University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Queue Management for Online Charging Station Brokering

Raphael Imfeld
Zurich, Switzerland
Student ID: 18-702-696

Supervisor: Prof. Dr. Bruno Rodrigues, Jan von der Assen, Prof. Dr. Burkhard Stiller
Date of Submission: January 09, 2025

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

Das vollständige Aufladen von E-Autos erfordert in der Regel mehr Zeit als das Betanken eines Fahrzeugs mit Verbrennungsmotor. Infolgedessen werden die Ladestationen über längere Zeiträume genutzt, was die Wahrscheinlichkeit erhöht, dass sich Warteschlangen bilden. Die Wartezeit für eine Ladestation, wächst proportional zur Länge der Warteschlange und verringert somit die Möglichkeit, eine Route zu planen. Obwohl sich dieser Effekt weniger auf E-Autos auswirkt, stellt er für E-LKWs eine Herausforderung dar, da sie ihre Routen unter Berücksichtigung von Zeit- und Entfernungsbeschränkungen planen, die zudem durch gesetzliche Faktoren bestimmt werden. In der aktuellen Forschung ist das Warteschlangenmanagement für E-Autoladestationen nur teilweise an den Anwendungsfall für E-LKWs angepasst und berücksichtigt nicht die Anforderungen, die sich aus der Routenplanung ergeben. In dieser Arbeit wird deshalb ein System für interaktives Reservierungs- und Warteschlangenmanagement (SIRQ) vorgestellt. Dieses bietet Funktionen zur Reservierung von Ladeplätzen während der Routenplanung oder einer Auktion, um die Warteschlange sofort zu überspringen wenn ein anderes Elektrofahrzeug bereit ist, den Ladeplatz gegen einen finanziellen Entgelt abzutreten. Der SIRQ-Prototyp wurde implementiert und anhand dreier realitätsnaher Szenarien evaluiert, die das unterschiedliche Ladeverhalten von Elektrofahrzeugen mit und ohne Zeitdruck modellieren.

Die Ergebnisse zeigen, dass SIRQ ein umfassendes und interaktives System für Warteschlangenmanagement bietet, welches unter bestimmten Annahmen die notwendigen Voraussetzungen für eine bessere Einplanung von Ladestopps gewährleistet.

Recharging electric vehicles to full capacity generally requires more time than refueling a vehicle with an internal combustion engine. As a result, charging stations remain in use for extended periods, which increases the likelihood of queues forming. The time spent at a charging station grows in proportion to the queue length, and therefore reduces the ability to plan a route. Although this effect is less impacting electric vehicles, it poses challenges for trucks, as they plan their routes according to time and distance constraints, which are further determined by regulatory factors. In current research, the queue management for electric vehicle charging station is only partially adapted to the electric truck use case and does not include the requirements imposed by route planning. Therefore, this thesis introduces a System for Interactive Reservation and Queue Management (SIRQ), which introduces functionalities to reserve charging spots during route planning or start an auction to immediately jump the queue if another electric truck is willing to cede the charging spot upon receiving a monetary incentive. The prototype of SIRQ is implemented and evaluated by applying three real-world inspired scenarios, which model the different charging behaviors of electric trucks with or without time pressure.

The findings demonstrated that SIRQ effectively developed a comprehensive and interactive queue management system. Under certain assumptions, SIRQ guarantees that electric trucks can either reliably plan their routes or reduce their wait times when a queue is formed at a charging station.

Acknowledgments

I wish to extend my sincere thanks to my thesis advisor, Prof. Dr. Bruno Rodrigues, for the essential guidance, encouragement, and engaging conversations over the recent months. The completion of this master thesis was made possible by his valuable feedback and the autonomy granted to explore a variety of methodologies independently.

To Yelena, your endless support and patience carried me throughout the entire journey of my studies. I am deeply grateful for the enlightening conversations, the probing inquiries, and your knack for brightening my day regardless of the challenging tasks I tackled. Your belief in my capabilities has inspired me to push boundaries and stay committed to what I am doing.

In addition, I would like to express my heartfelt gratitude to Jessica, Marion, Michael and Ramon who generously took the time to proofread my thesis. Their keen eye for detail, insightful feedback, and unwavering support were invaluable throughout the drafting process.

My appreciation also goes to Prof. Dr. Burkhard Stiller and the Communication Systems Group (CSG) at the University of Zurich for the opportunity to work on my thesis within their research team.

Contents

Chapter 1

Introduction

The charging of Electric Vehicles (EV), or batteries in general, typically shows a charging speed inversely proportional to the level of the battery's charge. The lower the battery level, the higher the charging rate. This standard effect is particularly observed in lithium-ion batteries, popular in EVs, due to the chemistry of these batteries and the battery management system [49]. For such batteries, lithium ions move from the cathode to the anode as the battery is recharged. When the battery level is low, there are several "vacant" points at the anode that these ions can occupy, allowing for a rapid flow of ions toward the anodes.

An analogy is to search for parking spots in an empty parking lot, which becomes more difficult with increasing number of parking spots taken. The practical result is that batteries charge at a slower rate as the battery level increases, implying that electric vehicles charge less efficiently at high battery levels while still occupying potentially valuable charging spots (CS).

This issue is accentuated with the increase in battery capacity of Electric Trucks (ET). The adoption is expected to grow significantly as companies and governments work to reduce emissions and meet sustainability goals. The International Energy Agency (IEA) estimates that ETs could account for 30% of all truck sales globally by 2030, driven by lower battery costs and stricter emission regulation [23]. Consequently, a larger amount of ETs long queues at CS could become a significant problem, resulting in delays and reduced productivity, as the American Transportation Research Institute (ATRI) states [40].

Previous studies explored several aspects of improving the efficiency of charging stations involving their optimal placement [46, 48] and scheduling [21, 41, 43]. In contrast to these approaches, in which the proposed solutions were discussed for private EVs, this Master Thesis explores a novel solution for ETs. The approach involves optimizing usage in peak utilization periods involving an online negotiation balancing the interests of the sellers (truck company already using a CS) and consumers (truck company in need of an immediate charge) to satisfy both.

An associated challenge in this scenario is the real-time management of queues, in which consumers may use an application to bid on a CS and skip an existing queue [2]. This

introduces a competitive layer to the queuing process that can be analyzed through the lens of a stochastic multiplayer game, as related studies suggested [44]. In this game, both sellers (of a CS) and consumers (vehicle owners) aim to maximize their respective benefits under practical real-world constraints.

Further, online auction systems are exposed to malicious intents of participants due to the required interactivity [34]. Hence, when introducing such approaches, countermeasures following the CIA (*Confidentiality, Integrity, Availability*) principle need to be implemented in order to allow practical applicability. By introducing a protocol that considers different usage scenarios for ET charging, an enhanced robust system is presented.

These scenarios are applied to define constraints which serve as a framework to address the specified challenges and establish parameters for the introduced functionalities. Consequently, the subsequent contributions of this thesis are outlined:

- Introduction of an end-to-end interactive queue management (QM) system by providing auction functionalities to jump queues when in urgent need of a charge
- Improve route planning through inclusion of reservations to guarantee charging
- Formalization of charging flows and the corresponding constraints to provide Quality of Service (QoS) for the implemented system

1.1 Motivation

Considering that the interests of several truck companies are different in a charging area, a general scenario can be formally modeled as a stochastic multiplayer game [44] in which both participants seek to maximize their benefits and reduce losses to achieve a Nash equilibrium [45] under practical considerations of real world constraints. As such, it is essential to design algorithms that can handle the stochastic nature of consumer arrivals and bidding behavior, which can account for factors including the urgency of consumers, the current state of the queue, and the bidding strategies employed by both parties. For example, when a consumer is in a hurry and may be willing to provide a higher bid to secure a quicker charge and jump an existing queue. On one side, the seller's benefit involves increasing the turnover of CS during peak times, and, on the other side, the consumer's benefit involves lower charging fees or bonuses in future recharges in response to the quick release of the station.

Although these motivations are based on similar economic thinking when comparing a private EV owner to a logistics company with an ET fleet, there are differences that need to be considered. In contrast to ETs, predictability includes a high amount of uncertainty when considering the nature of spontaneous usage of private EVs. Due to the regulations of driving times and the clear definition of the route [9], this uncertainty decreases. Additionally, the truck companies already plan the routes with possible stops, which allows to optimize these plans in terms of the economic efficiency. Hence, a bidding system for CS introduces an additional steering element when in urgent need of a charge to continue the delivery or even to increase profitability when on the side of the seller.

When aiming to improve the economic aspect of CS, the prerequisite of a trustworthy and robust system arises. The proposed system requires interaction between different users in two ways:

- **Passive** by waiting in the same queue or using the infrastructure at the same location
- **Active** by engaging with each other when negotiating CS or passing them to someone else

This introduces technical and social expectations of each user, which are based on the assumption of handling benevolent actors in the system process. However, since CS are open to the public without restrictions, all types of actors participate in the system, including those with malevolent intentions. These actors appear on both sides, sellers and consumers; therefore, the system needs to provide sufficient protection through effective countermeasures in order to fit the assumption of the participants following a set of predefined rules.

1.2 Problem Statement

In the discrete scenario of ETs, long charging times are accentuated due to the higher total capacity of the batteries [7] although an increase in the power outlet of chargers for trucks is still subject of ongoing research [5]. Predictions for the future forecast a continuous demand for freight delivery electrification [22], resulting in a higher frequency of use of the charging infrastructure and therefore increasing the potential for queues at the charging stations.

While increasing the number of CS or chargers in the respective CS could mitigate the risk of longer waiting times, it requires investments that target peak usages and in hindsight are not used in average scenarios. In contrast, as stated in Section 1.1, the efficiency of charging stations during peak times is aimed at improving the queue system. Taking this a step further leads to the question this work aims to answer: What adaptations does a queue management system need in order to be able to handle CS brokering for electric trucks?

1.3 Thesis Goals

Deriving from the described question in Section 1.2, the following goals are determined:

- **Analysis of background and related work**: A theoretical analysis of suitable algorithms is performed, which encompasses the brokering platforms based on the scope of game theory, including the literature on QM targeted to the scenario of the electric vehicle charging station. This results in an accurate mapping of literature on the topic, as well as a suitable collaborative algorithm applicable to the management of CS customers in a real-world scenario.

- **Design of a brokering platform including queue management**: Development of a brokering platform that implements QM. The platform will facilitate a brokering system for ET users, by integrating a dynamic brokerage that allows consumers to negotiate their charging spots. It will balance the interests of both platform users, who may accept high bids to skip the queue, and non-platform users, who follow the traditional first-come, first-served method.
- **Prototype Evaluation**: Evaluation of the prototype's effectiveness for predefined scenarios. This includes a detailed analysis of system performance, user engagement, and security aspects. The findings will inform strategies for real-world implementation and contribute to the overall enhancement of the EV charging infrastructure.

1.4 Methodology

The choice of the design for the envisioned system is based on a thorough study of current research. By analyzing and comparing existing solutions, a solid base is formed to combine suitable, already proven aspects while introducing new ones in this thesis. Deriving from these, requirements are finalized, which serve as a basis for an implementation-blueprint of an interactive QM system. The product of this experiment results in a prototype of the system, which is then verified with the requirements established previously.

1.5 Thesis Outline

Based on the methodology explained in Section 1.4, the structure of this thesis is as follows:

In Chapter 2, the core principles of QM and possible interaction with queues are presented and discussed. In addition, current research is presented and compared with each other, yielding insights on different approaches to mitigate different risks imposed by QM systems.

The approach presented in this work is then introduced in Chapter 3 by first defining the logical and technical requirements followed by explaining the assumptions made in the process of building the application.

Details about the implementation, such as the technical components of the system and the high-level architecture, are outlined in Chapter 4.

It is then compared to the defined requirements by conducting some experimental runs of different scenarios in Chapter 5. Additionally, the outcome will be discussed to outline elements that meet the requirements and those that need to be improved.

Ultimately, a summary of the findings is presented in Chapter 6.

Chapter 2

Fundamentals

In this Chapter, the fundamental knowledge required for this thesis is introduced. Section 2.1 explains the concept of queue management, auction schemes, and non-cooperative game theory. In addition, an overview of current charging station designs is provided, focusing on physical constraints and their implications on queue management systems.

Section 2.2 then discusses different applications of the core principles presented in current research. The results of these works are compared to the use case of this thesis.

2.1 Background

Although establishing a fair queue is one of the problems outlined in Section 1.1, it only highlights the output of a QM system. To shape this output, other given aspects of the whole ecosystem need to be considered. Therefore, the interplay between the constraints imposed by the environment in which the system acts and the different queuing principles are explained. One of these design factors is the intention of interactivity through the use of an auction system. The introduction of non-cooperative game theory allows one to discuss the optimal choice for the symbiosis of a QM system and an appropriate auction scheme.

2.1.1 Queue Management

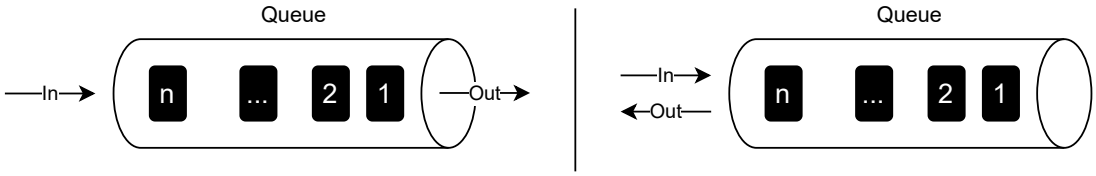
In computer science, the term "queue management" is related to approaches aimed at avoiding congestion and the resulting buffer overflow *e.g.*, when investigating TCP connections [29]. The Internet Engineering Task Force (IETF) issued recommendations for Active Queue Management (AQM), which discusses best practices to avoid congestion through QM [4]. In this case congestion describes the state of a queue, in which no further input can be handled and is dropped immediately. This affects two aspects of a system's QoS as inputs are handled with delay or not at all and by dropping input

data. Therefore, the sender needs to be informed about the requirement of resending the dropped parts again.

However, when users face a queue, they experience the allocation of the queuing system even before congestion appears. The perception of fairness correlates with the individual conception of an acceptable waiting time [27]. Thus, providing a queuing system in which the individual perceives the own waiting time as "fair" increases subjective satisfaction.

Here, the selection of queuing mechanisms fundamentally influences the perception of the implemented system, as it changes the above-mentioned waiting time, eventually pushing it above the individual acceptance limits. There exist different approaches for queuing mechanisms, such as *First In First Out* (FIFO) or *Last In First Out* (LIFO) as shown in Figure 2.1.

Figure 2.1: Concept of the queuing mechanisms *FIFO* (left) and *LIFO* (right)



FIFO: An element that first joins the queue will be processed first and eventually released. Hence, the arrival time determines which element is served in ascending order. The waiting time for an element e_n in this case is the processing time of each element $e_{n-1\ldots1}$ placed at the front. This mechanism is also known as "First Come First Serve"-principle.

LIFO: In contrast, using this mechanism, an element that joins the queue last will be processed and eventually released first. The waiting time for an element e_x in this case depends on the sum of the processing time of the elements $e_{n\ldots n-x+1}$ placed in front. The commonly known term for this type of queue is "Last Come First Serve". Counterintuitively, in certain scenarios where the queue is "invisible" to the participants, LIFO yields improved waiting times [25] with respect to a FIFO solution.

For both approaches, waiting times can be estimated with some margin of uncertainty, as the processing times may or may not be known. These calculations of expected waiting times are used in queuing theory, which aims at modeling and optimizing queuing systems [26]. To account for different characteristics of the queue, Kendall's notation was introduced with the following pattern [28]:

$$A/S/c \qquad \text{(Kendall's notation)}$$

- A Distribution of arrival intervals
- S Distribution of job service time
- c Number of servers

Thus, the notation $M/M/1$ describes a system where arrival times follow a Markovian process (Poisson or random) denoted by M. The service times are exponentially distributed and there is one server available to serve the queue's needs [38]. This notation

can also be extended to $A/S/c/K/N/D$ where K is the number of places in the queue, N is the population size, *i.e.*, where the customers come from, and D is the queuing discipline such as FIFO or LIFO. If these are not explicitly given, it is assumed that $K = \infty, N = \infty, D = FIFO$ [39].

The output of the corresponding probabilistic calculations yields insights into metrics such as average waiting times or average queue size. Considering the real-world scenario of the queue, the results are then incorporated as a base for the choice of the queuing system's design.

2.1.2 Auction Schemes

In the scenario in which vehicles requiring a charge are queuing for CS, the envisioned interactivity impacts the perspective of an individual user in the queue. As the value of each position in the queue increases, the closer it is to eventually be served, the higher the likelihood that other participants are interested in wanting to trade this favorable position.

