Article

Not peer-reviewed version

A Reference Architecture for Smart Car Parking Management Systems

Mert Ozkaya [*]

Posted Date: 1 October 2024

doi: 10.20944/preprints202409.2462.v1

Keywords: Smart car parking; reference architecture; software architecture; multiple viewpoints

Article

A Reference Architecture for Smart Car Parking Management Systems

Mert Ozkaya

Department of Computer Engineering, Yeditepe University; mozkaya@cse.yeditepe.edu.tr

Abstract: Finding out an available car parking space has been one of the most crucial problems for the crowded cities, which causes such issues as traffic congestions, air pollutions, and stressful drivers. To mitigate the issues here, smart car parking management system (SPMS) solutions can be used which digitalise the parking processes and maximise the drivers' productiveness. However, the existing SPMS solutions are either domain-specific or address the applications of particular techniques and technologies. None of the solutions in the literature propose any generic approach that can be re-used for the analysis and design of any SPMSs. In this paper, we propose a reference architecture (RA) for the SPMS product family. Performing comprehensive domain analysis, we provide a feature model of the common and varying features for SPMSs. We designed 4 architectural viewpoints, which are context, module, component&connector, and allocation. Each viewpoint addresses a different concern and proposes a generic solution that can be re-used for any specific SPMSs. To validate our RA design, we used the commercial 4Park SPMS application and specified its architecture using our RA viewpoints. We strongly believe that our RA design for SPMSs can be useful for the SPMS domain who develop SPMS solutions.

Keywords: smart car parking; reference architecture; software architecture; multiple viewpoints

1. Introduction

The world population is now more than 8 billion and expected to exceed 9 billion by 2040. Hundreds of mega-cities with 10+ million populations have been recorded so far [1]. So, such crowded cities suffer from traffic congestions that bring many disadvantages such as stressful life, air pollution, waste of time, and accidents [2]. While variety of local transportation options have been offered in the mega cities to reduce traffic congestions, most of the people still choose to use their own cars. Indeed, more than 1.4 billion cars have been recorded to be used in the world and this is expected to be doubled by 2040 [3]. One of the most crucial problems that contribute to the traffic congestions is the drivers who look for parking spaces to park their cars. A recent survey study [4] reveals that drivers spend in average 14 minutes to find a suitable parking space in each cruise. In another study conducted by INRIX [5], UK drivers have been found to spend 44 hours a year so as to find an available parking space. In USA, drivers have been found to spend 17 hours a year for the available parking spaces [6]. This essentially indicates not only the unproductive use of the drivers' time but also wasting fuel and causing air pollutions. Another interesting finding indicates that 30% of the traffic flows result from the drivers who look for available parking spaces [7].

To mitigate the issues with finding the available car parking places, the processes of car parking should be digitalised using smart techniques and technologies [8]. Many smart car parking management systems (SPMSs) have been developed so far through which various facilities can be performed including the online prior reservation, online payment, finding the closest/convenient car parking space automatically, navigating to the empty parking space, and integrating with other applications such as fuel payment and car charging applications. Thanks to such facilities that support the automated decision making, the time that drivers spend for finding any available parking spaces can be minimised, which will reduce the carbon emission and traffic congestion. Moreover, the facilities such as online reservation and payment, navigating to the parking area make it easier for the drivers and park attendants to manage the car parking effectively.

SPMSs are essentially complex systems [9], which are used by different stakeholders (e.g., drivers, park attendants, municipalities, etc.) for various purposes and integrate with various external systems (e.g., payment systems, data analytics systems, sensors, car charing systems, etc.). Also, SPMSs may consist of several different applications (e.g., navigation, reservation, street map, parking lot designer, etc.) each of which is designed with several software modules that are interrelated. Therefore, it is highly crucial to adopt a software architecture-centric perspective for the design and development of SPMSs. Indeed, software architecture is essentially the technique for managing the complexity of software systems, which focuses on decomposing software systems into components and mapping those software components into the physical components, the interaction details amond the components, and making design decisions for meeting the quality requirements to be satisfied [10].

To faciliate the architecture design of software systems, the notion of "reference architecture (RA)" can be used [11], which is the generic design of software architecture for a particular problem domain and can easily be re-used for designing the architectures of software systems in that domain which have commonalities and variations. An RA design can address different architectural concerns such as system boundaries, logical and physical architectures and offers a set of generic concepts, rules and constraints that can be used for designing a solution for a partciular problem domain. Therefore technical and non-technical stakeholders can easily communicate on the needs of any product using the RA design and quality software products that better meet the requirements can easily be designed and developed with the least effort and budget.

Given the cruciality of car parking, many different technqiues and technologies have been provided for digitalising car parking. Indeed, the literature includes several works on the design and development of SPMSs. As discussed in Section 2, the existing studies either *(i)* introduce specific SPMS applications, *(ii)* discuss the applications of particular data analytics, information or communication technologies for SPMSs, or *(iii)* discuss the analysis of the existing SPMS applications for some features of interest. However, none of the existing studies focus on the reference architecture design for SPMSs and consider providing an architectural guideline for the effective design of SPMS solutions.

In this paper, we aim at designing a reference architecture (RA) for SPMSs that can be used for designing application-specific SPMS architectures. We initially provide a feature model that indicates the common and varying functionalities and quality characteristics for SPMSs. Then, we design the RA in terms of multiple architectural viewpoints for managing the complexity, which are context, module, component&connector, and allocation. We believe that our RA design for SPMSs will be very useful for different stakeholders. Technical and non-technical stakeholders involved in a specific SPMS development can communicate on the product needs using our RA design and better decide on their application requirements and any design decisions to be considered for meeting those requirements. Also, software developers (or architects) can use our RA viewpoint definitions so as to design their application architecture (AA) for a specific SPMS product. Indeed, the architecture design viewpoints can be extended and adapted to the specific product needs by adding/removing new/existing elements. Moreover, software development companies that offer SPMS solutions can understand the missing features of their SPMSs and the architectural perspectives for addressing those features. Lastly, researchers can use our RA design to initiate research projects that develop smart, innovative solutions for car parking.

2. Related Work

Given the ever-increasing populations in big cities and the number of cars looking for parking places, the smart car parking problem has been shown great interest by the research communities in the recent ten years. Indeed, the literature includes tens of different studies about smart car parking, which shed light on different concerns.

Some of those studies in the literature propose the applications of different data source techniques and software techniques and technologies for the smart car parking. These include the approaches that *(i)* apply dynamic resource allocation algorithms [12], *(ii)* use cloud computing [13], *(iii)* use ultrasonic sensors for detecting empty parking spaces [14], *(iv)* use convolutional neural network for detecting empty spaces [15], *(v)* use genetic algorithm for scheduling the cars [16], *(vi)* use fog computing for desinging the smart parking system architectures [17], and *(vii)* use the RFID technology [18].

The literature also includes several different survey studies that aid in understanding and comparing the existing car parking solutions from diverse perspectives. Fahim et al. [19] compare a huge set of SPMSs for a set of features which includes the software technologies used (e.g., AI technologies), the sensor devices supported, different computational approaches, networking technologies, and user interface types. Paidi et al [20] analyse different sensor devices for collecting parking data and compare the SPMSs with regard to their support for different sensor devices. Lin et al. [8]'s survey reveals a comprehensive analysis of a set of SPMSs for a large feature set. Lin et al.'s feature set is mainly categorised into three groups, which are information collection, system deployment, and service dissemination, and each group is further divided into sub-groups. Xiao et al. [21] address the car parking prediction problem of finding an empty parking space. Xiao et al. compare different prediction methods in terms of their strengths and weaknesses, analyse the prediction problems, and suggest future research to tackle with the challenges on the car parking prediction. Barriga et al. [22] analyse a huge set of papers that discuss smart parking solutions for three main features. These features are the types of sensors used, the network infrastructure, and the services that the smart parking solutions offer for their users (e.g., information management, e-parking, and parking space prediction).

Given our focus on architecting SPMSs, we also observed some studies in the literature that discuss the architecture design of SPMS solutions. Fraifer et al. discuss a prototype-centric design and development of their SPMS and focus on understanding the user needs [23]. Anderson et al. discuss the architecture for a specific SPMS based on the fog computing [24]. Anderson et al. focus on the vehicle detection and propose a set of component types that can be used to design the SPMS architecture for secure anf fast communication. Soegoto et al. give brief information about the hardware and software units that can be composed for an SPMS [25]. Soegoto et al. focus on the car detection and developed a prototype about their solutions. Awaisi et al. discuss an architecture framework for SPMSs which is based on fog computing [26]. Awaisi et al. proposed a layered architecture consisting of 3 layers. The bottom layer consists of the camera devices (as the data sources), the middle layer consists of the fog nodes for collecting data from camera devices and processing them for detecting the parking space occupancies, and the top layer includes the cloud server for storing the processed data. Alharbi et al. proposed a web-based framework for the smart car parking [27]. In their work, Alharbi et al. briefly describe the algorithm for detecting car plates through which the car parking gate can be opened/closed and also . Alharbi et al. also discuss their web-based tool implementation, which provides such services as booking online for any parking space, managing the bookings, managing the visitors data, and viewing some reports.

