NTNU
Norwegian University of
Science and Technology

The Risks of Marine Cloud Computing

Adrian Alexander Eriksen

Master of Science in Communication Technology
Submission date: June 2017
Supervisor: Karin Bernsmed, IIK
Co-supervisor: Christian Frøystad, SINTEF
Per Håkon Meland, SINTEF

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title:	The Risks of Marine Cloud Computing
Student:	Adrian Alexander Eriksen

Problem description:

The maritime industry is a critical part of the European economy, which makes a disruption a significant threat to the society. The marine industry has in the later years become more reliant on information and communications technology (ICT), a dependency seen in other sectors as well.

Cloud computing gives access to a shared pool of computing resources on-demand and at possibly lower prices. The gains of utilising cloud computing should make it attractive to the maritime industry. But might also introduce new attack vectors that can disrupt the sector, leading to the research question.

What are the risks associated with using cloud-based services in maritime applications?

The master thesis project will be a case study trying to answer the research question, this will be done by performing a risk analysis on a system and obtaining a general overview of the risks of using cloud computing services within the marine sector.

The goals of the project are:

- Construct a simple application with a marine theme and deploy it into a real-life public cloud computing environment, which will act as a basis for the case study.
- Perform a risk analysis on the constructed system to point out risks for it, using a risk management framework as a support tool.
- Perform a literature study to get a general insight into the risks of cloud computing, and the cyber threats in the marine sector.
- Combine the results from the risk analysis with the literature study, to derive a general overview of the risks of using cloud computing in a marine context.

Responsible professor:	Dr. Karin Bernsmed, SINTEF Digital
Supervisor:	Christan Frøystad, SINTEF Digital

Abstract

The maritime industry has an increasing reliance on information and communication technology (ICT) systems to ensure efficient operations, moving their threat picture into the cyber domain. Meanwhile, cloud computing has impacted how ICT systems are operated, which has increased the efficiency and introduced cost savings. These benefits should be attractive to the maritime industry.

This master's thesis will investigate the risks associated with moving marine applications into cloud computing environments; it will address this problem with a divide and conquer approach. The first part is a literature review of recent research to gain an overview of issues related to general cloud computing and maritime ICT. The second part is performing a risk analysis of a marine themed Software as a Service (SaaS) solution running in a real life cloud computing environment, to get an insight into the risks related to the system.

The solution is constructed specially for this case and allows ships to report information, which port authorities can handle in a streamlined way. The risk assessment follows a qualitative approach and will give a general indication of risk.

The results from the literature review and the risk management process will provide input to approach the research question, giving an insight into multiple dimensions generating risks. The thesis will touch on issues from cloud computing, maritime environments, and human factors.

Sammendrag

Den maritime industrien har blitt mer avhengig av informasjons- og kommunikasjonsteknologi (IKT) for å sikre effektiv drift, en trend som har flyttet trusselbildet inn i den digitale verden. Samtidig har skytjenester hatt en innflytelse på hvordan IKT-systemer driftes, noe som har gitt økt effektivitet og kostnadsbesparelser. Disse fordelene burde være attraktive for den maritime industrien.

Denne masteroppgaven vil se nærmere på risikoene tilknyttet med å flytte maritime IT-tjenester inn i nettskyen, problemstillingen vil bli adressert i form av splitt og hersk. Den første delen er en litteraturstudie av nåværende forskning, dette studiet resulterer i oversikt over utfordringer tilknyttet skytjenester og maritim IKT. Den andre delen består av en risikoanalyse av en programvare som tjeneste-løsning (SaaS) med maritimt tema, dette skal gi innsikt i risikoene forbundet med systemet.

Løsningen er spesialutviklet for risikoanalysen og skal la skip rapportere inn informasjon som havnemyndigheter kan behandle på en mer strømlinjeformet måte. Risikoanalysen følger en kvalitativ metode som vil gi en generell indikasjon på risiko tilknyttet løsningen.

Resultatet fra litteraturstudiet og risikohåndteringsprosessen vil fungere som innspill for å tilnærme seg forskningsspørsmålet, som skal gi innsikt i flere dimensjoner som genererer risiko. Oppgaven vil behandle spørsmål tilknyttet skytjenester, maritime miljøer og menneskelige faktorer.

Preface

This master's thesis by Adrian Alexander Eriksen is the conclusion of the five-year master's degree in Communication Technology at the Department of Information Security and Communication Technology (IIK) at the Norwegian University of Science and Technology (NTNU). The work is done in collaboration with SINTEF.

I want to thank Karin Bernsmed and Christian Frøystad for the guidance during the spring semester. The weekly meetings have been a tremendous privilege which provided excellent and valuable feedback. This master's thesis would not be the same without your input.

Additionally, I want to thank my fellow students for making my five years in Trondheim a fantastic experience.

Contents

Appendices

List of Figures

List of Tables

List of Acronyms

API Application Programming Interface.

AWS Amazon Web Services.

CSA Cloud Security Alliance.

CSP Cloud Service Provider.

ENISA European Network and Information Security Agency.

IaaS Infrastructure as a Service.

ICT Information and Communication Technology.

IIK Department of Information Security and Communication Technology.

IoT Internet of Things.

JSON JavaScript Object Notation.

NIST National Institute of Standards and Technology.

NTNU Norwegian University of Science and Technology.

PaaS Platform as a Service.

RMF Risk Management Framework.

SaaS Software as a Service.

SLA Service Level Agreement.

Chapter 1 Introduction

The maritime industry is a critical component of the European economy; the industry accounts for over half of the goods transport within the European continent and the society's reliance on the marine sector is increasing.

The reliance on Information and Communication Technology (ICT) systems is growing, and the marine industry is no exception from this trend. The increasing reliance expands the threat picture into the cyber domain, and a report from European Network and Information Security Agency (ENISA) [1] states that a disruption in the marine ICT systems may threaten the European society. While physical security has been a priority to prevent accidents, there is a low awareness of cyber security issues.

Cloud computing gives an organisation access to a shared pool of computing resources on demand [13], which can yield benefits such as increased resource efficiency and lower costs. The benefits should make cloud computing attractive to the maritime sector, justifying the need to evaluate it.

There are security issues associated with cloud computing and efforts had been made to map out risks and issues. The Cloud Security Alliance (CSA) released a report [4] in 2016 covering the twelve most common security issues in cloud computing. The new threat picture combined with low cyber security awareness is a potential challenge [12].

We stated that cloud computing could yield potential benefits to the marine industry, while there are risks and issues related to it that must be considered. The combination of benefits and possible issues creates the foundation for the master thesis, allowing us to state the research question for this thesis, which goes as follows:

What are the risks associated with using cloud-based services in maritime applications?

This thesis is split into two parts, a literature study and a case study. The literature study tries to give us an insight into information security within cloud computing and the marine sector. The case study part is a risk analysis of a constructed Software as a Service (SaaS) solution used in a marine context; the solution is deployed into a real life cloud computing environment.

Using the results from the literature study and the case study, we will in the discussion try to generalise the findings to answer the research question. To facilitate the continuation of the work, we will try to propose new questions to work on to gain more insight into the field.

Chapter 2 Methodology

Analysing a case study will be the strategy used in this thesis to approach the research question, the case consists of a constructed application with a marine context running in a real life cloud computing environment.

We use Oates' definition of a case study as a foundation, which goes as follows:

"Case study: focuses on one instance of the 'thing' that is to be investigated: an organization, a department, a development project, an information system, a discussion forum, a systems developer, a decision and so on. The aim is to obtain a rich, detailed insight into the 'life' of that case and its complex relationships and processes." [14, p35]

Experiences, motivations, and a literature review formed the basis for the research question used in the thesis; this process helped us gain an understanding of the maritime domain and problems within it. The result of this process is the following research question:

What are the risks associated with using cloud-based services in maritime applications?

Inspired by a similar figure by Oates [14, p33], we have drawn a generalised overview of the methodology used in this master thesis shown in figure 2.1. The figure shows what created the foundation for the research question, which again led to a strategy with methods of sourcing material and analysing it.

This thesis will investigate the risks of using our cloud-based application targeted at the maritime industry; we will use a risk management framework to support this process. The output of this process is the risks specific to that system and will be used in the discussion.

In addition to our case study, a literature review will try to give us insight into

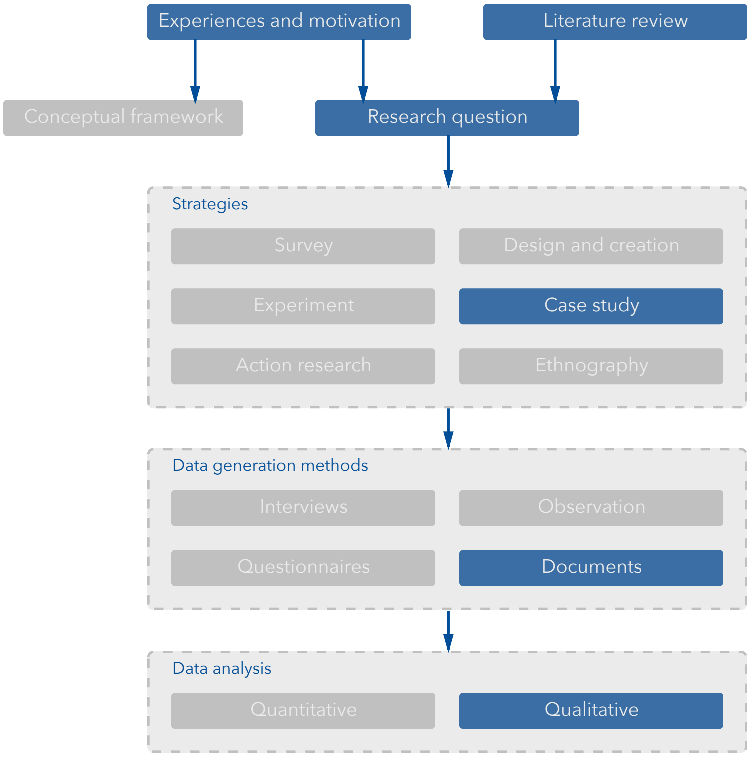


Figure 2.1: Overview of the thesis' methodology, showing the process from literature review to the case study using qualitative analysis.

where the research is today. The expected output is a more generalised overview of the threat picture. A literature search to obtain documents will serve as the primary data generation method.

By combining the specialised risks from the case study with the generalised risks from the literature review, we will try to obtain an overview of the risk picture related to using cloud services in maritime context, which will be used to address the research question.

2.1 Literature review

The literature review serves two purposes according to Oates [14]. The first is to gain an understanding of where the research is today, which can be used to create the research question. The second purpose is to gather evidence that supports the ideas presented in the thesis.