However, this is not a bilateral trade, and other participants might want to opt-in and join the trade. Such a scenario is the initialization of an auction: A valuable good is presented *e.g.*, the guaranteed access to a CS. Those participants who are interested in the offer can place their bids until eventually, based on the auction's rules, a winner is declared.

The rules of the auction system can be freely chosen. However, there are three basic auction schemes that are fundamental for all the other variations: The English auction, the Dutch auction, and the first-price sealed-bid auction [32].

Table 2.1: Different types of auction schemes

Type	**Rounds**	**Transparency**	**Bidding procedure**
English	Open	Fully transparent, bidders always know best price and other bids	Starting with lowest price, ascending
Dutch	Open	Non-transparent, bidders only know current price	Starting with highest price, descending
First-price sealed-bid	Single-rounded	Non-transparent, only final outcome is known	Bids are secretly handed in

The **English auction** starts at a comparatively low price for a specific good and is increased in each round [32]. Each bidder decides individually if the raised price is acceptable and informs the auction leader about the decision to accept the new price. This goes on until eventually only one bidder is left, since everyone else is not interested in paying this price. A property of this type of auction is the continuous transparency, as each bidder knows the current best price. This is the most costly approach with respect to the amount of transactions needed until the auction is successfully closed.

In contrast, the **Dutch auction** starts at a high price and is lowered for each bidding round. The auction ends as soon as the final price is announced and the first bidder decides to accept it [11]. Thus, the information flow of other bids is non-existent, as the bidders need to rely on their own prediction about the intentions of others. As soon as one fellow participant decides to deviate from this prediction in a non-favorable way, the auction is already lost. Dutch auctions are known to be efficient in terms of transactions as only one bid is required to end the auction.

Lastly, the **first-price sealed-bid auction** uses a similar approach as it is used for the English auction, by declaring the highest price to be the winning price, but changes one crucial aspect: Bidders are not informed about the other bids [32]. Therefore, for only one round, the bidder needs to predict the behavior of the other participants. Eventually, the highest bid will win the auction, minimizing the interaction with the other participants to a single round. Thus, the amount of bids is lower compared to the English auction but higher than a Dutch auction.

The efficiency of the auction schemes impacts the waiting times of the queue management systems if queue positions are traded. If these trades are to be concluded as soon as possible, the interactions should be kept to a minimum. Additionally, in the English scheme, malicious participants could try to increase the price without even wanting to eventually pay this price for the auctioned good.

2.1.3 Non-cooperative Game Theory

Although the interaction is indirect, as each bidder only interacts with the auction leader, there exists an implicit interaction with the other participants. This situation is similar to this thesis scenario, where trucks queue at a charging station: There is no need to interact with the other vehicles in line, but all participate in the same system of a distribution of CS. Hence, each actor implicitly decides on a strategy to achieve the best possible outcome, which in this case corresponds to optimizing waiting times.

The decision-making process of this dilemma is investigated in *game-theory*, which models different scenarios of situations where a game participant either knows or tries to anticipate the strategies of fellow participants [17]. Specifically, in the queuing scenario, the *non-cooperative* aspect of game theory applies, as each vehicle acts independently [35].

By analyzing each strategy and contrasting each other, a map of different game outcomes is drawn. Nevertheless, the results must subsequently be verified against the specific needs of the participants, as although some results appear logically valid, they may lead to a less favorable outcome when compared to the participants' expectations. Such scenarios occur if the decision of the participants is logical but is based on wrong predictions. However, the intersection of these strategies is considered an *equilibrium* as the term itself does not pose any positive or negative implications on the overall product.

The Nash equilibrium describes a situation in which the *pure strategies* of the game's participants resolve in one or multiple situations in which neither participant wants to change their strategy once all information is known [35]. Conversely, this does not imply

that the equilibrium is the best outcome on an individual participant's level, but rather when combining the individual values of the solution and comparing them among other combined outcomes.

Mapped on the Thesis scenario, a Nash equilibrium is found for a situation in which a truck urgently needs an immediate charge, but due to a *FIFO* queue needs to wait its turn, but another truck currently charging is willing to cede the CS given that the compensation is sufficient.

2.2 Related Work

Although the charging station scenario appears in the existing literature, it is typically applied to private electric vehicles, as the required power delivery through electric batteries has not yet been resolved, as mentioned in Section 1.2 [5, 7]. Hence, in Table 2.2 the different use cases are compared to each other, indicating if the paper aimed to resolve queuing for private EVs, ETs, or classic Internal Combustion Engine (ICE) Trucks. Furthermore, the implementations of queues are investigated in order to understand the scenario assumed for the queues to appear in.

The possibilities for interaction with the queue in the proposed approaches are outlined. Finally, to distinguish between the main targets of these papers, a brief description of the target is given, in order to embed the dimensions mentioned above into a wider context in which the work is situated.

Taking into account the comparison Table 2.2 of current research, the ratio of papers about private EVs and ETs is unbalanced. This reflects the current situation in research, where it is uncommon to find papers on queue management for ETs. The applicability of the private EV scenario is given due to the similarities in the charging station areas, yet it differs in the charging times required and the ability to plan the route, *i.e.*, the specific point in time an ET will charge.

When comparing the queue types used in state-of-the-art research and queuing systems, it is observed that different approaches were investigated to model queuing for chargers at an EVCS. One approach is to shift the queuing decision in time and do it earlier than the actual arrival. This is done by a central authority (CA), which collects charging requests. From there, the approaches deviate, as the CA assigns a charging station [50] or the CA facilitates data collection to improve economic efficiency by applying dynamic pricing based on the received demand [6]. In contrast, a distributed coordination framework is applied to the ET use case, sharing the idea shown in [50] where real-time in-transit data is received [42]. This data is then formed as the basis for a prediction of an arrival time for different EVCS leading to an assignment of the vehicle to a specific charging station.

Taking the view level one step higher, the queue is not managed based on individual decisions, but rather is seen as a product of the infrastructure decisions the charging station operator made [47]. By choosing a queuing system with the number of servers defined as s_j, where s_j is the number of chargers at a specific charging station j, the cost calculation used is heavily dependent on local factors in a specific scenario.

Table 2.2: Comparison of current research

Work	Use Case	Queue Type	Queue Interaction	Target
[13]	Private EV	$M/M/c$	None	Improvement of user's experience by considering the whole time spent between requesting and plugging in the EV while assuming that an EV owner sends charging request to central authority.
[47]	Private EV	$M/M/s_j/N$	None	Considers differing amount of chargers between multiple CS by targeting to reduce infrastructure costs, distance and queue waiting costs through an optimized QM.
[30]	Private EV	$G/G/c/N$	None	Improving QM in case of an emergency scenario where a larger amount of vehicles spontaneously require charging.
[33]	Private EV	$M/M/s/C$	None	Reducing burden of fixed EVCS by dispatching portable EVCS, such that peaks can be smoothened.
[15]	Private EV	Feedback controlled express station management	None	Keeping the delay rate in a defined boundary, which is determined by each individual owner, to optimize costs.
[36]	Private EV	Deep Learning $NSGA-II$ algorithm	None	Estimate waiting times by using a Deep Neural Network and use the estimation to minimize these times as well as reducing costs.
[6]	Private EV	Publish/Subscribe mechanism	Partial queue interaction through subscription	Providing a protocol which allows to communicate via Road Side Units to a remote reservation service.
[8]	ICE Trucks	point-wise stationary approximation	None	Reducing truck emissions through QM with minimizing engine idling times and therefore optimize stationary waiting times.
[3]	Electric Trucks	distributed coordination framework	None	The coordination framework receives real-time data from trucks about their estimated arrival time and based on uncertainty factors the CS are assigned to optimize waiting times.
[42]	Electric Trucks	$M/G/c$	None	Evaluating an optimal placement of CS with a model using QM in order to mitigate the risk of trucks running out of battery as the capacity does not cover the required distances yet.

To relieve local pressure, the proposal of dispatching portable charging stations during peak times was presented [33]. This requires a dynamic adaptation to real-world circumstances as demands appear chronically but also spontaneously. Such a spontaneous event, namely an emergency scenario, is investigated in [30]. As the queue type $G/G/c/N$ indicates with general *i.e.*, independent arrival times and independent service times, this scenario is particularly challenging to model as the situation does not allow to choose freely between EVCS.

Due to the challenges posed by the stochastic characteristic of the classic QM, different methodologies were introduced. Using a deep learning algorithm that learns from a virtual queuing system, it was shown that in complex queuing scenarios, the traditional queuing theory approach could be enhanced [36]. However, the QM predictions were approached differently by modifying the waiting areas for waiting vehicles. Although it was done for ICE Trucks, the proposed model uses a similar approach to a leaky bucket by regulating the flow of vehicles already at the beginning [12].

The principle of using a "bouncer" for the waiting area allows the system to closely follow the flows of each vehicle in the queue, and this knowledge is used to improve waiting times or delay as in [15]. In addition, different service classes were introduced to let the EV owner decide between different speeds of charging. While continuous feedback flow is costly, it improves the experience compared to a one-time decision made by a centralized system.

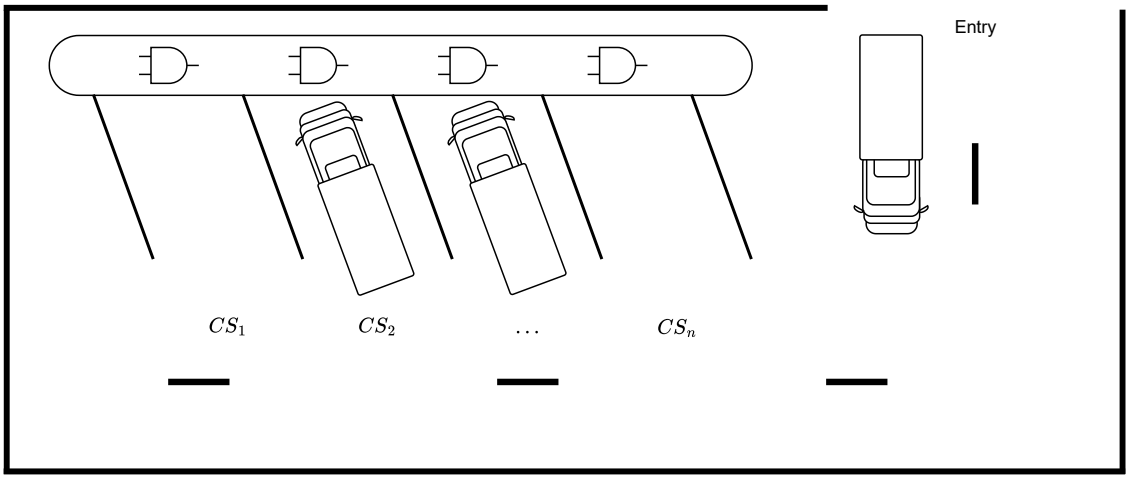


Figure 2.2: Simplified design of charging stations

From the third column of Table 2.2, a lack of interaction with queues is observed. In the EVCS scenario, interaction denotes the ability of a queue's user to influence the decision of the spot assignment. This is done either by "jumping" the queue and being assigned a spot upon arrival even though there are others waiting or by being able to block a spot for a specific time, so that no one else will be able to use it during this period. One of the only queue interactions observed in current research is a system in which EVs select a CS. As soon as they are approaching it, a reservation is sent to the respective CS [6]. These reservations are sent to the system through middle boxes, so-called roadside units, which enable the data transmission between EVs and EVCS.

The current infrastructure of charging stations is based on the common parking lot layout [18, 31]: Dedicated and spatially delimited areas are served by the corresponding charging station as shown in Figure 2.2. Since charging stations are either replacing old fuel stations or sharing the same space, there is no dedicated waiting area as this was not necessary (anymore) for classic fuel stations or a parking lot. For a reservation system proposed in [6] to work, charging stations need to be equipped with technical infrastructure to allow the matching of reservations and the corresponding usage of chargers.

While current research focuses on QM for EVCS, it does not consider the specific aspects for Electric Truck Charging Stations (ETCS). Although these approaches are applicable for the ET scenario as well, they do not suffice to cover the needs of truck companies. As they plan the route to gain efficiency in the delivery process, the charging is ideally used to connect breaks imposed by regulations with the need of having sufficient energy to cover the distance. Therefore, it is not applicable to assume that QM systems at the ETCS are able to provide CS in a beneficial period of time as it is done for EV. Hence, the QM systems need to provide a stronger guarantee to the truck companies in order to facilitate planning, while avoiding to strongly impact the already existing spontaneous flow.

Chapter 3

SIRQ's Design

In this Chapter, the design of the System for Interactive Reservation and Queue Management (SIRQ) is introduced. SIRQ presents a solution for scenarios that involve high utilization of electric vehicle charging, more specifically electric trucks, since these entail well-defined and time-constrained route planning. To embed the scenario, the assumptions made are discussed in Section 3.1. Combining the prototype blueprint with the assumptions, the logical requirements are outlined in Section 3.2, while Section 3.3 introduces the technical requirements. Deriving from the outlined requirements, an end-to-end model charging flow is presented in Section 3.4. The proposed interactions with the QM system are shown in Sections 3.5 and 3.6. These interactions impose some additional flows for the spot-assignment, a subpart of the charging flow. The changes are explained in Section 3.7. Based on the introduced charging flow and the targeted interactions, the environment in which the system is acting is defined.

3.1 Assumptions

The chosen scenario implicitly considers specific environmental and behavioral aspects as given. Therefore, these assumptions are introduced and discussed to incorporate a well-defined scenario.

To ensure the interaction between the different actors at an ETCS, they need to connect to SIRQ. In the chosen scenario, this is required, as reservations need to be transmitted and notifications require the ET companies to be reachable. The route planning of trucks is usually done end-to-end in a centralized manner [1]. Hence, it is not the driver who decides where to charge, but the internal truck dispatching service. Consequently, the participation of each ET company through a dedicated login is stated in Assumption *A1*. In addition, this connection is always stable and reliable to fulfill *A1* at any given time. As the goals of this thesis do not include the security aspect, the connection is assumed to be secure in the investigated scenario.

The queues appearing around ETCS impose spatial prerequisites: A dedicated waiting area allows ETs to stay close to the CS and be notified about new spot assignments. This

is defined in Assumption *A2* and is the prerequisite for all ETCS considered in the chosen scenario.

Furthermore, the waiting area described in Assumption *A2* does not restrict any ET to enter or leave the queue, whereas Assumption *A3* specifies the physical layout of these areas. By allowing ETs to enter or leave queues at any time, abandonment of the queue by either leaving the ETCS or jumping the queue due to a won auction is possible. This enhances the experience for ETs, as it provides a tool to actively manage individual acceptable waiting times.

Consequently, Assumption *A4* ensures this possibility of jumping the queue by stating that there are no physical constraints to bypass the queue. This applies for both scenarios: ETs which were already in the queue and through winning an auction now are assigned a CS or ETs with a reservation, which do not need to enter the queue at all.

To increase flexibility of the ETCS, reservations are available for any physical CS. Assumption *A5* ensures this by requiring equal accessibility. The proposed queue interaction through auctions would otherwise not work, as bias is introduced due to differing properties of the CS.

The ETCS are required to keep track of the different ETs, which is given through Assumption *A6*. This property supports the proposed queue interaction in SIRQ, by providing data about the state of each ET which is required for the decision-making of the QM.

To ensure successful auctions, the willingness of truck companies to trade their occupied CS is required. Hence, this property is stated in assumption *A7*, as the lack of this willingness blocks the auction mechanism if no or exorbitant prices are offered when requested.

Table 3.1: Assumptions for SIRQ's Design

Number	Assumption
A1	Each ET company uses a central login with a stable and secure connection
A2	The ETCS have a waiting area for ETs in the queue
A3	It is always possible to enter or leave a queue
A4	It is physically possible to bypass the queue without any further restrictions
A5	CS are openly designed *i.e.*, no physical barriers for reservable spots
A6	ETCS are able to track vehicles throughout the area
A7	Truck companies are willing to trade CS to reasonable prices

3.2 Logical Requirements

Combining the chosen scenario and the goals defined, logical requirements are derived to provide clear guidelines for the expected minimum functionalities of SIRQ.

Although interaction with the queue is offered, it is required to remain optional to ensure spontaneous charging. In addition, queues are not always existent and hence, the base

flow of arriving at the ETCS and seamlessly being assigned a CS should not be impacted by the added functionalities. The requirement *LR1* includes any queue interaction *i.e.*, auctions and reservations, since reservations interact with the queue if it exists due to the possibility to jump it. Accordingly, the requirement *LR2* states that the interaction functionality is used situational.

