CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Reference Architecture for Control Tower to Operate and Monitor Autonomous Heavy-duty Vehicles

Master's thesis in Computer science and engineering

MOHAMMAD JASIM UDDIN

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Reference Architecture for Control Tower to Operate and Monitor Autonomous Heavy-duty Vehicles

MOHAMMAD JASIM UDDIN

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

Reference Architecture for Control Tower to Operate and Monitor Autonomous Heavy-duty Vehicles

MOHAMMAD JASIM UDDIN

Supervisor: Christian Berger, Department of Computer Science and Engineering
Advisor: Mohammad Manjurul Islam, Volvo Autonomous Solutions
Examiner: Robert Feldt, Department of Computer Science and Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Reference Architecture for Control Tower to Operate and Monitor Autonomous Heavy-duty Vehicles

MOHAMMAD JASIM UDDIN
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

[**Context**] In recent years, automated vehicle technology has been rapidly evolved and enabled vehicles to perform a rich variety of autonomous tasks. This progress in vehicle automation has begun to indicate the possible benefits that automated vehicles can have on society and by scaling up the deployment of automated vehicles, the transportation networks will become increasingly more safe and efficient. In the maturity of certain autonomous functionalities, it has also become clear that automated vehicles will not and should not do certain tasks on their own. A cloud-based off-board system a.k.a control tower to operate and monitor the autonomous vehicle could help to scale up autonomous vehicle operation and improve productivity and efficiency.

[**Problem**] While the role of a cloud-based control tower is obvious, it needs to be investigated what are the mandatory functionalities and how the architecture of a control tower should look like. Along with the functional requirement, it is also required to investigate non-functional requirements and critical quality attributes of the control tower system.

[**Results**] Based on the investigation in an industry setup this study defines a reference architecture by identifying functional and non-functional requirements. This study identifies fifteen mandatory services for an operational control tower in a confined area. From the literature it is also found that *Performance*, *Reliability* and *Security* are most important quality attributes whereas from the stakeholder feedback it is found that *Security*, *Reliability* and *Maintainability* are the most important quality attributes. It is also observed that there is a lack of using common vocabularies among the stakeholders and in the literature, there is diversity in using and selecting the vocabularies while discussing control tower functions for autonomous vehicles.

[**Contribution**] The major contributions are the identified functional and non-functional requirements and designed architectural artifacts that aggregate knowledge from domain experts and literature. These artifacts can serve as a reference point for autonomous vehicles company to develop their own control tower for autonomous vehicles.

Keywords: autonomous vehicle, cloud, off-board system, control tower, functional requirement, non-function requirements, quality attributes, reference architecture, Architecture Trade-off Analysis Method.

Acknowledgements

I am grateful for the opportunity to write my master thesis within a highly relevant and interesting topic at Volvo Autonmous Solutions. I would like to thank my industrial supervisor, Mohammad Manjurul Islam for constructive discussions and continuous assistance throughout the study.

Last, but definitely not least, I would like to thank my academic supervisor, Christian Berger for his continuous feedback and supports.

Mohammad Jasim Uddin, Gothenburg, November 2021

Contents

List of Figures

List of Tables

1

Introduction

Autonomous Vehicles (AVs) have shown enormous potential to provide economic, societal, and environmental benefits [11]. Studies predict that with a 50% penetration, autonomous vehicles will result in 9,600 lives saved per year, 1.9 million fewer crashes, $50 billion in economic savings, 1.6 billion hours saved through less time traveled, and 224 million less gallons of fuel consumed [12]. The primary rationale behind these predictions is the AVs potentials in enhancing road capacity, improved road safety, increased transportation efficiency, and decreased traffic congestion and fuel consumption [13]. So far, the technological advancement of autonomous vehicles is dominated by the passenger vehicle manufacturer as in the case with Waymo [14], the commercial vehicle industries have also witnessed the need for complete automation. In fact, the need is much stronger as the economic case for autonomous technology is much higher for commercial vehicles as they could make the entire fleet operation efficient and productive. However, like autonomous cars, this also leads to the concern of safety and ethics during drive especially during boundary cases, where typically the human decision-maker would exercise some individual moral judgment [15].

The functionality of an autonomous vehicle is based on a complex Automated Driving System (ADS). Perception, plan & communication, and control are the three main functions of ADS [16, 4]. From the perception of the surrounding environment and predefined control, autonomous vehicles take decisions and execute missions. But when real situations dynamically interact, for example, a bad weather situation, it becomes a necessity to have a central point to monitor and coordinate the activities of the vehicle by giving extra input to vehicles beyond what their sensors can perceive. Because provisioning on-board human driver or a fallback action like an emergency brake is inefficient, costly, and to some extent unsafe. Additionally, it is unrealistic for the expected commercial mobility services with fleets of shared vehicles [17]. Hence on a larger futuristic scale, when providing dynamic fleet based autonomous solutions, it is paramount to have a control tower that allows the service provider to switch between the different transportation modes [18] within the different periods of the day according to the dynamic transportation demands of the users [19]and also to plan and optimize missions and control the fleet.

It is inspiring to learn from the idea of having control towers in air traffic control and management. The aviation industry has fully embraced automation in flight control and navigation systems since the mid-1970s (EBN 2016). The control tower can act as a decision-maker and can also be a decision support system for automated vehicles in dynamic driving scenarios [20]. It can also improve efficiency and increase revenue by implementing better and customize fleet management. For example, an autonomous vehicle manufacturer can deploy autonomous vehicles in a confined area with predefined missions like in mining. From a control tower, one person can manage multiple automated vehicles, take actions upon request, and take over the control of an autonomous vehicle based on an incident from autonomous drive to manually driven by the operator in teleoperation mode. One vital role of the control tower is assuring traffic safety and increase traffic efficiency. Other roles could be coordinating among the fleets, infrastructures, service providers, and traditional road users.

When it comes to the point to have a primary functional control tower for autonomous vehicles there is no industry standard or architectural guidelines or a Reference Architecture (RA). Most major Truck and passenger car OEMs across the world (Volvo, Daimler, BMW, Audi, Ford, Nissan, Volkswagen ...) have active development for autonomous driving solutions [21]. Although there is a great initiative from Drive Sweden [5] to bring the different automotive companies, service providers, and research institutes to work together to develop control tower for AVs, most of them are still working in an isolated manner that may lead to duplicated work and difficulties in interoperability. Even this phenomenon could be present among the different teams of a particular OEM for different projects as there is no guideline about how to address common problems. So having a RA will help the OEMs to create a more stable and robust product line, reduce development time and cost.

It can be argued why competing companies and organizations would like to develop a RA. They will not be interested in investing on developing a RA without optimal business opportunity as developing a RA is a complex job, needs initial investment, and there will be a learning curve down the road to adopt and maintain the RA [22]. To clarify that argument, the success story of AUTOSAR [1] (AUTomotive Open System ARchitecture), a well-known, mature, and accepted software RA for automotive applications used worldwide by more than 180 organizations [22] cab be taken. A survey study conducted by Martínez-Fernández et al. [22] showed that by adopting AUTOSAR, the automotive industry improved standardization by 88% and reusability by 80% which in turn helped them to have increased productivity and quality. So it is visible that the business opportunity is much higher than the business threat by developing a RA for AVs control tower which will help them to scale their services and generate better revenue.

The aim of this master's thesis is to apply the knowledge gained on software architecture in designing a reference architecture of a control tower for the autonomous commercial vehicles of Level 4 autonomy (Automated Driving – Levels of Driving Automation are Defined in New SAE International Standard J3016, 2014). The

[1] https://www.autosar.org/

contribution of this research can be useful for companies like Volvo Autonomous Solutions (VAS) to build the control tower for their autonomous fleet management. It can also be a starting point or point of reference for the other companies that want to develop control towers for their autonomous vehicles. Control tower for autonomous vehicles is a new research area and this thesis can be a contribution to the autonomous sector and help other researchers and developers in exploring and improving solutions and services for autonomous vehicles.

1.1 Problem Domain & Motivation

While the role of a control tower for autonomous vehicles is somehow evident, but the development of a control tower is still a difficult task. This is due to a lack of guidelines about how to apply specific patterns and/or practices to design and develop a control tower. The control tower needs to communicate with the systems which are complex by nature and which are developed by the different teams with a different mindset. The lack of architectural guidelines and common vocabulary hinders development, reduces effective communication among teams, and increases production cost. This also leads to ad-hoc design decisions while addressing non-functional requirements thus increase technical debt.

In general, the AVs control tower is subject to safety and security-critical attributes but is also necessary to identify other non-functional requirements like reliability, availability, fault-tolerance, maintainability, and scalability. Then a RA combining various quality requirements through appropriate design decisions can support the development of a control tower for AVs. The RA will do so by encompassing the knowledge about how to develop concrete architecture by 1. capturing the essence of the software architecture of this domain[23], 2. serving as standardization and evaluation of control tower system [24],3. avoiding the reinvention or re-validation of solutions to problems already solved [10], and 4. reducing costs of maintenance and development of software applications [25]. In the collaboration with VAS, this thesis work is intended to develop such a RA that would help to reap the aforementioned benefits.

1.2 Research Goal & Research Questions

The purpose of the study is to develop a reference architecture for autonomous commercial vehicle control towers by identifying functional and non-functional quality requirements. The intended architecture will be based on the inputs from domain experts on both functional and quality requirements and challenges to achieve those requirements.

RQ 1: What are the mandatory functional requirements for a control tower to operate and monitor autonomous heavy-duty vehicles in a confined area?

RQ 2: What are the non-functional quality requirements for a control tower in light of ISO/IEC 25010:2011 standard and how to handle them in the reference *architecture?*

RQ 3: How should a reference architecture for a control tower look like?

1.3 Contribution

This study contributes to the existing body of knowledge by aggregating information from domain experts and current literature. It can serve as a reference point for AVs company to develop their own control tower for autonomous fleet management which could help them to reduce the time to market and production cost. It can also serve as a contribution in the software engineering research on autonomous vehicle control tower research and software architecture as a whole.

1.4 Scope

The purpose of the study is to identify the functional and non-functional requirements and present a reference architecture. The scope of the eliciting functional requirement is limited to autonomous heavy-duty vehicles operating in a confined area. The proposed reference architecture is based on identified functional requirements. But it also aims to find out how the non-functional quality requirements can be addressed and validated in the reference architecture.

1.5 Structure of the Thesis

The structure of the rest of the document is as follows: Chapter 2 presents the related work for control tower reference architecture and how to design a reference architecture in general. Then in the chapter 3, the adopted methods for the study are explained. Results from the study are presented in chapter 4 and in the chapter 5 findings are analyzed and discussed and then reports the threats to validity. Finally, the document draws the conclusion and suggests a future outlook in the chapter 6.

2

Background and Related Work

This chapter aims to give some background about the study and discuss related work that exists in the literature. The research of related work is helpful to motivate the design choice of this study, for example how to start the process of designing a reference architecture. One of the main focuses of the related work, research is to find out the proven ways to design and evaluation of reference architecture in general. To find out functional and non-functional requirements for control tower development, the autonomous vehicle or similar fields are in focus. The searching for the related work closely following guidelines provided by Kitchenham [26]. The background related to *Reference Architecture* and *Architecture Tradeoff Analysis Method (ATAM)* intends to give some theoretical background for the reader.

2.1 Reference Architecture

According to IEEE, “Architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.” [IEEE Std.610.12, I.S. Board. 1990]. On the other hand, Reference Architectures (RA) is a special type of software architecture that systematically reuses architectural knowledge [27]. To Kruchten, [28], “RA is, in essence, a predefined architectural pattern, or set of patterns, possibly partially or completely instantiated, designed and proven for use, in particular, business and technical contexts, together with supporting artifacts to enable their use. Often, these artifacts are harvested from previous projects”.

Reference architectures have emerged as a special type of software architecture that achieves a well-recognized understanding of specific domains, promoting reuse of design expertise and facilitating the development, standardization, and evolution of software systems [2]. Different approaches have been proposed and applied to design the reference architecture of a system in the literature. Galster et al., [1] proposed an approach based on the empirical ground. Their approach consists of six steps performed by the software architect and domain experts. They argue that following these steps helps to design reference architectures either from scratch or based on existing architecture artifacts. In the later part of their study, they showed how to apply their approach to the design of two existing reference architectures found in the literature. The following figure 2.1 illustrates the steps they proposed. In step

3 the term 'Empirical acquisition of data' means collecting requirements from the domain and literature.

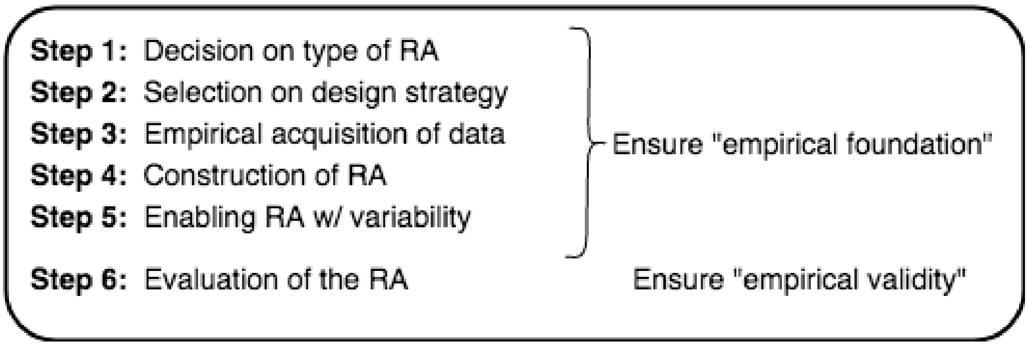

Figure 2.1: Empirically Grounded RA. Adapted from [1]

Nakagawa et al., [2] proposed another approach to design, representation, and evaluation of reference architectures. In their effort, they have designed a framework called 'ProSA-RA' to design, represent and evaluate a reference architecture. Rohling [29] proposed a reference architecture for "Satellite control systems" based on this framework. The framework consists of a four steps process and the first steps in 'Information Source Investigation' which heavily rely on the information from the literature and domain expert. The following figure illustrates the framework.

