Cloud connected vehicles : towards new business opportunities

/ Department of Mathematics and Computer Science / PDEng Automotive Systems Design

Cloud connected vehicles

Towards new business opportunities

October 2017

Andreas Krivas

Cloud connected vehicles
Towards new business opportunities

Eindhoven University of Technology
Stan Ackermans Institute - Automotive/Mechatronic Systems Design

PDEng Report: 2017/087

Andreas Krivas

The design that is described in this report has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct.

Partners

Vehicle for Innovation Brabant electric

Eindhoven University of Technology

Steering Group

Hans Brouwhuis – Site Manager NXP Semiconductors Eindhoven, VIBe Program manager

Dr.ir. Reinder J. Bril – Associate Professor
Department of Mathematics and Computer Science, SAN Group, TU/e

Date October 2017

Date October, 2017

Contact address Eindhoven University of Technology
Department of Mathematics and Computer Science
MF 5.072
P.O. Box 513
NL-5600 MB
Eindhoven, The Netherlands
+31 402743908

Published by Eindhoven University of Technology
Stan Ackermans Institute

Printed by Eindhoven University of Technology
UniversiteitsDrukkerij

PDEng Report PDEng Report nr. 2017/087

Abstract In this report, the concept of a cloud connected smart mobility platform is investigated focusing on enabling easy application development and deployment without vendor lock-ins via an open and multi-vendor environment. A prototype of the plaform along with a simple cloud application are developed to verify the feasibility of a first instantiation of the platform and evaluate its potential according to the aforementioned elements. In addition, a recommendation to the VIBe initiative is proposed containing the necessary steps from strategic and technical perspective for VIBe and the region of Eindhoven with respect to smart technologies and especially, smart mobility.

Keywords IoT, Connected Cars, Smart mobility, Cloud computing

Preferred reference Enabling new business by leveraging cloud technology via an open and multi-vendor environment, Eindhoven University of Technology, PDEng Technical report, October 2017

Partnership This project was supported by Eindhoven University of Technology and VIBe initiative

Disclaimer Liability	While every effort will be made to ensure that the information contained within this report is accurate and up to date, Eindhoven University of Technology makes no warranty, representation or undertaking whether expressed or implied, nor does it assume any legal liability, whether direct or indirect, or responsibility for the accuracy, completeness, or usefulness of any information.
Trademarks	Product and company names mentioned herein may be trademarks and/or service marks of their respective owners. We use these names without any particular endorsement or with the intent to infringe the copyright of the respective owners.
Copyright	

Foreword

We are in the middle of a smart transition entering uncharted IoT (Internet of Things) territory - and on the doorstep the migration of the physical and virtual world - with new rules, new stakeholders, new consortia and new modes in the way we have to deal with unsustainable trends demanding disruptive systems solutions to address the major challenges we face in health, energy, environment and mobility.

The world around us is changing fast, and so are our lives. We drive cars that automatically brake because the vehicle ahead has warned of an accident. We use our smartphones and smart pads for everything from counting calories to online banking and playing augmented reality games (Pokémon) in the dunes. The face of our cities is changing through interconnected traffic control, intelligent transport systems and smart power grids. In short, technological innovations have become an intrinsic part of our daily lives. They influence how we perceive the world around us, how we interact, how we move, work, and live. We are in the middle of a very disruptive smart transition.

Successful thought leadership in this area depends highly on our ability to share, working together and adapt our way of working very rapidly and earn and maintain the trust of partners and consumers in this smart transition related fast changing global marketplace.

This report is part of the VIBe initiative, a connected E-Mobility initiative which is more than ‘just’ equipping a car with a box. It involves making a secure connection to a service / node in the cloud that makes data accessible via easy to use open APIs (Application Programming Interface) to multiple parties who develop different applications whereby the driver decides who can have access to what data and when. The connected car becomes a living room on wheels with the same familiar options - including security and privacy - you get at home. Once this has been installed, virtually endless possibilities are created for potential new multi domain applications.

Since the VIBe initiative is one of the first to move on this we have high expectations of its impact, especially since the entire region around it is an internet of things breeding place; smart mobility, smart city, smart energy technology including a university next that has energy and smart mobility as two of its three strategic areas.

I’ve come to know Andreas as an entrepreneur, socially skilled, smart, hardworking, honest and too modest for his own good. He as we are not experts in this field but nobody is because this topic simply spans uncharted territories. A paradigm shift is going on; mobility solutions integrate with smart grids, sustainability initiatives, the transition to electrical transportation, changing mobility use concepts entering the new promising world of cyber-physical systems and the migration of our well known physical and unknown virtual world. Just imagine how the future could look like. It’s mind-blowing.

I think Andreas has done a commendable job in quickly coming to terms with this new field and in coming up with fresh ideas for issues that might help to tackle the challenge we have ahead and let's not forget keeping the VIBe initiative on speed.

I hope this report / his contribution will lead to the right decisions and the follow up efforts with which we will create the VIBe infrastructure facilitating the VIBe applications we actually already built.

I can hardly wait.

Hans Brouwhuis
Site Manager NXP Eindhoven
NXP Semiconductors Netherlands B.V.
Sept 28th, 2017

Preface

This report is part of the final project of the Professional Doctorate in Engineering (PDEng) program in Automotive System Design (ASD) and the work contained in this report was carried out under the partnership and support of VIBe (Vehicle for Innovation Brabant - electric) and Eindhoven University of Technology.

The focus areas of the work are the fields of 'Internet of Things', 'Smart Technologies' and especially 'Smart Mobility'. While there is still confusion around these terms and their potential impact on society, numerous cities and industries are transforming their infrastructure by leveraging cloud technologies in order to exploit new opportunities.

In this report, a cloud platform for smart mobility applications focusing on ease of development, open standards, multi-vendor functionality and prevention of vendor lock-ins regarding the infrastructure is proposed. In this way, a collaborative spirit and an open mentality are promoted towards enabling innovative applications. In addition, a recommendation to the VIBe initiative is conducted containing the necessary next steps for VIBe and the region regarding smart technologies and especially smart mobility.

This report is intended for the domain experts working in the smart mobility and connected car technology domains. However, it can also be useful for experts in cloud computing technologies, as cloud computing is at the core of the report.

A part of the report is also interesting to business enterprises and entrepreneurs, which would like to explore the new disruptive smart transformation and seek ways to exploit new business opportunities.

Andreas Krivas

October 1st, 2017

Acknowledgments

This final project has been more than a learning experience throughout the whole process. Apart from having the opportunity to engage in technical challenges, I was able to grow personally and professionally in multiple ways.

Special thanks to my company project mentor and supervisor ir.Hans Brouwhuis for giving me the opportunity to develop myself not only from technical point of view but also experience what it means to be an entrepreneur. I will never forget our very interesting and inspiring discussions. Furthermore, I would like to thank my TU/e supervisor, dr.ir.Reinder J.Bril, for his guidance and cooperation, as well as his attention to detail and shrewd observations, which helped me deliver results of higher quality.

I wish also to take this opportunity to express my gratitude to dr.ir.Peter Heuberger, Program manager of the Automotive Systems Design (ASD) PDEng program, firstly for giving me the opportunity to be part of the ASD community, as well as for his continuous support, understanding and motivation. In addition, I would like to thank dr.ir.Peter Cuijpers and PhD candidate Jeroen Redegeld for their help on crucial moments of the project. My gratitude also goes to Ms.Ellen van Hoof-Rompen, management assistant, for her support throughout the last two years.

The project would not be successful without the help of several people from the university, the industry and the authorities. Thanks to everyone who provided useful insights and information towards achieving the desired results. Special thanks to Wim Vossebelt from V-tron Technologies and Ronald de Beijer from Beijer Automotive for their continuous support and willingness to facilitate the project. In addition, I would like to extend my gratitude to prof.dr.ir.Maarten Steinbuch chair of Control Systems Technology and VIBe Steering Group member, dr.ir.Carlo van de Weijer from Strategic area smart mobility and Bert-Jan Woertman, Commercial director for the TU/e.

I would also like to thank all of my colleagues in the generation ASD 2015 for all the nice memories, inspiring brainstorming sessions, advice, laughters and support. I hope that we have the chance in the future to work again together. It will be an honor for me.

Last but not least, I owe my gratitude to all of my family and friends for their unconditional support at all times and for believing in me.

Andreas Krivas

October 1st, 2017

Executive Summary

The impact of the Internet of Things and cloud computing technologies is evident already in modern societies. Sectors such as mobility, healthcare or energy are being transformed and new business opportunities are created every day. While IoT devices can generate massive amount of data, cloud computing provides a pathway for that data to reach its destination. The VIBe initiative (Vehicle for Innovation Brabant electric) has been active since 2013 in enabling business development in the region of Eindhoven, especially with respect to smart mobility and cloud connected vehicles.

There are two main objectives of this report. The first is the investigation of a possible way forward for VIBe and the region of Eindhoven regarding exploiting new opportunities related to smart mobility. This way forward was conducted in the form of a recommendation directed to the VIBe initiative and strategic objectives were defined on a proposed vision towards new business opportunities.

The second main objective of this report is the investigation of a cloud connected Smart Mobility Platform. Having as distinctive features ease of development and deployment of applications, multi-vendor functionality, open development and prevention of vendor lock-ins regarding the infrastructure, this platform aims to start innovation in the context of smart mobility.

As a first demonstration of such a platform, a prototype was implemented focusing on ease of development and deployment and vendor independence, as well as cost effectiveness and flexibility regarding the underlying cloud infrastructure. In addition, a simple cloud application was developed with the purpose to satisfy the prototype requirements. Unfortunately, open development and multi-vendor functionality were not investigated in this prototype due to lack of available resources and are considered as future work.

The prototype and application were implemented successfully. The results showed that it is possible to achieve easy initial instantiation of a cloud platform and fast deployment of applications. The use of specific cloud services minimize the dependencies with cloud providers or hardware specifications. However, OS dependencies still exist and therefore, we can conclude that full independence is not possible yet. The application showed that it is possible to create a simple application and deploy it on the cloud in a matter of a few days. Regarding the costs, the implemented cloud services indicated that it is possible to create a cost effective first instantiation of the platform with zero initial costs and affordable recurring costs, while simultaneously providing flexibility regarding the infrastructure resources.

Andreas Krivas

October 1st, 2017

Glossary

ASD	Automotive Systems Design
OEM	Original Equipment Manufacturer
PDEng	Professional Doctorate in Engineering
PSGM	Project Steering Group Meeting
TU/e	Eindhoven University of Technology
SAN	System Architecture and Networking
BOM	Brabantse Ontwikkelings Maatschappij
RWS	Rijkswaterstaat
ANWB	Algemene Nederlandse Wielrijders Bon
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek
NDW	Nationale Databank Wegverkeersgegevens
ICT	Information and Communication Technology
ITS	Intelligent Transport Systems
EV	Electric Vehicles
MaaS	Mobility as a Service
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
API	Application Programming Interface
SQL	Structured Query Language
CPU	Central Processing Unit
OS	Operating System
RAML	RESTful API Modeling Language
URL	Uniform Resource Locator

List of Figures

Contents

1 Introduction

This report provides a comprehensive overview of the results of the Automotive Systems Design (ASD) Professional Doctorate in Engineering (PDEng) assignment on behalf of the initiative called 'Vehicle for Innovation Brabant electric', known as VIBe. The nature of the assignment was initially focused on entrepreneurial objectives that led to the creation of a concept considering a cloud platform for smart mobility applications, focusing on ease of development and deployment of applications, multi-vendor functionality with respect to hosting applications and components of different vendors, open development and vendor independence regarding the infrastructure. Consequently, the requirements of such a platform, as well as an implemented prototype are presented. Furthermore, an implementation of a simple web application that aims to verify these requirements was achieved and is explained in this report as well. Simultaneously, a recommendation to VIBe was conducted regarding proposed strategic steps to be taken in order to exploit the new business opportunities from leveraging smart technologies in the region of Eindhoven.

1.1 Background and context

Vehicle for Innovation Brabant electric (VIBe) is an initiative started in 2013 by a group of companies and TU/e in the Brainport region of the Netherlands in order to promote innovation in the context of smart applications, starting from connected cars and smart mobility [5]. More specifically, VIBe supports the creation of multi-vendor and multi-user solutions, making efficient use of toolboxes and components that are based on a common and open system architecture. Eventually, these solutions can enable fast and simple application integration and development at the end of the value chain.
As a first step towards that direction, VIBe decided to develop a common open Application Programming Interface (called VIBeX API) for electric vehicles (EVs) making use of the fast growing field of Internet of Things. Their aim is to act as an enabler of several businesses by allowing applications to be built against that API. More specifically, the VIBeX API aims to expose vehicle data to the cloud via applications that can be built against it. This API was investigated in the work of Preethi Ramamurthy [5], where an architecture design of an open source platform in the form of API focused on EV users was proposed.
Furthermore, the cloud computing technologies are in the core of operations of VIBe. Thus, cloud computing was used to realize the prototype of the smart mobility platform and the application. There are several reasons regarding the decision to base the work explained in this report on cloud computing technologies:

- **Existing experience:** Cloud is used by the majority of the companies comprising the VIBe Steering Group committee (V-tron, NXP, Driessen), as well as by VIBe partners (Beijer Automotive)

- **Flexibility:** The resources are scaled according to the demand and there is no need to build for the future and put constraints on the infrastructure.
- **Cost:** Using cloud technology reduces the maintenance fees. Many of the hidden costs typically associated with software implementation, customization, hardware, maintenance, and training are rolled into a transparent subscription fee.
- **Innovation:** The use of cloud computing in automotive is currently being explored by the automotive industry, which can unravel interesting opportunities ([11],[12])

1.2 Problem description

The use of the VIBeX API did not lead to achieving the desired goals. Apart from five main implementations (section 2.5), a wider acceptance of the API was not achieved. Moreover, all these implementations were followed individually by each company and there was a lack of monitoring the developments. Thus, it was decided that in the last quarter of 2017, the Steering Group of VIBe would reconvene to make a decision regarding the continuation of the initiative or the termination of activities. This decision would be based mainly on the results of this PDEng project and a recommendation on how VIBe should renew their goals and motivations.
The existence of such a recommendation is desired to be accompanied by a showcase that can highlight the key findings of that investigation. The intention is to investigate the potential of a cloud platform for smart mobility applications taking into consideration four factors:

- **Ease of development and deployment of applications:** Creating and deploying an application in a small amount of time (for example, in a few days) can really create numerous showcases of smart mobility applications.
- **Open development environment:** Via the use of open APIs, such as the VIBeX API, it is desirable to facilitate open development of applications towards maintaining open the interfaces between components.
- **Prevention of vendor lock-ins:** Dependencies of any kind (hardware or software) are to be avoided with respect to the required infrastructure. In this way, a flexible and independent platform can be created towards developing more diverse applications.
- **Multi-vendor functionality:** Combined with open development and prevention of vendor lock-ins, it is possible to create a multi-vendor environment, where applications from different vendors can be hosted.

1.3 Goals

After the context and problem description explained in sections 1.1 and 1.2 respectively, two main goals can be defined:

1. A proposal for the VIBe steering group committee is desired regarding the next steps for VIBe and the region in the smart technologies context. Based on this plan, the Steering Group of VIBe will decide on the continuation of the initiative
2. A prototype of the smart mobility platform enabling the development and instantiation of a smart mobility application, focusing on the four main aspects mentioned in section 1.2

1.4 Delimiting the scope

From the aforementioned goals the detailed scope of the report is presented below.

- Creation of a recommendation document to VIBe regarding the strategic next steps for VIBe and the region
- Requirements and a proposed setup for the smart mobility platform
- Evaluation of the design choices regarding a cloud connected smart mobility platform
- Implementation of a smart mobility platform prototype based on a sub-set of the requirements aiming to demonstrate ease of development/deployment of applications, vendor independence, open development and multi-vendor functionality
- Development and implementation of a simple application for the prototype

1.5 Approach

This section explains in detail the approach towards conducting the recommendation to VIBe, as well as implementing the prototype platform and the application.

1.5.1 VIBe recommendation

Initially, the main goal was to understand the global status of smart technologies especially in the mobility sector and their current trends. To get a better overview, two market maps were created (global and regional respectively, see Appendix B). In addition, worldwide events and conferences were analyzed, as well as interesting business partnerships. In order to get a better insight on the regional developments, several interviews were conducted. People from regional companies (NXP, Beijer Automotive, Vtron Technologies, Ericsson, HERE Technologies) were interviewed (see Appendix B for a full list of the interviewees) and their view on the current situation in Brabant was recorded and analyzed. In addition, people from the local authorities of Eindhoven (Gemeente) were interviewed on how the authorities view the digital transformation of smart cities. Finally, local events (Automotive Week 2017) were followed and discussion with representatives from local companies took place.

Taking into account the above, an investigation in the form of a SWOT analysis (strengths - weaknesses - opportunities - threats) was conducted. Eventually, proposed steps for VIBe and the region were defined on how to adapt to the smart technologies (more information about the recommendation can be found on Chapter 3 and Appendix C).

1.5.2 Prototype of the Smart Mobility Platform

As mentioned in section 1.3, the second goal was to create a prototype of the smart mobility platform towards enabling development and instantiation of a smart mobility application. Initially, the requirements for the platform were derived in cooperation with VIBe. The aforementioned benefits of using cloud computing technologies (section 1.1) were taken into consideration, along with the requirements in having an open and multi-vendor environment for easy mobility applications development and deployment, avoiding vendor lock-ins. Afterwards, the available resources (either complete cloud platforms or individual components) where evaluated in order to determine if there is a possibility to acquire an already developed platform and create the prototype with that reference. However, the required resources where not found and therefore, it was decided that a new prototype will be developed.
Finally, a simple smart mobility application was developed and deployed on the prototype platform. Different implementation and verification scenarios were used to verify several of the platform requirements and evaluate the different cloud options related to instantiating such a smart mobility platform.

1.5.3 Design process

The systems engineering process that was followed to implement the prototype platform is the V-Model (figure 1.1). More specifically, the left part was followed starting from the business requirements (section 4.2) for the smart mobility platform. Afterwards, the system requirements (section 4.3) for the prototype platform and the application were identified, as well as the specifications (section 6.5). Finally, a system and acceptance testing was perform via the implementation of a simple application (chapter 7) and the prototype platform specifications verification (chapter 8).

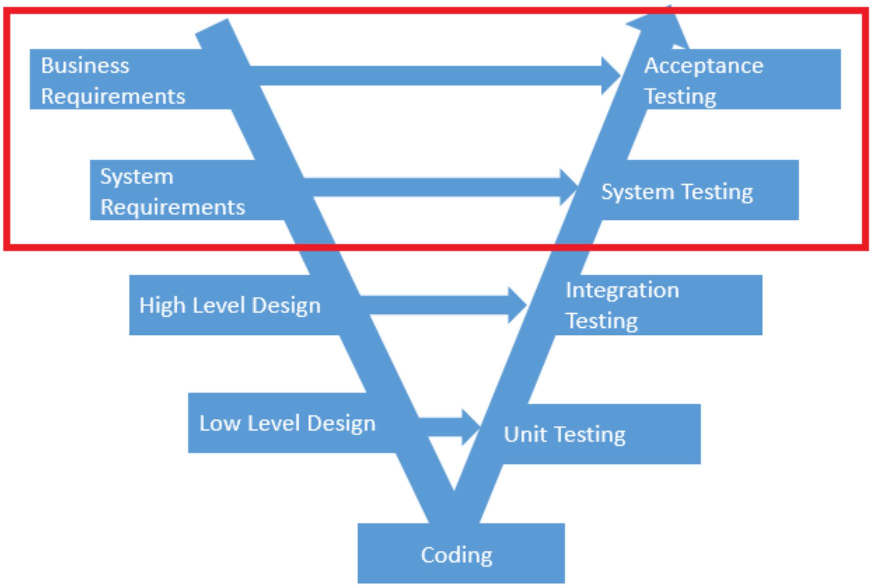

Figure 1.1: The V-Model. The red box represents the steps followed towards the implementation of the prototype.

1.6 Results

Several results were achieved following the aforementioned approach towards achieving the goals specified in section 1.3:

- The recommendation to VIBe was conducted. The recommendation was accepted by the program manager of VIBe. However, the presentation was not conducted during October as initially planned, due to unavailability of the the steering group committee members. Instead, it was published via email along with an executive summary and will be presented by the program manager of VIBe at a later date. *(Related to Goal No.1)*

- Investigation of different cloud providers and services, as well as proposed design choices for the smart mobility platform *(Related to Goal No.2)*

- A prototype of the smart mobility platform was implemented using cloud technologies and a vendor agnostic approach, focusing on ease of development and deployment. It was not possible to evaluate open development and multi-vendor functionality due to unavailability of resources (see chapter 6) *(Related to Goal No.2)*

- A simple application was developed and used as a showcase for the prototype platform. A series of verification tests were conducted towards verifying the prototype platform specifications. *(Related to Goal No.2)*

1.7 Report outline

The report continues by analyzing the problem more in depth, presenting the vision of VIBe as well as a thorough investigation of cloud computing working principles and models. Next, the most important findings of the VIBe recommendation are presented, which are also linked with the smart mobility platform. Afterwards, the requirements of the platform are presented, focusing on the applicable requirements to the prototype. Then, important design and implementation choices are discussed. The report continues investigating the implementation of the prototype platform and the application that was developed, as well as the verification of the prototype platform specifications. Finally, the report concludes with recommendations for future work and the presentation of the results. More specifically:

- **Chapter 2: Problem analysis**
  Analysis of the terms internet of things, smart technologies, smart mobility and connected car technologies. Moreover, detailed analysis of the global and regional developments regarding the connected car technologies is presented, as well as in depth information regarding the VIBe initiative. Finally, a presentation of cloud technology deployment and service models is included.

- **Chapter 3: VIBe recommendation**
  Includes the results regarding the recommendation to VIBe on how to approach from a strategic perspective the smart technologies and especially smart mobility

- **Chapter 4: Requirements elicitation**
  Describes the derived requirements for the prototype smart mobility platform. The requirements for the complete platform can be found in Appendix D.

- **Chapter 5: Prototype platform design choices**
  Investigates design choices regarding the prototype platform, focusing on vendor independence

- **Chapter 6: Prototype platform implementation**
  Presents important implementation choices regarding the setup of the prototype platform, as well as the derived specifications

- **Chapter 7: Application implementation**
  Explains the different implementation efforts regarding the developed application using the design choices of the prototype platform described in the previous chapters

- **Chapter 8: Prototype platform verification**
  Includes an evaluation of the prototype platform with respect to costs, ease of development/deployment and vendor independence

- **Chapter 9: Conclusions**
  Presents the results and recommendations for future work

2 Problem Analysis

In this chapter, the problem context is further analyzed. Initially, the terms Internet of Things (IoT), smart technologies and smart mobility are defined, which provide the domain context of the platform. Moreover, in depth information regarding the global and regional developments in the smart mobility and connected car context are provided. Afterwards, the goals and activities of VIBe are presented in detail, as well as the key cloud computing principles and models towards creating a smart mobility platform.

2.1 Domain context

The world is experiencing a new industrial revolution that impacts almost every aspect of modern societies. This revolution (often referred to as Industry 4.0 [13]) started with the constantly increasing evolution of the software technologies and is now led by the effort to connect everything to the Internet. However, there is still a lot of confusion around terms like IoT, smart technologies or smart mobility, as it is not entirely clear yet what will be the exact impact on society.

2.1.1 Internet of Things

The term known as Internet of Things is defined as the addition of connectivity to the Internet capabilities to previously unconnected devices (excluding computers). These IoT devices can range from connected home appliances (TVs, light bulbs, even refrigerators) to connected cars or consumer wearables (figure 2.1).

Typically, IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications. The interconnection of these embedded devices (including smart objects), is expected to usher in automation in nearly all fields, while also enabling advanced applications like a smart grid and expanding to areas such as smart cities. As well as the expansion of Internet-connected automation into a plethora of new application areas, IoT is also expected to generate large amounts of data from diverse locations, with the consequent necessity for quick aggregation of the data, and an increase in the need to index, store, and process such data more effectively.

The predictions indicate that by 2020 there will be 50 billion things connected to the Internet [14], while in 2017 there are already 8.4 billion devices connected according to Gartner [15]. This impressive growth demonstrates the reasons why the impact of these technologies is closely monitored and why the current business models are changing.

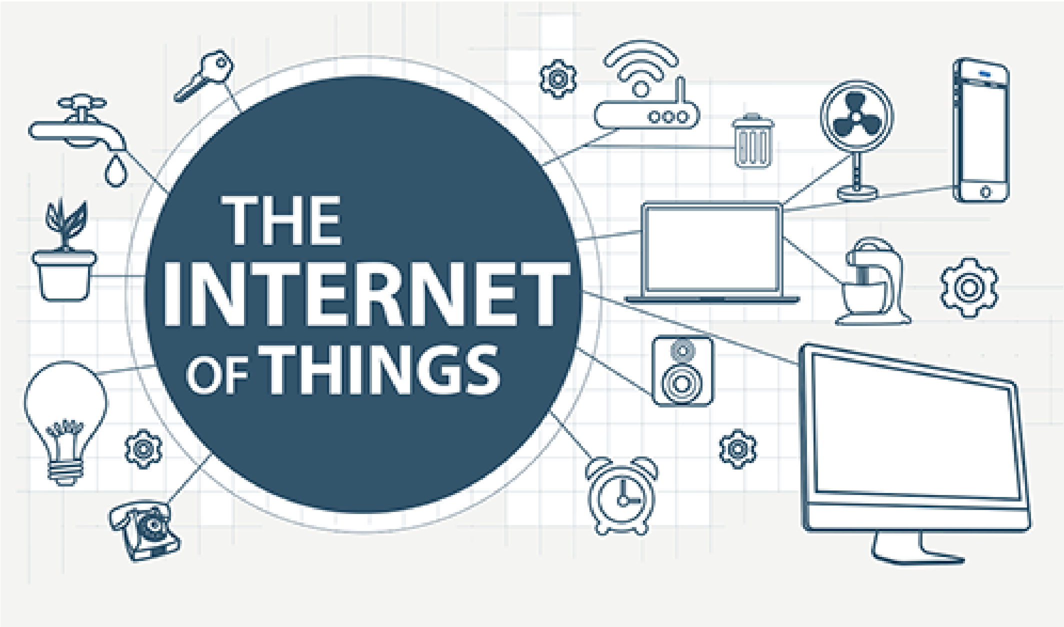

Figure 2.1: Internet of Things impact [1]

2.1.2 Smart Technologies

Solutions based on IoT technologies are often addressed as Smart Technologies. The influenced sectors, such as mobility or health, are becoming part of a bigger, connected and smart ecosystem that represents the vision regarding the Smart Cities of the future.
IGI Global defines Smart Technologies as the technologies (including physical and logical applications in all formats) that are capable to adapt automatically and modify behavior to fit environment, use technology sensors, thus providing data to analyze and infer from and drawing conclusions from rules [16]. Machine learning techniques are used to improve performance, thinking, and reasoning about what to do next, with the ability to self-generate and self-sustain. These actions aim to integrate seamlessly these smart systems into modern society providing better safety, introducing more functionalities, increasing efficiency while reducing costs and more importantly, focusing on the needs of the consumers.
For example, Philips Lightning Hue Light Bulbs and Bridge system [17] provides users with a connected device for home automation. Users have the ability to customize their interaction though a smartphone, as well as connect their system to the wider world. With it, a user can control their lights remotely or link them up to the rest of the web, news feeds, or even their inbox. Another example is the iRobot Roomba [18], which is a smart product vacuum cleaner with iAdapt Technology (an advanced system of software and sensors) that enables Roomba to find its way around any shape or size of home, covering every area of floor multiple times for a complete clean.
However, the impact and the added value of these technologies is still being investigated closely and their adoption from the consumers progresses at a slower pace. Although some sectors are more advanced in creating new smart applications compared to others (for example smart home, smart mobility, smart health care and smart energy), most cities are investigating how they can achieve smart transformation of their overall infrastructure.

Figure 2.2: Smart City vision[2]

2.1.3 Smart Mobility and Connected Cars

One of the front-runners in the development of smart concepts and applications is the mobility sector (referred to as Smart Mobility). The term 'smart mobility' is used more and more to describe new services that aim to allow seamless, efficient and flexible travel across various modes of transport.
Moving from services to technology, the term 'connected car' is mostly used. The connected car technologies are defined as the presence of devices in an automobile that connect the devices to other devices within the car and/or devices, networks and services outside the car including other cars, home, office or infrastructure [19].
Internet connection can provide functionalities that warn of traffic, collisions and other safety alerts. As an example, concierge services from applications or automakers alert the driver of the time-to-leave to arrive on time from a calendar and sends text message alerts to friends or business associates to alert them of arrival times.
Connected car features the use of vehicle sensor data and vehicle actuators (such as screens or speakers) to increase safety, improve location or navigation services, boost in-car entertainment, optimize energy usage, improve traffic conditions, reduce the operational and maintenance costs, provide real time diagnostics and telematics services and connect the vehicle with the outside world. Every day, more and more applications are being developed, tested and produced in all the above fields [20].

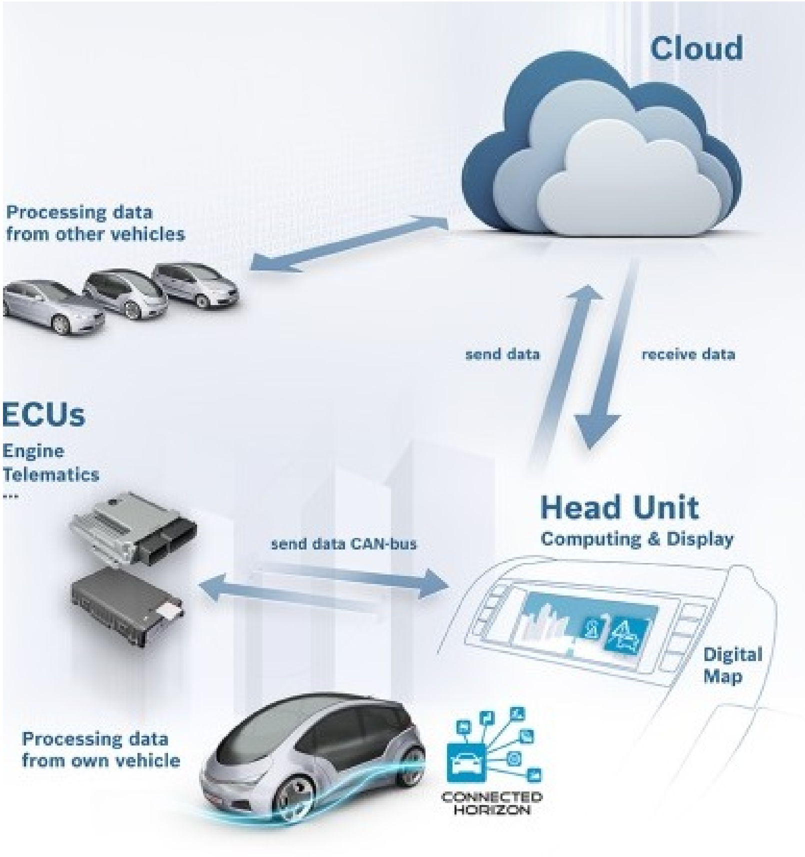

Figure 2.3: Connected car environment according to Bosch [3]

2.2 Developments

In this section, the connected car developments are explored from a global and regional (focusing on the region of Eindhoven) perspectives. Afterwards, VIBe is investigated in more detail regarding the initial goals, as well as past implementations of the VIBeX API.