Taking into account real-world behavior, every ET should be able to leave the ETCS at any time as it is defined in Requirement *LR3*. This implicitly requires SIRQ to handle dropouts of flows at any possible step. It should therefore be possible to enter a queue and leave it, while SIRQ continuously adjusts it based on the current state. Similarly, abandoning the ETCS after being assigned a spot should not cause any blocking of an empty CS while other ETs are waiting.

Table 3.2: Logical Requirements for SIRQ's Design

Number	**Logical Requirement**
LR1	ETs can be charged without the need of interacting with the queue
LR2	ETs can interact with the queue *i.e.*, to eventually jump the queue
LR3	ETs can terminate the charging flow whenever they desire

3.3 Technical Requirements

Publicly used multi-user systems that power essential infrastructure are required to fulfill technical properties, which increase their robustness to provide the expected QoS. These properties serve as a foundation to achieve the defined logical requirements of the brokering platform targeted in the thesis goals. As SIRQ's field of use is always-on *i.e.*, there is no planned downtime as ETs are operated at every time throughout a day. Sudden shutdowns of such systems can be reduced to a minimum, but not ruled out. Therefore, requirement *TR1* formulates the property of a system to recover the last known state even though an interruption of the service did occur.

Apart from the devices that provide data on the situation at an ETCS, SIRQ does not require any additional specific infrastructure. However, this requires the system to be flexible and use common standards to connect different devices or network topologies as stated in requirement *TR2*. Since ETCS and the corresponding infrastructure are already existing, SIRQ needs to be able to connect seamlessly. Connecting charging stations is an already existing model, as EVCS provide information about the amount and type of CS to the public through the onboard system or third-party services [24]. Moreover, the users are heterogeneous with different devices and, therefore, different security approaches are in place. Furthermore, it is not viable to require any whitelisting of SIRQ on the user devices.

The interaction between different users requires SIRQ to ensure confidentiality of the transmitted data. This is due to the sensitivity in terms of economic competition, since having access to the competitor's charging data yields insights on their expenses and reveals behavioral patterns. Furthermore, if the data about the queue interaction is shared

with a competitor, hostile strategies could emerge for the queue interaction mechanisms. This cannibalizes the functionalities of the system the charging data is coming from.

Similarly, data integrity needs to be ensured as mentioned in requirement *TR3*. It should not be possible to change the data on the elapsed charging time, as this causes financial damage to the truck companies. Additionally, queue interaction data is a target to be changed, as it results in an advantage over other participants in the system.

Due to above mentioned always-on scenario, users expect that the system will always respond. Consequently, the last part of Requirement *TR3* states that SIRQ should be available when a user needs it. Furthermore, the response time of the system is added to the waiting time which is correlated to the customer satisfaction. Hence, the lower the response time, the lower the waiting time will be.

Table 3.3: Technical Requirements for SIRQ's Design

Number	**Technical Requirement**
TR1	SIRQ is able to recover from spontaneous shutdowns
TR2	Common interfaces are used to allow for infrastructural flexibility
TR3	SIRQ ensures that the CIA principle is respected

3.4 Charging Flow

To satisfy the need for a charge, an ET owner goes through several stages of a procedure, which in the following Sections the term "*charging flow*" is used for.

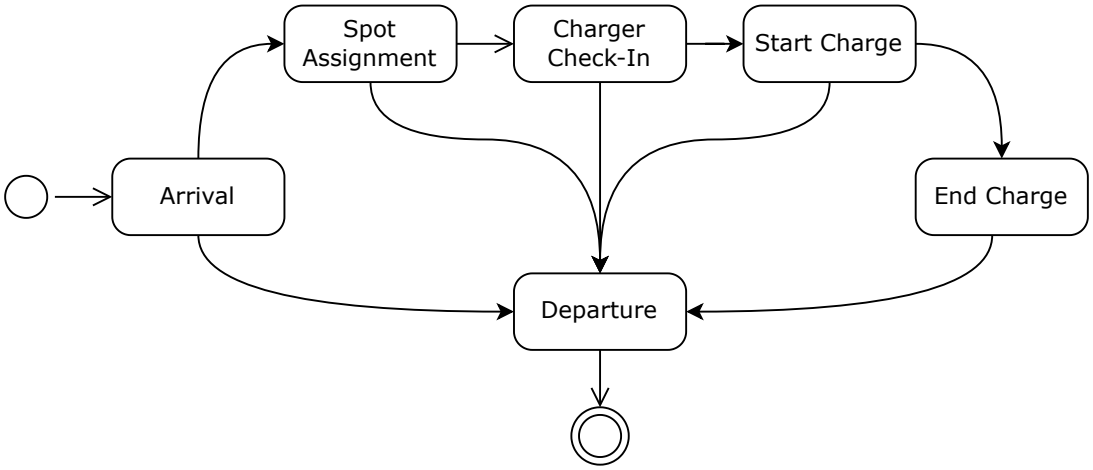
The first interaction with an ETCS is the entry into the area where the chargers are located. Mapping this to the scenario in which potential queues are expected, the entry must be recorded in order to keep track of all ETs present in the EVCS by storing a distinctive **arrival timestamp**. Subsequently, the system tries to match the arrivals with the available chargers. If no chargers are available, no spots will be assigned and the ETs will need to wait until a charger is freed. Hence, a queue is formed upon the arrival of additional ETs without free chargers.

Although queues are typically associated with fixed positions in a queue, *e.g.*, "position n in queue", this is not explicitly required for the management of queues. Since ET behavior is unpredictable and spontaneous, ETs leaving the queue is possible at all stages, the implicit queue reduces the complexity of reassigning positions while improving the robustness. Assuming the servicing process of a system will finish jobs in a finite period, the servicing spots are eventually freed up, and thus at least one position in the queue changes. In SIRQ, the queue is considered to be a $M/M/c$ queue, where c is the total amount of functional chargers in an ETCS. This holds for all periods in which there is no queue interaction that interferes with the available number of chargers, which is further discussed in Section 3.5 and Section 3.6.

The spot assignment is based on the *FIFO* principle introduced in Section 2.1.1: For all ETs in the queue the arrival timestamp is sorted in ascending order, such that the ET with

the earliest timestamp is served first. This type of queue is chosen because the charging levels of the ETs are not considered. If they were, the estimated charging time could be taken into account, and hence the efficiency of the system increases with respect to the overall waiting time but also in terms of the throughput *i.e.*, ETs charged in an arbitrary period of time similar to the service classes introduced in [15] discussed in Section 2.2. In addition, perceived queue fairness increases with a *FIFO* system if the queue is "visible" to the participants [25]. The architecture of ETCS is by design open and therefore every ET in the queue sees how the spot assignment takes place. Upon spot assignment, ETs have to shift from their current location to the assigned CS, which requires a **spot assignment timestamp** to be stored, allowing the system to immediately remove these ETs from the queue although they have not reached the charging spot yet.

Figure 3.1: State diagram of the Charging Flow

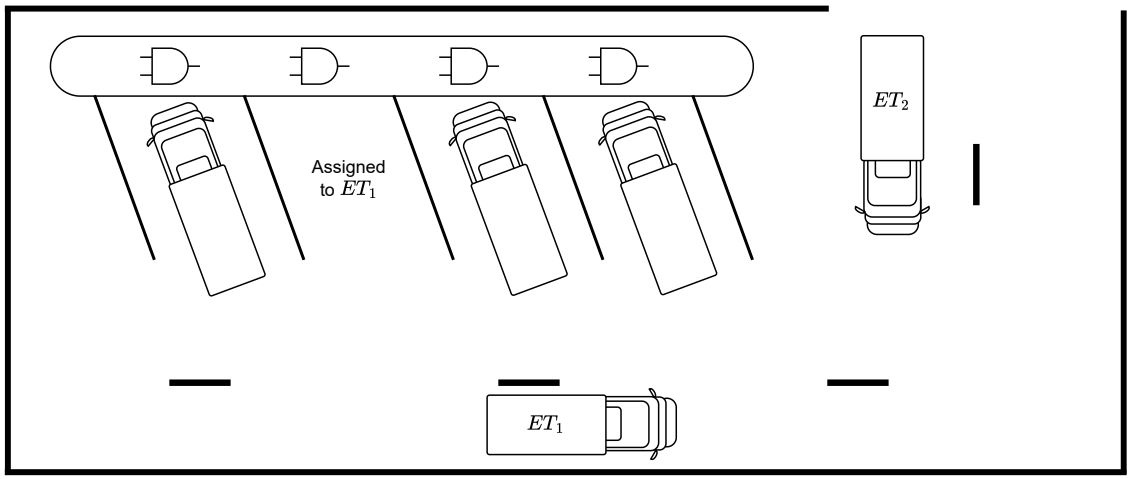


As soon as the ET arrives at the assigned CS, the **charger check-in timestamp** is set. This granularity of tracking has two advantages: First, SIRQ closely follows ETs through a distinguishable timestamp between arriving at the correct CS and *e.g.*, a plug failure. In addition, the cost of storing such a timestamp is relatively low compared to the flexibility it provides for further checks about the position of a system's user. Second, considering the accounting aspect, the blocking of a CS is charged differently, if blocking is charged at all, than the actual power consumption to charge the ET. Hence, the **charging start timestamp** is set when the plug is connected to the ET and the charging started. Analogously, the **charging end timestamp** is set as soon as the plug is removed or no current flows anymore. Upon receiving a charging end timestamp, the event of a freed CS is triggered. The system assigns a new spot to new arrivals or matches it with an ET waiting in the queue according to the rules mentioned above.

Ultimately, the ET moves from the previously used CS to the exit of the ETCS. To conclude the flow, setting the **departure timestamp** marks the end of the procedure as depicted in Figure 3.1. The departure is a generally valid next step in the charging flow, which avoids the procedure to remain stuck in a non-final state. This ensures that the servers will finish jobs in a finite period and hence, spots are freed eventually. If this is not given, the ETs in the queue suffer from *starvation* that describes a state of a system in which the waiting ETs are never served. An example is shown in Figure 3.2, which

shows a scenario where ET_1 never uses the assigned spot and therefore does not meet the intended charging flow. Such a scenario needs to be resolved by using the departure timestamp to eventually free the unused, assigned CS.

Figure 3.2: Starvation scenario without countermeasures



The flow mentioned above considers a scenario in which there is no interaction with the queue; the ETs arrive at the ETCS and either a spot is assigned or they wait until a charging spot is free. Nevertheless, this design of the charging flow serves as the baseline in the following sections.

3.5 Reservations

If an ET needs to ensure that a CS is available, there are two ways in SIRQ to achieve this:

1. The need is already known when planning the route and, therefore, can be done by reserving the specific period at a desired ETCS;
2. The need arises spontaneously and requires an auction to jump the queue.

The latter possibility includes the scenario in which an ET decides upon arrival at the ETCS that the queue is too long, and instead of joining it, a CS in use is released for the auction starter through a successful auction.

The **reservation** mechanism uses the classic approach of defining a time interval for which a CS needs to be exclusively available for a specific ET [37]. The purpose of the reservations is to provide truck companies with the ability to plan the route according to the availability of the CS at the ETCS for the desired time. However, this introduces new possibilities for SIRQ to allow users to suffer from starvation. To avoid this, the ETCS provider defines a maximum number of reservable spots, which is bounded by $r_i \in [0, c_i]$, where r_i

is the maximum amount of CS available at an arbitrary charging station i to be reserved and $\{c_i \in \mathbb{N} \mid c_i > 0\}$ is the amount of CS at an arbitrary ETCS i. If this property is not respected, the waiting time for ETs in the queue increases and hence the likelihood of reaching an unacceptable length of waiting time grows. Therefore, reservations are required to be seen in the context of the ETCS, which allows SIRQ to preemptively identify potential violations for certain periods as soon as a new reservation is received.

When a truck company successfully reserved a CS for a specific ETCS, SIRQ reacts accordingly upon arrival of the corresponding reservation holder. This implies that the ET is immediately assigned an available spot although other ETs without reservation are in the queue. Similarly to the implicit queue, SIRQ does not explicitly assign a CS to a reservation. Upon each assignment of a CS, not only is the current availability verified but also the future availability. However, to fulfill this requirement, a definition of a maximum charging time needs to be defined, which also implicitly sets a higher bound to the maximum reservation time of a CS.

This preceding availability check requires a time constraint for the latest possible reservation. Without this, the reservation spot is not guaranteed to be available at the start of the reservation interval, as the reservations could be made arbitrarily close to the start time. There are possibilities to circumvent this *e.g.*, by enhancing the pre-check done upon receiving a reservation with a check that treats CS for which the current charging time is still in the guaranteed minimum interval as blocked. However, if no minimum interval is defined between reservation time and reservation start time the following exploit is possible: Arriving ETs could try to jump the queue by reserving upon arrival at the ETCS, which cannibalizes the need for the auctioning system, that is, until the maximum of reservable spots is reached. Such a design causes ambiguity, as there is no deterministic rule when to use one of the two services and the user has to try which way to jump the queue is working at this moment.

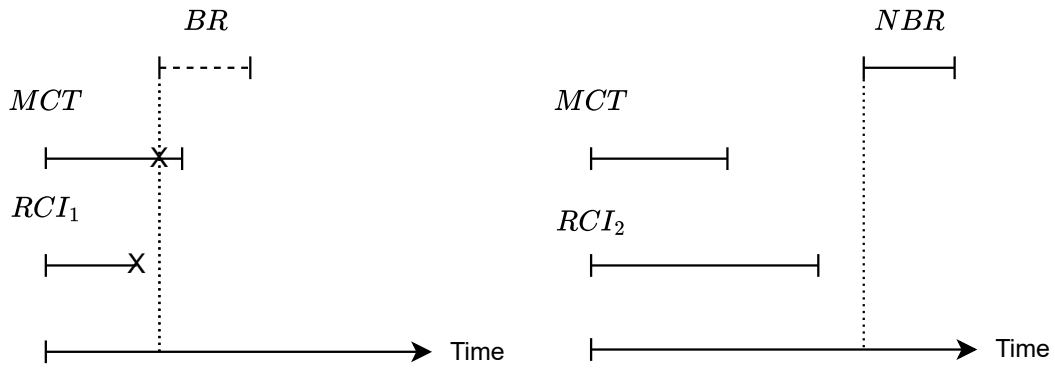
SIRQ allows for a distinctive setup of maximum charging time restrictions for reservations and non-reservations. This granularity provides flexibility during peak periods, as either one of these two charging flows is strategically favored. To provide an arbitrary illustration: If the ETCS provider decides to ask for a reservation fee and due to the reservation being able to already purchase the required energy beforehand, a longer maximum charging time could be used as an incentive for truck companies to reserve. Since this example is arbitrarily chosen, any or no favoring of a certain charging flow is possible.

Consequently, the maximum boundaries need to be enforced if the situation at the ETCS requires it. This is, if there is an existing queue due to no available CS including the previously mentioned blocked CS for upcoming reservations. In such a scenario, SIRQ finishes the charging for these CS which are exceeding the corresponding maximum boundaries defined by the ETCS provider. Assuming that there are no other incentives to prevent ETs from exceeding the maximum charging time, it is not required to enforce these limits proactively if no one is blocked by exceeding them. In contrast, a minimum guaranteed charging time is also provided to improve the customer journey and perceived fairness by the user. Similarly to the maximum bound, this is to be defined by the ETCS provider. The enforcement is done by setting the end charge timestamp for those ETs charging at that point and not respecting the maximum limit.

The choice of the minimum charging time influences the above-mentioned check during the spot assignment, which ensures that spots will be free at the latest when the reservation start time is reached. This interval should be at least of the length of the minimum charging time in order to prevent violation of the guaranteed charging interval. In Figure 3.3 this check is illustrated with the start of the interval being the beginning of a spot assignment. There are two existing reservations: BR and NBR.

If the reservation check interval RCI_1 is chosen to be shorter than the minimum charging time MCT the minimum is not guaranteed. This would cause the reservation BR to end the charging before the end of the MCT is reached. Consequently, RCI_2 is chosen to be at least as long as MCT and therefore does not violate the guaranteed minimum charging time. However, since the starting time of the blocking reservation BR is in the reservation check interval, the spot cannot be assigned, assuming that there is no other free spot available. If only a non-blocking reservation NBR was existing, the spot could be assigned as the reservation starts after the check interval ends.

