server (HNS), where uplink and downlink packets are forwarded between the SNS and HNS.

F.2 Handover roaming

In contrast to passive roaming, handover roaming transfers the control of the MAC-layer from one LNS to another, where the HNS maintains the control- and data-plane with the Join Server and Application Server even after the handover between LNSs has taken place. This gives the SNS the capability to control the RF settings of the end-node, allowing more flexibility.

Appendix G

Fact finding interview

G.1 Introduction

This interview is intended to function as a qualitative supplement to support the design and implementation decisions with respect to a future MIoT subsystem; soldier wearable. The interview guide is aimed primarily towards military ground forces leaders at squad or platoon level, in order to provide a better understanding of how the information flow works during a number of hypothetical cases.

G.2 Mission cases

G.2.1 CASE 1: Social patrol in urban environments during a peacekeeping mission

This case is similar to social patrols conducted by Norwegian troops during the Provincional Reconstruction Team (PRT) missions in Afghanistan, where a foot patrol is intended to interact with the local population, thereby building trust and confidence, as well as building a better understanding and awareness of what is happening in the area.

The timeline of the case is 1-4 hours, where you act as a foot-mobile infantry unit supported by aerial surveillance UAV and a QRF carrying heavy weaponry on 15 minute readiness.

Each patrol member is equipped with dual armament and one radio for internal communication. The squad leader and deputy squad leader is equipped with an additional radio for OPS reach-back.

G.2.2 CASE 2: Urban assault

This case is an offensive operation in which a platoon-sized unit is deployed to directly locate and eliminate a hostile target. The hostiles are a squad-sized unit which has taken defensive positions in multiple adjacent buildings of up to 3 floors in size, equipped with medium-weight MGs[1]

[1] MG: Machine Gun

and RPGs[2].

The timeline of the case is 3-9 hours, where you act as a platoon with 3 infantry squads dismounted from each of their armored patrol vehicles. The vehicles are equipped with a mounted machine gun, one radio for communicating with the leader of its dismounted infantry squad, and one radio for OPS reach-back.

The dismounted infantry squad are equipped with one lightweight MG. Each squad member have dual armament, one radio for squad communications, and hand grenades.

G.2.3 CASE 3: LRRP in rural terrain

This case is a typical ranger mission where a light LRRP unit of 4 men is deployed to establish visual surveillance over a large semi-static hostile force, e.g. a mobile OPS, using long-range optics. The intent is to collect information about the enemy and exfiltrate (exfil) while completely avoiding detection.

The timeline of the case is 5 days, where 1 day is used to move into position, and 1 day is used to exfiltrate (exfil) to a predesignated pickup location (i.e. by helicopter, boat, or ground vehicle, depending on the terrain), leaving a total of 3 days for surveillance while remaining static in position.

The patrol is equipped with one long-range HF radio for OPS reach-back. Each patrol member is equipped with assault rifles and carry all necessary gear for the duration of the mission in their personal backpacks. No squad radio systems is in use.

G.3 Informants

All three informants (abbreviated INF in the responses, where they are attributed with numbers 1 through 3, i.e. INF1..3) are currently serving personnel in differing units within the Norwegian Armed Forces, and have to a certain degree experience as leaders in both ranger and maneuver units.

G.4 Interview transcripts: Operational concepts

G.4.1 General

What kind of information would you generally need from the unit you are commanding (including down to each individual) throughout a mission with or without hostile activity?

INF1: As an individual soldier you largely only care about what you are to do for the next hours. You usually don't have the big perspective regarding what's happening in the world, you leave that to others. Normally you are

[2]RPG: Rocket Propelled Grenade

dealing with enough as it is with the tasks given to you. So the information you need is really the mission, where is the enemy, what can the enemy do against you, and what can you do to the enemy. What I as patrol leader need is therefore all changes in the situation that carry a meaning for you and your team. Based on the way of thinking as an individual soldier, you elevate it at bit as squad leader, and then you need to have control over your team, and preferably some information regarding the path ahead, what the company intend to do, so that you are slightly ahead so that you are ready when you receive an order from the platoon commander. You need to be able to command your platoon in accordance with the company intent, and of course higher elements' intent. If things change during the mission, and it always does, then it is important that this is received as early as possible, so that you are able to plan in accordance to those changes. Here, we are relatively weak occasionally, we have a tendency to fall a bit behind, and things go a bit too slowly. The reason for this is the ability to convey and disseminate that information into orders, and not necessarily that it goes fast enough. For example, if you're actively advancing, and you're not at the front where you tend to have control, but at the rear, rear left, rear right, then the information of the situation arrives late to you. Because everyone keeps focus on the front during an advance. Regarding comms usage, a bit after I started, the squads did not have anything, it was usually the platoon commander that had a radio, and when the squads are dispersed, it is occasionally a bit hard to control your unit. In particular when the situation escalates, there is a lot of shouting and yelling to get things done. Back then, you had a dedicated comms guy, working as an intermediary between the levels. But we didn't have radios for the squads. Later, the platoon commander got a radio.

INF2: We generally divide between 3 types of situations: blue, green, and red information. Blue means information regarding own forces, their status, personnel status, how much resources they got left, water, ammunition, etc. The red status is that particular units understanding of the enemy, with regards to observations on enemy activity. Green is information about the terrain, natural environments, sight, etc. My level of detail is platoon/company, where I am interested in as accurate descriptions of the enemy as possible, and their current location, in other words an exact grid, and if not grid, then NAI[3] blocks or TAI[4] box, and what kind of unit is it, an infantry platoon on foot, or a tank, in which case what kind of tank is it, and of course where they are going. At company level and higher you'd like to put an assessment on the situation, then you don't care as much about the accurate description, but rather how this particular enemy force relates to the bigger picture. With today's manual and analogue means of reporting, then this information needs to be as short and precise as possible, but at the same time as detailed as possible, and that is a challenge when you have tens of sensors deployed, which could be everything from static OPs to moving vehicles, UAVs and EW

[3]NAI: Named Area of Interest
[4]TAI: Target Area of Interest

sensors. In this case, blue information becomes just as important as the red information, so you know where you have your sensors, and you will know if the collected intelligence is consistent. At individual level, I carry the responsibility for the personnel. Consider a standard leadership motto, do the mission and take care of your soldiers, then I rely on information regarding how each individual is doing in order to take care of them. Then its mostly general status updates, are they OK, are they capable of doing the mission, what resources do they need. Right now I don't need any more information than what they themselves provide, whether they say they are OK or not, because that's their judgement while in the field. I'd be information overflow for me if I'm to sit there and interpret heart rates, blood pressures and such. If one have a twisted ankle, and they think they can't do the mission, then it is important for me to know that, so I can start stacking medical resources in the rear, or plan an evacuation if needed.
INF3: As maneuver platoon commander I would need personnel status, if there's been contacts with the enemy, any casualties, illnesses and such, just to know exactly how combat effective we currently are. Daily log reports of all kinds of logistics, water, ammo, fuel, etc. and status on the material, like primarily our vehicles. Then in addition we do a lot of reports regarding systems, in particular comms systems. The most important information however, is information during the mission, our units own position, contact reports, target acquisitions for those contacts, their assessments, their status after the contacts, how the enemy reacted, etc. Its the blue force and red force, and the current situation, that is the most critical information I need to coordinate both upwards and sideways. Usually, all reporting happens over text systems, whereas initial contact reports and that kind of critical information goes over voice, then a more complete contact report would be supplemented in text format later. Currently, we use 2 types of comms systems, but 3-4 different radio nets simultaneously, plus everything that goes over text, which is an additional 3 systems. But it is the voice systems that are primary for time critical information, for when I report to the brigade regarding contacts I received from the company and when I coordinate with neighboring battalions.

In general, would you prefer more or less radio equipment in the load-out? In either case, why?

INF1: Absolutely everyone could have one form for radio. Then you don't have to use so much force on your voice. I have an infantry voice, right? While today, I don't see the need to use your voice like that since we should all have internal comms. The squad leader can have comms with the platoon commander, meaning all leaders have two comms systems. Squad leader carries his own comms, platoon commander should have at least one with him. Operating on two comms is okay, but not simultaneously.
INF2: If I'm to answer that in an isolated setting, then I'd say less, because all radio equipment takes up space, weighs a lot, requires power. Power and batteries requires further resources in addition to transporting all that radio equipment. If I'm to bring more, then it'd be if we're working with

multiple types of communication systems that provides different types of information. If you can't get all types of information from one single communications system then you need more, for example due to differing frequency ranges.
INF3: Less text based systems, because a lot gets lost because of the large variety of platforms in active use. Because when a lot happens on the voice systems, and we start to receive a lot on the text systems from other actors that are not directly involved with what happens over voice, then it takes a lot of time until we are able to process that. For voice systems I'd like to keep it the same.

G.4.2 Case 1

If you were part of the patrol as a squad leader, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates, and how often? This can be either from each individual member of the unit, or from the unit as a whole.

INF1: I want continuous updates of changes, and as patrol leader I am basically responsible for talking to people over comms, and the communication with the locals is usually the deputy squad leaders responsibility. If nothing happens, then nothing happens, then you can have for example tick off and check that the comms are working, so that the deputy squad leader doesn't lose his nerve if he haven't heard anything in the last two hours. But you don't need continuous updates, you need updates of changes. If something happens, you need to know immediately. You're not interested in the meaningless chatter. In terms of granularity, I would only be interested in things that has an impact on the mission and my squad. For example, if the batteries on one of the comms is about to drop dead then of course I want to know that. I would also need to know the movements of the patrol. If a guy that stands guard behind a corner is suddenly knocked unconscious and dragged away, then that would be nice to know about, to put it that way.
INF2: Seeing as this is to be considered a routine mission, there is certain criteria on what we're to report back about. If someone sees something suspicious, then that would be reported back to OPS. Then they might provide us with a recommendation, and I'd take a stand based on that. I would also probably ask for a personnel status within the unit every hour or half hour, if they're struggling with something or anything at all. Then I'd sort of get the red picture, and the blue picture, which enables me to have a certain understanding of the situation that I can make decision on, if something happens. Depending on the red situation, then you'd have predetermined plans of action for given situation. In this case, you'd have interactions with civilians as well, I'd have one or two to communicate with the civilians, and the rest would secure the perimeter. I would want that they provide status updates regarding their sector to me and the OPS, so that I will have decision basis if something happens. This happens over

voice comms to me, then the complete situation considerations happens in my head while we conduct the mission. From me and the unit and upwards, I imagine using windows of reporting or criteria for reporting, so for example once every half hour, I'd provide a so-called GLA[5] message. What I report upwards is then my understanding of the situation based on my own observations, and what I get from my own unit, and what we get from the civilians. If I don't have any data communications with me then this would be transmitted using manual voice comms. If I am in conversations with civilians, or there is an on-going situation involving hostiles I have to handle up front, then I'd delegate the reporting and civilian interactions to the deputy squad leader, using the same format, while I take lead on the current situation in the field. In addition, if we have indications on enemy activity I would request that UAV as quickly as possible in order to either verify, or help us build SA on the enemy activity. As soon as it is circling over us, then the OPS would probably see the live feed, and they would be able to start tasking resources and support us without having us telling the OPS what to do.
INF3: On the internal comms I would have more or less continuous chatter regarding situation updates from my own patrol, or at least quite often. Things like sectors, observations, and such so that I can have a good SA. Including status on each individual. Depends on the situation, but from me to HQ, routine status updates once or twice every hour.