The primary goal of in this thesis is to retrieve documents from relevant and peer-reviewed academic journals and conferences to ensure the quality of the content. In addition to that, industry recognised technical reports can also be used as a source.

A challenge during the literature review is the lack of papers addressing the whole research question. A strategy to mitigate this problem is divide and conquer, where we split the literature search into multiple subthemes. An example when we want to obtain information about *cloud computing security*, where we can divide the search into general *cybersecurity* and *cloud computing*, where we merge the results later.

Scholarly search engines and university libraries serve as good starting points for this process. We will use specific keywords relevant to the subject; all used keywords will be recorded to do the search repeatable for later use. In this thesis, the following search engines will serve as a starting point:

Google Scholar Google's solution which indexes the full text and metadata of most peer-reviewed online academic journals, conferences and other scholarly sources.

Oria at NTNU University Library A search engine which keeps track of the literature available through the university library, with wide access to English and Norwegian sources.

The usage of synonyms is a strategy to ensure a wider search result. An example is to use *marine* when the original query contains the word *maritime*.

Both search engines are connected to huge databases, and we expect the results to beyond what a human can manually process. Narrowing down the number of results can be done using filtering, the year parameter can be used to obtain newer papers.

The discovered literature will be used for the discussion part of the thesis, to supplement our constructed case study. It will also be used in the case study to support the findings. The expected result is a generalised list of risks. In Oria, we will use a filter only to deliver papers from peer-reviewed sources.

The title, year, and journal name will be used in the first round of sorting papers. From this stage, we continue by reading the abstracts to determine relevance and cross check references, a paper without references will be rejected immediately. After these initial rounds, the paper will be read in detail to obtain relevant information.

The information obtained during the literature review will be used in the discussion to get an insight into the general risks related to the research question.

2.2 Case study

The goal of a case study is to gain insight into the life of the case we are studying, including its complex relationships; this is done by investigating one instance of what we want to learn more about [14].

A challenge mentioned in the literature review method (2.1) was the lack of material covering the research question in this thesis. This lack of material opens up for an exploratory study, which is used to define questions and hypothesis for subsequent studies. A real-life instance of a system to analyse must be present before performing a exploratory study.

We will in this thesis go outside the normal definition of an exploratory study. The system we construct for our case should end up as a proof-of-concept maritime SaaS solution running in a real life cloud computing environment, where the cloud operations will be an interesting technical subject to analyse.

A pitfall of this approach is that the results are specific to this case, which creates a need to generalise the results to address the research question.

2.3 Risk Management Framework

Evaluating the risks in our case study will be done using a risk management framework. We will use the ISO27005 standard [7] as our primary tool for risk management; this standard is specialised for information security risk management cases, which fits the case in our thesis.

The standard is comprehensive and implementing all aspects is unnecessary to answer our research question. The goal of this thesis is to map out the risks of a system, making the context establishment and risk assessment the interesting parts from the standard. Thus, we can take the steps of communication and consultation, monitoring and review, and risk treatment out of context of this thesis. An overview of modified risk management process is illustrated in figure 2.2.

There are two major stages in the risk management process, which are the following:

Context establishment Maps out the basic criteria, scope, and boundaries for the risk management process.

Risk assessment Lists the assessed risks prioritised according to the risk evaluation criteria. Consists of the sub steps of risk identification, risk analysis, and risk evaluation.

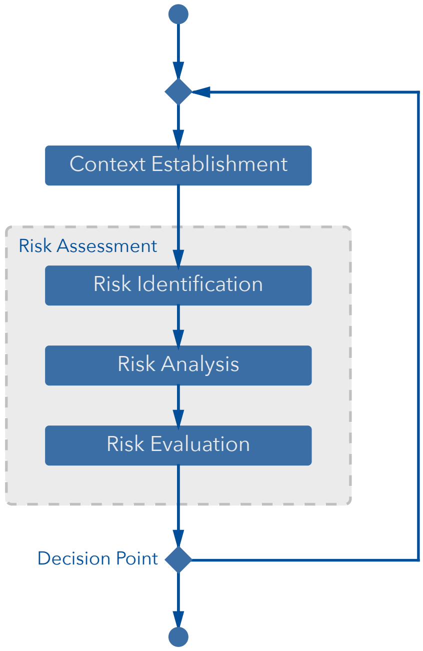


Figure 2.2: The simplified risk management process based on ISO27007 [7], showing context establishment and risk assessment as an iterative process.

The risk management process is iterative, allowing us to add new input found during the assessment continuously. The risk assessment steps lead to a decision point where we ask if results of the assessment cover enough to help us answer the research question.

2.3.1 Context Establishment

The context establishment phase of the risk management process involves establishing the context for the analysis, both internal and external; this includes steps such as setting the basic criteria, and defining the scope and boundaries.

There are different ways of approaching the risk management process, but one essential aspect to address is to establish some basic criteria for our risk management process, in our simplified process, this will be the risk evaluation criteria.

The risk evaluation criteria are the foundation used for determining a risk related to a threat, which can be used in the process of determining if a risk is acceptable or not. When determining a risk, we use a combination of likelihood and consequence; where we need to define different levels to both properties.

When we define the different levels, we need to consider what is the criteria for each of the levels. ISO27005 suggests that the following aspects should be taken into consideration:

- The criticality of the involved information assets.
- Operational importance of availability, confidentiality, and integrity.
- Expectations and perceptions of stakeholders, and negative consequences.

We should limit the risk management process by setting a scope and boundaries, ensuring that we prioritise the relevant risks; this is achieved by collecting information about the organisation to create a picture of the environment it is working in and relevance for the risk management process. The following aspects should be considered in the scope and boundaries:

- Business objectives and strategies.
- Function and structure of the organisation.
- Information assets.
- The expectation of stakeholders.
- Location and other geographical characteristics.

We are mainly interested in the technical risks associated with the constructed solution; as a consequence of this, regulatory and strategic aspects will be taken out from the scope. This decision simplifies the risk management process.

2.3.2 Risk identification

The risk identification phase maps out what could happen to cause a potential loss, gaining insight into how, where and why of the event is the essence of this step. We wish to include risks whether or not the source is under control by the organisation.

The following aspects should be identified in this phase:

Assets An asset is anything valuable for the organisation; thus, we wish to protect it.

Threats A treat is a potential harm against the organisation's assets; these come with a variety of characteristics and sources. Previous incidents and threats from other reports can also be used as input to identify threats.

Vulnerabilities Vulnerabilities can be exploited to harm an organisation's assets and can be identified in all aspects of the organisation. The presence does not cause harm itself, and action is not needed if no corresponding threat is present.

Consequences During a security incident, there may be consequences that can harm the operation of the organisation. These should be reviewed against a corresponding asset.

2.3.3 Risk analysis

The goal of this thesis is to gain a general understanding of risks associated with using cloud computing in marine environments; thus, a qualitative risk management methodology is sufficient for the risk analysis phase.

The qualitative analysis will generate a general understanding of each risk with an indication of the likelihood and the consequences. There should be a description of the likelihood and consequences with each of the risk; we should also point out the vulnerabilities creating the threat. The properties of likelihood and consequence should be classified according to the scale defined when setting the basic criteria during the context establishment.

We should take assets into considerations when we map out the consequence of a threat. The different assets affect the system differently, which again leads to different scales of severity.

After likelihood and consequence have been classified, we can determine the qualitative risk of each threat. This should be done using a risk matrix where likelihood and consequence serve as input, and the expected output is the risk level.

The expected result of this phase is a list of threats where each of them has a risk level assigned to them.

2.3.4 Risk evaluation

The risk evaluation is the final step in the simplified risk management framework, where we compare the risks to produce a prioritised list of risks. The prioritised risks should provide input for the discussion which should help us address the research question.

The results from the risk analysis act as an input for the evaluation step, where each threat has a risk level. Since there will be a finite number of qualitative risk levels, multiple risks could share the same risk level. Prioritisation of same level risks should happen considering the established context and to a qualitative consideration on which has the most impact.

We should also consider the threatened assets, threats against less valuable assets should be prioritised down. Other consequences is another factor; greater consequences may act as a reason for prioritisation.

Chapter 3 Background

To approach the research question, we can start by gaining insight into where the research is today; this is known as the literature study, which will help us source background material for this thesis.

We wish to gain an insight into cyber security risks related to cloud computing used in a marine context; the main challenge is that it is hard to find literature covering all these fields in one paper. Approaching this problem can be done using a divide and conquer strategy where we split the literature search in different themes and use the sourced material later in the discussion to answer the research question.

Cloud computing acts as a foundation for the thesis and needs to be defined; since there already is a well-made definition of cloud computing by Mell and Grance [13], it is not necessary to have a literature search to obtain it. But it should be presented before the rest of the literature study.

The literature study is split into different themes covering security and threats in cloud computing (3.1.3), maritime cloud computing (3.2), and ICT in the marine industry (3.3). To ensure repeatability to the literature search, the terms with results are documented in appendix A.

3.1 Cloud computing

Defining cloud computing is better done by pointing out characteristics of the term. National Institute of Standards and Technology (NIST) [13] says that cloud computing is a model where an organisation gets access to a shared pool of computing resources on-demand. The following five characteristics is considered as essential to the term:

On-demand self-service The consumer should be able to provision services without interaction with the service provider [13].

Broad network access The services should be available over a network and accessed through standard mechanisms [13].

Resource pooling The computing resources should be pooled together in a multi-tenant model. The resources are dynamically assigned according to demand and should be location independent [13], but location specification at a higher level – such as country or data centre – is often possible.

Rapid elasticity The delivered services should be provisioned and released elastically, often automatically by demand [13].

Measured service Resource use is controlled and optimised by using metering capabilities at some level of abstraction [13].

A cloud service is also defined by two other properties, the service model describing what is delivered, and the deployment model describing who owns and utilises the resources.

3.1.1 Service models

The service model describes what the Cloud Service Provider (CSP) delivers to the customer. The delivered solution can range from only infrastructure resources to a complete application.

Infrastructure as a Service (IaaS) The customer gets access to infrastructure components, e.g., network, processing resources, and storage. The customer controls what goes on the server, but the CSP owns and manages the hardware [13].

Platform as a Service (PaaS) The customer gets access to deploy an application onto the cloud infrastructure owned and managed by the CSP, i.e., the customers develops an application and the CSP takes care of the deployment of the application [13].

Software as a Service (SaaS) The customer gets access to a full application, which is developed and managed by the CSP. The underlying infrastructure is not available to the customers and managed by the CSP [13].

3.1.2 Deployment models

The deployment model describes the ownership of the cloud computing infrastructure. It is possible for a customer to combine different deployment models within an organisation.