Figure 3.3: Spot Assignment with Check for Reservations



Lastly, a minimum reservation interval can be defined to ensure that the reservation checks do not preemptively block spontaneous spot assignments for reservations lasting only a few minutes. This is similar to the restaurant scenario in which a guest wants to reserve a table for only having a drink, which is usually rejected as the profit from other guests having a full meal is higher. However, typically these guests are then informed to just spontaneously arrive, as there could be free tables or free seats at the bar.

The same approach applies to the ETCS scenario: If an ET wants to charge only for a short amount of time charging is still possible, but it requires patience if there is a queue. Nevertheless, to reduce the waiting time an auction for a CS could be initiated.

3.6 Auctions

While reservations introduce a mechanism to avoid queues if the charging period is known early, SIRQ introduces the possibility for using auctioning models to negotiate immediately over CS. Consequently, in the scenario in which an ET arrives at the ETCS and faces a queue because all CS are occupied, the ET could issue a request to trade a currently occupied CS and upon success is able to jump the queue. To accelerate the auction, the request either includes a signal indicating that every price is accepted or an upper bound *i.e.*, a maximum accepted price is defined. Although "every" price could be accepted, SIRQ still allows to define an upper bound to avoid any extortion in the auction.

Similarly to reservations, the available spots to consider are defined first. For auctions, the constraints are less strict, since each ET currently charging decides individually on the willingness to release the occupied CS. Consequently, the check for the limits of the minimum and maximum charging times is omitted. However, the check for future reservations is still carried out as the guaranteed minimum charge time introduces a potential violation of the reservation premise stated in Section 3.5. This violation occurs if a CS is traded, which is designated to be freed for a reservation in the defined check-interval for future reservations.

After the exclusion of the reserved spots, the auction starts by notifying the corresponding ETs. The auction is following the *first-price-sealed-bid* principle. Since the target of the auction is to still offer an attractive charge price for the ET starting the auction, the winning price is the lowest offer. Therefore, the *first-price-sealed-bid* is chosen, as the auction participants are bidding only one round without disclosing their bid. This prevents strategies of price optimization where the lowest offer is adjusted to marginally beat the second lowest price. In addition, the maximum acceptable price defined is not disclosed either, to avoid the bids to be adjusted to this height.

Upon submitting a bid, there is no positive or negative response shown to the bidder. The auction is done silently by SIRQ and only discloses the result to the auction starter and, if successful, to the "winner" of the auction. This is required as the spot will be assigned the same way as it is done for reservations: The end charging timestamp ends the charging session by setting the charging end timestamp for the auction winner and assigns this CS to the auction starter.

With the introduction of reservations and the finite character of these, another good to trade emerges: The reservation itself. SIRQ in addition, allows a user to request an auction for reservations too. This covers the scenario in which there are no reservable spots left at a ETCS, but a truck company still wants to reserve despite having to pay a surcharge *i.e.*, the auction price. To avoid the need of requesting many users to decide if they want to enter the auction, the period of the request is required to exactly match the start and end time of an existing reservation.

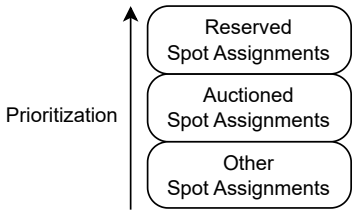
Lacking this constraint, overlaps could be possible and consequently a complex auction needs to be implemented, which aggregates offers for different periods to be matched. However, this increases the amount of participants being notified about auctions and therefore the dependency on user engagement grows as well as the danger of overwhelming

users. Eventually such a system could lead to users reject using the specific feature or the system as a whole.

3.7 Interaction Enhanced Spot Assignment

The introduction of the reservation and auction mechanism requires adjustments to the spot assignment, when either all of the CS are occupied or a queue is existing. Instead of assigning a charging spot to the first ET in the queue, the reservations and auction winners should be considered. SIRQ prioritizes reservations first, followed by spots reserved through auctions and finally all other assignments *i.e.*, for ETs spontaneously requiring a charge without wanting to start an auction or having a reservation.

Figure 3.4: Prioritization Hierarchy of Spot Assignment



This prioritization is shown in Figure 3.4 and is based on the economic incentives of the ETCS provider. Reservations improve energy allocation planning, supporting flattening of charging costs as energy is not bought on demand during expensive peak times. Therefore, the reservation flow is the most privileged flow and served first. After reservations have been settled, the CS that are reserved through an auction trade are assigned. In SIRQ, a specific interval is used to keep these auctioned spots reserved. The length of this interval is arbitrarily defined and serves the purpose of not blocking these spots forever and incentivizing the auction starters to not block charging spots even though they are not present at the ETCS. As blocking increases the waiting time for ETs in the queue, SIRQ provides tools to prevent it.

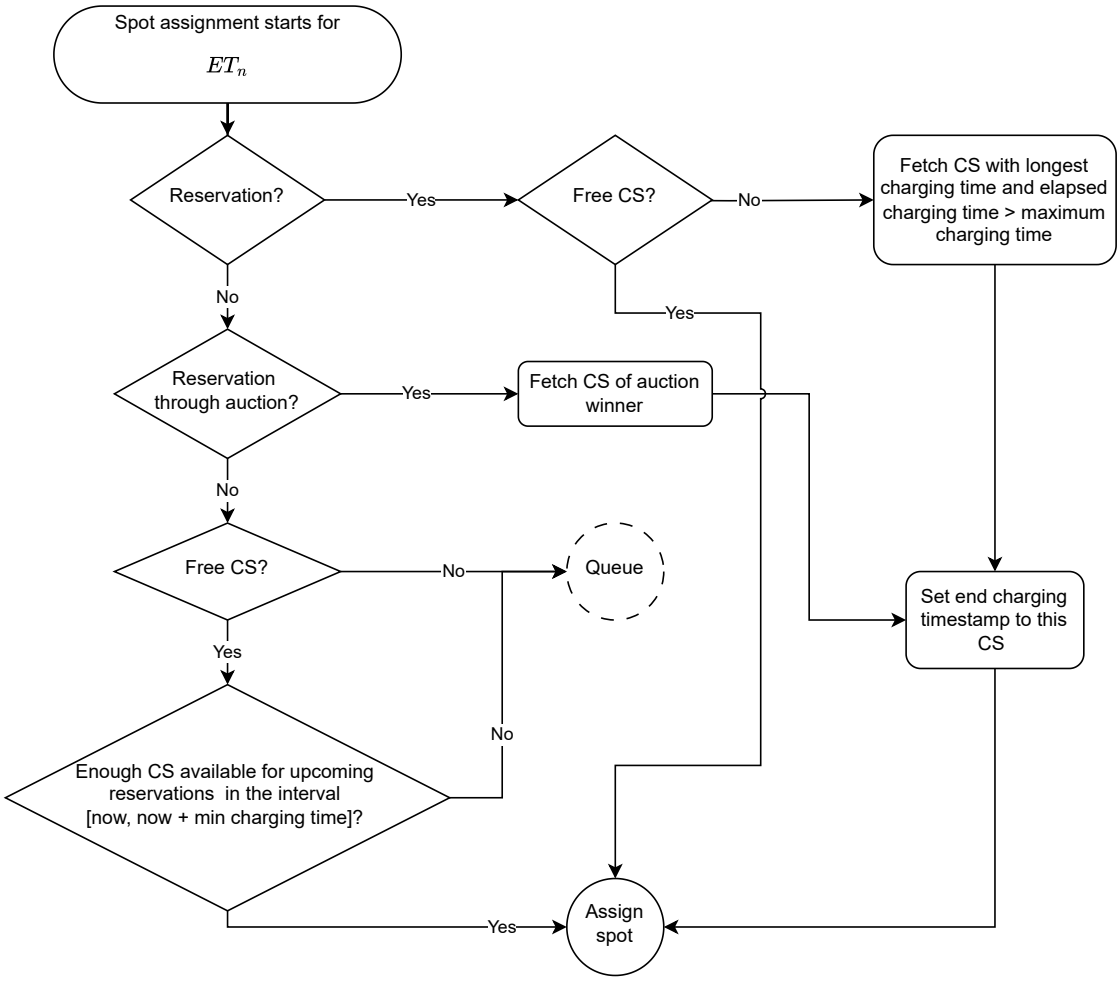
The combination of all the prioritization and control mechanisms which ensure that these properties are not violated is shown in Figure 3.5. Although comprehensive flexibility is added with these two interaction functionalities, the spot assignment algorithm remains relatively simple, as only three new distinctive decision points are added.

Assuming that there is no queue, at least one free CS and the spot assignment is done for a specific ET without a reservation, the first check will be negative. Subsequently, SIRQ compares the CS which were subject of an auction but are not yet assigned with the received license plate number of the ET. In this case, as there is no queue, the ET does not need to get a CS through an auction.

Finally, the available spots are fetched to compare these with the upcoming reservations. As explained in Section 3.5, the check interval is limited to at least the defined length of

the minimum charging time. Provided that there is a sufficient amount of available CS during this time period, the ET is allocated one of the open slots.

Figure 3.5: Spot Assignment Including Auction and Reservations



In case the arriving ET reserved a spot, it is by design ensured that at least one spot is available. However, as the minimum charging time does not restrict the duration of the charging session only guarantees a certain period, the CS could still be occupied. This increases the efficiency of the system as the capacity is fully utilized. Consequently, SIRQ assesses which of the spontaneous charging sessions exceeded the minimum charging time and charged for the longest time. Once this is determined, the corresponding charging session will be ended by setting a charging end timestamp and the spot is available to be assigned to the ET arriving with the reservation.

Finally, if an ET decided to jump the queue by starting an auction, the corresponding CS of the auction "winner" is first retrieved from the auction data in order to assign this spot to the ET which started the auction. The timestamp marking the end of the charging is set either upon the arrival of the auction initiator at the ETCS, or, if the ET has already arrived, the change of CS is executed immediately.

Chapter 4

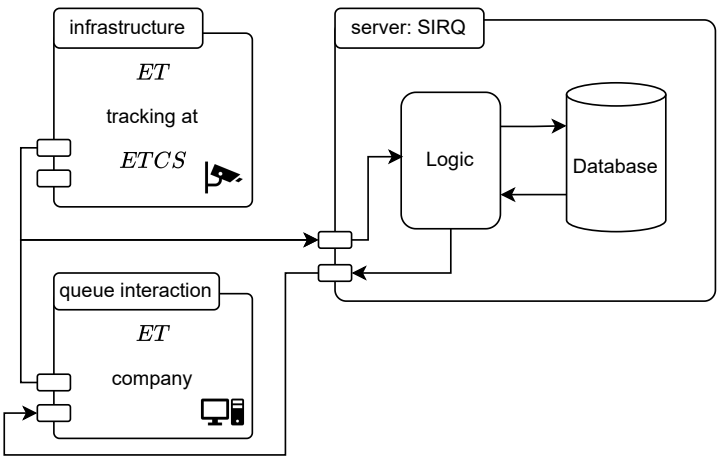
Implementation

This Chapter presents the prototype of an implementation of SIRQ. Section 4.1 provides an overview of the system's topology. The charging flow as a subpart of the architecture is discussed in Section 4.2. Subsequently, the features that surround and support the core functionalities are presented in Section 4.3. Finally, the implementation of the queue interaction procedures is described in Sections 4.4 and 4.5.

4.1 Architecture

Based on the design introduced in Chapter 3, three high-level components are implemented as an abstraction of the real world scenario. First, the infrastructure to track ET at the ETCS is modeled in order to provide data that is used to control the charging flow. Second, to ensure that the technical requirement *TR2* is met, a proof of concept is provided which showcases the simplicity of connecting an ET company. This connection is used to let the truck companies interact with the queue either through a reservation or an auction.

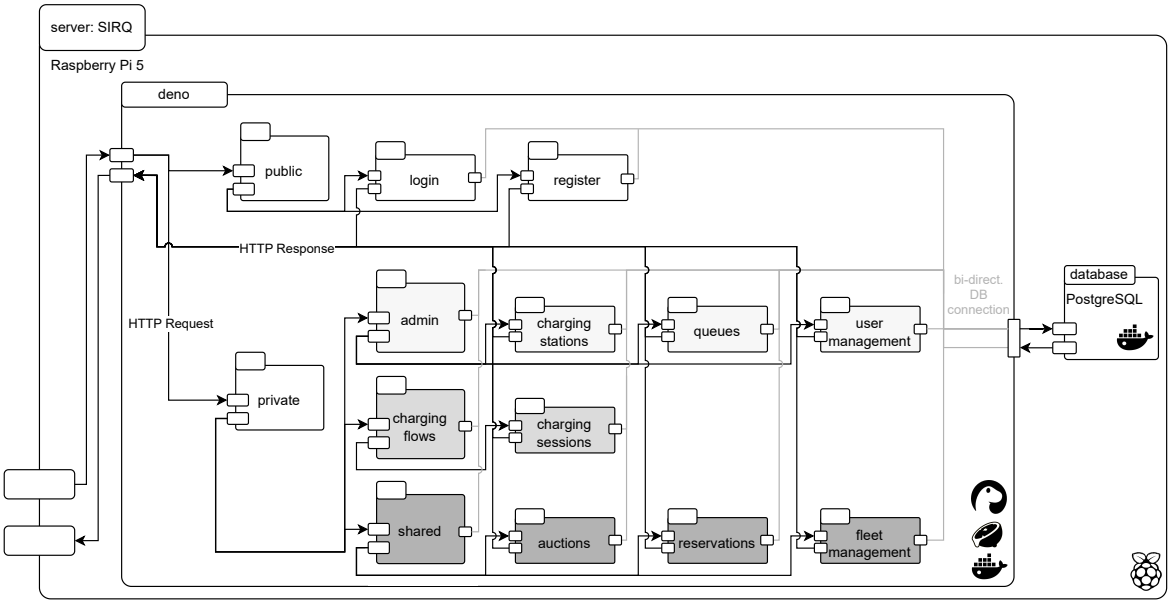
Figure 4.1: High level architecture



These two external parts are connected through an Application Programming Interface (API) to the server running SIRQ, the core of the ecosystem. It is divided into a logical part which ensures the incorporation of the aspects described in the logical requirements in Section 3.2 and the database (DB). As depicted, each request from the outside first needs to go through the logic part which includes the verification.

In addition, by keeping the database behind the logic acting as a gatekeeper, the requirement *TR3* in terms of integrity can be partially fulfilled. The connection is not encrypted in the prototype, as this is dependent on the existing infrastructure, both on the ETCS part and the truck company side. However, the implementation allows common protocols to be included such as TLS/SSL without any further changes to the logic. Furthermore, this design improves the availability, as the logic part manages the connections to the DB. This ability to handle DB connections with any desired prioritization logic is an additional aspect of improvement to provide a specific QoS. The database container used the Postgres Docker image [10], while SIRQ was implemented in a container running a Deno image [20] with the Fresh Framework [16]. Deno, combined with the full stack framework Fresh provides Representational State Transfer (REST) API integration out of the box and hence is working with HTTP requests. This fulfills the requirement *TR2* as HTTP is a commonly integrated protocol and does not require any additional implementations to work with existing devices.

Figure 4.2: Component view of the SIRQ server



As illustrated in Figure 4.2, the access to use SIRQ is restricted and needs to initially be authenticated. This is done by sending a `POST` request to the `register` endpoint. After a successful registration, the `POST login` endpoint authenticates the given credentials. The response contains a JSON Web Token (JWT) which encodes a unique company ID, represented as a version 4 universally unique identifier (UUID). This encryption is not considered cryptographically secure, but due to the assumption *A1* the bearer token is considered to be a placeholder of a secure authentication token.

API requests are routed through middleware, which is organized as layers on top or in between the different interface levels. On the top route *i.e.*, the `private` route, the bearer token sent as a cookie with the request is verified and decoded. This design allows messages to be intercepted before entering the logic of the requested endpoint. Hence, the underlying business logic can expect that the passed user is permitted to access the requested data. This is how the confidentiality of requirement *TR3* is ensured. Based on this design, other implementations of mechanisms to support the CIA principle can be implemented without affecting the actual business logic.

The SIRQ-specific architecture shown in Figure 4.2 reveals a Role-Based Access Control (RBAC) [14] based on the three flows:

- `admin`
- `charging-flows`
- `shared`

These are abstractions of the real-world roles which were introduced in Chapter 3. The `admin` role maps to the ETCS provider, the `charging-flows` role instantiates the infrastructural part of an ETCS discussed in Section 4.3 and the `shared` flow implicitly implements the truck company user. However, since the `admin` is expected to share the same rights as a company user, these endpoints are not exclusive to one role.