If you were the OpsOff stationed in OPS, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the patrol, and how often? This can be either from each individual member of the unit, or from the unit as a whole.

INF1: Then it is suddenly different, then you want updates all the time. Because then you feel the responsibility for everyone that is out on patrol. So if there is 4-6 patrols deployed, then it starts to be a little worse. Then you are preoccupied with receiving updates from the patrols as often as possible, that all is OK. Then radio checks are very nice, just to check that they are still there. Which is really annoying as patrol leader on the ground, when you're out there speaking to the locals. It's not so great to keep interrupting the conversation with the "loud and clear". You also don't think much about low-level details, you are more concerned with the bigger picture. That no one is shot and such. This lies really in the preparations, that people bring what they are supposed to, and that the plans are laid correctly. If resupplies during the mission is needed, that have to be taken into account in the planning phase. Both as platoon commander and squad leader, otherwise you don't do a proper risk assessment for your mission. If it is the case that you cannot carry everything you need for the duration of the mission, including all possible fuck-ups that might occur, then you have to plan for resupplies.

[5]GLA: Gruppering, Lokalisering, Anbefaling (i.e. grouping, location, and recommendation

INF2: Depending on what communications system we have, if we only have voice and no data comms, then I'd receive these updates over voice, which would have to be written down and converted into a visual presentation on a map, regarding where the enemy is located, where our patrol is, whether there is enemy activity, civilians, where is the UAV, etc. If we cannot provide the patrol with a map update over data comms, then that update will have to be provided over voice to everyone in a simple fashion, so that everyone knows where all friendly forces are located, what they are doing, do they have any enemy activity, and what the UAV is doing. This has to be as short and concise as possible, so as to not take up so much time on the comms. A challenge is of course the balance between providing as much information as possible and not take up so much time on the comms. In addition, it is very challenging for one individual to receive that much information at once, especiallyMbed OS Lo over voice. That is also the challenge using voice-only comms in general. Individual status updates would be too much information over comms, so if I'm to get updates from multiple deployed units, it would be from the unit as a whole.
INF3: The initial detail level would be GLA message. If nothing really happens, then that would probably be enough. However if something have happened then I'd be reaching out a bit more, depending if the information I receive from the patrols are good enough and detailed enough, or if the reporting procedures are up to standard. However if I don't feel I receive good enough reports, then I'd intervene to acquire more. If it's very quiet and nothing happens, I'd request updates every half hour, maximum.

As patrol leader, what information would you normally receive or request from OPS during the mission, other than what you've already mentioned?

INF1: Changes in the situation. Normally it's not you who is monitoring UAV feeds. That's normally someone else's job, so that comes from HQ to you. Everything that might affect your mission. For example, if there is a wedding convoy approaching dead ahead, then you want to know it. Even if it per definition isn't supposed to be dangerous, but you don't know that. I would push up status updates, radio checks, needs for alterations in the mission, if you have to do something else that isn't laid out in the plans for the mission. If something unforeseen happens, you have to report it so that you have time to react.
INF2: The most important information is the understanding of actors from higher elements in the area. If we are conducting social patrols, and higher elements receives updates on enemy activities which potentially alters enemy intent, that can affect my mission and my personnel, then that would matter, and not necessarily everything that happens within the AO[6]. It could be relevant in certain situations, what the situation is for neighboring squads, even if you don't have a direct interaction with

[6] AO: Area of Operation

them in the field, primarily to be able to support them. But this is about information overload again, in what capacity can I as squad leader handle all information. From me as squad leader and upwards to OPS, I would normally push flash messages, everything that can affect my unit as a whole, with regards to mission and/or threats. For example, if I receive threat information from a villager that insurgents have planned an attack against a logistics convoy, then I'd push that flash message immediately. Other than that, I report in accordance with routines relating to the mission.
INF3: Depending on the situation, if there's been contacts, I would have requested information from neighboring patrols, and OPS for updates on the situation. I would have pushed updates regarding own forces using the GLA message format, so that they know where we are so that we avoid blue on blue situations. Resource needs, resupply needs and such. Then of course my assessments, especially if there has been changes in the plan. I would also request information regarding the enemy situation, in particular if there's been contacts, in addition to the whereabouts of neighboring patrols and their enemy situation. So in general I would have pushed all information that affects the unit as a whole, and requested information that affects the mission.

Given the assets standing by at your disposal (UAV and QRF), how would you normally activate these, and what criteria do you feel would need to be fulfilled for you to do so? What information would you normally have to provide with the request?

INF1: I would have preferred to have my own UAV from own friendly forces, because if you have only one available covering a large area, then you are per definition a long way from friendly forces. When you conduct social patrols, then you only have a tiny patrol and you only have control over a small part of the area. Then you have to do a risk assessment, how often do you need control over external areas, how much time would an enemy need to come near you. Then the question is if you should create a time interval where the UAV circles above you, only to report back "no changes". Or should you take the choice to request the UAV when you feel something is not right. Usually when you feel something is not right, then it is too late. To activate something you need to have indicators that triggers the reason for you requesting them. The reason you request it is because you feel insecure, you're starting to become a bit nervous, that you've been static in the area for too long, or you get that feeling that something is not quite right. You have to give them a reason for them to bother sending that UAV to you. That request have to go via HQ. You can of course also plan with predefined flight paths for it to check regularly for activity in a given area. You also have to provide a reason with the request, but when you have the UAV, then you should be able to define what the UAV is supposed to do, where it should go, what you want it to look for, and then it is some operator sitting somewhere that receives that information. At the same time it will be a lot of chatter on the comms. I think that can be improved greatly.

INF2: If the UAV is only an image sensor, and not a weapon platform, then everything that that UAV can do to provide us with better SA, I would task it for. Then I would handle that via OPS, since they are normally the ones that control and prioritizes it. But it would all be about providing us with force protection with respect to hostile threats. In regards to recommendations for the QRF, I have a rather high threshold for activating it. Because if I activate it, then it can't be activated quickly for any of the other squads, then it must be a rather high threat against my squad's safety. For example, if there is an enemy OP observing us from up high while we're en-route through a choke point, and I know that I can drive a different route so as to avoid driving through that choke point, then I'd rather task the UAV in order to help me choose the best and safest possible path. If I however do not have any other option but to drive through that choke point, which means I would be ambushed, then I'd activate the QRF.
INF3: I would have requested the UAV quite often, unless someone of the neighboring patrols are in contact. I would however not request the QRF unless I've found myself in a contact that I couldn't handle myself, and if the same situation didn't involve possible assistance from neighboring forces. The information I'd send with such a request would still be the GLA message, in addition to make it clear that it is a superior enemy force, what they are doing, and then tie my recommendations to that we need the QRF, otherwise we'd be defeated. Regarding the UAV request, I'd add as much information regarding the enemy situation as possible, in order to help the UAV.

G.4.3 Case 2

If you were part of the deployed unit as the ground force commander (e.g. platoon commander), how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the deployed unit, and how often? This can be either from each individual member of the unit, or from the unit as a whole.

INF1: As platoon commander you normally only care about your squad leaders, you don't care about others. That is the squad leader's concern. The platoon commander's most important task is to divide the squads with respect to the attack itself, as force components. When you have tree squads to command for combat, it is manageable. If it's more than that, it will get increasingly harder. Therefore it is ideal to divide into tree squads, especially when you get extremely stressed out.
INF2: For simplicity, we can divide the operation into 3: reconnaissance before the mission, fire and maneuver during the mission, and finally re-organization and re-deployments after the mission. If nothing particular in the situation changes during the reconnaissance, then I'm not particularly not that interested in the details. If there is changes in the situation, then I simply expect the squad leaders to report accordingly, presumably with their recommendation to continue as planned. Short and concise mes-

sages. During the fire and maneuver, then it is important that there is as little chatter on the comms as possible so as to give the squad leaders space to conduct the mission, and not be blocked by unnecessary chatter. In this case, the radio net is open for the ones that are in contact, then I am not to block the net, nor anyone else. It is the squad leaders that are taking the building, or providing suppressing fire, that is the prioritized actors for using the comms. I would only intervene if I see something that I don't want to happen, for example if we get indications from higher elements that enemy reinforcements are en-route. When we have taken control of the building and the enemy is neutralized, then we have to re-organize, how many enemies are neutralized or taken captive, what's the status on the rooms, do we need external engineer support. Here, the chatter on the comms may start again, then I as platoon leader will need as much information as possible that I can push upwards.
INF3: I would have established a reporting wheel for everything administrative. When the timeline is this manageable then I wouldn't really need personnel log and such, which would have been reported once per 24 hours. During the mission itself, I would have required and expected to receive continuous reports regarding the squads position, localization of the enemy, contacts, contact reports, their outcomes, such as casualties, on both sides. Then of course a recommendation for further actions relative to the laid plan. Coordination between the squads would take a lot of space on the comms, usually controlled by me. Like who is the breaching team, who is covering what sector. Then afterwards, we would do a complete re-organization, a situation report from all the squads, and an assessment for further action.

If you were the OpsOff stationed in OPS, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the deployed unit, and how often? This can be either from each individual member of the unit, or from the unit as a whole.

INF1: You get continuous information normally from the platoon commander in relation to advancements. You receive requests on intelligence from the platoon commander, which you provide if you have access to it. Intelligence updates during the attack regarding the target takes a long time to reach its intended recipients. With today's technology it should be quicker to convey such information, that is, from UAV, etc., to the platoon commander. Everything that can have some form of impact on the mission should be conveyed to the platoon commander. Everything goes from platoon commander to you, it is the platoon that usually requests everything they need during the mission. In the planning phase, everything you see the need for, should be made available and be planned to be accessible at the right time.
INF2: Same principle as before, the plan is detailed and discussed, so I would only expect a GLA message, where they are grouped, located, and whether they are going ahead as planned. Then I expect to know as soon as

they are in contact, and to be kept informed by ground force commander regarding where we have our own forces, if we have any casualties, or if we've killed or captured enemies. Mostly in order to stack and make resources ready in the rear, since the mission itself is solved by the deployed unit. I as OpsOff am not to tell them how to do things, that is up to them, I am to provide them with resources and support. When I know that they are in contact, then the platoon commander has to have freedom of action, thus, he will report back to me when he has time. This is often challenging in the rear since we want to have as much control as possible, but we simply have to trust that the people in the front know what they are doing.
INF3: I wouldn't get too involved in the coordination, unless coordination between platoons is required. What I would have requested is updates on their position and advancements, including the enemy, localization and contacts, and the platoon commanders assessment. As long as that is transmitted from the commander when it is required, or when there is a change in the situation, or when it is logical that it should be reported, I would not have intervened, but if I don't receive it, I would have pushed to get that information.