Private Cloud The customer gets exclusive access to the cloud infrastructure, i.e., the resources will not be shared with others. The customer can own the underlying hardware, or it can be provided by a third-party [13].

Community Cloud The customer shares the cloud infrastructure with other customers sharing the same concerns, e.g., a cloud solution for only marine organisations. The hardware can be owned and operated by one or more customers, or a specialised third-party can deliver the service [13].

Public Cloud The customer shares the cloud resources with the general public, i.e., all customers of the CSP. The services are owned and managed by specialised CSPs, delivering resources on their premises [13].

3.1.3 Security and threats

Like every technology, there are threats associated with the usage of cloud computing services. Cloud services are attractive targets for cyber-criminals since an incident may affect multiple parties [10]. Research teams and the industry has put effort into gaining insight into security issues in the cloud; one example of effort is a technical report from the CSA [4] which lists the twelve most common threats in cloud computing.

Data breaches are the event where sensitive or confidential information is released to non-authorised parties; this is one of the highest ranked threats [4]. The source of the breach can be an intentional attack or simply human errors; this threat is not unique to cloud computing, but the multi-tenancy of cloud computing makes the threat more significant [4].

Issues related to multitenancy and virtualization are typical in cloud computing [6]. This threat is related to the characteristic of shared resources which support the characteristic of increased resource utilisation [16], where underlying components are not designed to promote strong resource isolation [4].

The data retained in the cloud should be available for the end-user at request. However, at certain points, this service can be denied to the end-user, which makes it an issue to consider [6, 16].

The deployed application should be considered as an asset which needs protection against cyber threats, secure communication between the components is essential [3]. Software security issues are relevant in this context, and an initiative from OWASP has mapped out the most common software security issues [18]. A worst case scenario in the application security context is that an application is compromised and the attack escalates into the cloud environment [6].

3.2 The Maritime Cloud

The Maritime Cloud is an effort to combine different communication solutions within the marine sector; the vision is to provide an architecture that ensures secure communication between entities in the maritime sector [12]. The goal is to prove an open gateway between different authorised stakeholders who exchange information, the integrity, confidentiality, and authenticity are guaranteed during data exchange [20].

Many use cases have been proposed for the Maritime Cloud, supporting e-navigation is one of the promising opportunities where it is possible to make the information exchange more secure. One specific improvement is to replace the current paradigm with signed and authorised paper documents, with more standardised and automated solution where the digital information is sent through the Maritime Cloud [20]. Much of the information a ship must report comes from sensors and other digital sources, by digitalising and streamlining the reporting, we can free time from the navigators and let them focus on their core task, which is safe and secure navigation.

There have been proposals on how to implement the Maritime Cloud with a draft on architecture and technical concerns [5]. A properly designed and implemented platform can yield benefits such as improved resource utilisation and more rapid application development. There is a lack of standardisation of cloud services – including on how to store user data and applications. Issues regarding users' privacy, data security, legacy services, and transition needs to be investigated; research on forensics, information management, and continuous reporting needs to be done.

The Maritime Cloud concept does not overlap with traditional cloud computing and storage [20]. Existing cloud computing services can still be relevant for the marine industry, where SaaS solutions replace in-house development and operations [9]; this will reduce the need for in-house resources to handle ICT needs; this move would allow rapid scaling of the software, and services accessed over networks, allowing the use of thinner clients.

3.3 ICT in the marine industry

ICT systems are involved in every aspect of marine operations, ranging from business functions to safety critical control systems; the computers are connected with each other which presents a security challenge [19].

A maritime organisation is complex, and in a paper by Jensen [8], a large shipping line is used as an example. The described shipping line controls offices in 150 countries and 300 vessels, and the company itself owns half of the offices and vessels, the rest is controlled by third-parties. All entities have IT infrastructure, and the organisation can't control it at the chartered vessels and the offices ran by contracted local agents. Jensen also describes the responsibility for IT operations, where the IT department handles infrastructure on shore, the marine technical department handles the IT onboard, the latter has often limited IT background.

Ports are important within the marine ecosystem and dependent on ICT infrastructure, these systems contain critical and sensitive data [15]. In a paper by Polemi et. al. [15], port ICT systems are described with seven layers and is considered secure when all seven layers satisfy the three dimensions of security. Another important idea from this paper are that the systems can't be viewed as an isolated unit, but must be considered with is multi-order dependencies from the rest of society, an example of such dependency is electricity.

A paper by Tucci [19] discusses cybersecurity issues within the US Coast Guard which has faced minor incidents, but not deliberate attacks. The events made marine terminals to adopt good cyber practice and principles to avoid accidents in the future. He points out that the maritime industry has previously had success in risk management, which needs to be expanded into the cyber domain.

Jensens [8] points out a lack of practical guidelines for cybersecurity within the marine industry, the global nature of the industry makes that task hard since guidelines at local levels tend to become conflicting. IMO is considered as an actor who can help create a consensus, but a standard process through IMO can take years.

Chapter 4
Case: A maritime cloud service

The case in this master thesis is a simplified approach clearance system delivered to the end-user using the SaaS cloud model. In general, a ship request clearance to approach a port and submits information to the solution; the port authority can get an overview of all requests and respond to them.

The information to be sent in the initial request is basic information about the ship and an estimated time of arrival, which is assigned a status as pending. The port authority can accept or deny the request; they should also be able to update the journey with actual times of arrival and departure.

The application solution will be a SaaS cloud solution, which the end-users will consume over a network connection. To be able to scale on-demand and utilise resources efficiently, the solution will run in a cloud computing environment provided by a third-party CSP.

4.1 Users and stakeholders

Before mapping out requirements, we need to have a look at who is going to use the system and other relevant stakeholders; the users of the system are the primary stakeholders and will be used in the functionality description.

The primary stakeholders are the one that will be using the system, and we are interested in including which benefits they gain from it and how they use the system. In this case, we define the roles on an organisational level, instead of a specific position or individual; thus, the end-user is a person within the defined entity.

Ship Seeking approval to approach a port, which is done by submitting journey information to the SaaS solution. Should also be able to get an overview of all its requests through the solution.

Port authority Manages the port and handles journey requests through the SaaS solution. Should be able to obtain a list of all journey requests and update status and information on single journey requests.

The secondary stakeholders do not use the SaaS solution, but they are still relevant since they affect it in some way directly or indirectly.

Development team Responsible for development of business logic and define the needed infrastructure to run the application, may have direct access to data.

CSP Owns the infrastructure and provisions it on demand, giving them physical access to the hardware containing the data. Access to data is regulated through a Service Level Agreement (SLA).

4.2 Functionality

Using our defined primary stakeholders, which will use the application, we are now able to map out how the application is used by explaining the functionality.

We can start by mapping out the typical flow on how the application is used, which gives us an indication about how the users will interact with the system. A typical use will go as follows:

1. A **ship** must be registered in the ports database before submitting a journey to the port; this step can be done by contacting and submitting the required information to the service.

2. The **ship** submits the required information, which is last and next port of call and estimated arrival time.

3. The **port authority** can fetch a list of all requested arrivals, and respond to the requests by accepting or cancelling the requests.

4. The **ship** can get a list of their requests with statuses.

5. Upon arrival and departure, the **port authority** can set actual times of arrival and departure.

4.3 Data model

The data model used in the solution is a simplified version of an approach call notification system, which has taken inspiration from similar forms from Sandnessjøen[1], Trondheim[2], and Oslo[3].

The data is modelled using a document-oriented style syntax in JavaScript Object Notation (JSON), which is a database design in the NoSQL paradigm. There is no fixed schema, meaning that the storage doesn't have hard constraints on how the data is saved, leaving it up to the business logic to ensure valid data. The advantage of this model is the flexibility. The document style does not support direct relations between data, thus, joining data in the database layer will not be available.

The data model for a vessel is simple and takes a name and home port, the latter is an object that contains country and city. The vessel name will be the value used to retrieve a document, forcing it to be unique. In real life, multiple vessels can have the same name, but in this case, we assume that names are unique.

```json
{
  "name": "Midnatsol",
  "home_port": {
    "city": "Tromsø",
    "country": "Norway"
  },
  "created_at": 1496160000,
  "updated_at": 1496160000
}
```

Journeys are the other collection of documents in our data model; it records information about ports and times of arrivals. `vessel_name` and `estimated_arrival_time` work together as an index, the combination of them must be unique. The fields for last and next port has the same object structure as the home port in a vessel document.

```json
{
  "vessel_name": "Midnatsol",
  "estimated_arrival_time": 1496314800,
  "status": 3,
```

[1] http://www.alstahaughavn.no/anloep.384674.no.html

[2] http://trondheimhavn.no/skipsanlop.aspx

[3] http://www.oslohavn.no/no/gods/priser_og_anlopsmelding/anlopsmelding/

```
  "last_port": {
    "city": "Brønnøysund",
    "country": "Norway"
  },
  "next_port": {
    "city": "Nesna",
    "country": "Norway"
  },
  "actual_arrival_time": 1496312100,
  "actual_departure_time": 1496325600,
  "created_at": 1496239200,
  "updated_at": 1496246400
}
```

There is an additional field in a journey that needs attention, the status field which tells us how the request is handled. A journey can have one of the following status values:

0 – pending Request sent from ship, but not reviewed by port.

1 – accepted Request accepted by port.

2 – docked Ship has arrived to the port.

3 – departed Ship has departed from the port.

4 – cancelled Ship or port has cancelled the journey.

4.4 The ecosystem around the solution

The ship and the port authority are the primary actors and will do all of the interaction with the system. Meanwhile, other stakeholders are indirectly involved in our solution. The developers will provide code, which is deployed in a cloud computing environment provided by a CSP.

While the stakeholders mentioned above are the most important, the solution will be a part of a much bigger ecosystem. There is infrastructure between the end-users and the CSP, which the information will flow through. While the port authorities can access the solution over a regular Internet connection, that is not the case for a ship which may be dependent on satellite links to exchange information. The satellite links may not provide perfect global coverage, opening up a possibility that the ship can't connect to the cloud solution.

4.5 Architecture on Amazon Web Services

One of the goals is to construct a simple application and deploy it into a real-life public cloud computing environment. We want to achieve this by building a server-less application, where the developer cares about the business logic, not the underlying infrastructure.

Amazon Web Services (AWS) is a provider of on-demand cloud computing resources and services with a pay-as-you-go pricing model, i.e., the customers pay for the resources consumed. Billing of resources happens by hours, GB, or per request, depending on the service. AWS provides a broad range of different services, within the IaaS and SaaS service models. This makes AWS an ideal candidate for a public cloud environment to run the application with.