While the literature includes diverse studies on the smart car parking, none of the existing studies essentially address the problem of proposing a generic, re-usable solution for the smart car parking design and development. Therefore, considering the importance of smart car parking for the cities with ever-increasing population, we consider this as a crucial issue that needs to be addressed. Indeed, the literature already reveals many RAs for different domains and industries such as healthcenter information systems [28], smart farming [29], autonomous driving [30], digital twin [31], no any similar works have been conducted for SPMS so far. Considering the complexity of SPMS solutions, we consider the lack of RA for smart parking as a gap in the literature and aim in this study to bridge this gap.

3. Research Questions

In our research, we aim to design an RA for the SPMS product family. To this end, we investigate 4 key research questions that are given below.

- RQ1: Is it possible to analyse the smart car parking domain so as to determine the common and varying features for the SPMS products?

- RQ2: Is it possible to determine the stakeholders who/which interact with SPMSs and their responsibilities?
- RQ3: Is it possible to design an RA for SPMSs in a manageable way for the facilitated understandability and re-usability?
- RQ4: Is it possible to re-use the RA design for specifying application-specific SPMS architectures?

We address RQ1 in Section 4, where we discuss the results of our domain analysis for the smart car parking and introduce our feature model of common and varying features. We address RQ2 in Section 5.1, where we discuss the system boundaries for SPMSs in terms of the stakeholders and their system interactions. We address RQ3 in Section 5, where we discuss our RA design in terms of multiple viewpoints. Lastly, we address RQ4 in Section 6, where we discuss the AA specification of a real SPMS by means of the re-use of the RA design.

4. Domain Analysis for SPMSs

To understand the problems about car parking management, we aimed to obtain the domain knowledge about car parking and analyse the collected information. We initially performed a literature review and determined a list of papers about smart car parking management. We considered the papers that *(i)* analyse the existing park management tools for relevant innovative techniques and technologies, e.g., [8,19,22], *(ii)* introduce smart parking management solutions, e.g., [32–34], and *(iii)* discuss the applications of some crucial techniques and technologies, e.g., [35–37]. We further analysed the commercial smart car parking management technologies, e.g., Fleximodo[1], Parkable[2], and 4Park[3]. In total, we have reviewed around 50 different papers and 10 different commerical smart car parking tools. So, were able to identify the potential stakeholders who are involved. We also identified all the functionalities and quality properties that are addressed in the selected list of papers and tools and their level of importance.

Besides, we conducted a face-to-face survey with a group of parking attendants who work for ISPARK[4], which is a subsidiary for the municipality of Istanbul, Turkey and responsible for mananging the car parking services in Istanbul. The parking attendants do not currently use any mobile applications. They rather direct the drivers who enter their street to the empty parking places (if any) and charge the drivers depending on their estimated time of departure. The park attendants are responsible for detecting any drivers who exceeded the stated departure time and charging them extra. We asked 15 questions to 15 different staffs so as to understand the challenges that they face while managing car parking without the use of any smart car parking system and their expectations from a smart system[5]. The survey results helped us validating the functionalities and quality expectations that have been determined from the papers and tools. Indeed, we understood how important smart technologies are for drivers and park attendants in managing their parking processes and analysed the functionalities that are crucial and those that are optional for drivers and park attendants.

After collecting and analysing the domain data about car parking management, we were able to identify the stakeholders and their responsibilities that are shown in Table 1. The driver acts as the main user who interacts with an SPMS so as to perform the driver-related tasks, e.g., finding out an empty parking space, booking and payment, and navigating to the empty space processes. The park attendant interacts with an SPMS so as to control the parking processes, e.g., confirming payments and checking parking spaces. The municipality staff and parking provider interact with an SPMS so as to perform administration tasks such as monitoring, generating and viewing useful reports, budget management, tariff management, billing and payment management, etc. The software developers

[1] Fleximodo: https://www.fleximodo.com/
[2] Parkable: https://parkable.com/
[3] 4Park: https://smartparkingsystems.com/
[4] ISPARK: https://ispark.istanbul/
[5] The survey questions are accessible here: https://zenodo.org/records/12704679

are responsible for designing and developing an SPMS with regard to the system requirements and managing the software testing processes. The administrator is concerned with administering an SPMS and performing any maintenance tasks such as bug fixing and adding new features. The data sources acquire data from environment such as the data for identifying the occupancy of any parking place. Government systems and any other external systems can integrate with SPMSs so as to provide some services or require some data such as parking fine management, fuel management, and charging station management applications.

Table 1. The stakeholders involved in SPMSs and their goals.

Stakeholder	**Goal**
Driver	To use SPMS and perform driver-related tasks
Park Attendant	To use SPMS and control the car parking processes
Municipality	To use SPMS and get informed about the car parking processes
Parking Provider	To use SPMS and get informed about the car parking processes
Software Developer	To design, develop and test SPMS
Administrator	To administer and maintain SPMS
Data Scientist	To analyse the data maintained by SPMS
Sensor Data Source	To provide data about the parking place occupancy
Camera Data Source	To provide data about the parking place occupancy
Government	To integrate government e-services and require/provide data
External System	To integrate with the car parking management systems

To model the functionalities and quality properties determined via the domain analysis, we consider FODA's feature modeling methodology [38]. Feature modeling promotes the specifications of common, alternative, and optional features that represent the user-visible functionalities and quality characteristics for a product family. So, a feature model for a product family indicates the features that any product of that family must possess and those features that are optional and thus does not always have to be supported. Feature models are specified as a tree-form in a hierarchical way where any features can be described in terms of its sub-features. Figure 1 shows the feature model that we specified for the SPMS product family. The mandatory features that are expected to be provided by any SPMS are parking lot, parking map, security and data management. Parking lot serves for designing the parking layout and editing the car parking spaces for different needs such as editing the car enterance and flow directions, accessible parking areas, and locating parking spaces around other physical objects. Parking layout is also supported with another sub-feature for the optimisation of the car parking places. Parking map serves for displaying the parking spaces on a visual street map and provides many functionalities such as offline parking map, online parking map that shows the parking occupancies at real-time, exporting selected parking area as an image file, saving parking locations, editing locations (e.g., adding photos), sharing parking locations in social media, and facility management (e.g., viewing the nearest hotels, restaurants and cafes). Note that the online parking map and real-time parking lot occupancy monitoring are the mandatory sub-features for the parking map feature. The driver feature serves for managing the driver profile and enabling drivers to *(i)* view some useful reports about their system usages, *(ii)* manage the notifications that they want to receive, their preferences, the payment methods, and their memberships, *(iii)* view their parking history, and *(iv)* get help&support. The security feature is for securing the system data and can be considered in terms of the important principles of information security (also called as "CIA triad"), which are confidentiality, integration, and availability. The data management feature supports collecting data from data sources (e.g., sensors, camera, and user interfaces), storing the collected data persistently/temporarily, processing the collected data, making decisions using data analytics techniques (e.g., machine learning), and visualising the decisions.

Besides the mandatory features, Figure 1 show various features that are considered optional for SPMSs. These are navigation, reservation, user profile, and administration. The navigation feature serves for finding the optimal route from the driver's current location to the selected parking place and can furher enable other facilities such as route saving, voice assistance, and closest facility routing. The reservation feature serves for reserving any available car parking place for a particular time slot, making payment for any available parking space, and other facilities such as cancellation, invoicing, applying discounts, and receiving notifications. Lastly, the administration feature serves for administering the SPMS and enabling the administrators to *(i)* monitor the system for some indicators, *(ii)* obtain reports about e,g., users, payments, parking areas, etc., *(iii)* manage users (e.g., adding/removing users), *(iv)* manage the billing&payment, discounts, tariffs, parking fines, and budget, and *(v)* plan and schedule for any system maintenance.