2.2.1 Global developments

The connected car technologies have risen in popularity in the last 5 years immensely as part of the current and disruptive invasion of the smart technologies in various domains. Everyday new partnerships, merges, and collaborations are formed. The automotive OEMs still explore how they can maintain their market share either by trying to create a complete solution in house or by forming partnerships with expert companies. Latest example is the partnership between BMW, Intel (along with the recently acquired Mobileye) and Delphi (since May 2017) for autonomous vehicles.
Many of the contemporary companies recognize that eventually the car will be another connected to the cloud device, just like modern smartphones or wearables, being part of the smart ecosystem. Emphasis will be given to customization of applications that can be accessed from anywhere, even from the car. Each user will have a unique, customized profile that can be accessed by each mode of mobility (an initiative promoting this vision is MaaS Alliance) and would enable the user to choose

the navigation system or infotainment application that satisfies the requirements. The attention will be shifted from the device to the person using that device, creating a unique experience. This is very important as traveling from A to B which used to be the main goal of mobility, will now be just one of the many services available in the car.
Going more into depth regarding the global landscape, it is important to mention the number and total raised funding of connected car startups. In 2014, 369 startups were registered as connected car companies and raised 3.15 billion dollars [21]. One year later, the total number of companies almost doubled (622) and the funding increased to 17.6 billion dollars [22]. In 2016, a number of 1053 companies raised an amount of 51 billion dollars [23], showing the rapid growth of the connected car industry. The latest numbers for 2017 (recorded in August) indicate that 1148 startups raised 84 billion dollars [24]. A summary is presented in figure 2.4.

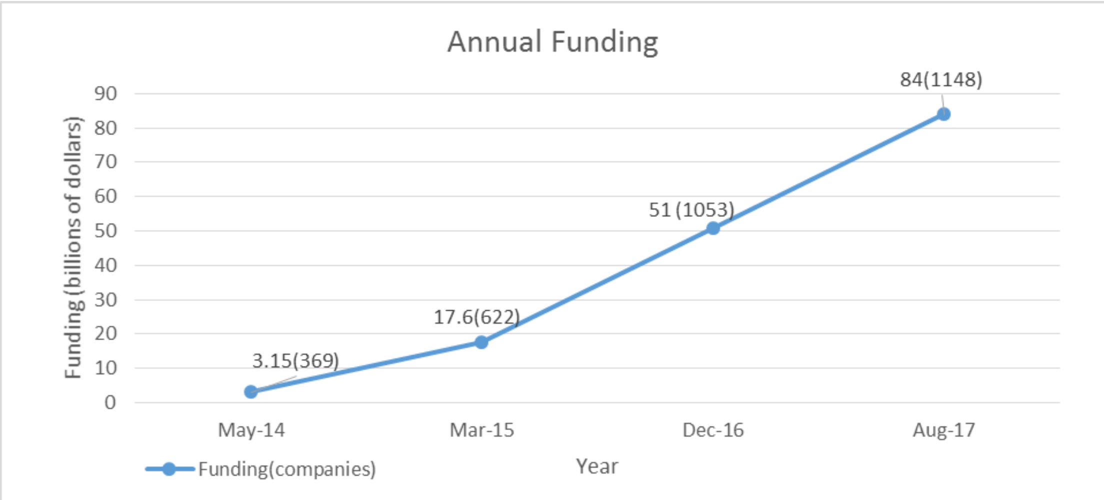

Figure 2.4: Accumulative funding from connected car startups

By the end of 2022, more than 120 billion euro revenue will be generated from leveraging connected car technologies [25], with a tendency to grow every year. That is easily verified by the projected number of connected cars on the road. Currently, almost 20% (15 million) of all new cars have some connectivity embedded. By the year 2020, the number of new cars sold with connectivity services will by around 75% (69 million) of all cars sold that year, reaching a total number of 220 million connected cars sold globally from 2014 until 2020.
All these new companies are not traditional automotive suppliers. They are active in Information and Communication Technology (ICT) being connectivity providers, service providers (car sharing, ride hailing, etc), infotainment developers, etc. This creates the need for partnerships as the car technology becomes more complex.
An important gray area is the choice between open standards and architectures or closed systems. The automotive OEMs still try to maintain their business model and market share, investing in creating complete, closed and customized solutions, while they explore the alternative path of initiating more collaborations at the same time. However, they are not experts in the ICT components of the connected car, which results in not achieving the desired quality level or the multi-domain functionality needed.

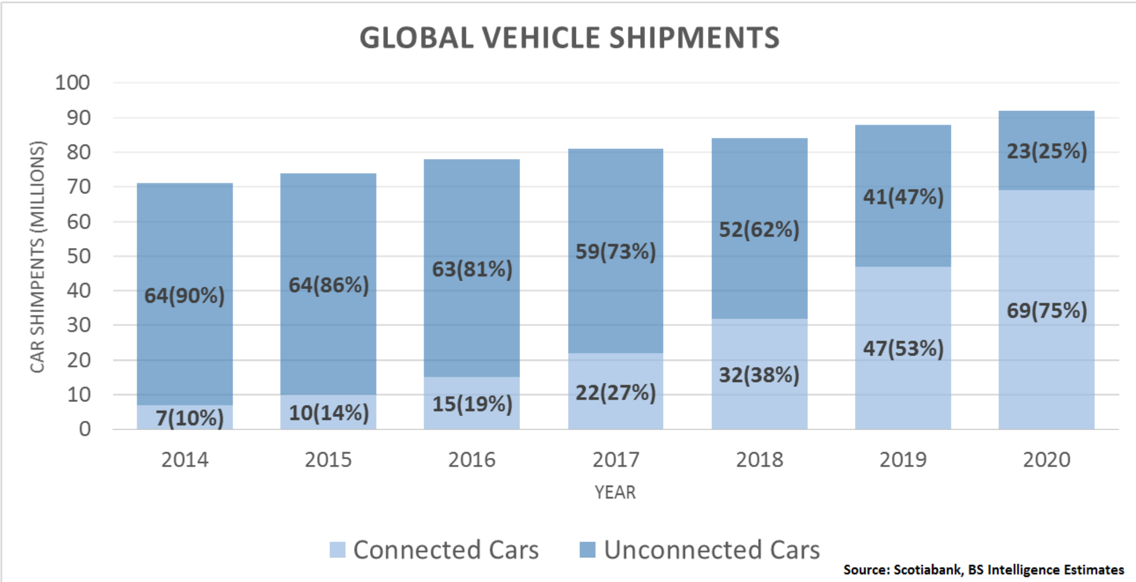

Figure 2.5: Number of projected connected new cars [4]

2.2.2 Regional developments

The Netherlands, and especially the Brainport region, have been active in the past years in initiating innovative projects related to Intelligent Transport Systems (ITS). Authorities, companies, universities, institutes are executing pilot projects to demonstrate the added value of these systems and their impact on traffic management in the Netherlands. To mention a few [26]:

- Initiatives: VIBe, Cooperative ITS Corridor, SPITS, Automotive Week, Shockwave Traffic Jams, Talking Traffic, Smartwayz.nl
- Collaborations: Connected, DITCM, CHARM, Connekt, CoastToCoast EV Connection, Connecting Mobility
- Authorities/Institutes: RWS, Traffic Innovation Centre, ANWB, ndw
- Knowledge institutes: TU/e, TNO
- Private sector: TomTom, Beijer Automotive, V-tron, ULU, Driessen and around 20 more

Focusing on the motto 'learning by doing', the region encourages the initiation of new concepts and projects with the purpose of gathering data and learning the challenges of the autonomous and connected vehicle technologies. Although this represents a proper foundation to build upon, a vision regarding the future of overall smart mobility is needed and not only initiatives related to ITS. Learning by doing simulates a more traditional way of entrepreneurship, while the imminent and incoming tsunami of smart technology demands a more visionary, proactive and decisive approach, addressing everyone that wants to capitalize on the new business opportunities on the horizon.
However, only a few companies are deploying business models regarding the connected car technology, trying to have a head start in the market. Most of the initiatives are side projects or proof of cases, which means that in Brainport, while there is some awareness about the technology, there is

lack of activities and leadership. If we consider the life cycle of innovation in business, feedback from interviews with people from the region (the list of interviewees can be found in Appendix B) suggests that the companies in the region are in the ’Valley of Death at the moment’(figure 2.8), meaning that initiatives and leadership towards the realization and commercialization of pilot and experimental projects are still missing.

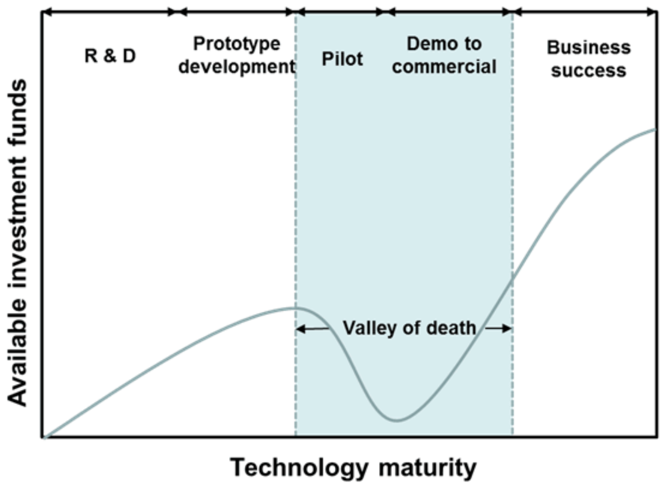

Figure 2.6: Technology maturity versus available investment funds: The Brainport region is in the "Valley of Death" when it concerns smart mobility.

2.2.3 VIBe - Vehicle for Innovation Brabant electric

The VIBe initiative was initiated by four organizations (TU/e, NXP, BOM and Autogroep Driessen). Founded in 2013, they started with the shared vision of an innovation platform and the realization that only through a collaborative environment it is possible to accomplish a swift establishment of new, smart technologies in the automotive and energy sector (among the partners of VIBe is also Enexis).

Initial Goals

VIBe was started to enable businesses in the Brainport region of the Netherlands to make best use of the existing infrastructure. The other reason behind VIBe is to integrate knowledge from various domains like EV technology and charging, weather, etc to enable smart application development. One of the subprojects of the VIBe initiative is aimed at standardizing the API as an interface against which several applications can be built (VIBeX API project). This is useful when individual applications want to communicate with each other and exchange information (figure 2.9). An important goal of the VIBe initiative is to set-up an ’International Living Lab’ example in the Netherlands to showcase the possibilities or the potential that arises when the cars become connected to the cloud and the infrastructure.

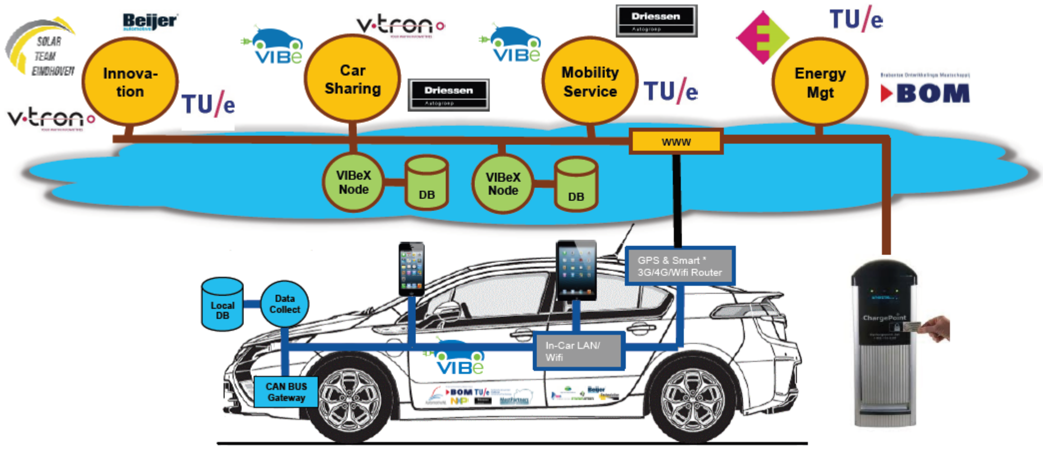

Figure 2.7: VIBe platform vision [5]

The VIBe partnership is a facilitator towards the standardization of the Information Technology (IT) infrastructure and for the additional growth of electric mobility. The intention is that through making a secure connection to a service or node in the cloud, data become accessible via easy to use APIs to multiple parties who develop different applications.
Summarizing, below the goals of VIBe are mentioned explicitly:

- Enable businesses in the Brainport region
- Integrate knowledge from various domains (EV technology, connected cars, smart application development)
- Standardization of the Information Technology (IT) infrastructure (effort via the VIBeX API project
- Creation of an 'International Living Lab'

Past projects

There are several projects carried out in the last 4 years that utilized the VIBeX API:

- Stella solar car application built by Ericsson (optimal route selection based on energy demand) in collaboration with Solar Team Eindhoven in TU/e
- Driessen car share application built by Accenture
- Enexis energy management application
- Vetuda connected car platform built by Beijer Automotive in collaboration with Microsoft [27]
- V-tron customers are using the VIBeX API to build their own end-user solutions. V-tron intends in the near future to rebuild their own applications integrating them with the API

From the above partners, TU/e, Driessen and V-tron are in the VIBe Steering Group committee, whereas Beijer Automotive and Enexis are partners of VIBe. All these projects investigated the use of an open API to communicate with a vehicle, having in mind that it can lead to a standardized API, which is in line with the initial goals of VIBe. Unfortunately, there is no public information regarding the aforementioned applications (except for the Vetuda platform) and the only available source were the VIBe steering group committee members.

2.3 Related work

As mentioned in section 1.2.4, the cloud computing technology and its applicability in smart mobility is investigated in this report. Thus, it is necessary to explore the principles and characteristics of cloud technology [28] and how they can be implemented in the prototype platform.

2.3.1 Cloud computing basics

Cloud computing (an overview is presented in figure 2.8) is a new form of Internet based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications, and services), which can be rapidly provisioned and released with minimal management effort.

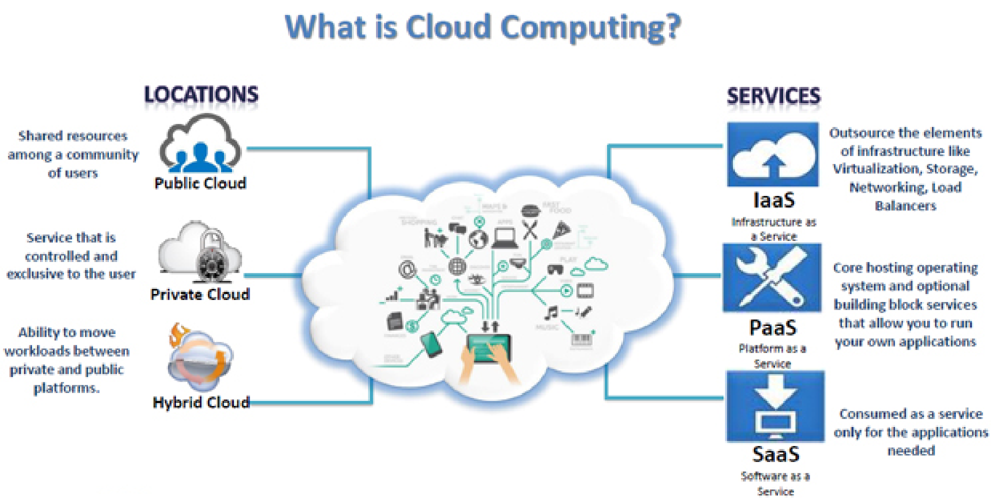

Figure 2.8: Cloud computing overview [6]

Basically, cloud computing allows the users and enterprises with various capabilities to store and process their data in either privately owned cloud, or on a third-party server in order to make data accessing mechanisms much more easy and reliable. Data centers that may be located far from the user, ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale.
In the next sections, the basic characteristics of cloud computing are explained in more detail. More specifically, the cloud deployment models are initially investigated. A cloud deployment model represents a specific type of cloud environment, primarily distinguished by ownership, size, and access.

Afterwards, the cloud service models are explored, which represent a specific, pre-packaged combination of IT resources offered by a cloud provider. This design choice defines which features are the responsibility of the cloud provider and which of the customer.

2.3.2 Cloud deployment models

There are three main cloud deployment models provided by the majority of the cloud providers: public, private and hybrid cloud. Below, the differences and similarities of these models are explained in more detail [29].

Public cloud

The cloud infrastructure is provisioned for open use by the general public [29]. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider and can be acquired via a monthly subscription based paying model.
The purpose of this model is to gain experience with cloud computing, rather than to create an enterprise cloud infrastructure. The advantages of public cloud mainly include automatic scalability, elasticity, minimal management, and maintenance. All these are handled by the cloud provider. This services can be acquired with zero upfront costs as no installation on premises is required. The disadvantages of this model are the limited security options and the dependence on reliable Internet connection. Utilizing a data center of a cloud provider means that the customer relies on the security of that provider, which is caused by the fact that all the data are hosted on shared computational resources. Additionally, to use the public cloud, a reliable Internet connection is needed as primarily remote connection services are used.

Private cloud

The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
This model has high upfront costs and needs concrete, long-term planning, as the goal is to create a customized, unique cloud infrastructure owned by a single organization. The purpose of this model is to keep tight control of the cloud data and applications and it is ideal for security sensitive systems. Unlike the public cloud model, it is not bound by a constant Internet connection, as private local networks can also be used. This results in generally better performance compared to the public cloud model. However, it is harder to scale as it requires more time and capital. Furthermore, the maintenance and management are completely a responsibility of the customers and their IT departments.

Special case: A particular case of the private cloud model is the Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
The community model is similar to private cloud with the difference that it is handled by a closed

group of customers instead of just one organization. It is ideal for a small number of companies that aim to create a shared, private cloud infrastructure.

Hybrid cloud

The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).
This solution is the most popular of the four. It combines two or more of the aforementioned models to create a hybrid system. A commonly used combination is using public cloud services and private data storage facilities. In that way, the advantages of both models are preserved and the drawbacks are minimized. In addition, there is still a need for IT personnel to manage and maintain the private components of the cloud. However, the main disadvantage of this option is the introduced added complexity (not only for the IT personnel responsible for the management of the infrastructure but also for the developers, as a clear overview and design of the system should exist regarding which components are used via on-premise private solutions and which via public services that are used at that particular moment), which makes this solution suitable for enterprises that are already experienced with cloud technologies (either public or private).

Overview

All three options have advantages and disadvantages that make them suitable depending on the requirements and purpose of the designed system. Below, an evaluated overview of the characteristics of these models is presented:

	Public	Private	Hybrid
Payment model	Pay-as-you-go	Upfront payment	Both
Initial investment	++ Lower upfront costs	- Higher initial investment but lower total costs	- Considerable initial costs (less than private but more than public)
Availability	++	++	++
Security	- Generic security, not suited for compliance and security sensitive data	++ Ideal if security is a strong requirement	+ Option to use the private cloud for security applications
Management and maintenance	++ Handled by the cloud service provider	-- Handled by the company IT department	- The private cloud components are managed by the company
Tenancy	++ Multiple tenants on a shared basis	- Only one tenant	+ Some parts can support multi tenancy
Scalability	++ Economies of scale, all scalability is handled by the provider	- Not easy or cheap to scale after the initial setup	+ The on-premise components will not be easy to scale (similar to private)
Setup	++ Fast deployment as everything is already setup by the provider	-- A lot of effort in setting up the on premise components	- Effort on setting up the desired private cloud
Performance	- Still relying on internet connection	++ Use of private networks or internet	+ Generally fast as you can use private network

Figure 2.9: Cloud deployment models overview

2.3.3 Cloud service models

As mentioned before, the Private Cloud deployment model means that all the cloud infrastructure is on-premise. In this case, the cloud provider has the task to setup the initial cloud environment and afterwards, the management, ownership, and responsibility of the cloud platform go to the customer. In case of a public or hybrid model is chosen, the following service models can be used [30]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

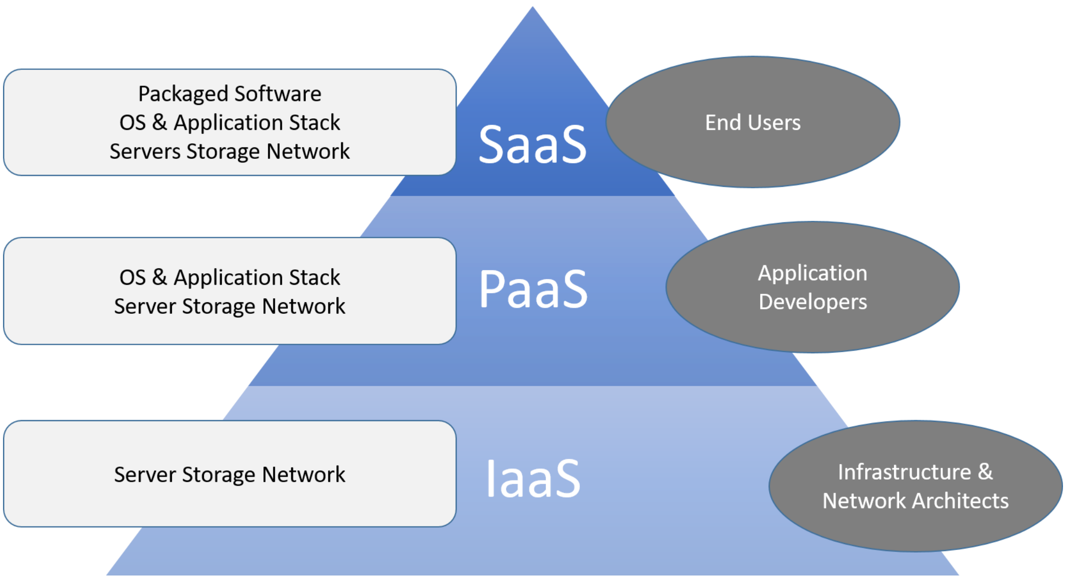

Figure 2.10: Cloud service models overview

Infrastructure as a Service

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications. IaaS providers offer these cloud servers and their associated resources via dashboard and/or APIs. IaaS clients have direct access to their servers and storage, just as they would with traditional servers but they are provided with more scalability. Users of IaaS can outsource and build a 'virtual data center' in the cloud and have access to many of the same technologies and resource capabilities of a traditional data center without having to invest in capacity planning or the physical maintenance and management of it.
IaaS is the most flexible cloud computing model and allows for automated deployment of servers, processing power, storage, and networking. IaaS clients have true control over their infrastructure whereas users of PaaS or SaaS services have not. The main uses of IaaS include the actual development and deployment of PaaS, SaaS, and web-scale applications.

Platform as a Service

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
The PaaS model typically provides a platform on which software can be developed and deployed. PaaS providers abstract much of the work of dealing with servers and give clients an environment in which the operating system and server software, as well as the underlying server hardware and network infrastructure, are taken care of, leaving users free to focus on the business side of scalability, and the application development of their product or service.
As with most cloud services, PaaS is built on top of virtualization technology. Businesses can acquire resources as they need them, scaling as demand grows, rather than investing in hardware with redundant resources.

Software as a Service

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
SaaS is the most familiar form of cloud service for consumers. SaaS moves the task of managing software and its deployment to third-party services. Among the most familiar SaaS applications for business are customer relationship management applications like Salesforce, productivity software suites like Google Apps, and storage solutions brothers like Box and Dropbox.
Use of SaaS applications tends to reduce the cost of software ownership by removing the need for technical staff to manage install, manage, and upgrade software, as well as reduce the cost of licensing software. SaaS applications are usually provided on a subscription model.

Overview

To summarize, the three different cloud service models represent the abstraction layer of the cloud infrastructure [28]. IaaS has the purpose of hosting a cloud environment providing full customizability to the cloud developer. PaaS handles the infrastructure and middleware to provide an environment for development and deployment of applications. Finally, SaaS addresses the consumers of cloud applications providing a complete, end-to-end solution.

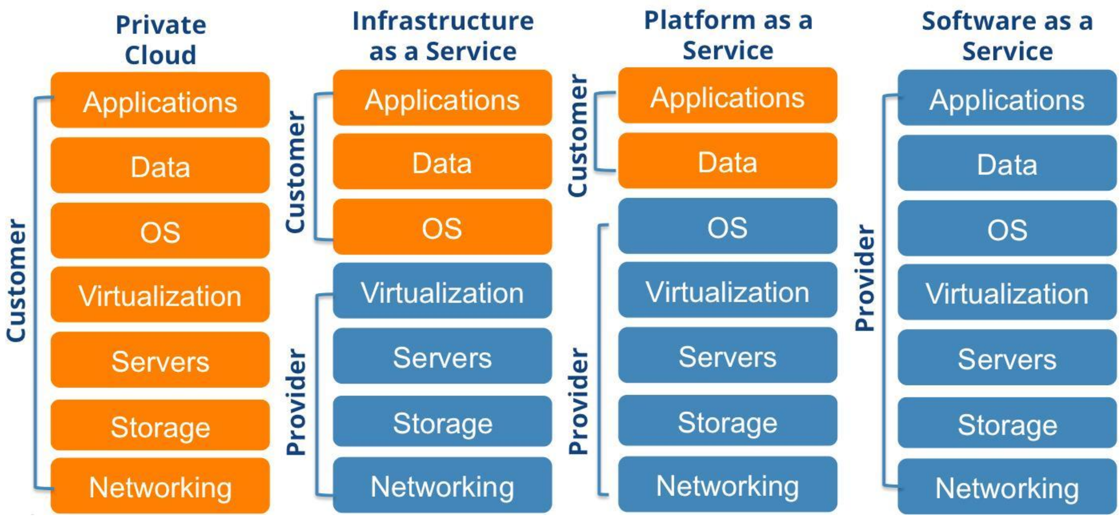

Figure 2.11: Cloud service models overview [7]

3 VIBe Recommendation

In this chapter, an overview of the recommendation to VIBe Steering Group is presented (the full recommendation can be found in Appendix C). These results inspired the concept of the smart mobility platform and consequently, the prototype platform that was developed. It is important to note that the recommendation is intended for a managerial audience and therefore, it is written in a different style. The purpose of the recommendation was to provide an alternative view on a possible way forward for VIBe and the region of Eindhoven. As mentioned in section 1.5.1, the main source of information were interviews with people from the industry around Eindhoven (for a detail list see Appendix B), local events and conferences and publicly available information.
The recommendation proposes an alternative vision, embracing more smart sectors. However, the rest of this report focuses on a smart mobility prototype platform, as a starting point, which could be linked with more domains in the future. However, these potential links were not included in the scope of this project.

3.1 Analysis

After investigating the global and regional developments regarding smart technologies and especially smart mobility, a SWOT analysis (strengths, weaknesses, opportunities and threats) for VIBe and the region was conducted, in order to get an indication of the potential of the region as well as identify warning signs of wrong direction.

3.1.1 Region

This section presents the SWOT analysis for the region of Eindhoven (an overview is given in figure 3.1). More specifically:

Strengths

- **Location:** VIBe and in extension the Brainport, reside at the heart of Europe, having the potential to become a technological hub of the continent. In addition, Brainport is considered to be the smartest part of Europe, accommodating some of the most innovative companies of the continent, knowledge institutes and international experts.

- **Experienced in ITS:** The Netherlands is a leader in Traffic Management innovation. A lot of institutes, projects, initiatives confirm that every year. All these projects have provided very

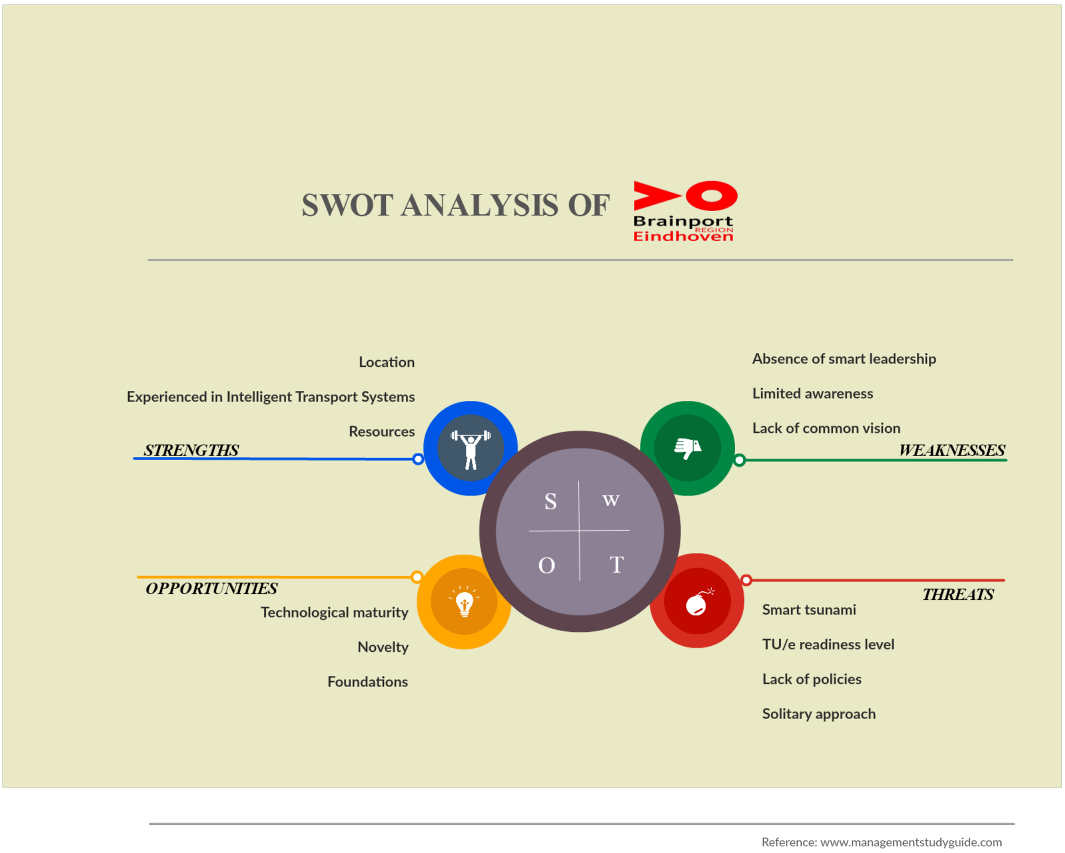

Figure 3.1: Brainport SWOT analysis matrix

useful test data and learning lessons for the future and also prove that the region is open to innovation.

- **Resources:** The individual growth of the companies in Brainport is suggesting that there is room for investments and entrepreneurship. The region is expanding business wise and the question is ultimately, how to better spend the available resources, serving the right vision.

Weaknesses

- **Absence of smart leadership:** As mentioned before, many prerequisites of being successful in the smart domain exist in the region (capital, experts, companies, innovation). However, there is one element missing: leadership. While in the past companies such as Philips could drive a region forward, today no such company exists, especially in the smart mobility area. Consequently, it is highly important to define clear roles and responsibilities, as at the moment the absence of a clear vision creates obstacles in defining the way forward (investments, concepts, planning, etc).
- **Limited awareness:** During the Automotive Week 2017, a survey from Connecting Mobility was presented regarding the perception of the connected car among professional users. Almost 3 out of 4 people were not aware of the term connected car. That example indicates that the

region is still not aware of the potential of smart mobility (and in extension smart technology) and what that means for the consumers or the business players of the region.

- **Lack of common vision:** The absence of leadership influences heavily the the creation of a common vision regarding smart mobility and smart cities. The lack of such a vision about the region creates uncertainty regarding the direction of new (or existing) smart projects and eventually fails to convince the region about the potential of these technologies. It also hampers the placement of smart investments in platforms and toolboxes that could transform the region (such as open and/or multi-vendor platforms).

Opportunities

- **Technological maturity:** The days when the ideas were mature but the technology could not support them, are passed. Currently, technology is rapidly progressing (especially in the mobility sector) and is able to satisfy great demands. It is the first time in history, that it is a matter of putting the right ideas in place rather than focusing on the technology behind it.

- **Novelty:** Smart mobility and smart technologies have yet to be evaluated and standardized. While significant progress is being done worldwide, no one has all the answers yet. This gives a huge opportunity to innovate and create original concepts in this domain.

- **Foundations:** The region of Eindhoven, despite being relatively small, has proven to be open to innovation. From the ITS projects in Helmond to smart lighting grids in Eindhoven, the region has shown that it welcomes new smart projects.