Before sketching out the architecture, we need to have different components to help us construct the application which has a server-less architecture. All components should be available in the AWS ecosystem.

Amazon API Gateway is a service that let developers create and maintain an Application Programming Interface (API) for an application, which works as a public gateway to backend infrastructure containing the business logic of the application [2]. We will use this service to define the public API for our service, while Amazon will take care of the underlying infrastructure.

Amazon DynamoDB is a NoSQL database service, providing a hybrid between document and key-value data models [2]. It is a service fully managed by Amazon, and it will handle or storage needs in our application.

AWS Identity and Access Management is an access control service for services and resources on the AWS cloud platform [2]. The service allows us to define users and groups to access AWS services and specify which operations each of them is authorised to perform.

Amazon Lambda is a computing service that allows code to be executed without provisioning and managing servers [2]. This service will handle the business logic in our application, where we develop it using JavaScript. Amazon will handle deployment and scaling.

The desired outcome is a backend API which front-end systems can utilise. The solution should have a server-less architecture, where we don't need to handle issues such as scaling.

Amazon API Gateway will fill the role as a public gateway taking requests from a client and determine how to handle it. It the request is valid, it will be forwarded to Amazon Lambda which contains the business logic of our application, and is responsible for invoking other services such as Amazon DynamoDB which handles storage of our data. The architecture is drawn in figure 4.1.

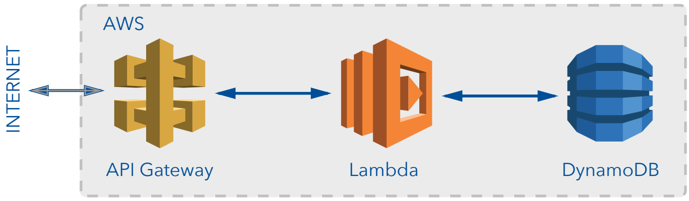


Figure 4.1: Relation between AWS components to achieve a server-less application architecture.

To give access to the constructed service to stakeholders and to restrict it from the general public, AWS Identity and Access Management is integrated into Amazon API Gateway. This service will issue and control credentials for all stakeholders, including owners, developers and users.

Chapter 5
Risk Assessment of the case

With the SaaS solution at its environment described in chapter 4, we can continue by performing the risk analysis of the case. The process will comply with the described Risk Management Framework (RMF) (2.3) defined in the methodology.

5.1 Context establishment

Before evaluating the risks, we need to establish the context for our analysis (2.3.1). We will use the context to as a foundation for evaluating possible threats and determine the risk.

The port will serve as the point of view in this analysis. The reasoning behind this is that the SaaS solution should be used to manage arrivals and departures in a single port. Thus, the port will be leasing the application.

We assume that the port represents a complex organisation which has a significant amount of employees that will use the solution.

While the ports will be the daily manager of the data and the communication in our case goes from ship to shore and back again. The ships and the port authority two entities will communicate through the Internet; when a ship is at sea, satellite links might be the only way of connecting to the Internet, introducing availability challenges. Each ship has their own credentials to the SaaS solution.

In information security, we have three security properties referred to as the CIA triad. These properties give us an indication of what we expect from the service, and goes as follows:

Confidentiality Information in the application should not be disclosed to unauthorised parties; this is critical if there are sensitive information related to a journey.

Integrity To ensure sound operations of the port, we are reliant on correct and accurate information; the solution must not threaten this.

Availability To collect information from a ship, the application must be available for journey submission. This aspect is important since there may be limited connectivity between ship and shore.

One important note on the confidentiality aspect is that information about ships are often publicly available through registers[1]. On the other side, information about a journey can be sensitive – for example in military operations. The latter justifies the need for confidentiality, since sensitive or secret information may be stored in the database.

5.1.1 Risk evaluation criteria

To estimate a risk, we need to classify the likelihood and consequence. We need indicators associated with each level for both indicators.

The foundation for classifying the consequence level of an incident is to consider the consequence against the security properties of confidentiality, availability, and integrity. Since were not only dealing with technical aspects, we also include the financial aspect as a possible source of consequence. The consequence index is provided in table 5.1.

Likelihood should give an indication how possible it is that an incident happens. Two aspects make up this indicator; the first is how often an incident can happen, while the other tells us the needed resources to create an incident. The likelihood index is provided in table 5.2.

We can derive the final risk by combing the consequence and likelihood. The scale goes from insignificant to extreme and will be used in the risk evaluation to prioritise risks. In this case, we have the following seven risk classes:

- Insignificant
- Low
- Low-Medium
- Medium
- Medium-High
- High
- Extreme

[1] Information about Norwegian ships are available trough the registers NOS and NIS.

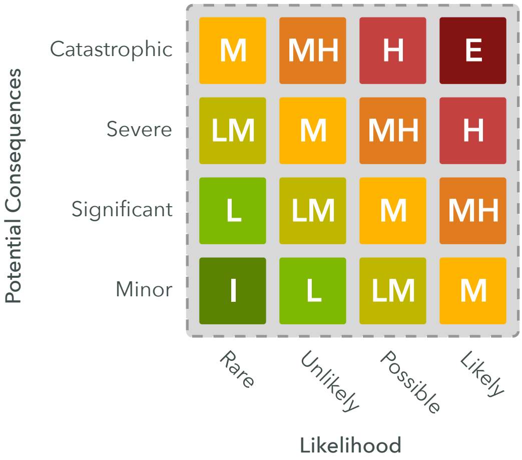
Classification	Criteria
Minor	Few minutes of service unavailability.
	Insignificant economic loss.
Significant	Few hours of service unavailability.
	Unauthorised access to a smaller data set.
	Information without significance missing or tampered.
	Recoverable economic loss.
Severe	Maximum a day of service unavailability.
	Unauthorised access to a significant data set.
	Unauthorised access to classified information.
	Critical or sensitive information partly missing or tampered.
	Economic loss impacting business.
Catastrophic	Multiple days of service unavailability or complete shutdown.
	Unauthorised access to the full data set.
	Unauthorised access to secret information.
	Critical or sensitive information missing or tampered.
	Bankrupcy due to economic loss.

Table 5.1: Consequence classification scale

Classification	Criteria
Rare	Rarer than every fifth year.
	External actors with expertise or insiders.
Unlikely	On a yearly basis.
	External actors with knowledge and good resources.
Possible	Multiple times per year.
	Actors with common knowledge and intention.
Likely	Multiple times per month.
	Negligently by an internal or external actor.

Table 5.2: Likelihood classification scale

The risk classes are mapped into a 4x4 matrix, which will be the tool used to determine the relative risk. The matrix used in this risk evaluation is shown in figure 5.1.



Potential Consequences	Rare	Unlikely	Possible	Likely
Catastrophic	M	MH	H	E
Severe	LM	M	MH	H
Significant	L	LM	M	MH
Minor	I	L	LM	M

Likelihood

Figure 5.1: Risk matrix showing risks as a product of consequences and likelihood.

5.1.2 Scope and boundaries

We assume that the constructed SaaS solution is a part of a much larger maritime ecosystem, in this case, the port authority is the end-user of the application and we assume that they run a single port. The application tries to address the following business goal:

Collect all approach and departure information from ships in a single solution available to ensure a more reliant operation of the port.

Since ships are an important actor in this solution, we consider the communication link between the ship and port as within the scope of risk management. At the same time, the port authority cannot be responsible for the ICT operation within the ship, leaving that out of the scope.

Within the port authority organisation, the ICT infrastructure is important and will be included, the same goes for physical infrastructure protecting these entities. Other physical equipment available on the harbour is considered to be out of scope for the risk assessment.

Both stakeholders expect that the system is available since ships want to submit information and the authorities need the information for its operations. The authorities have a need for precise and correct data, creating the need for data integrity.

The nature of the maritime sector is global, and we expect that the ships can be everywhere, creating a need for global availability. The port can be in an arbitrary location, but we assume it handles international ships, which also sails through arctic waters where geostationary satellites have coverage problems.

5.2 Assets

Mapping out the assets is a major step in the risk identification phase (2.3.2). We will map out a primary asset which is the most important asset in our solution; the rest is considered to be secondary assets. The primary asset will have priority when two threats are classified with the same risk. We consider the following asset as the primary asset:

A01 – **Journey information** The solution tracks all arrivals and departures at the port, including the previous and next port. All journeys are linked to a vessel.

The secondary assets do also impact our solution and need protection from malicious intent. The following assets are secondary in our risk analysis:

A02 – **Vessel information** The solution has a register over vessels, which has visited the port or approaching it. The register contains the name and the home port of the vessel.

A03 – **Source code** The source code contains all business logic of the application, including the configuration. The code is an asset maintained by the developers, and malicious intent might introduce vulnerabilities threatening the other assets.

A04 – **Provisioned cloud capacity** Our solution runs on the infrastructure provided by the CSP, as long as this contains the data, it should be considered as an asset which needs to be protected. An SLA will regulate the expected quality of service.

A05 – **Physical infrastructure** A port contains a lot of physical infrastructures. While not directly linked to our solution, the physical infrastructure also contains the terminals used to access the solution. While the solution contains data about a journey, the port must also provide facilities for the vessels to ensure good operation.

A06 – User credentials The end-users need credentials to communicate with the SaaS solution. This can be in the form of API tokens; the solution can also be extended to accept passwords which will fit into this category.

5.3 Risk analysis

We wish to gain a qualitative understanding of the risks of using cloud computing in maritime environments. Our approach is to map out threats and classify likelihood and consequences; the result is used to state the risk using the risk matrix from risk evaluation criteria (5.1.1).

5.3.1 Vulnerabilities

The vulnerabilities represent a way of harming the organisation's asset and mapping them out is a major step of the risk identification phase (2.3.2). These vulnerabilities are used to support the discovered threats, where each of the treats points to associated vulnerabilities.

General

The following vulnerabilities are general cyber security vulnerabilities that are relevant to our solution, some of them are described generically, and some has a maritime angle.

V01 – Authentication vulnerabilities Authentication is a procedure where we try to confirm the identity of an entity; failure could lead to unauthenticated access to resources.

Password, tokens, and certificates are examples of authentication methods. Amazon IAM assigns an access ID and a key to each user, which should be protected. Password-based authentication is a feature that should be considered in future development iterations since it gives end-users a more intuitive way of authenticating, but would also create a new asset that must be protected.

V02 – Authorisation vulnerabilities Authorisation is a procedure where we change the privileges to an identified entity; failure could lead to unwanted access to resources.

Amazon IAM can be used to handle authorisation, where we can specify roles and which Lambda functions they are allowed to trigger.