Each of the endpoints persists data in the DB as the connections to the DB container show. This is the technical foundation of requirement *TR1*, as the state of the system can be restored in the event of sudden failure or shutdown. As this system is used concurrently by different users, a single DB connection causes lockouts of multiple concurrent requests. To resolve this, a connection pool is established which offers concurrent DB connections with different contexts. The connections are set up lazily, which reduces setup time by reusing already available connections rather than tearing them down and setting up anew.

```
1  // Create a database pool with 20 connections that are lazily
       established
2  import { Pool } from "https://deno.land/x/postgres@v0.19.3/mod.ts";
3  export const sqlPool = new Pool(
4    {
5      user: databaseUser,
6      password: databaseScrt,
7      hostname: databaseHost,
8      database: databaseName,
9    },
10   20, // connections
11   true // lazy?
12 );
```

Listing 4.1: DB connection pool setup

4.2 Charging Flow

The charging flow is implemented using an event-driven approach. These events are stored atomically as timestamps for a unique charging session in the table `charging_flows`, which allows recreation of an application state at any time, and hence forms the foundation of the requirement *TR1*. Instead of requiring SIRQ to actively maintain all states, it should rather use the incoming events, such as charging flow updates and auction or reservation requests as a trigger to first serve the incoming request and after that settling other open flows or auctions. However, this was not entirely fulfilling requirement *LR3* as ET could leave after being assigned a spot. Assuming that at least one ET is in the queue and SIRQ did not capture the departure timestamp of the ET abandoning the flow, the system would not recover until a second ET arrives and queues in a finite period of time.

To prevent SIRQ from being stuck in such a situation, a recurring job is implemented that settles all time-critical flows. This job runs on an arbitrarily defined schedule and checks first if it is necessary to refresh the flows, since this could have happened seconds ago through an incoming event as shown in Listing 4.2. The updates are logged with a timestamp in the DB to provide full transparency on the automatic behavior and to allow for decreasing the load on the system when it is already busy.

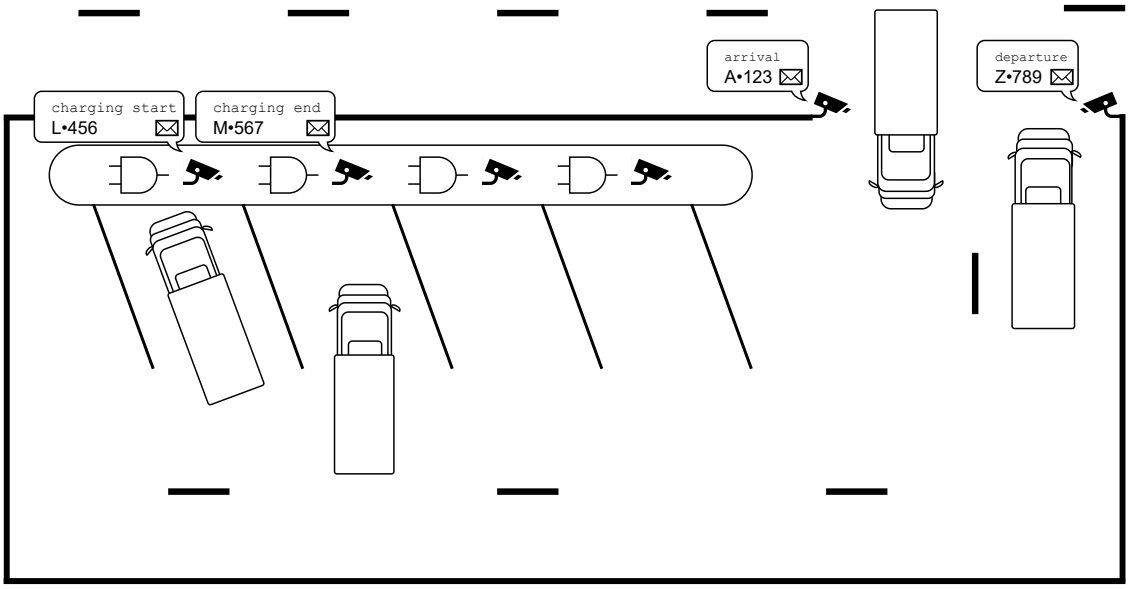
```
log(
      "INFO",
      `Refresh Flows CRON Job: Checking for last job timestamp`,
    );

    const lastExecutionResult = await sqlConnection.queryObject<
      {
        executionTime: string;
      }
    >({
      text: `
                    SELECT execution_time
                    FROM masterthesis_schema.jobs
                    WHERE CURRENT_TIMESTAMP < (execution_time + INTERVAL
                        '${threshold}')
                    LIMIT 1
                    ;
            `,
      fields: [
        "executionTime",
      ],
    }).then((itm) => itm.rows);

    if (lastExecutionResult.length > 0) {
      log(
        "INFO",
        `Last update was done in less than ${threshold} ago, scheduled
           run will skip this occurrence`,
      );
    } else {
      refreshFlows(sqlConnection);
    }
```

Listing 4.2: Recurring job settling flows proactively

4.3 Infrastructure

The integration of SIRQ with the infrastructure at an ETCS is based on the technical requirement *TR2* and further on the logical requirement *LR3*. The technical part is already partially covered in Section 4.1 with the decision to use REST API. However, this is focused on the receiving part of the data transmission. To examine how SIRQ can be integrated at an ETCS, the data capturing was mocked as there was no accessible testing charging station where the system could be evaluated.

Figure 4.3: Envisioned scenario of *ET* state capturing



Previous research introduced in Chapter 2 considered having terminals, similar to parking lots, that track arrivals at the ETCS. Despite this being a proven approach, it does not take advantage of recent technology, which increases user convenience by requiring fewer interactions. In Figure 4.3 an infrastructure is introduced based on visual recognition performed by cameras. The registration numbers are fetched using text recognition algorithms and then sent together with other metadata to SIRQ. An example payload of such a transmission is shown in Listing 4.3

```
{
    "licensePlate": "ABC*123",
    "chargerId": "a79974c8-80dd-4bf9-966d-8a2f7acce36e",
    "chargingStationId": "378d205b-6b19-4b3f-8a33-7cd30b131540",
    "timestamp": "2024-11-23T18:25:43.511Z"
}
```

Listing 4.3: Request body of a charging start request

A proof of concept of this automated recognition is implemented by using Google's ML Kit as the baseline of an Android app [19]. This includes a static and a real-time text recognition algorithm to retrieve the license numbers. Using the device's camera, the recognized value is sent upon a button click as an HTTP request to the SIRQ API.

As recognition is done in real time and automates a manual check-in system, the idea is further taken in Figure 4.3 by placing a dedicated camera for each charger, the entrance and exit. However, this is a proposal and is not required, since the implementation would also allow for any amount of cameras as long as the license number recognition can be distributed to the different states of the charging flow. Furthermore, this design allows for the tracking of departures at any time which denotes the end of a charging flow and therefore establishes the foundation that *LR3* is ensured.

Upon triggering the event when receiving the API request, SIRQ does not immediately persist the timestamp in the DB. First, it verifies whether the previous step presented in Figure 3.1 has been fulfilled. These algorithms are adjusted for each of the steps as there are different checks required to guarantee flow integrity. The charging end timestamp can only be set if for the given charging session at the corresponding charger a start charge timestamp has already been inserted into the `charging_flows` table as shown in Listing 4.4.

```
SELECT session_id,
    charging_station_id,
    end_charge_ts
FROM masterthesis_schema.charging_flows
WHERE license_plate = '${reqObj.licensePlate}'
    AND charging_start IS NOT NULL
    AND charger_id = '${reqObj.chargerId}'
    AND departure_ts IS NULL;

-- If there is a charging session fulfilling this, the update is done

UPDATE masterthesis_schema.charging_flows
SET end_charge_ts = CURRENT_TIMESTAMP
WHERE session_id = '${sessionId}'
RETURNING session_id,
    charger_id,
    end_charge_ts
```

Listing 4.4: Pre-check for ending a charging session including update

The charging session also includes the `companyId` which links the license plate to a specific company. Since truck companies include charging in their route planning, SIRQ offers the functionality to manage the ET fleet. For this prototype a key-value table is used, which stores a unique license number with the `companyId`.

Spot assignments, reservations, or auctions require preliminary checks, as outlined in Chapter 3, to guarantee anticipated functionality. To achieve this, an abstraction of the ETCS infrastructure is used in terms of charging spots. Hence, SIRQ offers endpoints for the admin role, which are used by the ETCS provider, to set up and alter the different ETCS.

The setup of a new ETCS is done by providing the values for the parameters indicating the total amount of charging spots and how many thereof are intended to be reservable, for which an example payload is shown in Listing 4.5. SIRQ enforces logical soundness of these parameters by checking if the property $c_i \geq r_i$ where c_i is the total amount of CS and r_i is the maximum amount of CS which are allowed to be reserved concurrently.

```
{
    "chargingStations": [
        {
            "chargingStationName": "testChargingStation",
            "totalChargingSpots": 2,
            "maxReserveSpots": 1,
            "active": true
        }
    ]
}
```

Listing 4.5: Request payload to set up a new charging station

4.4 Reservations

The endpoint `reservations` is located in the shared flow as it can be accessed by both the admin and the company user. However, the middleware part responsible for the shared flow will authenticate the requesting party and pass it to the endpoint if it is accessed as an admin or a company user. In this case, it is required to separate the two access rights, as an administrator can access any reservations made for all ETCS, while the company user retrieves the subset of data for the ETs linked to the corresponding `companyId`.

For data retrieval about reservations, the `GET` endpoint queries the DB including the required filters based on the user information received from the middleware. To create or delete a reservation, the `POST` endpoint is used. The implementation flexibly allows sending one or more creation or deletion of a reservation in a single request. An example payload for a mutation of a reservation is shown in Listing 4.6.

```
{
    "reservationIdsToCreate": [
        {
            "chargingStationId": "378d205b-6b19-4b3f-8a33-7cd30b131540",
            "reservations": [
                {
                    "licensePlate": "ABC*123",
                    "startTimestamp": "2024-11-20T15:15:00.000Z",
                    "endTimestamp": "2024-11-20T15:30:00.000Z"
                }]
        }],
    "reservationIdsToRemove": [
        {
            "chargingStationId": "378d205b-6b19-4b3f-8a33-7cd30b131540",
            "reservations": [
                {
                    "licensePlate": "ABC*123",
                    "startTimestamp": "2024-11-20T14:15:00.000Z",
                    "endTimestamp": "2024-11-20T14:30:00.000Z"
                }]
        }]
}
```

Listing 4.6: Request payload to delete an old reservation and set up a new one

The creation requires the payload to be formally checked first. As explained in Section 3.5, the reservation should respect the minimum acceptable interval between the current timestamp and the start time. It implicitly requires that the start time is in the future, as the given timestamp needs to be after the end of the control interval, which is always in the future. Since it does not require complex calculations and hence can be done quickly, this check is done first. Upcoming checks are not performed if this first one fails. By implementing this design, complex calculations for requests that do not meet basic requirements are avoided.

Subsequently, the requested reservation period is verified to respect the maximum length. These checks are executed in memory *i.e.*, without the need to fetch data from the DB. Upon successful completion of the time constraint checks, the given information about the ET is verified. This prevents unauthorized creation or deletion of reservations as the requested license number and the company are matched with the fleet data from the DB.

The last check performed verifies if the maximum amount of reservations for this specific ETCS is not violated as shown in Listing 4.7. Therefore, the situation is first simulated in a temporary table, which models the new potential situation. The table combines the current reservations with the requested reservations received through the API while aggregating the amount of reserved spots for each interval. To save computational resources, all intervals for which there are no reservations present are excluded, as the amount of reservations are implicitly set to 0 and hence by definition do not violate any constraints.

After passing all of the above-mentioned checks, the reservations are inserted or removed based on the received instruction. SIRQ detects without any further notice if there was already a reservation with the same start time and changes the end time accordingly, as the insertion uses a `INSERT INTO ... ON CONFLICT (license_plate, start_ts) DO UPDATE` logic.

```
1     SELECT DISTINCT ON (start_ts) charging_station_id,
2         start_ts,
3         (
4             SELECT COUNT(1) AS rv_count
5             FROM coalesced_res AS rv -- coalesced_res is a temporary
                  view which includes all reservations to create and delete
6             WHERE (
7                     r.start_ts <= rv.start_ts
8                     AND r.end_ts > rv.start_ts
9                 )
10                OR (
11                    r.start_ts >= rv.start_ts
12                    AND r.end_ts <= rv.end_ts
13                )
14                AND r.charging_station_id = rv.charging_station_id
15                AND r.license_plate != rv.license_plate
16        ) AS reservation_count
17    FROM masterthesis_schema.reservations AS r
```

Listing 4.7: Main part of reservation check: Counting reservation including requested ones

4.5 Auctions

Similarly to reservations, the `auctions` endpoint, located in the shared flow, offers a `GET` and `POST` method which can be accessed by administrators and company users. The `GET` method returns an overview of the auctions grouped by the ETCS and filtered by the access rights of the requesting entity's role. Apart from metadata fields, which provide unique identifiers for the auction, ETCS and truck company there are the start and end time of the auction as shown in Listing 4.8. Since the requesting user expects a quick confirmation if the auction was successful, the maximum duration is pre-defined but can be changed as there is no other logical dependency in SIRQ's implementation.

Furthermore, the response object contains the maximum accepted price, which is defined by the auction starter. If this value is not defined, the auction initiator needs to explicitly set the automatic accept flag to `true`, which accepts the best offered price from the initiator's perspective. Although this option is chosen, SIRQ still limits the maximum accepted price to protect the auction starter from extortion by the other auction participants.

```
{
  "chargingStationId": "f8b139af-82c2-4996-b9ac-7f557cbd0e44",
  "auctions": [
    {
      "auctionId": "d9999de5-56c0-4aaf-ad9a-1e2dc54440f7",
      "chargingStationId": "d8e76b85-3d12-4190-ad73-79ad3ffd98eb",
      "companyId": "166bb335-da3a-496d-a362-5cbfba72d178",
      "licensePlate": "ABC*123",
      "auctionStartTimestamp": "2024-11-20T15:15:00.000Z",
      "auctionEndTimestamp": "2024-11-20T15:20:00.000Z",
      "maxAcceptedPrice": 15,
      "autoAccept": false,
      "auctionType": "chargingSpot",
      "winningPrice": 10,
      "auctionFinished": true
    }
  ]
}
```

Listing 4.8: Response body for auctions

Based on the type of auction requested, a CS or a reservation, the payload of the `POST` request requires additional attributes. A CS auction includes all base attributes, which provide information about the ET (license number), the ETCS and implicitly, by sending the authentication token, the company identifier. Apart from this administrative information, the maximum accepted amount or the will to accept any offer is passed in the request body. In case of a reservation auction, the desired reservation start and end timestamps are sent, which implicitly indicate that the request aims to auction for a reservation in the future. If an auction should be canceled, the unique auction identifier and the flag indicating a request for removal is sent instead of the other fields.

The received payload is first routed to the flow corresponding to starting an auction or removing one. If the requested action is the initiation of an auction, the request is passed to the matching auction type. In the spot auction flow, SIRQ preemptively checks that

no spot is available for the requested ETCS. In case there is still a spot available, the auction initiation is rejected.

If the request targets an auction for reservations, the check searches for matching reservation periods. However, it is not required that the start and end time of an existing reservation is equal to these timestamps in the request. At minimum, the requested reservation period needs to be in between of an existing one. Although this decreases the chance of a match, it also reduces the amount of auction requests sent to the truck companies, and hence avoids rejection of using the functionality because it tends to be overwhelming or annoying. In the system's point of view, this design decreases the response time due to a reduction of computational complexity as fewer variations of overlaps are considered.

To avoid a circumvention of the introduced constraint about the latest possible request for reservations, a check verifies that the start time respects the specified time interval in which it is too early to create a reservation. Otherwise, the auction for reservations would provide a circumvention of this constraint and potentially cannibalize the functionality provided for a spot auction. After passing this check and determining the suitable reservations, the auction is started by inserting the details, including a unique identifier for each auction, into the `auctions` table.

Consequently, in the same query to the DB, the selected reservations are inserted into the `reservation_auction_offers` table. This design improves memory usage, as the table is reused for a `GET` endpoint `auction-offers/<id>` returning the current entered data for a specific auction offer. The corresponding `POST` endpoint is then used by the participants of an auction to enter their offers. Each offer will be stored with a timestamp that denotes the receipt of the auction offer.

The auction ends when all notified truck companies send their offers or the auction end time is reached. At this point, SIRQ filters and ranks the offers ascending by the price and the timestamp the offer was received. As the DB entry for the winning offer includes the company identifier and the start time, the reservation in the `reservations` table can be linked to this offer. Finally, the company identifier and license plate for the original reservation will be adjusted to the values of the auction's initiator. This settlement of an auction is included in the recurring procedure introduced in Section 4.1 to ensure that auctions are processed if no or not all of the requested participants return an offer. By design, it is not required to respond to an auction request.