Threats

- **Smart Tsunami:** As mentioned in previous sections, a smart tsunami is coming our way, meaning that many technology sectors are changing. Traditional practices and business models are being replaced by new ones, that focus on the user instead the device itself. Cars are becoming greener, smarter and more software oriented, demanding a change in the existing business models. Therefore the question is how can each company find the way to be part of this smart tsunami, utilizing the opportunities that arise and being agile enough to adapt to the new conditions. Considering the second incoming wave related to augmented reality, the danger of staying at the observer's role rather than actively engaging in these technologies is real.

- **TU/e readiness level:** The Eindhoven University of Technology is an important source of neutral and pure innovation. However, it is not clear yet if the university is ready to embrace the smart technologies, addressing not only the application layer but the complete infrastructure. Therefore, it is important to organize smart leadership inside the university as well.

- **Lack of policies:** The slow adaptation of the policy makers (in a country but also at European level) could pose a challenge. The struggle to resolve the data ownership and privacy issues is slowing down the process of enabling connected car technologies. However, the need to establish legislation is identified in the Declaration of Amsterdam as highly critical towards the development of connected car technologies and intelligent transport systems.

- **Solitary approach:** Historically, in case of new technology, most companies try to create complete, vendor specific solutions and claim novelty. This often costs capital, effort and time to all the competing companies, who engage in a last-man-standing competition. If the region is drawn into this process, Brainport will be falling behind compared to other regions that approach the smart transition from a collaborative and unified perspective.

3.1.2 VIBe

Similarly, the SWOT analysis for VIBe is explained in this section (similarly an overview is given in figure 3.2)

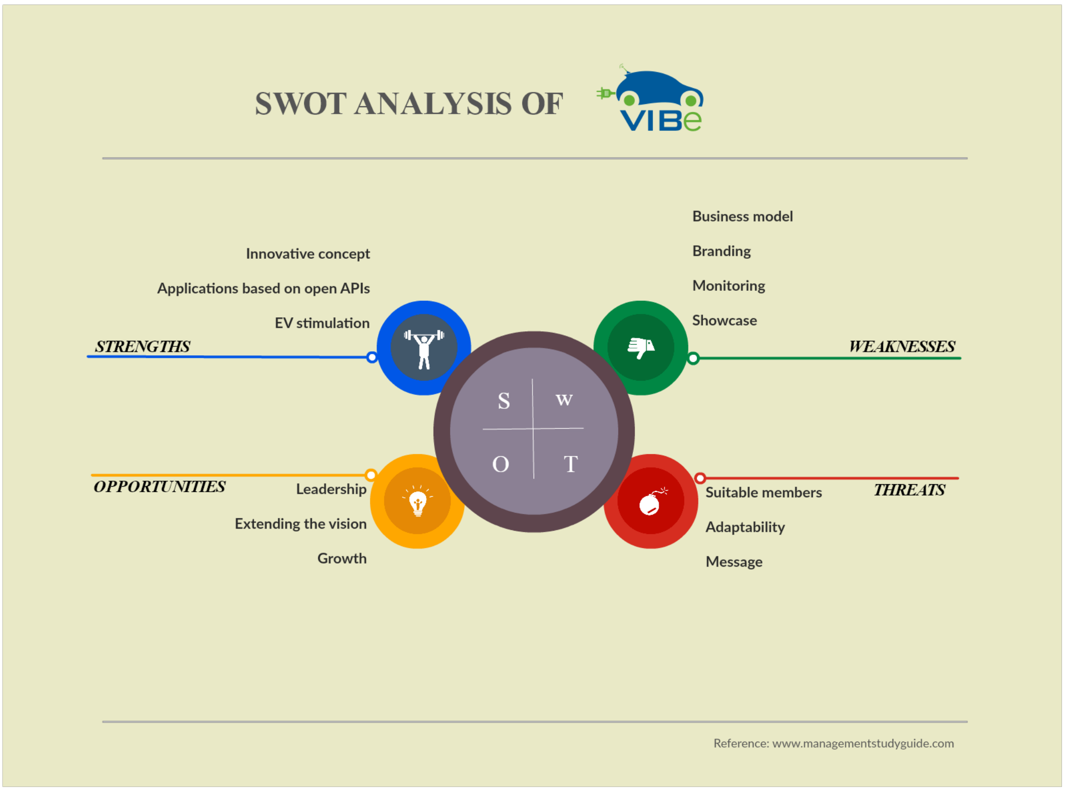

Figure 3.2: VIBe SWOT analysis matrix

Strengths

- **Innovative concept:** The initial objective of VIBe was to design and develop an open platform for electric vehicles in the format of a Living Laboratory. The rationale behind this objective was to stimulate entrepreneurs and companies to develop and test their innovative products and services in a real condition environment and produce results faster and cheaper. Innovation is the reason for existence of VIBe and should remain us such.

- **Applications based on open APIs:** VIBe has succeeded in initiating discussions about open architectures and joined initiatives with collaborative spirit. More specifically, VIBe has been responsible for the development of applications by members and partners using the VIBeX API.

- **EV stimulation:** The Netherlands are among the leaders globally in EV adaptation. VIBe contributes by stimulating EV leasing schemes to employees of the VIBe company members but also, by providing a fleet of EVs to be used for application development.

Weaknesses

- **Business model:** The industry is not convinced regarding the value of open architectures and standards, questioning the existence of a viable business model. The reason for that is that companies approach open standards and architectures from a business perspective, as companies try to protect their intellectual property and individual commercial value generators.
- **Branding:** The publicity of VIBe is not high enough at the moment, meaning that apart from the current members and partners, very few are aware of VIBe and its mission.
- **Monitoring:** The first idea behind VIBe was to create a communication platform in order to inspire companies to develop their own application using that API. However, after the first release of the API, there was not enough monitoring of the updates and modifications implemented by the companies that used the API.
- **Showcase:** Although there have been a few implementations of the VIBeX API, still the absence of a strong showcase is hampering the effort to attract more partners and to inspire the region. Several times during the interviews with people from the industry, the remark was made that a working proof of case is missing. Apart from the API, there is no establishment of open innovation platforms yet and the absence of relevant showcases is an issue.

Opportunities

- **Leadership:** As mentioned in the regional SWOT analysis, the region is suffering from leadership absence. VIBe has a real chance to tackle this problem by emitting the right vision about the smart cities of the future and provide a clear direction.
- **Extending the vision:** Assuming leadership means to not only address one smart domain (e.g. mobility). The vision need to be adapted in the smart ecosystem context. Having mobility as a showcase, the first steps can be made into understanding how the bigger picture looks like.
- **Growth:** Through embracing the smart ecosystem vision, more sectors are addressed that could also be linked with smart mobility (smart energy, smart business, etc). That could give the chance to VIBe to connect all these domains and inspire multi-domain solutions.

Threats

- **Suitable members:** It is key to have members and partners that believe in the same vision. In this way, everyone is an ambassador of VIBe. In the opposite case, it is difficult to set a strong example in the region and therefore, the message will not be persuasive enough regarding the urgency to adapt to the smart transformation.

- **Adaptability:** The current developments have shown that creating an open API and providing it to interested parties is not enough to create the avalanche effect that can change an industry and alter mindsets. Adaptation is needed to introduce something bigger that demonstrates the added value of using smart technology in an open and collaborative way.
- **Message:** VIBe should not be seen as a business entity. VIBe should represent the vision on what it means to be smart and a plan on how to get there. It should be a hub of members that believe in that vision and a facilitator of projects aiming at the right direction.

3.2 The vision

It is clear from all the above that a collaborative spirit is needed in order to survive in the new smart reality. The private sector, universities or research institutes, and the authorities need to work closely together, driving innovative and disruptive projects, as well as creating inspiring smart showcases.

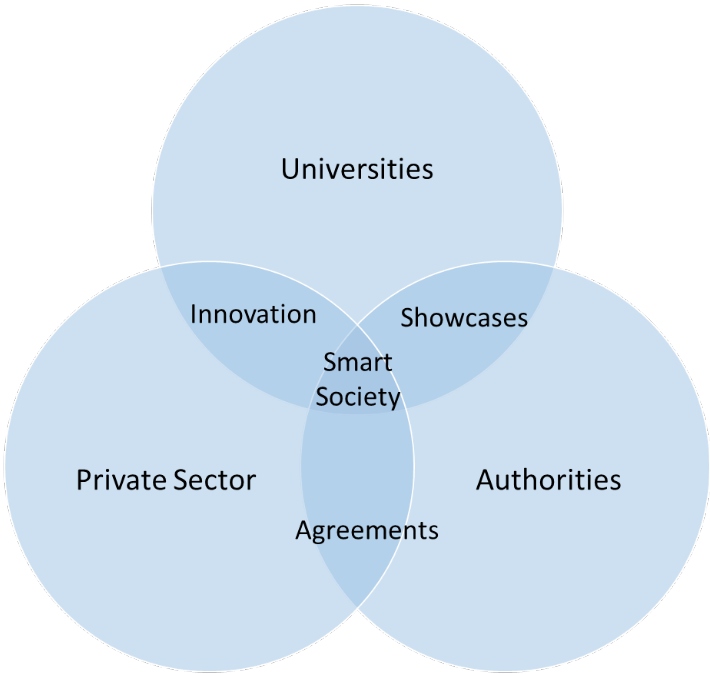

Figure 3.3: The Way Forward

A lot of cities are already claiming that they are on the way to become 'Smart Cities' or 'Smart Societies'. The common denominator is always the efficient collaboration of the above three partners. The private sector companies form mutually beneficial agreements with the authorities regarding use of smart systems and handling of public data. The authorities rely on research and educational institutions that can provide showcases to demonstrate the added value of the smart platforms, utilizing the core purpose of these institutions which is to promote new technologies and to provide a peek into the future. Finally, the business world combines forces with the universities to create innovative concepts that can also be implemented in the real world. The combination of all these factors is able to create the necessary foundations to transform a city into a smart society, having as core belief that the cities of the future are people oriented.

3.2.1 Private sector role

As mentioned in the previous chapter, the region has a lot of potential when it comes to having the necessary financial and technical infrastructure to realize innovative concepts. However, leadership and awareness of the actual impact of these technologies are missing, which explains the delay in adapting to the new reality.
The role of the industry around Brainport involves adopting a collaborative spirit and dismissing the solitary approach. The realization that the transition to the smart era can unravel huge business opportunities for everyone involved only if the region follows a unified approach.
However, there are a few aspects that need attention. Firstly, working together also means embracing the open mentality. While this does not mean that everything should be open and free, it certainly means that open development and sharing data under specific agreements are integral ingredients. The individual vendor products that add commercial value to each company will remain black boxes but the interfaces should be open and shared. The industry needs to be an example to follow on how to adapt to the new smart reality, showing vision and decisiveness while understanding the numerous opportunities that will arise, if a more collaborative and open mentality is adopted.

3.2.2 Universities Role

The Eindhoven University of Technology is a strong founding partner of VIBe. Being the place where innovation starts and heavily involved in smart mobility concepts, the TU/e should play an elevated and crucial role.
Supporting open standards and architectures is a common belief of TU/e and VIBe. That, combined with young engineers with fresh ideas can be the spark to create these applications and be the source of inspiration regarding smart (mobility) applications. These students are engineers that will play an important role in the next 10 years in establishing the region as a smart era leader.
However, as mentioned in the SWOT analysis, the TU/e has not still reached the desired engagement level. It is crucial that a world-renowned institute as TU/e understands what it means to be part of the new smart reality. By taking actions such as organizing leadership at a higher level for the purpose of connecting the different departments and smart projects. A unified communication stream is desirable, as well as a integration of different projects that use the same infrastructure (especially related to cloud computing). The university has a great chance to be the leader of the transition to smart, cloud connected platforms, starting from mobility applications and expanding to all smart domains. In that way, partnering with companies, very innovative projects can be created that would provide a chance to build up on previously earned valuable lessons. TU/e can take the lead to demonstrate what it means to be smart in the future and in this way prepare for the next tsunami related to augmented reality.

3.2.3 Authorities Role

The role of authorities is very crucial towards enabling connected car technologies and smart solutions in the region. Policies and regulations can really accelerate the establishment of the required infrastructure.
VIBe and the local authorities have several goals in common. Similarly with the TU/e, one of them is the support for open standards and architectures. The first steps have been made from the local authorities by publishing guidelines for using open architectures (IoT Principles document). That shows eagerness to build the legal foundations for smart city and mobility initiatives. On the other hand, the

business world understandably wants to protect its business models and is very skeptical towards that mentality. If VIBe and the authorities align their goals, it could directly link the policy making with the business world and the public interest, which could lead to more fruitful discussions and agreements.
The collaboration of VIBe and the private sector of the region with the authorities is positive so far but there is room for improvement. The local authorities really want to involve the industry in creating innovative concepts and they want to form beneficial collaborations. In addition, they consider VIBe to be a very good example of an attractive initiative, that can be inspiring for the government to invest in. Therefore, it is definitely interesting to investigate ways for closer cooperation. The local authorities aim to introduce several smart city initiatives in Eindhoven, which combined with smart mobility and intelligent transport systems can really make Brainport a high-tech hub of the region. To realize that transition, the authorities decided to create a smart leadership team that will be able to lead the region with decisive actions. It was widely accepted that until now a lot of capital was spent without having a clear goal and understanding of what it means to be smart. The authorities decided to change that which provides a great opportunity for new agreements, showcases and projects that convey the right vision.

3.3 Strategy management

To capitalize on all the above information, a strategy needs to be created taking advantage of the strengths and opportunities but also try to mitigate the threats as well as improve the weaknesses. A clear vision needs to be formulated regarding the future of the region in the smart era and how VIBe can contribute.

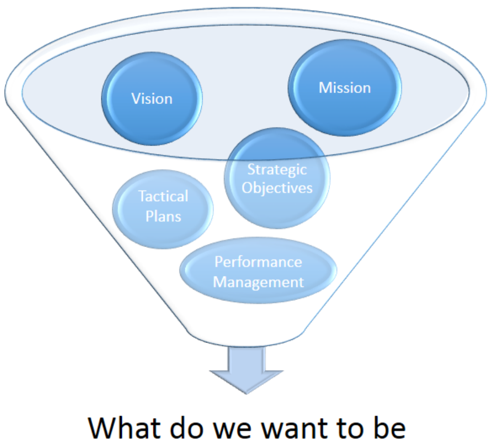

Figure 3.4: Strategy management

3.3.1 Vision Statement

VIBe was initiated with the purpose to promote open architectures and standardization, utilizing EV technology. This innovative mindset should remain a core element in VIBe. However, it is clear by

now that addressing only one domain is not enough. The realization that the smart technologies should be approached as a whole is needed.
Therefore, VIBe should represent the long term vision that the region needs. It is important to inspire the region to see the bigger picture and how to transform their businesses (regardless of the domain) to be smarter and in this way, unravel new business opportunities.
The Why of VIBe, the purpose, should be to help Brainport come together, adopt a collaborative spirit, accept open architectures and standards and promote innovation in all domains. The How of VIBe is to start from smart mobility and connected electric vehicles as a means to show what it means to be smart today using smart platforms designed to enable innovation in an open and fast way.

3.3.2 Mission Statement

The vision statement can be the foundation of the mission of VIBe and in general the region in the connected car and smart domains context. More specifically, the following questions can shape the mission statement:

- ***Where do we want to be in 5 years?***
  As a region, we want to show leadership and initiative in the smart domain. On one hand, we need to reach a common understanding and form agreements internally to face the smart tsunami together.
  On the other hand, we want Brainport to be used by every major stakeholder (OEMs, authorities, suppliers) as a testing environment for the technologies that will shape our future. We want to be considered innovative (EV technology, ITS initiatives, automated driving,etc), resourceful (past experiment data, lessons learned, history), cooperative (open mindset in regulations and policies), proactive (ensuring the required technology is in place) and most importantly, united (authorities, companies, knowledge institutes all together). We want to be at the front of the innovation and not trying to keep up.
  Ideally, VIBe will have facilitated towards the creation of smart applications on several domains using open standards and multi-vendor platforms. Mobility, will be the leading sector, demonstrating the added value of the applications and the infrastructure behind them. In this way, VIBe can also attract more members which will provide opportunities to create multi-domain solutions.

- ***What do we do?***
  On the regional level, we accelerate the establishment of the necessary smart infrastructure and encourage the local industry to initiate more pilot projects not only in ITS domain but in smart mobility in general. The goal would be to have in place a smart platform, where everyone can develop smart applications easily. This is the necessary extension from supplying an open API to supporting an open, smart, multi-vendor platform.

- ***How do we do it?***
  VIBe and the Technical University of Eindhoven are already discussing the establishment of an open multi-vendor platform that will enable rapid prototyping of smart mobility applications. Similarly, this can be extended or connected with similar projects in other smart sectors to form a unique smart city profile. By providing such infrastructure, we acquire a vast amount of data and access to most of European and international projects that can provide valuable information regarding the deployment of these systems. In addition, in this way an acceleration

in policy establishment can be achieved. Using the VIBeX API along with platforms similar to the aforementioned smart mobility platform, showcases of applications can be created fast which is the first step in understanding what it means to be part of the smart ecosystem. Finally, rethinking the way of approaching investments and projects at the authorities level is necessary, as the people with the right vision and drive need to be involved.

- ***Whom do we do it for?***
  These are the necessary steps for the region to establish a profitable, innovative but mainly sustainable future in the smart domain context. The private sector companies, the authorities and the research institutes need to jump on the smart wagon as soon as possible. As explained in the beginning of the document, there are already cities that take serious steps towards a long term vision in a unified and collaborative way. Failing to follow a similar approach will result in isolated efforts without a common vision.

- ***What value do we bring?***
  As mentioned above, the region will bring the needed leadership in implementation of innovative, smart projects. In addition, for the region to be able to do that, a common consensus is needed among the stakeholders. VIBe can create that value by being the bridge between the authorities, the industry and TU/e.

3.3.3 Strategic Objectives

From all the above, it is necessary to derive more specific, strategic objectives: **For the region:**

- Organize smart leadership at authority and university level
- Form agreements and alliances among the three main sectors (private sector, authorities, universities)
- Initiate more smart projects, starting from mobility and connected car technology but extending the vision to all smart domains
- Ensure the establishment of open, multi-vendor smart platforms (such as the TU/e Smart Mobility Platform)
- Smart investments with clear goals and following a clear vision
- Support new promising startups with funding (Lightyear, Amber Mobility, etc)

For VIBe:

- Be the vision that the region needs
- Adapt the vision to include all smart domains towards the establishment of the smart city through mainly addressing smart mobility and its connection with other domains.
- Promote open mentality and fight vendor lock-ins
- Take the lead to facilitate closer collaboration between the authorities, the industry and the university
- Maintain and promote the EV mentality but support all types of vehicles

3.3.4 Tactical Plans

The question now is how to realize these objectives and what are the steps and guidelines to follow. Regarding the VIBe objectives:

- Evaluation of the current members and partners regarding support of the vision of VIBe
- Effort to include partners from different domains (cloud providers, authorities, smart energy, smart city, etc)
- Create showcases of smart multi-vendor, mobility applications using open standards (VIBeX API)
- Help TU/e by proposing (or investing in) applications or prototypes to be built regarding the Smart Mobility Platform
- Integrate the feedback from the use of the VIBeX API into a new unified version

3.3.5 Performance Management

It is highly important to have an agile approach in the new smart era. Developments are happening so fast that in order to keep up, regular refinements on the strategy and planning should be scheduled. To make that measurable, at least every 6 months the members of VIBe should meet and adjust the next steps. Close monitoring of the developments regarding open platforms, the VIBeX API and applications developed is needed. In order to be successful in the connected car era, agility and proactive behavior play an important role.

3.4 Presentation to the VIBe Steering Group committee

The presentation to the VIBe Steering Group committee was not possible to take place as the steering group members were unavailable during the project available time. However, the recommendation was published via email, along with an executive summary. The discussion will happen at a later stage regarding the direction of VIBe and will be organized by the program manager of VIBe.

4 Requirements Elicitation

In this chapter, the most important requirements for the prototype of the smart mobility platform are addressed, as well as the stakeholders. The complete list of requirements for the smart mobility platform can be found in Appendix D.
Before the system requirements for the prototype platform are presented, the business requirements of the platform are investigated. More specifically, the business requirements relate to a specific need that must be addressed to achieve an objective and address the 'why' for a project [31]. Once the business requirements are established, the system requirements are defined.
Furthermore, the requirements were derived focusing on the features mentioned in section 1.2: ease of development and deployment, vendor independence, open development and multi-vendor functionality. Apart from the description and rationale of the requirements, the relevant stakeholders as well as a possible use cases are presented.
As mentioned in section 1.1, the PDEng Thesis of Preethi Ramamurthy [5] is related to open development as well. However, the main objective of that report is to address the issues with developing an open platform towards reducing the range anxiety problem of EV users in the form of an open API. Therefore, the requirements in that report are related to the VIBeX API and more specifically address security and privacy, performance and scalability, availability and resilience, as well as evolution and interoperability. The requirements for the prototype of the smart mobility platform are related to the underlying cloud infrastructure rather than the API VIBeX API itself and are influenced by the factors mentioned in section 1.2.

4.1 Stakeholders of the platform

It is necessary to investigate the stakeholders of the platform and their view of the system. At a higher level, the stakeholders are mainly the university (TU/e), private companies (at least in the Brainport region), the regional authorities (Gemeente Eindhoven), individual developers and vehicle drivers. Depending on their view on the system, they can be further categorized as Infrastructure administrators, Infrastructure developers, Data providers, Data service providers, Data service developers and Data users [32]. The next figure displays the correlations between the stakeholders and their roles.

	Infrastructure provider (physical)	Infrastructure administrator (cloud)	Infrastructure developer (cloud)	Data provider	Data service provider	Data service developer	Data User
TU/e		X	X	X		X (via student projects or competitions)	X
Private Vendors				X	X	X	X
Public Authorities	X			X	X	X	X
Individual Developers				X		X	X
Drivers							X

Figure 4.1: Stakeholders of the platform and their roles

- **Infrastructure provider (physical):** The availability of physical infrastructure such as test tracks, road networks and road side units (RSUs) is necessary to deploy applications. Especially the public authorities can provide the necessary permissions to use such infrastructure elements.
- **Infrastructure administrator (cloud):** This role belongs to the administrators of the smart mobility platform (in our case, it is proposed to be the TU/e). Their concerns involve user access management, maintenance, availability and scalability of resources.
- **Infrastructure developer (cloud):** These stakeholders are responsible for developing functionalities for the cloud platform and extending its capabilities. They are driving automation processes regarding development or deployment of applications and towards creating a true multi-vendor and open platform. This can be achieved in cooperation with Data service providers.
- **Data provider:** This group is closely associated with the Data service developers. Promoting an open platform and multi-vendor functionality, sharing of data is necessary and data lock-ins are not desirable. Therefore, private companies, public authorities or even developers can provide data. Eventually, the platform could host a vast amount of data for research or development purposes for the Data service developers and Data Users.
- **Data service provider:** The cloud providers and infrastructure component suppliers belong to this category. In case of private service providers, their main concern is commercial value and profit. On the other hand, the public authorities can provide such services to explore ways to make existing public systems more efficient and improve their quality.
- **Data service developer:** These are the application developers of the smart mobility platform. They expect easy development and deployment of applications, comprehensive developer tools and environments, as well as available data sets and testing infrastructure.
- **Data user:** Referring to the user of the application that requires access to in-vehicle data. These stakeholders expect support during the lifetime of an application such as updates to improve stability and functionality. Application management function such as installing, updating, enabling, disabling, and removing of applications are expected [32].

4.2 Business requirements

The business requirements of the platform demonstrate the added value of the platform from a business perspective. They address directly the needs of the region that were explained in the SWOT analysis in chapter 3. For the prototype platform only BR05 was taken into account. It was not possible to use the VIBeX API to satisfy BR02, as well as there was no possibility during the available time to investigate BR03. Finally, BR01, BR04 and BR06 are not relevant to this prototype and are more applicable for the long term potential of the smart mobility platform.

Requirement	Description	Rationale	Stakeholder	Use Case
BR01	**The system should be hosted in the context of TU/e**	**The platform needs to be at a neutral ground in order to avoid vendor lockins. TU/e is the most suitable place to host such a platform which can be beneficial for the university too.**	**Infrastructure administrator** **Infrastructure developer**	

Figure 4.2: Business Requirement No.01

It is important that the platform is initiated and hosted on a neutral ground from a business point of view. That is necessary in order to avoid potential issues from competing companies. In addition, the purpose of a university is to promote technology and innovation, which perfectly fits with the goal of the platform and provides a safer environment to attract more companies. It is equally important that TU/e will be the chosen university as it resides at the heart of Brainport and can really influence the industry. It is also one of the most prestigious technical universities in Europe which adds extra value. Furthermore, this system will create many opportunities for the university to open new research areas in cloud computing, big data and smart applications. It can potentially create a smart ecosystem by bringing together in one platform applications from different smart domains (smart health care, smart lighting, smart mobility, etc).

Requirement	Description	Rationale	Stakeholder	Use Case
BR02	**The system must support open development and APIs**	**It is necessary for the smart multivendor platform to support open APIs and standards. An open mentality is desired as an end result and therefore, the application programming interfaces should support open development.**	**Data service developer** **Data provider** **Data service provider**	**The latest VIBeX API (v1.5) will be one of the supported APIs.**

Figure 4.3: Business Requirement No.02

The support of open development is one of the core characteristics of the platform. Supporting open APIs and standards and ensuring the availability of Software Development Environments and Kits (SDEs and SDKs) for the development of applications and integrations of components are necessary requirements. In this way, the platform can really accelerate innovation by providing the opportunity of easy application development.

Requirement	Description	Rationale	Stakeholder	Use Case
BR03	The system must support multi-vendor functionality	The platform should be able to support components from different vendors, as well as providing the integration functionalities needed. New stakeholders should be able to integrate their own components and use the system to create a unique solution.	Data service developer Data service provider	

Figure 4.4: Business Requirement No.03

One of the most innovative requirements is the support of multi-vendor functionality. Different companies can use already developed components by different vendors and their products can also be used by others. It is important to mention, that the individual components or applications will be treated as black boxes and therefore, their business value can be maintained. Only the interfaces will be open.

Requirement	Description	Rationale	Stakeholder	Use Case
BR04	The system should host a marketplace of different components and applications	It is desirable that the platform serves as a marketplace of different components and applications that enable the developers to create unique prototypes. The different vendor components that are hosted in the marketplace are treated as black boxes. However, it is important to have defined interfaces (APIs) to facilitate fast integration of new components.	Infrastructure developer Infrastructure administrator Data service developer Data user Data provider	A developer uses an API from Company A to create an application that communicates with a vehicle from Company B

Figure 4.5: Business Requirement No.04

By satisfying the first three business requirements, it is possible to create a shared marketplace of components and applications. Similarly to the widely known Google Play Store, this marketplace will provide developed smart applications and components for developers or users of the platform. It could also be possible that the different vendors can sell their applications through the marketplace. This can create an environment where openness and ease of development play a critical role in creating new innovative concepts.

Requirement	Description	Rationale	Stakeholder	Use Case
BR05	The system must enable the seamless creation of showcases of smart mobility applications	The main goal of the platform is to enable applications to be developed and deployed easily. The system should be able to host several showcases of applications without time-consuming restrictions	Data user Data service developer Infrastructure developer	TU/e student competition/ integration with existing compan applications.

Figure 4.6: Business Requirement No.05

By enabling easy application development through providing a smart and open innovation platform, showcases can be created that demonstrate the added value of smart mobility applications. The wider acceptance of the platform can potentially lead to an increasing number of hosted applications and the system must still be able to provide the resources and integration functionality needed.

Requirement	Description	Rationale	Stakeholder	Use Case
BR06	**The system should support experimental and commercial development**	**On one hand, the goal is to open new opportunities for experimental applications for showcase. On the other hand, it should also be possible for a company to use the platform and create their own commercially-oriented prototype**	**Data user** **Data service developer**	

Figure 4.7: Business Requirement No.06

The final business requirement is related to the purpose of the platform. Hosted by a technical university that has strong bonds with the industry, the goal of the platform is to inspire innovation and at the same time, provide opportunities for companies to develop product oriented applications. In addition, this will create more collaborations between the university and the industry and can lead to the realization of new innovative applications.

4.3 System requirements

This section investigates the high level system requirements for the prototype of the smart mobility platform which are further categorized as functional or non-functional requirements. The complete list of system requirements can be found in the Appendix D.

4.3.1 Functional requirements

Requirement	Description	Rationale	Stakeholder	Use Case
FR01	**The system must provide comprehensive developer tools and environments**	**The smart multivendor platform should provide all the necessary developer tools and environments (SDKs, etc) to facilitate the rapid development of applications from developers of the platform. Similarly to website or smartphone application development, the effort needed to create a new component should be minimized.**	**Data service developer** **Infrastructure administrator**	

Figure 4.8: Functional Requirement No.01

It is highly critical that the platform provides all the developer tools and environments needed in order to create and deploy applications fast. The end goal of the platform is to minimize the effort and costs needed in order to develop new concepts.

Requirement	Description	Rationale	Stakeholder	Use Case
FR02	**The system must provide heterogeneous systems support and be hardware and cloud-provider agnostic**	**Not only should the smart mobility platform leverage the latest hardware, virtualization and software solutions, but it should also support a data center's existing infrastructure. While many of the early movers based their solutions on commodity and open source solutions, larger service providers and enterprises have requirements around both commodity and proprietary systems when building out their clouds. Furthermore, the solution should not be hardware dependent and it should be able to support all new technologies.**	**Data service developer** **Data service provider** **Infrastructure developer**	**Different vendors can use their existing systems to communicate with the platform**

Figure 4.9: Functional Requirement No.02

One of the main business requirements and simultaneously drivers behind the creation of such a platform, is the independence from software/hardware/cloud-provider specifications and support of heterogeneous systems. This is a key functional requirement towards development a platform without vendor lock-ins, especially at the infrastructure layer.

Requirement	Description	Rationale	Stakeholder	Use Case
FR03	**The system must provide service management and maintenance tools**	**It is important that administrators have a simple tool for defining and metering service offerings. A service offering is a quantified set of services and applications that end users can consume through the provider. The service management functionality should tie into the broader offering repository such that defined services can be quickly and easily deployed and managed by the end user. Finally, there should be clear guidelines regarding the maintenance of the toolbox.**	**Infrastructure administrator**	

Figure 4.10: Functional Requirement No.03

The platform administrators should have tools available for monitoring and maintaining the platform, which is an important requirement for open systems. The number of developers of applications can increase fast and therefore, the administrators will need to ensure the proper and uninterrupted functionality of the platform. In addition, service management tools that can be deployed easily by the end user are facilitating towards developing and deploying seamlessly smart applications.

4.3.2 Non-functional requirements

Finally, the non-functional requirements for the prototype platform are presented. Similarly, the complete list can be found in Appendix D.

Requirement	Description	Rationale	Stakeholder	Use Case
NFR01	The system must support seamless on-demand integration of components	Cloud applications should be built from the ground up to lower the cost, time, and risk of integrating them with existing on-premise and on-demand applications. As a customer you should expect your cloud provider to offer an integration platform and tools, a strong partner ecosystem, and generally whatever assistance you require for pain-free integrations.	Data service developer Data service provider Infrastructure developer	A company can use its existing modules to create a new application.

Figure 4.11: Non-functional Requirement No.01

One of the core added values of the system is the easy and seamless integration of existing components towards the creation of new applications. The different companies that desire to use their own components should be able to do so with limited effort in integrating them with the platform.

Requirement	Description	Rationale	Stakeholder	Use Case
NFR02	The system should provide high reliability (95%) and availability (95%) and basic security services	Whether the cloud serves as a test bed for developers prototyping new services and applications or it is running the latest version of an application, users expect it to be functioning every minute of every day.	Data user Infrastructure administrator Infrastructure developer	Applications should have constant connection with the platform, handling vehicle data reliably

Figure 4.12: Non-functional Requirement No.02

To be fully reliable and available, the cloud infrastructure needs to be able to continue to operate while data remains intact in the virtual data center regardless if a failure occurs in one or more components. Services need to be able to provide access to only authorized users and in this shared resource pool model the users need to be able to trust that their data and applications are secure.