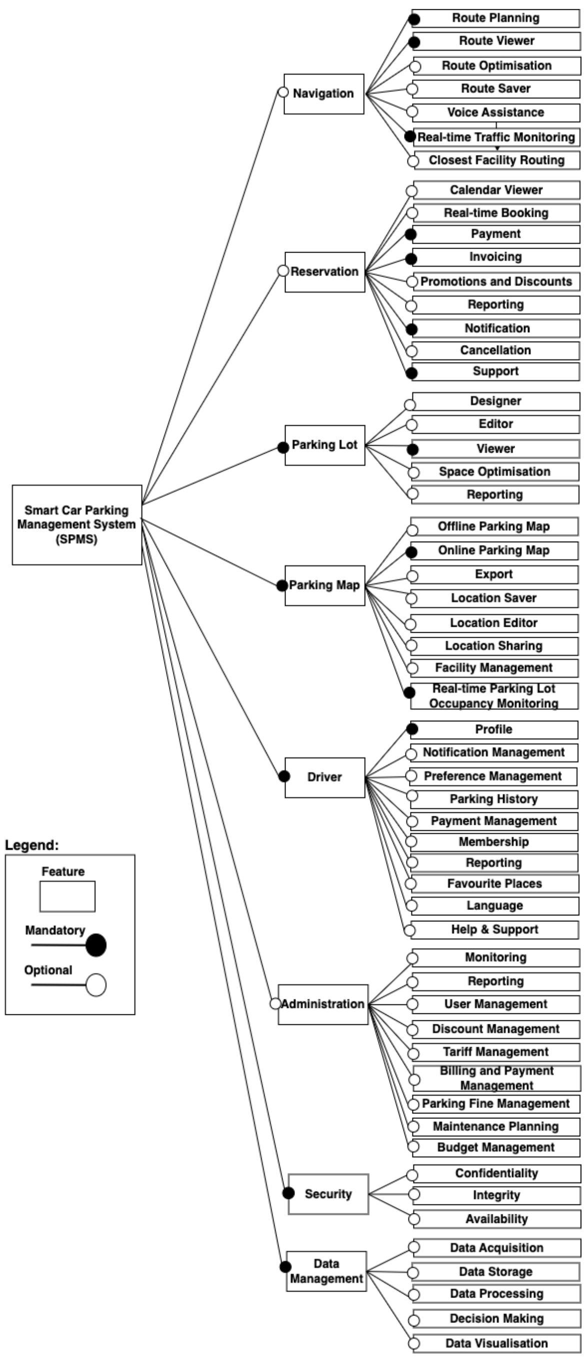

Figure 1. The feature model for SPMS.

5. Reference Architecture (RA) Design for SPMS

A software architecture design can be complex and encompass the solutions of different concerns that are relevant to different stakeholders (e.g., hardware designer, software designer, and programmer). To manage the complexity of software architecture, multiple viewpoints modeling can be adopted, where the solution designs for different concerns (e.g., systems' decomposition into software unuits and physical architecture) are separated and thus better managed and understood [39]. Here, each viewpoint addresses a particular concern and proposes *(i)* a particular method of dividing the system into parts and *(ii)* any rules and constraints so as to achieve the stakeholders' goal.

To design an RA for the SPMS product family, we focussed on the viewpoints that are encouraged by Clements et al.'s seminal book on architecture viewpoints [40] and thus commonly accepted by the software architecture communities. These are the context, module, component&connector and allocation viewpoints. The context viewpoint is concerned with establishing the system boundaries, the stakeholders' communications with the system, and determining the integration with external systems. The module viewpoint is concerned with decomposing the system into implementation units where each unit is supposed to realise a cohesive set of responsibilities (e.g., functionalities and quality expectations). The component&connector viewpoint is concerned with the runtime behaviour of the system, where the components represent the running units of the system and connectors represent the protocols of interactions among the components. To organise the components and connectors composing a system, we considered here the layered architecture style. Indeed, we strongly think that the layered style fits very well with such types of information systems as SPMSs where the components are typically separated into the layers for user interaction, business logic and data layers. The allocation viewpoint is concerned with mapping the software components into the physical devices that are physically connected.

An RA design for SPMSs can be re-used for the AA of a particular SPMS solution and the AA design consists of the view specifications derived from RA's viewpoint definitions. As illustrated in Section 6, any view specification can extend the viewpoint definition that it derives from by adding new elements or removing existing elements as long as the viewpoint rules and constraints are not violated.

5.1. Context Viewpoint

The context viewpoint is depicted in Figure 2, which shows the stakeholders that can interact with an SPMS. We consider two types of stakeholders, mandatory and optional stakeholders. The mandatory stakeholders must exist for each SPMS and those are the software developer, administrator and driver stakeholders. The optional stakeholders are not essential and their existences depend on the requirements for the SPMS solution to be developed. The optional stakeholders can be data scientists, park attendants, municipality staffs, research institutes, any external systems such as the data sources (sensors and camera), e-government systems, and electric car charging systems.

We categorise the communications between stakeholders and an SPMS as optional and mandatory. Any mandatory communication links are always established for the data exchange between the associated stakeholders and the SPMS. However, the optional communication links do not have to be established for each SPMS. Indeed, not all SPMS have to send updates to the system provider or administrator, while each administrator manages the system. Also, the stakeholders can perform either one-way or two-way communication with an SPMS. For instance, the municipality staff can manage the system and at the same time receive any updates from the system. So, while both communications are optional here, Concerning the driver, he/she needs to use the system and that is considered as a mandatory relationship. However, the driver may receive updates from an SPMS and that is an optional relationship which may be the case if it is required by an SPMS under development.

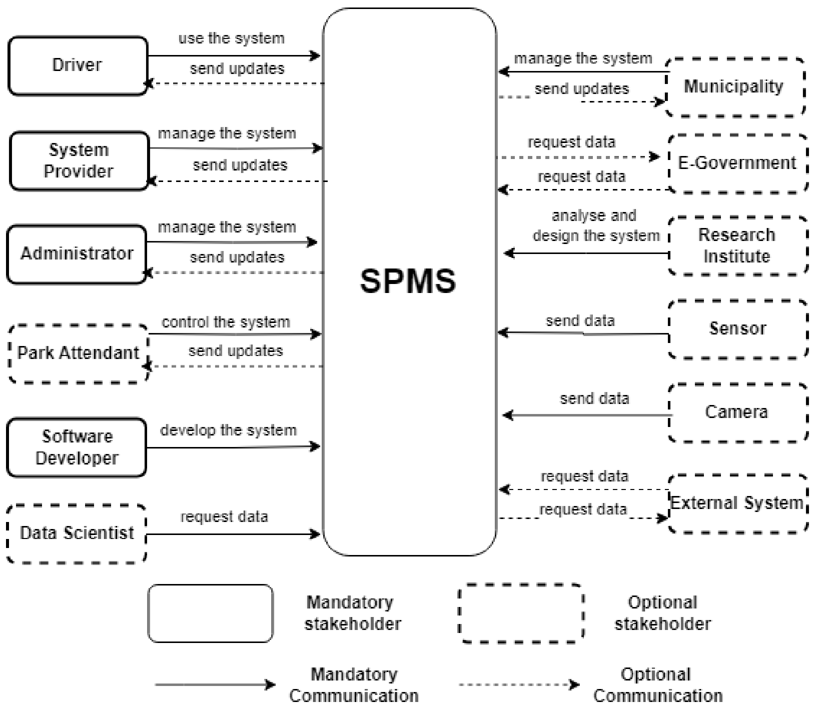

Figure 2. The context viewpoint for SPMS.

5.2. Module Viewpoint

The module viewpoint for our RA design is depicted in Figure 3, which shows the implementation units that can be designed and developed for an SPMS. Here, we used the feature model given in Figure 1 and performed the decomposition of 8 different main units each of which consists of modules (and sub-modules).

The navigation unit is to do with the functionalities about navigating to the available car parking spaces safely. The navigation unit consists of four main modules which are voice assistance, real-time traffic monitoring, closest facility management, and route management. The route management module here are further decomposed into a cohesive set of sub-modules which are route planning, route viewer, route searcher, route saver, route optimisation, and routing for the closes facility. The reservation unit is to do with the functionalities about reserving an empty car parking place. The reservation unit consists of several modules, which are calender viewer, real-time booking, payment, promotions & discounts, cancellation, notification, invoicing, reporting and support. The parking lot unit is to do with the functionalities about designing the parking lots that can be viewed by the drivers and includes such modules as lot designer, editor, lot viewer, optimisation for using the parking space effectively, and reporting. The parking map unit is to do with the functionalities about the map that shows the streets with empty/occupied parking spaces and any other facilities (e.g., restaurants, cafes, hospitals, etc.). The parking map is decomposed as the modules for exporting to an image file, facility management, parking lot occupancy monitoring, and location management. The map and location management modules are further decomposed into sub-modules. The driver unit is to do with the functionalities that any drivers can perform for their user accounts. The modules considered here includes the profile module, membership, preference management, notification management, billing management, favourite places, parking history, help & support and language. The administration unit is to do with the functionalities for administering the SPMSs by the relevant stakeholders (e.g., administrator, parking providers, and municipality staffs). These modules include the user management (e.g., adding/deleting users), monitoring the system performance, reporting, payment management (e.g., invoicing, secure payment, etc.), and tariff and discount management. The security unit is for ensuring the security properties to be satisfied so as to keep the data maintained by an SPMS safe and secure. The authentication module is for ensuring that the data is accessed by the correct person under the correct access rights. The integrity is for ensuring that the system data is always maintained as complete, consistent, and correct. The availability module is for ensuring that the data is always accessible for the authenticated users. Lastly, the data management unit consists of five different modules for collecting data from the environment, storing the collected data, processing those data using different data analytic techniques, making decisions using the processed data and then visualising the decisions.