As ground force commander, what information would you normally receive or request from OPS during the mission?

INF1: The platoon commander will report back how the attack is going, normally via his own comms guy, or he can communicate directly. Normally the platoon commander wants to primarily keep control over his platoon, then secondarily to keep the updates flowing to the OpsOff. You would normally push upwards everything that is not in accordance with the laid plan, the enemy's weapons, the weapons at our disposal, everything that can affect the battle that you cannot control. The squads themselves should have simple, solvable tasks.
INF2: Does the UAV see something that differs from our understanding, is EW intercepting enemy chatter regarding intent, reinforcements, etc. So everything that affects my mission.
INF3: More pressure on my understanding of the enemy, especially if I assess or experience that the enemy deviates from likely course of action, and recommendations based on that. Based on the updated situation, I might request additional resource, more people, UAV support and such. And of course assessments from neighboring units in the field if they were present, and conduct a further assessment combined with HQs understanding of the enemy situation.

In the event one member of the unit is wounded during the operation, how would this particular event affect the information flow? How would you as ground force commander handle this?

INF1: What you need to know, is if the squads medic can fix it, or if they need the platoon medic. Does he need life-saving aid, or can he remain there until the attack is over. Do we have time or do we not have time.

Can we let him die, because the mission is more important, or should we save him. If you have medical support units with you, and you are able to tell the status of the wounded soldier, they can move in and extract him, depending on their position relative to yours. Normally, during an attack, that is what would happen. As the battle ceases, it is not up to you as platoon commander to coordinate with MEDEVAC[7] or the likes, that is the medics job. You do not wish to spend resources on wounded soldiers, because this means loss of combat forces. If you have, say, a squad of 8 men, and you have to carry a wounded soldier out, you need 4 to do so. That leaves 3 men to do the actual fighting. Everyone in the squad has a specialized task, that you really need. As a squad leader, you take the exact same approach as the platoon commander.
INF2: We would have medics on the ground that would handle the medical stuff, so they are to be given the freedom of action they need to keep the wounded alive and do what they need. So my task is just to get the medical resources we need, such as reporting the 9-liner, and to task the MEDEVAC capacities we need, be it in the air or on the ground, and to evacuate the wounded. The medic would provide me with the minimum amount of information I would need to get the medical resources deployed to us, and when it arrives then that medic would conduct HOTO[8] to the MEDEVAC.
INF3: Hopefully I will receive that information rather than to request it. I would communicate internally with the affected squad first, and then get an understanding regarding whether we can continue with the mission or not. While we can still conduct the mission then I would focus on the remaining squads. I would task the platoon sergeant to organize evacuation and coordination with external medical units, while my focus would be to understand its impact on the mission, and if something needs to be changed in regards to the plan. This information I would push upwards, while also requesting necessary resources.

G.4.4 Case 3

If you were part of the deployed unit as the patrol leader, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the patrol, and how often? This can be either from each individual member of the unit, or from the unit as a whole.

INF2: We would be walking in the 50-10 pattern, so all communication in the patrol without internal comms happens using signs and signals[9]. Then we'd do a status from everyone during each 10 minute break, how is everyone doing, is anyone tired, do you need water, etc. If you have internal comms, then you don't need signs and signals as much, depending on the EW threat. Then you have time slots for reporting back to HQ, every fourth hour for instance, then you provide them with a grid, and

[7] MEDEVAC: Medical Evacuation
[8] HOTO: Handover Takeover
[9] Using predetermined physical gestures, somewhat analogous to sign language

if everything is in accordance with the plan. When we've established ourselves in a static position, then you operate more or less the same way, in addition to enemy observations.

INF3: During infil, assuming we have internal comms, we would've used it as little as possible provided we can still use signs and signals, depending on the readiness level. This is because of the possible presence of an EW threat, so I don't want to take unnecessary risks to be detected. Upwards, we would have reported in accordance with transmission windows, unless there's been situations that deviates or heavily affects the mission, such as contacts. While static in position, I would have needed status regarding battery levels on the HF, how people are doing in general, food and water, and reporting with the HF in accordance with the given transmission windows, unless we have flash messages, like if the enemy have put a tank company in formation 24 hours ahead of what we thought. For exfil we would have done much the same like infil, probably reporting on given phase lines whenever we cross these.

If you were the OpsOff stationed in OPS, how would the information flow throughout the mission with respect to status updates and radio system usage? How would you acquire status updates from the deployed unit, and how often? This can be either from each individual member of the unit, or from the unit as a whole.

INF1: Not too great to just receive very short messages on rare occasions. Wouldn't coordinate much with the patrol at all, leave them at it by themselves. There is of course situations where the mission have to be aborted, like that they are caught in TIC with the enemy, but this has to be planned.

INF2: Lets say we have windows for reporting every fourth hour, so every fourth hour we would receive a status update regarding their whereabouts, which we will plot into a map, and if they are doing OK. If everything is OK, then we continue as scheduled. If something isn't OK, such as injuries in the patrol, then we have to see if we can evacuate that individual. When they are static in position, it is more or less the same, we receive enemy observation, which we also plot into the map, or we write a single-discipline report which contains a summary of what we've observed for the last 24 hours, personnel status, and whether we're continuing as planned, before sending that to higher elements, such as analysis cells.

INF3: Then I would be completely static and waiting for the transmission windows to receive whatever they are reporting. Primarily just that, unless there is something else I need an update on. If a given number of transmission windows has been lost, then we might have to execute additional plans, such as a PACE plan. A lot less active OpsOff in this scenario compared to the second scenario, so as to not put too much noise or signature, and thus increase the risk, for the patrol.

As patrol leader, what information would you normally receive or request from OPS during the mission?

INF1: Standard observation reports, very short messages, preferably pre-made. Because counter-intelligence is super-interested in foreign signals. And they cannot be sent too often.
INF2: Where is own, most forward, friendly forces located, where is the FLOT[10]. This has a lot of impact regarding our exfil plans. What we are observing, and if there is any changes in the enemy situation, that we can put into a higher context.
INF3: Provided that everything goes in accordance with the plan, then I'd request updates regarding own forces during the transmission windows, in particular maneuver and other sensors within the same NAI, areas of interest with regards to intelligence and such.

In the event of TIC during infil or exfil, how would the information flow internally in the squad and between you and OPS?

INF1: You have to keep the squad co-located and under control, under visual control, especially at night if we are without internal comms. As OpsOff you have to consider either sacrificing them or attempt to save them from the situation. But that information must be reported back to HQ, and there must be a predefined plan for that. During infil this is pretty simple, just retreat, or get out of there. During exfil is a bit worse, but the goal is to get back to own, friendly forces. With internal comms and for example blue force tracking then you can control and steer the squad.
INF2: If we're in contact, the most important is to get out of the situation. So as soon as we've gotten out of it, we need to get control on the team, how much ammunition is left, personnel status, water, and determine if we're still able to conduct the mission as planned. If we can't, we have to execute a planned evasion plan. So what we have to report upwards, is that we've been in contact so that the OPS knows, so that they can task aerial resources for instance, in order to help us on the ground. When we've reorganized, we send a status update, personnel status, ammunition, water, and whether we're to exfil or continue as planned. If we continue with the plan, it doesn't really affect the OPS as much other than that the enemy situation has changed, and if we're pulling out then resources will be tasked in order to extract us.
INF3: I need to establish comms with HQ immediately, and I'd have to provide an update and do necessary requests, such as EVAC. Hopefully I'd have a medical plan as well, where I'd have designated pickup locations along the infil and exfil axis. Internally we'd have to handle as much as possible, as well as transporting the wounded ourselves to one of the pickup locations.

[10]FLOT: Forward Line of Own Troops

G.5 Interview transcripts: Discussing the soldier wearable

This section involves presenting an outline of the soldier wearable to the object, before hypothetically applying them to the given cases outlined in Section G.2.

G.5.1 Case 1

Hypothetically, to which level do you think using such a system would benefit you in the role as OpsOff in OPS or patrol leader in this case?

INF1: You really get full control over your squad, mainly with respect to healthcare and logistics. You automatically know the losses, if the ammunition drops drastically then we are suddenly facing something that isn't quite right. As squad leader, you will always know where people are. In combat, people get separated. In my opinion, everything that can be automated, shall, not should, shall be automated. The less time spent on your third arm, the better. Then you can focus on the mission. If somebody falls then somebody else gets that message and acts accordingly, then you as squad leader, platoon, or company commander don't have to execute everything on this earth to fix it. To do so is resource demanding. In continuous combat you need ammunition. Ammunition is a gigantic problem.
INF2: For social patrols, so-called low-intensity operations, where the enemy pressure isn't necessarily that high, not as high EW threat, then a think such a thing would be very good. In particular on the logistics side, especially water, so that we in the rear can be ready with resupplies, but also in terms of biometrics, since personnel status in such situations is very important. In a peacekeeping mission then you are more micromanaged by higher levels as well, then I think such a system is very helpful in terms of both logistics and healthcare. A challenge might be false positives, like when a person reports that he is OK, but the sensor reads the same individual as unhealthy. But that's more an error handling thing. I imagine false positives in terms of logistics is easy to handle, but false positives in terms of biometrics is a bit worse. In terms of logistics, as patrol leader, I would be very happy with the OPS being given such sensor data. However I would be more skeptical to the OPS making decision based on biometric data, because it would often be a subjective interpretation regarding health care information. Here, the patrol leader needs confidence from OPS. But if a soldier just collapses, then you wouldn't necessarily know why, but the OPS might have an indication of what it might be, for example he could be dehydrated. Therefore they could provide us on the ground with decision making pointers, which would be a safety factor for the patrol leader, provided that OPS knows how to use this information for the patrol leader rather than overriding him.
INF3: Especially as OpsOff it would be extremely useful, because it would be incredibly easy to get detailed status updates from the unit. As given in

the presentation, it would be incredibly fast as well to react on the provided data, this and that unit needs resupplies, this and that unit has MEDEVAC needs, which would save an awful lot of time spent on the radio. If I as a patrol leader had, say, a tablet or an Android device with the processed data, then that too would have been very nice, maybe more a convenience if its just for my own patrol. But if the same user interface shows detailed information about neighboring patrols as well, then I will get a lot better understanding of the situation, which would make me able to assess the situation and saved a lot of chatter time, since the core of voice chatter is assessment.

In the role as a OpsOff in OPS or patrol leader, do you have a positive or negative view regarding using such a system in this case? Please explain why.

INF1: Positive as hell, assuming that the information is aggregated correctly at all levels so that unnecessary information does not get displayed. It is easy to get stuck on details.
INF2: In social patrol scenarios, I would be positive.
INF3: Exclusively positive, as long as the size and weight of the system wouldn't affect the wearer too much.

G.5.2 Case 2

Hypothetically, to which level do you think using such a system would benefit you in the role as OpsOff in OPS or ground force commander in this case?