Requirement	Description	Rationale	Stakeholder	Use Case
NFR03	The system must support fast deployment functionality (ideally deployment in a few days)	Since cloud applications don't require investments and installation of hardware and software, you should be able to get them running and productive in a fraction of the time compared with on-premise software.	Data service developer Infrastructure developer	A TU/e student can create a smart mobility application in a few days for his/her project.

Figure 4.13: Non-functional Requirement No.03

New applications might need new services or specifications (storage, speed, memory, CPU, etc). Even in this cases, the developer's effort should not be hindered and the deployment should still be seamless and fast. The goal is that a new applications can be developed in a matter of a few days regardless of the computing power demand.

Requirement	Description	Rationale	Stakeholder	Use Case
NFR04	**The system must be scalable and elastic**	**It is important for the platform to be scalable and elastic with respect to computational resources in order to enable fast growth and expansion of the capabilities of the platform.**	**Data service developer** **Infrastructure developer** **Infrastructure administrator**	**With the addtion of more applications, the hardware resources are scaled (CPU, memory, storage, etc) to meet the demands**

Figure 4.14: Non-functional Requirement No.04

Finally, scalability and elasticity are two necessary features of the platform that will ensure the uninterrupted functionality of the system and will guarantee that there will always be enough resources available. Problems encountered by increasing number of connected vehicles, developed applications and users or developers using the platform should be avoided and the system must be able to handle increased workloads and number of applications.

5 Prototype platform design choices

Before the setup and implementation of the prototype platform is explained, it is necessary to introduce important design choices. More specifically, a service which can provide hardware, software and cloud-provider independence is introduced (container technology) and additionally, data replication strategies are investigated towards a better reliability functionality.

5.1 Container service

One of the most important design choices for this platform is the container service. It provides the desired independence from hardware or cloud providers. Containers are an IaaS solution, as the access to a virtual machine is needed. On an important note, recently the cloud providers have started to support PaaS solutions with containers as well [33], since the PaaS models are more popular among their customers. However, they have not reached yet full PaaS container functionality.
The container technology is introduced by Docker and is supported by all major OS (macOS, Linux and Windows) and cloud providers (see section 6.3). Docker can be integrated into various infrastructure tools, including: Kybernetes, Apache Mesos, CoreOS rtk and DC/OS. More details about containers can be found in the following sections.

5.1.1 Background

Containers decouple applications from operating systems, which means that users can have a clean and minimal operating system and run everything else in some form of containers. Also, because a container offers a convenient unit to encapsulate a small application component, it becomes an infrastructure of choice for building micro-service applications, which enables more manageable application infrastructure and continuous application deliveries. Before the working principles of containers is explained, it is necessary to understand the concepts of virtualization and virtual machines first.

Virtual Machines

In computing, a virtual machine (VM) is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination [36].
Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, one or more apps, necessary binaries and libraries - taking up tens of GBs [8]. Multiple virtual machines can run simultaneously on the same physical computer. For servers, the multiple

operating systems run side-by-side with a piece of software called a hypervisor to manage them, while desktop computers typical employ one operating system to run the other operating systems within its program windows. Each virtual machine provides its own virtual hardware, including CPUs, memory, hard drives, network interfaces, and other devices. The virtual hardware is then mapped to the real hardware on the physical machine which saves costs by reducing the need for physical hardware systems along with the associated maintenance costs that go with it, plus reduces power and cooling demand.

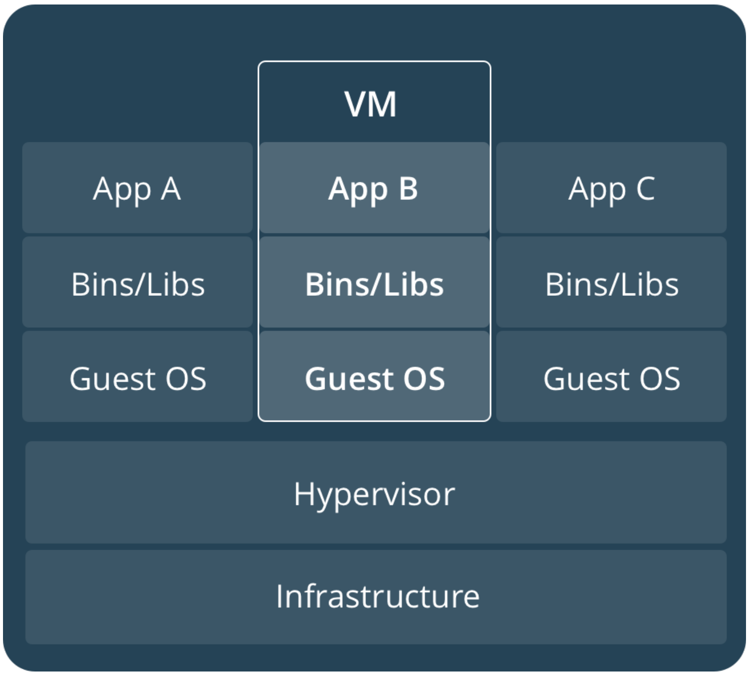

Figure 5.1: Architecture of a virtual machine [8]

Operating-system-level virtualization

Operating-system-level virtualization, also known as containerization, refers to an operating system feature in which the kernel allows the existence of multiple isolated user-space instances. A computer program running on an ordinary person's computer's operating system can see all resources (connected devices, files and folders, network shares, CPU power, quantifiable hardware capabilities) of that computer. However, programs running inside a container can only see the container's contents and devices assigned to the container [37].
On ordinary operating systems for personal computers, a computer program can see (even though it might not be able to access) all the system's resources. They include:

- Hardware capabilities that can be employed, such as the CPU and the network connection
- Data that can be read or written, such as files, folders and network shares
- Connected peripherals which can be interacted, such as webcam, printer, scanner, or fax

The operating system may be able to allow or deny access to such resources based on which program requests them and the user account in the context of which it runs. The operating system may also hide those resources, so that when the computer program enumerates them, they do not appear in the enumeration results. Nevertheless, from a programming point of view, the computer program has interacted with those resources and the operating system has managed an act of interaction.

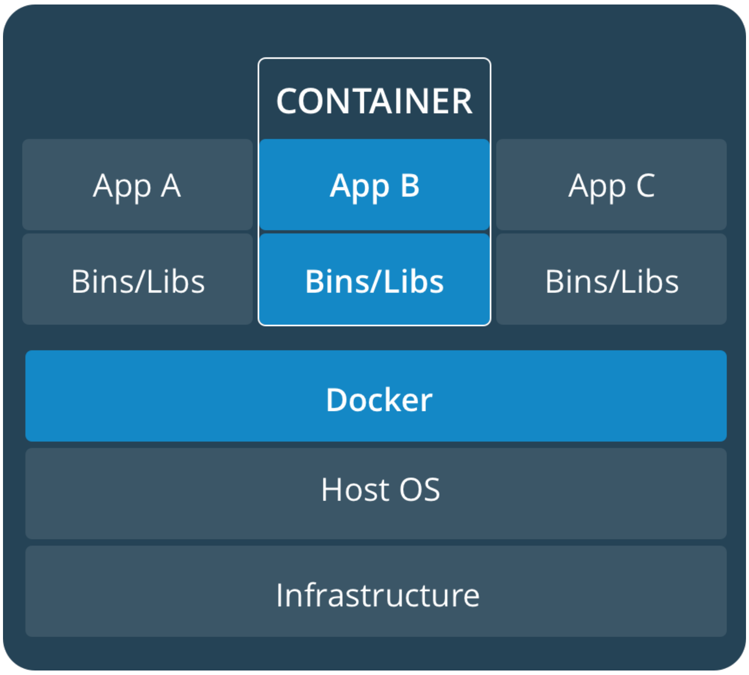

Figure 5.2: Architecture based on Docker container service [8]

With operating-system-virtualization, or containerization, it is possible to run programs within containers, to which only parts of these resources are allocated. A program expecting to see the whole computer, once run inside a container, can only see the allocated resources and believes them to be all that is available. Several containers can be created on each operating system, to each of which a subset of the computer's resources allocated. Each container may contain any number of computer programs. These programs may run concurrently or separately, even interact with each other.
Containers are an abstraction at the application layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), and start almost instantly. An overview of virtual machines and containers is shown in figure 5.3.

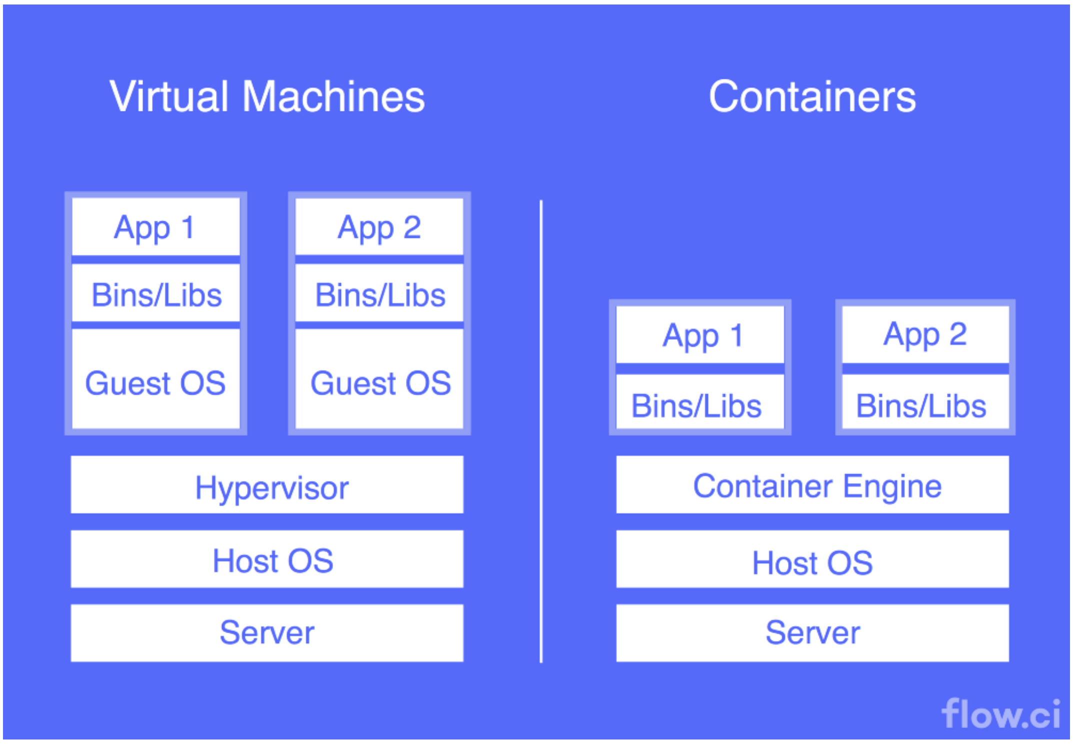

Figure 5.3: Comparison of Virtual Machines and Containers [9]

5.1.2 Consequences

The quick uptake of containers makes a lot of sense given what they offer. At a high level, containers provide a lightweight platform abstraction without using virtualization. Containers are also much more efficient for creating workload bundles that are transportable from cloud to cloud. In many cases, virtualization is too cumbersome for workload migration. Thus, containers provide a real foundation for moving workloads around hybrid clouds and multiclouds without having to alter much, if any, of the application.
More specifically, containers provide the following advantages [34]:

- Reduced complexity through container abstractions. Containers remove dependencies on the underlying infrastructure services, reducing the complexity of dealing with those platforms. They are truly small platforms that support an application or application services that sit inside of a very well-defined environment.
- The ability to use automation with containers to maximize their portability from one cloud to another
- Better security and governance from placing services around containers rather than inside
- Better distributed computing capabilities, because an application can be divided into many separate domains, all residing within containers. Containers can be run on any number of different cloud platforms, including those that provide the most cost and performance efficiencies. Therefore, applications can be distributed and optimized, based on their utilization of the platform from within the container. For example, an I/O-intensive portion of the application can

be placed on a public cloud service without virtualized infrastructure (bare metal cloud [35]) to provide the best performance, a compute-intensive portion of the application on a public cloud to provide the proper scaling and load balancing, and perhaps even place a portion of the application on traditional hardware and software. They can all work together to form the application, which has been separated into components that can be optimized.

- The ability to provide automation services that offer policy-based optimization and self-configuration, meaning providing an automation layer that finds the optimum place to run the container, as well as manages the changes in the configurations

On the other hands, there are still some disadvantages from using containers:

- One of the main disadvantages of container-based virtualisation compared to traditional virtual machines is security. Containers share the kernel, other components of the host operating system, and they have root access. This means that containers are less isolated from each other than virtual machines, and if there is a vulnerability in the kernel it can jeopardize the security of the other containers as well [9].
- The kind of tools needed to monitor and manage containers are still lacking in the industry. This is not a new phenomenon. The early days of hypervisor-based virtualization were marked by a shortage of suitable tools. And just as capable VM monitoring and management tools are now readily available, new tools are starting to appear for container management. These include Google's open source Docker management tools Kubernetes, DockerUI to replace Linux command line functions with a web-based front end, Logspout to route container logs to a central location and many more.
- In addition, managing a huge amount of containers is challenging, especially when it comes to clustering containers. Infrastructure tools that can facilitate container clustering are Kubernetes, Docker Swarm and Apache Mesos.

Containers provide a standard application architecture that offers both managed distribution and service orientation. In essence, containers can move from cloud to cloud and system to system, and thus can also provide automation for this process. In other words, we not only can leverage containers, but also can have them automatically 'live migrate' from cloud to cloud as needed to support the application's requirements.
Docker provides the fundamental building block necessary for distributed container deployments. By packaging application components in their own containers, horizontal scaling becomes a simple process of spinning up or shutting down multiple instances of each component.

5.1.3 Docker swarm

Docker Swarm provides native clustering functionality for Docker containers, which turns a group of Docker engines into a single, virtual Docker engine. The Docker Engine includes commands to manage swarm nodes (e.g., add or remove nodes), as well as deploy and orchestrate services across the swarm [38].
A node is an instance of the Docker engine participating in the swarm. One or more nodes can be run on a single physical computer or cloud server, but production swarm deployments typically include Docker nodes distributed across multiple physical and cloud machines. To deploy an application to

a swarm, a service definition to a manager node is submitted. The manager node dispatches units of work called tasks to worker nodes. Manager nodes also perform the orchestration and cluster management functions required to maintain the desired state of the swarm. Manager nodes elect a single leader to conduct orchestration tasks. Worker nodes receive and execute tasks dispatched from manager nodes. By default manager nodes also run services as worker nodes, but they can be configured to run manager tasks exclusively and be manager-only nodes. An agent runs on each worker node and reports on the tasks assigned to it. The worker node notifies the manager node of the current state of its assigned tasks so that the manager can maintain the desired state of each worker.
There are several features that make Docker Swarm useful [39]:

- Cluster management integrated with Docker Engine: Use of Docker Engine CLI to create a swarm of Docker Engines where application services can be deployed.
- Decentralized design: Instead of handling differentiation between node roles at deployment time, the Docker Engine handles any specialization at runtime. Nodes, managers and workers, can be deployed using the Docker Engine. This means it is possible to build an entire swarm from a single disk image.
- Scaling: For each service, the number of tasks to be executed can be declared. The swarm manager automatically adapts by adding or removing tasks to maintain the desired state during scaling procedures.
- Load balancing: The ports for services to an external load balancer can be exposed. Internally, the swarm allows the user to specify how to distribute service containers between nodes.
- Service discovery: Swarm manager nodes assign each service in the swarm a unique DNS name and load balances running containers. Every container running in the swarm can be queried through a DNS server embedded in the swarm.
- Multi-host networking: An overlay network for the services can be specified. The swarm manager automatically assigns addresses to the containers on the overlay network when it initializes or updates the application.

5.2 Data replication

When an application is deployed to more than one data center, such as cloud and on-premises locations, it is important to consider how to implement the replication and synchronization of the data that each instance of the application uses, in order to maximize availability and performance, ensure consistency, and minimize data transfer costs between locations.
The data in a Microsoft Azure storage account is always replicated to ensure durability and high availability. Replication copies the developer's data, either within the same data center, or to a second data center, depending on which replication option is chosen. Replication protects the data and preserves the application's up-time in the event of transient hardware failures. If the data is replicated to a second data center, it is protected from a catastrophic failure in the primary location. The replication strategy applies to the data contained in a storage account and therefore, different storage accounts is possible to have different replication strategies.
When a storage account is created, the following replication options are available [40]:

- Locally redundant storage (LRS): Replicates the data three times within a storage scale unit, which is hosted in a data center in the region in which the storage account is created.
- Zone-redundant storage (ZRS): Replicates the data asynchronously across data centers within one or two regions in addition to storing three replicas similar to LRS, thus providing higher durability than LRS. Data stored in ZRS is durable even if the primary data center is unavailable or unrecoverable.
- Geo-redundant storage (GRS): Replicates the data to a secondary region that is hundreds of miles away from the primary region. If the storage account has GRS enabled, then the data is durable even in the case of a complete regional outage or a disaster in which the primary region is not recoverable. For a storage account with GRS enabled, an update is first committed to the primary region, where it is replicated three times. Then the update is replicated asynchronously to the secondary region, where it is also replicated three times.
- Read-access geo-redundant storage (RA-GRS): Maximizes availability for the storage account, by providing read-only access to the data in the secondary location, in addition to the replication across two regions provided by GRS. When read-only access to the data in the secondary region is enabled, the data is available on a secondary endpoint, in addition to the primary endpoint for the storage account.

Replication strategy	LRS	ZRS	GRS	RA-GRS
Data is replicated across multiple datacenters.	No	Yes	Yes	Yes
Data can be read from a secondary location as well as the primary location.	No	No	No	Yes
Number of copies of data maintained on separate nodes.	3	3	6	6

Figure 5.4: Replication strategies

6 Prototype platform implementation

This chapter explains the approach and options towards implementing a prototype of the smart mobility platform enabling the development of a simple mobility application. This application is not required to be complex, as the focus is on the cloud infrastructure. Thus, a simple tracking web application of vehicles using the VIBeX API was conceived. The application requests the coordinates of the VIBe cars and displays the location on Google Maps.
The chapter starts by presenting a high level architecture of the cloud computing architectural layers and their basic components in the smart platform context, which provides a reference of required components. This reference was used to investigate possible implementation scenarios of the prototype platform and the application. Finally, the decision on which concept was used for the prototype platform is presented and the chapter concludes with the specifications of the prototype platform.

6.1 Reference

Through the high level architecture, it is possible to identify components and services that populate the different layers. However, these components are defined at a higher level and are independent from the chosen cloud deployment model (public, private or hybrid).
The cloud high level architecture can be divided in three layers (similarly to the cloud service models): Infrastructure, Middleware and Application layers. More specifically:

- **Infrastructure Layer:** The Infrastructure as a Service main components (storage, servers, virtualization, networking) are needed to host the applications and the different vendor components. This layer ensures that the cloud infrastructure needed to store data, establish (remote) connectivity with the servers and provide virtualization options (hypervisor, virtual machines, etc) exist.
- **Middleware Layer:** The Infrastructure layer will guarantee the access to data centers. However, due to the increased number of possible applications, it is certain that traffic management and load balancing functionalities are needed, as well as virtual networks and OS. These services ensure the efficient distribution of the workload and the automatic scalability of computational resources in case there is a sudden increase in the demand. This layer is also responsible for the messaging and communication protocols of the platform.
- **Application Layer:** Finally, this final layer is of most importance. Here, we want tools that enable easy integration of different vendor components and fast development/deployment of applications. It is important to note that the different vendor modules will be treated as black boxes, interfacing only through APIs. This creates the concept of a marketplace, where the

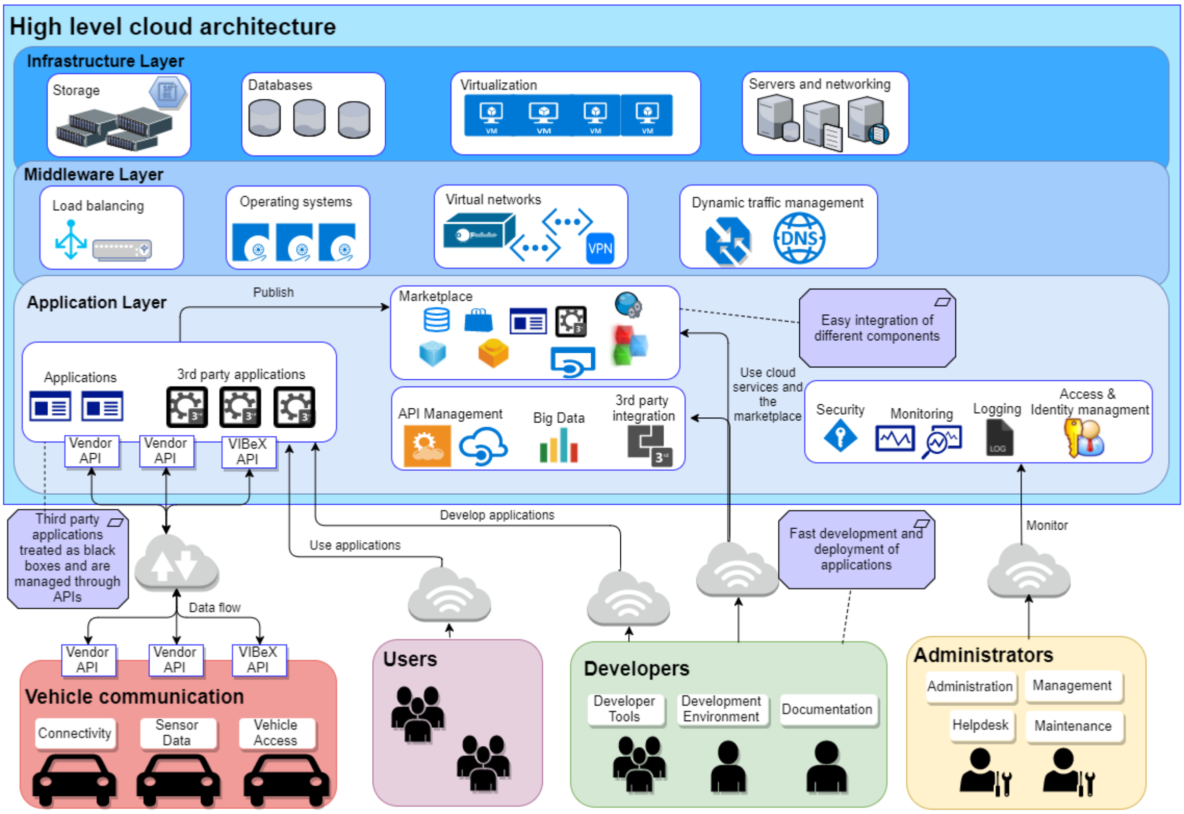

Figure 6.1: High level architecture as a reference. The arrows represent how the actors interface with the components.

different vendors can place their modules for other developers to use during development. In addition, monitoring, identity management, security and logging services are included in this layer. Finally, it is important to use API management services as the intention is to accommodate a variety of APIs for developers to use.

As shown in the picture above, there are four actors that interact with the cloud platform. These actors can be linked with the stakeholders described in section 4.1. Therefore, data providers are represented as vehicles, data users as users, data service developers and providers as developers and infrastructure administrators as administrators.

- **Vehicles:** The starting point for prototype platform is mobility applications This creates the need for vehicle communication and module integration to extract data from the vehicle and expose them to the platform via APIs. In addition, one of the predefined APIs will be the VIBeX API.
- **Developers:** This group should be provided with developer tools and environments that will enable them to create and deploy applications easily and efficiently. The different vendor components are available for the developers as black boxes with predefined inputs and outputs and can be used to create new applications.
- **Administrators:** Such a system will need administrator services to monitor the usage of the

platform but also to handle access rights and user groups.

- **Users:** Finally, towards enabling showcases, the integrated as well as newly developed applications should be available to use. These users can be fleet owners, public authorities, OEMs, research institutes, etc.

Typical examples of such applications that can be initially developed are applications in the areas of telematics, fleet management, driver behavior, electric vehicles, weather conditions, traffic management, navigation and location based applications.

6.2 Initiation

Since one of the goals was to have a working prototype of the platform and an application by the end of October, companies were approached in order to investigate possible partnerships. The goal was instead of creating the platform from zero ground to acquire components from different companies and integrate them to form the prototype of the smart mobility platform. In this way, the effort and cost needed to set the platform up would be minimized and at the same time, more companies would be invited to contribute. Most importantly, the application would be developed and deployed on the proposed approach, to provide an opportunity for continuation after the PDEng assignment using the same (or similar) setup.
The companies in consideration were: Beijer Automotive, V-tron Technologies, Ericsson and HERE Technologies. From these discussions, two separate concepts were created and investigated in parallel until only one of them was continued.

6.2.1 Multi-vendor concept

In this option, every company contributes towards the assembly of the platform. As shown in the figure below, HERE Technologies can contribute by supplying the cloud infrastructure through their Open Location Platform. This company has a very similar mindset as VIBe and therefore they can be a good fit. Additionally, Ericsson offered to be responsible for the communication between the vehicles and the platform (in other words, implementing the VIBeX API). Finally, V-tron and Beijer can provide applications and test cars in order to create showcases. It is also a possibility to use the fleet of electric cars from VIBe as test vehicles. As proposed, the concept will be under the umbrella of the university, which will be the administrator of the platform. The main challenge of this concept is the integration strategy, as many partners will be included from an early stage.
This option uses the existing infrastructure of specific vendors and requires mutual agreements, defined roles and clear responsibilities. The solution can be considered as a private cloud (can even evolve to community cloud, if a group of dedicated companies that has access to the platform exists).

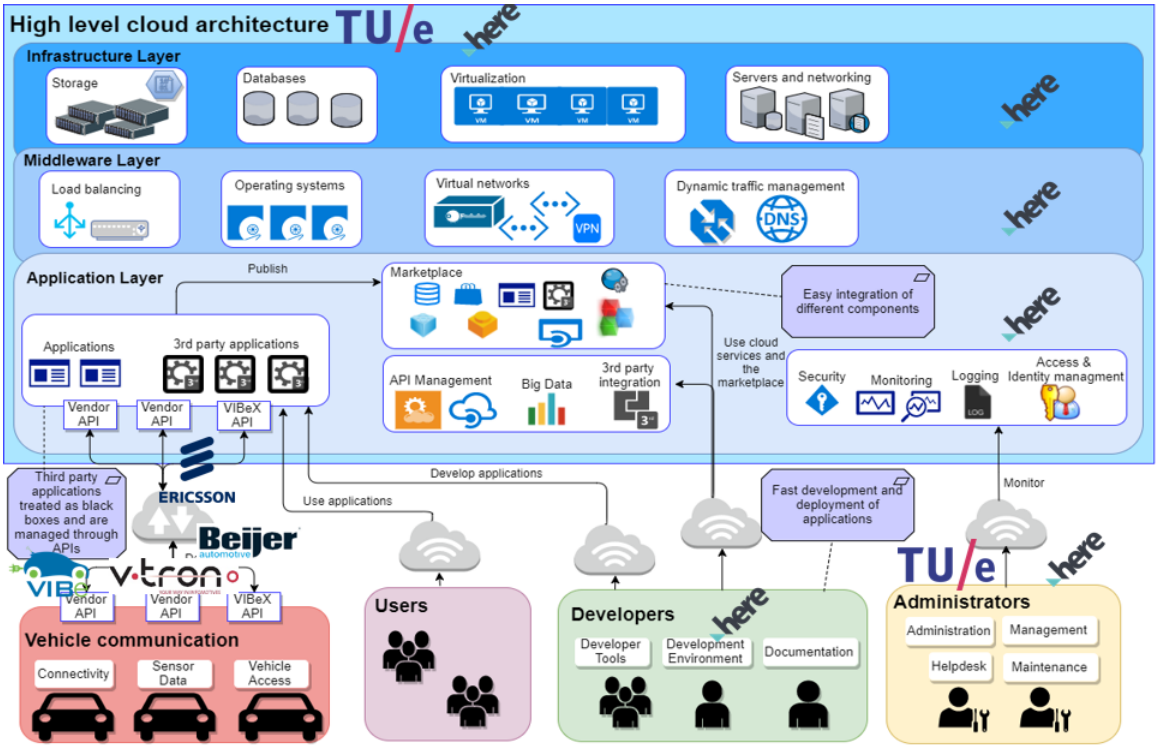

Figure 6.2: Multi-vendor concept high-level architecture

Advantages

- True multi-vendor platform from the beginning
- Less TU/e human resources needed
- Integration technology on the spotlight
- More attractive showcase

Disadvantages

- Clear roles and concrete arrangements are needed and fast
- Increased complexity due to many different components
- Potential vendor lock-ins
- Unclear feasibility during the project timeframe

6.2.2 Single cloud provider based concept

For the second concept, TU/e has an elevated role. In this case, fewer companies are involved initially and TU/e is additionally the owner of the platform. This involves investigation of a possible partnership with a cloud provider, who can facilitate in setting up the cloud environment. In addition, V-tron and Beijer can still play the role of the application and testing facilities provider, whereas HERE Technologies can provide an extra stream of location data to inspire location/navigation based smart applications. Similarly with the first concept, the VIBe electric car fleet can be used for showcases. However, the main challenge of this option is the elevated role of the university, which will have more responsibilities.

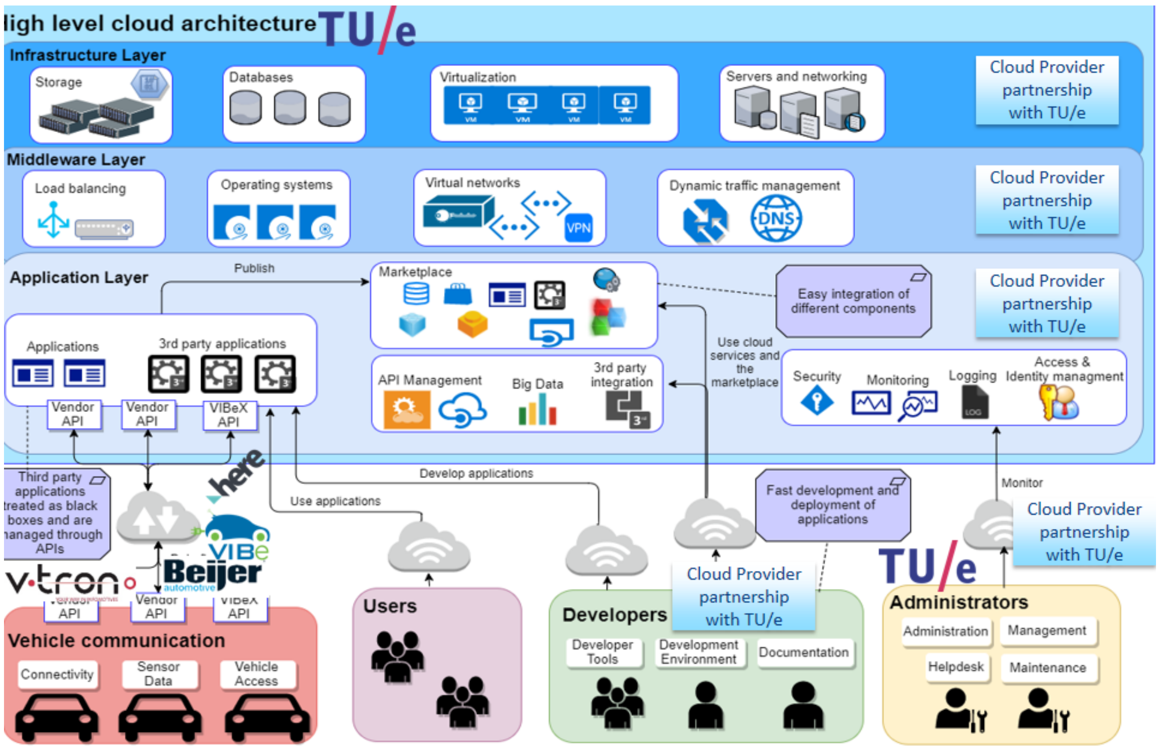

Figure 6.3: Single cloud provider based concept - high level architecture

Advantages

- TU/e in the lead (closer to the recommendation)
- Opportunity to build knowledge on creating smart systems
- Simpler integration
- More flexibility

Disadvantages

- More TU/e human resources and funding needed
- More basic showcase than the first concept
- Need to clarify ownership of the system
- Need for more changes internally in TU/e

6.2.3 Approved concept

Due to the time constraints, after two weeks of parallel investigation, it was agreed to implement a prototype for the platform with respect to the second concept . While the first concept seemed very promising, the companies did not react on time and therefore, the simpler but more feasible second concept was followed. Taking into consideration the chosen concept, the next section presents a market analysis to identify the optimum cloud provider to base the prototype platform and develop a simple application.