Similar to auctions for reservations, pre-checks are performed for auctions targeting CS. As these auctions are started with the intent to jump a queue, the first check ensures that there is no free spot available in the respective ETCS. This ensures that auction requests are sent to the participants only in case the situation at the ETCS fits an auction scenario.

To determine which of the occupied spots should be included in the auction, not only the current situation needs to be covered, but also the interval of the minimum charging time as shown in Listing 4.9. The check models the future amount of reservations covering the duration of the minimum charging time and excludes the calculated amount of spots from the auction to ensure that the reservations are not impacted by the auction.

```
SELECT r.charging_station_id,
    LEAST(COUNT(*), cl.max_reservation_spots) AS future_rsv_count,
    total_charging_spots
FROM masterthesis_schema.reservations AS r
    LEFT JOIN masterthesis_schema.charging_stations AS cl ON cl.
       charging_station_id = r.charging_station_id
WHERE start_ts BETWEEN CURRENT_TIMESTAMP
    AND (CURRENT_TIMESTAMP + INTERVAL '${minChargeTime}')
    AND r.charging_station_id = '${reqObj.chargingStationId}'
GROUP BY r.charging_station_id,
    cl.max_reservation_spots,
    cl.total_charging_spots
```

Listing 4.9: Counting future reservations to determine auction participants

The exclusion of specific chargers raises the question of how to decide which CS is excluded. Listing 4.10 shows how SIRQ decides based on the elapsed charging time as these CS are most likely to be abandoned soon and will not be assigned anymore by the spot assignment algorithm due to the constraint that guarantees reserved spots to be available.

```
charger_order AS (
    SELECT session_id,
        cf.charging_station_id,
        charger_id,
        spot_assignment_ts,
        start_charge_ts,
        ROW_NUMBER() OVER (
            ORDER BY COALESCE(start_charge_ts, spot_assignment_ts)
        ) AS spot_order,
        future_rsv_count
    FROM masterthesis_schema.charging_flows AS cf
        LEFT JOIN count_res AS cr ON cr.charging_station_id = cf.
           charging_station_id
    WHERE cf.charging_station_id = '${reqObj.chargingStationId}'
),
filtered_insertions AS (
    SELECT *
    FROM charger_order
    WHERE spot_order > future_rsv_count
)
```

Listing 4.10: Exclusion of CS to determine auction participants

The bidding procedure is similar to the one implemented for reservations. For the `spot_auction_offers` table the columns are different, as it is not necessary to store a start and end timestamp. Instead, an identifier for the charger is required to map the auction outcome to the spot assignment, since the CS will be swapped between the auction "winner" and the auction initiator.

Performing the auction is the same for spot auctions as for reservation auctions. The offers are ordered in the same way and in both auctions the lowest price wins. However, for both reservation types a constraint has to be introduced to avoid that the good *i.e.*, the CS or the reservation, which is subject to the auction, is still in possession of the auction winner. Hence, SIRQ will automatically remove the auction offer for a reservation if the

reservation is removed during the course of the auction, and the same applies to the CS when the ET occupying the respective CS stops charging and abandons it. This is achieved by removing the corresponding auction offers whenever an ET removes the plug, thus setting the end charging timestamp.

As shown in Figure 3.4, an additional check needs to be added to include spots which were the subject of an auction. Therefore, a record of these CS is implemented as a table `spot_assign_locks`. In addition to the data required to map the auction initiator with the CS, two timestamps are introduced: A lock start timestamp and a lock end timestamp. This principle allows to lock a specific CS for the interval defined by the two timestamps and is not considered as an available spot even though it might be free. Hence, the lock determines how long two rules can be overridden: During the locked period of time, the CS is not considered available even if the auction "winner" decides to leave before the auction initiator arrives. Secondly, even if the spot is occupied when the ET with the respective license plate arrives and is registered by SIRQ, it will be assigned the CS stored in the table `spot_assign_locks` and automatically end the charging flow of the auction "winner" as shown in Listing 4.11.

```
1  WITH just_arrived_flows_with_lock AS (
2      SELECT session_id, sal.charger_id
3      FROM masterthesis_schema.charging_flows AS cf
4          JOIN masterthesis_schema.spot_assign_locks AS sal ON cf.
               license_plate = sal.license_plate
5      WHERE arrival_ts IS NOT NULL
6          AND spot_assignment_ts IS NULL
7          AND departure_ts IS NULL
8          AND cf.charger_id IS NULL
9  ),
10 charging_locks AS (
11     SELECT charger_id FROM masterthesis_schema.spot_assign_locks
12     WHERE CURRENT_TIMESTAMP BETWEEN lock_start_ts AND lock_end_ts
13 ),
14 currently_charging_with_lock AS (
15     SELECT session_id, cf.charger_id
16     FROM masterthesis_schema.charging_flows AS cf
17         JOIN charging_locks AS cl ON cl.charger_id = cf.charger_id
18     WHERE start_charge_ts IS NOT NULL
19         AND end_charge_ts IS NULL
20 ),
21 to_update AS (
22     SELECT jafwl.session_id AS session_id_new, ccwl.session_id AS
          session_id_old, ccwl.charger_id
23     FROM just_arrived_flows_with_lock AS jafwl
24         JOIN currently_charging_with_lock AS ccwl ON ccwl.charger_id =
              jafwl.charger_id
25 )
26 INSERT INTO update_data
27 SELECT * FROM to_update;
28 UPDATE masterthesis_schema.charging_flows AS cf
29 SET end_charge_ts = CURRENT_TIMESTAMP
30 FROM update_data AS ud
31 WHERE session_id = ud.session_id_old;
```

Listing 4.11: Spot assignment flow changes to include auctions with locks

Chapter 5

Evaluation

In this Chapter, the implementation of SIRQ is analyzed and evaluated. Using well-defined scenarios, different aspects are investigated to see how the system performs under the defined circumstances. The system used to run the defined scenarios on is introduced in Section 5.1. These scenarios are explained in Section 5.2. Based on these scripts the execution is presented in Section 5.3. Furthermore, the scenarios are used and analyzed in Section 5.4, providing a summary of the situations where SIRQ performs well and highlighting areas where design enhancements are necessary. Finally, in Section 5.5 the requirements defined in Chapter 3 are compared to the output of the evaluation to determine if they were met.

5.1 Technical Setup

The analysis of the SIRQ including the Deno container and the PostgreSQL database was run on a Raspberry Pi 5 with a 2.4 GHz 64-bit Quad-Core ARM Cortex-A76 processor, 4 GB DDR4 RAM, Raspberry Pi OS Kernel Version 6.6 and Debian Version 12 (bookworm). There, two Docker containers are instantiated for running the server and the database.

The app for the automated visual recognition of the license number was deployed and tested on a Xiaomi Redmi Note 13, Qualcomm Snapdragon 685 8-Core 2.8 GHz, 8GB RAM running HyperOS 2, Android 15. As it is a prototype, the functionality of the app focuses on recognizing license numbers on static images or by using the camera function. To achieve a higher accuracy and match the DB fleet entries, the recognized value is shown on the phone's screen and could be corrected through an input text field before sending it to the running SIRQ server.

5.2 Scenario Script

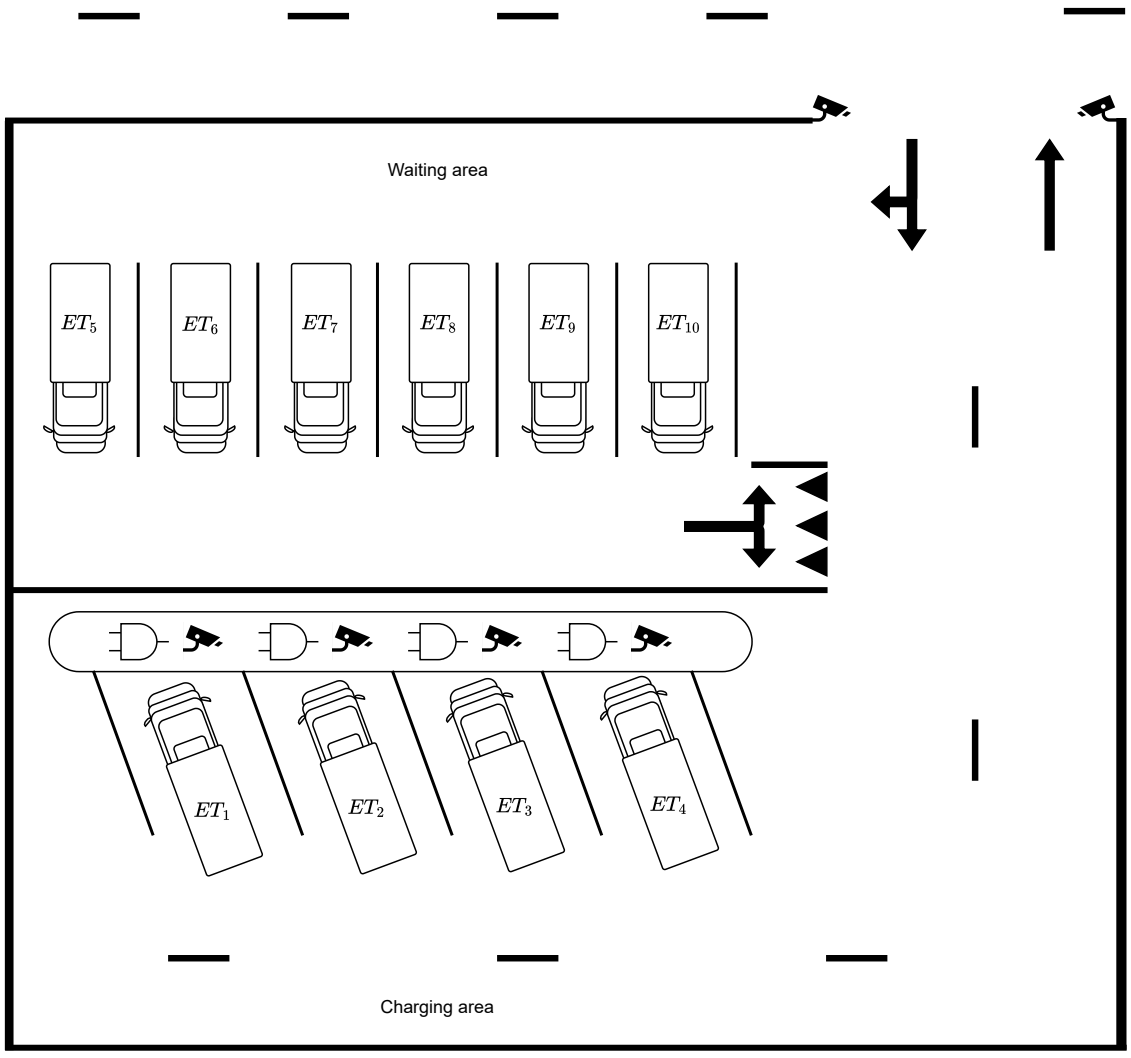
The following definition of scenarios is based on the selected real-world scenarios with which SIRQ is confronted. All of the assumptions stated in Section 3.1 are applied for

the specified flows in order to focus on the main characteristics of SIRQ. The scenarios are established from scratch through the APIs provided by SIRQ. The starting point for each scenario is SIRQ's server running with a healthy connection to the DB.

Although SIRQ offers the possibility to manage users, ETCS and fleets for trucking companies, it is assumed that the administrative setup was done previously as it is not the main focus of the evaluation. Each scenario includes an ETCS having 4 CS, with a maximum of 2 reservable spots and 10 ETs $\{ET_1, ET_2, ..., ET_{10}\}$ from 10 different truck companies available to model usage of the ETCS.

The parameters introduced for the different check intervals used by SIRQ are kept in an interval of (0, 20] minutes to test the system more efficiently by reducing the idle time *i.e.*, waiting for charge cycles to finish. Finally, each scenario uses the infrastructure proposed in Section 4.3 and enhances it to meet the requirements of the outlined scenario script as shown in Figure 5.1. Thus, SIRQ immediately receives an update for each of the states introduced in Figure 3.1.

Figure 5.1: Base state of evaluation scenarios



Scenario 1 The aim of this scenario is to explore a situation where there are no queue interaction features utilized, and SIRQ functions only as a QM system. Hence, the constraint parameters to be set are limited to the minimum and maximum charging time. These are set very low as outlined in the general scenario description to increase the efficiency of the investigation while still being slow enough to get an overview of each state the system is in. These overviews are captured and used to analyze the scenario.

Scenario 2 In contrast to scenario 1, the queue interaction features of SIRQ are available and frequently used by each of the queuing ETs. This scenario aims to exclusively use these functionalities by setting the maximum charging time to exceed the investigation period, while ETs are not leaving before the maximum charging time is reached when they are assigned a spot. However, no new reservations are possible as the ones set initially cover the full investigated period.

Scenario 3 Improvements in trucking route planning are examined in the final scenario. The state of the arrived ETs at the ETCS remains static as the non-static flows are covered by the previous two scenarios and the route planning investigated here is not dependent on ETs finishing their charging session nor the queue moving. The minimum charging time is set to 1 hour and according to SIRQ's specifications explained in Section 3.5, the period in which no reservations are allowed is set of equal length. Each ET owning a reservation during the investigated period is willing to trade this through a reservation auction.

5.3 Scenario Execution

The defined scenarios are executed using the HTTP endpoints of SIRQ to imitate ET behavior following the given constraints. Moreover, the Android application conducts the tracking by utilizing visual recognition. The execution procedures per scenario are as follows:

Scenario 1 Starting with an empty ETCS, $\{ET_1, ET_2, ..., ET_4\}$ begin to fill the available CS until none are available. After that, the remaining 6 ETs $\{ET_5, ET_6, ..., ET_{10}\}$ arrive and form a queue, waiting for CS to free up. All arrivals are random and do not follow a specific order or interval. Each ET arrives at the ETCS during the investigated interval. The ET $\{ET_5, ET_6\}$ randomly decide to leave the area without having charged. $\{ET_2, ET_9, ET_{10}\}$ decide to end the charging early *i.e.*, without being fully charged. After ET_{10} departs the ETCS, no further ET is present and the investigation interval ends.

Scenario 2 Similar to scenario 1, the initial state of the ETCS is empty. Incoming ET $\{ET_1, ET_2\}$ occupy the available slots, whereas pre-existing reservations for the remaining CS have been made for $\{ET_3, ET_4\}$. The arrivals are randomly distributed throughout the interval in random order, but all ETs eventually arrive at the ETCS. Due to the given constraints the CS are only freed through auctions, which require

the occupying ETs to offer an attractive price. This is given in the current scenario, yielding the auction pairings shown in table 5.1. Similar to scenario 1, the investigation interval ends with ET_{10} leaving.

Table 5.1: Auction pairings for scenario 2

Auction starter	**Auction winner**
ET_5	ET_1
ET_6	ET_2
ET_7	ET_3
ET_8	ET_4
ET_9	ET_5
ET_{10}	ET_6

Scenario 3 The planning procedure for 3 specific ET $\{ET_1, ET_2, ET_3\}$ is examined. ET_3 owns a reservation which covers the span of the investigated scenario. The remaining 7 ETs are either occupying CS $\{ET_4, ET_5, ET_6\}$ or queuing $\{ET_7, ET_8, ..., ET_{10}\}$. As defined by the scenario script $\{ET_4, ET_5, ..., ET_{10}\}$ do not change their state in this scenario.

The truck company of ET_1 plans a route diversion based on an unexpected change in delivery location. This new location requires ET_1 to get a recharge as the distance is too far considering the current state of charge. Hence, in order to be on time, a guaranteed charge at ETCS is required at latest in 15 minutes after the start of the investigation interval. Therefore ET_1 tries to reserve a spot, guaranteeing availability upon arrival.

In contrast, the timing constraints for the route allows the truck company of ET_2 to reserve a CS after the minimum charging time ends *i.e.*, a point in time after the investigation period. ET_3 already owns a reservation for the interval covering the duration of this scenario and is willing to trade it for a favorable price.

5.4 Scenario Evaluation

The scripts outlined in the specified scenarios were executed and interactively reviewed utilizing the designated endpoints of SIRQ. The corresponding results are presented in Table 5.2. Three distinct categories were analyzed according to the objectives defined in the thesis:

Performance Since SIRQ was not tested in a real world setting with an existing ETCS and infrastructure, the waiting time was used as an approximation to compare the different scenarios. However, as the intervals were chosen to be short, the obtained data does not represent any real data and therefore it is discussed relatively instead of quantitatively. The baseline to be compared with is a system without any minimum and maximum charging times as proposed in most of the work presented in Section 2.2.