INF1: No difference with the previous scenario, really. Everything that helps is a good thing, in regards to acquiring information superiority. The shorter time you spend to streamline needs, medic, resupplies, reserves, then the OODA-loop gets significantly smaller, and the quicker it gets. Ideally, you don't have to say anything whatsoever, everything just happens automatically.
INF2: Just a quick disclaimer first, what you're presenting here is a disruptive technology that will alter some of the cultural procedures in the armed forces. So I don't necessarily see all the possibilities that such a system provides. Because during my education and my experience I've been affected by existing procedures and culture, and regarding this scenario, I think everything that exclusively handles logistics, especially water and ammo, I am exclusively positive towards that elements in the rear get this information. I have been working on CV90s, and they send these logistical statuses back, such as fuel and ammunition status. I have also worked with combat support, and it is great to receive this information quickly in order to provide necessary logistical support to the front. What I'm a bit more skeptical towards is the use if biometric sensors in this specific case. I don't quite see how this data is supposed to help the situation on the ground. I think it would be too much information for

a ground force commander to monitor heart rates, and other biometric conditions. I also took notice of the suggested use of gas sensors, which I think would be extremely valuable. If this defending force is using toxic gases, which could be odorless, then this would help pick up any presence of such, which could help save lives.
INF3: I think it is in this scenario that this technology would be of most value, since it is a local high-intensity scenario, complex environment and mission, lot of internal chatter, challenging to lead externally and in relation to higher elements. So this would have been an extremely good thing as platoon commander, since HQ would receive all this information a lot earlier than if I've had to provide it for them. So it would help us be one step ahead, rather than waiting for me to identify the same needs that the technology would pick up. In addition it would save a lot of time spent on the radio systems. As OpsOff I would receive time critical information quicker, which would enable me to react a lot quicker too, in addition to have a lot better understanding of the situation on the ground.

In the role as a OpsOff in OPS or ground force commander, do you have a positive or negative view regarding using such a system in this case? Please explain why.

INF1: Dissemination of information is something I think is important using such a system, but with certain constraints. If one of the teams gets eliminated, then that would be a huge demoralizing piece of information for the remaining squads to receive. So then that information shouldn't reach them so that it doesn't affect the battle. People die in war, that's a fact, that's just how it is.
INF2: As long as it made as simple as possible for the recipient of the information, then I am positive.
INF3: As long as it doesn't affect my budget, then I am exclusively positive.

G.5.3 Case 3

Hypothetically, to which level do you think using such a system would benefit you in the role as OpsOff in OPS or patrol leader in this case?

INF1: If you can lead within a bubble, and all internal communications happens within that bubble, then it isn't a problem at all using such a system. But then again, when you're in OP, the problem is being detected. By bubble, I mean that all communication that goes through the air does not leave that bubble.
INF2: With existing knowledge, then I think such a system would be very vulnerable for detection by enemy EW, which would give away the position of the patrol, especially if they are behind enemy lines. But if we disregard that, then I think it would be a very positive thing for the OpsOff to receive this kind of information. It would also provide a certain safety factor for the patrol if they can be sure that OPS knows where they are, and how much ammunition and water they have. The challenge would

be at what scale the mission is lead by in particular higher elements, this must be implemented with existing leadership culture in mind. It is for a reason they use these time slots for reporting back, so that they can focus on the mission, and to avoid detection by hostile EW. So I'm not entirely sure for such a system's use case in these scenarios, where you have to be a bit cynical and say that personnel in LRRP units isn't that important in the context of a larger, conventional scenario. If we implement a sort of alarm button that activates the sensor system when you need it, then that would be sort of a safety factor both for the patrol and OPS.
INF3: Much the same as before, except for the EW threat of course. That would be the only big challenge I think. But apart from that, this would have been very valuable. In the role as OpsOff then I'd have appreciated such a system much more, since it provides a lot more information regarding the situation on the ground. As patrol leader I have my guys directly nearby, and I wouldn't really need such automated updates. If as you say a sort of on-off button is implemented, then that would kind of compensate for the EW threat, as you use it only when it is needed. Then you could hit that button if you're caught in contact, and turn it back off when you try to stay covert after you've gotten yourself out of the situation.

In the role as a OpsOff in OPS or patrol leader, do you have a positive or negative view regarding using such a system in this case? Please explain why.

INF1: For the patrol, the OP, it is bad shit that there is signals transmitted from them, and therefore I am not a big fan of using anything wireless in such a scenario. But aside from the counter-intelligence threat, I am all for it. The exception is if you find yourself in TIC. A compromise would be to implement some big red button that activates such a system that by default is in radio-silence mode, which you could use when you find yourself in situations like TIC.
INF2: As OpsOff I would be very positive to such a system as long as what we've already talked about, in particular detection by hostile EW, is something we're aware of, and that we don't micromanage the patrol based on the sensor data. The more information you have in the rear, the more likely you are to micromanage ground forces. But in general, I'm positive.
INF3: With the exception of the EW threat, I would be very positive, provided that the on-off function is an option for us as a patrol.

G.5.4 Other comments or thoughts regarding usage of such a system, independent of the described cases outlined in Section G.2.

INF1: I want to develop as far as possible, within all thinkable categories, automated solutions for everything. As long as they are integrated and rugged enough. There is no point in having a rugged laptop that weighs 25kg. Take maximum advantage of the available technology, the people in

the field want as small a OODA-loop as possible. It is after all the ones in the front that dies.

INF2: I am a bit skeptical with regards to detection by hostile EW and micromanagement by commanding elements in the rear. So for example a large, red button that I decide to push when it is necessary to activate it, and that it is implemented as part of the leadership culture, would probably be good considerations with respect to the mentioned challenges. We see this already with BMS usage, it provides higher elements a lot more information, and it makes it possible to micromanage the units a lot more. The more streams of data from the ground, the less mission-based leadership you get, and the more micromanagement you get. But I think it is important for the armed forces to implement IoT systems, not just to get control over the blue situation, but also the red situation. Essentially, this is about the OODA-loop, and get adequate information about the enemy to actually win the battle or solve a mission, so I think part of challenge for implementing such a system, is firstly how to make this a combat-efficient system that doesn't make life more difficult for the wearer, that it works in harsh conditions, and finally what leadership culture it is supposed to be used within, so as to not facilitate for a order-controlled organization. I also think it will be certain resistance within the organization simply because the armed forces, in particular the army, is very conservative, where leaders are educated in how the Germans conducted infantry- and maneuver combat during WW2. Generally, there is a lot of resistance against new things, especially against things that there might be some insecurities surrounding its usage. For example for me, my concerns are about how this will affect the individual soldier and the leadership culture. But in general, I'm very positive for implementation of such a system, provided that it is done right. Similar systems have been a game changer with respect to the blue situation. So I think the project needs to discuss how this system fits into a assignment-based leadership, and in armed forces leadership philosophy, with regards to a very conservative officer corps. The more information the commanding officer has in regards to me and my mission then he can fall into the trap of micromanaging me, which I've experienced a lot before when using BMS.

INF3: Using such a system, with the brigade the way it is now, would have been extremely good to have, because it visualizes a lot of output, especially for mechanized forces, where there is a lot of voice chatter. 4 vehicles, and 4 infantry squads, are coordinating and talking about all kinds of things, in addition to the coordination needs between levels, such as foot-mobile units to vehicle, vehicle commander to platoon commander, platoon commander to company commander, and so on. So such a data stream that passes through all levels would save a lot of time, in addition to being a lot more accurate and precise. I imagine a challenge would be the cost of implementing and operating such a system.

Appendix H

Feedback interview

H.1 Introduction

This interview is intended to function as a means to measure the applicability of the prototype developed after the first interview, of which the input from the respondents were a crucial part towards identifying operational functionalities, and, by extension, the implementation logic.

The interview will start by presenting a high-level architecture of the system and its inner workings, and subsequently continued with a hands-on demonstration by the informant in which he or she will provide a first-impression and general feedback on the system. During the hands-on demonstration, a simulation showing the movement of a 5-man squad walking in formation through terrain will be showed, in which it will be indicated that they are caught in enemy contact, during which the patrol will sustain one injury as reported by the biometrics sensors.

After the simulation have finished executing, the respondent will be asked a series of questions which aims at acquiring the respondents first impressions, their thoughts on applying such a system in operational use, what challenges it may face, and general feedback based on the impressions they acquire during the simulation.

H.2 Interview transcripts

Hypothetically, how do you think such a system fits with existing operational patterns and doctrines?

INF1: Quite okay. Because then you don't have to keep monitoring your comms, which is the whole purpose with this, to keep tabs. You make it very easy when you're out there with personnel, and here you have control over them. And when you're sitting in HQ then you almost never have the ability to affect the situation anyway, unless you're going to drop indirect fire on your own, which shouldn't happen. I am all for this kind of stuff, because it simplifies the job when you're in HQ, and if you start gnawing at them over the comms then you stress them out. Not everyone think that's particularly great. But here you get the information directly. Such

as MEDEVAC can be executed, you know where they are, if you need fire support then you can add that directly, because you know where they are, direction, distance, everything you need. And then you can plot that in. Then they don't need to carry that kind of comms with them.
INF2: We are already doing this on vehicles today, using our most modern vehicles, like the CV90, where we have similar "biometric" data and position data for those. So this is sort of an extension to that concept for individual riflemen. And everything that provides an enhanced SA is a good thing. Like here for instance, I can see their position, their formation. The patrol leader can spend more time leading what's happening on the ground rather than keep a report with the rear, because they receive most of the information through this instead. If I as OpsOff are wondering about something, rather than hailing the unit using voice comms I can instead look at the screen, where are they, what are they doing, they are doing OK. No need to nag the patrol leader, as I can fulfill my need for information by looking at a screen. That's one thing, the other is that if something unforeseen happens, then I can prepare resources immediately when something happens, like a QRF or MEDEVAC, so that they are ready to move. So when I then get voice comms with the patrol leader saying he's in this or that position, then I can press that dispatch button, then it's just to drive out and get them. So it's really about increased operational tempo, in addition to increased SA, thus improving the decision basis for the commanders. However, as said last time, we have to be careful regarding micromanagement from the higher ups. I personally know about officers and NCOs that would use this to micromanage them, "go a bit more to the left", "don't go that way", "don't do that", which is a pitfall in itself. But that's more about leadership culture, and not the technology. As long as we manage to integrate the technology into existing leadership culture, then it is a good thing. Then it would increase the operational tempo and improve the decision basis. We have to use it the correct way. If we disregard the UI here, and consider a hypothetical end-state, then the biggest threat isn't just EW, but it's also a leadership culture that must be trained in the use of such tools, so that you don't end up micromanaging anyone. Because that happens today with our current use of BMSs, depending on who it is of course.
INF3: Provided that the sensors can handle both harsh weather and potentially a bit of a beating, then I think it is very compatible with the doctrines, the tactical and strategical operational methods and procedures we use today. As long as the hardware can handle said treatment and is compatible with existing software we already have, and that it doesn't require a lot of extra stuff, then I think it is very compatible. I hope and think that this is something that would have been embraced and viewed as a form of leadership support and seen the positive sides about this. Especially among the ones at lower levels, which are often younger, would have seen the value of this. The only potential friction in terms of leadership culture would probably be the older, senior officers, that probably would be a bit more skeptical regarding such a system. The big reward, the way I see it, is the amount of time you can save, such

as making MEDEVAC ready when you get all the information you need when something happens, rather than having someone report about it. In a leadership culture where the focus is to solve the mission and take care of your unit, I cannot see any downsides regarding such a system, as long as the mentioned prerequisites are met.