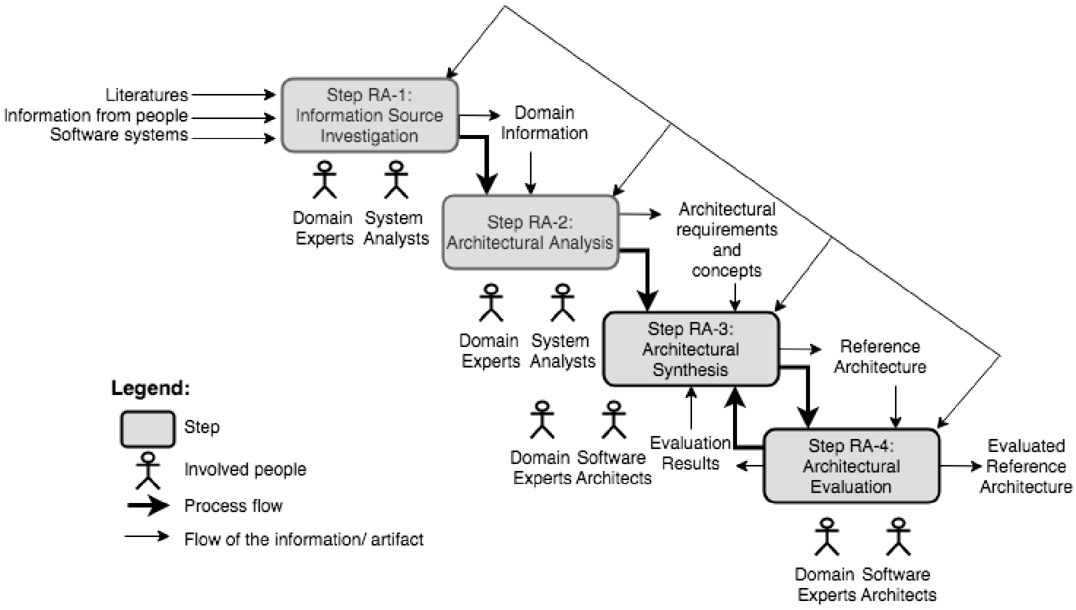

Figure 2.2: Outline Structure of ProSA-RA. Adapted from [2]

2.2 Architecture Tradeoff Analysis Method

Architecture Tradeoff Analysis Method (ATAM) is a technique for analyzing software architectures. The purpose of the ATAM is to assess the consequences of architectural decisions in light of quality attribute requirements [3]. The ATAM gets its name because it not only reveals how well an architecture satisfies particular quality goals (such as availability, and security), but it also provides insight into how those quality goals interact with each other [3]. The following figure 2.3 illustrates ATAM activities. A conceptual flow of ATAM is also provided in the figure 2.4.

2.2.1 ATAM Purpose

The purpose of the ATAM is to assess the consequences of architectural decision alternatives in light of quality attributes [30]. The method ensures the right questions are asked early to discover

- risks: alternatives that might create future problems in some quality attribute
- sensitivity points: alternatives for which a slight change makes a significant difference in a quality attribute
- tradeoffs: decisions affecting more than one quality attribute [3]

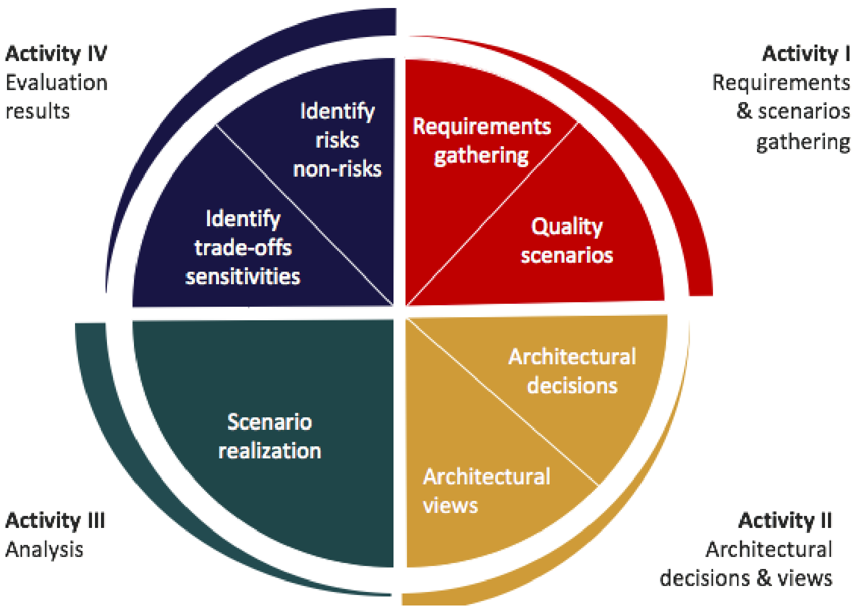

Figure 2.3: ATAM Activities. Adapted from [3]

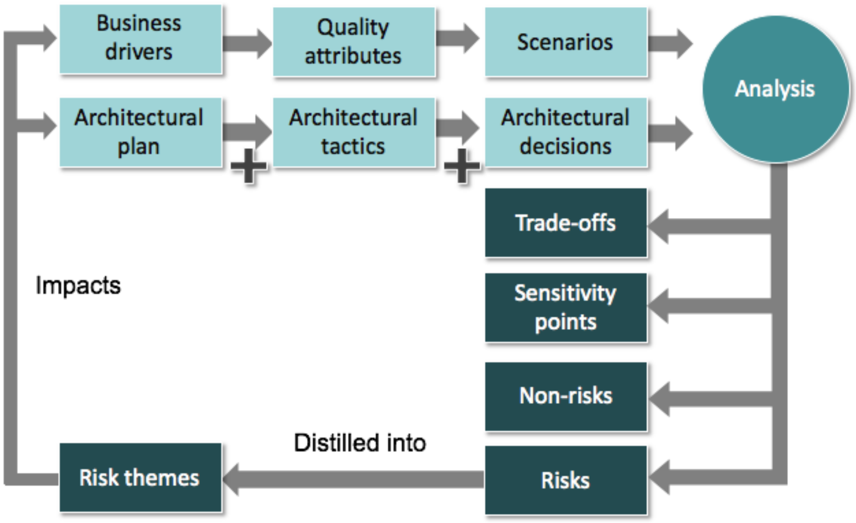

Figure 2.4: ATAM Conceptual Flow. Adapted from [3]

2.2.2 ATAM Steps

The ATAM process consists of the following steps. The steps are adapted from [30].

1. Present the method: a brief overview by the evaluation team of the ATAM steps, techniques used, and outputs from the process.

2. Present the business drivers: a brief presentation by the project manager describing the business drivers and context for the architecture.

3. Present the architecture: architect presents the architecture.

4. Identify architectural styles: architectural styles is discovered as a result of the previous step.

5. Generate the quality attribute utility tree: identification, prioritization, and refinement of the most important quality attribute goals are represented in the form of a utility tree.

6. Elicit and analyze architectural styles: a analysis of the architectural styles in light of the quality attributes in order to identify risks, sensitivity points, and tradeoffs.

7. Generate seed scenarios: a representation of the stakeholder's interest to understand quality attribute requirements.

8. Brainstorm and prioritize scenarios: addition of scenarios from stakeholders and an understanding of their relative importance.

9. Map scenarios onto styles: continuing to identify risks, sensitivity points, and tradeoffs while noting styles and components within styles that are affected

by each scenario.

10. Present out-brief and/or write report: recapitulation of the execution of the ATAM steps, results, and recommendations.

2.3 Autonomous Vehicles

An autonomous vehicle (AV) is one that is capable to operate itself and capable of complete the assigned mission without or limited human intervention. The core competencies of an AV can be categorized into three categories: *perception, planning and control* [4]. *Perception* is the ability to collect information and extract required knowledge from the environment to operate safely. Perception is be divided into two subcategories. The first one is *Environmental Perception* which is responsible for developing a contextual understanding of the environment such as where are the obstacles are located, detection of path tracking (e.g. road sign or marking), and categorizing the extracted data by their semantic meaning. The last one is *Localization* which refers to the ability of the vehicles to determine their position with respect to the environment. *Planning* is the process of making purposeful decisions to achieve the vehicle's higher-order goals. An example of a goal is to bring the vehicle from a start location to a target location while avoiding obstacles and optimizing over-designed heuristics[4]. Finally, the *Control* competency is the vehicle's ability to execute the planned missions that have been created by the higher-level processes. The following figure 2.5 illustrates a system overview of a typical AV.

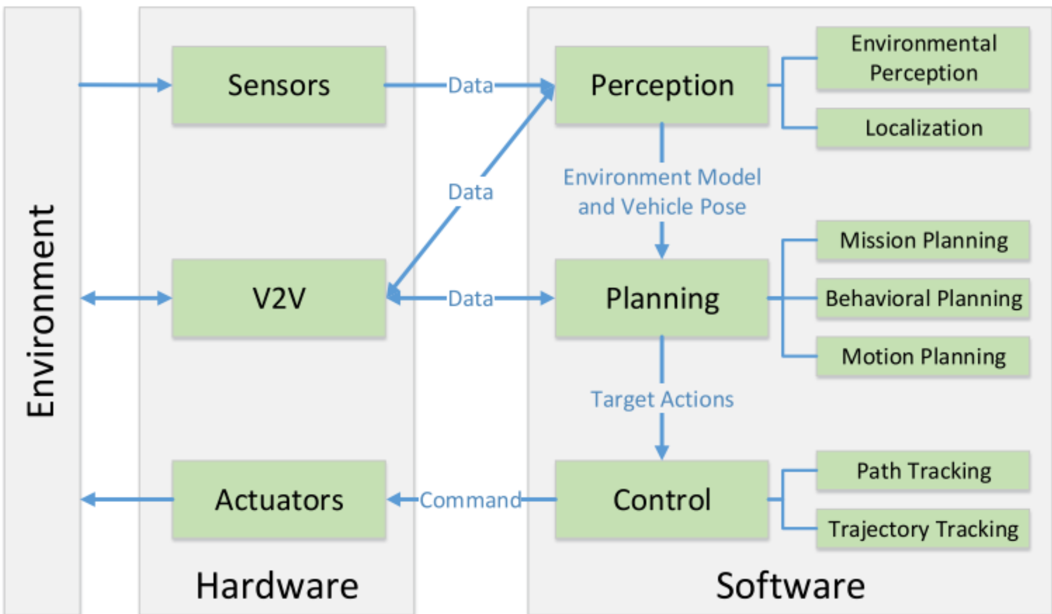

Figure 2.5: System overview of a AV. Adapted from [4]

2.4 Control tower for autonomous vehicles

A control tower or a remote station is relatively known in the field of aviation, marine, and railways. Especially the aviation sector that has started to leveraging the control tower to ensure safe operation and effective management of air traffic. There are several airports across Europe that are now delegating *Aviation Traffic Control* to a remote station. Since 2018, the international airport of Saarbrücken is remotely controlled from the remote tower center in Leipzig, Germany [1]. Norway has recently set up the world's largest Remote Towers Centre [2]. Here in Sweden Luftfartsverket and SAAB have developed a remote tower service with Örnsköldsvik Airport being the first in the world to have this service [3].

In the railways sector, the term control tower is also known. The primary target of a control tower in railways is to reduce travel delays, improved timeliness, and efficient use of railway infrastructure [31]. In marine, autonomous submersibles are controlled remotely which helps to explore the wide difficult range of operations such as deep-ocean and under-ice exploration, tasks in hazardous areas, in natural or man-made disastrous regions, automated searches, surveillance missions, etc [32]. Controlling a satellite is done from a ground station. Important services, such as telecommunication, global positioning system, weather forecast, earth and space observation, meteorology, resource monitoring, military observation, and many others are provided from a ground station which is a control tower like setup [29].

Examples from the related domain have demonstrated that the autonomous vehicle control tower should be able to operate and monitor and take action upon unexpected behavior. However, the key difference between control towers for autonomous road vehicles and other autonomous vehicles (aviation, marine, railway, etc.) is the intensive road network and complex infrastructure [20]. In a joint effort by the Integrated Transport Research Lab and some industry partner, the Automated Vehicle Traffic Control Tower(AVTCT [4]) project have described the role of a traffic control tower (TCT) for AVs. In their study, gaps, barriers, and potential different scenarios were identified. In an article, possible functionalities and benefits of AVTCT as well as challenges are discussed, which has set the foundation for a conceptual model, simulation, and real application of AVTCT [20]. The following figure 2.6 illustrates the potentials roles of AVTCT.

[1]https://www.youtube.com/watch?v=oPq8CPTktao
[2]https://www.youtube.com/watch?v=VN-MrX$_{p}khc$
[3]https://www.saab.com/products/digital-tower
[4]https://www.drivesweden.net/en/projects-5/avtct

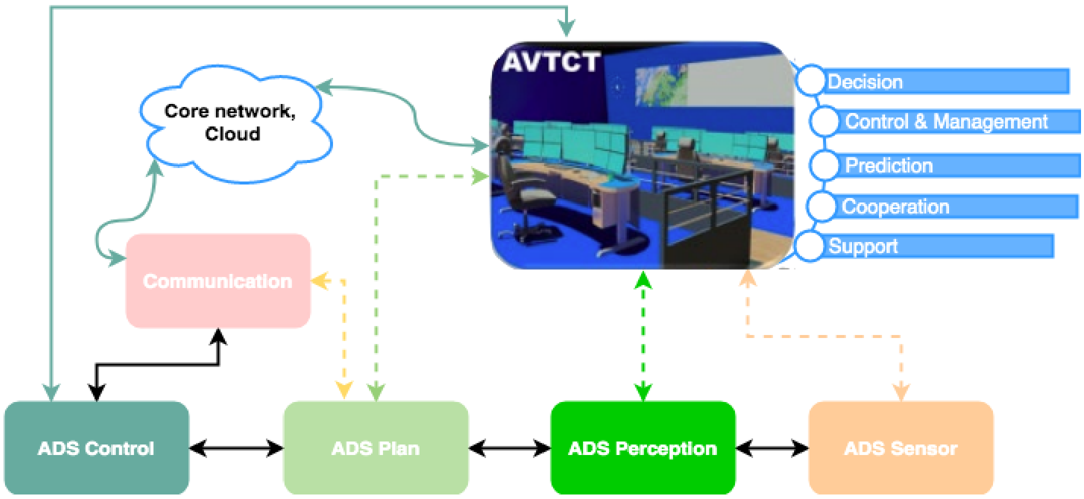

Figure 2.6: Illustration of AVTCT in controlling automated vehicles. Adapted from [5]

According to AVTCT finding the control tower have the potentials to control vehicles when the ADS control fails. It can help to take proactive, reactive, and responsive decisions. It also can provide better control and management and make the whole transport system more efficient and intelligent. Furthermore, it can improve cooperation among stakeholders, provide better prediction situations and improve support.