6.3 Cloud providers comparison

While there are dozens of different cloud providers, the investigation was focused on the four biggest [41]: Amazon Web Services, Microsoft Azure, IBM Bluemix and Google Compute Engine. There are several reasons for this decision. Firstly, in the case of public cloud services, the existence of a data center in West Europe is essential, as the performance in terms of connectivity speed and reliability is highly dependent on that aspect. In addition, these companies offer the widest variety of services and support, which makes them ideal for organizations new to cloud technologies. The usability of these services is highlighted by the user experience criterion. This aspect indicates the level of difficulty in the use of the services by developers, as well as the available on-line support community. Furthermore, these cloud providers offer the most reliability choices, guaranteeing more than 99% reliability of applications. Regarding the costs, the comparison considered only the public deployment model costs and therefore, the on-demand monthly subscription model (recurring and usage costs). The most important factors on determining the cost of a specific set of resources are the geographic location of the physical resources and the use of solid-state drives (SSDs) [42]. Finally, all the cloud providers in consideration support Docker container technology.
Another factor playing a role in the cloud provider decision, is the possibility to use the same cloud provider for future prototypes and possibly the base the smart mobility platform on the same provider. In this way, through this prototype, valuable conclusions about the chosen provider can be derived.

Amazon Web Services

Amazon is the global market leader in cloud services. They represent 31% of the market share in cloud technologies and they are increasing the gap with the second. Amazon Web Services (AWS), founded in 2006, offers the best variety of cloud services and every month new options are introduced. However, the increasing number of services adds more complexity to the design. In addition, Amazon

The Big 4	Amazon Web Services (AWS)	Microsoft Azure	IBM Bluemix and Softlayer Cloud	Google Compute Engine (GCE)
IaaS	Yes	Yes	Yes but more enterprise oriented	Not strong
PaaS	Yes	Yes	Yes	Yes
Global Market share	31%	9%	7%	4%
Locations	World wide	World wide	World wide	Mainly USA
Launched	2006	2010	2013	2013
User experience	++	+++	+	+
Services	+++	++	+	+
Resources	+++	+	+	++
Reliability	99.995%	99.95%	99.982%	99.95%
Cost	+	++	-	+
University connection	+	++	-	+

Figure 6.4: Cloud providers overview [10]

offers the most computational resources compared to its competitors. The price of using AWS is considered moderate, being neither the cheapest nor most expensive option in most cases [42]. Moreover, AWS is used by TU Delft, which gives an indication that there is affiliation with universities. Finally, AWS guarantees 99.995% reliability and has data centers worldwide.

Microsoft Azure

Microsoft introduced Azure services in 2010 and at the moment, it is the 2nd global cloud provider (9% of the market share). The main advantages of Microsoft Azure are the better user experience as it uses popular Microsoft Windows tools and the reduced cost compared to AWS in general, as well as to GCE when SSDs and data centers in Europe are needed. In addition, Microsoft is more university friendly having a precedence in collaborating with universities which could lead to a more extensive collaboration with TU/e. Azure provides university packages for educators, students or researches and is used by more than 750 universities. Finally, Microsoft Azure has a reliability percentage of 99.95% and has also worldwide presence when it comes to data centers.

IBM Bluemix

The third place in the global cloud market (7%) is held by IBM and IBM Bluemix cloud services, founded in 2013. The difference in IBM Bluemix is that it is more suitable for enterprise solutions. IBM is providing extensive tools and services for organization, promises reliability of 99.982% and has data centers worldwide. Bluemix does not offer the variety of services or resources that Amazon and Microsoft offer and the enterprise mentality (targeting the corporate rather than the consumer market) categorizes Bluemix as the most expensive option among the top four. Finally, IBM is not university friendly as it targets the corporations.

Google Compute Engine

Google introduced their cloud platform named Google Compute Engine (GCE) in 2013. Taking advantage of their pre-existing infrastructure, they are now at the 4th place in the global market, representing 4% and growing at a high rate. GCE is considered to be the cheapest option in the USA, offering a lot of computational resources but not many services yet. While the GCE reliability meter is at 99.5%, only one data center exists in Europe so far, which is an important drawback, as it can potentially introduce reliability and data speed issues due to limited resources, while at the same time increase the costs. Finally, GCE is used by several universities in the UK, which make use of the only GCE data center in Europe (also residing at UK).

Design choice

From the market analysis of the cloud providers, it seems that the best option for the prototype and eventually the platform would be to use Microsoft Azure. Despite being the 2nd biggest market player, Microsoft is known for working well with universities, forming attractive (also from financial point of view) partnerships. In addition, their vast variety of services ensure that the requirements will be met and the existing Microsoft developer tools are not only comprehensive but they are also widely known to developers already. It is also important to note that there is a vast online community working with Microsoft tools, which contributes to better support. Furthermore, using the public model of azure services, reliability and good performance is guaranteed as there are plenty of Azure data centers in Europe (currently, the closest one is in Amsterdam). Finally, one of the partners of VIBe (Beijer Automotive) has already experience with integrating Microsoft Azure services with the VIBeX API, which is valuable expertise and can be utilized. As an alternative in case that a cooperation with Microsoft is not feasible, Amazon Web Services can be used. Apart from being the market leader, we know that TU Delft is using AWS to educate students about cloud technologies, which means it is possible that TU Delft has already some collaboration with Amazon.

6.4 Recommended setup

Towards establishing the required initial environment for the prototype platform and the implementation of the application, several choices were made.

- **Cloud provider:** As mentioned in section 6.3, Microsoft Azure is the desired cloud services provider. Although the purpose is to create vendor and cloud agnostic applications, still there is need for infrastructure to accomplish that. Azure represents for the reasons mentioned in section 6.3 the most suitable choice for developing the prototype platform and the application.
- **Cloud deployment model:** From the analysis in chapter 2, it is clear that the most fitted model for the prototype platform is the public cloud model. It is desirable to have minimal initial effort and cost to setup the platform as one of the goals is to gain experience with cloud technologies. In addition, the cloud provider handles the maintenance and management of the platform, as well as scalability, elasticity and reliability of the services. Furthermore, with the subscription

payment model, the university is provided with the flexibility of stopping and re-enabling the services easily, without having any additional costs. The disadvantages of less flexible security options and dependence on an Internet connection are not that important in this case, as the purpose is not to create security critical and sensitive applications for the time being (e.g. personal, financial, corporate information), which means that the basic security that Microsoft provides is sufficient. In addition, the university has always a reliable and fast Internet connection. Finally, this model provided the only feasible solution to create a simple application, as there was no availability of a private cloud solution in the project context (section 6.2.3).

- **Cloud service model:** Ideally, the IaaS model is most suitable as it provides more flexibility and customizability in creating the desired development environment. In addition, the container technology is supported better for IaaS solutions currently. However, using an IaaS model requires human resources, time and effort to design properly. To assess the ease of development and deployment of a new application in Azure for the prototype platform, both IaaS and PaaS service models were implemented separately.

- **Container technology:** As mentioned above, Microsoft Azure is the desired cloud provider to initiate the platform. Docker is supported by Azure for the creation of single containers. However, Azure also supports several popular container orchestration technologies, namely, Docker Swarm, Kybernetes and DS/OS. In this way, the developer can choose which container service to use and build their application cluster in a hardware and cloud provider agnostic way. However, the application intended for the prototype platform requires a single container only and therefore, the aforementioned container orchestration technologies were not used. Finally, Microsoft provides detailed guidelines on how to work with containers and deploy applications in them.

- **Data replication:** As the data flow in the application is minimal and not safety critical, the chosen data replication is LRS for the prototype and the simple application. This choice includes less costs and is sufficient for demonstration purposes.

- **Vehicle integration:** A fleet of VIBe vehicles with integrated VIBeX API is the primary option for test cars regarding exposing real car data to the application.

These choices can be used to evaluate the feasibility of the system requirements of the prototype platform and provide a solid basis to gain experience from working with such platforms, as well as enable the creation of the simple application. In addition, it provides a flexible and agile option for a cloud platform with less initial costs and the ability to stop or continue the services at any given moment.

6.5 Specifications

This section explores the specifications of the prototype platform that will be tested for verification via the application implementation. The list for the complete platform can be found in Appendix D.

6.5.1 Functional specifications

These specifications ensure that the basic functionality of the platform will be present from the beginning via the prototype platform. Developer tools and environments, monitoring and access management services, as well as services that ensure hardware and cloud provider independence are all included, meaning that porting existing applications in the platform or developing new ones will be possible.

Specification	Requirement	Description
FS01	FR01:The system must provide comprehensive developer tools and environments	Microsoft PaaS provides all the needed developer tools and environments such as Visual Studio. In addition, more than 5 programming languages are supported.
FS02	FR02: The system must provide heterogeneous systems support and be hardware and cloud-provider agnostic	Using the Container service, any application or component developed will be hardware and cloud provider agnostic. At the same time, most of the popular OS are supported by Microsoft Azure.
FS03	FR03: The must provide service management and maintenance tools	Microsoft Azure services include numerous management and maintenance tools (Azure Management Portal, API Management, System Center, PowerShell Command-Lets).
FS04	FR04: The system should provide flexibility and access control mechanisms	Microsoft Azure services include access control mechanisms (Microsoft Azure Access Control Service, Active Directory, Azure Active Directory Authentication Extensions)
FS07	FR07: The system should handle regularly delivered, vendor managed updates	Multi-tenancy will be handled via Microsoft Azure scheduled updates and with guaranteed functionality during the downtime.

Figure 6.5: Functional Specifications

6.5.2 Non-functional specifications

The non-functional specifications ensure the required on-demand elasticity and scalability of the resources (depending on the demand), as well as the reliability and availability of the platform. All these functionalities will be handled automatically by Microsoft Azure public services ensuring the uninterrupted deployment of applications and the scaling of the resources without implications to the development environment.

Specification	Requirement	Description
NFS02	**NFR02: The system should provide high reliability (95%) and availability (95%) and basic security services**	**Microsoft guarantees 99.95% reliability and availability (through geo-replication services and Service Level Agreements or SLAs) and several options for security.**
NFS03	**NFR03: The system must support fast deployment functionality (ideally deployment in a few days)**	**Microsoft PaaS model incorporates tools and services for fast deployment of new applications and easy integration of existing ones.**
NFS04	**NFR04: The system must be scalable and elastic**	**Using Microsoft Azure's public services, it guarantees automatic scalability and elasticity.**
NFS05	**NFR05: The system should account for dynamic workload and resource management**	**Microsoft Azure public services in combination with specific PaaS models, can provide premade load balancers and dynamic workload functionality.**

Figure 6.6: Non-functional Specifications

6.5.3 Business specifications

Finally, the business specifications for the prototype platform are intended to demonstrate that the prototype platform is on the right direction. As mentioned in section, the purpose is to satisfy the requirements for open development and the creation of simple showcases of smart mobility applications. The simple application will serve as a showcase of the prototype platform via using the VIBeX API.

Specification	Requirement	Description
BS02	**BS02: The system must support open development and APIs**	**One of the first APIs will be the VIBeX API, which is open. Also, the developer tools from Microsoft are already free and open to use.**
BS05	**BR05: The system must enable the seamless creation of showcases of smart mobility applications**	**By enabling open development and use of existing components and developer tools, a lot of applications can be created quickly. For the prototype, a simple application will be developed as a showcase**

Figure 6.7: Business Specifications

7 Application implementation

This chapter investigates the implementation options of the prototype platform and the application. Several implementation scenarios were defined depending on the availability of human and data (particularly the VIBeX API) resources. Additionally, Microsoft Azure services were used to implement the application and thus, a choice of a SDE and SDK was made.
The initial goal was to create a prototype of the smart mobility platform with the help of Microsoft and implement a simple application using the VIBeX API. However, the time restrictions and the uncertainty of available human or data resources, did not guarantee the successful implementation of such a scenario. Therefore, two alternative scenarios were defined with a minimum requirement to have a simple and working application using simulated data and Microsoft Azure services. All scenarios are explained in the next section.
The chapter continues by presenting the requirements, specifications and structure of the developed application and concludes with two separate implemented cases regarding the use of Azure services (both the IaaS and PaaS models were implemented as mentioned in section 6.4).

7.1 Available human and data resources

After the two implementation concepts regarding the initiation of the prototype platform were explained in section 6.2 and the chosen concept was presented, three scenarios were identified as possible implementations of the application using that concept, depending on the availability of human and data resources.

- **Minimum scenario:** The most basic scenario could be implemented if there is unavailability of real vehicle data. In this case, the coordinates could be simulated and taken from a database. Only the required Microsoft Azure services to create a functional application would be used, as the whole application would be created without external assistance.

- **Desirable scenario:** In order to have real car data, V-tron technologies was approached to explore possible collaboration. V-tron has knowledge on how to communicate via the VIBeX API with a car and therefore, this option was promising. However, basic Microsoft Azure services would still be used as, similarly to the minimum scenario, there would be no external assistance regarding the use of Azure services.

- **Ideal scenario:** The best case scenario includes Microsoft, who could assist with connecting a car to the platform easily, as they have experience with the VIBeX API after their collaboration with Beijer Automotive in creating the Vetuda connected car platform [27]. Additionally, a more sophisticated and advanced Microsoft Azure environment could be designed in cooperation with

Microsoft Azure experts and include services like data analytics or application telemetry data. In this way, the opportunity is provided to create a more solid basis for possible steps regarding the development of the platform, as this option could be used to have a first instantiation of the platform. However, it also represented the least feasible option in the project timeframe.

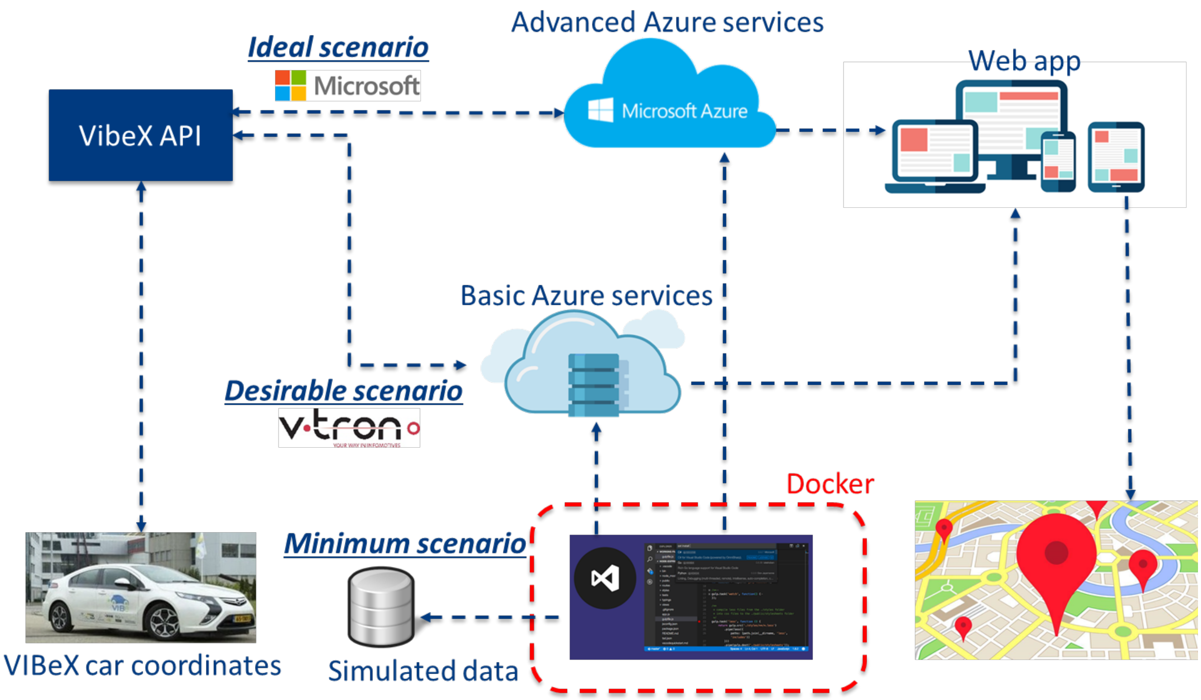

Figure 7.1: Implementation scenarios for the application. In all three cases, a docker application is developed. The arrows represent how the components of the prototype are linked in each scenario.

In all three scenarios, a modular and extendable approach will be followed. The application will be designed in a cloud provider and hardware agnostic way using container technology. It was chosen to implement the minimum scenario for the prototype platform and the application as it has the fewest dependencies with external required resources. The desirable scenario was not feasible to implement due to technical problems of V-tron's back-office and the VIBeX server. Finally, the ideal scenario was decided to be impossible to implement due to time restrictions in forming a partnership with Microsoft.

The goal is to demonstrate the ease of development and deployment, the hardware and platform independence, the presence of basic platform functionalities (networks, storage, communication), as well as the container service and data replication strategies and evaluate the development time and costs for creating a containerized cloud application in Azure.

7.2 Web application

As mentioned in the beginning of the chapter, a simple application was developed for the prototype platform. There was no requirement regarding the actual functionality of the application, as the purpose was to be used as a means to verify the prototype platform requirements. However, in this section, the requirements and specifications of the application are explained, as well as its structure.

7.2.1 Application requirements and specifications

The following table displays the requirements and corresponding specifications of the application, based on the chosen scenario mentioned in section 7.1.

Requirement	Specification
R01: The application must be related to smart mobility	**S01: The application functionality will be tracking a fleet of cars on Google Maps**
R02: The application should be related to VIBe	**S02: The fleet of cars will be VIBe vehicles from the available fleet of cars in VIBe inititative**
R03: The application should use the VIBeX API to expose vehicle data	**S03: The VIBeX API v1.5 is implemented in the VIBe cars and will be used. If this is not feasible, static vehicle data will be simulated via an SQL Server database.**
R04: The application must be compatible with the prototype platform	**S04: The application will be developed using Microsoft Azure SDKs as a cloud application. The supported browsers will be Google Chrome, Mozilla Firefox and Internet Explorer**
R05: The application should be simple to use	**S05: An one-page cloud application will be developed. Seven buttons will represent the VIBe cars and a Google Maps windows with moving ROI (Region Of Interest) will be used**

Figure 7.2: Requirements and specifications of the simple application

Summarizing, the application was not required to fulfill a specific functionality, as long as it is related to mobility (and ideally VIBe) and serves as a means for the prototype platform to verify its requirements (mainly vendor independence and ease of application development/deployment). The next section presents the technical specifications of the application (SDKs and frameworks used).

7.2.2 Technical specifications

The chosen tool for developing the application is Microsoft Visual Studio 2017 Community edition and moreover, the .NET Framework v4.6 was used. However, it should be noted that Azure provides several options for SDKs [43]. The reason to use .NET framework and Microsoft Visual Studio is mostly due to better Azure support and more integrated developer tools included. In addition, the Azure Portal was used to create the azure services and to host the application. The application was created as an ASP.NET Web Application and the Model-View-Controller architectural pattern [44] was used and the corresponding Microsoft Visual Studio template [45]. For storing the data, a SQL database was preferred. Finally, the application uses the AngularJS [46] framework (version 1.4.6), which makes the development of MVC applications easier [47]. This JavaScript-based, open-source and front-end web application framework aims to simplify both the development and testing of single-page applications by especially supporting MVC architectures.

7.2.3 Application structure

This section provides an insight on how the MVC framework was used in the application context. As mentioned in the introduction of Chapter 6, the tracking application displays the locations of VIBe cars in Google Maps.
The application displays the location of seven VIBe Mitsubishi Outlanders. The first six have single dedicated fixed coordinates and the seventh vehicle simulates a path. Every time the corresponding button is pressed, a query to the database is performed, the respective coordinates are retrieved, and finally displayed on the map.

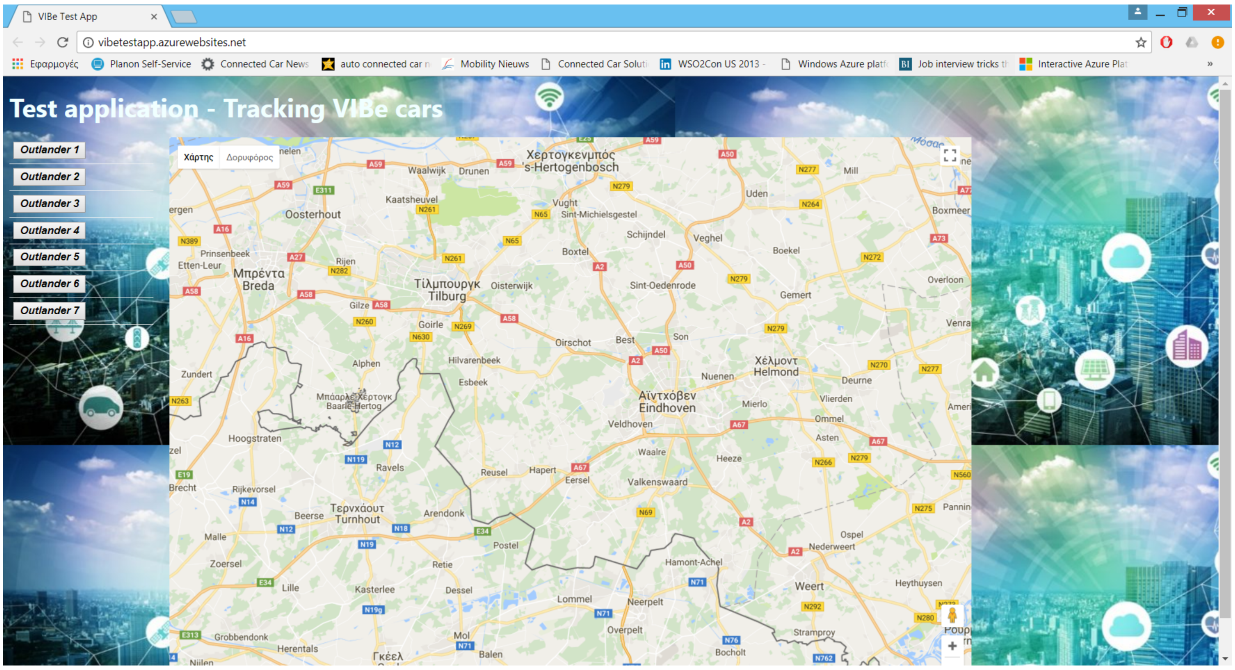

Figure 7.3: User interface of the application

Model

Since the minimum scenario was followed, simulated data from a database were used. This data are real vehicle data recorded from a past demonstration of VIBeX API in 2015. The creation of an MVC model was not necessary, as the application only needed to interface with a database for that scenario. However, in case of successfully implementing the desirable scenario with V-tron, an MVC model to communicate with the car will be created.
The database was created using the ADO.NET Entity Framework [48] and more specifically the Entity Data Model (EDM) [49], which is a set of concepts that describe the structure of data, regardless of its

stored form. This framework is useful when the data have a relational structure as data access, storage and scalability are very efficient.
For the creation of the database, SQL was used. More specifically, two tables were created, one for the first six vehicles and one for the seventh vehicle.

View

The MVC view file is responsible for the user interface (UI) of the application. The algorithm creates the initial UI by loading the Google Maps plug-in and the background image. Afterwards, requests the names of the vehicles and displays them as buttons upon successful response from the controller. Finally, after every new request for coordinates , the view displays on the map the corresponding vehicle and its location.
At the end of this file, the necessary JavaScript scripts are loaded. This scrips are responsible for loading the Google Maps using the AngularJS framework. Moreover, a Google API key is generated to grant access to the map.

Controller

The MVC controller file represents the interface between the view and the database. It is responsible for querying the database regarding the coordinates and data of each corresponding vehicle. Furthermore, updates the view with the responses of the database.

7.3 Implementation with Azure

The next step is publishing the application to Azure desired service model (PaaS or IaaS). It was decided to use both models as a means to evaluate the requirement of ease of development and deployment, as well as compare the costs. As mentioned in chapter 5, the container technology is a key design choice and the intention was for that to be also included. The SQL database server and SQL database were created to be the same for both options using the LRS data replication strategy as mentioned in section 5.2.

7.3.1 App service - PaaS Model

Initially, the PaaS model was implemented. Microsoft offers three PaaS services: App Service, Cloud Services and Service Fabric [50]. The choice of the suitable PaaS service was based on the criteria of fast application deployment, support of SQL databases, scaling of resources without redeploying, and associated cost. After evaluating the options, it was decided to use the App Service model, as it satisfied all the aforementioned criteria. The application was developed on a local machine and then deployed to the cloud. The Cloud Services option does not support seamless redeployment in case of resources scaling and configuration. However, this model provides limited access to the underlying infrastructure (while the App Service model does not). However, for our case, since the virtual machines will be used for the IaaS model and access to the infrastructure is included, the use of Cloud Services model did not add value and thus, the App Service PaaS model was preferred. As far as the Service Fabric option is concerned, it was rejected due to a considerable difference

in accumulative costs while providing much more functionality than needed, as Service Fabric is intended for computation demanding applications.
The infrastructure and middleware layers are completely handled by Azure and the developer has no way of accessing the computing resources. Furthermore, Azure determines the most computation efficient way to allocate the resources and takes care of load balancing, automatic scaling of resources, networking, operating systems and storage.

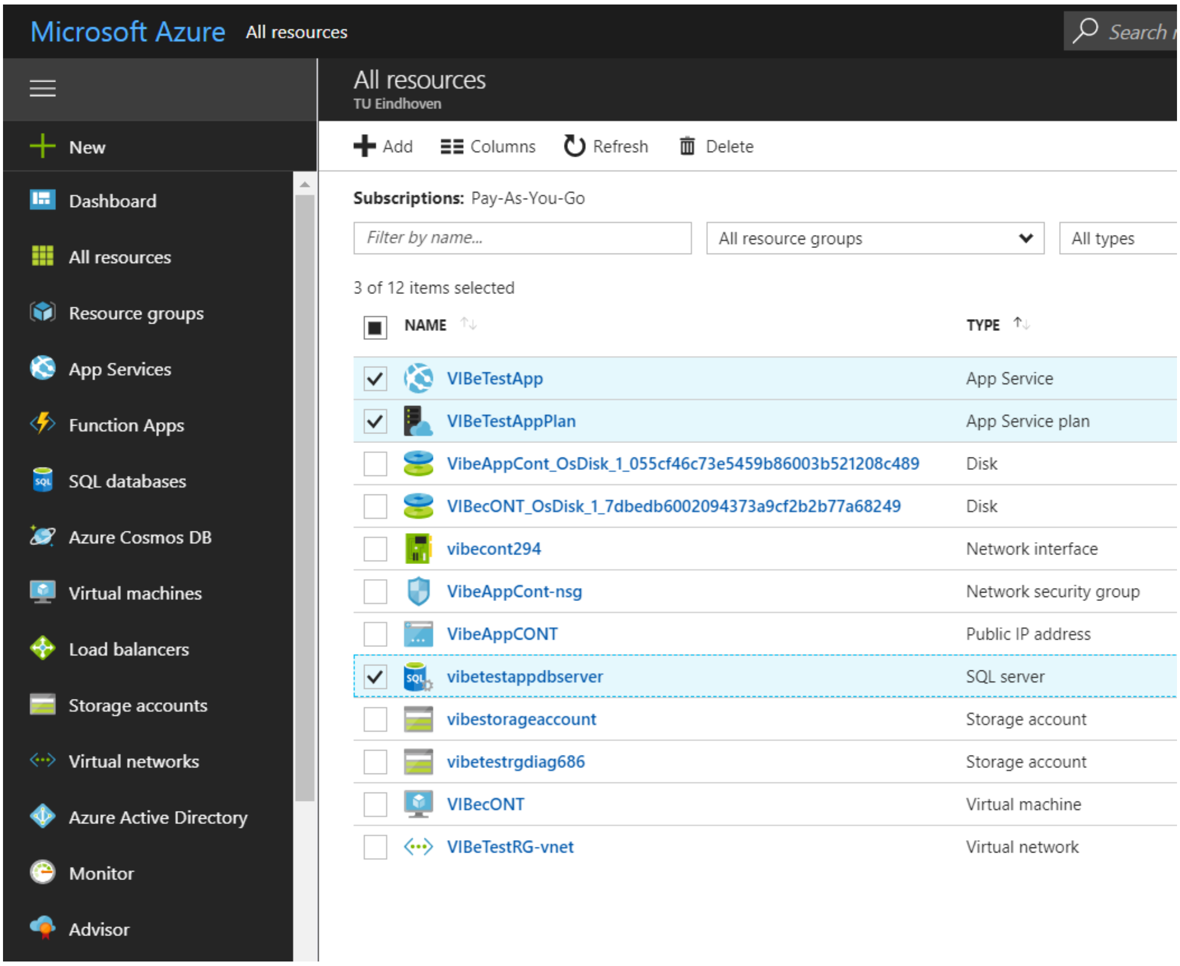

Figure 7.4: Azure services associated with App Service model

The figure above shows the services generated by Azure after publishing the application to the cloud. Only an App Service resource is created and a SQL server. While the App Service model provides extensive tools for monitoring the resource health and activity of the application, it is not possible to have access to the infrastructure. However, it is an almost instant way of publishing application to Azure.
Unfortunately, the support of the PaaS models regarding the container technology is immature yet. Microsoft only introduced migration of an App Service application to a specific version of Docker in April 2017 and only for Linux OS. Although it is still under development, the use of PaaS model is getting more popular than the IaaS models and therefore, Microsoft intends to fully support PaaS services and all OSs with containers. However, the container service was used only in the IaaS Model for this application, which is fully supported.

7.3.2 Virtual machine - IaaS Model

Using the IaaS service enables the developer to have an influence on the underlying infrastructure. Although many flavours of Linux OS are supported, it was chosen to host the application on a virtual machine running Windows Microsoft Server 2016, mostly for familiarity reasons. As mentioned in section 5.1, Docker is supported as well. However, to be able to use Docker in that virtual machine, effort and time was needed to install Docker and setup the development environment on that machine. Afterwards, the developed application was migrated to a Docker container and successfully deployed on the cloud.
It is clear that Azure only takes care of the infrastructure layer in this case (storage, networking, virtualization). The developer needs to take care of the virtual networks, load balancing features and traffic management. In addition, figure 7.5 shows the resources needed to operate a virtual machine. The developer has access to storage accounts, virtual networks and even the virtual machine itself. Firewalls, security groups and networks need to be configured by the developer as well.