Hypothetically, do you think a full-scale deployment of such a system would enhance SA, and why? If so, at what level (i.e. in the military organization) would it be the most applicable?

INF1: Of course. You can have it at all levels, but you need aggregated levels, and that should be a locked feature. Like when the Americans got their new cameras and all that, then everyone wanted to be co-shooters. Then the management went ineffective. So you can have that at squad, platoon, company, battalion, at all levels, no problem, but it have to be aggregated. So as a brigade commander, then you see the battalion as a box, and then downwards to the patrol leader that can see all the members of the patrol as individuals. That is absolutely the biggest problem, that leaders get stuck on details they are not really supposed to have. When the brigade commander is interested in what rifleman 1 is doing then he doesn't know his own job. He is supposed to care about what the battalion is doing. For a platoon commander, he has X amount of teams or patrols, and I wouldn't give individual information to the platoon commander. I would have given the patrols or the squads to the platoon commander, their position. But usually it is the platoon commander that is responsible for executing MEDEVAC. So I would rather use APP-6A symbols for battalions and such, because it is so easy to get stuck on "where is the battalions most forward guy", but why would you need to know that? It is completely uninteresting and takes away the focus from leading larger units. This happened with the Americans during an exercise, where a 2-star general that paid close attention to the front line, the most forward squad, using those new cameras and all, it was all very fancy. That was the focus. So where is the other units then? Then you've forgotten about them, because you're too fixated on these things. They are not supposed to be watching this kind of stuff, others are.
INF2: That would probably depend on the situation, and what kind of mission you're supposed to solve. In my head, there is a lot that indicates that such a system may produce information overload during high-intensity, steel versus steel, warfare. Where it is a matter of minutes or hours until a unit has either been eliminated or eliminated the enemy. So I think in that case then this might just be an added complexity to the scenario, and not help the SA in any remarkable way. However it might help with a foot-mobile squad conducting a long flank march for instance, but that foot-mobile squad also needs to dismount from armored vehicles and attack enemy positions, so life or death could be determined within 20 seconds. However, for units conducting stabilization missions or mentoring in for instance Iraq then I think such a system have a completely different role, majorly due to the very low acceptance for loss of life during

such international missions compared to the previously mentioned large-scale warfare. So I think it is more in the low-intensity operations that such a system would truly shine, mainly at platoon and company levels. In a high-intensity operation for platoon-, company-, and battalion level, then it really doesn't matter if a couple guys dies, to be quite cynical. But that's just something to expect in such scenarios, about 10-20 percent losses. For ranger patrols and cavalry units however, where the mission are a bit more slow-paced while also operating with significantly larger time frames, then such a system would be a lot more usable. Each individual in such units are also a lot more essential, compared to combat battalions, where each individual are more or less single use during high-intensity warfare. So I think the levels where the individual soldier are important and operations where each individual are important, where there are less acceptance for loss of life, that's where such a system would really be useful. During such operations you also have the time to digest this kind of information, but during combat operations then you don't really have the capacity to handle this sort of information, and certainly not to pay attention to all details like in this simulation. So for a platoon- or company commander in combat units I don't think such a system would be all too useful due to potential information overload. However for a squad leader using for example a tablet where he receives information about his unit alone, then I think that would give him a lot. If something happens then he have to move up to the front anyway, but if he sees this prior to that, then he will already have a good understanding of the situation before arriving. So I think this could be very useful for a squad leaders and patrol leaders in combat and reconnaissance units. For company commanders and upwards I think this would be a little too high-resolution, since he is more concerned where his platoons and squads are located. So then it is up to the platoon commanders and squad leaders to care about the individual soldiers. You'll find that the lower in the organizational structure you are, the more detail-rich information you have or need. As a digression, if we can "hook" an AI into this information flow, then I think the AI would provide a more accurate decision-basis for the commanders. Since it could predict how long the ammunition would last, when should the reserves be activated, and so forth. An AI "thinks" two steps ahead compared to a human. Where the human would use for example half an hour to figure out the best course of action in a given situation, the AI might use two minutes based on all the information provided through such a system, the biometric data, the ammo count, how tired people are, and so forth, then we can predict how things will be like in the future, thus reaching a decision much quicker.

INF3: It would undoubtedly improve SA. On all levels that have access to the data basically, that have the software. Realistically at least down to platoon level. I don't know if it'd require some sort of hardware to get it out on squad level, but at least on platoon level, if it's been integrated with what we have now in the form of BMSs. Something I've really missed as a platoon commander is a live feed of the foot-mobile infantry whenever they were out, where I've had to receive a GPS position from the foot-mobile team leader and plot that manually. So if I as platoon commander have

had access to this data in a live feed, then it would have built an incredible SA at platoon, company, and battalion level. It would have been insane amounts of time saved. My biggest concern as platoon commander in a mechanized platoon is when I hear small arms fire from an area where you know you foot-mobile infantry teams are, and of course they cannot start feeding me with information. So it will take a long long time until I hear anything, since he has to handle the contact first. If we had this system, I would have received it momentarily. I would say, for contact situations, I think we would have saved, my guess, half an hour. It takes so much time for the foot-mobile to get control over the situation initially, and then do something about that, and then finally afterwards get control over himself and his own, a complete reorganization after the contact. To have had this information right away, and to be receiving continuous updates almost like a livefeed for the medical resources. So I see this as a leadership support tool in the context of stacking resources, to be not just one, maybe two steps ahead, which would potentially be very critical, in particular in cases where you have wounded. In terms of negative effects, I think there is the concern of the costs, and how much it would require from the users, as it could be an added complexity. Potentially, there could be leadership or leadership support levels, for example staffs, that is unable to handle the amount of information and transform it to something useful. Regarding the GUI, it would of course be great for the SA if I could see everything from everyone, but if I could choose to select through a filter what unit I want to see, so that I could scale according to the level you are on. But of course it would be pretty smooth if you could turn on other units, if I for example know that the Armored Battalion is conducting maneuvers in a valley east of my location, and I hear a lot is going on, and then see what is happening there. So if they've for example lost a squadron, then it's quite likely that the brigade will re-task us to reinforce them. So I think it should be possible to be a bit selective, so that you're not overflowed with information.

Would you consider the level of information in the system to be too little, or too much?

INF1: Can always add more. I like the kind of detail like "Wounded in Action", "Killed in Action". That type of thing. Because if it shows up like this then you immediately know, then you don't have to think about it much more. A suggestion would be that the squad leader could send a TIC message for instance, by pressing a button. He doesn't need to go on the comms. In the heat of battle and yelling "we're in TIC" and that kind of thing. As a patrol leader then you would want to know everything about your men. At the level above, you could just use a summary. The platoon commander doesn't need to know "Per Hansen" this and "Ole Olsen" that, he just need a status. As platoon commander all you're supposed to do is say for instance "execute MEDEVAC" this or that or just press that button. But all information regarding injuries and such that is not your concern, that is the medical units concern. If they should bring a lot of bandages or just a large black bag. They take care of that stuff. So you

should disseminate the information to the right people. But as platoon commander, I am happy with this.
INF2: To start off with the good stuff, I think the information-resolution as presented here was useful, to know where people are located in itself greatly improves the SA and saves a lot of time, where people would normally fetch their GPS and manually read the location using voice comms. It greatly improves the tempo on the battle grounds, so you don't drop artillery on your own forces, you know where not to drive if they are firing in certain directions, and so on. I also think its useful to be able to zoom in and out to see the units formation and such, since this tells me a lot about their threat assessment. That it updates real-time is also something I appreciate. I also think its good to be provided with information regarding their state, such as if they are physically exhausted, unhealthy, or healthy, as long as you know what those terms means. Perhaps you could divide these into sub-categories, is he unhealthy due to lung collapse, massive blood loss, or is he simply sick from consuming spoiled food or water. For a squad leader that would be essential to know, because then I know if I have to send in my medic in a high-risk situation or is he just sick. From the presentation I noticed the possibility of a shot counter as well, which I as squad leader would really like to know about, how much ammo do they have left. Reorganization takes a lot of time as you know, it is a highly manual task. If we use AIs to take care of that for us, then it can send the status back automatically, then the platoon commander would instantly know if we have a man down, one man exhausted, the squad is out of ammo so we must swap them with a new one, send a medic to conduct MEDEVAC, stuff that we could handle in a matter of seconds rather than spending, say, 20 minutes to assess the situation before we take concrete action.
INF3: If we've had the shot counter as well, then I think it would be spot on. To get the position and their physical state in addition to ammunition count then I think we're well covered. What I think would be a good feature, although I think it should be able to be turned on or off because I think it could cause absolute chaos, a function that shows the direction that the soldiers are pointing their rifles at. That would have been very interesting, for example if they're dug in to defensive positions, are we covered or not, and to see quickly where potential contacts may come from.

Do you consider the level of control provided to you in the role as OpsOff, team leader, or platoon commander over the devices as sufficient? Why, or why not?

INF1: As platoon commander you see where your patrols are. Critical information, even if everything is planned, where they are going to go, routes, and all that, is nice to see. Regarding the button to dispatch MEDEVAC, it depends on the information you get, you get one UNHEALTHY here, then the patrol leader himself can also press that same button, in a smart phone fashion. The platoon commander is only concerned with if the wounded soldier is still usable or not. Then the patrol

leader could for instance just press a button regarding his state.
INF2: As patrol leader I would want the information as high-res as possible, all the way down to the pulse of each individual. For example, if a person has a pulse of 200, he is most likely going to have tunnel vision or tunnel hearing, while a person having a pulse of 120 is just alert. So if two co-located soldiers have such differing pulses then it might have to with their combat reactions and such, which I would need to attend to. If I was a squad leader commanding several teams then I think what I saw here was adequate.
INF3: Action buttons as shown here is potentially a very useful feature, because you might have to execute for example MEDEVAC using a completely jammed down radio net, so to have the possibility to just push the button is just awesome, as long as the one pushing it still making the decision based on the situation. But control-wise, as both OpsOff and platoon commander, then I think this is a good level of control and gives me great grounds for decision making. But I wouldn't give up my comms just yet, the data is easy to read, but I would still have the need to talk to the people on the ground. I have the data, but I still need their assessment. A feature I think would have been very useful, although I am not quite sure how we'd have acquired the information, is if the system somehow could acquire any information about the enemy.

Other comments or thoughts regarding the implementation and deployment of such a system?