V03 – Communication encryption vulnerability Encrypting the communication to and from our application is essential, in our case, the Transport Layer

Security (TLS) protocol is used to achieve this. The API deployed with Amazon API Gateway takes care of deploying the service with TLS, which abstracts this responsibility from both the developers and end-users.

It is reasonable to assume that Amazon use third-party libraries – such as OpenSSL – to implement TLS in their services. Unfortunately, Multiple vulnerabilities have been discovered earlier; examples are attacks such as BEAST, CRIME, and POODLE. Given the unfortunate history of breaches, it is reasonable to assume that new vulnerabilities will be found in the future.

V04 – Linear network topology A linear network topology is a situation where there is only one path between two points, i.e. no redundancy is available. The creates a single-point-of-failure vulnerability which threatens availability

In a maritime environment, this risk is present when a vessel is at sea and relies on satellite communication. Commercial grade satellites have strict SLAs which guarantees minimal downtime, making incidents with an impact rare.

V05 – OS or dependency vulnerabilities Most software is dependent on other software, either in the form of an operating system or third-party software dependencies. These components must be secure; else they can be exploited as an attack vector into our solution.

Since we are using Lambda for our computing, we don't have access to the underlying Operating System (OS) and infrastructure. Thus, we rely on Amazon to maintain the service. We are also using the Serverless framework to build our application, which also needs to be up to date.

V06 – Poor key management Keys are used in a broad range of situations. It can be keys for cryptography or as a token to access a service.

Access to the AWS environment is done using keys generated for each user. Thus, privileges follow a key. A compromised key can in the worst case be used to access DynamoDB, compromising all stored data. Amazon takes care of handling the cryptographic keys, which can't be accessed by the developers.

Extending our application to support key based authentication is a natural candidate for the next development iteration. These keys need protection, else we risk data breach.

V07 – Poor software quality assurance The software we deploy to AWS can also be vulnerable. Following best practices and using automated tests are examples of ways to improve the quality of the product. The responsibility relies on the development team, which are responsible for the day to day maintenance of the solution.

Maritime related

The following vulnerabilities are specific to the maritime domain, but can potentially disrupt the operation of our solution.

V08 – Insecure connected control systems The automated systems onboard make the operation of the vessel easier, allowing the crew to focus on fewer tasks. The systems handle critical information ensuring reliant operation of the ship; these systems can be connected to the Internet and share important data with the staff onshore.

By taking the control systems into the Internet of Things (IoT) era, the systems are potentially vulnerable to remote attacks. Depending on the component, the consequences can range from simple data leakage to hijacking.

V09 – Satellite coverage Modern satellite networks – such as Iridium and Inmarsat – strive for global connectivity, but there will be spots where the connection is poor or not available. The orbit of a satellite determines coverage; a geostationary orbit is known to have poor coverage in the arctic regions.

V10 – Weather and natural causes While the weather will not directly affect our solution, but changing weather may affect vessels on approach to the port, which will make the estimated arrival time imprecise.

Cloud related

There are some vulnerabilities tightly linked to cloud computing, justifying the need for a separate category. Some of the vulnerabilities can be present outside cloud computing, but are more appropriate to classify as cloud computing issues in this context.

V11 – Poor resource isolation Resources for one cloud service customer should not be accessible to other customers on the same cloud. So a virtual machine belonging to one user should share the same physical hardware as other virtual machines, but act as a stand-alone machine, this is achieved by using a hypervisor.

Our solution does not use virtual machines directly and won't give us directly access to the computing resources outside what Lambda and DynamoDB offers, but it is reasonable to assume that the code is executed on shared hardware with other users of these services.

V12 – Poor resource provisioning CSPs allows us to provision capacity according to need, meaning that we can scale up when necessary. Undercapacity may threaten the reliability of the service, while overcapacity introduces unnecessary costs.

In our solution, Lambda allows us to provision memory and set a timeout limit for our process, helping us setting a cap for resource usage. DynamoDB allows us to define read and write capacity, which acts as a resource cap, but overcapacity is also possible, introducing unnecessary costs.

V13 – Vendor lock-in When an application is tightly coupled with a service provider's proprietary technology, we create a vendor lock-in since other service providers can't be used to run the application. Thus, we create a heavy reliance on one provider's services.

In our case, we are using the API Gateway, DynamoDB and Lambda services from AWS, which is only offered by them. Other CSPs can provide similar services, but moving would require rewriting parts of the solution to fit into their environments, which can increase costs.

Non-technical

In addition to the technical vulnerabilities, non-technical sources such as humans, law, and physical surroundings might be exploited. Non-technical vulnerabilities are an important aspect which often represents the weakest link and is a preferred target for malicious intent.

V14 – Inadequate physical security The physical facilities of the port should also be protected. Inadequate security can give unauthorised actors access to terminals used to interact with our solution.

V15 – Insufficient service level agreements An SLA defines the required quality characteristics for a service and is an agreement between the customer and the service provider [11, p43]. Maximum downtime is one of many properties that can be included in the agreement.

There will be multiple SLAs in our constructed solution. We assume that the developers are responsible for delivering the solution. There will be an SLA between the development team and the CSP, and there will be an agreement between the development team and the ports using the system – which are the end-users.

V16 – Lack of forensic readiness Forensic readiness prepares an organisation for future investigations after incidents. Good preparations make it easier to gain a picture of an attack. Continuous monitoring and logging are examples of measures that increases the readiness.

V17 – Missing jurisdictional information Legal concerns will also have an impact on the solution, and we need an awareness of jurisdictional aspects concerning the solution. In our case, there may be multiple jurisdictions to comply with, since the data will most likely be retained in a cloud computing environment in another country than where the end-user is.

V18 – Poor security awareness Actors working with ICT systems needs to be aware of potential threats that might affect them since a wide range of attacks is targeted against humans, not the technology itself.

The maritime industry has previously had success with their risk management, but these efforts have to be expanded into the cyber domain [19]. Meanwhile, the cyber security awareness has been low [12]. The low cyber security awareness can make the system vulnerable since crews at sea or staff at the ports are more prone to mistakes.

5.3.2 Threats

The threats represent potential harm against the organisation, and each of them is analysed to obtain the risk.

R01 – Account compromise

<table>
<tr><td colspan="2"><i>Likelihood description</i>
Common due to weak credentials set by end-users of the SaaS solution, also a preferred attack due to human involvement. Breaking the cryptography should be infeasible.</td></tr>
<tr><td colspan="2"><i>Consequence description</i>
Depends on the compromised account and its privileges, higher ranked staff leads to more severe consequences in the form of more available information.</td></tr>
<tr><td colspan="2"><i>Affected assets</i>
A01 – Journey information
A02 – Vessel information</td></tr>
<tr><td colspan="2"><i>Vulnerabilities</i>
V01 – Authentication vulnerabilities
V03 – Communication encryption vulnerability
V14 – Inadequate physical security
V18 – Poor security awareness</td></tr>
<tr><td><b>Likelihood</b></td><td>Likely</td></tr>
<tr><td><b>Consequence</b></td><td>Severe</td></tr>
<tr><td><b>Risk</b></td><td>High</td></tr>
</table>

Table 5.3: Risk description of R01 – Account compromise

This threat concerns the SaaS solution and the information residing in its environment. The credentials are not stored within our solution, but vulnerable if communication encryption is broken.

R02 – Data breach

<table>
<tr><td colspan="2"><i>Likelihood description</i>
A wide range of vulnerabilities, including human factors which is considered to be a preferred target. While breaking cryptography is infeasible to break, issues in software – both in-house developed and from third-parties – are common.</td></tr>
<tr><td colspan="2"><i>Consequence description</i>
Depends on which data is breached, but it is reasonable that a complete breach will have catastrophic consequences where the trust to the solution is highly questionable.</td></tr>
<tr><td colspan="2"><i>Affected assets</i>
A01 – Journey information
A02 – Vessel information
A06 – User credentials</td></tr>
<tr><td colspan="2"><i>Vulnerabilities</i>
V01 – Authentication vulnerabilities
V02 – Authorisation vulnerabilities
V03 – Communication encryption vulnerability
V05 – OS or dependency vulnerabilities
V07 – Poor software quality assurance
V11 – Poor resource isolation
V14 – Inadequate physical security
V18 – Poor security awareness</td></tr>
<tr><td><b>Likelihood</b></td><td>Likely</td></tr>
<tr><td><b>Consequence</b></td><td>Catastrophic</td></tr>
<tr><td><b>Risk</b></td><td>Extreme</td></tr>
</table>

Table 5.4: Risk description of R02 – Data breach

R03 – Satellite link unavailable

Likelihood description While the Internet provides redundancy, the satellite connections provide a linear topology, making this a likely event for ships sailing in Arctic regions.	
Consequence description Most network outages are fixed within hours. In the satellite case, these should also be handled in hours.	
Affected assets A01 – Journey information A02 – Vessel information	
Vulnerabilities V04 – Linear network topology V09 – Satellite coverage V15 – Insufficient service level agreements	
Likelihood	Likely
Consequence	Significant
Risk	Medium High

Table 5.5: Risk description of R03 – Network downtime

The case revolves around the ports, but vessels are important users, and their connectivity problems are relevant for this threat.

R04 – Unexpected shut down

Likelihood description The development team and third-party vendors are the sources for this threat since both are capable of shipping software with bugs. The SLA often gives an indication of maximum expected downtime that should occur.	
Consequence description We expect that the service will be restarted, which should be completed within minutes. Rollback to a previously stable version should also help avoid the problem in the future.	
Affected assets A04 – Provisioned cloud capacity	
Vulnerabilities V05 – OS or dependency vulnerabilities V07 – Poor software quality assurance V15 – Insufficient service level agreements	
Likelihood	Possible
Consequence	Minor
Risk	Low Medium

Table 5.6: Risk description of R04 – Unexpected shut down

R05 – Communication interception

Likelihood description Requires that the malicious actor can break the cryptography, which is infeasible. However, there has been discovered vulnerabilities in cryptographic libraries, but severe discoveries are rare.	
Consequence description Gives the actor access to wiretap or modify all transmitted data, which has severe consequences.	
Affected assets A01 – Journey information A02 – Vessel information A06 – User credentials	
Vulnerabilities V03 – Communication encryption vulnerability V06 – Poor key management V16 – Lack of forensic readiness	
Likelihood	Unlikely
Consequence	Severe
Risk	Medium

Table 5.7: Risk description of R05 – Communication interception

Threat affects communication between end-users and SaaS solution and can happen at any point along the network route.