2.5 Reference architecture for autonomous vehicles

Research in reference architecture on autonomous vehicles support systems is active and diverse. Although most of the researches is not defined as a control tower the motives of the research are to develop a support system for the connected vehicles. So the terms like autonomous driving, cooperative driving, connected vehicles can be considered as control tower systems for autonomous vehicles. S. Behere et al. [6] have proposed a reference architecture for cooperative driving that is focused on needed functions and key architectural elements and their relationships. The architecture fits into and extends the existing vehicle architecture in a minimally invasive manner. It provides a clear definition of needed services and the architectural elements that realize them. There is a good separation of concerns through modularization, enabling the compartmentalization of related data handling and control functions into hardware and software modules. This permits domain experts to focus narrowly on their specific parts. Furthermore, possible errors are isolated and contained within respective architectural modules. The reference architecture can be used to implement various cooperative driving applications like vehicle platooning or conveying. The following figure 2.7 illustrates the conceptual view of the reference architecture they proposed.

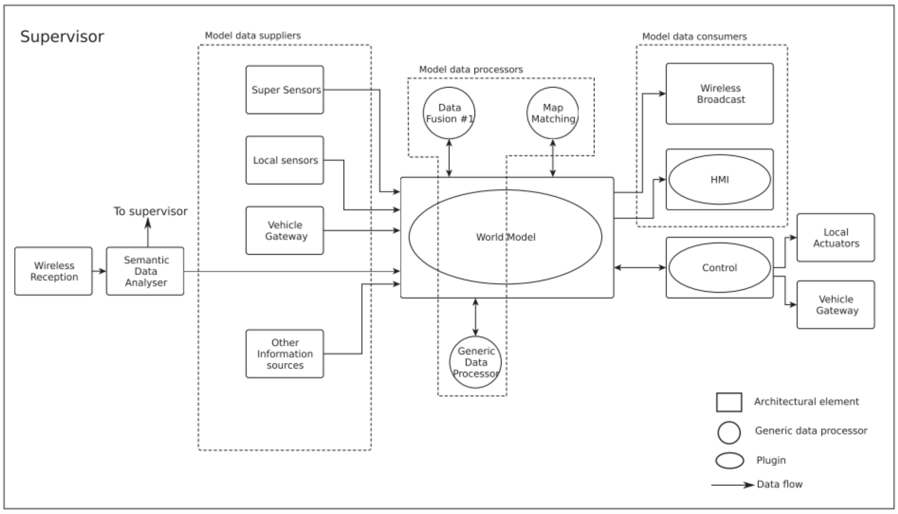

Figure 2.7: Conceptual view of the reference architecture. Adopted from [6]

Schroeder et al. [7] did an industrial case studies that report a systematic process to elicit, integrate and validate functional and non-functional requirement for a multi-domain reference architecture concerned with transport mission planning, execution, and tracking. After identifying the functional and non-functional requirements systematically they have presented the design and evaluation of the reference architecture and given a realization of the architecture in the context of the construction site domain. That work can be considered as the most relevant study to this research work and can be taken as a motivational. The main difference to this study is that they have taken both on-board and off-board systems into consideration.

They have conducted the study in two folds. In the first phase, they have elicited the requirements and design patterns to design the reference architecture and in the end, they did the evaluation of the resulting reference architecture to validate the compliance with previously elicited attributes. The figure 2.8 illustrates the objective and applied methods of their study.

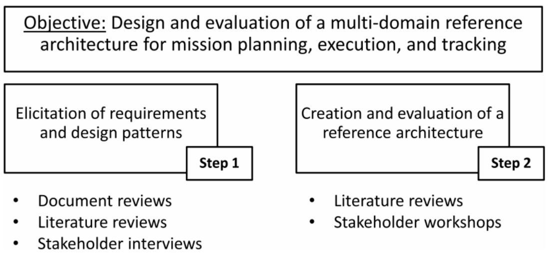

Figure 2.8: Main objective together with the derived research steps and the applied methods. Adapted from [7]

As shown in the figure 2.9 they have presented the functional requirements in the form of a use case diagram for the sake of clarity and confidentiality. The main actors are the mission provider, vehicle environment, and heavy vehicle motion control (HVMC). The mission provider supplies mission input and monitors mission execution, the vehicle environment module is responsible for perceiving the surrounding environment and the HVMC controls the vehicle's motion like acceleration, steering, or braking.

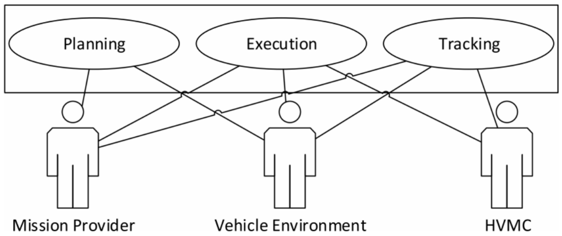

Figure 2.9: Main actors and use cases for the reference architecture. Adapted from [7]

The main use cases they have identified are the plan transport mission and execute the planned missions in a graceful manner so that HVMC reacts flexibly to dynamic changes in the environment. They have also proposed that the reference architecture should be able to aggregate and combined all the data related to mission planning and execution so that it can be used for future improvement.

For the non-functional requirements, they did a thorough literature review along with feedback from industry experts through interviews and workshops. The intention of the literature review was to validate the results in terms of data triangulation with other reference architecture in similar domains. The figure 2.10 illustrates the resulting reference architecture proposed by Schroeder et al. [7].

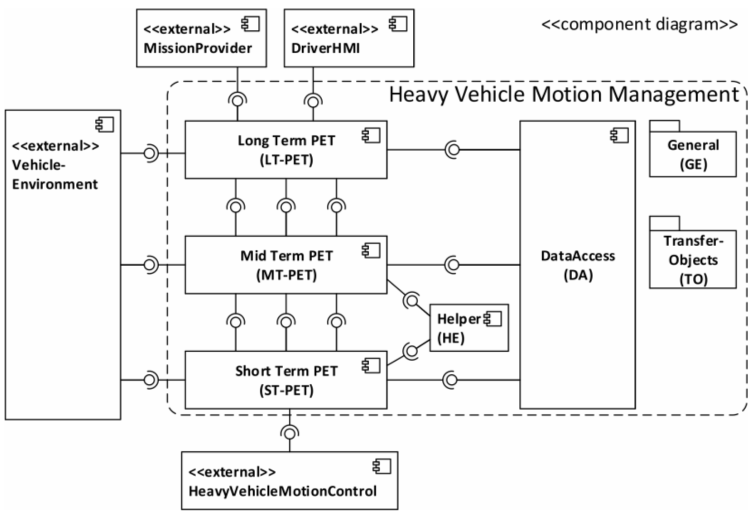

Figure 2.10: The resulting reference architecture proposed by [7]

3

Research Method

This study is divided into three distinct methods: Literature Review, Interviews & Workshops, and Survey. The first two methods are used to find answers to all three research questions whereas the last method 'Survey' is used to evaluate the implemented reference architecture related to research question RQ3.

The study begins with a literature review to explore AVs domain knowledge and software architecture in general. This is done by some ad-hoc searches on Google and reading some pilot papers. This pre-study helped to design the following study methods in a systematic and organized manner. Then the two methods literature review and interview were conducted in almost parallel and analyzed the collected data side by side to see triangulation of data as the triangulation has been viewed as a qualitative research strategy to test validity through the convergence of information from different sources[33].

Data and information extracted through literature review and interview have been analyzed using the thematic analysis process. Interesting pieces of information have been organized into codes and then defined codes are organized into themes to get an overall picture of the domain requirements. This has been done to elicit both functional requirements and non-functional quality requirements. After that, different architectural artifacts (e.g. functional decomposition diagram, deployment diagram) have been generated. In the end, the generated artifacts have been evaluated using ATAM (Architecture Trade-off Analysis Method) and conducted a survey to get feedback from domain experts and software architects. The following sections describe the study methods in detail.

3.1 Literature Review

The search to find out related literature started with searching in Google and Google Scholar. The intention behind this strategy is to gain initial knowledge about the problem domain and find out the necessary keywords. The findings from this phase were used in the literature review study design. Although the literature review conducted in this study is not systematic, but the process closely followed the guidelines provided by Kitchenham [26].

3.1.1 Identification of Research

Identification of relevant studies to answer the research questions is a crucial step in a systematic literature review. In order to do that the search queries need to developed and there are many approaches are available in the literature of systematic review. In this study, an iterative approach is followed for the gradual improvement of the search query. Besides this, some pilot articles have been used to define and improve the search query. Those papers gave a foundation to define keywords for the search query. For example, the keywords *Vehicular Cloud* or *IoT Cloud* are extracted through the pilot study of some literature related to AVs cloud support.

There are two parts to the defined search query. The first part is included to limit the literature within a software architecture. The second part of the query is more related to the domain to include the study that focuses on autonomous vehicles' cloud support functionality. The following table 3.1 illustrates the search query.

Table 3.1: Search query

Search string
("Reference Architecture" OR "Software Architecture") AND ("Vehicular Cloud" OR "Connected Vehicles" OR "Vehicular Network" OR "Control Tower" OR "Autonomous Vehicles Cloud" OR "Automated Vehicles Cloud" OR "IoT Cloud")

The search query were applied to four different scientific databases. The selected databases are Scopus [1], ACM Digital Library [2], IEEE Xplore [3] and Science Direct [4]. The following table 3.2 listed the databases with the number of retrieved papers.

Table 3.2: Selected databases and number of retrieved papers

Database	Filter	Papers
Scopus	Limited to Title/Abstract/Keyword	50
ACM Digital Library	None	87
IEEE Xplore	Metadata	49
Science Direct	Limited to Title/Abstract/Keyword	180

3.1.2 Inclusion Criteria

The studies included for the literature review if they presented a scientific contribution to the body of software architecture knowledge in the context of the autonomous vehicle control tower, connected vehicles, or vehicular cloud. Specifically, the research which focuses mainly to cloud support system for autonomous vehicles by proposing, discussing requirements (RQ1), quality attributes and non-functional

[1] https://www.scopus.com/search/form.uri?display=basic
[2] https://dl.acm.org/
[3] https://ieeexplore.ieee.org/Xplore/home.jsp
[4] https://www.sciencedirect.com/

requirement (RQ2), architecture style and pattern, techniques, and tools (RQ3). The following listing illustrates the detailed inclusion criteria.

1. Studies that addressed the software architectures of autonomous vehicles cloud system or control tower at any level of abstraction, including design patterns, styles, views, scenarios, evaluation methods, quality attributes, etc.

2. Studies that identified procedures and techniques for software architecture management of the cloud-based IoT system.

3. Studies that addressed the reference architecture for a related or neighboring domain like the aviation domain.

Accordingly, the following criteria were considered for exclusion:

1. Studies that were conducted out of Computer Science & Engineering domain.

2. Studies that addressed topics other than software architectures.

3. Articles that were duplicates.

4. Literature other than Conference paper, Article, Review Article, and Research Article.

5. Studies that were not written in English.

3.1.3 Study Selection

The selection process started by reading the title and abstract of all papers whether they should be further investigated. The inclusion criteria were interpreted liberally following the suggested approach by Kitchenham [26] when reading the title and abstract of a paper. Following figure 3.1 illustrates the study selection overview.

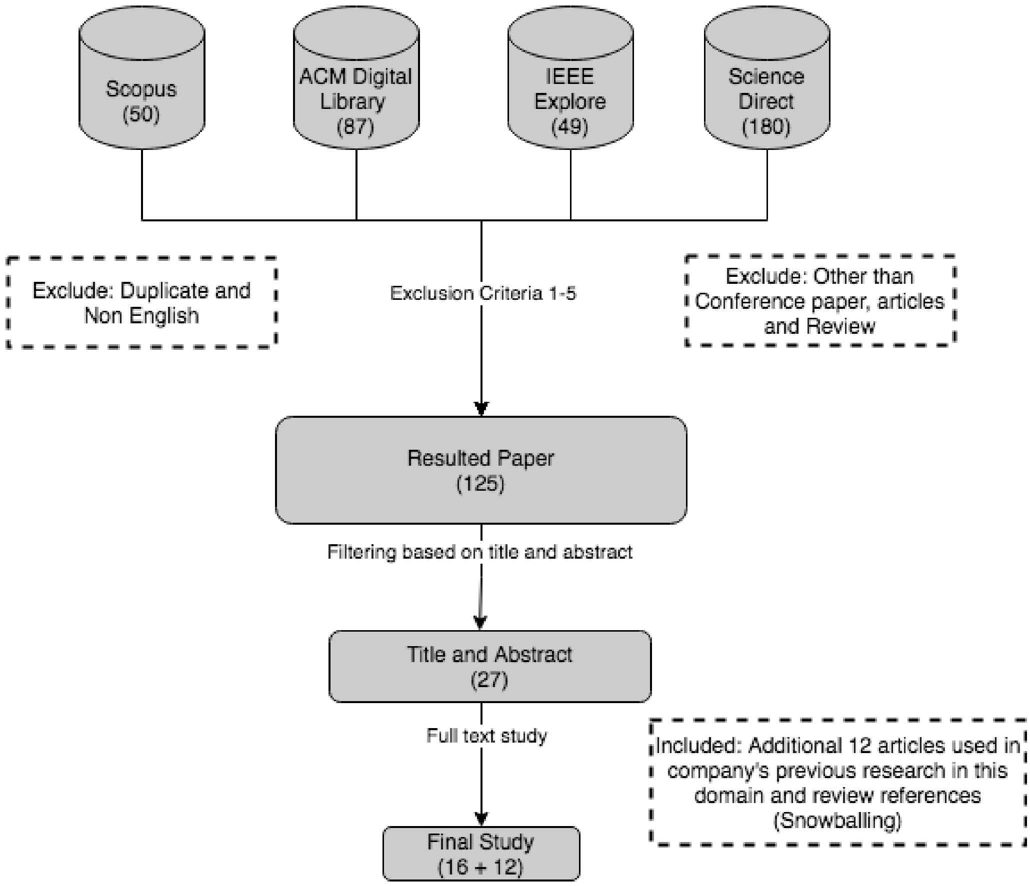

Figure 3.1: Selection of studies

3.1.4 Data Extraction and Synthesis

The primary goal of the literature review was to elicit functional requirements, identifying and prioritizing non-functional requirements for the autonomous vehicle control towers. Other goals of the literature review were to find out the way to define and evaluate a reference architecture. So if a paper provides information or knowledge about all these goals then the paper is taken into further consideration to use as a source of functional requirements and non-functional requirements or a tool to define and evaluate reference architectures. Finally, the studied papers are grouped based on research questions.