INF1: I think this is very nice. For future soldier systems, I think this should be part of it. The technology is available, and it is relatively cheap. It isn't like 20 years ago when such equipment cost millions.
INF2: If we consider end-state where each individual combat soldier have such a system and uses it, then I think we need to test the system on smaller sub-parts of the organization for it to gain success among the users. For example, within a unit, we could test it on a single platoon only to begin with, and then fine-tune the system before it is deployed to the whole unit. I imagine we should test it on squads first, preferably at the weapons school where they are very preoccupied with testing such new things. How is it for the soldiers to wear the system, how is it for the commanders to receive the information, and when you start to understand the experience to this squad alone, then you can start scaling up to larger parts of the organization. To show the users that the system works and that it makes their lives easier through incremental testing and subsequent fine-tuning I think is crucial. If you start with the whole unit, and the test results are initially bad, then you will most likely have lost the possibility for positive reception among the larger parts of the organization, then you will most likely meet resistance from the users. So small-scale, incremental testing and fine-tuning will most likely spread the word about such a system in a positive manner, for instance, platoons using the system perform better during exercises, they make decisions faster, keep a faster tempo, and so forth. So when you then reach a larger group of users, then the doors

are open, and the cultural challenges are no longer there. I would suggest using the special forces for the initial testing and fine-tuning, they are units known to think innovative and wants to test new things.

INF3: It is not a given that sitting there and hitting those action buttons as soon as they pop up is the correct tactical choice. If I've pressed or not would have been dependent on the terrain, whether there is enemy contact and how it develops, and such. That is basically what I'm concerned about, that the users need to be competent enough and experienced enough to keep making considerations, and use this as a support tool, and not that they themselves hit that button as soon as it pops up. The point is that the user needs to still keep making decisions based on experience and knowledge in addition to the data you get here, and not exclusively on the received data.

List of Abbreviations

ABP Activation By Personalization. 27

ADR Adaptive Data Rate. 25, 30, 115, 121

AES Advanced Encryption Standard. 27, 29, 126

AES-CMAC Advanced Encryption Standard-Cipher-based Message Authentication Code. 29

AES-CTR Advanced Encryption Standard-Counter Mode Encryption. 29

AI Artificial Intelligence. 15

API Application Programming Interface. 17, 64, 68, 74, 80, 81, 87, 93, 100, 106, 110

ATC Air Traffic Control. 4

BLE Bluetooth Low Energy. 43

BMS Battle Management System. ii, 4, 31, 39, 75, 99, 100, 106, 109, 113, 115, 116

BPSK Binary Phase Shift Keying. 18

C2 Command and Control. 11, 12, 34, 66, 67

C3 Command, Control, and Communication. 11

C3I Command, Control, Communication, and Intelligence. 11, 12, 116

C4ISR Command, Control, Communication, Cyber, Intelligence, Surveillance, and Reconnaissance. 11, 31, 32, 35, 45, 87, 89

CID Command Identifier. 26

COP Common Operational Picture. 32

COTS Commercial Off-The-Shelf. 2, 3, 42, 109, 110

CRC Cyclic Redundancy Check. 23

CSS Chirping Spread Spectrum. 20, 26, 119

CUPS Configuration and Update Service. 64, 89

DIO Digital Input/Output. 126

DoDAF Department of Defence Architecture Framework. 13

DR Data Rate. 26, 37, 61, 121

DSSS Direct-Sequence Spread Spectrum. 119

ECG Electrocardiography. 48, 77, 84

ECM Electronic Counter Measures. 36

EDT Emerging and Disruptive Technologies. 2

EIRP Effective Isotropically Radiated Power. 118

EMCON Emission Control. 60, 96, 100

EMG Electromyography. 48, 77

EPM Electronic Protection Measures. 8, 36

ESM Electronic Support Measures. 36

ETSI European Telecommunications Standards Institute. 118

EW Electronic Warfare. 8, 36, 57, 59–61, 65, 66, 75, 114

FFI Norwegian Defence Research Agency. 37, 42, 45

FG Functional Group. 5, 71, 72

FMN Federated Mission Networking. 41

FNS Forwarding Network Server. 127

FPGA Field Programmable Gate Array. 2

FSK Frequency Shift Keying. 26

GFC Ground Force Commander. 6, 50–56, 58, 60, 74

GPIO General Purpose Input/Output. 43, 125

GSI Global Standards Initiative. 1

GUI Graphical User Interface. 4, 96, 100, 101

HAT Hardware-Attached-on-Top. 43

HNS Home Network Server. 127

HUMINT Human Intelligence. 12

HV Home Guard. 33

I2C Inter-Integrated Circuit. 77, 80, 86, 87, 94, 95, 125

ICMP Internet Control Message Protocol. 38

IdAM Identity Management and Access Control. 35

IM Instant Messaging. 31

IMINT Imagery Intelligence. 12

IRQ Interrupt Request. 126

ISAF International Security Assistance Force. 7

ISM Industrial, Scientific, and Medical. 18, 20, 43, 60, 65, 117

ITU International Telecom Union. 1, 117

JIE Joint Information Environment. 35

JSON JavaScript Object Notation. 47, 61, 88, 91, 97, 100

KFOR Kosovo Force. 7

LBT Listen-Before-Talk. 125

LBT-AFA Listen-Before-Talk Adaptive Frequency Agility. 118

LNS LoRaWAN Network Server. 21, 22, 28, 29, 38, 62, 64, 72, 88, 89, 115, 125, 127

LoRa Long Range. 16, 18–24, 26, 28, 42–44, 47, 61–65, 68, 70, 72, 78, 87–89, 91, 110, 111, 117, 119, 125–127

LoRaWAN Long Range Wide Area Network. vi, 18, 21, 22, 25–31, 37, 39, 40, 42, 44, 45, 47, 60, 62–65, 70, 74, 78, 89, 93, 94, 97, 99, 110, 111, 119, 121, 125, 127

LoS Line-of-Sight. 37

LPD Low-Probability of Detection. 36

LPI Low-Probability of Interception. 36

LPP Low-Power Payload. 62, 65, 94, 97, 110, 111, 114, 115

LPWAN Low Power Wide Area Network. 11, 15, 16, 18, 31, 47

LRRP Long Range Recon Patrol. 8, 49, 130

M2M Machine-to-Machine. 41, 42

MAC Medium Access Control. 21, 24–28, 30, 66, 127

MANET Mobile Ad-Hoc Network. 41

MCU Microcontroller Unit. 19, 30, 63, 65, 80, 125

MIC Message Integrity Code. 24, 28, 29

MIL-STD Military Standard. 4, 35

MO Modus Operandi. 9, 109

MPE Mission Partner Environment. 35

MQTT Message Queueing Telemetry Transport. 40–42, 46, 47, 72, 74, 89, 92, 93, 97, 99, 100, 111

MQTT-SN Message Queueing Telemetry Transport for Sensor Networks. 41

NAFv4 NATO Architecture Framework Version 4. 13

NB-IoT Narrowband IoT. 16, 18, 20

NCW Network-Centric Warfare. 13, 32, 35

OODA Observe-Orient-Decide-Act. 11, 12, 30, 32, 56, 116

OP Observation Post. 8

OPS Headquarters. 49–52, 55, 129, 130, 135, 139, 142

OpsOff Operations Officer. 6, 49–51, 53, 55, 56, 60, 74, 101

OTAA Over-the-Air Activation. 27

OWASP Open Web Application Security Project. 16

PCW Platform-Centric Warfare. 12

PHY Physical Layer. 21, 24, 27, 42

PRT Provincional Reconstruction Team. 129

QoS Quality of Service. 20

QRF Quick Reaction Force. 49, 129

RF Radio Frequency. 4, 18, 20, 60, 61, 75, 117, 119

RFID Radio Frequency Identification. 2, 15, 117

RSSI Received Signal Strength Indicator. 21

RTC Real Time Clock. 126

RTT Round Trip Time. 42

SA Situational Awareness. 3, 4, 9, 13, 32, 33, 39, 45, 60, 75, 101

SDR Software-Defined Radio. 18, 42, 43

SE Secure Element. 30

SF Spreading Factor. 26, 37, 39, 40, 42, 45, 121

SIGINT Signals Intelligence. 12

SNR Signal-to-Noise Ratio. 26

SNS Serving Network Server. 127

SOA Service-Oriented Architecture. 35

SOC System-on-Chip. 2

SPI Serial Peripheral Interface. 88, 125

SSA Single Security Architecture. 35

STANAG Standardization Agreement. 4

STO Science & Technology Organization. 2

TIC Troops in Contact. 50

TPM Trusted Platform Module. 30

UART Universal Asynchronous Receiver-Transmitter. 77, 81, 95, 125

UAV Unmanned Aerial Vehicle. 49, 129

UI User Interface. 1, 4, 67, 70, 74, 75, 89, 91–93, 99, 103, 106, 114, 115

UX User Experience. 4, 75, 99, 115

WS-N Web Services Notification. 41, 46

Bibliography

3rd Generation Partnership Project (3GPP) (2016). *About 3GPP*. https://www.3gpp.org/about-3gpp, Accessed 2020.03.27.

A. Stanford-Clark, H. L. Truong (2013). *Message Queuing Telemetry Transport For Sensor Networks*. Protocol Specification Version 1.2. Organization for the Advancement of Structured Information Standards (OASIS). URL: https://www.oasis-open.org/committees/download.php/66091/MQTT-SN_spec_v1.2.pdf.

Adafruit Industries (2012). *Adafruit Ultimate GPS Breakout V3*. https://www.adafruit.com/product/746, Accessed 2021.03.25.

Advancer Technologies (2015). *MyoWare Muscle Sensor Datasheet (AT-04-001)*. http://www.advancertechnologies.com/p/myoware.html, Accessed 2021.02.20.

Alberts, David S, John J Garstka and Frederick P Stein (2000). *Network Centric Warfare: Developing and Leveraging Information Superiority.* Tech. rep. Assistant Secretary of Defense (C3I/Command Control Research Program.

Arm Mbed (2014). *SX1276MB1xAS LoRa shield*. https://os.mbed.com/components/SX1276MB1xAS/, Accessed 2021.04.10.

— (2015). *FRDM-KL25Z development board*. https://os.mbed.com/platforms/KL25Z/, Accessed 2021.04.10.

— (2017a). *DISCO L072CZ LRWAN1 development board overview*. https://os.mbed.com/platforms/ST-Discovery-LRWAN1/, Accessed 2021.03.30.

— (2017b). *Example LoRaWAN application for Mbed-OS*. https://github.com/ARMmbed/mbed-os-example-lorawan, Accessed 2020.10.20. Arm.

— (2019). *Power optimization*. https://os.mbed.com/docs/mbed-os/v6.9/apis/power-optimization.html, Accessed 2021.03.31.

— (2020). *Arm Mbed LoRaWANInterface API*. https://os.mbed.com/docs/mbed-os/v6.4/mbed-os-api-doxy/class_lo_ra_w_a_n_interface.html, Accessed 2021.01.11.