R06 – Unauthorised data modification

Likelihood description Made possible by wrong authentication and authorisation, but can simply be done by account compromise which creates a human attack vector. Faulty business logic is another factor that makes this possible.	
Consequence description Faulty information which will impact the operation of the port, since the information will be used to determine availability for the port.	
Affected assets A01 – Journey information A02 – Vessel information	
Vulnerabilities V01 – Authentication vulnerabilities V02 – Authorisation vulnerabilities V07 – Poor software quality assurance V18 – Poor security awareness	
Likelihood	Possible
Consequence	Severe
Risk	Medium High

Table 5.8: Risk description of R06 – Unauthorised data modification

The SaaS solution can only modify journey and vessel data, keeping authentication data out of the threat which is handled by AWS.

R07 – Wrong information reported

Likelihood description Depends on who reports the information, humans are more prone to report wrong information than if an automated system takes care of it.	
Consequence description Faulty information which will impact the operation of the port, but less critical than an intended data modification.	
Affected assets A01 – Journey information	
Vulnerabilities V08 – Insecure connected control systems	
Likelihood	Likely
Consequence	Significant
Risk	Medium High

Table 5.9: Risk description of R07 – Wrong information reported

Threat introduced by the vessels reporting the information, not affecting other information in the solution.

R08 – Long delay or missed approach

Likelihood description Extreme weather can realise this threat, which happens a few times a year. Technical concerns may also be a reason for this threat.	
Consequence description Information about expected arrival will not match with actual times, affecting the data used for the operation of the port.	
Affected assets A01 – Journey information	
Vulnerabilities V08 – Insecure connected control systems V10 – Weather and natural causes	
Likelihood	Unlikely
Consequence	Severe
Risk	Medium

Table 5.10: Risk description of R08 – Long delay or missed approach

R09 – Denial of service against SaaS solution

Likelihood description A common attack on the Internet often done with the help of a botnet, the techniques are primitive and have low knowledge requirements. The attack can often be intended, but in some cases, the service is a random target.	
Consequence description Threatens availability by exhausting the available resources, making the service unavailable. Lasts until attack stops or defensive measures are deployed.	
Affected assets A04 – Provisioned cloud capacity	
Vulnerabilities V05 – OS or dependency vulnerabilities V12 – Poor resource provisioning	
Likelihood	Possible
Consequence	Significant
Risk	Medium

Table 5.11: Risk description of R09 – Denial of service

Resource exhausting will affect one of the AWS services with a resource usage cap. Vulnerabilities in software can accelerate exhaustion.

R10 – Lock-in

Likelihood description Our SaaS solution is tightly coupled with AWS, meaning that this threat is always present and it relies on the CSP to provide their services. New laws are continuously introduced, which may also trigger this threat.	
Consequence description A lock-in might make the data hard to access the services provided by the CSP, and in worst case make the data unavailable for the user. Another problem is that it might be hard and pricey to change CSP.	
Affected assets A01 – Journey information A02 – Vessel information A04 – Provisioned cloud capacity A06 – User credentials	
Vulnerabilities V07 – Poor software quality assurance V13 – Vendor lock-in V15 – Insufficient service level agreements V17 – Missing jurisdictional information	
Likelihood	Likely
Consequence	Catastrophic
Risk	Extreme

Table 5.12: Risk description of R10 – Lock-in

R11 – Changing jurisdictions

Likelihood description New jurisdictional notices are passed continuously to follow the technology development; the process takes time and happens a few times a year.	
Consequence description The time of passing new laws makes it possible to move data out of the jurisdiction before the law is adopted, which should end up with a minor economical loss.	
Affected assets A01 – Journey information A02 – Vessel information A03 – Source code A04 – Provisioned cloud capacity A05 – Physical infrastructure A06 – User credentials	
Vulnerabilities V17 – Missing jurisdictional information	
Likelihood	Unlikely
Consequence	Significant
Risk	Low Medium

Table 5.13: Risk description of R11 – Changing jurisdictions

The assumption is that both the cloud service and the end-user resides in stable jurisdictions where the process of passing new laws is transparent.

Laws and jurisdictional notices can affect every aspect of the society. Thus, all assets may be threatened.

R12 – Incomplete data deletion

<table>
<tr><td colspan="2"><i>Likelihood description</i>
A rare software error when using battle-tested libraries and operating systems.</td></tr>
<tr><td colspan="2"><i>Consequence description</i>
Creates a small risk of unauthorised data access, but physical access to cloud storage is restricted.</td></tr>
<tr><td colspan="2"><i>Affected assets</i>
A01 – Journey information
A02 – Vessel information
A03 – Source code
A04 – Provisioned cloud capacity
A06 – User credentials</td></tr>
<tr><td colspan="2"><i>Vulnerabilities</i>
V05 – OS or dependency vulnerabilities
V07 – Poor software quality assurance</td></tr>
<tr><td><b>Likelihood</b></td><td>Rare</td></tr>
<tr><td><b>Consequence</b></td><td>Significant</td></tr>
<tr><td><b>Risk</b></td><td>Low</td></tr>
</table>

Table 5.14: Risk description of R12 – Incomplete data deletion

R13 – Malicious insiders

<table>
<tr><td colspan="2"><i>Likelihood description</i>
Multiple human factors and complex organisations make yearly incidents possible.</td></tr>
<tr><td colspan="2"><i>Consequence description</i>
Worst case is leakage of secret information which may be catastrophic for the trust.</td></tr>
<tr><td colspan="2"><i>Affected assets</i>
A01 – Journey information
A02 – Vessel information
A03 – Source code
A04 – Provisioned cloud capacity
A05 – Physical infrastructure
A06 – User credentials</td></tr>
<tr><td colspan="2"><i>Vulnerabilities</i>
V02 – Authorisation vulnerabilities
V16 – Lack of forensic readiness
V18 – Poor security awareness</td></tr>
<tr><td><b>Likelihood</b></td><td>Possible</td></tr>
<tr><td><b>Consequence</b></td><td>Catastrophic</td></tr>
<tr><td><b>Risk</b></td><td>High</td></tr>
</table>

Table 5.15: Risk description of R13 – Malicious insiders

R14 – Cloud service outage

Likelihood description Data on previous outages show two events per year as a maximum; Amazon Web Services has suffered from major incidents causing regions or services to go down.	
Consequence description Outage for several hours has been the worst case, threatening availability. Events causing permanent data loss is unknown.	
Affected assets A01 – Journey information A02 – Vessel information A04 – Provisioned cloud capacity A06 – User credentials	
Vulnerabilities V05 – OS or dependency vulnerabilities V12 – Poor resource provisioning	
Likelihood	Unlikely
Consequence	Significant
Risk	Low Medium

Table 5.16: Risk description of R14 – Cloud service outage

One of biggest happened in October 2012, causing popular site on the Internet to go down. Recently in March 2017, the Northern Virginia region went down for a few hours.

Our web service is directly dependent on three AWS services, which again may be dependent on other services within the ecosystem; each service represents a vulnerability on its own. Our service is residing in the EU West region, which may have downtime.

5.4 Risk evaluation

An overview of the threats with their risks is presented in figure 5.2, where the axises represent likelihood and consequences.

The matrix lets us roughly compare the risks against each other, but we see that many risks share the same risk classification, which we need details to prioritise.

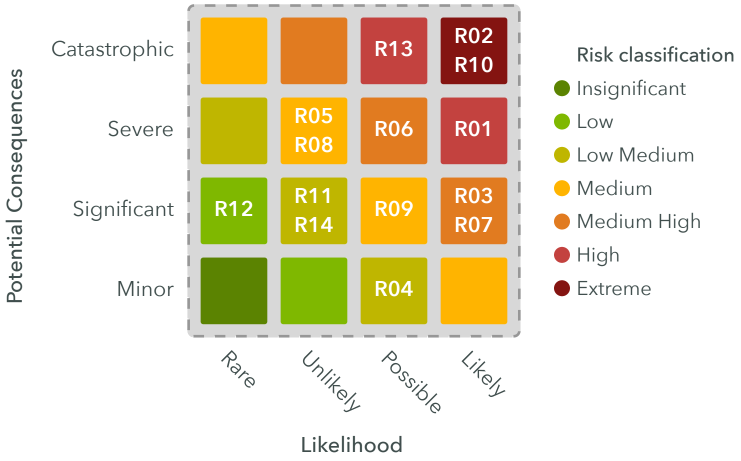


Figure 5.2: Risk matrix showing threats classified with a risk.

We wish to create a final list of risks, but to limit the risk comparing, we will only discuss the risks classified as medium or higher.

We have two threats that should be considered as extreme and needs priority, data breaches (R02) and lock-ins (R10). Both threatens the complete data set, but the latter also targets the cloud computing infrastructure. The dependency is a consequence of the reliance on specific service provided by the CSP, which other CSPs cannot offer direct replacements for. Thus, the lock-in vulnerability gets prioritised.

Account compromise (R01) can affect the journey and vessel data, depending on the privileges of the compromised account. The other risk classified as high is malicious insiders (R13), which can affect every aspect of the ecosystem around the solution. The latter takes the priority since it can affect more assets.

Unauthorised data modification (R06) takes priority since it can represent a permanent integrity breach to the whole data set. Wrong information reported to our system (R07) may affect the operation of the port – which relies on journey information which is our primary subject of the risk assessment. Thus, the latter risk is the second priority, while the satellite link threat (R03) gets the lowest since it only affects single ships.

The denial of service threat (R09) affects primarily the provisioned cloud capacity, not the journey data; this allows us to give this threat a low priority. Communication interception (R05) has severe consequences and may affect the whole data set, depending on what traffic that is intercepted. The long delay or missed approach (R08) will threaten the integrity of the data, affecting the operations of the port.

A list of the prioritised risks can be found in table 5.17, where the threats with equal risk are prioritised according to the discussion.

Threat	Risk
R10 – Lock-in	Extreme
R02 – Data breach	Extreme
R13 – Malicious insiders	High
R01 – Account compromise	High
R06 – Unauthorised data modification	Medium High
R07 – Wrong information reported	Medium High
R03 – Satellite link unavailable	Medium High
R05 – Communication interception	Medium
R08 – Long delay or missed approach	Medium
R09 – Denial of service	Medium

Table 5.17: Threats ranked by risk classification

Chapter 6 Discussion

The primary goal of this thesis is to evaluate the risks of using cloud computing in maritime applications. We started the project with a literature study to gain insight into where the research is today; in early stages, we focused on the Maritime Cloud and assumed that it was a cloud computing concept targeted towards maritime applications.

The literature review revealed later that the Maritime Cloud is an architecture to achieve secure information exchange between marine entities, not a cloud computing platform. This discovery left the Maritime Cloud out of the scope for the risk assessment, since we were interested in evaluating the usage of cloud computing. However, the Maritime Cloud architecture can still be relevant since it is a communication architecture which we can integrate with our constructed solution.