User engagement With the queue interaction functionalities introduced, each participant finds a strategy to increase the chance of a favorable outcome, as the principles of non-cooperative game theory describe. Thus, the flexibility provided creates a new dependency on the other users. Therefore, the user engagement measures to which extent the actions of the users are required to achieve their goals defined in the investigated scenario.

Security Due to the increased dependency on interaction between different parties, measures were taken to ensure that users follow the rules set by SIRQ. Therefore, established constraints are investigated in the defined scenarios to detect potential malevolent actions preventing the outlined goals of the different actors.

As the interaction functionalities were not used during **Scenario 1**, the empty ETCS could be filled by the first 4 ETs arriving. In this case, SIRQ requires a short setup time due to the implemented charging flow functionality, which assigns spots to all of the 4 ETs as shown for the first one in Figure 5.2. However, once the charging process is established by using the implemented visual recognition, no further time is required.

Figure 5.2: SIRQ log output after ET_1 is assigned a spot upon arrival

```
2024-12-20 14:10:54 --- [INFO] --- Assigning spots for all charging stations where there are
available spots
2024-12-20 14:10:54 --- [INFO] --- Successfully assigned spots
2024-12-20 14:10:54 --- [DEBUG] --- Assigned spots '[{"sessionId":"5aab72b9-10b0-471d-9619-
dadbe71293ca","spotAssignmentTimestamp":"2024-12-2-T14:10:54.616Z","chargerId":"ed9bcc6f-cd1b-
4f74-92bb-52dbf9a03010","chargingStationId":"378d205b-6b19-4b3f-8a33-7cd30b131540"}]'
2024-12-20 14:10:54 --- [INFO] --- Releasing SQL connection from /api/private/charging-
flows/arrivals
```

When all CS are filled, the waiting time depends on the position and the length of the queue, similar to the solutions proposed in current research. Through the constraints introduced by SIRQ, this waiting time is bounded by $[0, |ET| \cdot maxCT]$ where $ET = \{ET_1, ET_2, ..., ET_{n-1}\}$ and $maxCT$ denotes the maximum charging time, since each ET at maximum has to wait for every other ET charging first to end the charging session.

Since the scenario prohibited the usage of auctions and reservations, user engagement is kept minimal. It is lower than the user engagement required in similar approaches discussed, as visual recognition does not require manual registering at the ETCS *e.g.*, getting a ticket with a queue number.

The constraints on maximum charging time and maximum transition times between different states of the charging flow ensure a secure procedure. In addition, the authorization token combined with assumption *A1* ensures that data is handled confidentially. Nonetheless, while the charging flow can be virtually guaranteed, it does not reduce the possibility of a vehicle physically obstructing a CS.

In **Scenario 2** the ETCS is filled by ETs with reservation and the first ones without. ETs queuing can only get a CS if an auction is done. These constraints model a QM system without queue interactions, which is encountered in various works presented in Table 2.2.

SIRQ allows an ET to actively decrease waiting times by using the provided functionalities in such a scenario, in which an ET waits for another user to eventually stop charging.

This enforcement of using auction results in high user engagement due to the need of charging ETs to react to the requests for auctions received. Based on assumption *A7*, the queuing ETs will get a CS through these auctions as the charging ETs are willing to cede their CS. However, if multiple auctions are started, SIRQ does not aggregate the notifications and only ask once for an offer. Therefore the user gets a notification for each of the requests and has to respond to them separately, which could result in overwhelming the user.

As there is no limiting constraint for requesting auctions, a distributed denial of service (DDoS) is a potential threat of SIRQ. In addition, there is no guarantee that the auction is non-cooperative, as both sides of the auction *i.e.*, requesting party and offering party could be working for the same truck company. This violates the non-cooperative property of the auction and hence it could lead to "insider" auctions, which by definition is a pseudo auction as the offer will have the same originator and receiver.

Table 5.2: Result of SIRQ's effectiveness across defined scenarios

Scenario	**Performance**	**User Engagement**	**Security**
Scenario 1	Waiting time depending on queue	Minimal due to automatic recognition	Constraints provide security
Scenario 2	Ability to actively decrease waiting times	High due to interaction needed for auctions	Auction might include cooperative actors
Scenario 3	No waiting time regarding guarantee to jump the queue through reservation	High in case of reservation auction, low otherwise	Parameter choice and multiple reservation lead to starvation

The route planning for $\{ET_1, ET_2, ET_3\}$ in **Scenario 3** is simplified through SIRQ's functionalities, as a successful reservation by definition guarantees the spot to be available during the requested period. Hence, the waiting time is reduced to a minimum even if a queue is faced at the ETCS upon arrival.

However, due to the constraints given, ET_1 cannot successfully reserve a CS as no reservations are allowed in an interval below 1 hour as the output log in Figure 5.3 shows. As this was required to ensure the minimum charging time, ET_1 is still able to try to start an auction for a reservation. By definition, SIRQ includes ETs with an existing reservation for which the requested reservation time is either between the existing reservation period or matching the same start and end time. Assuming this is the case, ET_1 is able to take over the reservation from ET_3.

This improves the planning situation for ET_1 as the route can be planned even though there was an ad-hoc change. However, it depends on the willingness of the other ETs owning a reservation. In this scenario, the willingness and an acceptable offer for the CS were explicitly given through the definition. While the constraints ensure that the SIRQ

guarantees QoS in terms of minimum charging time and guaranteed available spots, the choice of the blocking interval for reservations could lead to situation in which there is no available spot guaranteed at the ETCS. Assuming ET_3 did not have the reservation, ET_1 would have to either hope for no queue at the ETCS or start an auction for a CS if there was a queue. As there is uncertainty in both scenarios, the planning for this specific situation is not given.

Figure 5.3: SIRQ log output when trying to reserve for ET_1

```
2024-12-20 15:11:26 --- [INFO] --- /shared/reservations POST request received
2024-12-20 15:11:26 --- [DEBUG] --- Request:
{"reservationIdsToCreate":[{"chargingStationId":"378d205b-6b19-4b3f-8a33-
7cd30b131540","reservations":[{"licensePlate":"TRUCK*1","startTimestamp":"2024-12-
20T15:15:00.000Z","endTimestamp":"2024-12-20T15:20:00.000Z"}]}],"reservationIdsToRemove":[]}
2024-12-20 15:11:26 --- [INFO] --- Fetching fleet data
2024-12-20 15:11:26 --- [DEBUG] --- Fleet data for company ID: 53fe4cc8-7e1b-440a-b62f-
eaafb89d9f2e, name: TruckCompany_1
2024-12-20 15:11:26 --- [INFO] --- Timestamp of reservation to add is too close in the future,
reservation is not added
2024-12-20 15:11:26 --- [DEBUG] --- Reservation for license plate: 'TRUCK*1', start timestamp:
'2024-12-20T15:15:00.000Z' < '2024-12-20T16:11:26.019Z' (now + 3600 seconds)
```

Although tools are given to decrease the likelihood of starvation, it still depends on the decision of the ETCS provider about the amount of maximum reservable spots r_i and the total amount of CS c_i at the ETCS i. If $r_i = c_i$ and the reservation start and end times are the same for all ETs with a reservation currently charging, then the queue will not move until the first reservation ends or the maximum charging time is reached.

5.5 Requirement-based Evaluation

Throughout the scenarios, the logical requirements described in Section 3.2 and the technical requirements outlined in Section 3.3 were assessed. To rank the state if the requirement was successfully implemented, a scale is introduced:

- ★ Fully implemented
- ⯪ Partially implemented
- ☆ Not implemented

The defined logical requirements *L1-3* were implicitly covered by at least one of the three scenarios. As there was no queue interaction possible in Scenario 1, SIRQ managed the queue without using the additional functionalities implemented based on requirement *LR1*. These functionalities were used in Scenario 2 & 3 in which queuing ETs successfully were able to jump the queue by starting a CS auction as defined in *LR2*. However, this only works if assumption *A4* holds, since each ET has to be able to bypass the queue even if it was already part of the queue.

Scenario 3 demonstrated the necessity of reservations being subject to an auction. SIRQ successfully handled such a case with the assumption *A5*, as reservation auctions without further information about the capabilities of the charger require all of them to be equal. Otherwise, the price expectations differ due to the alternating specifications of the charger.

In Scenario 1 ETs did leave the ETCS at different stages of the charging flow without any negative impacts on the other participants or SIRQ itself. Hence, requirement *LR3* is ensured by settling the flows using the recurring CRON job.

Setting up the different scenarios and reusing some of the parts from the previous ones showed that the state can be rebuilt based on the DB contents. In addition, the DB Docker container was working with a mounted volume, to persist data according to the design based on the requirement *TR1*.

Throughout all scenarios, the proof-of-concept for visual recognition of the license plate was used and therefore demonstrated a simple end-to-end flow. As the connection of this part of the infrastructure worked with simple HTTP requests, requirement *TR2* is considered to be successfully implemented.

Table 5.3: Analysis of logical and technical requirements for SIRQ

Requirement	**Achieved**	**Comment**
$LR1$	★	Scenario 1 revealed that SIRQ works without additional interaction
$LR2$	★	Scenario 2 & 3 demonstrated variations of jumping the queue
$LR3$	★	Scenario 1 showed that ETs could leave spontaneously
$TR1$	★	States can be rebuilt at any given time by storing atomic data in DB
$TR2$	★	Proof of concept for end-to-end implementation of a QM system including infrastructure shows simple integration
$TR3$	⯪	Authentication and transactions for DB provide confidentiality and integrity while sacrificing availability

Authentication and RBAC increased confidentiality by structuring accesses and distributing the rights only to the required roles. To ensure integrity of the data, the DB uses transactions to quickly lock tables and prevent unwanted states of the application through concurrent access to the same row of a DB table. Using connection pools does not resolve this limitation since the DB table remains locked, and various contexts do not bypass this issue. Hence, availability is sacrificed as requests were rejected due to table locks during the evaluation. Therefore, the CIA principles could not be fully covered and hence requirement *TR3* is only partially achieved.

5.6 Flow Security Evaluation

The evaluation of the outlined scenarios involved testing the constraints presented in Chapters 3 and 4. In combination, these constraints are necessary to ensure that participants adhere to the established flow, even though they are allowed to make random decisions. Therefore, the scenarios allow for an assessment on how the constraints perform upon application.

The scenario baseline requires the initial setup of an ETCS with 4 CS and thereof 2 can be reserved by users. As this is done in an automated way, SIRQ ensures that the input is handled transactional *i.e.*, the intended state provided by the input parameters can be represented throughout the whole system or it is rejected. Hence, the endpoint includes the logic to create missing CS if these are not existent yet.

In addition, SIRQ also considers the life cycle of an ETCS, due to physical failures of CS or expansion of the ETCS, CS are added or removed. When the parameters are modified, SIRQ verifies that the adjustment does not conflict with any anticipated future state based on the booked reservations. Therefore, the check ensures that the scenarios are not facing any infrastructural errors due to unsafe operations.

Furthermore, there are checks to prevent ETs from starting two charging flows at once. Primarily, this is done to avoid adding the same ET to the queue twice and hence blocking other participants from getting a spot. However, this could also be done without malicious intent by a simple malfunction of the visual recognition which accidentally captures the ET twice. In this case, SIRQ asks the user to settle the open charging flow first before starting a new one.

This check is done across all ETCS to prevent concurrent charging with the same license number, as these are unique. To prevent malicious actors from trying to deceive a visual recognition system by mocking the license plate, the attempt to start a second open charging flow needs to be resolved manually in this case. In this situation, if the system autonomously terminates an active charging process, it could potentially impose the end of the charging process for another ET.

The first scenario depends on the operation of the recurring procedure detailed in Section 4.2, which guarantees state safety by enforcing specified intervals. Due to the defined property, that no queue interaction functionality is used, the queue is guaranteed to move due to the chosen minimum charging time. As SIRQ ends the charging sessions if the defined minimum charging time is exceeded, a continuous flow of new charging sessions and departures is established.

In scenario 2 the spot assign locks introduced in Section 4.5 are applied to prevent the spot assignment procedure to assign these CS to other participants. However, this lock is only valid for the specified duration and ensures that the auction is only started by ETs which have already arrived or are about to do so. Otherwise, the auction functionality could be used as a reservation replacement and prevent a spot assignment for the specific CS.

Apart from the earliest possible reservation constraint discussed in Section 5.4, the reservation check in scenario 3 is used twice, which requires the aforementioned soundness of the infrastructure. When the reservation request is received, SIRQ checks for any breaches of the maximum permitted CS occupied by reservations and declines the request if a breach is detected. The same check is performed when the proposed auction for reservations is taking place. Hence, these checks ensure that the infrastructure parameters are still respected after using the interaction functionalities.

5.7 Discussion

The analysis of the defined scenarios yields a positive performance assessment when considering the increase in user flexibility through the auction and reservation functionalities. Although user engagement is increased, SIRQ transforms it from bare administrative tasks without any direct benefits of the user *e.g.*, using a kiosk to inform the QM system about the arrival, to optional activities which aim to directly improve the situation of the user.

However, these interactions are based on an assumption (*A7*) made on the user behavior. If the truck companies are exclusively sticking to their plan and do not consider it a benefit to reduce their costs by receiving money from an auction, the interaction functionalities are not beneficial. Furthermore, the user engagement requires truck companies to be registered in order to use SIRQ. This is covered by assumption *A1* in the chosen scenario, but is not given in a real world scenario.

The constraints introduced kept the charging flow secure in each scenario. However, they increase the complexity to setup the system for an ETCS provider, as they need to be defined first. Moreover, if not selected carefully, SIRQ's capabilities may become restricted and, in the worst scenario, may not function as intended. Although this denotes a strong dependency on these definitions, the benefit of a flow secure system is inevitable for an interactive system which users measure by the perceived fairness.

As SIRQ implements state safety by carefully managing data, it is highly dependent on the database used. While this provides data persistence and therefore fulfills the technical requirement *TR1*, it also introduces concurrency problems due to increased transaction usage, which locks DB tables upon writing. Due to the required checks, which are also done in the DB, this design generates a bottleneck. It affects the system for a single ETCS in peak times, but assuming that the system is widely used for multiple ETCS it could cause availability issues.

Apart from the queue interaction functionalities, SIRQ already showcases a proof-of-concept of an end-to-end integration between the external systems of truck companies, the physical infrastructure of ETCS and the QM system. The assessment verifies that the selected design is capable of simulating a realistic ETCS scenario. It achieves this by representing various users as roles and modeling the infrastructure as a virtual ETCS, implementing built-in automation through visual recognition. In addition, by integrating the interaction functionalities and the QM with the charging flow, the system satisfies different complex user needs while still allowing all functions to run in parallel.

Chapter 6

Final Considerations

Concluding the thesis with this chapter, the summary is done in Section 6.1. Furthermore, the drawn conclusions based on the evaluation are discussed in Section 6.2. Finally, the future work Section 6.3 outlines remaining challenges and subsequent ideas worth exploring.

6.1 Summary

In contrast to ICE vehicle refueling, charging times for EVs are considerably longer due to the methods used to replenish energy. Consequently, CS are used for a certain period to get the state of charge needed by the consumer. This introduces new challenges such as queues that arise at the EVCS during peak times, as multiple EVs request a charge but the amount of CS is limited. Enhancing the attractiveness of using EVs thus requires advances in reducing waiting times and improving the overall user experience.

Recent studies explored various methods for optimizing queue management systems at EVCS identifying ways to minimize wait times and improve charging efficiency. While privately owned EVs typically charge spontaneously, company-owned ETs plan their routes in advance, and thus, the time expected to charge is also known. To reduce the uncertainty in planning, the QM is required to provide guarantees for available spots.

This Master Thesis introduced SIRQ, a prototype which provides interactivity with a QM system. The implemented functionalities allow the users of the system to reserve CS at ETCS, which allows arriving ETs to jump the queue if they own a reservation for this period of time.

In addition, the auction functionality facilitates jumping the queue spontaneously, as CS in use can be freed if the ET occupying the spot offers the lowest price in the auction. Hence, the requesting ET pays a surcharge on top of the charging costs to reduce the waiting time. Upon successful auction, the requesting ET is allowed to take the corresponding CS from the auction "winner". SIRQ incorporates the complete charging process, from when an ET arrives to the point it departs the ETCS, to automate spot assignments using

suitable logic depending on the chosen queue interaction. As this requires integration with existing infrastructure to keep track of the current state of the ETCS, a subsystem was implemented as a proof-of-concept to achieve automated visual recognition of license plates. This integration facilitates the enforcement of set constraints to ensure improved QoS through clearly established limits on system usage.