3.2 Interviews

Domain knowledge and experiences from the experts on that domain are at the core in reference architectures as it tries to aggregate[23] knowledge on the selected domain. As this study was conducted in an industry setup, continuous discussion and analysis were conducted frequently with domain experts. However, after defining the study scope finalizing the research question, it was essential to investigate the problem domain and find the answers to the search questions systematically with larger participants. Thus the interviews with the domain expert were organized as the interview attempts to understand the world from the subjects' point of view [34] which would be a valuable input in designing a reference architecture.

The entire selection of interviewees followed careful and purposeful sampling which was done by the identification of participants who were knowledgeable or experienced in the area of software development in general and experts in the automotive domain. Necessary research ethics have been followed by selecting participants who were willing to participate, taking consent about sharing their views anonymously as part of data collection. The duration of the interview was around 50 to 60 minutes.

The materials for the interview were prepared systematically. The Goal-Question-Metric (GQM) approach [35] was used to define the objectives of the interview and design interview questions. At the beginning of the interview, the participants were given an introduction about the problem domain and context of the study to avoid any misunderstandings and to stimulate thoughts and discussion from the participants. Careful measurement has been taken to avoid bias. This involved presenting some video clips of a similar control tower solution in the aviation domain and research project (AVTCT [5], NordicWAY [6]).

The following table 3.3 represents the list of the interviewees with their roles. All the interviews were recorded to analyze and extract the data properly as this study was conducted by a single person and it is not practical to collect the data while interviewing.

Table 3.3: List of Participants

Name	Role	Area of Expertise
Interviewee A	Software Architect	Experienced Software Architect with vast knowledge in automotive industry
Interviewee B	Domain architect for AV	Technology advisor and research director
Interviewee C	Domain architect for cloud support	
Interviewee D	Safety & Regulation	Closely work with policy making public agency
Interviewee E	Lead Engineer	Requirement Engineering
Interviewee F	Project manager	
Interviewee G	Research Engineer	Research focus to Control Tower for Autonomous Track
Interviewee H	Consultant for control tower solution	Developer consultant; mainly focus to development

[5] https://vimeo.com/514990723
[6] https://www.youtube.com/watch?v=gTrrl4ymvyc

3.3 Thematic Analysis

Thematic analysis is a way to identifying, analyzing, and reporting patterns (themes) within data in primary qualitative research [36]. According to [37], thematic analysis is a useful method for examining the perspectives of different research participants, highlighting similarities and differences, and generating unanticipated insights. This method has been used to analyze and synthesize data collected from literature reviews and interviews. Following figure 3.2 illustrates the steps taken for thematic analysis.

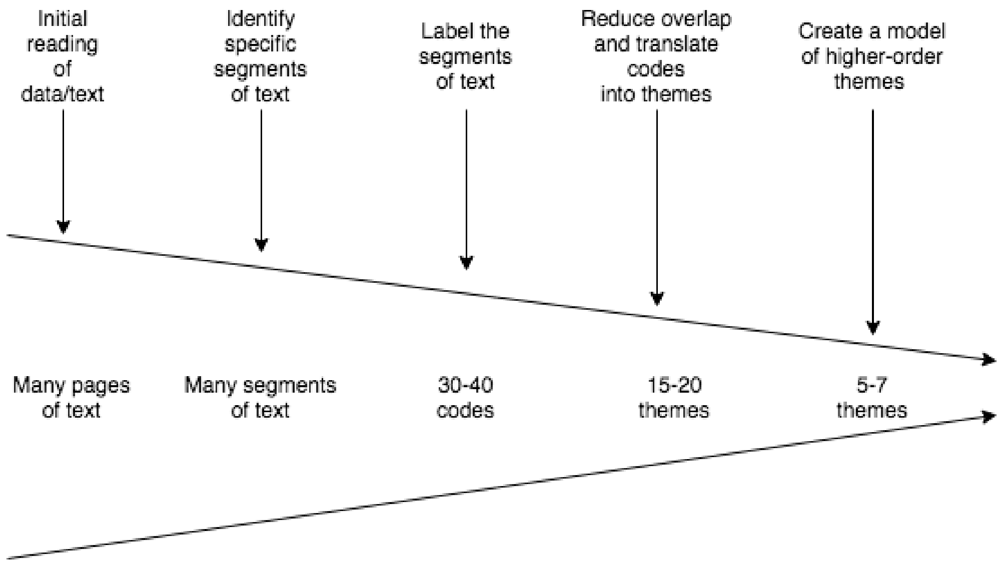

Figure 3.2: Thematic synthesis process (adapted from Creswell [8]).

3.4 Evaluation

The reference architecture can be evaluated according to specific criteria to make sure it fulfills certain quality attributes [38]. However, no quality can be maximized in a system without sacrificing some other quality or qualities, thus there is always a trade-off between the different quality attributes [38]. Two well know techniques propose by Bosch [38] are *Scenario-based evaluation* and *Architectural prototype evaluation.* Another well know technique is Architecture Trade-off Analysis Method (ATAM). The ATAM gets its name because it not only reveals how well an architecture satisfies particular quality goals (such as availability, and security), but it also provides insight into how those quality goals interact with each other [3] and how they trade-off.

The proposed reference architecture is evaluated based on ATAM. The ATAM process can be very extensive and can include 40 to 50 stakeholders. However, in this study, a mini-scale ATAM is conducted with six stakeholders.

4

Results

This chapter describes details findings of the different methods applied to investigate research questions stated in chapter one.

4.1 Functional Requirements - RQ1

The goal of the first research question is to investigate and elicit mandatory functional requirements for a control tower to operate and monitor autonomous heavy-duty vehicles. Research method 'Literature Review' and 'Interview/Workshop' has been applied to dig down the research question. The following sub-sections illustrate the findings.

RQ 1: What are the mandatory functional requirements for a control tower to operate and monitor autonomous heavy-duty vehicles in a confined area?

4.1.1 Literature Review

The literature review to elicit functional requirements for control tower or cloud-based support functions for autonomous vehicles resulted in twelve relevant papers. The resulted papers have different focuses with different levels of detail. Some papers discussed a concept of a cloud function and some papers discussed details of the proposed functions.

In general, it is observed that there is a diversity in using and selecting the vocabularies while discussing cloud support functions for connected vehicles. For example, both Lovas et al. [39] and Olariu et al. [40] discuss a support function to handling traffic lights and monitoring the surrounding area after an accident but they use different vocabulary. While Lovas et al. [39] discusses this use case in proposing PaaS oriented IoT platform for connected cars, Olariu et al. [40] discussed this use case while proposing Autonomous Vehicular Clouds (AVCs). This challenge has been mitigated through thematic analysis of different use cases and other useful information found in the literature. The analysis has been done using the process proposed by [8]. By the thorough analysis of the text, different domain-related codes have been identified. Then with further analysis, the codes are grouped into different domain themes. The following table 4.1 illustrates the domain theme with the number of use cases identified from the literature.

Table 4.1: Functional requirements classification

Domain Theme	Purpose	# of Use Cases
Traffic management	Improving traffic management	7
Fleet management	Managing fleet, optimizing mission, Provide business value	7
Situational Awareness	Aggregate data from different sources to provide better situational awareness	6
Monitoring & Diagnostics	Monitoring vehicle, performing diagnosis activities	5

Details of the identified use cases are given below according to their thematic order.

Theme: Traffic management

Use case: The traffic lights can be synchronized after handling an accident (Olariu et al. [40])
Description: Traffic authority can be notified through the control tower to reschedule the traffic single which will help to reduce the congestion quickly in the afflicted area.

Use case: The control tower can play role to mitigate recurring congestion problem automatically (Olariu et al. [40])
Description: Through the collaboration with the control tower the vehicles in the site can be able to query the mission plan of each other at the time of rerouting during congestion and estimate the impact on the road network to prevent vehicle flooding to a specific segment or lane.

Use case: Efficient use of road network (Maheswaran et al. [41])
Description: With the efficient implementation of the traffic planner algorithm, the road network can be divided into operating vehicles in the site effectively.

Use case: Traffic signals optimization (Whaiduzzaman et al. [42])
Description: Through on-board vehicle navigators, the vehicular cloud (aka control tower) can sense the segment traffic congestion by transferring the GPS coordinates and destination to a navigation server. Then the navigation server implemented in the cloud can take the charge of computing the optimal routes by constructing a traffic load map and the traffic pattern matrix and estimating road segment loads and delays.

Use case: High occupancy vehicle (HOV) lanes management(Olariu et al. [40])
Description: Control tower can play a role to manage HOV dynamically. Utilizing the occupancy sensor data and traffic load, the HOV lane can be configured optimally.

Use case: Evacuation management (Whaiduzzaman et al. [42])
Description: The inductive loops information, cameras feed, roadside sensors data, and surveys data of the individual vehicles can be integrated together with the road network data to form a real-time picture of the traffic situation and plan evacuation accordingly in case of emergency.

Use case: Monitoring of accidents (Lovas et al. [39])
Description: Vehicles sensors can be a source of field data related to accidents. The spatial (map data) and non-spatial (sensor parameters) data from the sensors can be sent to a database center where it gets merged into one database. Then the cloud analyzer analyses the incoming data to achieve various types of statistical reports and provides further insight.

Theme: Fleet management

Use case: Optimizing parking facilities (Olariu et al. [40])
Description: The vehicle can be directed to the most suitable parking location by real-time pooling of the information from the vehicles in the site about the parking space occupancy of the site.

Use case: Managing charging stations (Herrnleben et al. [43])
Description: Charging station management can be optimized through cloud collaboration as the cloud can get battery or fuel level information from the vehicles and estimate the charging lifetime and book a time slot for charging or refueling for the vehicle.

Use case: Maintain a steady speed at controlled RPM on demand (Lovas et al. [39])
Description: The cloud can play role in controlling the RPM as intensive braking and unnecessary speeding both waste fuel and increase safety risks. So it plans a controlled speed profiles for certain segments or lanes in the road network.

Use case: Shift-up early for information on improper shifting up tendencies of the vehicle (Lovas et al. [39])
Description: Malfunctioned shifting up tendencies and habits can be analyzed and fixed later on with the help of the vehicle's RPM and speed information.

Use case: Cloud-based algorithm that generates an optimal speed trajectory to reduce the fuel consumption (Milani and Beidl [44], Al Najada and Mahgoub [45])
Description: Collecting the associated traffic and geographical information (i.e. road network), the cloud generates a route to reach the intended destination. It can also solve the optimization problem by a spatial domain dynamic programming (DP) algorithm that utilizes accurate vehicle and fuel consumption models to determine the optimal speed trajectory along the route.

Use case: Optimizing HD map usage (Liu et al. [46])
Description: The autonomous vehicles collect data by LiDAR and camera. This data then can use to compare with the real-time LiDAR/camera feed which to assists

these vehicles in precisely self-localizing in real-time.

Use case: Remote engine start (Lovas et al. [39])
Description: The cloud can play role in remote engine start which allows to remotely start and run the vehicle for a definable period. It can be handy when it is required to pre-warm the engine for example.

Theme: Situational awareness

Use case: Information sharing through an outside-the-car observer (Maheswaran et al. [41])
Description: The cloud can collaborate and pre-process the data feeds obtained from the vehicles or other road objects before passing them along to other vehicles. For example, pothole information or changing drive-ability conditions during wildfires can be relayed to the other vehicles from the cloud.

Use case: Detecting dead animals and other objects on the road network(Aydin et al. [47])
Description: The cloud support function can notify the cleaning crew and police in real-time where exactly these objects lie which will facilitate them to remove the objects very quickly from the road network.

Use case: Optimized weather forecast (Lovas et al. [39] Marosi et al. [48])
Description: Typical weather forecasting gives very short-term predictions based on recent and localized measurements. However, incorporating the data from the large set of vehicles allows these predictions to be more precise and detailed than traditional weather forecasting.

Use case: Improving traffic sign recognition (Lovas et al. [39])
Description: The existing traffic sign database and the one collected from the vehicle can be used to identify new, changed, missing/stolen traffic signs and update the traffic sign database with the respective information. It can also notify road/site maintenance service authorities about the missing road signs.

Use case: Provide information about obstacles and blind spots on-road (Kumar et al. [49])
Description: Autonomous driving can be significantly improved by providing vehicles greater access to critical information and blind spots about the site or road network from the control tower.

Use case: Monitoring road condition (Lovas et al. [39])
Description: The vertical acceleration data from the vehicle can be used to identify road faults and for assessing the vibration load. This information then combined with collected trajectory and speed data can be used to profile routes according to road quality, goods vibration load, and other potential factors.

Theme: Monitoring and diagnosis

Use case: Check and maintain optimal tire pressure before the drive, after, and at high speeds (Lovas et al. [39])
Description: Consumption of fuel can be optimized by automatic checking of tire pressure frequently.

Use case: Regular vehicle maintenance (Lovas et al. [39])
Description: Control tower can collect data on different parts of the vehicles and analyze and then schedule servicing of the faulty parts.

Use case: Simulating a complex battery lifetime model in the cloud (Adhikaree et al. [50])
Description: Complex battery life cycle can be simulated in the cloud to make it possible to predict the state of health of high voltage batteries.

Use case: Collaborating plausibility checks of vehicles sensor specification and other on-board systems (Milani and Beidl [44])
Description: Compromised electronic control unit (ECU) software, forged sensor specifications, or foreign data can be very harmful to the vehicle's normal operation. The control tower can play role in verifying the plausibility of sensors and ECU software, the models of vehicle software with series data, and initial sensor specification.

Use case: Providing back office support for all the errors, warning, etc. (Marosi et al. [48], Lovas et al. [39])
Description: A could based system can act as a back-office support center and provide the interfaces to the different stakeholders (e.g. users, administrators, etc.) to access error and warning information.