— (2021a). *I2C API reference*. https://os.mbed.com/docs/mbed-os/v6.9/apis/i2c.html, Accessed 2021.02.25.

— (2021b). *LoRaWAN API reference*. https://os.mbed.com/docs/mbed-os/v6.9/apis/lora-tech.html, Accessed 2021.03.29.

— (2021c). *Mbed OS LoRaWAN configuration*. https://os.mbed.com/docs/mbed-os/v6.9/apis/lorawan-configuration.html, Accessed 2021.03.31.

Arm Mbed OS Components Team (2014). *SerialGPS: Library for EM-406 and MTK3339 GPS modules*. https://os.mbed.com/teams/components/code/SerialGPS/, Accessed 2020.12.20.

Augustin, Aloÿs et al. (2016). 'A Study of LoRa: Long Range & Low Power Networks for the Internet of Things'. In: *Sensors* 16.9. ISSN: 1424-8220. DOI: 10.3390/s16091466. URL: https://www.mdpi.com/1424-8220/16/9/1466.

Baeyens, Pieter (May 2017). 'Use of IoT technology in a Battlefield Management System'. MA thesis. Brussels: Royal Military Academy.

Bahga, A. and M. Vijay (2014). 'Internet of things: A Hands-On Approach'. In: Arsheep Bahga & Vijay Madisetti. Chap. 5, pp. 99–120.

Barreiro, S. et al. (Nov. 2018). *Sigfox Device ETSI Mode*. Whitepaper. Sigfox. URL: https://support.sigfox.com/docs/sigfox-device-etsi-mode-white-paper.

Beitler, A. and A. Singh (2019). *LoRa Basics Station*. https://doc.sm.tc/station/, Accessed 2021.02.01. Semtech Corporation.

Biswajeeban, M. and M. Biswaranjan (June 2020). 'Evaluating and Analyzing MQTT Brokers with Stress-testing'. In:

Bloebaum, T. H. and F. T. Johnsen (2015). 'Evaluating publish/subscribe approaches for use in tactical broadband networks'. In: *MILCOM 2015-2015 IEEE Military Communications Conference*. IEEE, pp. 605–610.

Bluetooth Special Interest Group (SIG) (1989). *About Bluetooth*. https://www.bluetooth.com/specifications/bluetooth-core-specification/, Accessed 19.05.2020.

Boyd, John (1987). *The Essence of Winning and Loosing*. https://www.danford.net/boyd/, Accessed 2021.01.20.

Brehmer, Berndt (2005). 'The dynamic OODA loop: Amalgamating Boyd's OODA loop and the cybernetic approach to command and control'. In: *Proceedings of the 10th international command and control research technology symposium*. Citeseer, pp. 365–368.

Brocaar, O. (2019). *ChirpStack Architecture*. https://www.chirpstack.io/project/architecture/, Accessed 2021.04.07.

Cerrado, C., E. M. Fayo and M. Sequeira (Jan. 2020). *LoRaWAN Networks Susceptible to Hacking: Common Cyber Security Problems, How to Detect and Prevent Them*. Whitepaper. IOActive Inc.

CH2I (2018). *LoRa Gateway Shield*. https://github.com/ch2i/iC880A-Raspberry-PI, Acessed 2021.02.02.

Cisco (June 2014). *The Internet of Things reference model*. Whitepaper. Cisco Systems Inc. URL: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf.

— (2021). *Cisco Wireless Gateway for LoRaWAN*. https://www.cisco.com/c/en/us/products/routers/wireless-gateway-lorawan/index.html, Accessed 2021.02.08. Cisco Systems Inc.

Covington, M. J. and R. Carskadden (2013). 'Threat implications of the Internet of Things'. In: *2013 5th International Conference on Cyber Conflict (CYCON 2013)*, pp. 1–12.

Dragino Technology Co. (2016). *Raspberry Pi HAT featuring GPS and LoRa technology*. https://www.dragino.com/products/lora/item/106-lora-gps-hat.html, Accessed 2020.05.14.

Eclipse Foundation (2010). *libmosquitto — MQTT version 3.1.1 client library*. https://mosquitto.org/api/files/mosquitto-h.html, Accessed 2021.03.23.

— (2013). *Eclipse Paho JavaScript Client*. https://www.eclipse.org/paho/index.php?page=clients/js/index.php, Accessed 2021.03.23.
ECMA (2017). *The JSON Data Interchange Syntax, ISO/IEC 21778*. Standard ECMA-404. European Computer Manufacturers Association (ECMA). URL: https://www.ecma-international.org/publications/standards/Ecma-404.htm.
FitBit Inc. (2021). *FitBit*. https://www.fitbit.com/global/no/home, Accessed 2021.03.30.
Fongen, A. and F. Mancini (2015). 'Integrity attestation in military IoT'. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 484–489.
Fraga-Lamas, P. et al. (Oct. 2016). 'A review on internet of things for defense and public safety'. In: *Sensors* 16.10, p. 1644. ISSN: 1424-8220. DOI: 10.3390/s16101644. URL: http://dx.doi.org/10.3390/s16101644.
Ghoslya, Sakshama (2017). *LoRa decdoding*. https://www.sghoslya.com/p/lora_9.html, Accessed 2020.02.07.
Global Platform (May 2018). *Secure Element (SE) Configuration*. https://globalplatform.org/specs-library/secure-element-configuration-v2/, Accessed 2021.02.23. Version 2.0.
Global Standards Initiative (GSI) (2015). *Internet of Things Global Standards Initiative*. https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx, Accessed 2021.01.25.
Google (2009). *Google Maps JavaScript API*. https://developers.google.com/maps/documentation/javascript/overview, Accessed 2021.03.23.
Great Scott Gadgets (2014). *HackRF One SDR*. https://shop.hak5.org/products/hackrf, Accessed 2021.03.30.
Hayes, Adam (2020). *Wearable Technology*. https://www.investopedia.com/terms/w/wearable-technology.asp, Accessed 2021.04.10.
IEEE Standards Association (2016a). *IEEE Standard for an architectural framework for the internet of things, IEEE Std. 2413-2019*. Standard. Institute of Electrical and Electronics Engineers (IEEE). URL: https://standards.ieee.org/standard/2413-2019.html.
— (2016b). *IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks, IEEE 802.11-2016*. Standard. Institute of Electrical and Electronics Engineers (IEEE). URL: https://standards.ieee.org/standard/802_11-2016.html.
IETF (2011). *The WebSocket Protocol*. RFC 6455. Internet Engineering Task Force (IETF). URL: https://tools.ietf.org/html/rfc6455.
IMST (2021). *iC880A-SPI LoRa Concentrator*. https://wireless-solutions.de/products/lora-solutions-by-imst/radio-modules/ic880a-spi/, Accessed 2021.02.08.
Ionascu, M. A. (2015). *Heart Rate Monitor*. https://os.mbed.com/users/maryannionascu/code/HeartRateMonitor/, Accessed 2021.02.23.
ISO (2011). *Systems and software engineering - Architecture description, ISO/IEC/IEEE 42010:2011*. Standard. International Organization for Standardization (ISO). URL: https://www.iso.org/standard/50508.html.
— (2015). *Information Technology - Trusted Platform Module (TPM) library - Part 1: Architecture, ISO/IEC 11889*. Standard. International Organiza-

tion for Standardization (ISO). URL: https://www.iso.org/standard/66510.html.

ISO (2018). *Information technology - Radio frequency for identification for item management, ISO/IEC 18000-6:2013*. Standard. International Organization for Standardization (ISO). URL: https://www.iso.org/standard/59644.html.

ITU (2012). *Overview of the Internet of Things, ITU Recommendation Y.4000/Y.2060*. Recommendation. International Telecommunication Union (ITU). URL: https://www.itu.int/rec/T-REC-Y.2060-201206-I.

Jalaian, B. et al. (2018). 'Evaluating LoRaWAN-based IoT devices for the tactical military environment'. In: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 124–128. DOI: 10.1109/WF-IoT.2018.8355225. URL: http://dx.doi.org/10.1109/WF-IoT.2018.8355225.

Johnsen, F. T., T. H. Bloebaum, N. Jansen et al. (2019). 'Evaluating Publish/Subscribe Standards for Situational Awareness using Realistic Radio Models and Emulated Testbed'. In: *International Command and Control Research and Technology Symposium (ICCRTS) proceedings*. Norwegian Defence Research Establishment.

Johnsen, F. T., T. H. Bloebaum and P. Puente (2019). 'Towards friendly force tracking with MQTT over LoRa'. In: *International Command and Control Research and Technology Symposium (ICCRTS) proceedings*. Norwegian Defence Research Establishment.

Johnsen, F. T. and P. Ø. Puente (Nov. 2018). 'Towards IoT in a military context'. In: Eksternnotat 18/00718.

Johnsen, F. T., Z. Zieliński et al. (2018). 'Application of IoT in military operations in a smart city'. In: *2018 International Conference on Military Communications and Information Systems (ICMCIS)*. IEEE, pp. 1–8. DOI: 10.1109/ICMCIS.2018.8398690. URL: http://dx.doi.org/10.1109/ICMCIS.2018.8398690.

Kamba-Mpiana, H. et al. (2019). *Minimal printf and snprintf*. https://github.com/ARMmbed/mbed-os/blob/master/platform/source/minimal-printf/README.md, Accessed 2021.04.02. Arm.

Knight, Matthew (2017). *GNU Radio OOT module implementing the LoRa PHY*. https://github.com/BastilleResearch/gr-lora, Accessed 2021.03.30.

LatLong.net (2019). *DMS to Decimal Degrees*. https://www.latlong.net/degrees-minutes-seconds-to-decimal-degrees, Accessed 2021.03.03.

Lawson, Joel (1981). 'Command control as a process'. In: *IEEE Control Systems Magazine* 1.1, pp. 5–11.

Lohmann, N. (2013). *JSON For Modern C++*. https://json.nlohmann.me/, Accessed 2021.03.20.

LoRa Alliance (2017). *LoRaWAN Backend Interfaces*. 1.0. LoRa Alliance.

— (Feb. 2019). *LoRaWAN Security: Full end-to-end encryption for IoT application providers*. Whitepaper. LoRa Alliance. URL: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf.

— (2020a). *LoRaWAN Link-Layer Specification*. 1.0.4. LoRa Alliance.

— (2020b). *LoRaWAN Regional Parameters*. 1.0.2. LoRa Alliance.

Manyika, J. et al. (June 2015). 'Mapping the Value Beyond the Hype - Executive Summary'. In: *McKinsey Global Institute - The Internet of Things*.

Mariani, J., B. Williams and B. Loubert (2015). 'Continuing the march: The past, present, and future of the IoT in the military'. In: *Deloitte Insights*. URL: https://www2.deloitte.com/us/en/insights/focus/internet-of-things/iot-in-military-defense-industry.html.

Marr, B. (2020). *How Does Big Data Help Companies?* https://bernardmarr.com/default.asp?contentID=1738, Accessed 04.05.2021.