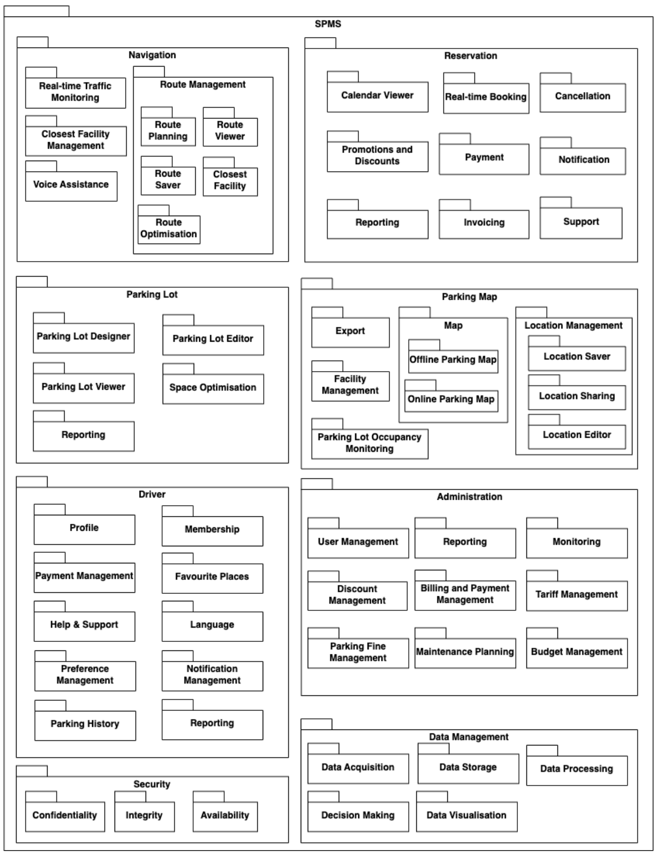

Figure 3. The module viewpoint for SPMS.

5.3. Component & Connector Viewpoint

To specify the components that represent the processing units for an SPMS and their interactions (i.e., connectors), we consider the layered architecture style [41], which is one of the most widely used style for information management systems such as SPMSs and suits best with the organisation of SPMS components. With the layered architecture style, components are organised into horizontal layers where each layer has a specific goal (e.g., the presentation layer for visualising the user interfaces to the end users) and consists of a cohesive set of components that aim to achieve the layer's goal. Connectors here are considered as simple connectors that act as links between layers and enable the data communications (or service requests) between the components of any two connected layers. Any layer may be connected with the layer right below and the components in the above layer may send data to the components in the layer below or request services.

Given the module viewpoint definition in Section 5.2, the software units and their modules that are considered for SPMSs can be used so as to determine the executing components of an SPMS. Each component can essentially execute one or multiple software units/modules that are relevant to each other. Figure 4 shows the application of the layered style for the SPMS domain. Therein, we consider the presentation layer at the top which encapsulates the components for executing the visual user interfaces in different platforms (i.e., Web and Mobile). The middle layer is the business logic layer where the components for executing the software units/modules about the SPMS functionalities that are discussed in Section 5.2 are encapsultated. The busines logic components are grouped into 6 categories that represent the main executing components of SPMSs at run-time. These are are navigation, reservation, administration, parking map, parking lot, and driver. Each component here implements the relevant software units/modules that perform the associated functionalities (e.g., navigation component implementing the navigation software unit). Upon receiving requests from the presentation layer components, the business logic layer components handle the requests and take any necessary actions which may require any data to be requested from the data management layer components. The bottom layer is the data management layer, whose components implement the data management software units/modules from the module viewpoint. The data management layer thus consists of 5 different components, one for acquiring data from diverse data sources (e.g., sensors and camera), another for storing the collected data, another for processing the acquired and stored data using the data analytic techniques (e.g., artificial intelligence and statistical techniques), another for supporting the decision making using the processed data, and lastly one for visualising the decisions. Lastly, we specify the security layer that is supposed to implement the security software unit from the module viewpoint. So, the security components here receive requests from all the layers so as to ensure the security of the data maintained by an SPMS.

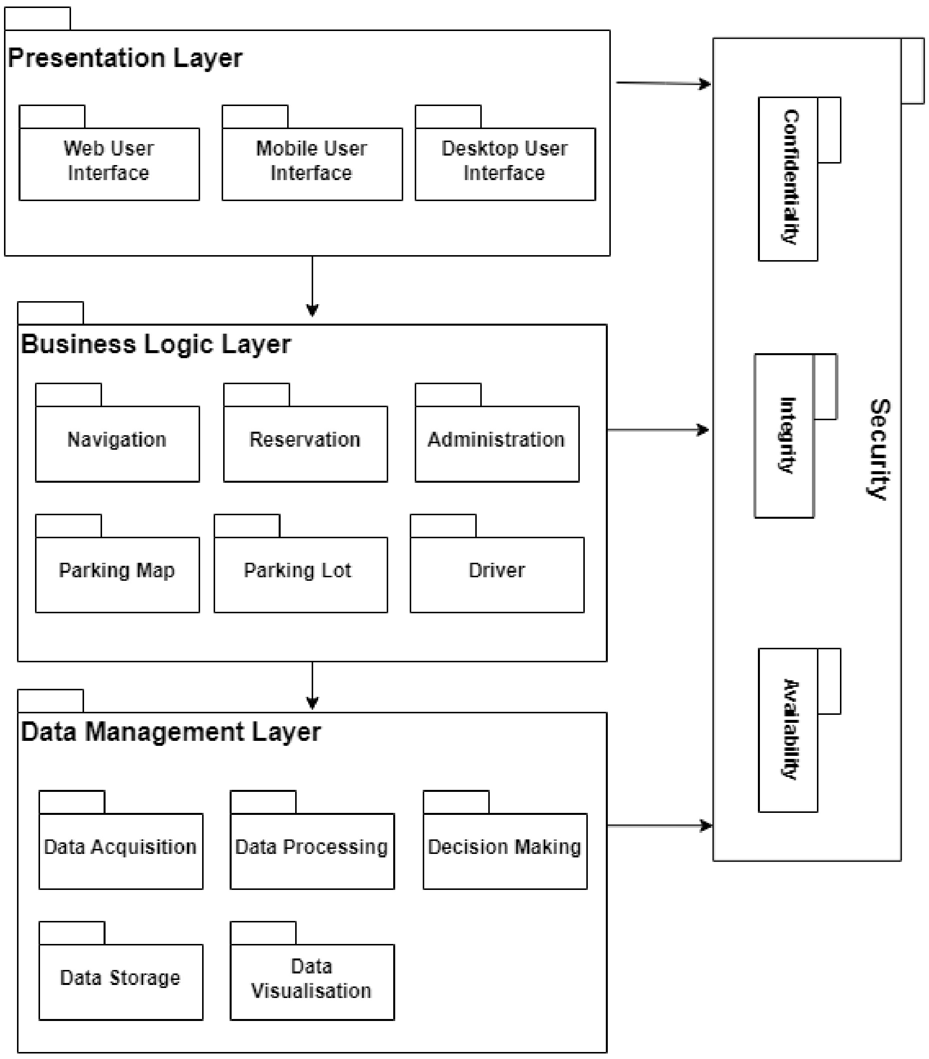

Figure 4. The component & connector viewpoint for SPMS (layered style).

5.4. Allocation Viewpoint

In the allocation viewpoint, we focus on mapping the software components from the component&connector viewpoint into physical components. We consider the client-server architecture style and focus on two types of physical components that are client and server. Any client device runs the software that performs requests for obtaining data or getting some tasks to be done such as the mobile devices for drivers and PC machines for parking providers. The server device runs the software that provides the SPMS business services, makes calculations, and shares the resulting data upon receiving requests from the client devices. Note that the server here can be deployed physically on a physical device or virtually on a cloud environment. The server and client devices communicate over network using the appropriate communication protocols e.g., HTTP and TCP.

Figure 5 depicts the allocation viewpoint. The presentation layer components (e.g., mobile, web or desktop applications) from the component & connector viewpoint in Figure 4 are mapped into the client devices, while the business logic and data management layer components are mapped into the server devices.