There are different levels of abstractions that can be found in the use cases mentioned in the literature relevant to autonomous vehicle's cloud support functions. That is because different authors defined these use cases in their defined context thus they vary in level of abstraction. This problem is mitigated by domain expert opinion by analyzing requirements mentioned during the interview with experts and continuous discussion with the industry supervisor.

4.1.2 Interview Analysis

Data collected through the interviews have been organized into different Requirement groups and functions under this group. The Following table 4.2 illustrates this. A brief description of all these requirement groups and functions is given after the table.

Table 4.2: Functional group

Requirement Group	Functions
Plan	Operation Flow Integrator, Production Manager, Fleet Manager
Operate & Monitor	Site Manager, Traffic Controller, Mission Planner, Road Network Provider, Situational Awareness, Dynamic Replanner, Monitoring, Remote Operation, Diagnostics & Maintenance, State Maintainer
Evaluate	Productivity Evaluator, Mission Evaluator

In the following section, the identified functions are discussed in detail based on the extracted data from interviews.

Operation Flow Integrator is a component in the cloud that can act as an operation flow integrator which will provide an ecosystem to connect other services like *Work Order Service* and *Billing Service.*

Work Order Service is a service through which a customer can place a work order. For example, a port authority can place a work order to an AVs company to execute loading and unloading in a port area for a certain period.

Billing Service is a service that can be connected to a company's business/enterprise system to provide data about vehicles in operation so that the business system can generate bills to the customer.

Production Manager is responsible overall planning of the production. It requires a certain amount of vehicles to transport goods from point A to point B in given time constraints. This information moves to the *Fleet Manager* that allocates a set of vehicles to accomplish this task. After allocating the vehicles by the fleet manager the production planning is sent to *Site manager* to plan and execute the mission.

Fleet Manager is responsible for managing the fleet of vehicles on a site. It allocates vehicles to the site, maintains the vehicles operating on the site.

Site Manager is responsible operate and monitor the autonomous vehicles on a site. It gets inputs from the *Production Manager* about missions to execute and with the support of other services like *Mission Planner, Situational Awareness* accomplish the mission. It is also responsible for executing the given mission effectively. With help of other services, it will have the capability to re-route or reschedule in case of non-predicted or emergency situations.

Mission Planner primarily concern to plan a complete set of missions with one or more predefined goals within set boundaries. For example, a company has a fleet of trucks and a given amount of goods that should reach its destination in the defined time.

Traffic Controller component provides the services so that several vehicles with different tasks interact and share resources. It can also support to prevents emergency stops and queues when more and more varied tasks are performed. It is also responsible for planning a given mission optimally by generating the best path for a given mission with a proper distribution of load.

Road Network Provider is a service that will provide a road network to the other services. The road network can also be viewed as a map service.

Situational Awareness is a service that provides situational data to the vehicles. For example, the fleet has access to traffic and weather data through this service. For other related functions, a good reference would be NordicWay project [1].

Dynamic Replanner function is responsible for re-planning a failed mission.

Monitoring is a service that can monitor vehicles in real-time. Several interviewees mentioned a cloud function like Volvo's Dynafleet service [2].

State Maintainer is a service or a component that can be used to maintain vehicle state on the cloud. In case of connectivity loss with the control tower, it can simulate and predict the vehicle's actual position.

Remote Operation is a service that allows an operator from the control tower to operate the vehicles in case of emergency situations and evacuation.

Diagnostics & Maintenance System is a component that can provide vehicle diagnostics service and backup service. It is also responsible for the vehicle's internal health by identifying if the vehicle has issues with its engine, exhaust, transmission, ignition coils, oil tank, and more. A cloud service in the control tower can augment the onboard diagnostics and provide improved vehicle health monitoring and maintenance.

Evaluate is a component that evaluates the planned activity with results. The evaluation can be done at the production planning level and also specific mission level.

HMI/GUI A separate service for Human-Machine Interaction (HMI) and Graphical User Interface (GUI).

Based on the finding from the literature review and interviews a functional decomposition diagram has been established which demonstrates the mandatory functional requirements discussed. The following figure 4.1 illustrates the diagram.

[1] NordicWay Film: https://www.youtube.com/watch?v=gTrrl4ymvyc

[2] https://www.volvotrucks.com/en-en/services/optimized-business/fleet-management.html

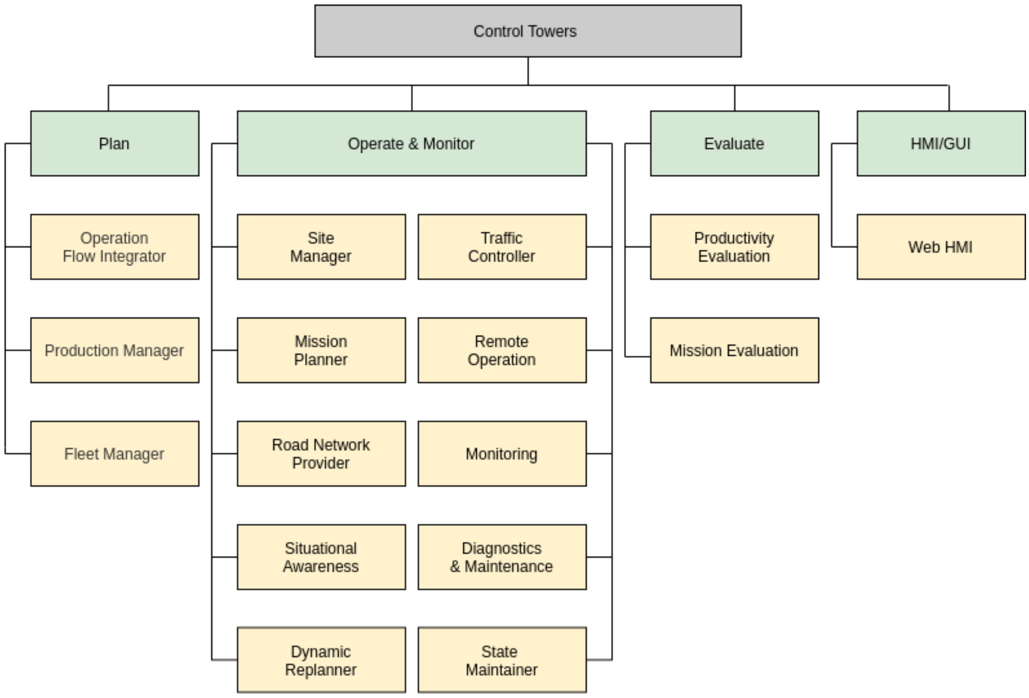

Figure 4.1: Functional Decomposition

4.1.3 Reconciliation of requirements

It is also important to report how the finding from the literature reconciles with the finding from the interview. the reconciliation is done based on thematic analysis. The use cases found in the literature were grouped into different themes (e.g. Traffic management, Fleet management, etc.) as given in the 4.1. Then those themes were realized with themes extracted from the interview data analysis. Due to the company secrecy issue, only the top-level theme are included from the interview data instead of detailed use cases. For that reason, the requirements might be seen as generic and top-level.

4.2 Quality Requirements - RQ2

The goal of the second research question is to investigate the non-functional quality requirements for an autonomous vehicle control tower. ISO 25010:2011 standard has been taken as a reference while investigating quality attributes. The following subsections discuss finding from the literature review and interviews.

RQ 2: What are the non-functional quality requirement for a control tower in light of ISO/IEC 25010:2011 standard and how to handle them in the reference architecture?

4.2.1 Literature Review

Current research on non-functional requirements and quality attributes is very diverse. Some researches have addressed the quality attributes from the point of general software design perspective and some research addressed that from a domain perspective. In this study, the focus was given to the research that is related to cloud support function for autonomous vehicles as in any system development some quality attributes (e.g. performance, availability) are inherently important.

The literature review was also a baseline for the interview questionnaires. From the literature following challenges as shown in the table 4.3 were found.

Table 4.3: Technical Challenges relevant to Control Tower development from the literature review

Technical Challenges	Relevant ISO Attributes
The communication between the control tower and the vehicle is subject to varying latency and delay time of cellular and wireless network [44] [51]	Availability
The Data load in the network gets amplified by high vehicle density corresponding to exchange in control message and data.[44] [51]	Performance
It plays an important role in accessing the cloud-based vehicle functions (CB-VF), e.g. in case of temporary lack of cellular coverage in a dead-spot zone, the onboard ECUs do not have access to the CB-VFs [44]	Availability
A lost or erroneous message might cause a malfunction of the vehicle control algorithms and create a safety risk [51]	Reliability
In the first generation of V2X communication systems (1G-V2X), vehicles periodically broadcast safety messages with an interval between 100 ms and 1s. This will lead to a high message rate as the monitored vehicles increase [51]	Availability, Performance
The architectural blocks must be designed to withstand structurally stresses induced by the inherent instability in the operating environment [42][40]	Performance
There is a critical need to efficiently manage mobility, resource heterogeneity (including sensing, computation, and communication), trust and vehicle membership (change in interest, change in location, resource denial and/or failure) [40]	Performance
Includes data security, cloud access control, securing vehicular communication, securing vehicular communication, securing location information such as traffic status reports, collision, etc [42]	Security
It is essential to efficiently sense and aggregate various types of sensor data, including traffic data, vehicle's health information, information about the environment (disasters, fire, etc.), movements of vehicles and citizens on roads, etc [52]	Reliability

4.2.2 Interview Analysis

During the interview, participants view that the control tower is subject to basic quality attributes like Security, Reliability, and performance efficiency that are common in any software system. However, there are some control tower specific non-functional requirements. Following are some non functional requirements that participants viewed as important during the interview.

Requirement: Capacity to Handling Multiple AVs (Performance)
Description: The control tower shall be able to handle (for example, registration, monitor, dispatch) multiple Autonomous Vehicles (AVs). This indicates to the total number of AVs from various autonomous projects.
Requirement's criteria:

- Minimum: 100 AVs
- Should have: 150 AVs
- Good to have: 200 AVs

Requirement: Performance (Latency): Multi-Vehicle Mission Planning Time
Description: Control tower needs to plan or re-plan multiple vehicle missions whenever there is a new transport booking or there is a delay in the operation sourcing from different actors. The result of the multi-vehicle mission planning will tell us whether there is a feasible solution given various resources e.g., number of registered vehicles, loading, unloading areas, charging stations, intermediate parking areas, and constraints e.g., number of transport bookings, driving speed, driving distance, time to load/unload/charge, gates. Since the arrival of the new transport booking is stochastic, there will be a need to run multi-vehicle mission planning to understand whether, given the current resources and constraints, it can meet the transportation goal or not. The criteria define the time for vehicle mission.
Requirement's criteria:

- Minimum: 30 seconds
- Should have: 15 seconds
- Good to have: 5 seconds

Requirement: Performance (Latency):Time to read the vehicle status
Description: Control Tower needs to monitor all the operating vehicles continuously to get information e.g., speed, heading, GPS location.
Requirement's criteria:

- Minimum: 500 ms
- Should have: 400 ms
- Good to have: 300 ms

Requirement: Time to send safety related command to vehicle (Latency)
Description: Control Tower needs to send various commands to a vehicle, e.g., an emergency stop that is related to safety. This requirement points to the end-to-end latency for such safety related commands. For example, if an "emergency stop" command is issued by a third party system this refers to the total time starting from the issue of the command (by an operator (time from the press of a button associated with sending a command) or a system) to the time the command reaches the vehicle.
Requirement's criteria:

- Minimum: 100 ms
- Should have: 80 ms
- Good to have: 50 ms

Requirement: System Error Logging
Description: Error log files should be saved on web server. Critical Errors should be written to Windows event log.
Requirement's criteria: NA

Along with the non-functional requirements the participants were also asked to give their view on ISO 25010: 2011 standard quality attributes. The participants viewed the following quality attributes that are very critical for the control tower:

1. Security.
2. Reliability
3. Maintainability

The prioritization of the attributes has been done by the 100$ prioritization technique. All the participants were given an imaginary 100 dollars to spend on the different quality attributes. The following table 4.4 illustrates the dollar distribution on different quality attributes. The attributes are listed along with their system perspective (e.g. Runtime, Transition, etc.).

Table 4.4: Quality Attribute Prioritization

Perspective	Quality Attributes	Sub Attributes	Score
Runtime	Functional Suitability	Functional Completeness Functional Correctness Functional Appropriate-ness	10
Revision	Maintainability	Modularity Reusability Analysability Modifiability Testability	14
Runtime	Reliability	Maturity Availability Fault Tolerance Recoverability	22
Runtime	Performance	Time behaviour Resource Utilisation Resource Utilisation	11
Transition	Portability	Adaptability Installability Replaceability	7
Transition	Compatibility	Interoperability	5
Runtime	Usability		3
Runtime	Security	Integrity Confidentiality Non-repudiation Accountability Authenticity	28
		Total	100

The table 4.4 represents the aggregated distribution of all participant's data. However, it is also interesting to see how different participants with different roles in the organization spend the money on different attributes. For example, the project manager spent more money on *Maintainability* than *Reliability*. But the domain architect put more value to *Reliability* and *security*.

While it comes to the point about how to address these quality attributes in the control tower architecture, then the opinion is diverse. Some participants think it is context-dependent and constrained by the SLA (service-level agreement) and some participants think its best effort with existing resources. One of the participants viewed, we can always have better availability with some sort of redundancy. For example, for typical operations, public cloud service can be used because it brings flexibility but for certain critical operations, we can have redundant communication channels. We can create redundancy by direct radio communication either through 5G or other means of radar communications. For example, radio communication over cellular communication can be used for certain types of functionalities e.g., an emergency stop that is not dependent on infrastructure around it.

An autonomous vehicle control tower system is essentially a collection of heterogeneous systems. Some participants viewed it as very important to have a system architecture that is easy to integrate with other systems. So, it is important to design a system with an open mind so that it can be integrated with other systems. Some participants also viewed that when a nonfunctional requirement like latency or availability is very critical for a system operation then it becomes a functional requirement. For example, when there is a government regulation that a human operator needs to take control of the vehicles, the latency immediately becomes a functional requirement. The overall quality aspect of a control tower should be taken as a mission-critical system. Most of the human factor designs are nonfunctional, and these are critical. Anything related to human factor design is extremely important.

4.3 Reference Architecture - RQ3

The goal of the third research question is to find out how to design and present a reference architecture for a control tower. The focus of the literature review for this research question was to find out how to represent the reference architecture and the focus for the interview was to identify architectural domain requirements and architectural technical requirements. The research question is stated here again.