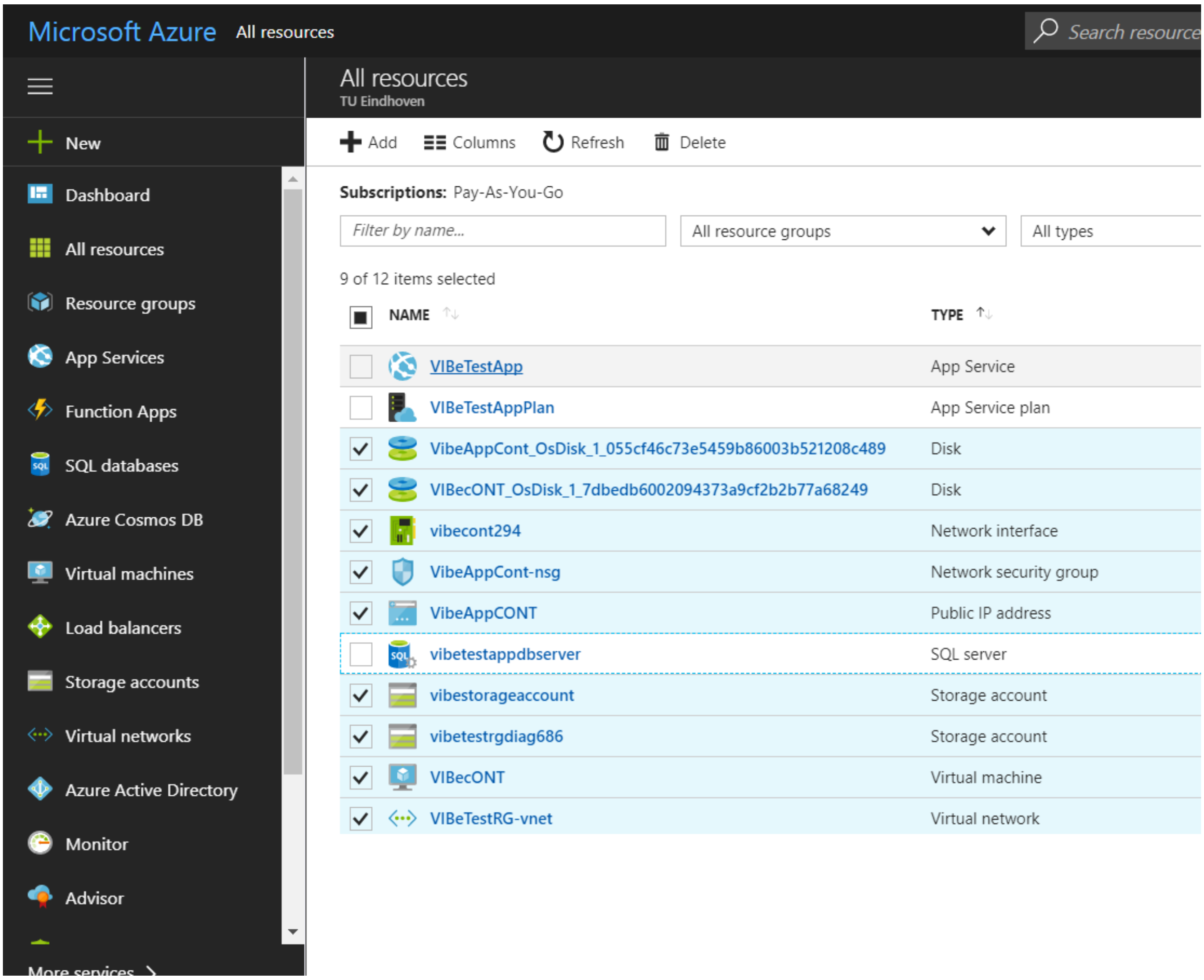

Figure 7.5: Azure services associated with Virtual Machines model

8 Prototype platform verification

In this chapter, the implementation results of the simple application are verified against the specifications (section 6.5) regarding the prototype platform. Initially, the container functionality was tested by migrating the developed application from Azure and Windows Server 2016 to Amazon Web Services and same OS (section 8.1.1). In this way, the cloud provider independence can be verified. Next, the docker container is migrated to a Linux Ubuntu 16.04 OS, to test the OS dependence in container technology (section 8.1.2). Afterwards, an evaluation of the implemented Microsoft Azure services is performed (section 8.2).

8.1 Container functionality

As mentioned in chapters 5, the container service is one of the important design choices that will provide independence of cloud provider and software/hardware specifications. The Docker technology is the main supported container service by the majority of the cloud providers. However, it was only on April 2017 that Azure officially supported Docker and provided relevant azure documentation and manuals. Therefore, it is still interesting to verify the level of integration of container technology in cloud services and the current capabilities regarding migrating to different operating systems or infrastructure.

8.1.1 Migration to AWS

Similarly with Azure services, a virtual machine was created in Amazon Web Services cloud platform and Windows Server 2016 was chosen as the OS. Afterwards, docker was installed on the virtual machine. The next step was to export the container application to AWS virtual machine. The container image was initially pushed to a Docker Hub repository from the Azure virtual machine.
Afterwards, it was pulled by the AWS virtual machine. The operation was completed successfully and the image was loaded to the AWS virtual machine. Both procedures are shown below.

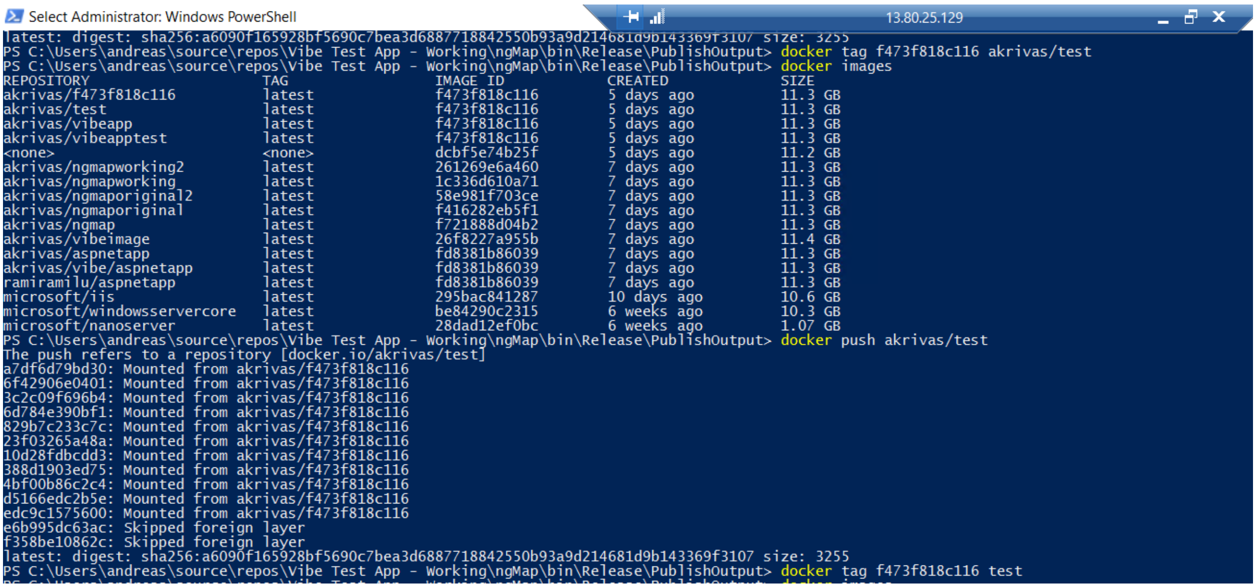

Figure 8.1: Pushing the image to a repository from Azure Windows Powershell

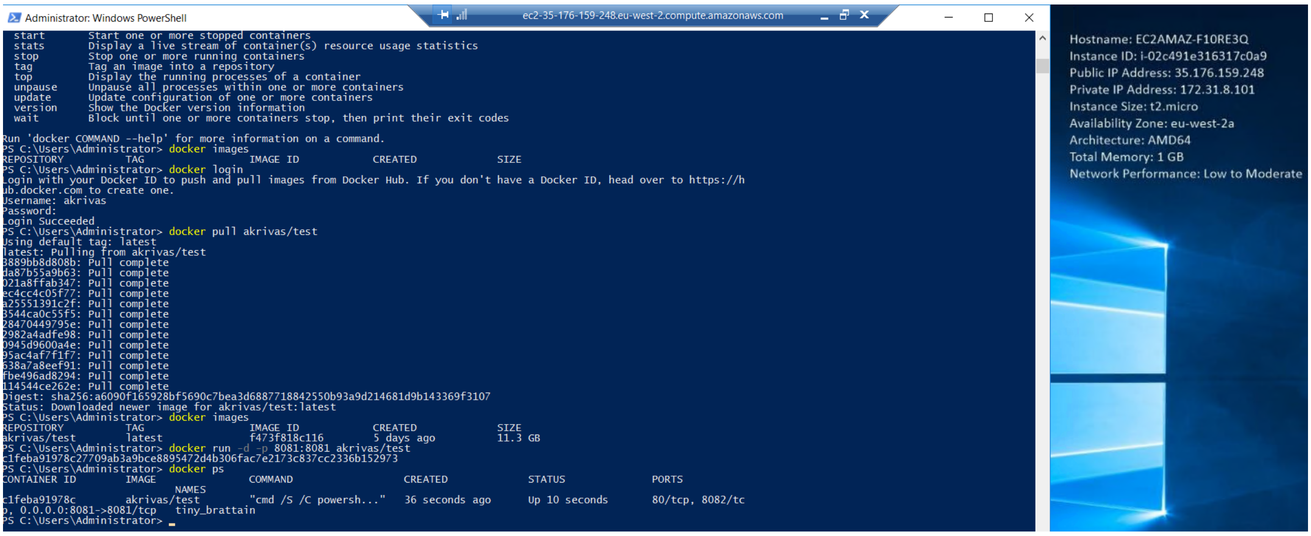

Figure 8.2: Successful image loading in AWS Windows Powershell

8.1.2 Migration to Linux

The next verification test is attempted migration to Linux Ubuntu OS. This test aims to test the migration functionality of a container from one OS to another, since all cloud providers support Windows and Linux OS. However, the test was unsuccessful and the image could not be pulled to Ubuntu. The reason is that containers are not meant for virtualization, and they are using the resources of the host machine. As a result, for now, windows containers cannot run directly on a Linux machine.
To be able to run a Windows container on Linux, a Windows virtual machine on top of the Linux OS is needed. In case of desired opposite operation (a Linux container on a Windows machine), at April 2017 a Docker toolkit was released (LinuxKit[51]). This toolkit provides enough features of a Linux-based platform at a layer beneath the application, for a container to run a Linux-based application on any operating system platform, (including Mac OS and Windows) and on any major cloud platform.

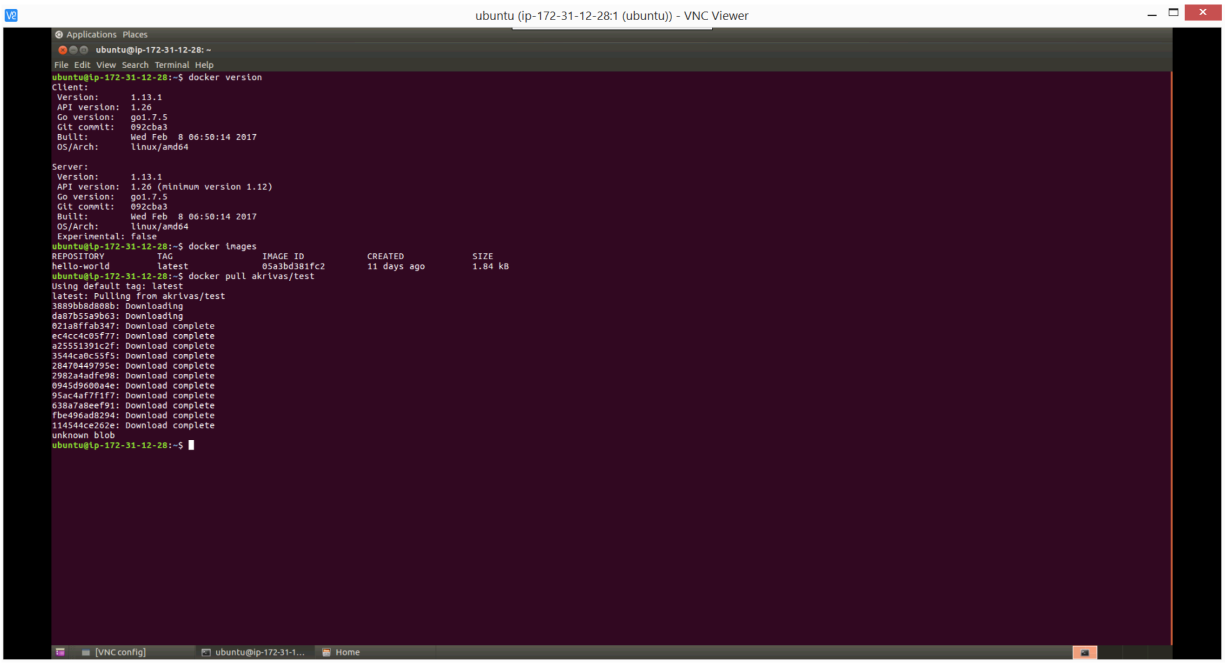

Figure 8.3: Unsuccessful image loading in AWS Ubuntu terminal

8.2 Evaluation

This section provides an analysis of the prototype platform and application implementation results and consequently, the feasibility of initiating such a smart mobility platform is explored. The application is evaluated against ease of development/deployment, costs, scalability/elasticity, maintainability and vendor independence.

8.2.1 Development effort and costs

The evaluation of the time and costs spent on creating the application will demonstrate if the requirements explained in chapter 4 are met. Four separate stages can be identified regarding the development: creating the web application, publishing to Azure using the App Service (PaaS) without Docker, publishing to Azure using the Virtual Machine (IaaS) with Docker and migrate the Docker container of the application to another cloud platform and operating system.

Web application

The development of the Web application considers the minimum scenario regarding the availability of resources. As described in chapter 7, the MVC framework was used and the application was developed using the ASP.NET Web Application template of Microsoft Visual Studio 2017 Community Edition. Effort was needed to reach the desired knowledge level regarding developing web applications, which had implications on the development time needed.
Regarding the costs, the actual development of the application before it was deployed to the cloud, is zero. All the tools and frameworks are provided by Microsoft for free.

Activity	Duration (Days)
Setting up the developer tools and environments	1
Application code	10
Total Duration:	11
Azure Resource	Cost (Euro)
Developer tools and environments	Initial: 0 Daily: 0
Total Cost	0

Figure 8.4: Web application development effort and costs

Concluding, the development of the application had zero cost and in total 2 weeks of development. Another 2 weeks were dedicated to acquiring the knowledge regarding building web applications based on SQL databases.

App Service

Using the PaaS model had the advantage of almost instantaneous deployment to cloud. Azure takes care of allocating the necessary resources to support the deployment of application and its continuous functionality. Therefore, the only actions needed by the developer is choosing the desired services and publishing the application.
The initial costs are zero, since the App Service is a public cloud service. However, the use of the required resources have a daily cost associated with them. For the specific application the cost is shown below.

Activity	Duration (Days)
Publish to Azure via App Service	0.5
Total Duration:	0.5
Azure Resource	Cost (Euro)
App Service plan	Initial: 0 Daily: 2
SQL Server Database	Initial: 0 Daily: 0.41
Total Cost	2.41 per day

Figure 8.5: App Service development effort and costs

As a reminder, the docker service was not used with this model, since the integration of docker with PaaS models is not fully operational yet. However, Microsoft indeed offers almost instant deployment of applications to the cloud with minimal effort to configure the infrastructure.

Virtual Machine

Creating a virtual machine in Azure gave the opportunity of directly having an influence to the underlying resources. Therefore, Docker needed to be installed and configured on the remote machine, as well as development environments and frameworks. However, Microsoft has detailed guidelines on these topics, and thus, setting up the environment was achieved in a day. Afterwards, the migration of the web application to docker and the virtual machine required almost another day. The chosen virtual machine specifications are 4GB RAM, dual core processor, 128GB hard disk drive and 64-bit OS. Similarly with the App Service, the virtual machine service is a public cloud service for Azure. Therefore, there are zero initial costs. However, a daily cost is associated with the use of the resource and is shown below.

Activity	Duration (Days)
Virtual Machine initiation	0.5
Installing and configuring Docker	0.5
Publishing the application to Azure as a Docker container	1
Total Duration:	2
Azure Resource	Cost (Euro)
Virtual Machine	Initial: 0 Daily: 2.45
Storage accounts	Initial: 0 Daily: 0
Disks	Initial: 0 Daily: 0.16
Network interface	Initial: 0 Daily: 0
Network security group	Initial: 0 Daily: 0
Virtual Network	Initial: 0 Daily: 0
Public IP Address	Initial: 0 Daily: 0.08
Docker EE for Windows Server 2016	Initial: 0 Daily: 0
SQL Server Database	Initial: 0 Daily: 0.41
Total Cost	3.1 per day

Figure 8.6: Virtual machine development effort and costs

Concluding, the effort needed to use the IaaS service is indeed higher, as well as the associated costs. However, the user has more flexibility to configure the resources, as well as fully functional container abilities. In addition, more applications can be hosted in the same virtual machine, which the same costs, where as in the PaaS model, every new application needs a new App Service plan.

Docker image migration

Migrating to another OS and cloud provider has also some costs and development effort associated. Firstly, the required environment in AWS needed to be set up. While the instantiation of a virtual machine in AWS is faster than in Azure, connecting remotely to a Linux OS is slightly more complex than connecting to a Windows Server OS. Since this option was used to verify the functionality of containers, the free subscription of AWS was used and therefore, no cost association was needed.

Activity	Duration (Days)
Virtual Machines initiation in AWS (Windows and Linux)	0.5
Installing and configuring Docker (Winodws and Linux)	0.5
Attempts to port the docker image to both OS	0.5
Total Duration:	1.5

Figure 8.7: Migration development effort

8.2.2 Microsoft Azure evaluation

After evaluating the two different service models (IaaS and PaaS), several conclusions were drawn regarding Microsoft Azure services.

- Almost 85% of the development time was dedicated to building knowledge on how to create web applications. Therefore, it is definitely possible to deploy applications fast with the current cloud service models. Still, effort is needed to help the developer create applications faster.
- The App Service option is very useful for fast and easy deployment of applications. Combined with the extensive tooling and environments, is indeed possible to have fast application development. There is no need to redeploy the application in cases of scaling or reconfiguring resources and several maintenance/monitoring tools are provided.
- The Virtual Machines service provides more choices and freedom in customizing the development environment, but is also associated with higher monthly costs and development effort. However, extensive support exists that enable even a single developer with minimum experience to create an application in weeks.
- Migrating the docker application to another cloud provider was demonstrated successfully. Thus, cloud vendor and infrastructure independence is possible. However, it was verified that OS dependence exists for the time being. Virtualization techniques are needed to run docker applications in different operating systems. For now, it seems better to create Linux containers, as it is possible with the use of LinuxKit to run them in other operating systems.
- Docker containers are fully supported by Azure IaaS services and are well documented. In PaaS services, Microsoft constantly increases support but containers are still not completely functional.
- In all investigated cases, maintenance and management of the infrastructure was taken care of by Microsoft, as well as elasticity and scalability. On one hand, in the App Service model, the user has no influence on how the resources are allocated and therefore, Microsoft scales them according to the demand. On the other hand, in the Virtual Machines case, the resources were manually scaled successfully with no implications to the development environment or the application, other than redeploying the docker container in the cloud.
- Regarding the costs, the App Service model charges only for the use of the needed resources, where as the Virtual Machines model charges for a user-defined set of resources. This can produce more costs than needed, since the resources used are not optimized according to the needs of the application. On the other hand, with increased number of applications, the virtual machines become more cost effective than the PaaS model.
- The public deployment model of Azure certainly fits the purpose of building knowledge regarding cloud technologies and is ideal for the initial setup of the smart mobility platform.

8.3 Specifications revisited

In this section, the results of the evaluation of the prototype platform are associated with the specifications and requirements defined in Chapter 6. From the specifications in section 6.5 only three were not verified (FS04, NFS05, and BS02). The use of open APIs (BS02) was not satisfied due to unavailability of real car data (section 7.1). As already mentioned, the vehicle data were taken from a static database. Regarding access control mechanisms (FS04), it was not necessary at the end to include these services. While it was possible, it did not add any valuable functionality at this specific application. Finally, dynamic workload (NFS05) was not needed with a static database and therefore, not evaluated.

Specification	Requirement	Description	Verification
FS01	FR01:The system must provide comprehensive developer tools and environments	Microsoft PaaS provides all the needed developer tools and environments such as Visual Studio. In addition, more than 5 programming languages are supported.	Microsoft Visual Studio and the frameworks used to develop the application are free, comprehensive and well documented.
FS02	FR02: The system must provide heterogeneous systems support and be hardware and cloud-provider agnostic	Using the Container service, any application or component developed will be hardware and cloud provider agnostic. At the same time, most of the popular OS are supported by Microsoft Azure.	A Docker container of the PoC was successfully implemented on an IaaS model and migrated on a different cloud provider. Still, OS dependence exists.
FS03	FR03: The must provide service management and maintenance tools	Microsoft Azure services include numerous management and maintenance tools (Azure Management Portal, API Management, System Center, PowerShell Command-Lets).	Maintenance of the infrastructure was handled by Microsoft in all test cases. Extra tools were not used.
FS07	FR07: The system should handle regularly delivered, vendor managed updates	Multi-tenancy will be handled via Microsoft Azure scheduled updates and with guaranteed functionality during the downtime.	The LRS data replication method guaranteed that updates will be installed seamlessly, an event that happened during the PoC development.
NFS02	NFR02: The system should provide high reliability (95%) and availability (95%) basic security services	Microsoft guarantees 99.95% reliability and availability (through geo-replication services and Service Level Agreements or SLAs) and several options for security.	During the 3 months that the Azure services were active, there was not a reliability related incident.
NFS03	NFR03: The system must support fast deployment functionality (ideally deployment in a few days)	Microsoft PaaS model incorporates tools and services for fast deployment of new applications and easy integration of existing ones.	The longest development cycle lasted 2 weeks. It is definitely possible to reduce the development time needed, once the Docker container is fully integrated with PaaS services.
NFS04	NFR04: The system must be scalable and elastic	Using Microsoft Azure's public services, it guarantees automatic scalability and elasticity.	Elasticity was not tested, since the application was not handling a large amount of data. Automatic scalability of resources without implications was successfully tested on the IaaS model. On the PaaS model, Microsoft handles the allocation of resources depending on the application demands
BS05	BR05: The system must enable the seamless creation of showcases of smart mobility applications	By enabling open development and use of existing components and developer tools, a lot of applications can be created quickly.	The simple application was successfully developed and deployed on Azure

Figure 8.8: Specifications revisited

9 Conclusions

This chapter presents the conclusions. Important findings and results, as well as recommendations regarding future steps, are discussed in detail. Initially, the conclusions regarding the prototype platform are presented. Afterwards, the recommendation to VIBe conclusions are explained.

9.1 Prototype of the smart mobility platform

This section presents the conclusions after the successful implementation of the application for the prototype platform. In section 1.3, the goals of the prototype platform were discussed. The intention was to evaluate the platform on four aspects: ease of development/deployment, vendor independence regarding the cloud infrastructure, open development and multi-vendor functionality. However, the attention was focused on ease of development/deployment and vendor independence, as open development (specifically the VIBeX API) and multi-vendor functionality were not included due to unavailability of resources (see below and chapter 6). As far as the achievements with respect to the prototype of the smart mobility platform are concerned:

- The requirements for the smart mobility platform were derived, focusing on ease of development/deployment, multi-vendor functionality, open development and hardware/software/cloud-provider independence (sections 4.2, 4.3 and Appendix D)

- A cloud-provider analysis was conducted regarding the cloud infrastructure for the prototype platform and the Microsoft Azure services were chosen (section 6.3)

- Design choices regarding cloud computing deployment and service models, as well as associated services (i.e. containers) were investigated (chapter 5, sections 6.4, 8.2.2)

- A working prototype of the smart mobility platform and a simple application were successfully implemented, focusing on the utilization of cloud infrastructure, ease of development/deployment and cloud/hardware independence (chapter 7)

- Cloud-provider and infrastructure independence was demonstrated via the successful migration of the application from Microsoft Azure infrastructure to Amazon Web Services using Docker container technology (section 8.1.1)

- A cost and development time evaluation was conducted regarding the associated costs and time needed to create and maintain a simple mobility application on the prototype platform (section 8.2.1)

On the other hand, due to unavailability of resources and time restrictions, several aspects were not explored. More specifically:

- Open development was not included in the prototype platform as the application did not make use of the VIBeX API, since there were technical issues involved with the VIBeX server and there was no connectivity with the VIBe car fleet. This issue could not be resolved during the available time.
- The multi-vendor functionality was not investigated as there was no possibility to integrate an existing, vendor-specific application or software component in the prototype platform
- Operating-system independence was not successful as currently Docker containers provide limited support for that functionality especially in the case of migrating a Windows Docker container to Linuxs

From the above results, the prototype platform and the implemented application successfully demonstrated some of the potential benefits of a smart mobility platform. On the one side, scalability, maintenance, elasticity and allocation of resources can be handled by the cloud provider using public services at affordable costs. In addition, comprehensive development tools and environments are provided for free (including several tools aiming to facilitate the developer), as well as an extensive on-line support community. For these reasons, Microsoft Azure proved to be a user-friendly option for an initial setup of the platform. Moreover, docker was successfully used and proved to be functional in IaaS Azure services. Thus, cloud-provider independence and ease of deployment are demonstrated. On the other hand, container technology is currently in the process of being integrated with existing cloud PaaS service models, and therefore, it is not completely functional yet. However, all the major cloud providers are projected to provide full integration of container technology. For now, IaaS models can be used for container technologies, although it requires more setup and development effort. In addition, effort is needed to create tools that enable the development of applications even faster. More details on possible steps for technical future work are presented in section 9.3.1.

9.2 VIBe recommendation

Regarding the recommendation to VIBe, a detailed analysis of the region was conducted that resulted in defining strategic objectives and tactical plans towards a proposed vision of regarding smart technologies. The VIBe recommendation will be a possible way forward for VIBe and the region in order to prepare for the smart transformation and exploit the new business opportunities that arise. A clear road map is defined regarding the steps needed to achieve this purpose. In this recommendation:

- An analysis of the reasons behind the smart transformation of cities around the world was done (Appendix C)
- An analysis of the global and regional developments was conducted in the context of smart mobility technology (Chapter 2)
- Two separate SWOT analysis processes were conducted, one for VIBe and another for the region of Eindhoven (section 3.1 and Appendix C)

- A proposed vision regarding the future, based on the findings of the above investigations was presented (section 3.2 and Appendix C)
- Proposed strategic steps and objectives for VIBe were documented (section 3.3 and Appendix C)

The recommendation was not presented to the VIBe Steering Group committee, due to unavailability of the VIBe members. However, it was published via email along with an executive summary.

9.3 Future work and recommendations

This section provides recommendations for future work regarding the initiation and implementation of the smart mobility platform.

9.3.1 Technical work

Firstly, the more technical next steps are identified.

- **Docker integration with PaaS:** Investigation of migrating an App Service application (or any PaaS service based application) to a Docker container in Linux [33], since this case is supported currently by Microsoft (there is still no PaaS support of a Windows container). Combining the PaaS functionality regarding ease of deployment with the hardware and cloud provider independence provided by a container technology such as Docker, can provide better optimization and automatic scalability of resources needed for containers, as well as minimizes the development environment setup effort, while simultaneously isolating the application from any dependence.
- **Application using the LinuxKit:** The use of LinuxKit can provide insight on how a Linux-based docker application can be not only cloud provider and hardware agnostic, but also OS independent.
- **Implementing the VIBeX API:** Real car data using the open API of VIBe in combination with a docker application, can provide a better showcase of the smart mobility platform, as the requirement of using open standards will be satisfied.
- **Migrating a vendor application to Azure:** Towards investigating the multi-vendor functionality, an existing, vendor-specific mobility application can be migrated to a docker container and eventually deployed through Azure.
- **Continuous Delivery, Continuous Integration and Continuous Deployment tools:** Towards facilitating the developer, continuous delivery, integration and deployment tools can be investigated (Microsoft Visual Studio already provides support for this kind of tools). Continuous integration focuses on integrating work from individual developers into a main repository multiple times a day to catch integration bugs early and accelerate collaborative development. Continuous delivery is concerned with reducing friction in the deployment or release process, automating the steps required to deploy a build so that code can be released safely at any time. Continuous deployment takes this one step further by automatically deploying each time a code change is made [52]. In this way, the developer's effort can be minimized and the development process can be automated.

- **API Management service:** API management is the process of creating and publishing web APIs, enforcing their usage policies, controlling access, nurturing the subscriber community, collecting and analyzing usage statistics, and reporting on performance. In case of multiple APIs, this service can simplify the developer's effort.
- **Investigation of Docker Swarm:** Orchestration of containers can be explored by using multiple containers in the case of multiple running applications. In this way, management of containers and maintenance can be investigated (section 5.1.3).

9.3.2 Smart Mobility Platform initiation

The recommendation to VIBe is accompanied by the proposal of a cloud platform for smart mobility applications, hosted in the TU/e context.
One of the strong supporters of the effort is NXP Semiconductors, as they have committed in facilitating further even after the completion of the PDEng assignment. Similarly, V-tron technologies are considering the option of developing a new application for one of their clients on an initial setup of the smart mobility platform.
The interest of the public authorities is also quite high. Gemeente Eindhoven is considering using the platform for smart city applications and smart mobility showcases. However, there have not been any concrete arrangements with the Gemeente yet. In addition, the Organization of Dutch Municipalities (Vereniging van Nederlandse Gemeenten) is fully supportive of the project.
Thus, several options can be interesting regarding an initial setup of the platform and are proposed below:

- TU/e can assume the responsibility of creating an initial setup of the Smart Mobility Platform with the help of Microsoft. After two initial meetings, Microsoft is very interested in helping TU/e in developing knowledge regarding cloud computing and specifically Azure services. The requirement is that the project is officially part of a department of the university.
- The members and partners of VIBe are eager to help with more projects and proof-of-concept applications in the smart mobility platform (V-tron application for example)
- NXP has already agreed on financing at least one more PDEng trainee to further work on the platform. In such a case, more prototypes can be created to get a better understanding of the platform
- As mentioned above, the external interest is high which can create more opportunities for projects and applications using an initial setup of the platform. However, concrete arrangements were not discussed during the available project time.
- The Department of Mathematics and Computer Science has shown interest in being responsible for the platform
- Thorough investigation regarding the required costs and human resources for a first instantiation of the platform is needed

Bibliography

[1] How To Implement an IoT-Focused Marketing Strategy. https://kitewheel.com/iot-internet-things-customer-engagement-blog/. Accessed August 2017.

[2] Dubai Wants to Become a Global Benchmark for Smart Cities. http://www.channelpostmea.com/2017/03/02/dubai-wants-to-become-a-global-benchmark-for-smart-cities/. Accessed August 2017.

[3] Bosch is seamlessly connecting mobility. https://www.bosch.com/products-and-services/connected-products-and-services/connected-mobility/. Accessed September 2017.

[4] Connected cars could cost consumers' privacy. http://www.businessinsider.com/connected-cars-could-cost-consumers-privacy-2015-10?international=true&r=US&IR=T. Accessed January 2017.

[5] Preethi Ramamurthy. An IoT architecture for cloud connected electric vehicles. http://repository.tue.nl/799344, September 2015. PDEng-Thesis.

[6] The Lessons of Cloud Computing - What Have We Learned So Far? http://www.xorlogics.com/2017/07/31/the-lessons-of-cloudcomputing-what-have-we-learned-so-far/. Accessed August 2017.

[7] Monitoring Cloud Infrastructure Performance to Eliminate Visibility Gaps. https://www.sevone.com/white-paper/monitoring-cloud-infrastructure-performance-eliminate-visibility-gaps. Accessed June 2017.

[8] What is a Container. https://www.docker.com/what-container. Accessed August 2017.

[9] Introduction to Containers: Concept, Pros and Cons, Orchestration, Docker, and Other Alternatives. https://medium.com/flow-ci/introduction-to-containers-concept-pros-and-cons-orchestration-docker-and-
Accessed September 2017.

[10] Cloud Providers Comparison. https://www.cloudorado.com/cloud_providers_comparison.jsp. Accessed June 2017.