However, SIRQ does not impose exact specifications of these constraints, but rather, allows the ETCS provider to decide how to specify these. Therefore, it is possible to limit the amount of spots which can be reserved. To increase the flexibility, the reservation in such a scenario can be subject of an auction if an ET is in urgent need of a guaranteed charge, following a similar logic as for spot auctions.

Finally, this Thesis presented a requirement-based evaluation by defining multiple scenarios to define the behavior of the actors in the scenario *i.e.*, ETs. This allowed to investigate if the requirements for the system were met and how the system performs in different conditions. These experimental scripts demonstrated that the interaction requirements of SIRQ were satisfied. As the initial setup of the system became more complex, the increased level of interaction requires the implementation of strategies to maintain fairness among SIRQ participants.

6.2 Conclusions

This thesis designed a brokering platform including QM based on the analysis of current research and finally evaluated a prototype of the developed design. With the implementation of SIRQ, the targets were achieved while including a broader scope than the goals denoted in the introduction of this thesis. The key contributions are detailed as follows:

- Introduction of a QM system including a reservation function for improved preplanning of routes
- Inclusion of an auction function for spontaneous ability to jump the queue
- Automation and virtualization of real world charging flows for ETs
- Formalization of QoS features to conform with user expectation by introducing constraints
- Possible end-to-end integration with existing infrastructure by providing administrative features and using common interfaces

The literature review revealed that the use case for ETs is not represented, although the use case for EVCS is not fully applicable due to different prerequisites in terms of planning routes and acting less spontaneous when it comes to charging. This is why reservations are a complementary functionality added to SIRQ allowing to improve guarantees needed for planning a route when dealing with potential queues appearing.

In case of ad-hoc changes and time pressure, the ETs are dependent on minimal waiting times at an ETCS. Hence, the auction functionality allows to reduce the risks of this dependency by allowing a mechanism to jump the queue upon paying a surcharge directly to another ET currently occupying the CS. This interaction requires the ET to be willing to trade. The willingness cannot be enforced by SIRQ but the choice of auction scheme provides an incentive to cede the CS. Therefore, the auction scheme is *first-price sealed-bid* as this decreases the interaction needed by the ETs charging at a CS, due to the need to submit the offer only once per auction. The lowest price is considered the "winning" price to reduce costs for the auction starter.

The added interactivity between users, infrastructure and ETCS provider required the definition of assumptions, which support specific desired behavior patterns during the usage of an ETCS. Without well-defined assumptions, the scenarios tend to grow in complexity due to the random factor of a multi-user system with different expectations and strategies to improve the own situation.

These design choices and assumptions impact the user engagement needed, compared to a QM system without queue interaction functionalities. Hence, SIRQ carefully selects the participants in the auction to reduce unnecessary user engagement while still providing enough auction participants to increase the chance of success for the auction starter.

The evaluation of a scenario in which the auction could be used successfully resulted in a higher ET turnover when comparing the same period for a QM system, which does not provide a queue interaction functionality. SIRQ's design allows modeling a scenario without these functionalities, as the queue interaction is optional and users simply can refrain from using them if desired.

Implementing these queue interaction mechanisms involved deeply integrating a virtual representation of the physical actions executed by an ET at an ETCS to ensure that the interactive QM does not deteriorate the system's soundness. Hence, SIRQ implements an end-to-end flow including visual recognition of license plates and a model of a charging flow to efficiently inject interactivity at the desired moments of a charging flow.

To avoid that this end-to-end implementation introduces technical dependencies, the architecture is chosen to be open by using common interfaces *i.e.*, REST APIs over HTTP. This allows the prototype to be adjusted to the specific needs in the real world while also preserving ease of use to encourage further development.

6.3 Future Work

While SIRQ was thoroughly tested in complex scenarios during the evaluation of the prototype, it only indicates how it potentially behaves in a real world scenario. Hence, by deploying the system for a real ETCS the assumptions made could be verified. In addition, evaluating the system in such a scenario could test the constraints introduced and adjust them to fulfill the requirements imposed by the circumstances.

Furthermore, there is no clear causal connection between decreased waiting times and the auction mechanism, as the decreased waiting times were mainly driven by the enforced constraints introduced through minimal and maximal charging times. Hence, an integrated setup with an existing ETCS could investigate this aspect by testing the queue performance with and without the auction functionality and a QM system without any of these constraints implemented.

This overhaul might affect the technical complexity, which needs to be investigated, as the prototype's performance was tested requirement-based and therefore does not provide any quantitative data to assess the system's performance.

The intricacy arises primarily due to the choice of performing checks within the DB and as discussed in Section 5.7, the implementation of transactions may decrease the availability essential for a multi-user distributed network.

Although a proof-of-concept was provided to showcase the integration with existing systems, it is required to investigate the implications of communication between ETCS, truck companies and ETs, as ETs interact with ETCS but SIRQ assumes to communicate with them through the truck company as a middle man.

Alternatively, this does not need to be done if the system provides a functionality to handle the auction and reservation processes in an automated manner. It could be worth investigating how preset parameters could be used to automatically request auctions or request reservations when planning the route or diverting from it. In this case, the truck company could reduce the user engagement needed, which is a main drawback of the implementation of SIRQ.

Bibliography

[1] Stéphane Alarie and Michel Gamache. „Overview of solution strategies used in truck dispatching systems for open pit mines“. In: *International Journal of Surface Mining, Reclamation and Environment* 16.1 (2002), pp. 59–76.

[2] Gonçalo Alface, João C Ferreira, and Rúben Pereira. „Electric vehicle charging process and parking guidance app“. In: *Energies* 12.11 (2019), p. 2123.

[3] Ting Bai et al. „Distributed Charging Coordination for Electric Trucks under Limited Facilities and Travel Uncertainties“. In: *arXiv preprint arXiv:2407.10307* (2024).

[4] F Baker and G Fairhurst. *RFC 7567: IETF Recommendations Regarding Active Queue Management*. 2015.

[5] Marie Rajon Bernard et al. „Charging solutions for battery-electric trucks“. In: *ICCT, Washington, USA* (2022).

[6] Yue Cao et al. „An electric vehicle charging management scheme based on publish/subscribe communication framework“. In: *IEEE Systems Journal* 11.3 (2015), pp. 1822–1835.

[7] Emir Çabukoglu et al. „Battery electric propulsion: An option for heavy-duty vehicles? Results from a Swiss case-study“. In: *Transportation Research Part C: Emerging Technologies* 88 (2018), pp. 107–123.

[8] Gang Chen, Kannan Govindan, and Mihalis M Golias. „Reducing truck emissions at container terminals in a low carbon economy: Proposal of a queueing-based bi-objective model for optimizing truck arrival pattern“. In: *Transportation Research Part E: Logistics and Transportation Review* 55 (2013), pp. 3–22.

[9] European Commission. *Driving time and rest periods*. URL: `https://transport.ec.europa.eu/transport-modes/road/social-provisions/driving-time-and-rest-periods_en` (visited on Dec. 7, 2024).

[10] PostgreSQL Docker Community. *postgres - Official Image | Docker Hub*. URL: `https://hub.docker.com/_/postgres/` (visited on Dec. 25, 2024).

[11] Vicki M Coppinger, Vernon L Smith, and Jon A Titus. „Incentives and behavior in English, Dutch and sealed-bid auctions“. In: *Economic inquiry* 18.1 (1980), pp. 1–22.

[12] Tien Tuan Anh Dinh et al. „Blockbench: A framework for analyzing private blockchains“. In: *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM. 2017, pp. 1085–1100.

[13] Fadi Elghitani and Ehab F El-Saadany. „Efficient assignment of electric vehicles to charging stations“. In: *IEEE Transactions on Smart Grid* 12.1 (2020), pp. 761–773.

[14] David Ferraiolo and David Kuhn. „Role-Based Access Controls“. In: *15th National Computer Security Conference (NCSC)*. 1992.

[15] Nilgun Fescioglu-Unver, Melike Yıldız Aktaş, and Coşku Kasnakoğlu. „Feedback controlled resource management model for express service in electric vehicle charging stations“. In: *Journal of Cleaner Production* 311 (2021), p. 127629.

[16] the Fresh authors. *Fresh - The simple, approachable, productive web framework.* URL: `https://fresh.deno.dev/` (visited on Dec. 25, 2024).

[17] Drew Fudenberg and Jean Tirole. „Noncooperative game theory for industrial organization: an introduction and overview“. In: *Handbook of industrial Organization* 1 (1989), pp. 259–327.

[18] Ali Ghasemi-Marzbali et al. „Fast-charging station for electric vehicles, challenges and issues: A comprehensive review“. In: *Journal of Energy Storage* 49 (2022), p. 104136.

[19] Google. *ML Kit | Google for Developers.* URL: `https://developers.google.com/ml-kit/` (visited on Dec. 25, 2024).

[20] Andy Hayden. *denoland/deno - Docker Image | Docker Hub.* URL: `https://hub.docker.com/r/denoland/deno` (visited on Dec. 25, 2024).

[21] Yifeng He, Bala Venkatesh, and Ling Guan. „Optimal scheduling for charging and discharging of electric vehicles“. In: *IEEE transactions on smart grid* 3.3 (2012), pp. 1095–1105.

[22] Anna Herlt, Eugen Hildebrandt, and Henrik Becker. *Building Europe's electric-truck charging infrastructure.* URL: `https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/building-europes-electric-truck-charging-infrastructure` (visited on Dec. 7, 2024).

[23] Paris IEA. *Global EV Outlook 2024.* URL: `https://www.iea.org/reports/global-ev-outlook-2024` (visited on Jan. 8, 2025).

[24] ChargeFinder Inc. *ChargeFinder - Charging stations for electric cars (EV).* URL: `https://chargefinder.com/` (visited on Dec. 22, 2024).

[25] Oualid Jouini. „Analysis of a last come first served queueing system with customer abandonment“. In: *Computers & Operations Research* 39.12 (2012), pp. 3040–3045.

[26] Vladimir V Kalashnikov. *Mathematical methods in queuing theory.* Vol. 271. 2013.

[27] Karen L Katz, Blaire M Larson, and Richard C Larson. „Prescription for the Waiting-in-Line Blues: Entertain, Enlighten, and Engage“. In: *Sloan Management Review* 32.2 (1991), p. 44.

[28] David G. Kendall. „Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain“. In: *The Annals of Mathematical Statistics* 24.3 (1953), pp. 338 –354.

[29] Long Le et al. „The effects of active queue management on web performance“. In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. 2003, pp. 265–276.

[30] Craig David MacDonald, Lina Kattan, and David Layzell. „Modelling electric vehicle charging network capacity and performance during short-notice evacuations“. In: *International journal of disaster risk reduction* 56 (2021), p. 102093.

[31] Muhammad Shahid Mastoi et al. „An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends“. In: *Energy Reports* 8 (2022), pp. 11504–11529.

[32] R Preston McAfee and John McMillan. „Auctions and bidding“. In: *Journal of economic literature* 25.2 (1987), pp. 699–738.

[33] Valeh Moghaddam et al. „Dispatch management of portable charging stations in electric vehicle networks“. In: *ETransportation* 8 (2021), p. 100112.

[34] Moni Naor, Benny Pinkas, and Reuban Sumner. „Privacy preserving auctions and mechanism design“. In: *Proceedings of the 1st ACM Conference on Electronic Commerce*. 1999, pp. 129–139.

[35] John F Nash. „Non-cooperative games“. In: *The Foundations of Price Theory Vol 4*. 2024, pp. 329–340.

[36] H Pourvaziri et al. „Planning of electric vehicle charging stations: An integrated deep learning and queueing theory approach“. In: *Transportation Research Part E: Logistics and Transportation Review* 186 (2024), p. 103568.

[37] Kanniga Gayathri S.B et al. „On Spot Reservation and Parking System“. In: *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*. 2023, pp. 1–5.

[38] Maike Schwarz et al. „M/M/1 queueing systems with inventory“. In: *Queueing Systems* 54 (2006), pp. 55–78.

[39] Rathindra P Sen. *Operations Research: Algorithms And Applications: Algorithms and Applications*. 2010.

[40] Jeffrey Short, Alexandra Shirk, and Alexa Pupillo. „Charging Infrastructure Challenges for the US Electric Vehicle Fleet“. In: *December 2022 - Charging Infrastructure Challenges Full Report* (2022).

[41] Yongju Son, Hyeon Woo, and Sungyun Choi. „A study on the application of game theory for optimal operation of electric vehicle charging station“. In: *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2022, pp. 338–340.

[42] Daniel Speth, Verena Sauter, and Patrick Plötz. „Where to Charge Electric Trucks in Europe-Modelling a Charging Infrastructure Network“. In: *World Electric Vehicle Journal* 13.9 (2022).

[43] Xiaoqi Tan et al. „Optimal scheduling of battery charging station serving electric vehicles based on battery swapping“. In: *IEEE Transactions on Smart Grid* 10.2 (2017), pp. 1372–1384.

[44] Michael Ummels. *Stochastic multiplayer games: Theory and algorithms*. 2010.

[45] Eric Van Damme. *Stability and perfection of Nash equilibria*. Vol. 339. 1991.

[46] Mohammad M Vazifeh et al. „Optimizing the deployment of electric vehicle charging stations using pervasive mobility data“. In: *Transportation Research Part A: Policy and Practice* 121 (2019), pp. 75–91.

[47] Dan Xiao et al. „An optimization model for electric vehicle charging infrastructure planning considering queuing behavior with finite queue length“. In: *Journal of Energy Storage* 29 (2020), p. 101317.

[48] Yanhai Xiong et al. „Optimal electric vehicle fast charging station placement based on game theoretical framework“. In: *IEEE Transactions on Intelligent Transportation Systems* 19.8 (2017), pp. 2493–2504.

[49] Caiping Zhang et al. „Charging optimization in lithium-ion batteries based on temperature rise and charge time“. In: *Applied Energy* 194 (2017), pp. 569–577.

[50] Jie Zhu et al. „Planning of electric vehicle charging station based on queuing theory“. In: *The Journal of Engineering* 2017.13 (2017), pp. 1867–1871.

Abbreviations

AQM	Active Queue Management
CA	Central Authority
CIA	Confidentiality, Integrity and Availability
CS	Charging Station
DB	Database
DDoS	Distributed Denial of Service
ET	Electric Truck
ETCS	Electric Truck Charging Station
EV	Electric Vehicle
EVCS	Electric Vehicle Charging Station
FIFO	First In First Out
ICE	Internal Combustion Engine
JWT	JSON Web Token
LIFO	Last In First Out
UUID	Version 4 Universally Unique Identifier
REST	Representational State Transfer
QM	Queue Management
QoS	Quality of Service

List of Figures

List of Tables

Listings

Appendix A

Contents of the Repository

The code was provided in a Github repository `https://github.com/raffi07/mt-sirq` instead of a physical CD as agreed upon with the supervisor during the meetings in course of this thesis.

In the repository two folders are found:

- `mt-cs-sirq-app` contains the code for SIRQ
- `mt-cs-camera-infra` contains the code for the Android App to use the visual recognition and send the result to SIRQ.

A.1 Installation Guide SIRQ

The following guide is used to install SIRQ. In any case, please refer to the `README` found at the root of the repository for the most recent and extensive information about how to run it.

A.1.1 Install Docker

Run the following commands in a `bash` shell to install Docker.

- `curl -fsSL https://get.docker.com -o get-docker.sh`
- `sh get-docker.sh`

If you are not using a bash shell, please refer to:
`https://docs.docker.com/engine/install/`

A.1.2 Download Source Code

Download the source code from the following repository:

- `https://github.com/raffi07/mt-sirq`

Search for the commit with the `v.1.0.0` release tag to get the version of the application discussed in the thesis.

A.1.3 Fill `.env`-variables

Navigate to the downloaded source code of SIRQ and locate the `.env`-file. Open it with any text editor to fulfill it. Please refer to the `README` to fill in the different environment variables which are needed to run the application and provide an entry point to the application's API by defining the credentials.

A.1.4 Build & Run SIRQ

Navigate a shell instance to the top directory of the downloaded SIRQ source code and run

- `docker-compose up -build`

After a successful build, refer to `README` to start using the application.

A.2 Optional: Installation Guide Visual Recognition

The following guide is used to install the Android app for visual recognition. This is optional and not required to use all functions of SIRQ.

Move the `.apk` file to an Android phone and install it. Ensure that the installation of unknown sources is enabled in the settings.

For further instructions refer to the `README`.