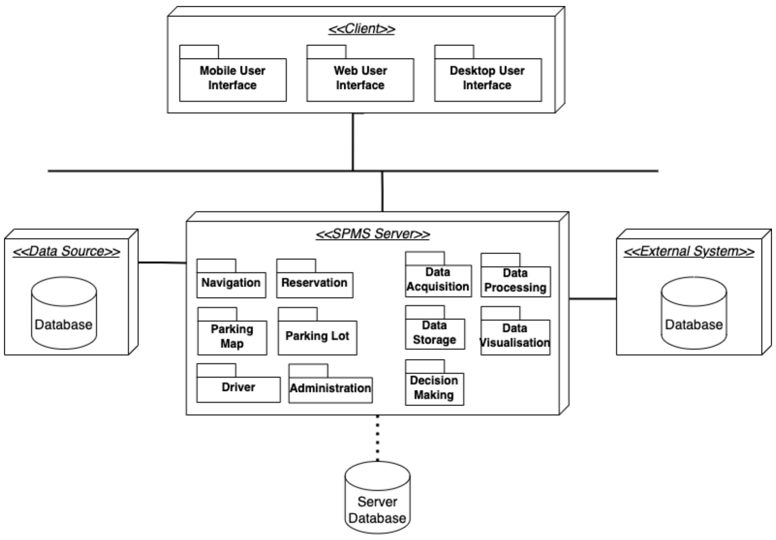

Figure 5. The allocation viewpoint for SPMS.

A separate client device can be specified for each type of SPMS user (e.g., driver, park attendants, and parking providers). Also, different types of devices such as KIOSK can be specified as a separate client device. Multiple server devices can be specified for the physical architecture of any SPMSs. Indeed, backup servers or different types of servers (database server, web server, etc.) can be specified.

In any physical architecture, an intermediate load balancing software applications can also be specified for directing the requests to different servers depending on the current loads of the servers. Furthermore, any data sources (e.g., sensor and camera devices) can be involved in the physical architecture so as to share data with the servers at real-time. Lastly, any external systems (e.g., electric car charging applications and fuel payment applications) can be involved, which can handle specific tasks that are not supported by the system servers and exchange data with the server(s). The data sources and external systems can have their own local database and interact with the server for communicating their data or providing any services.

6. Case Study Evaluation

To validate our RA design for SPMS, we considered the 4Park SPMS[6] which is one of the top used commercial tools for smart car parking and has been developed by the Smart Parking Systems company

[6] 4Park Website: https://smartparkingsystems.com/en/4-park-smart-parking-app/

based in Italy. Our goal here is to specify the AA for 4Park by re-using the RA design introduced in Section 5. To obtain the domain knowledge, we used the 4Park web-site where case-studies, videos, tutorials, white papers, and technical support contacts (e.g., e-mail and phone) are easily accessible. Firstly, we used the feature model specified with the RA design and ended up with a sub-set of features that 4Park supports. Then, we used the RA viewpoints and specified the views that are specific for the 4Park AA.

6.1. Feature Model

Figure 6 depicts the feature model for the 4Park SPMS. The feature model here includes all the features specified for the RA design. Moreover, the parking map feature includes an additional sub-feature for managing the parking locations for disabled people (colored pink)[7].

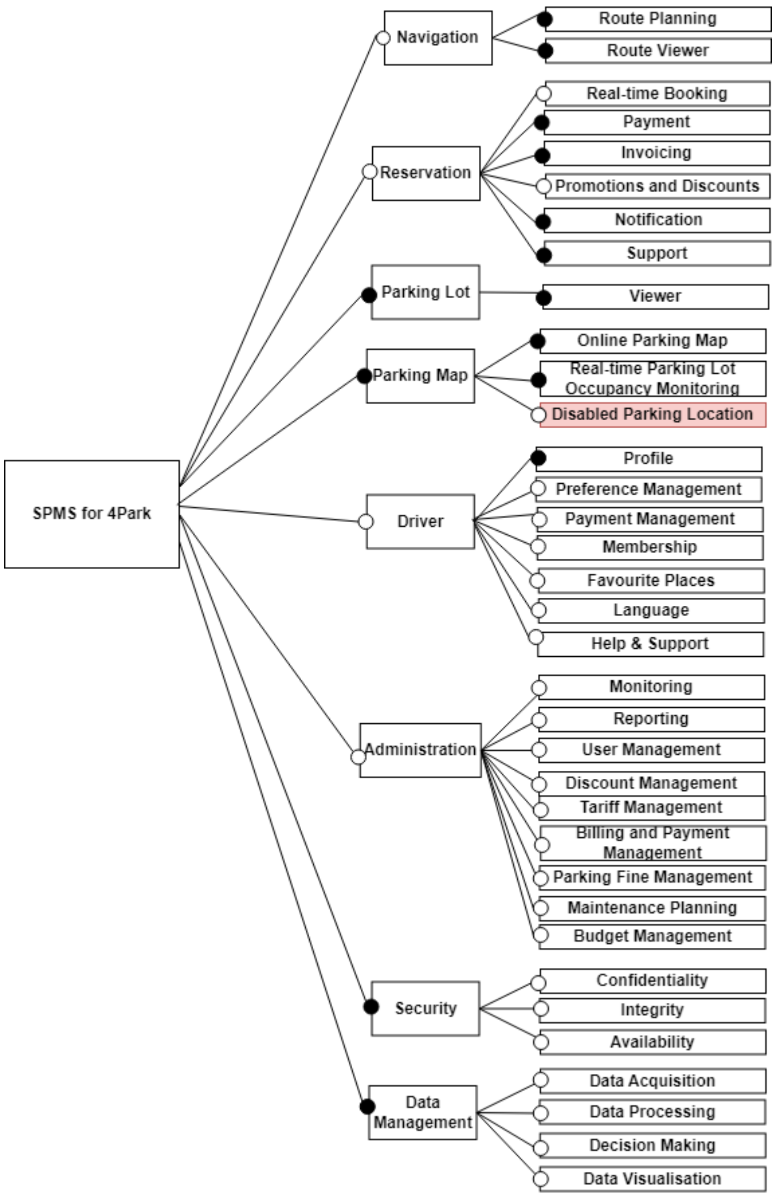

Figure 6. The feature model for 4Park.

[7] The sub-feature for the disabled parking location management: https://smartparkingsystems.com/en/disabled-parking-management/

While all the features considered by the RA design have been re-used in 4Park, not all the sub-features have been re-used. The exceptions here are the administration, security and data management whose all sub-features are re-used by 4Park. However, the navigation feature for 4Park is just considered in terms of route planning and viewer. Also, the parking lot feature does not seem to support designing and editing parking lots. Similarly, the parking map feature does not support location saving, editing, and sharing, and the management of facilities (e.g., nearest hospitals and cafes). In total, 62% of the sub-features that are considered by the RA design are used by 4Park.

6.2. Context View

The context view for 4Park is modeled in Figure 7. Here, the newly introduced stakeholders that are not existing in the RA's context viewpoint is colored pink, while those stakeholders that already exist in RA's context viewpoint with a different name is colored blue. So, the SPMS product for 4Park is accessed by the driver, system manager, parking controller, police, administrator, and municipality staff. The system manager here corresponds to the system provider stakeholder in the RA's context viewpoint and the parking controller corresponds to RA's park attendant. The sensor and camera devices send data to the SPMS. The external system involved in 4Park is the electric car charging system, which works as integrated with the 4Park SPMS server. 66% of the stakeholders that are defined in the RA context viewpoint depicted in Figure 2 are used by 4Park. 4Park additionally involves the police stakeholder who can be notified about any issues through the administration features of the SPMS.

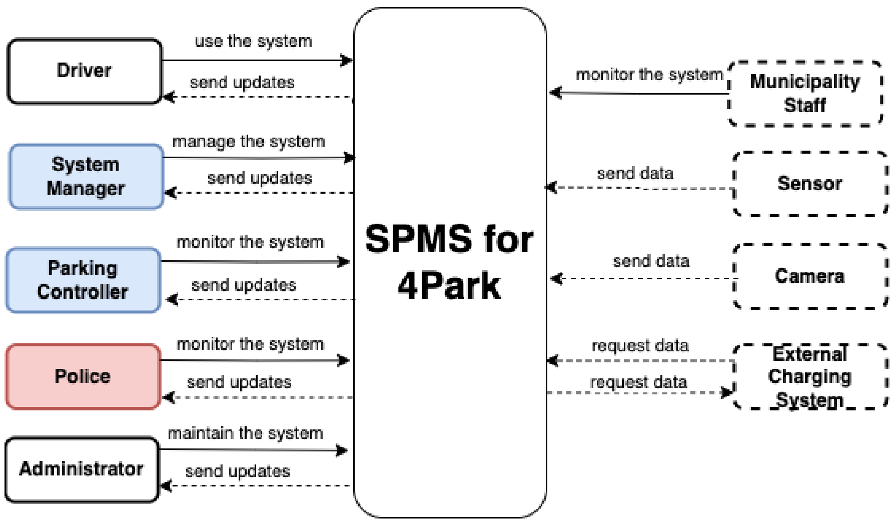

Figure 7. The context view for 4Park.