RQ 3: How should a reference architecture for a control tower look like?

4.3.1 Literature Review

The state of the literature to representing an architecture is mainly influenced by ISO/IEC/IEEE 42010:2011 standard and according to this, an architecture can be illustrated with the aid of different views and viewpoints. As defined by ISO 42010 a view is ”A representation of a whole system from the perspective of a related set of concerns.”. A view is often materialized in a model and corresponds to one of the facets of software. For example, a logical view (usually represented in class diagrams in UML), or a physical view, often represented as a deployment diagram. In turn, an architecture viewpoint is used to establish notations, conventions, techniques, and methods, which frame particular concerns and are conceived for specific system stakeholders [9]. So a viewpoint corresponds to a collection of patterns, templates, and conventions for establishing a particular view. The following figure 4.2 illustrates ISO/IEC/IEEE 42010 contents of an architecture framework.

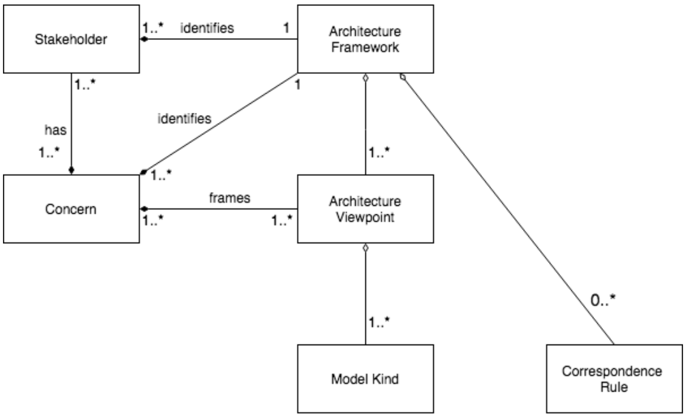

Figure 4.2: ISO/IEC/IEEE 42010: contents of an architecture framework (adapted from [9])

4.3.2 Interview Analysis

According to participants a reference architecture for an autonomous vehicles control tower's architecture should address both architecture domain requirements and the architectural technical requirements into the architecture. The domain requirements are the key functional requirements that are identified in finding answers to the first research question (RQ1). The technical requirements that participants viewed as important are illustrated in the table 4.5.

Table 4.5: Architectural technical requirements

SL #	Description
1	The RA must enable the development of CT with blocks developed in different programming languages
2	The RA must enable the development of CT through dynamic architecture (changing and moving shapes of structures, rapid development)
3	The RA must enable the instantiation of CT through the composition of components
4	The RA must enable the instantiation of CT with monitoring and management of quality attributes
5	The RA must enable the development of CT that achieves a high degree of security and reliability
6	The RA must enable the development of CT that allow the use of heterogeneous databases
7	The RA must enable the development of CT that allow the aggregation of data from heterogeneous sources (e.g. Weather data, sensor data, etc.)
8	The RA must enable the continuous development of CT

4.4 The Reference Architecture

The reference architecture is presented as a collection of different kinds of views in the following subsections. The first view is the *Structural View* of the different components. It represent the functional decomposition of the control tower. The second view is the *Logical View* of the component and finally, the third view is the *Deployment View* of the reference architecture.

4.4.1 Structural View

The structural view makes it possible to delimit the context of the RA, as well as its internal elements. The main structures of the reference architecture are components, which run independently and could be developed using any programming language. This view has 19 components that encapsulate the control tower functionalities. All components have APIs for the definition of provided and required interfaces. The following figure 4.3 illustrates the structural view of the reference architecture.

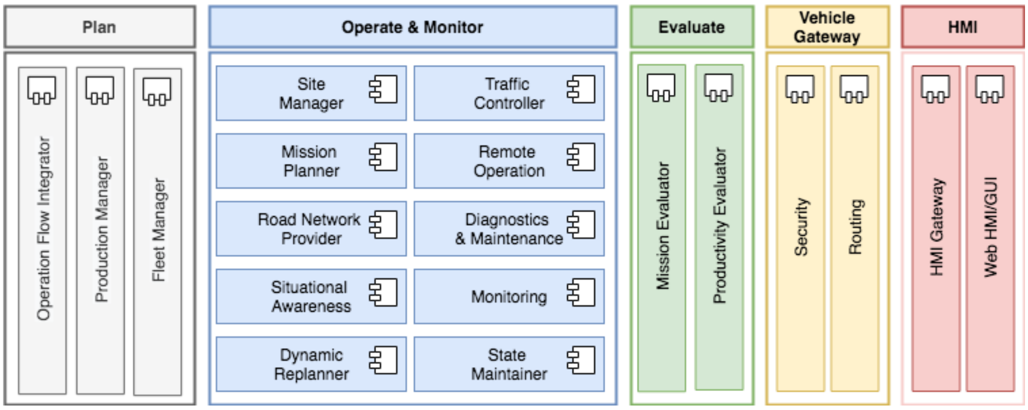

Figure 4.3: Structural view of the reference architecture

4.4.2 logical View

The purpose of the logical view is to visualize the order of how the different components interact with each other and how the data flow from component to component. The provided and required interfaces are replaced with arrows for the sake of simplicity. The basic concept of this view is that a component represents a modular part of the system and the manifestation of the component is replaceable within the environment. The following figure 4.4 illustrates the logical view of the reference architecture.

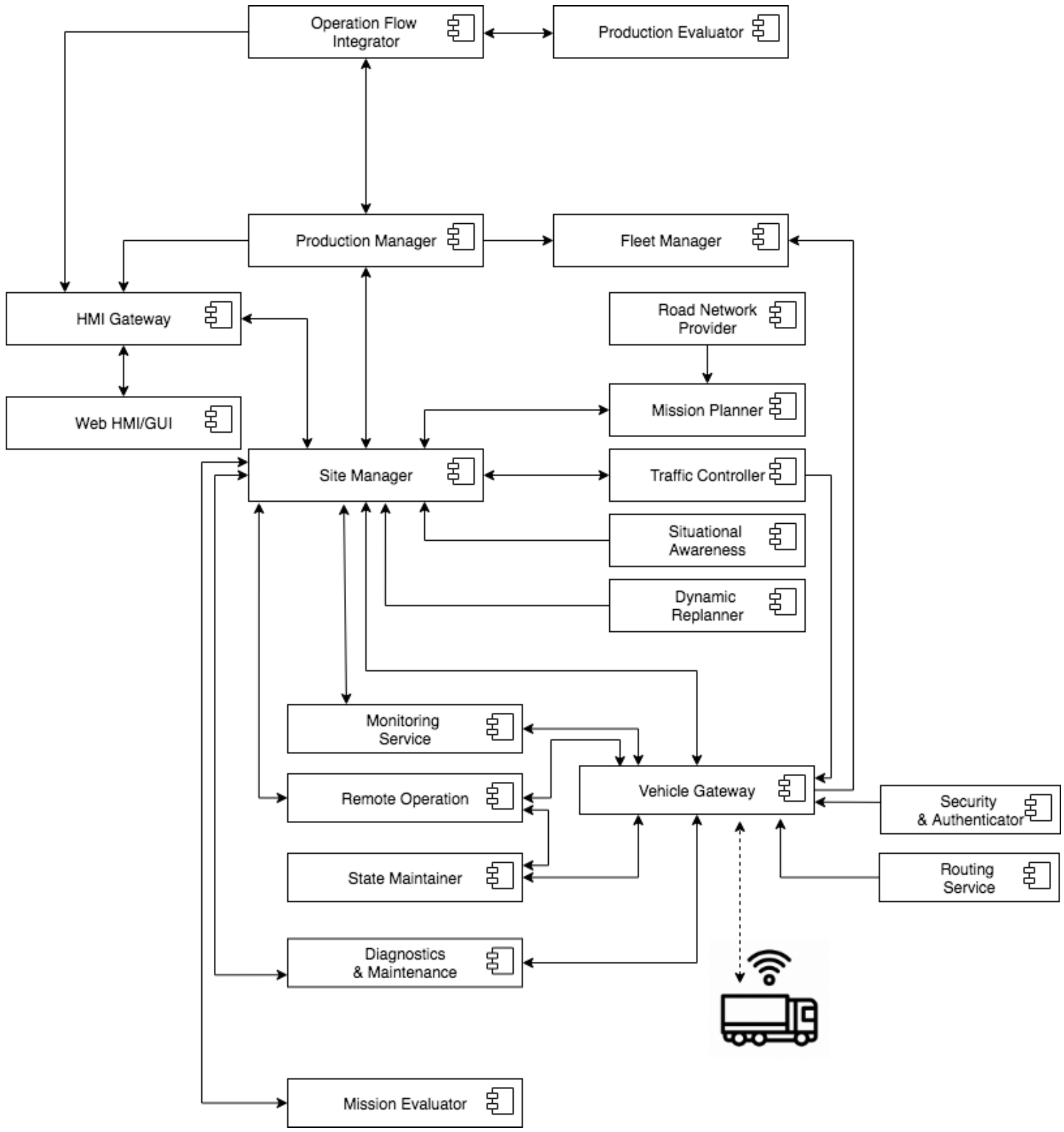

Figure 4.4: Logical view of the reference architecture

4.4.3 Deployment View

The deployment view presents an instantiated state of the reference architecture. One of the main purposes of this view is to explain how the architectural decision has been made to address critical quality attribute like security and availability.

Security has been viewed as the topmost quality attribute by the different participants during the interview and evaluation sessions. Several architectural decisions have been made to address the security of the control tower. The components inside the control tower use the specification of Open Container Initiative (OCI [3]) as an execution environment that favors the execution of components in independent

[3]https://opencontainers.org/

platforms. So it provides isolation, restricts access, and improves security. However, containerized deployment can security nightmare if properly not controlled. The figure 4.5 illustrates the deployment view of the architecture.

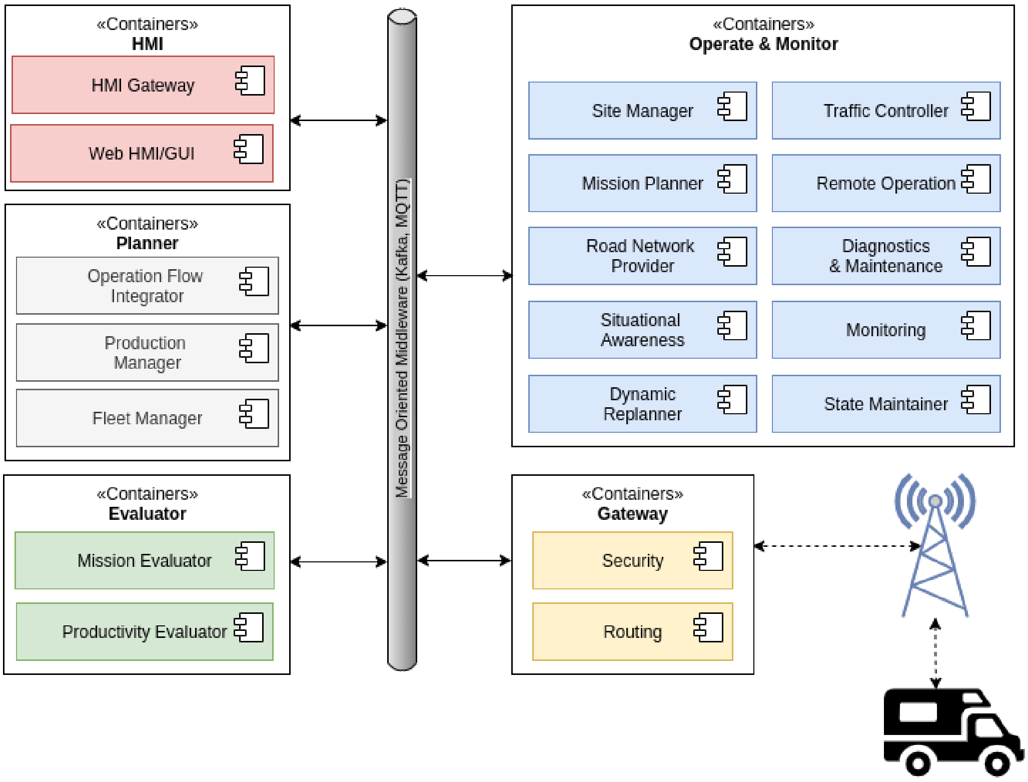

Figure 4.5: Deployment view of the reference architecture

4.5 Evaluation of the architecture using ATAM

The proposed reference architecture was evaluated using Architecture Tradeoff Analysis Method(ATAM). Running an ATAM can involve as few as three to five stakeholders or as many as 40 to 50 [3]. A small-scale ATAM has been conducted with six stakeholders. The following sub-section represents the activities and output of the ATAM.

4.5.1 Performed Steps

The steps for the full phase of ATAM are quite extensive. However, the process may not need every step and stakeholders[3]. The following table 4.6 illustrates the steps the were performed to evaluate the proposed reference architecture using ATAM.

Table 4.6: Architectural technical requirements

Step	Activity	Stakeholder Groups
1	Present the ATAM	All stakeholders
2	Present the architecture	All stakeholders
3	Identify architectural approaches	Stakeholders from Architecture team
4	Generate quality attribute utility tree	Stakeholders from Architecture team
5	Analyze architectural approaches	Stakeholders from Architecture team
6	Present the results	All stakeholders

4.5.2 Outcome from the Evaluation

The ATAM exposed some serious problems about the clarity of the functional requirements and in the documentation of the architecture. Thus it was hard to evaluate some quality attributes like performance and maintainability. In the following section, a brief summary has been given.

4.5.2.1 Documentation

The architectural documentation was not up to mark during the evaluation. The structural and logical view was not enough to evaluate the architecture.

4.5.2.2 Requirements

Some of the requirements were surfaced out during the evaluation as the ATAM involves different stakeholders and improves communication among them.

4.5.2.3 Sensitivities and Tradeoffs

Security and Reliability are considered the most sensitive points of the system. The availability of the system is highly sensitive to the latency of the communication channel.

4.5.2.4 Architectural Risks

The overall architecture is subject to the single point of failure as there is separate a gateway for the communication between the vehicles and the control tower.