[11] Cloud for automotive. https://www-935.ibm.com/services/multimedia/Cloud_for_Automotive_Exec_Summary.pdf. Accessed September 2017.

[12] How Cloud Computing Is Changing The Automotive Industry. https://www.knowarth.com/how-cloud-computing-is-changing-the-automotive-industry/. Accessed September 2017.

[13] i SCOOP. Industry 4.0: the fourth industrial revolution - guide to Industrie 4.0. https://www.i-scoop.eu/industry-4-0/. Accessed September 2017.

[14] Matthew Murray. Moving Toward a World of 50 Billion Connected Devices. https://www.pcmag.com/news/347086/moving-toward-a-world-of-50-billion-connected-devices. Accessed March 2017.

[15] Rob van der Meulen. Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. http://www.gartner.com/newsroom/id/3598917. Accessed March 2017.

[16] What is Smart Technology. https://www.igi-global.com/dictionary/smart-technology/38186. Accessed September 2017.

[17] Light your home smarter. http://www2.meethue.com/en-us. Accessed September 2017.

[18] Roomba 980. http://www.irobot.com/For-the-Home/support/FAQs/roomba-980. Accessed September 2017.

[19] Connected Car Definition. http://www.autoconnectedcar.com/definition-of-connected-car-what-is-the-connected-car-defined/. Accessed September 2017.

[20] Connected cars: top 5 innovations. https://www.raconteur.net/technology/connected-cars-top-5-innovations. Accessed February 2017.

[21] Organizing Connected Transportation. https://venturescannerinsights.wordpress.com/2014/05/06/organizing-connected-transportation/. Accessed January 2017.

[22] Connected Car Market Update. https://venturescannerinsights.wordpress.com/2015/03/20/connected-car-market-update/. Accessed January 2017.

[23] Venture Scanner Connected Transportation Report 2016 Q1. https://www.slideshare.net/NathanPacer/venture-scanner-connected-transportation-report-2016-q1. Accessed January 2017.

[24] Transportation Technology Exits by Category and by Year - Q3 2017. https://www.venturescanner.com/blog/tags/connected%20transportation. Accessed January 2017.

[25] PWC. Connected car report 2016. Technical report, PWC, 2016.

[26] Smart Mobility - SmartTogether. https://www.tno.nl/media/7613/magazine-smart-mobility.pdf. Accessed March 2017.

[27] Vetuda Connected Car Platform. https://www.vetuda.com/. Accessed June 2017.

[28] Ralph Retter Walter Schupeck Christoph Fehling, Frank Leymann and Peter Arbitter. *Cloud Computing Patterns*. Springer, 2014.

[29] Thomas Erl, Zaigham Mahmood, and Ricardo Puttini. *Cloud Computing: Concepts, Technology, and Architecture*. Pearson, 1st edition, 2013.

[30] Eamonn Colman. When to use SaaS, PaaS and IaaS. https://www.computenext.com/blog/when-to-use-saas-paas-and-iaas/. Accessed August 2017.

[31] Business Requirements vs Functional Requirements. http://www.requirementsnetwork.com/business-functional.htm. Accessed May 2017.

[32] Jeroen Redegeld, Pieter J.L. Cuijpers, Johan J.Lukkien. Stakeholder impact on the software architecture of intelligent transport systems implementations. http://ieeexplore.ieee.org/document/7958429/. 2017 IEEE International Conference on Software Architecture Workshops (ICSAW).

[33] Dockerization of Azure PaaS. https://blog.appliedis.com/2017/01/04/dockerization-of-azure-paas-beyond-azure-container-service/. Accessed September 2017.

[34] David Linthicum. Moving to Autonomous and Self-Migrating Containers for Cloud Applications. https://www.cloudtp.com/doppler/moving-to-autonomous-and-self-migrating-containers-for-cloud-applications/. Accessed August 2017.

[35] What is bare-metal cloud? http://www.computerweekly.com/blog/CW-Developer-Network/What-is-bare-metal-cloud. Accessed September 2017.

[36] Virtual machine. https://en.wikipedia.org/wiki/Virtual_machine. Accessed September 2017.

[37] Operating-system-level virtualization. https://en.wikipedia.org/wiki/Operating-system-level_virtualization. Accessed September 2017.

[38] Swarm mode key concepts. https://docs.docker.com/engine/swarm/key-concepts/. Accessed September 2017.

[39] Swarm mode overview. https://docs.docker.com/engine/swarm/#feature-highlights/. Accessed September 2017.

[40] Marsh Macy and Tom Pratt. Azure Storage replication. https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy. Accessed August 2017.

[41] Cynthia Harvey. Public Cloud Computing Providers. http://www.datamation.com/cloud-computing/public-cloud-providers.html. Accessed June 2017.

[42] How to compare cloud costs between Amazon, Microsoft and Google. https://www.networkworld.com/article/3145470/cloud-computing/how-to-compare-cloud-costs-between-amazon-microsoft-and-google.html. Accessed June 2017.

[43] Microsoft Azure SDKs. https://azure.microsoft.com/en-us/downloads/. Accessed July 2017.

[44] A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. https://web.archive.org/web/20100921030808/http://www.itu.dk/courses/VOP/E2005/VOP2005E/8_mvc_krasner_and_pope.pdf. Accessed July 2017.

[45] Learn About ASP.NET MVC. https://www.asp.net/mvc. Accessed July 2017.

[46] AngularJS. https://en.wikipedia.org/wiki/AngularJS. Accessed July 2017.

[47] 10 Reasons Why You Should Use AngularJS. https://www.sitepoint.com/10-reasons-use-angularjs/. Accessed July 2017.

[48] ADO.NET Entity Framework. https://msdn.microsoft.com/nl-nl/library/bb399572(v=vs.100).aspx. Accessed July 2017.

[49] Entity Data Model. https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/entity-data-model. Accessed July 2017.

[50] Azure App Service, Virtual Machines, Service Fabric, and Cloud Services comparison. https://docs.microsoft.com/el-gr/azure/app-service/choose-web-site-cloud-service-vm. Accessed August 2017.

[51] Finally, Linux Containers Could Run on Windows with Docker's LinuxKit. https://thenewstack.io/finally-linux-containers-really-will-run-windows-linuxkit/. Accessed September 2017.

[52] Justin Ellingwood. An Introduction to Continuous Integration, Delivery, and Deployment. https://www.digitalocean.com/community/tutorials/an-introduction-to-continuous-integration-delivery-and-deployment. Accessed September 2017.

A Project Management and Retrospective

In this chapter, the project management activities of the project are explained. The objectives of the assignment were multidisciplinary and not strictly technical, which made the whole process a lot more interesting and special.

A.1 Project approach

The initial questions of the project were at a higher level of abstraction, which required a more entrepreneurial approach. While the direction and the path to follow were not clear, the goal was of great importance not only for the region and its adoption to smart technology but also for VIBe and its continuity. Therefore, it was clear from the beginning that the technical part of the assignment would derive from the analysis conducted for the VIBe recommendation. Consequently, effort and time were invested in understanding deeply the smart domain and afterwards, the status of the region in the smart area.
The creation of a prototype of the Smart Mobility Platform and an application were the technical aspects of the project. As mentioned in section 1.5.3, the V-Model engineering process was followed from deriving the requirements up to creating a working prototype of the platform and the application. On parallel, throughout the whole project frequent iterations of the VIBe recommendation were conducted that resulted in the final version on October 2017.

A.2 Project stakeholder analysis

In this section, the stakeholders of the project are presented. The project owner and major stakeholder is VIBe. An equally important stakeholder is considered to be TU/e and the reasons are analyzed below. Finally, the importance and impact of the results of the project have a direct influence on the business world of the region of Eindhoven, as well as the regional authorities. Below, a stakeholder matrix presents the overview of the stakeholder analysis (figure A.1).

A.2.1 Technical University of Eindhoven

The Technical University of Eindhoven has multiple roles in the project. Firstly, as aforementioned above, TU/e is an important member of VIBe via Professor Maarten Steinbuch and therefore very interested in the results of the project. Additionally, it is necessary to keep the appointed supervisor from the university satisfied, who has the task of ensuring that the project is in the correct direction

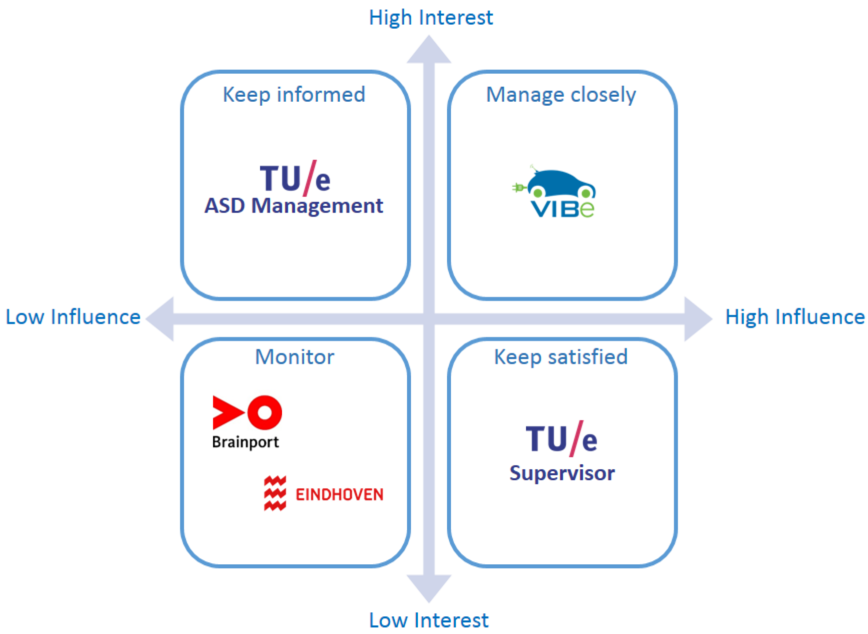

Figure A.1: Stakeholder matrix

and fulfills the PDEng requirements. Finally, the ASD Management needs to be informed regularly about the progress of the project, in order to be reassured that there are no issues with the process.

A.2.2 Region of Eindhoven - Brainport

Although Brainport is not directly linked with the project, several companies provided valuable insights on the status of the region, which proved to be crucial in understanding the current regional issues. Furthermore, these companies can potentially aid with the technical aspects of the project and therefore, their feedback should be taken into consideration.

A.2.3 Local authorities - Gemeente Eindhoven

Similarly with Brainport, the local authorities represented by Gemeente Eindhoven, are not directly linked with the project. However, it is very important for them to understand the problems of the region in order to act accordingly. Thus, the authorities are an equally important stakeholder that need to be monitored and informed about the results of the project.

A.3 Planning

The following figure displays the final planning as it was defined. It is important to note that the planning kept being adjusted according to the progress of the project and with the goal in mind to meet the deliverables by the end of September.

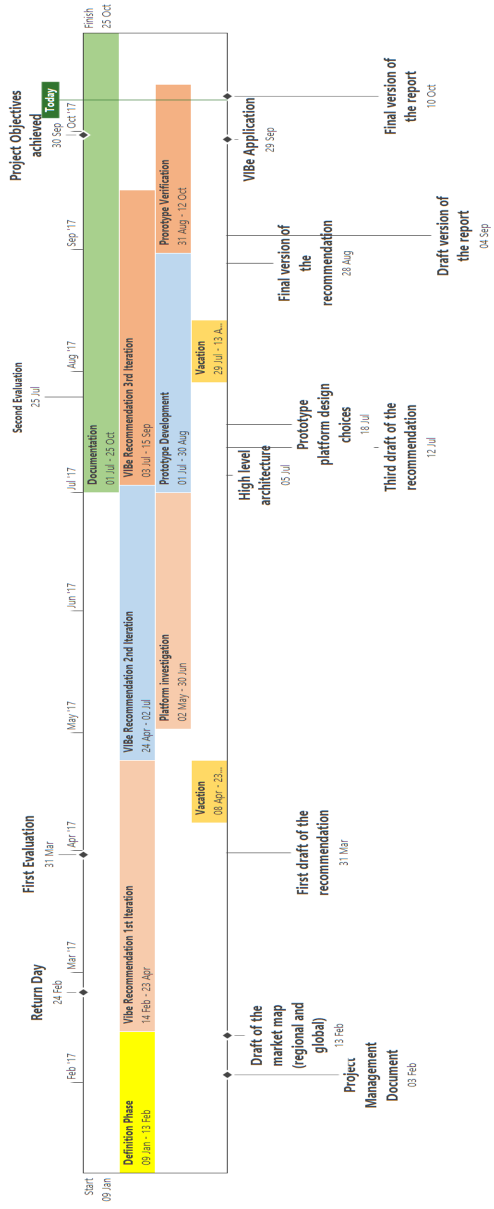

Figure A.2: Final planning

A.3.1 Activity-cost estimations

From the beginning of the project, the intention was that an application will be created using cloud services. This resulted in a small cost of using Microsoft Azure services, which was presented in Chapter 8.
The activities of this project had many facets. From discussions with Chief Executive Officers about strategy and visions regarding the future to understanding cloud technologies and how to design smart systems. The following figure gives an overview of the activities performed:

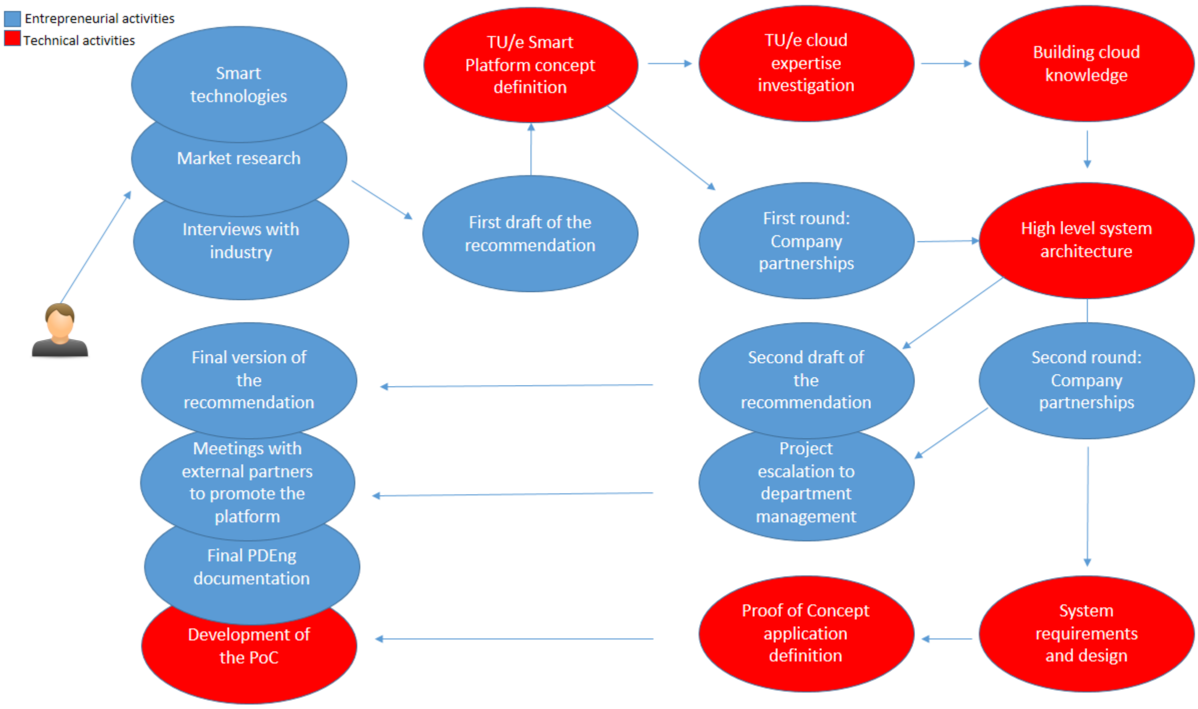

Figure A.3: Activities estimations

A.4 Communication management

As mentioned many times before, the project required a lot of meetings with people from the industry for various reasons (VIBe recommendation, TU/e Smart Platform). Through the course of the project more than 25 meetings with people from the industry were conducted, 9 Project Steering Group Meetings (PSGMs) and countless meetings with the Project Owner and Mentor, Hans Brouwhuis. The plethora of meetings often coincided with technical work about the prototype plaform and the application development. Therefore, careful planning and arrangements beforehand were made. Before every meeting, an agenda was sent so that the participants have a chance to prepare. Similarly, afterwards, the minutes of each meeting were written, recording the agreements and conclusions of that meeting.

A.5 Risk management

To mitigate potential risks of the project, a risk register was followed and reviewed every two weeks. In case of critical risks, actions were discussed during the PSGMs and the discussions with the Project Owner. The final version of the register is presented below:

Risk Identification	Qualitative and Quantitative Risk Analysis				Risk Response Planning	Monitor
Name	Probability	Impact	Exposure	Proximity	Risk Response	Risk Status
1. The available time will not be enough to finish.	M	VH	H	Within project	Limit the scope. Define highest priority deliverables.	Active
2. Difficulties in syncing everyone's agenda.	L	M	M	Imminent	Organise meetings early. Send reminders to late emails.	Active
3. Interviews might be difficult to organise and cause loss of time.	M	L	M	Within stage	Organise the interviews early. Ensure the best possible outcome so that more meetings are not needed.	Inactive
4. Technical design aspect will not be clearly and timely defined.	L	VH	H	Imminent	Make use of the meetings with the industry. Request feedback from the PSG meetings. Utilize experts.	Mitigated
5. The design will not have a connection with the other objectives.	L	M	M	Imminent	PSG meetings. Project Management Document.	Mitigated
6. Lack of support for the business objectives	M	L	M	Within stage	Request early on support from Hans Brouwhuis	Mitigated
7. The interview questions will not inspire useful answers.	L	H	M	Within stage	Many reviews of the questions before the interviews. Put effort in the preparation. Ensure the proper level of understanding	Inactive
8. Lack of business knowledge will cause poor result in these objectives.	M	M	M	Within stage	Regural feedback from appropriate people. Ask for support if needed.	Mitigated
9. Verification of the design will not be done due to lack of time or resources.	M	H	H	Within stage	Agile planning. Define the verification strategy from the early stages.	Active
10. The platform will not be acquired (the meetings with companies will be unsuccessful or on time)	M	VH	H	Imminent	Strict, agile planning. Frequent reminders of the time left. Secure the minimum deliverable from early on.	Active

Figure A.4: Risk register

A.6 Project Retrospective

This section provides an opportunity to reflect on the PDEng assignment. As in every project, there were aspects that progressed very well and other aspects that were more challenging. In any case, valuable lessons were learned and personal growth was achieved in several areas.

A.6.1 What went well

The project started with the goal of creating a strategic recommendation for VIBe regarding the next steps in the smart domain. Based on this recommendation the continuance of VIBe would be decided. In these 10 months, the expectations of the client were surpassed as not only the recommendation was conducted but a prototype of the platform and a simple application were designed, realized and demonstrated. In this way, the project really opened the discussion regarding taking the necessary steps towards exploiting new opportunities in the smart area. Furthermore, through this project, TU/e has an opportunity to open new research areas and lead the region with respect to the smart technologies, as well as attract more talent, students and companies. Similarly, the region of Eindhoven and the local authorities can now link with TU/e in creating showcases of smart applications and dive into the smart domain. Finally, the management of the project was successful despite that it involved several multidisciplinary activities (technical design, entrepreneurship).

A.6.2 What was challenging

As mentioned before, the project did not have a clear technical goal in the beginning. On the one hand, this provided freedom to explore all possible directions but on the other hand, required a lot of time to identify which direction would make sense for VIBe and the PDEng assignment. Due to this situation, it was not until May that the Smart Mobility Platform was conceived as a possible technical route that makes sense for all the stakeholders involved. Therefore, only 5 months were remaining to design and implement the prototype. This period included also the summer holidays which made the efforts to arrange meetings with interested parties more difficult.
It is also worth mentioning that there were limited resources in the university regarding building up knowledge in the cloud computing domain. Consequently, it required personal and extensive effort to reach the appropriate level of understanding in order to design the simple web application.

A.6.3 What I have learned

I entered a project having been trained in approaching a problem from only technical point of view and at the end of the assignment, I experienced what it means to be an entrepreneur and a systems engineer, trying to find multidisciplinary solutions that satisfy not only technical criteria but also business and strategic objectives.
I strongly encourage the initiation of more similar projects, as they give an opportunity to understand the roots of a problem (which are rarely only technical) and try to find the solution that satisfies different kinds of requirements (system, business, etc).
Personally, I developed skills outside my comfort zone, which include being able to discuss at different levels of abstraction (from strategic high-level concepts to low-level technical implementations), to link technology with business and to manage stakeholders.

B Entrepreneurial content

This appendix contains more details regarding the business aspects of the assignment. The interviews and meetings conducted, as well as the construction of the market maps that gave useful insights are presented.

B.1 Meetings and Interviews

In this section, the most important meetings and interviews are explained. However, numerous informal discussions took place throughout the project that gave crucial information towards the finalization of the VIBe recommendation.

B.1.1 Initial interviews

- Olha Bondarenko - Gemeente Eindhoven, Strategic Advisor
- Ronald de Beijer - Beijer Automotive B.V., CEO
- Wim Vossebelt - V-tron Technologies, CEO
- Maurice Geraets - NXP Netherlands, Board Member
- Tim Wouda - Ericsson, Head of Transport Systems (ITS)

B.1.2 Meetings for the TU/e Smart Platform

- Ronald de Beijer - Beijer Automotive B.V., CEO
- Wim Vossebelt - V-tron Technologies, CEO
- Tim Wouda - Ericsson, Head of Transport Systems (ITS)
- Rene van der Mast - Ericsson, IoT and Applications, Business Unit
- Bart Coppelmans - HERE Technologies, Business Development IoT, Solutions Program Manager
- Sander van den Hoven - Microsoft Netherlands, Principal Technical Evangelist (1st round)
- Sander van den Hoven - Microsoft Netherlands, Principal Technical Evangelist (2nd round)

B.1.3 TU/e meetings

- Aleksandra Kuzmanovska - PhD candidate
- Prof.dr.ir. Dick Epema - Former faculty member TU/e, Chairman TU Delft Distributed Systems
- Prof.dr.ir. Maarten Steinbuch - Chairman Control Systems Technology
- Dr.Ir. Carlo van de Weijer - Director Strategic Area Smart Mobility
- Bert-Jan Woertman - Commercial Director

B.1.4 External interest meetings

- Guido Dierick - NXP Netherlands, General Counsel, Management Team member, Country Manager
- CeesJan Mol - Gemeente Eindhoven, Strategy Digital Ecosystem advisor
- Marijn van der Poll - vanderPolloffice
- Cas Hariri - TU/e Innovation Lab

B.2 Market maps

As mentioned in chapter 1, two market maps were created in order get a better understanding of the global and regional developments in the context of connected car technologies.

B.2.1 Global market map

Regarding the global developments, the top 300 companies worldwide were analyzed and categorized according to their product domain. The identified categories were chosen based on general functionality and main business case. These information were mainly gathered from numerous sources from the web and particularly the technology market intelligence company CBInsights and the goal was to understand the reasons behind the success of these companies.

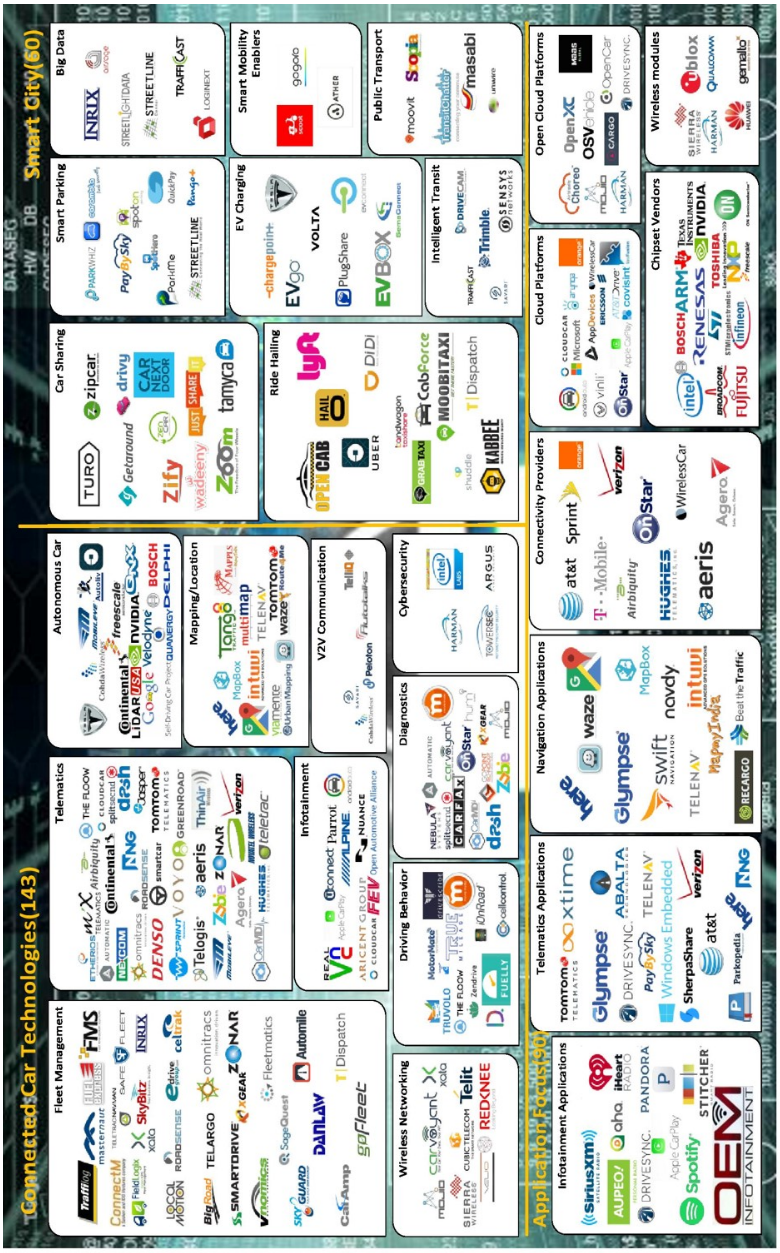

Figure B.1: Global market map

B.2.2 Regional market map

The region of Eindhoven was investigated regarding the companies, institutes or initiatives related to connected car technologies. Information was gathered mainly from the interviews, as well as from the Automotive Week 2017 in Helmond, which took place in March 2017.

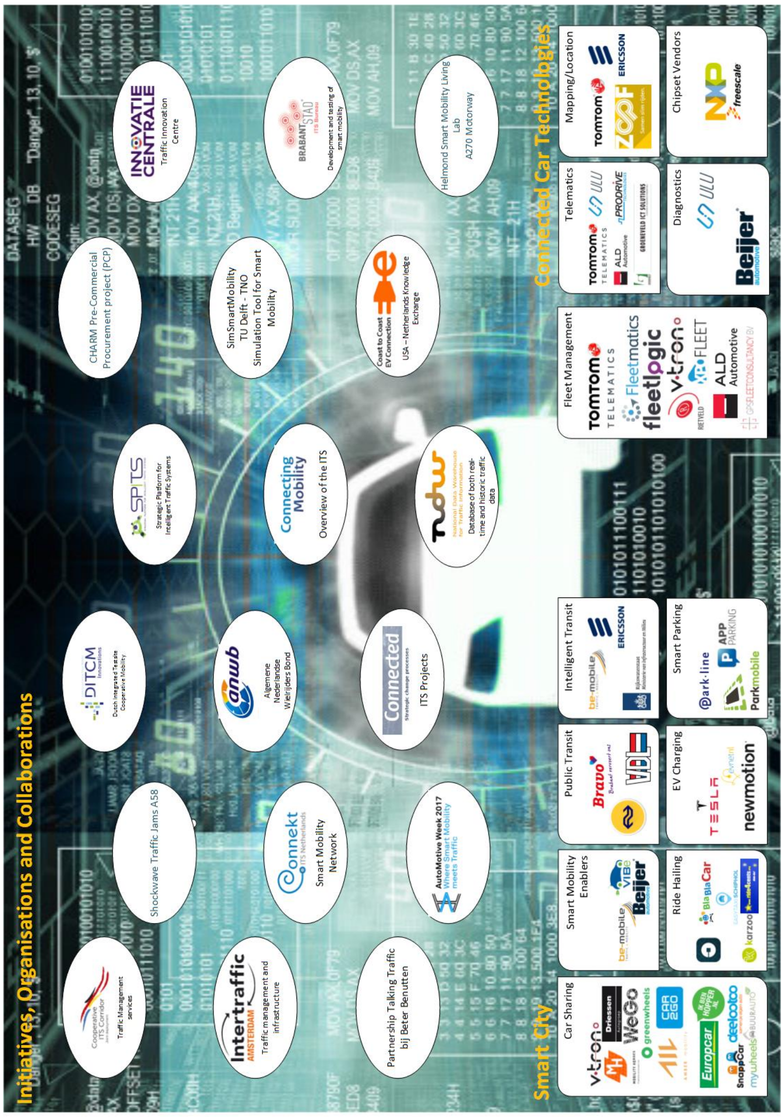

Figure B.2: Regional market map

C VIBe recommendation

This section presents the proposed recommendation to the VIBe Steering Group committee regarding the necessary next steps for VIBe but also the region of Eindhoven in the context of smart mobility and in general the smart arena. Moving from examples of smart cities and zooming in VIBe and smart mobility, key steps are identified in order to propose a vision regarding the future.

C.1 Introduction

Initially, examples of arguably the currently smartest cities in the world are explained. In addition, global developments regarding smart mobility and connected cars are included and thoroughly investigated.