We constructed a simple port approach reporting system to serve as the subject for our risk assessment. To make the case realistic, we targeted the application to run in the AWS cloud, which allowed us to sketch up an application with a serverless infrastructure, allowing the developers to focus only on the code, not the provisioning of the underlying infrastructure. An advantage of this approach is that we get real-life cloud characteristics as input to the analysis. However, the application is constructed and has not been used in real life situations, meaning that we were challenged to imagine how it would operate in a marine environment. Experiences from a real-life ICT solution targeted at maritime entities could have been good input to our analysis.

We will in the rest of this chapter discuss the main findings from the literature study and results from the risk assessment. The goal is to obtain the risks of using cloud computing for marine applications. In our risk assessment, we mapped out threats, where we assigned a risk level to each of them.

6.1 Issues from cloud computing

An observation from the risk assessment is that many of the mapped out risks can be applied to computing in general, not only cloud computing. Even if the risks are not unique to cloud computing, the characteristics of sharing resources can make an attack more severe since there is a potential to affect all residents on the infrastructure.

Another observation is that few of the risks mapped out are related to marine environments. Thus, we see that by introducing cloud computing into marine environments, we introduce the risks into the marine ecosystem.

We evaluated lock-in to be the biggest threat against our system, this is a consequence of the design decision where we have tight integration with services only AWS offers – DynamoDB, Lambda, and API Gateway. The three services are proprietary and exclusive to AWS, meaning that we cannot directly migrate our data to competing CSPs.

The lock-in should not be problematic when the services are stable, and the CSP do not drastically change the service or its pricing model. Thus, trusting the CSP is essential when we work with a serverless architecture. A potential worst case scenario is if the CSP shut down the service, without giving the end-user an opportunity to dump the data; even with data dumps available, migration over to another CSP can be a time-consuming process leaving our solution unavailable until we have completed the migration. However, a random shutdown of a service is not likely since it will harm the customer relationships between the CSP and their users. Legal notices, financial matters and malicious intentions are still a part of the threat image, creating a possibility for permanent damage against the CSP.

Data breaches are another prioritised risk. This threat is not exclusively related to cloud computing since it can affect other types of computing equipped such as communication links. But cloud computing introduces new vulnerabilities related to sharing the computing resources; poor resource isolation can in worst case allow an attack against another service to escalate into our service, given that both services reside on the same cloud computing infrastructure. Thus, we see that the advantages of resource efficiency delivered by cloud computing can be a potential vulnerability. The literature review supports the fear of data breaches in the report on the Cloud Computing Top Threats report [4].

Availability is the security property ensuring that our service is available on-demand for our end-users. Denial of service attacks has been sketched out as a threat against out service, and poor resource provisioning is one of the vulnerabilities making this possible. The serverless architecture scales well since the idea is to pay per request. The only service in our stack we have specified capacity for is DynamoDB, which allows us to specify read and write capacity. But there are issues related to provisioning capacity, too little makes resource starvation possible, while too much introduces unnecessary cost. Even with indefinitely scaling, we are faced with the question if we want to pay for the resource usage of handling garbage requests. Maybe we should allow the service at one point to go down to avoid an unnecessary cost.

Our setup at Amazon makes them responsible for handling the cryptography between the end-user and the API Gateways service. The model takes the cryptographic keys away from the development team; this leave us with a question, can we trust AWS to handle out secret keys? We are already trusting AWS to handle all of our data, including the sensitive one. On the other hand, it is reasonable to assume that our service is not the only service trusting keys to AWS, this fact makes AWS an attractive target for cybercriminals since a breach can leak the secrets of many cloud customers.

We can observe during this discussion that traditional information security threats are transferred into the cloud and are not new. But many customers share the same resources on the infrastructure, making cloud computing environments attractive targets.

6.2 Challenges in the maritime environment

We have assumed that a port authority uses the solution with many internal users; also each ship has their set of credentials.

We described the maritime ecosystem as global, and a vessel is reliant on satellite links when communication happens at sea. The satellite links often provide less bandwidth than a regular Internet connection, and our solution design must keep that in mind to avoid obtaining unnecessary capacity. We described denial of service attacks as a threat against our cloud service, but similar effects can occur in overloading network equipment such as the satellite links. However, the crew has an expectation of accessing Internet services at sea, meaning that network owners try to create capacity that meets those demands.

Continuing on the availability of satellite networks, we can characterise the network to have a linear network topology, which is a contrast to the Internet which can provide many redundant paths inside the core network. However, this is not necessary the case, since a ship may communicate with multiple satellites in the network given that there are a no blockades on the path between the ship and the satellite.

We can go further by linking this issue up against cloud computing. Latency is the time it takes for a network package to get from one point to another; a satellite link will have higher latency than a fibre link due to the transmission medium and distance. But the placement of the cloud data centre can also have an impact on latency. While we cannot change the satellite latency, we can plan the deployment of the solution to be near the point where the connection goes from land to the satellite.

If we recall the background on ICT in the maritime industry, Tucci stated that there have not yet been any deliberate cyber attacks against ships. On the same time, our risk analysis pointed that insecure connected control systems may be a vulnerability. Previously the cybernetic systems have been used internally on the ship to control it, but our case should be able to handle electronic reporting directly from the vessels ICT systems without human interaction. To allow electronic reporting, the control systems must be connected to send data over the Internet, which can expose them to actors with malicious intent. So even if there have not been any major incidents, we should not rule it out when more systems are connected to the rest of the world.

6.3 The human aspect

So far, we have discussed the technological aspects of security, but humans are often a target for malicious intent and often end up as the weakest link in the security chain.

Unfortunately, the cyber security awareness within the maritime industry is not sufficient [12], again making crews and staff potential targets for cyber security attacks. Tucci's paper on cyber security within the US Coast Guard points out that the industry has previously shown success in their risk management efforts, this should be expanded into the cyber domain. However, this can be a challenging task due to the complexity of marine organisations. Another challenge is that you can give staff formal training, but relying on that all of the staff will follow the training is a risky assumption.

While we have focused on the crew and port staff, our ecosystem is more complex and includes developers and the CSP. The development team needs to be aware of recent security issues; else they can introduce the vulnerability of poor software quality. Good developers make mistakes, mainly due to the complexity of their work. Another challenge is the domain knowledge, the developers are specialised in their discipline, but they also need to understand the environment their solution will run in, a poor design can realise some of the vulnerabilities we can find in maritime environments. The CSP is also an important actor to achieve secure operation of our solution, but they do also have humans involved in their work making them prone to mistakes. However, the maritime organisations will most likely not be able to influence the work of the public cloud provides, leaving that out of control. However, if there is a private or community cloud model used, influence should be easier, but also introduces new responsibilities on the marine organisations.

A flaw pointed out during the literature review is that the ICT onboard is handled by the marine technical department, which might not have the competence to handle it. Performing maintenance on the cyber systems without proper knowledge can introduce vulnerabilities in both software and hardware. However, it might be necessary that the marine technical department handles the maintenance of the ships since the ICT is just a part of a more complex system. This leads us to the question of how the future technical departments should be organised and which competence should be present.

Summarised, while not technical, humans are important in the cyber security work since they represent a vulnerability which is attractive for malicious intent. Humans are also the masters of technology, meaning that all technological issues can be tracked back to human activity.

Chapter 7 Conclusion

We observe that by using cloud computing as a component in maritime environments, we directly inherit the risks mapped out in risk assessments of general usage of cloud computing. However, the marine aspect also introduces some challenges, especially related to availability. Human-related vulnerabilities are also present, and the lack of cyber security awareness within the sector are an issue.

While not exclusive to cloud computing, data breaches are one of the biggest fears since a cloud computing platform is an attractive target for cybercriminals, due to the potential of a significant breach since multiple customers share the infrastructure. The gains of using cloud computing end up as a vulnerability.

Technology lock-in is another challenge to consider; more CSPs are offering proprietary cloud services where the end-user gets a service without needing to maintain OS and other underlying infrastructure. The most common issue is the costs of moving to a similar service provided by another CSP, a worst case scenario is if the CSP cannot hand out data dumps.

The marine aspect introduced some issues related to availability, where satellite communications represent a linear network topology, creating single points of failures. The continuous automation of the industry is another subject that will have an impact in the future, which needs systems that can handle electronic reporting.

Risk management culture and security awareness go hand in hand, but the marine sector has a lack of cyber security awareness which makes the organisation vulnerable to cyber attacks. By introducing more complex ICT infrastructure like cloud computing, we risk introducing new threats to organisations who are not ready to handle cyber incidents.

By introducing cloud computing to the maritime industry, we are not creating new and unique risks. But we expand the threat picture for the marine organisation, which needs to adopt cloud computing threats and vulnerabilities into their risk management processes. We should consider both technical and human factors since both represent possible attack vectors. However, the people often represent the weakest link; awareness is needed to prepare them for the cyber future which the maritime industry will be involved.

7.1 Future work

Integrating our constructed SaaS solution with the Maritime Cloud is a natural next step. We started this project by looking at the Maritime Cloud but rejected it since it is not related to cloud computing, but rather an architecture to ensure secure communication between marine entities. Our solution is a natural candidate to be integrated into the Maritime Cloud architecture since it tries to give reporting a digital interface before approaching a port.

In this thesis, we constructed a case to act as a subject for a risk analysis. We deployed the created solution into AWS, which served as a real-life cloud computing environment in our analysis. Analysing a SaaS solution in production – targeted for usage by marine entities – is a natural continuation of this project, allowing to use experiences from production in the risk analysis. We can extend the research into a survey where we collect data on experiences.

We pointed out the lack of cyber security awareness as a vulnerability within the maritime industry; this could lay the foundation for research on cyber security awareness. We can use the results to gain an understanding of how the crew and port staff interact with ICT systems, exposing possible vulnerabilities.

References

[1] The European Union Agency for Network and Information Security. Analysis of Cyber Security Aspects in the Maritime Sector, 2011. https://www.enisa.europa.eu/publications/cyber-security-aspects-in-the-maritime-sector-1/at_download/fullReport [Accessed: 18-May-2017].

[2] Amazon Web Services, Inc. Overview of Amazon Web Services?, 2015. https://d0.awsstatic.com/whitepapers/aws-overview.pdf [Accessed: 22-Mar-2017].

[3] H. Bennasar, A. Bendahmane, and M. Essaaidi. An overview of the state-of-the-art of cloud computing cyber-security. *Lecture Notes in Computer Science*, 10194: 56–67, 2017.

[4] CSA. The Treacherous 12: Cloud Computing Top Threats in 2016. Technical report, Cloud Security Alliance, 2016. https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf [Accessed: 11-May-2017].