4.5.2.5 Reporting Back and Mitigation Strategies

Some corrective steps were taken based on evaluation feedback. A deployment view has been added to provide an instantiated view of the reference architecture. With that view, some of the architectural sensitive points are revealed clearly.

5

Analysis and Discussion

The purpose of the chapter is to discuss the results from the research goal perspectives and explain the outcome considering the research question. It also explains how the requirements were prioritized and who are the targeted users of the proposed reference architecture. Furthermore, there is a discussion about the level details of the proposed architecture. In the end, it describes the limitation and ethical considerations of the study.

5.1 Analysis on Research goal

The goal of the study was to develop a reference architecture for autonomous commercial vehicle control towers by identifying functional and non-functional quality requirements. The aim was to develop an architecture that could be used by multiple OEMs. However, while conducting the study it was observed that it is a very challenging goal considering the current maturity of knowledge about the requirements of control tower functionalities. There are also some other factors like lack of common vocabulary about the requirement and involving the right stakeholders in the study.

5.2 Outcome considering RQ1

Considering the research question the outcome could be viewed as very useful for the industry especially for Volvo Autonomous Solutions. To answer the first research question, this study elicits the required functional requirements that are needed for a secure and reliable control tower. Some of the requirements like *Vehicle State* and *Dynamic replanning* are the result of multiple sessions with different stakeholders which values are less obscure. But to have a reliable and efficient control tower those are very important.

Some of the findings from the literature for the functional requirements (i.e. RQ1) do not exactly match feedback from the stakeholder as the context of the study found in the literature is a bit different. However, those requirements provide a baseline for the discussion with the stakeholder during the workshops and interviews and helped while analyzing interview data. For example, Maheswaran et al.

[41] reports a use case for efficient use of road space while proposing a fog computing framework for autonomous driving assistance. During the interviews, several participants mentioned the control tower capability of having an efficient traffic planner so that the road network in the site can be utilized properly. In that sense, there is a kind of data triangulation between literature and stakeholder feedback. But it is apparent there is a miss-match in the vocabulary used in literature and industry while expressing the requirements or use cases.

Maturity of the search and development of autonomous vehicle control would be a reason for the contradiction between literature and industry. However, when it comes to the bare minimum functionality of the control tower, both literature and feedback of the stakeholder reflect almost the same. Finding from both methods suggest that an autonomous vehicle control tower must have three functionalities which are *planning mission*, *execute mission* and *monitor the vehicle*.

5.2.1 Requirement prioritization

Requirement prioritization is usually done based on certain criteria like objectives, risks, quality factors, or viewpoints of stakeholders. There was a challenge to select the right set of requirements out of superset of candidate requirements so that all the key interests and preferences of the critical stakeholders are fulfilled. The 100$ dollar method was not feasible to use as the requirement was explored through semi-structured interview and were fine-tuned gradually. One possibility was there to go back to all stakeholders and do the prioritization using the 100$ method. But it was not possible due to time and resource constraints.

To mitigate this challenge a requirement technique called a grouping (also known as Numerical Assignment) has been used to define the right set of requirements. This prioritization was done during the evaluation process. In this process, all the requirements were grouped into the critical levels to have a functional control tower. The requirements were tagged as compulsory, very important, important, or not important. But the process was not that rigorous as that was done by only six stakeholders thus the potential key interest of some key stakeholders were overlooked. For example, in the proposed architecture there is a component called 'State Maintainer' to maintain the state of the vehicle in the cloud but if we involve stakeholders from the finance team then they would view that it is not important functionality for the control tower. Because developing a component like a state maintainer or digital twin is complex and expensive thus they might be viewed that requirement as not mandatory.

5.3 Outcome considering RQ2

The second research question was to investigate non-functional requirements and critical quality attributes. This study produces some non-functional requirements that could be very important for further study. Analyzing the contrasting view of different stakeholders on different quality attributes is another outstanding outcome of this research question. It gives the mandate that it is always good to add various stakeholders in the architectural design process and from the beginning of the process.

Eliciting every kind of non-functional requirement is very difficult upfront [53]. All participants viewed that it is an iterative process and the requirements expand with the system life cycle. This might be the case that part of the system does not uncover until the system reaches to certain milestone [53]. So it is very impractical to claim the completeness of non-functional requirements identification.

Validating the non-functional requirements against the proposed architecture is another difficult job. Very vague feedback is found when asked the different participants how they validate if certain quality attributes are addressed in an architecture. For example, when asked how they address *Performance efficiency* some of the participants mentioned that they do some sort of load test to verify that.

Security, *Reliability* and *Maintainability* are viewed as most critical quality attributes by the stakeholder. However, it is difficult to see how these attributes are addressed properly in the proposed reference architecture. The proposed reference architecture grouped different functional requirements to boost the maintainability of the system. But that does not guaranty the *Maintainability*. Having a maintainable system is much more than that. Having a maintainable code base, appropriate separation of concern and low dependency among the module play a big role in maintainability. But that is hard to validate through the reference architecture and there is no visible measure in the proposed architecture.

There is a contrasting finding between literature review and interview regarding *performance* of the control tower. According to the current literature, performance should be considered as a critical quality attribute where participants of the interview do not view that as important as *Security*, *Reliability* and *Maintainability*. One reason could be is that the control tower is a cloud enteric microservice-based application that enables to increase the number of microservice instances during high load, and reduces them during times with less load. But still, the requirements like transaction response times and throughput are as critical as service reliability and availability. However, industry people viewed that these requirements might be served by best effort using the current technology and infrastructure.

5.4 Outcome considering RQ3

The outcomes from the third question are the architectural artifact and the technical requirement for the reference architecture. Four different architectural artifacts have been produced as part of that. The type of artifacts (i.e. different views) is backed by industry experts and also supported by the literature.

5.4.1 Target user of the architecture

The target audiences of the architectural guideline given in the study are mainly the autonomous vehicle manufacturers who offer transportation as a service to other businesses or organizations. Volvo Truck, Navistar, Volkswagen, PACCAR, MAN Truck, and Scania are in the list to name some of these types of companies. They all are heavy-duty vehicle manufacturers and they all are active in autonomous solutions development. Volvo Group and Scania both have created separate wings for autonomous solutions for their truck fleet. This study could be influential for other research projects like AVTCT by Drive Sweden [1]. Volvo Autonomous Solutions and Scania both are very active partners of the AVTCT project [2] and using that channel they can collaborate and share learning experiences.

In general, it is unrealistic to anticipate the competing company would adopt a single reference architecture. However, the targeted domain is a bit special and the autonomous solution stack is not their core business. They are competition mainly on a vehicle manufacturing level. So even though there is competition among them but the market opportunity is really high if they can draw attention from end-users and provide reliable solutions. Furthermore, autonomous vehicle control tower requires some services that are very difficult and costly to build a single company. For example, services like situational awareness need to integrate many other services like weather forecast, road network, traffic control, etc. to provide real situational awareness. Having a reference architecture would help to build such services and expedite the overall autonomous vehicle development. It will also encourage entrepreneurs to start up new companies to provide these kinds of services and companies like Volvo and Navistar can use that service.

A Reference Architecture is also strongly linked to company (or consortium, e.g. MIPI) mission, vision, and strategy [10]. The strategy determines what multi-dimensions have to be addressed, what the scope of the Reference Architecture is, what means, such as synergy, are available to realize mission and vision [10]. But the mission and vision are company-specific and the reference architecture is materialized based on them. The following figure 5.1 illustrates how multiple organizations can use a reference architecture.

[1] https://www.drivesweden.net/en/projects
[2] https://www.drivesweden.net/en/projects-5/avtct

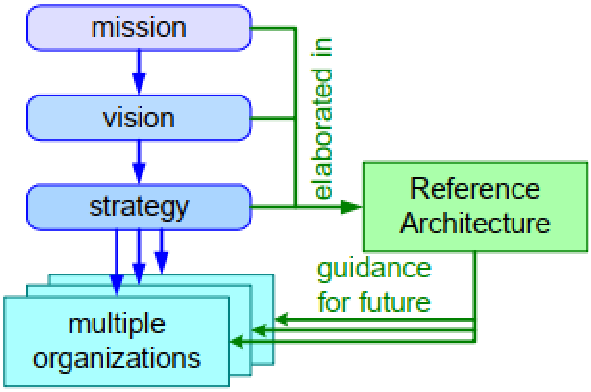

Figure 5.1: A Reference Architecture elaborates mission, vision and strategy to provide guidance to multiple organizations (adapted from [10])

5.4.2 Abstraction level of the proposed reference architecture

Overall, the identified functional and non-functional would feel a bit generic. For example, continuous development and support for multiple languages are pretty common in every software development. So it can be argued that they are not domain-specific requirements. Indeed they are not domain-specific requirements but implementing these requirements in the autonomous vehicle domain is very complex. For example, continuous development for an e-commerce site would be fairly easy but in control tower development the decision to what to develop first is a bit hard and that decision needs to be wise as testing with a real vehicle is expensive, and it is very tough to recall the vehicle for the operational site.

Also, the nature of the reference architecture is kind of abstracted as the competing company would not like to share sensitive detail about a certain part of the system. For example, in the proposed reference architecture there is a component named 'Traffic Controller' which is responsible for planning the mission optimally considering the given constraints. But the actual implementation of the component can be different. One company could implement a rule-based traffic controller to determine the best path whereas another company could implement a traffic controller based on machine learning where they define a model and use that model to define the best path. So here the reference architecture provides the notion of having a traffic controller but the actual implementation of the service will win the competition and the company that excels in that implementation would be able to generate better revenue. According to Robert Cloutier et al. [10] the nature of reference architecture is indeed at a high level and actual instantiation of the architecture provides details of the requirements. They provide a step-by-step process of how a reference architecture gets realized into an actual system. The Following 5.2 figure illustrates that step-by-step process.

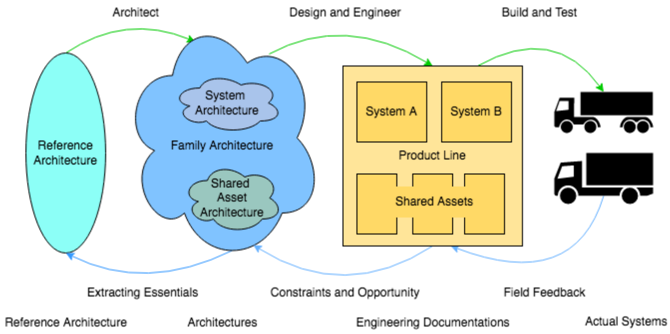

Figure 5.2: The steps a Reference Architecture is transformed into actual systems (adapted from [10])

It might still be considered the proposed architecture is not refined enough for actual implementation However doing so could miss the whole point of a reference architecture. Certain details of implementation are missing precisely as they are implementation-specific. To conclude, no matter how good a reference architecture is, it can always be ruined by bad implementation. Therefore, a good question would be how can we go for instantiating the given reference architecture?

5.5 Threats to Validity

5.5.1 Internal validity:

Since this work is done individually, the main threat that could be seen is experimenter bias. However, it was tried to mitigate with the usage of iterative approach and by engaging various stakeholders. All the results are collected from domain experts and the individual results are refined with discussion with the supervisors during the process. The participants of the interview were also asked to give their feedback on the materials of the interview and workshops.

5.5.2 External validity:

This thesis work was done in Volvo Autonomous Solutions, which raises questions on the generalizability of the findings. So, instead of focusing specifically on their company's truck, which is still in its developing stage, this study tried to generalized to autonomous vehicles. Using the study results, the company should be able to tune in the requirements that they need, which is not part of this work. It is also supported by the fact that most of the general functionalities should be applicable to autonomous vehicles.

5.5.3 Construct validity

Construct validity is concerned with the extent to which the study measures what it claims to be measuring. This is potentially compromised by clarifying the terms that are used in the study. To ensure non ambiguity with the definition of a control tower, apart from asking their opinion on the view of the control tower, a video of a remote control tower was presented. The participants were asked about their opinion and how do they think the concept of a control tower would vary when it comes to autonomous trucks.

5.5.4 Reliability

Reliability is concerned with producing the same consistent result with other researchers on repeating the study. We have tried to ensure reliability by including the details of interview participants and survey respondents in terms of their role and years of experience. The interview template and survey questions have also been presented in the appendix. For the literature review, the databases and keywords used in the search process are also provided.

5.5.5 Conclusion validity

The sampling is very important and a wrong sampling may severely impact the results of the study. Idealistically, the sampling should be random from a relatively homogeneous population (where each person in the population has a similar experience with coding, programming, programming language, etc). For this study, the participants were picked using convenience which could lead to the threat of conclusion validity.

5.5.6 Informed consent

The interview participants were informed about the study purpose and their consent was taken orally for recording the interview. They were also informed that the recording will be deleted after the completion of this work. The survey respondents were informed about the motive to undertake the survey and the duration to complete it. The email invitees of the survey were also informed about the deadline to complete it. For both interview and survey, the participants were ensured of safeguarding anonymity and were informed on how their feedback will be used in this work.

5.5.7 Confidentiality and Anonymity

Anonymity requires the researcher not to know who the participants are and confidentiality involves the researcher knowing the identity, but not disclosing the information. Before the start of this study, a confidentiality form was signed in agreement with the company as to not disclose any company-related information.

6

Conclusion and Outlook

This case study reports a process of developing and evaluating a reference architecture for autonomous heavy-duty vehicle control towers by identifying functional and non-functional quality requirements. The process starts with eliciting functional requirements and non-functional requirements by literature review and interviewing the industry experts. Then a reference architecture is developed and evaluated by the architecture tradeoff analysis method (ATAM). This study followed a systematic approach to develop the reference architecture (RA) as it is a special type of architecture.

The functional requirements identified in this study are based on literature review and feedback from experienced domain experts. The requirements for the autonomous vehicle control tower are not abundant is in the literature. There is very little research that directly matches with control tower development. But the findings from the literature are quite handy especially to create a context for workshops and interviews. So there is a little data triangulation when it comes to functional requirements.

This study gathered important quality attributes for the control tower development. It then prioritized the attributes based on their severity for the control tower. It is true that most of the identified quality attributes are common for every software development however the challenges reported from the literature review are very insightful and ranking them based on the feedback from industry people is helpful in developing a robust reference architecture for the autonomous vehicle control tower.

The primary artifact of the proposed reference architecture is the *structural view* which presents the basic functionality supported by the architecture and how they should be grouped into the different components. The proposed logical and deployment views are mere for illustration purposes and can be done in a different way based on development technologies (i.e. programming language, framework, etc.) and deployment environment (i.e Cloud, On-premise).