6.3. Module View

The module view for 4Park is modeled in Figure 8. Considering the RA's module viewpoint discussed in Section 5.2, 4Park's module view has several differences. Since some of the sub-features defined in RA are not supported by 4Park as discussed in Section 6.1, the associated software modules are not specified in the corresponding software units. For instance, the navigation unit and parking lot unit each include one software module only, despite the RA design considering several modules (and sub-modules). Note that 4Park includes a different software unit called "Management Software POLIS" which we consider as the composition of the existing Data Management software unit with the "Administration" software unit. Management Software POLIS provides services for the parking controller, system manager, police and administrator stakeholders and serves for analysing the data maintained by system so as to perform real-time monitoring and therefore produce useful reports and notifications. Also, the "parking map" software unit includes a new module called "disabled parking location" whose goal is to enable disabled people to reach the convenient parking locations.

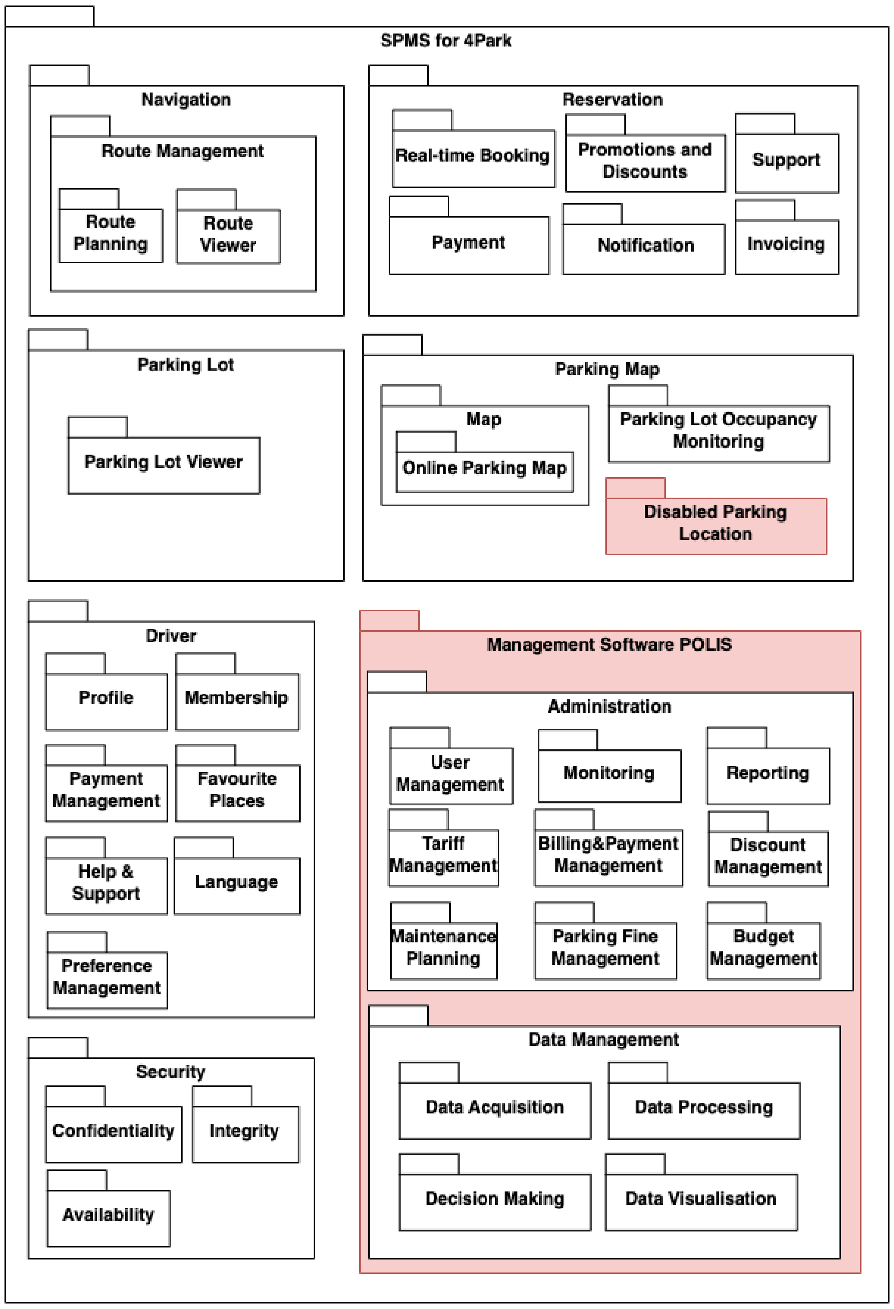

Figure 8. The module view for 4Park.

6.4. Component&Connector View

The component&connector view for 4Park is modeled in Figure 9. Given RA's component&connector viewpoint in Section 5.3, the view specification for 4Park includes an additional layer between the business logic and data management called "Management and Monitoring Layer". The management and monitoring layer encapsulates a new component that executes the newly introduced software unit called "Management Software Polis" specified in the module view depicted in Figure 8. The management and monitoring layer therefore offer services related to the real-time monitoring of the car parking system and the reporting facilities, which can be requested by the presentation layer components and business logic layer components. The management and monitoring layer reaches the data maintained by the data management layer and produces knowledge that can be used by the presentation layer and business layer components.

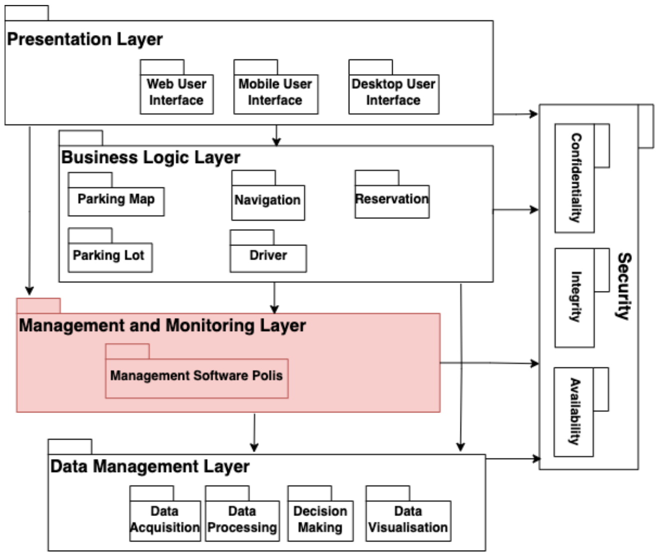

Figure 9. The component&connector view for 4Park.

6.5. *Allocation View*

The allocation view for 4Park is modeled in Figure 10. Given RA's allocation viewpoint in Section 5.4, 4Park's allocation view re-uses the client and server elements from the viewpoint definition. 4Park works with a centralised server that is running in cloud and the server handles all the software components specified in the component&connector view. The server is accessible via different types of clients such as police, municipality staffs, drivers, car parking manager, and parking controllers. While the police, municipality staff, parking controller, and park manager can access the server via their mobile devices or PCs (i.e., either using the desktop or web user interface), the driver can access the system via their mobile device only. Another client is the KIOSK device, which enables the drivers to manage their user profiles and perform parking payment.

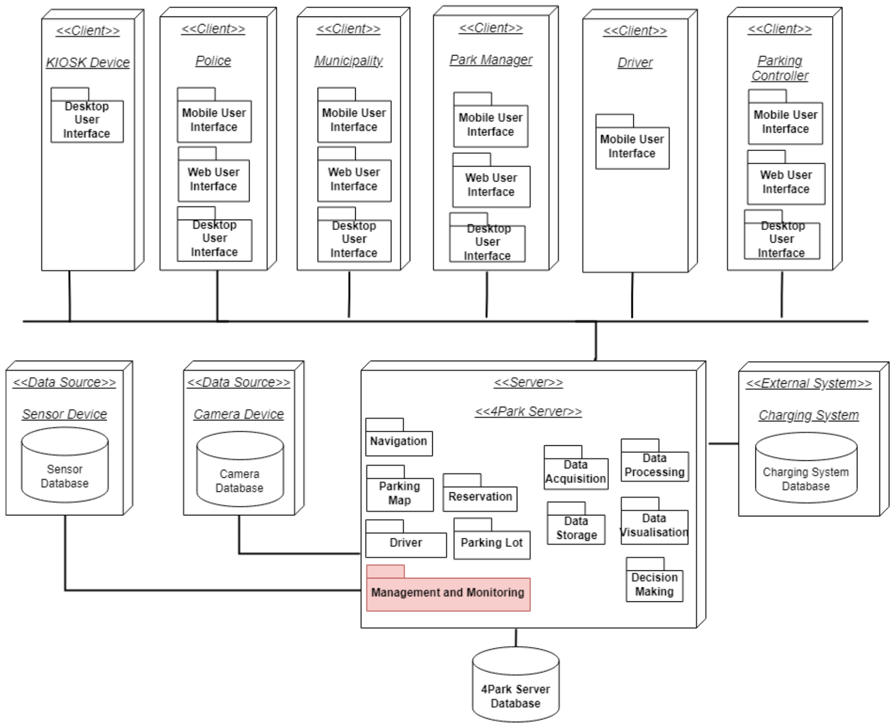

Figure 10. The allocation view for 4Park.