[5] K. Dellios and D. Papanikas. Deploying a maritime cloud. *IT Professional*, 16: 56–61, 2014.

[6] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio. Security issues in cloud environments: a survey. *International Journal of Information Security*, 13(2):113–170, 4 2014.

[7] ISO/IEC. Information technology – Security techniques – Information security risk management. Standard, International Organization for Standardization, Geneva, CH, 2011. ISO/IEC 27005:2011(E).

[8] L. Jensen. Challenges in maritime cyber-resilience. *Technology Innovation Management Review*, 5(4):35–39, 2015.

[9] J. Joszczuk–Januszewska. Importance of cloud-based maritime fleet management software. *Communications in Computer and Information Science*, 395:450–458, 2013.

[10] L. M. Kaufman. Data security in the world of cloud computing. *IEEE Security & Privacy*, 7(4):61–64, 2009.

[11] K. T. Kearney and F. Torelli. The SLA Model. In *Service Level Agreements for Cloud Computing*, pages 43–67. Springer, 2011.

[12] Lysneutvalget. Digitale sårbarheter maritim sektor. Technical report, DNV GL, Stavanger, Norway, 2015.

[13] P. Mell and T. Grance. The NIST Definition of Cloud Computing. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2011. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf [Accessed: 24-Mar-2017].

[14] B. J. Oates. *Researching Information Systems and Computing*. SAGE Publications Ltd, London, UK, 1 edition, 2006. ISBN 1-4129-0223-1.

[15] D. Polemi, T. Ntouskas, E. Georgakakis, C. Douligeris, M. Theoharidou, and D. Gritzalis. S-port: Collaborative security management of port information systemsk. In *Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on*. IEEE, 2013.

[16] S. Sehgal, Naresh K.and Sohoni, Y. Xiong, D. Fritz, W. Mulia, and J. M. Acken. A cross section of the issues and research activities related to both information security and cloud computing. *IETE Technical Review*, 28(4):279–291, 2011.

[17] Serverless Team. Serverless documentation. Website. https://serverless.com/framework/docs/ [Accessed: 10-June-2017].

[18] The OWASP Foundation. OWASP Top 10, 2013. https://www.owasp.org/images/f/f8/OWASP_Top_10_-_2013.pdf [Accessed: 29-May-2017].

[19] A. E. Tucci. Cyber risks in the marine transportation system. *Protecting Critical Infrastructure*, 3:113–131, 2016.

[20] A. Weintrit. E-navigation revolution – maritime cloud concept. *Communications in Computer and Information Science*, 471:80–90, 2014.

Appendix A
Literature review statistics

We wish to obtain statistics from our literature search, showing how we obtained the papers for the master thesis.

We have used two sources in our literature search, *Google Scholar* and *NTNU Universitetsbiblioteket Oria*. For each of the search strings, we record the number of results from each of the search engines.

Search string	**# Oria**	**# Scholar**	**Selected**
cloud cyber security	4 166	51 500	[10, 3]
cloud information security	37 656	1 080 000	[6, 16]
marine cloud	35 097	608 000	N/A
marine cyber security	1 369	25 600	[19]
marine cyber risk	1 165	22 300	[8]
marine information security	36 517	1 080 000	N/A
maritime cloud	19 288	148 000	[5, 20, 9]
maritime cyber security	1 506	20 200	N/A
maritime cyber risk	1 075	15 200	[15]
maritime information security	24 479	489 000	N/A

The results from Oria were limited by only letting articles from peer-reviewed journals pass through.

The technical reports *Analysis of Cyber Security Aspects in the Maritime Sector* [1], *Digitale sårbarheter maritim sektor* [12], *The NIST Definition of Cloud Computing* [13], *OWASP Top 10* [18], and *The Treacherous 12* [4] are used in the literature study, but found outside the literature search. All papers are public reports recognised by the industry.

Appendix B
Service API documentation

Vessels – List all vessels

```
GET /vessels
```

Success 200

Field	Type	Description
`vessels`	Object[]	List of vessels.
`name`	String	Name of the vessel.
`home_port`	Object	Home port information.
`city`	String	Home port name.
`country`	String	Home port country.
`created_at`	Number	Timestamp of creation.
`updated_at`	Number	Timestamp of last update.

Error 5xx

Name	Description
`InternalServerError`	Couldn’t fetch vessel list

Vessels - Create new vessel

`POST /vessels`

Parameter

Field	Type	Description
`name`	String	Name of the vessel.
`home_port`	Object	Home port information.
`city`	String	Home port name.
`country`	String	Home port country.

Success 200

Field	Type	Description
`name`	String	Name of the vessel.
`home_port`	Object	Home port information.
`city`	String	Home port name.
`country`	String	Home port country.
`created_at`	Number	Timestamp of creation.
`updated_at`	Number	Timestamp of last update.

Error 4xx

Name	Description
`ValidationError`	Couldn’t create the vessel

Error 5xx

Name	Description
`InternalServerError`	Couldn’t create the vessel

Vessels – Request vessel information

```
GET /vessels/:name
```

Parameter

Field	Type	Description
`name`	String	Vessel's name.

Success 200

Field	Type	Description
`name`	String	Name of the vessel.
`home_port`	Object	Home port information.
`city`	String	Home port name.
`country`	String	Home port country.
`journeys`	Object[]	List of journeys.
`estimated_arrival_time`	Number	Estimated arrival time.
`last_port`	Object	Last port information.
`city`	String	Last port name.
`country`	String	Last port country.
`next_port`	Object	Next port information.
`city`	String	Next port name.
`country`	String	Next port country.
`status`	Number	Journey status.
`created_at`	Number	Timestamp of creation.
`updated_at`	Number	Timestamp of last update.

Error 4xx

Name	Description
`NotFoundError`	Couldn't find the vessel

Error 5xx

Name	Description
`InternalServerError`	Couldn't fetch the vessel

Vessels – Update vessel information

`PUT /vessels/:name`

Parameter

Field	Type	Description
`name`	String	Vessel's name.
`home_port`	Object	Home port information.
`city`	String	Home port name.
`country`	String	Home port country.

Success Response

```
HTTP/1.1 204 No Content
```

Error 5xx

Name	Description
`InternalServerError`	Couldn't update the vessel

Journeys – List all journeys

`GET /journeys`

Success 200

Field	Type	Description
`journeys`	Object[]	List of vessels.
`vessel_name`	String	Vessel name.
`estimated_arrival_time`	Number	Estimated arrival time.
`last_port`	Object	Last port information.
`city`	String	Last port name.
`country`	String	Last port country.
`next_port`	Object	Next port information.
`city`	String	Next port name.
`country`	String	Next port country.
`status`	Number	Journey status.
`actual_arrival_time`	Number	Actual arrival time (Optional)
`actual_departure_time`	Number	Actual departure time (Optional)
`created_at`	Number	Timestamp of creation.
`updated_at`	Number	Timestamp of last update.

Error 5xx

Name	Description
`InternalServerError`	Couldn’t fetch journey list

Journeys – Create new journey

`POST /journeys`

Parameter

Field	Type	Description
`vessel_name`	String	Vessel name.
`estimated_arrival_time`	String	Estimated arrival time.
`last_port`	Object	Last port information.
`city`	String	Last port name.
`country`	String	Last port country.
`next_port`	Object	Next port information.
`city`	String	Next port name.
`country`	String	Next port country.

Success 200

Field	Type	Description
`vessel_name`	String	Vessel name.
`estimated_arrival_time`	Number	Estimated arrival time.
`last_port`	Object	Last port information.
`city`	String	Last port name.
`country`	String	Last port country.
`next_port`	Object	Next port information.
`city`	String	Next port name.
`country`	String	Next port country.
`status`	Number	Journey status.
`created_at`	Number	Timestamp of creation.
`updated_at`	Number	Timestamp of last update.

Error 4xx

Name	Description
`ValidationError`	Couldn’t create the journey

Error 5xx

Name	Description
`InternalServerError`	Couldn't create the journey

Journeys – Request journey information

```
GET /journeys/:vessel_name/:expected_time
```

Parameter

Field	Type	Description
`vessel_name`	String	Vessel's name.
`expected_time`	Number	Estimated arrival time.

Success 200

Field	Type	Description
`estimated_arrival_time`	Number	Estimated arrival time.
`vessel`	Object	Vessel name.
`name`	String	Home port information.
`home_port`	Object	Home port information.
`city`	String	Home port name.
`country`	String	Home port country.
`last_port`	Object	Last port information.
`city`	String	Last port name.
`country`	String	Last port country.
`next_port`	Object	Next port information.
`city`	String	Next port name.
`country`	String	Next port country.
`status`	Number	Journey status.
`actual_arrival_time`	Number	Actual arrival time (Optional)
`actual_departure_time`	Number	Actual departure time (Optional)
`created_at`	Number	Timestamp of creation.
`updated_at`	Number	Timestamp of last update.

Error 4xx

Name	Description
`NotFoundError`	Couldn't find the journey
`NotFoundError`	Journey's vessel doesn't exist

Error 5xx

Name	Description
`InternalServerError`	Couldn't fetch the journey

Journeys – Update journey information

```
PUT /journeys/:vessel_name/:expected_time
```

Parameter

Field	Type	Description
`vessel_name`	String	Vessel's name.
`expected_time`	Number	Estimated arrival time.
`status`	Number	Journey status.
`actual_arrival_time`	String	Actual arrival time (`new_status=2`)
`actual_departure_time`	String	Actual departure time (`new_status=3`)

Success Response

```
HTTP/1.1 204 No Content
```

Error 4xx

Name	Description
`NotFoundError`	Couldn't fetch the journey
`ValidationError`	Couldn't update the journey

Error 5xx

Name	Description
`InternalServerError`	Couldn't fetch the journey
`InternalServerError`	Couldn't update the journey

Appendix C
Structure of SaaS solution

The case study concerns a constructed SaaS solution deployed on AWS; this appendix will give a short overview of how the project is laid out and what the different files do.

We use a framework called Serverless to orchestrate the project; this framework allows us to build event-driven functions with a pay-per-execution model [17], which fits into the serverless category.

The following files are present in the project:

api/journeys/create.js Function to create a journey.

api/journeys/list.js Function to list all journeys.

api/journeys/show.js Function to show a specific journey with vessel.

api/journeys/update.js Function to update a journey with status and times.

api/vessels/create.js Function to create a vessel.

api/vessels/list.js Function to list all vessels.

api/vessels/show.js Function to show a specific vessel with journeys.

api/vessels/update.js Function to update a vessel.

common/errors.js Library of error responses used by the API functions.

serverless.yml Configuration script for serverless, used to specify how functions are triggered and resource usage.