To develop and adopt industry-wide reference architecture it is required to have a common platform where different organizations can take part in and contribute to developing the architecture. So a study similar to this one could be conducted in some potential organizations that are active in autonomous vehicle solutions development. Then the requirements both functional and non-functional can be validated and prioritized by a wide range of stakeholders and eventually refine into a reference architecture. Currently, Drive Sweden provides that kind of platform but the projects they carried out so far are limited to pre-study or not focusing to develop an industry-wide reference architecture for the autonomous vehicle control tower.

6.1 Contribution

The main contribution of this study is the identified functional requirements. Another significant contribution of the study is the prioritization of the quality attributes as the prioritization is done systematically and in an industry setup. The proposed architecture can be taken as a motivation or point of reference for any autonomous vehicle control tower or site management. The requirements identified in this study are relevant for autonomous vehicle site management in confined areas or hub-to-hub transportation systems. Process wise the study also contributes to how to develop a reference architecture. The steps performed to develop the reference architecture are verified with the research done in the other related domain.

This study also provides some industry insight back to researchers. The things that are critical from a research perspective could be considered optional to the industry people. For example, in literature, the quality attribute Performance is considered highly critical. But to industry people, this is more a customer requirement that comes as a service level agreement (SLA).

Developing a reference architecture demands experience and domain maturity. The research about control tower development for autonomous vehicles is relatively young. Thus the study contributes to the software engineering body of knowledge when it comes to eliciting functional and non-functional requirements. Overall this study will help industry practitioners to develop references for their respective domain and software engineering researchers to research further on control tower reference architecture.

It is true that the primary beneficiary of this research work is Volvo Autonomous Solutions. However, this study also contributes to the research community. One of the key contributions is that the proposed reference architecture is developed by following well know process found in the current literature. So this study validates the concept found in the literature in an industry setup which could be considered as a scientific contribution as it will encourage future research workers to follow a similar kind of process to develop reference architecture in other domains.

6.2 Future Work

This study is done in a limited context. Most interviewees who participated in this study have experience only in developing a site control system for a confined area. So it would be nice to how it fits with other related domains. For example, it would be quite interesting to compare how the gathered functional requirements fit into the control tower for the autonomous hub-to-hub transport system where it will include not only a confined area but also a public road.

This study only listed and ranked important quality attributes. But it would be nice to see how they can be addressed systematically in the reference architecture. The evaluation of the proposed architecture is not that extensive due to time and resource constraints as this thesis is done by a single person. So an extensive study can be done on how to apply full phased ATAM to evaluate reference architecture and compare the ATAM evaluation technique with other evaluation techniques that exist in the literature.

Bibliography

[1] M. Galster and P. Avgeriou, "Empirically-grounded reference architectures: a proposal," in *Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality of software architectures–QoSA and architecting critical systems–ISARCS*, 2011, pp. 153–158.

[2] E. Y. Nakagawa, M. Guessi, J. C. Maldonado, D. Feitosa, and F. Oquendo, "Consolidating a process for the design, representation, and evaluation of reference architectures," in *2014 IEEE/IFIP Conference on Software Architecture*. IEEE, 2014, pp. 143–152.

[3] R. Kazman, M. Klein, and P. Clements, "Atam: Method for architecture evaluation," Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, Tech. Rep., 2000.

[4] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

[5] A. Drive Sweden. (2020) Automated vehicle trafic control tower. [Online]. Available: https://www.drivesweden.net/sites/default/files/content/bilder/avtct_final_report.pdf

[6] S. Behere, M. Törngren, and D.-J. Chen, "A reference architecture for cooperative driving," *journal of Systems Architecture*, vol. 59, no. 10, pp. 1095–1112, 2013.

[7] J. Schroeder, D. Holzner, C. Berger, C.-J. Hoel, L. Laine, and A. Magnusson, "Design and evaluation of a customizable multi-domain reference architecture on top of product lines of self-driving heavy vehicles-an industrial case study," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2. IEEE, 2015, pp. 189–198.

[8] J. W. Creswell, *Educational research: Planning, conducting, and evaluating quantitative*. Prentice Hall Upper Saddle River, NJ, 2002.

[9] P. Pelliccione, E. Knauss, S. M. Ågren, R. Heldal, C. Bergenhem, A. Vinel, and O. Brunnegård, "Beyond connected cars: A systems of systems perspective," *Science of Computer Programming*, vol. 191, p. 102414, 2020.

[10] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, and M. Bone, "The concept of reference architectures," *Systems Engineering*, vol. 13, no. 1, pp. 14–27, 2010.

[11] S. Gopalswamy and S. Rathinam, "Infrastructure enabled autonomy: A distributed intelligence architecture for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 986–992.

[12] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[13] S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *Journal of modern transportation*, vol. 24, no. 4, pp. 284–303, 2016.

[14] A. Khoury. (2018) Retrieved from digital trends. [Online]. Available: https://www.digitaltrends.com/cars/waymo-now-a-full-ride-hailing-service/

[15] S. Burton, I. Habli, T. Lawton, J. McDermid, P. Morgan, and Z. Porter, "Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective," *Artificial Intelligence*, vol. 279, p. 103201, 2020.

[16] L. Kang, W. Zhao, B. Qi, and S. Banerjee, "Augmenting self-driving with remote control: Challenges and directions," in *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, 2018, pp. 19–24.

[17] D. J. Fagnant, K. M. Kockelman, and P. Bansal, "Operations of shared autonomous vehicle fleet for austin, texas, market," *Transportation Research Record*, vol. 2563, no. 1, pp. 98–106, 2015.

[18] M. Vagia and Ø. J. Rødseth, "A taxonomy for autonomous vehicles for different transportation modes," in *Journal of Physics: Conference Series*, vol. 1357, no. 1. IOP Publishing, 2019, p. 012022.

[19] S. Bsaybes, A. Quilliot, and A. K. Wagler, "Fleet management for autonomous vehicles using flows in time-expanded networks," *Top*, vol. 27, no. 2, pp. 288–311, 2019.

[20] X. Zhao, R. Darwish, and A. Pernestål, "Automated vehicle traffic control tower: A solutionto support the next level automation," *International Journal of Transport and Vehicle Engineering*, vol. 14, no. 7, pp. 283–293, 2020.

[21] S. Behere and M. Törngren, "A functional reference architecture for autonomous driving," *Information and Software Technology*, vol. 73, pp. 136–150, 2016.

[22] S. Martínez-Fernández, C. P. Ayala, X. Franch, and E. Y. Nakagawa, "A survey on the benefits and drawbacks of autosar," in *Proceedings of the First International Workshop on Automotive Software Architecture*, 2015, pp. 19–26.

[23] S. J. Martinez Fernandez, “Towards supporting the adoption of software reference architectures: An empirically-grounded framework,” in *Proceedings 11th International Doctoral Symposium on Empirical Software Engineering (IDoESE@ ESEM)*, 2013, pp. 1–8.

[24] E. Y. Nakagawa, P. O. Antonino, and M. Becker, “Reference architecture and product line architecture: A subtle but critical difference,” in *European Conference on Software Architecture.* Springer, 2011, pp. 207–211.

[25] K. D. Wentzel, “Software reuse-facts and myths,” in *Proceedings of 16th International Conference on Software Engineering.* IEEE, 1994, pp. 267–268.

[26] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[27] L. Bass, P. Clements, and R. Kazman, “Software architecture in practice,” 2012.

[28] P. Kroll and P. Kruchten, *The Rational Unified Process Made Easy: A Practitioner’s Guide to the RUP: A Practitioner’s Guide to the RUP.* Addison-Wesley Professional, 2003.

[29] A. J. Rohling, V. V. G. Neto, M. G. V. Ferreira, W. A. Dos Santos, and E. Y. Nakagawa, “A reference architecture for satellite control systems,” *Innovations in Systems and Software Engineering*, pp. 1–15, 2019.

[30] B. P. Gallagher, “Using the architecture tradeoff analysis methodsm to evaluate a reference architecture: a case study,” CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, Tech. Rep., 2000.

[31] A. Kauppi, J. Wikström, B. Sandblad, and A. W. Andersson, “Future train traffic control: control by re-planning,” *Cognition, Technology & Work*, vol. 8, no. 1, pp. 50–56, 2006.

[32] J. Yuh, G. Marani, and D. R. Blidberg, “Applications of marine robotic vehicles,” *Intelligent service robotics*, vol. 4, no. 4, pp. 221–231, 2011.

[33] D. S. Triangulation, “The use of triangulation in qualitative research,” in *Oncology nursing forum*, vol. 41, no. 5, 2014, p. 545.

[34] S. Kvale, *Interviews: An introduction to qualitative research interviewing.* Sage Publications, Inc, 1994.

[35] D. R. van Solingen and E. W. Berghout, *The Goal/Question/Metric Method: a practical guide for quality improvement of software development.* McGraw-Hill, 1999.

[36] D. S. Cruzes and T. Dyba, “Recommended steps for thematic synthesis in software engineering,” in *2011 international symposium on empirical software engineering and measurement.* IEEE, 2011, pp. 275–284.

[37] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[38] J. Bosch, *Design and use of software architectures: adopting and evolving a product-line approach.* Pearson Education, 2000.

[39] R. Lovas, A. C. Marosi, M. Emödi, Á. Kisari, E. Simonyi, and P. Gáspár, "Paas-oriented iot platform with connected cars use cases," in *2018 International Conference on Sensor Networks and Signal Processing (SNSP).* IEEE, 2018, pp. 409–420.

[40] S. Olariu, M. Eltoweissy, and M. Younis, "Towards autonomous vehicular clouds," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 1, no. 1, 9 2011.

[41] M. Maheswaran, T. Yang, and S. Memon, "A fog computing framework for autonomous driving assist: architecture, experiments, and challenges," *arXiv preprint arXiv:1907.09454*, 2019.

[42] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer applications*, vol. 40, pp. 325–344, 2014.

[43] S. Herrnleben, M. Pfannemüller, C. Krupitzer, S. Kounev, M. Segata, F. Fastnacht, and M. Nigmann, "Towards adaptive car-to-cloud communication," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops).* IEEE, 2019, pp. 119–124.

[44] F. Milani and C. Beidl, "Cloud-based vehicle functions: Motivation, use-cases and classification," in *2018 IEEE Vehicular Networking Conference (VNC).* IEEE, 2018, pp. 1–4.

[45] H. Al Najada and I. Mahgoub, "Autonomous vehicles safe-optimal trajectory selection based on big data analysis and predefined user preferences," in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON).* IEEE, 2016, pp. 1–6.

[46] S. Liu, J. Tang, C. Wang, Q. Wang, and J.-L. Gaudiot, "Implementing a cloud platform for autonomous driving," *arXiv preprint arXiv:1704.02696*, 2017.

[47] M. Aydin, M. Samarah, and K. O. Elish, "Enabling smart cities through v2x communication," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC).* IEEE, 2018, pp. 550–554.

[48] A. C. Marosi, R. Lovas, Á. Kisari, and E. Simonyi, "A novel iot platform for the era of connected cars," in *2018 IEEE International Conference on Future IoT Technologies (Future IoT).* IEEE, 2018, pp. 1–11.

[49] S. Kumar, S. Gollakota, and D. Katabi, "A cloud-assisted design for autonomous driving," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 41–46.

[50] A. Adhikaree, T. Kim, J. Vagdoda, A. Ochoa, P. J. Hernandez, and Y. Lee, "Cloud-based battery condition monitoring platform for large-scale lithium-ion battery energy storage systems using internet-of-things (iot)," in *2017 IEEE*

Energy Conversion Congress and Exposition (ECCE). IEEE, 2017, pp. 1004–1009.

[51] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of v2x communication in support of cooperative autonomous driving," *IEEE communications magazine*, vol. 53, no. 12, pp. 64–70, 2015.

[52] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, 2015.

[53] D. Ameller, C. Ayala, J. Cabot, and X. Franch, "Non-functional requirements in architectural decision making," *IEEE software*, vol. 30, no. 2, pp. 61–67, 2012.

A

Appendix 1

The Goal-Question-Metric (GQM) approach was used to define the objectives of the interview and design interview questions. Every question was defined keeping a goal in mind. The following table illustrates interview questions based on defined goals.

Table A.1: Functional requirements classification

Introductory question

1. What is your role in the organization?
2. What is your designation?
3. How long are you working in this domain?

Goal: Finding out the functional requirements of a Control Tower

1. What are the roles of the control towers for the autonomous trucks?
2. What are mandatory functions that a reference architecture for the autonomous truck's control tower should include?

Goal: Elicitate and prioritize non functional (technical) requirements

1. What are the critical non-functional/technical requirements for autonomous trucks control tower?
2. Would you like to give some quality scenarios?
3. If you would like to spend 100$ how would you distribute this amount to the aforementioned quality attributes?
4. What kind of tactics can be taken to address the quality attributes (e.g. Security and Availability) in the architecture?
5. Any other technical requirement you would like to highlight?

Goal: Establish a security framework to ensure secure communication among Control Towers and Vehicles

1. Do you think we should have a special security framework for control tower communication? Why? Why not?
2. Would like to recommend specific frameworks, tools, techniques, or protocols to address security concerns?

Goal: Identifying architectural style and communication/network protocol

1. Which architectural style(s) are most suitable for control tower development?
2. Which kind of network protocols (both transport and application-level) can be used for communication between Vehicle and Control Tower?
3. Do you think the message queue approach is suitable for communication between different control towers in general? why? why not?

Goal: Identifying development platform and data infrastructure and reliable connectivity

1. What are the advantages of selecting a cloud platform as the data and deployment platform?
2. What would be potential challenges if we select cloud infrastructure as a deployment platform and data infrastructure?
3. Do you think the development of edge computing and the 5G network could mitigate the aforementioned limitations and challenges?

Goal: Finding out the necessity of established a communication standard or protocol to communicate between different control towers and vehicles

1. Do you think a communication protocol should be established to exchange information between different control towers?
2. What other safety regulations and standards should be taken into consideration and why?

Goal: Finding architectural challenges and carry out architecture evaluation

1. What are the main challenges in architecting complex and distributed systems like control towers for autonomous vehicles?
2. How to carry lessons learned from one project to another project and evaluate and update the architecture accordingly?