Two different data sources are involved in the 4Park SPMS and communicate their data with the server at real-time - one for the sensors and another for the camera. The electric car charging system that acts as an external system is also integrated, which can provide the charging-related services for electric cars and exchange data with the server.

7. Discussion

To the best of our knowledge, our RA design for the SPMS product family that is introduced in this paper gains its novelty as being the only generic approach that can be re-used as a guide for designing the architecture of any SPMSs. As discussed in Section 2, the existing approaches in the literature do not aid in understanding *(i)* which types of stakeholders are involved in the SPMS domain and their interactions with the system, *(ii)* the common and varying features for SPMSs, *(iii)* the software units that can be designed for implementing an SPMS, *(iv)* the architecture styles that can be used to design the software architecture of an SPMS, and *(v)* the styles for mapping the software units to the physical devices. Therefore, we strongly believe that our paper will bridge the gap here and facilitate the architecture design of any SPMS solutions.

Our RA design for SPMSs is essentially based on our comprehensive domain analysis where our goal was to create a feature model of the SPMS product familty. We searched on the *google scholar* for any papers that discuss SPMS solutions and *google* for any commercial SPMS solutions. We analysed the papers and the commercial tools and identified a set of SPMS features, which we categorised as the mandatory features and optional features. We further conducted a face-to-face survey among 15 park attendants who work for ISPARK in Istanbul, Turkey which is the main parking provider controlled by the municipality. The survey results helped us review the feature set and add/change/remove new/existing/redundant features. We validated our feature set using one of the most popular commercial product called 4Park and did not identify any missing features. For each feature, we further determined the sub-features that contribute the feature goal (e.g., route optimisation for the navigation feature). Note however that we did not consider going deeper in the hierarchy for sub-sub-features as this can make the feature model very complex given so many features and sub-features already identified.

Upon specifying the feature model for SPMSs, we focussed on designing our RA in a way that is easy to manage and understand. Therefore, we considered the separation of concerns into the context, module, component&connector, and allocation viewpoints that have been suggested by Clements et al.'s seminal book on architecture viewpoints [40]. Furthermore, Nakagawa et al. encourage for the same set of viewpoints for the RA designs [42]. Indeed, many attempts have been made so far which have developed RAs for different problem domains in terms of the same set of viewpoints, e.g., the RA for smart farming [29], the RA for digital twins [43], the RA for ontology engineering tools [44], and the RA for smart warehouses [45].

We considered the layered architecture style for the component&connector viewpoint modeling and promote the components that represent the processing units of SPMSs to be organised as a set of cohesive layers whose communications are directed from top to down. We adopted the layered architecture style because it is considered as one of the top-used architecture patterns in industry and facilitates the understandability of the system components and their interactions. Moreover, the architectures of information systems (such as SPMS solutions) fit very well with the layered style, where the top layer represents different user interfaces which require services from the main logic layer (e.g., driver requesting to perform reservation) and that in turn require services from the data layer (e.g., the reservation component requesting the most up-to-date data about the park occupancies).

Our RA design for SPMSs can always be extended for the particular needs of any specific SPMS and its AA design decisions. This can require *(i)* composing/decomposing the existing elements from the RA viewpoints (e.g., software units in the module viewpoint) or *(ii)* adding new units (e.g., adding new intermediate layer for the component&connector viewpoint) or removing the existing units to/from the RA viewpoints. Indeed, we re-used the RA design for the 4Park commercial product, where we removed many of the sub-features that are not considered for 4Park, introduced new sub-features, added a new software layer for the component&connector viewpoint, and composed the software units into a new one for the module viewpoint.

Our effort here in designing an RA for SPMSs is essentially considered as a domain-driven design (DDD) [46], as we performed a comprehensive analysis of the SPMS domain first and then designed a set of abstract modeling viewpoints using the domain features. Also, we do not put emphasis on any technologies that can be used for developing SPMSs (e.g., programming, database, and networking technologies) and rather focussed on proposing generic, architecture-level solution using abstract modeling. Therefore, our RA design is essentially useful in the inital stages of SPMS development life-cycle. Indeed, with our RA, the communications between non-technical stakeholders (e.g., users, customers, and managers) and technical stakeholders (e.g., developers) are facilitated, the system boundaries can easily be established, and high-level design decisions can easily be figured out using the RA's viewpoint definitions. We strongly believe that our RA design can highly reduce the time and effort needed for analysing and designing SPMS solutions and support the quality SPMS development that better meet the requirements.

8. Conclusion

In this paper, we designed an RA for the SPMS product family so as to provide an architectural guidance for analysing and designing the application architectures of any SPMSs in the most productive and quality way. We initially provided a feature model that describes the common and varying features and their sub-features for SPMSs in a hierarchical way. Using the feature model, we proposed four essential viewpoints for the specifications of SPMS architectures which are context, module, component&connector, and allocation. The context viewpoint is concerned with specifying the system boundaries in terms of the stakeholders and their interaction with the SPMS under consideration. The module viewpoint is concerned with specifying the software units that can be designed and implemented for an SPMS and their software modules. The component&connector viewpoint is concerned with specifying the runtime structure of an SPMS in terms of the components that represent the independent processing units and the connectors for the component interactions. The allocation viewpoint is concerned with specifying the physical architecture and the mapping between the software and physical components.

We validated our RA design for SPMSs using a commercial SPMS product called 4Park and show how the RA viewpoints can be re-used and extended. We strongly believe that our RA design is a unique contribution for the literature and has high potential to aid developers in analysing and designing quality SPMS products.

We are currently designing and developing a prototype SPMS application using the RA design introduced in the paper. By doing so, we aim to validate the RA design to a further extent. Moreover, we are planning to analyse the commercial SPMS products with regard to their support for the feature model specified in this paper. Lastly, we have already started conducting interviews with diverse industries including government, logistics and transportation, universities, automative, where parking management is one of the most crucial concerns, and aim to evolve our RA design with diverse problems and their needs.

References

1. Population, C. Major Agglomeratıons of the World. https://www.weforum.org/agenda/2016/04/the-number-of-cars-worldwide-is-set-to-double-by-2040, 2024. Accessed: (30.05.2024).
2. Isa, N.; Yusoff, M.; Mohamed, A. A Review on Recent Traffic Congestion Relief Approaches. 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, 2014, pp. 121–126. doi:10.1109/ICAIET.2014.29.
3. Forum, W.E. The number of cars worldwide is set to double by 2040. https://www.weforum.org/agenda/2016/04/the-number-of-cars-worldwide-is-set-to-double-by-2040, 2016. Accessed: (30.05.2024).