Mekki, K. et al. (2019). 'A comparative study of LPWAN technologies for large-scale IoT deployment'. In: *ICT Express* 5.1, pp. 1–7. ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2017.12.005. URL: http://www.sciencedirect.com/science/article/pii/S2405959517302953.

Messler, D. et al. (2018). *OWASP Internet of Things Top 10*. https://owasp.org/www-project-internet-of-things/, Accessed 2021.02.09. Open Web Application Security Project (OWASP).

Michaelis, J. et al. (2019). 'Leveraging LoRaWAN to Support IoBT in Urban Environments'. In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 207–212. DOI: 10.1109/WF-IoT.2019.8767294. URL: http://dx.doi.org/10.1109/WF-IoT.2019.8767294.

Miermont, S. and M. Coracin (2013). *LoRa Packet Forwarder*. https://github.com/Lora-net/packet_forwarder, Accessed 2020.12.19. Semtech Corporation.

MojIoT Lab (2020). *LoRa concentrator*. https://www.mojiot.com/blog/resolved-error-main-failed-to-start-the-concentrator/, Accessed 2021.04.05.

Mukherji, A. and S. Sadu (2016). 'ZigBee performance analysis'. In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 325–329.

MultiTech Systems Inc. (2018). *MultiTech Conduit IP67*. https://www.multitech.com/brands/multiconnect-conduit-ip67, Accessed 2021.02.09.

— (2021a). *LoRaWAN Gateway, Module, and Transceiver Technology*. https://www.multitech.com/technology/lorawan, Accessed 2021.02.08.

— (2021b). *MulitTech mDot Box*. https://www.multitech.com/brands/multiconnect-mdot-box, Accessed 2021.02.09.

myDevices Inc. (2018). *Cayenne Low-Power Payload*. https://developers.mydevices.com/cayenne/docs/lora, Accessed 2020.12.05.

National Institute of Standards and Technology - Federal Information Processing Standard (NIST-FIPS) (2001). 'Announcing the Advanced Encryption Standard (AES)'. In: *Federal Information Processing Standards Publication* 197.1-51, pp. 3–3.

Next-Generation Internet of Things (NGIoT) (2020). *Standardization bodies*. https://www.ngiot.eu/community/standardization-bodies/, Accessed 2020.02.02.

NMEA Standards Committee (2018). *NMEA 0183 Interface Standard*. NMEA Standard. National Marine Electronics Association (NMEA). URL: https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard.

North Atlantic Treaty Organization (NATO) (2015). *Federated Mission Networking (FMN)*. https : / / act . nato . int / activities / fmn, Accessed 2021.04.04.

North Atlantic Treaty Organization (NATO) Architecture Capability Team, Consultation, Command & Control Board (2020). *NATO Architecture Framework Version 4*. https://www.nato.int/nato_static_fl2014/assets/pdf/2021/1/pdf/NAFv4_2020.09.pdf, Accessed 2021.03.23. Version 2020.09.

OASIS (2006). *Web Services Base Notification*. OASIS Standard 1.3. Organization for the Advancement of Structured Information Standards (OASIS). URL: https://www.oasis-open.org/committees/wsn/.

OMA (Open Mobile Alliance) SpecWorks (2018). *Lightweigth M2M (LwM2M) Specification*. http : / / openmobilealliance . org / wp / OMNA / LwM2M/LwM2MRegistry.html, Accessed 2021.03.25.

Owens, William A (1996). *The emerging US system-of-systems*. 63. National Defense University, Institute for National Strategic Studies.

Pradhan, M. et al. (2019). 'Leveraging Crowdsourcing and Crowdsensing Data for HADR Operations in a Smart City Environment'. In: *IEEE Internet of Things Magazine* 2.2, pp. 26–31. ISSN: 2576-3180. DOI: 10.1109/iotm.001.1900013. URL: http://dx.doi.org/10.1109/iotm.001.1900013.

Pycom Ltd. (2017). *LoPy4 development board*. https://pycom.io/product/lopy4/, Accessed 2020.05.14.

Ray, P. P. (2015). 'Towards an Internet of Things based architectural framework for defence'. In: *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 411–416. DOI: 10.1109/ICCICCT.2015.7475314.

Reding, D. F. and J. Eaton (2020). 'Science & Technology Trends 2020-2040'. In: *NATO Science & Technology Organization, Office of the Chief Scientist, Brussels, Belgium*. URL: https://www.nato.int/nato_static_fl2014/assets/pdf/2020/4/pdf/190422-ST_Tech_Trends_Report_2020-2040.pdf.

Rjaanes, M. et al. (Aug. 2020). 'Possible implications for the Norwegian Air Force'. In: *Norwegian Defence Research Agency (FFI) Report - Technological Trends*. URL: https://publications.ffi.no/nb/item/asset/dspace:6819/20-01894.pdf.

RTL-SDR (2021). *RTL-SDR*. https://www.rtl-sdr.com/, Accessed 30.03.2021.

Russell, S. and T. Abdelzaher (2018). 'The Internet of Battlefield Things: The Next Generation of Command, Control, Communications and Intelligence (C3I) Decision-Making'. In: *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pp. 737–742. DOI: 10.1109/MILCOM.2018.8599853.

Seeed Technology Co. Ltd. (2018). *Grove Multichannel Gas Sensor V2*. https://wiki.seeedstudio.com/Grove-Multichannel-Gas-Sensor-V2/, Accessed 2021.02.25.

Semtech Corporation (2014). *LoRaMAC reference implementation and documentation of a LoRa node*. https://github.com/Lora-net/LoRaMac-node, accessed 2021.05.10.

— (May 2015). *LoRa Modulation Basics*. https : / / web . archive . org / web / 20190718200516 / https : / / www . semtech . com / uploads / documents / an1200.22.pdf, Accessed 2020.03.03.

— (2016a). *LoRa Device Mobility: An Introduction to Blind ADR*. https://lora-developers.semtech.com/library/tech-papers-and-guides/blind-adr/, Addendum 2021.01.08.
— (2016b). *Understanding Adaptive Data Rate (ADR)*. https://lora-developers.semtech.com/library/tech-papers-and-guides/understanding-adr/, Accessed 2021.01.08.
— (Dec. 2017). *MCU Requirements for LoRaWAN*. Revision 3.
— (2019a). *Building a custom integration*. LoRaWAN Academy. URL: https://lora-developers.semtech.com/learning-center/lorawan-academy/courses/building-a-custom-integration-1.
— (2019b). *Data Packet Transmissions*. Tech. rep. Semtech Corporation. URL: https://lora-developers.semtech.com/library/tech-papers-and-guides/the-book/data-packet-transmissions.
— (2019c). *Don't waste airtime!* LoRaWAN Academy. URL: https://lora-developers.semtech.com/learning-center/lorawan-academy/courses/dont-waste-airtime.
— (2019d). *LoRa Basics MAC*. Tech. rep. Semtech Corporation. URL: https://lora-developers.semtech.com/resources/tools/basic-mac/welcome-basic-mac/.
— (2019e). *The FPort Field*. Tech. rep. Semtech Corporation. URL: https://lora-developers.semtech.com/library/tech-papers-and-guides/the-book/the-port-field.
— (2021a). *LoRa Gateways*. https://www.semtech.com/products/wireless-rf/lora-gateways, Accessed 2021.04.02.
— (2021b). *LoRa Transceivers*. https://www.semtech.com/products/wireless-rf/lora-transceivers, Accessed 2021.04.02.
Søndrol, T., B. Jalaian and N. Suri (2018). 'Investigating LoRa for the Internet of Battlefield Things: A Cyber Perspective'. In: *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pp. 749–756. DOI: 10.1109/MILCOM.2018.8599805. URL: http://dx.doi.org/10.1109/MILCOM.2018.8599805.
Sourmey, Isabelle (2020). *The impact of the communication technology protocol on your IoT application's power consumption*. https://www.saftbatteries.com/energizing-iot/impact-communication-technology-protocol-your-iot-applicationÅŻ-power-consumption, Accessed 2021.01.23.
Sparkfun Electronics (2016a). *AD8322 Heart Rate Monitor Hookup Guide*. https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide?_ga=2.102135937.1690406824.1619266228-350243993.1619266228, Accessed 2021.03.25.
— (2016b). *Single Lead Heart Rate Monitor - AD8232*. https://www.sparkfun.com/products/12650, Accessed 2021.03.25.
Springer, A. et al. (2000). 'Spread spectrum communications using chirp signals'. In: *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No. 00EX405)*. IEEE, pp. 166–170.
Stanton, Neville A, Peter RG Chambers and John Piggott (2001). 'Situational awareness and safety'. In: *Safety science* 39.3, pp. 189–204.

Statista Research Department (Mar. 2020). *Global number of connected IoT devices 2015-2025*. https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/, Accessed 2020.02.02.

STMicroelectronics (n.d.). *I-CUBE LRWAN*. https://www.st.com/en/embedded-software/i-cube-lrwan.html, Accessed 2021.03.29.

Suri, N. et al. (2016). 'Analyzing the applicability of internet of things to the battlefield environment'. In: *2016 international conference on military communications and information systems (ICMCIS)*. IEEE, pp. 1–8. DOI: 10.1109/ICMCIS.2016.7496574. URL: http://dx.doi.org/10.1109/ICMCIS.2016.7496574.

The Things Industries (2017). *The Things Stack*. https://github.com/TheThingsIndustries/lorawan-stack-docs, Accessed 2021.04.11.

Tortonesi, M. et al. (2016). 'Leveraging Internet of Things within the military network environment — Challenges and solutions'. In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 111–116. DOI: 10.1109/WF-IoT.2016.7845503.

U.S. Department of Defence (2010). *The DoDAF Architecture Framework Version 2.02*. https://dodcio.defense.gov/Library/DoD-Architecture-Framework/, Accessed 2021.01.27.

W. A. Carter, D. E. Zheng (Sept. 2015). 'Leveraging the Internet of Things for a More Efficient and Effective Military'. In: *Center for Strategic & International Studies*.

W3C (2007). *SOAP Version 1.2 Part 1: Messaging Framework*. W3C Recommendation. World Wide Web Consortium (W3C). URL: https://www.w3.org/TR/soap12/.

— (2008). *Service-Oriented Architecture*. W3C Open Standard. World Wide Web Consortium (W3C). URL: https://www.w3.org/2008/11/dd-soa.html.

Weng, W. (2019). *Grove Multichannel Gas Sensor V2 Arduino Library*. https://github.com/Seeed-Studio/Seeed_Arduino_MultiGas, Accessed 2021.02.25. Seeed Technology Co. Ltd.

Wixted, A. J. et al. (2016). 'Evaluation of LoRa and LoRaWAN for wireless sensor networks'. In: *2016 IEEE SENSORS*, pp. 1–3. DOI: 10.1109/ICSENS.2016.7808712.

ZigBee Alliance (2004). *About ZigBee*. https://zigbeealliance.org/solution/zigbee/, Accessed 2020.05.19.