4. Lee, J.B.; Agdas, D.; Baker, D. Cruising for parking: New empirical evidence and influential factors on cruising time. *Journal of Transport and Land Use* **2017**, *10*. doi:10.5198/jtlu.2017.1142.
5. INRIX. Searching for Parking Costs the UK £23.3 Billion a Year. https://inrix.com/press-releases/parking-pain-uk, 2017. Accessed: (30.05.2024).
6. Das, S. A Novel Parking Management System, for Smart Cities, to save Fuel, Time, and Money. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019, pp. 0950–0954. doi:10.1109/CCWC.2019.8666537.
7. Shoup, D.C. Cruising for parking. *Transport Policy* **2006**, *13*, 479–486. Parking, doi:https://doi.org/10.1016/j.tranpol.2006.05.005.
8. Lin, T.; Rivano, H.; Le Mouël, F. A Survey of Smart Parking Solutions. *IEEE Transactions on Intelligent Transportation Systems* **2017**, *18*, 3229–3253. doi:10.1109/TITS.2017.2685143.
9. Ladyman, J.; Wiesner, K. *What Is a Complex System?*; Yale University Press, 2020.
10. Garlan, D.; Shaw, M. An Introduction to Software Architecture. Technical report, Pittsburgh, PA, USA, 1994.
11. Muller, G. A Reference Architecture Primer. Technical report, University of South-Eastern Norway-NISE, 2020. Available from: https://www.gaudisite.nl/ReferenceArchitecturePrimerPaper.pdf.
12. Kotb, A.O.; Shen, Y.C.; Zhu, X.; Huang, Y. iParker—A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing. *IEEE Transactions on Intelligent Transportation Systems* **2016**, *17*, 2637–2647. doi:10.1109/TITS.2016.2531636.
13. Mohammadi, F.; Nazri, G.A.; Saif, M. A Real-Time Cloud-Based Intelligent Car Parking System for Smart Cities. 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP), 2019, pp. 235–240. doi:10.1109/ICICSP48821.2019.8958543.
14. Kianpisheh, A.; Mustaffa, N.; Limtrairut, P.; Keikhosrokiani, P. Smart parking system (SPS) architecture using ultrasonic detector. *International Journal of Software Engineering and Its Applications* **2012**, *6*, 55–58.
15. Thomas, T.; Bhatt, T. Smart Car Parking System Using Convolutional Neural Network. 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 172–174. doi:10.1109/ICIRCA.2018.8597227.
16. Thomas, D.; Kovoor, B.C. A Genetic Algorithm Approach to Autonomous Smart Vehicle Parking system. *Procedia Computer Science* **2018**, *125*, 68–76. The 6th International Conference on Smart Computing and Communications, doi:https://doi.org/10.1016/j.procs.2017.12.011.
17. Tandon, R.; Gupta, P.K. Optimizing Smart Parking System by Using Fog Computing. Advances in Computing and Data Sciences; Singh, M.; Gupta, P.; Tyagi, V.; Flusser, J.; Ören, T.; Kashyap, R., Eds.; Springer Singapore: Singapore, 2019; pp. 724–737.
18. Thakre, M.P.; Borse, P.S.; Matale, N.P.; Sharma, P. IOT Based Smart Vehicle Parking System Using RFID. 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1–5. doi:10.1109/ICCCI50826.2021.9402699.
19. Fahim, A.; Hasan, M.; Chowdhury, M.A. Smart parking systems: comprehensive review based on various aspects. *Heliyon* **2021**, *7*, e07050. doi:https://doi.org/10.1016/j.heliyon.2021.e07050.
20. Paidi, V.; Fleyeh, H.; Håkansson, J.; Nyberg, R.G. Smart parking sensors, technologies and applications for open parking lots: a review. *IET Intelligent Transport Systems* **2018**, *12*, 735–741, [https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2017.0406]. doi:https://doi.org/10.1049/iet-its.2017.0406.
21. Xiao, X.; Peng, Z.; Lin, Y.; Jin, Z.; Shao, W.; Chen, R.; Cheng, N.; Mao, G. Parking Prediction in Smart Cities: A Survey. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 10302–10326. doi:10.1109/TITS.2023.3279024.
22. Barriga, J.J.; Sulca, J.; León, J.L.; Ulloa, A.; Portero, D.; Andrade, R.; Yoo, S.G. Smart Parking: A Literature Review from the Technological Perspective. *Applied Sciences* **2019**, *9*. doi:10.3390/app9214569.
23. Fraifer, M.; Fernström, M. Designing an IoT smart parking prototype system. Thirty Seventh International Conference on Information Systems. ISCA, 2016, pp. 2–12.
24. Anderson, E.C.; Okafor, K.C.; Nkwachukwu, O.; Dike, D.O. Real time car parking system: A novel taxonomy for integrated vehicular computing. 2017 International Conference on Computing Networking and Informatics (ICCNI), 2017, pp. 1–9. doi:10.1109/ICCNI.2017.8123788.

25. Soegoto, E.S.; Pamungkas, V.Y.; Herdiawan, A. Designing Smart Parking Application for Car Parking Space Arrangement. *IOP Conference Series: Materials Science and Engineering* **2018**, *407*, 012185. doi:10.1088/1757-899X/407/1/012185.
26. Awaisi, K.S.; Abbas, A.; Zareei, M.; Khattak, H.A.; Khan, M.U.S.; Ali, M.; Din, I.U.; Shah, S. Towards a Fog Enabled Efficient Car Parking Architecture. *IEEE Access* **2019**, *7*, 159100–159111. doi:10.1109/ACCESS.2019.2950950.
27. Alharbi, A.; Halikias, G.; Yamin, M.; Abi Sen, A. Web-based framework for smart parking system. *International Journal of Information Technology (Singapore)* **2021**, *13*, 1495–1502. Publisher Copyright: © 2021, Bharati Vidyapeeth's Institute of Computer Applications and Management., doi:10.1007/s41870-021-00725-8.
28. Tummers, J.; Tobi, H.; Catal, C.; Tekinerdogan, B. Designing a reference architecture for health information systems. *BMC Medical Informatics and Decision Making* **2021**, *21*. Publisher Copyright: © 2021, The Author(s)., doi:10.1186/s12911-021-01570-2.
29. Krisnawijaya, N.N.K.; Tekinerdogan, B.; Catal, C.; van der Tol, R. Reference architecture design for developing data management systems in smart farming. *Ecological Informatics* **2024**, *81*, 102613. doi:https://doi.org/10.1016/j.ecoinf.2024.102613.
30. Behere, S.; Törngren, M. A functional reference architecture for autonomous driving. *Information and Software Technology* **2016**, *73*, 136–150. doi:https://doi.org/10.1016/j.infsof.2015.12.008.
31. TAO, F.; SUN, X.; CHENG, J.; ZHU, Y.; LIU, W.; WANG, Y.; XU, H.; HU, T.; LIU, X.; LIU, T.; SUN, Z.; XU, J.; BAO, J.; XIANG, F.; JIN, X. makeTwin: A reference architecture for digital twin software platform. *Chinese Journal of Aeronautics* **2024**, *37*, 1–18. doi:https://doi.org/10.1016/j.cja.2023.05.002.
32. Melnyk, P.; Djahel, S.; Nait-Abdesselam, F. Towards a Smart Parking Management System for Smart Cities. 2019 IEEE International Smart Cities Conference (ISC2), 2019, pp. 542–546. doi:10.1109/ISC246665.2019.9071740.
33. Kubler, S.; Robert, J.; Hefnawy, A.; Cherifi, C.; Bouras, A.; Främling, K. IoT-based Smart Parking System for Sporting Event Management. Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services; Association for Computing Machinery: New York, NY, USA, 2016; MOBIQUITOUS 2016, p. 104–114. doi:10.1145/2994374.2994390.
34. Ajchariyavanich, C.; Limpisthira, T.; Chanjarasvichai, N.; Jareonwatanan, T.; Phongphanpanya, W.; Wareechuensuk, S.; Srichareonkul, S.; Tachatanitanont, S.; Ratanamahatana, C.; Prompoon, N.; Pipattanasomporn, M. Park King: An IoT-based Smart Parking System. 2019 IEEE International Smart Cities Conference (ISC2), 2019, pp. 729–734. doi:10.1109/ISC246665.2019.9071721.
35. Al-Kharusi, H.; Al-Bahadly, I. Intelligent parking management system based on image processing. *World Journal of Engineering and Technology* **2014**, *2014*.
36. Bura, H.; Lin, N.; Kumar, N.; Malekar, S.; Nagaraj, S.; Liu, K. An Edge Based Smart Parking Solution Using Camera Networks and Deep Learning. 2018 IEEE International Conference on Cognitive Computing (ICCC), 2018, pp. 17–24. doi:10.1109/ICCC.2018.00010.
37. Perković, T.; Šolić, P.; Zargariasl, H.; Čoko, D.; Rodrigues, J.J. Smart Parking Sensors: State of the Art and Performance Evaluation. *Journal of Cleaner Production* **2020**, *262*, 121181. doi:https://doi.org/10.1016/j.jclepro.2020.121181.
38. Kang, K.; Cohen, S.; Hess, J.; Novak, W.; Peterson, A. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-021, 1990. Accessed: 2024-Jun-6.
39. Rozanski, N.; Woods, E. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*, 2 ed.; Addison-Wesley Professional, 2011.
40. Clements, P.; Garlan, D.; Bass, L.; Stafford, J.; Nord, R.; Ivers, J.; Little, R. *Documenting Software Architectures: Views and Beyond*; Pearson Education, 2002.
41. Richards, M. *Software Architecture Patterns*; O'Reilly Media, Inc., 2015.
42. Nakagawa, E.Y.; Oquendo, F.; Maldonado, J.C., Reference Architectures. In *Software Architecture 1*; John Wiley & Sons, Ltd, 2014; chapter 2, pp. 55–82, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118930960.ch2]. doi:https://doi.org/10.1002/9781118930960.ch2.
43. van Dinter, R.; Tekinerdogan, B.; Catal, C. Reference architecture for digital twin-based predictive maintenance systems. *Computers & Industrial Engineering* **2023**, *177*, 109099. doi:https://doi.org/10.1016/j.cie.2023.109099.
44. Braun, G.A.; Estevez, E.; Fillottrani, P. A Reference Architecture for Ontology Engineering Web Environments. *Journal of Computer Science and Technology* **2019**, *19*, e03. doi:10.24215/16666038.19.e03.

45. van Geest, M.; Tekinerdogan, B.; Catal, C. Design of a reference architecture for developing smart warehouses in industry 4.0. *Computers in Industry* **2021**, *124*, 103343. doi:https://doi.org/10.1016/j.compind.2020.103343.
46. Evans. *Domain-Driven Design: Tacking Complexity In the Heart of Software*; Addison-Wesley Longman Publishing Co., Inc.: USA, 2003.