C.1.1 Smart cities

Over the past several years, the idea of the being 'smart' has emerged as a key mechanism for cities to find innovative solutions to the challenges that they are facing, as well as to offer an attractive environment for new talent and entrepreneurship. In addition, increased demand for infrastructure, housing, transportation, jobs, energy, food and water are all straining city governments and infrastructure, as people around the world flock to urban centers in hopes of a better life and more opportunity.
For many years, the push to create smarter cities was led by technology companies looking for uses (and buyers) for their products. However, in recent years cities have begun to think more holistically about what being a smart city could mean, and have innovated new ways to modernize how a city serves its citizens. As shown in figure C.1, several sectors play a role in defining the concept of the smart city. From smart energy and smart mobility to smart education and smart healthcare, these main pillars play in important role in the smart transformation of cities.
Below, some examples of arguably the smartest cities in the world are presented:

- **Copenhagen:** Copenhagen is a living laboratory for testing smart technologies to handle the challenges of urbanization and climate change. Unique access to data and efficient public-private sector partnerships attract many multinationals. Copenhagen has become a preferred living lab for testing and developing smart city technologies, owing to easy collaboration with academia, the public sector and industry. Moreover, the Danes are early adopters of new technologies, and Denmark has a long tradition of citizen involvement in urban planning and development.
  Copenhagen is a breeding ground for smart city start-ups. Around 250 companies are involved in smart city activities in Copenhagen, and small companies make up two thirds of the smart

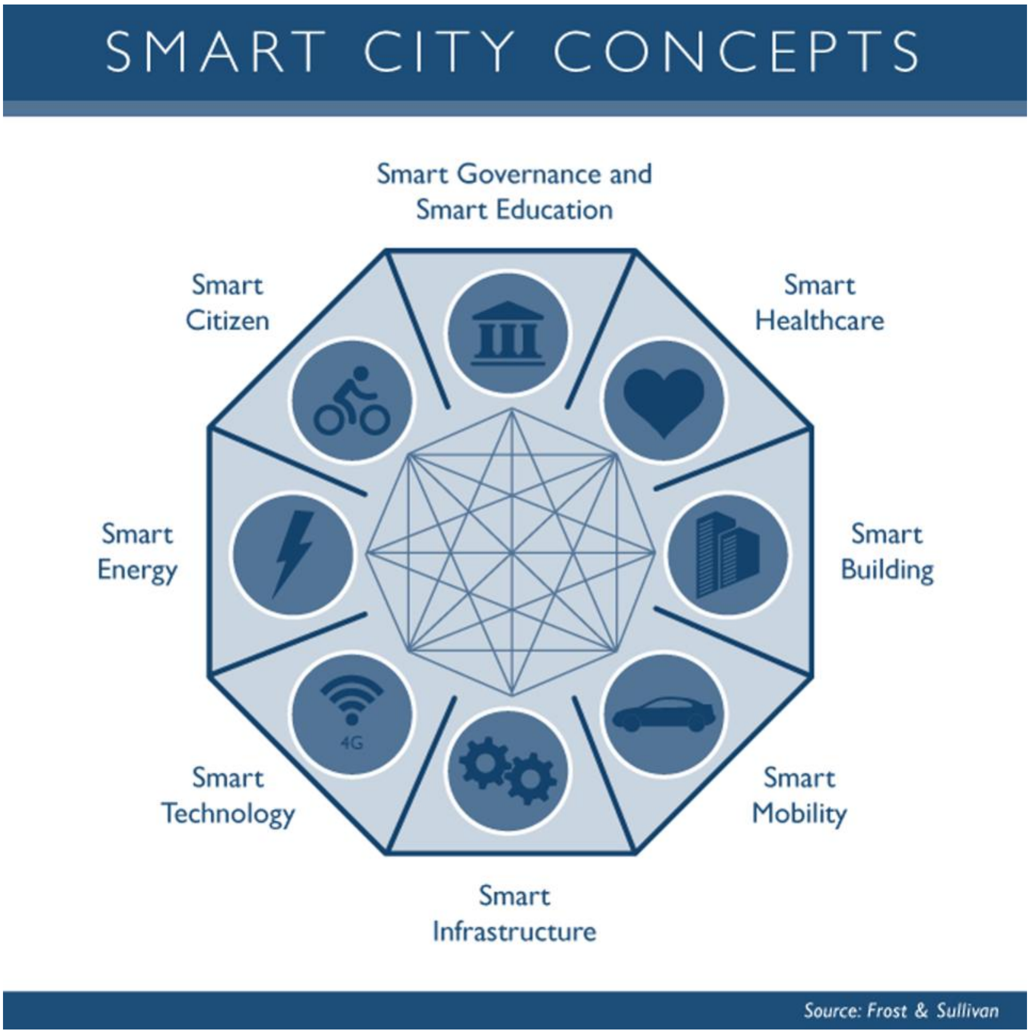

Figure C.1: Smart Ecosystem

city industry, offering attractive investment opportunities as well as bridgeheads for collaboration with the public sector in Denmark. For decades, the Danish authorities have collected and stored basic data about citizens, businesses and real estate in order to digitize services across administrations and sectors. A new government program provides free access to public data sources in the aim to drive smart city innovation.

- **Barcelona:** Barcelona has regularly ranked highly on the annual smart cities rankings. Perhaps it is no surprise then that Barcelona is working on becoming the Mobile World Capital, is the host of the largest annual smart cities event (the Smart City Expo) and was recently awarded the title of Europe's Innovation Capital.
  The city has also made its Sentilo sensor and actuator platform available on the Internet. The open-source software platform can be found on Github. The availability enables city planners around the world to study data from Barcelona's smart city projects and learn from them. Barcelona's 22@ innovation district is also an impressive mix of smart urban planning and entrepreneurial innovation. This sector of the city has been transformed into an innovation home

attracting local and international entrepreneurs to set up shop.

- **Helsinki:** Helsinki really shines in the Smart Government arena. They have more than 1,200 open datasets and 108 applications have been built and are in operation which leverage their open data program. They also played host to the first global Open Knowledge Festival in 2012. Helsinki also launched their Forum Virium Smart City Project to provide ubiquitous data to their citizens in hopes of improving quality of life.
  Helsinki has a strong commitment to digital technology. A full 100% of residential and commercial buildings have smart meters, and 70% of commercial buildings leverage automation systems to enhance efficiency.

- **Singapore:** Singapore is aiming to become the world's first 'smart nation' through a range of initiatives leverage intelligence, integration and innovation to become a major player on the world stage. .Singapore also has an open data platform related to data collected by sensors located on the island. Some cities conduct smart cities projects and do not make their data available to the public. When that happens, it is just the city deciding what will be done in isolation. But Singapore is trying to engage its citizens to help determine what to do.

While there are many more cities that try to transition to the smart area, these are among the front runners for specific reasons. They all follow a people-centric approach, realizing that the purpose is to involve people in the process. Moreover, easy collaboration with academia, the public sector and industry is seemingly the mindset and the difference maker compared with efforts of other cities. Finally, innovating in an open and collaborative environment can capitalize on the collaborative spirit and enable more innovation in the right direction.

C.1.2 Connected car and smart mobility

One of the main pillars of the smart cities is smart mobility. Undeniably, smart mobility and the connected car technologies have been driving the efforts to understand what it means to be smart from technological but also business point of view.

Developments

Especially the connected car technologies have risen in popularity in the last 5 years immensely as part of the current and very disruptive invasion of the smart technologies in various domains. Everyday new partnerships, merges, and collaborations are formed, while the automotive OEMs still explore how they can maintain their market share. Either by investing in creating a complete solution in house or by forming partnerships with expert companies, they try to leverage new technology to open new business opportunities in order better adapt to the new reality. Latest example is the partnership between BMW, Intel (along with the recently acquired Mobileye) and Delphi (since May 2017) for autonomous vehicles.

Many of the contemporary companies recognize that eventually the car will be another connected to the cloud device, just like modern smartphones or wearables, being part of the smart ecosystem. Emphasis will be given to customization of applications that can be accessed from anywhere, even from the car. Each user will have a unique, customized profile that can be accessed by each mode of mobility (an initiative promoting this vision is MaaS Alliance) and would enable the user to choose the navigation system or infotainment application that satisfies the requirements. The attention will

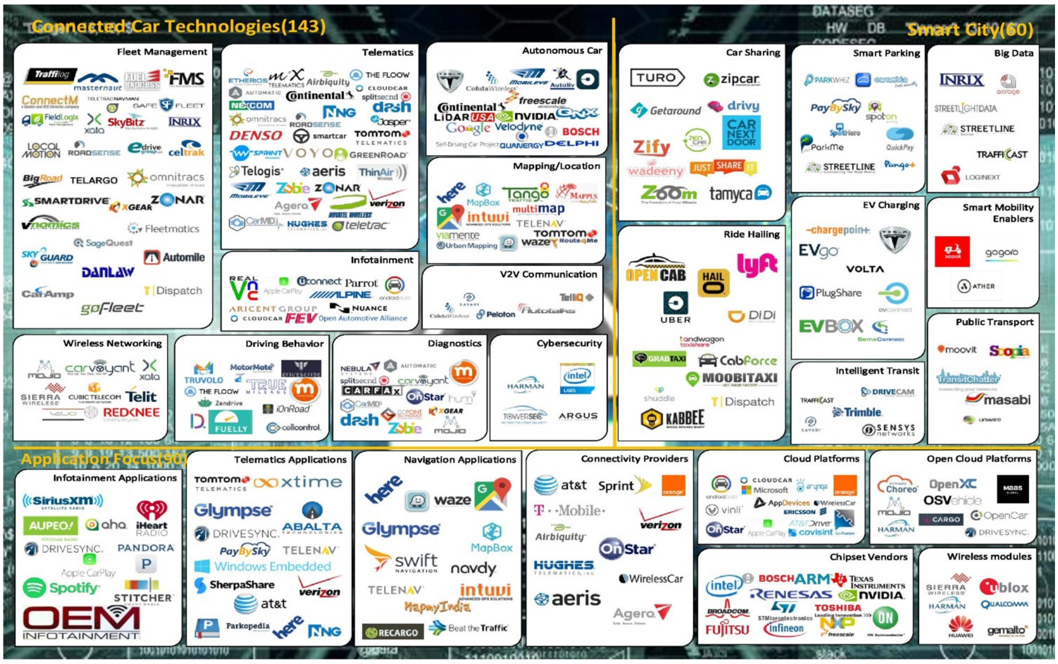

Figure C.2: An overview of the connected car companies around the world

be shifted from the device to the person using that device, creating a unique experience.
Traveling from A to B used to be the main goal of mobility. However, it is expected to be just one of the many services available in the car, which by itself will be another component in the smart ecosystem, providing connectivity with the other smart domains.

Trends

There are several trends in the connected car landscape that are shaping the future of mobility. The most important ones are:

- **Radically new technology at low prices:** Technological innovation is accelerating, particularly the quality of connectivity and sensors. Innovative companies, both established automotive manufacturers and new entrants from the technology industry, are investing accordingly in new technologies and services.

- **New mobility concepts focusing on urban customers:** Urban residents in Western markets appear to be losing interest in owning their own cars, a trend exacerbated by their desire to move to urban areas, where cars simply are not a requirement, and where public transport and ride hailing applications can easily fulfill their needs.

- **Evolving regulatory and policy constraints:** Policy and regulations typically lag behind technological progress, at least in the beginning of a new phase.

- **New high-tech entrants drive faster:** Nontraditional tech companies are gaining traction in the very technology that makes cars run and in so doing, they are disrupting the traditional vehicle technology value chain. Their data-centric business models are different too, being far more dependent on revenue from ongoing services and the sale of information.

- **Smart domain and augmented reality:** Being smart is being able to seamlessly integrate new technology into daily operations. As connectivity (to the cloud, infrastructure, etc) is becoming more and more important, the system integrators face new challenges such as: complexity, security, development speeds and finally, integration of radically new technologies like augmented reality, which introduce a new era of user experience.
  Augmented reality is regarded as a new wave that is expected to have a serious technological and societal impact, even in automotive applications. There are already concepts like augmented reality dashboards or infotainment systems and they are expected to drive the automotive AR market at an annual growth rate of almost 18% by 2020.

C.2 Analysis

After investigating the global and regional developments, a SWOT analysis (strengths, weaknesses, opportunities and threats) for VIBe and the region was conducted, in order to better understand the potential of the region as well as warning signs of wrong direction.

C.2.1 Region

Below, the SWOT analysis for the region and the results are presented:

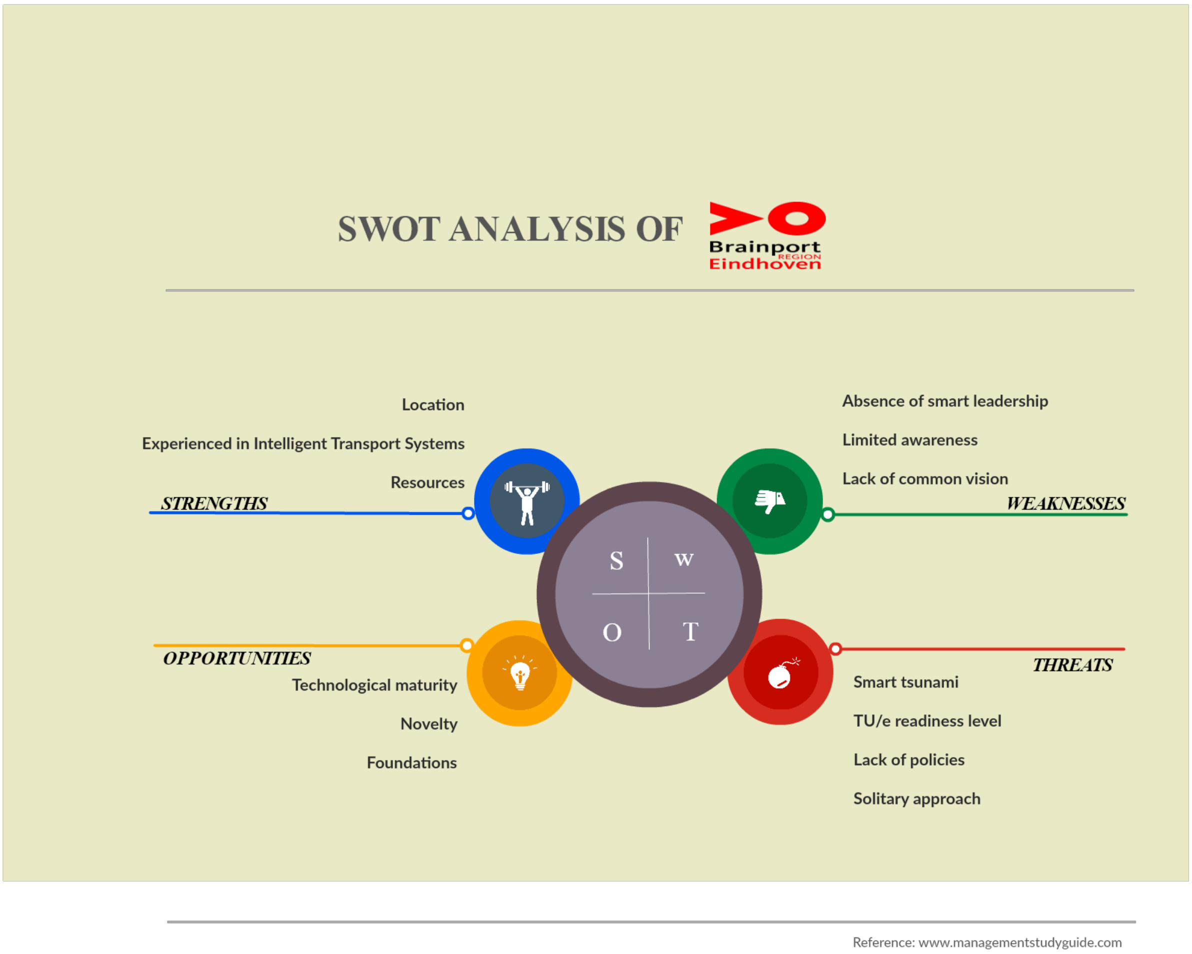

Figure C.3: Brainport SWOT analysis matrix

Strengths

- **Location:** VIBe and in extension the Brainport, reside at the heart of Europe, having the potential to become a technological hub of the continent. In addition, Brainport is considered to be the smartest part of Europe, accommodating some of the most innovative companies of the continent, knowledge institutes and international experts.
- **Experienced in ITS:** The Netherlands is a leader in Traffic Management innovation. A lot of institutes, projects, initiatives confirm that every year. All these projects have provided very useful test data and learning lessons for the future and also prove that the region is open to innovation.
- **Resources:** The individual growth of the companies in Brainport is suggesting that there is room for investments and entrepreneurship. The region is expanding business wise and the question is ultimately, how to better spend the available resources, serving the right vision.

Weaknesses

- **Absence of smart leadership:** As mentioned before, many prerequisites of being successful in the smart domain exist in the region (capital, experts, companies, innovation). However, there is one element missing: leadership. While in the past companies such as Philips could drive a region forward, today no such company exists, especially in the smart mobility area. Consequently, it is highly important to define clear roles and responsibilities, as at the moment the absence of a clear vision creates obstacles in defining the way forward (investments, concepts, planning, etc).
- **Limited awareness:** During the Automotive Week 2017, a survey from Connecting Mobility was presented regarding the perception of the connected car among professional users. Almost 3 out of 4 people were not aware of the term connected car. That example indicates that the region is still not aware of the potential of smart mobility (and in extension smart technology) and what that means for the consumers or the business players of the region.
- **Lack of common vision:** The absence of leadership influences heavily the the creation of a common vision regarding smart mobility and smart cities. The lack of such a vision about the region creates uncertainty regarding the direction of new (or existing) smart projects and eventually fails to convince the region about the potential of these technologies. It also hampers the placement of smart investments in platforms and toolboxes that could transform the region (such as open or multi-vendor platforms).

Opportunities

- **Technological maturity:** The days when the ideas were mature but the technology could not support them, is passed. Currently, technology is rapidly progressing (especially in the mobility sector) and is able to satisfy great demands. It is the first time in history, that it is a matter of putting the right ideas in place rather than focusing on the technology behind it.
- **Novelty:** Smart mobility and smart technologies have yet to be evaluated and standardized. While some regions are further along than others, no one has all the answers yet. This gives a huge opportunity to innovate and create original concepts in this domain.

- **Foundations:** The region of Eindhoven, despite being relatively small, has proven to be open to innovation. From the ITS projects in Helmond to smart lighting grids in Eindhoven, the region has shown that it welcomes new smart projects.

Threats

- **Smart Tsunami:** As mentioned in previous sections, a smart tsunami is coming our way, meaning that many technology sectors are changing. Traditional practices and business models are being replaced by new ones, that focus on the user instead the device itself. Cars are becoming greener, smarter and more software oriented, demanding a change in the existing business models. Therefore the question is how can each company find the way to be part of this smart tsunami, utilizing the opportunities that arise and being agile enough to adapt to the new conditions. Considering the second incoming wave related to augmented reality, the danger of staying at the observer's role rather than actively engaging in these technologies is real.
- **TU/e readiness level:** The Eindhoven University of Technology is an important source of neutral and pure innovation. However, there are issues suggesting that TU/e is not ready yet to lead the region in the smart era. First of all, there are no faculty members with expertise relative technologies and the focus is on the application level only rather than on the complete infrastructure. Moreover, the departments seem unconnected and unaware of the individual smart projects that are being carried on by students. All the aforementioned elements demonstrate that the importance of organizing a smart leadership inside the university as well.
- **Lack of policies:** The slow adaptation of the policy makers (in a country but also at European level) could pose a challenge. The struggle to resolve the data ownership and privacy issues is slowing down the process of enabling connected car technologies. However, the need to establish legislation is identified in the Declaration of Amsterdam as highly critical towards the development of connected car technologies and intelligent transport systems.
- **Solitary approach:** Historically, in case of new technology, most companies try to create complete, vendor specific solutions and claim novelty. This often costs capital, effort and time to all the competing companies, who engage in a last-man-standing competition. If the region is drawn into this process, Brainport will be falling behind compared to other regions that approach the smart transition from a collaborative and unified perspective.

C.2.2 VIBe

Similarly, the figure shows an overview of the SWOT analysis for VIBe and afterwards, the findings are explained in more details.

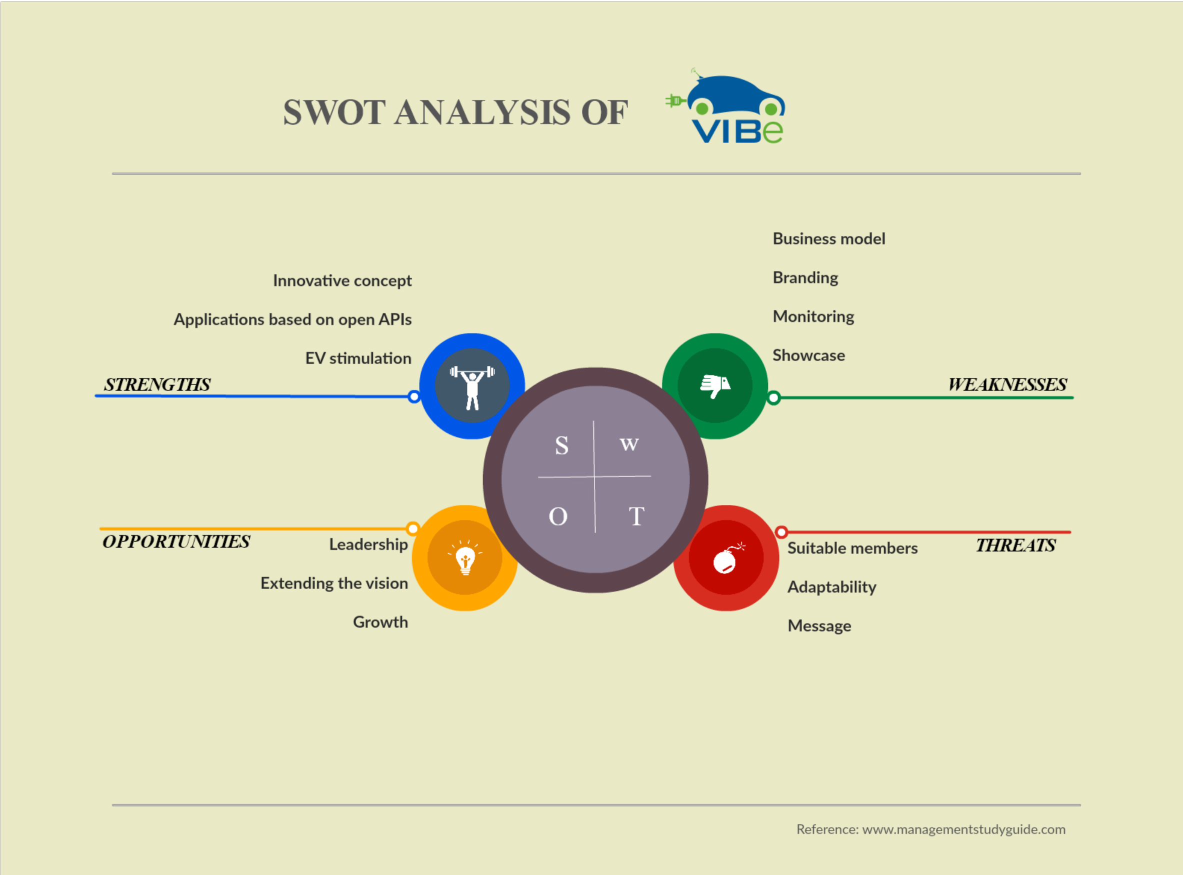

Figure C.4: VIBe SWOT analysis matrix

Strengths

- **Innovative concept:** The initial objective of VIBe was to design and develop an open platform for electric vehicles in the format of a Living Laboratory. The rationale behind this objective was to stimulate entrepreneurs and companies to develop and test their innovative products and services in a real condition environment and produce results faster and cheaper. Innovation is the reason for existence of VIBe and should remain us such.

- **Applications based on open APIs:** VIBe has succeeded in initiating discussions about open architectures and joined initiatives with collaborative spirit. More specifically, VIBe has been responsible for the development of applications by members and partners using the VIBeX API.

- **EV stimulation:** The Netherlands are among the leaders globally in EV adaptation. VIBe contributes by stimulating EV leasing schemes to employees of the VIBe company members but also, by providing a fleet of EVs to be used for application development.

Weaknesses

- **Business model:** The industry is not convinced regarding the value of open architectures and standards, questioning the existence of a viable business model. The reason for that is that companies approach open standards from a business perspective, as companies try to protect their intellectual property and individual commercial value generators.
- **Branding:** The publicity of VIBe is not high enough at the moment, meaning that apart from the current members and partners, very few are aware of VIBe and its mission.
- **Monitoring:** The first idea behind VIBe was to create a communication platform in order to inspire companies to develop their own application using that API. However, after the first release of the API, there was not enough monitoring of the updates and modifications implemented by the companies that used the API.
- **Showcase:** Although there have been a few implementations of the VIBeX API, still the absence of a strong showcase is hampering the effort to attract more partners and to inspire the region. Several times during the interviews with people from the industry, the remark was made that a working proof of case is missing. Apart from the API, there is no establishment of open innovation platforms yet and the absence of relevant showcases is an issue.

Opportunities

- **Leadership:** As mentioned in the regional SWOT analysis, the region is suffering from leadership absence. VIBe has a real chance to tackle this problem by emitting the right vision about the smart cities of the future and provide a clear direction.
- **Extending the vision:** Assuming leadership means to not only address one smart domain (e.g. mobility). The vision need to be adapted in the smart ecosystem context. Having mobility as a showcase, the first steps can be made into understanding how the bigger picture looks like.
- **Growth:** Through embracing the smart ecosystem vision, more sectors are addressed that could also be linked with smart mobility (smart energy, smart business, etc). That could give the chance to VIBe to connect all these domains and inspire multi-domain solutions.

Threats

- **Suitable members:** It is key to have members and partners that believe in the same vision. In this way, everyone is an ambassador of VIBe. In the opposite case, it is difficult to set a strong example in the region and therefore, the message will not be persuasive enough regarding the urgency to adapt to the smart transformation.
- **Adaptability:** The current developments have shown that creating an open API and providing it to interested parties is not enough to create the avalanche effect that can change an industry and alter mindsets. Adaptation is needed to introduce something bigger that demonstrates the added value of using smart technology in an open and collaborative way.

- **Message:** VIBe should not be seen as a business entity. VIBe should represent the vision on what it means to be smart and a plan on how to get there. It should be a hub of members that believe in that vision and a facilitator of projects aiming at the right direction.

C.3 The Way Forward

C.3.1 Towards a smart society

It is clear from all the above that a collaborative spirit is needed in order to survive in the new smart reality. The private sector, universities or research institutes and the authorities need to work closely together, driving innovative and disruptive projects, as well as creating inspiring smart showcases.

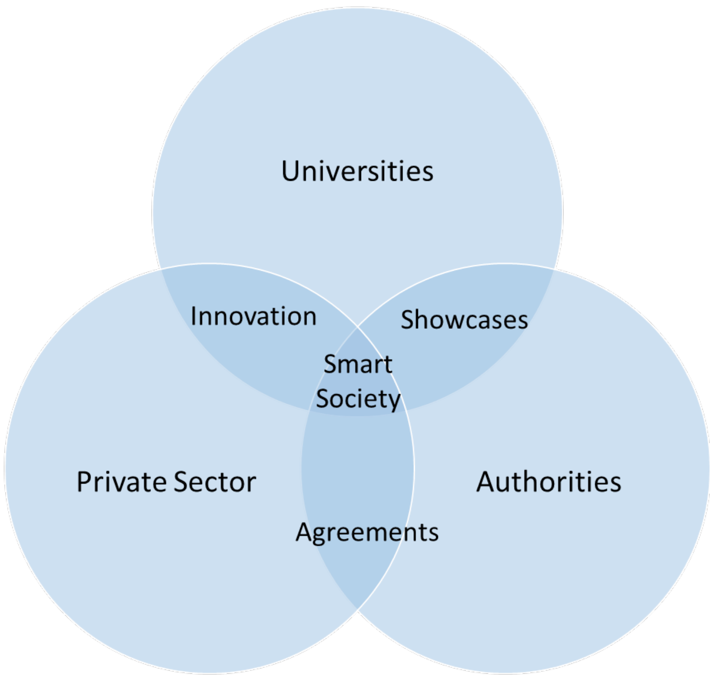

Figure C.5: The Way Forward

A lot of cities are already claiming that they are on the way to become 'Smart Cities' or 'Smart Societies'. The common denominator is always the efficient collaboration of the above three partners. The private sector companies form mutually beneficial agreements with the authorities regarding use of smart systems and handling of public data. The authorities rely on research and educational institutions that can provide showcases to demonstrate the added value of the smart platforms, utilizing the core purpose of these institutions which is to promote new technologies and to provide a peek into the future. Finally, the business world combines forces with the universities to create innovative concepts that can also be implemented in the real world. The combination of all these factors is able to create the necessary foundations to transform a city into a smart society, having as core belief that the cities of the future are people oriented.

C.3.2 Private sector role

As mentioned in the previous chapter, the region has a lot of potential when it comes to having the necessary financial and technical infrastructure to realize innovative concepts. However, leadership and awareness of the actual impact of these technologies are missing, which explains the delay in adapting to the new reality.
The role of the industry around Brainport involves adopting a collaborative spirit and dismissing the solitary approach. The realization that the transition to the smart era can unravel huge business opportunities for everyone involved only if the region follows a unified approach.
However, there are a few aspects that need attention. Firstly, working together also means embracing the open mentality. While this does not mean that everything should be open and free, it certainly means that open development and sharing data under specific agreements are integral ingredients. The individual vendor products that add commercial value to each company will remain black boxes but the interfaces should be open and shared. The industry needs to be an example to follow on how to adapt to the new smart reality, showing vision and decisiveness while understanding the numerous opportunities that will arise, if a more collaborative and open mentality is adopted.

C.3.3 Universities Role

The Eindhoven University of Technology is a strong founding partner of VIBe. Being the place where innovation starts and heavily involved in smart mobility concepts, the TU/e should play an elevated and crucial role.
Supporting open standards and architectures is a common belief of TU/e and VIBe. That, combined with young engineers with fresh ideas can be the spark to create these applications and be the source of inspiration regarding smart (mobility) applications. These students are engineers that will play an important role in the next 10 years in establishing the region as a smart era leader.
However, as mentioned in the SWOT analysis, the TU/e has not still reached the desired engagement level. It is crucial that a world-renowned institute as TU/e understands what it means to be part of the new smart reality. By taking actions such as organizing leadership at a higher level for the purpose of connecting the different departments and smart projects. A unified communication stream is desirable, as well as an integration of different projects that use the same infrastructure (especially related to cloud computing). The university has a great chance to be the leader of the transition to smart, cloud connected platforms, starting from mobility applications and expanding to all smart domains. In that way, partnering with companies, very innovative projects can be created that would provide a chance to build up on previously earned valuable lessons. TU/e can take the lead to demonstrate what it means to be smart in the future and in this way prepare for the next tsunami related to augmented reality.

C.3.4 Authorities Role

The role of authorities is very crucial towards enabling connected car technologies and smart solutions in the region. Policies and regulations can really accelerate the establishment of the required infrastructure.
VIBe and the local authorities have several goals in common. Similarly with the TU/e, one of them is the support for open standards and architectures. The first steps have been made from the local authorities by publishing guidelines for using open architectures (IoT Principles document). That shows eagerness to build the legal foundations for smart city and mobility initiatives. On the other hand, the

business world understandably wants to protect its business models and is very skeptical towards that mentality. If VIBe and the authorities align their goals, it could directly link the policy making with the business world and the public interest, which could lead to more fruitful discussions and agreements.

The collaboration of VIBe and the private sector of the region with the authorities is positive so far but there is room for improvement. The local authorities really want to involve the industry in creating innovative concepts and they want to form beneficial collaborations. In addition, they consider VIBe to be a very good example of an attractive initiative, that can be inspiring for the government to invest in. Therefore, it is definitely interesting to investigate ways for closer cooperation. The local authorities aim to introduce several smart city initiatives in Eindhoven, which combined with smart mobility and intelligent transport systems can really make Brainport a high-tech hub of the region. To realize that transition, the authorities decided to create a smart leadership team that will be able to lead the region with decisive actions. It was widely accepted that until now a lot of capital was spent without having a clear goal and understanding of what it means to be smart. The authorities decided to change that which provides a great opportunity for new agreements, showcases and projects that convey the right vision.

C.3.5 Translating the words into actions

After the thorough analysis, it became clear that an initiative is needed to help the region understand the long term vision.

A possible implementation of this initiative is the creation of a smart, multi-vendor toolbox led by TU/e that can support smart applications, stressing mostly topics regarding the integration technology, the seamless and open application development/deployment and the prevention of vendor lock-ins in integration technology and open platforms.

Focusing initially on smart mobility applications, this platform will enable any developer (students, companies, individual developers) to create and deploy smart applications fast by providing a toolbox with all the necessary tools and infrastructure required. It is important to note, that it will be a multi-vendor platform, meaning that the components and modules can be shared among the developers. In this way, an open, collaborative and shared cloud platform is created that is able to host all different kinds of smart applications in a vendor agnostic and independent way.

The creation of such a toolbox is very important for various reasons. First of all, a showcase environment is created where anyone can develop and test an application but also create prototypes of components. Moreover, it can potentially lead to an open multi-domain platform, where different types of applications can exist and thus, evolving into a smart ecosystem. Finally, it will attract attention and spread the word that the region and TU/e are acting and taking the leadership.

This system not only has the potential to create a vast variety of demonstrators, but also will act as a self-evolving system that could at the end be standardized. Services that create a hardware and cloud provider agnostic platform can be developed and can provide true independence. It is also very important to host the platform in the TU/e context, as it provides a neutral and safe ground for pure innovation.

As potential users of the platform, there are already talks of hosting TU/e connected car hackathon competitions for students and in addition, the High Tech Campus startups are interested in using the platform to create prototypes. It is only logical that the interest will keep rising.

With the struggle between the open standards and the closed systems still in place, it is probable that eventually the system integration strategy will be the most important aspect of the smart architecture.

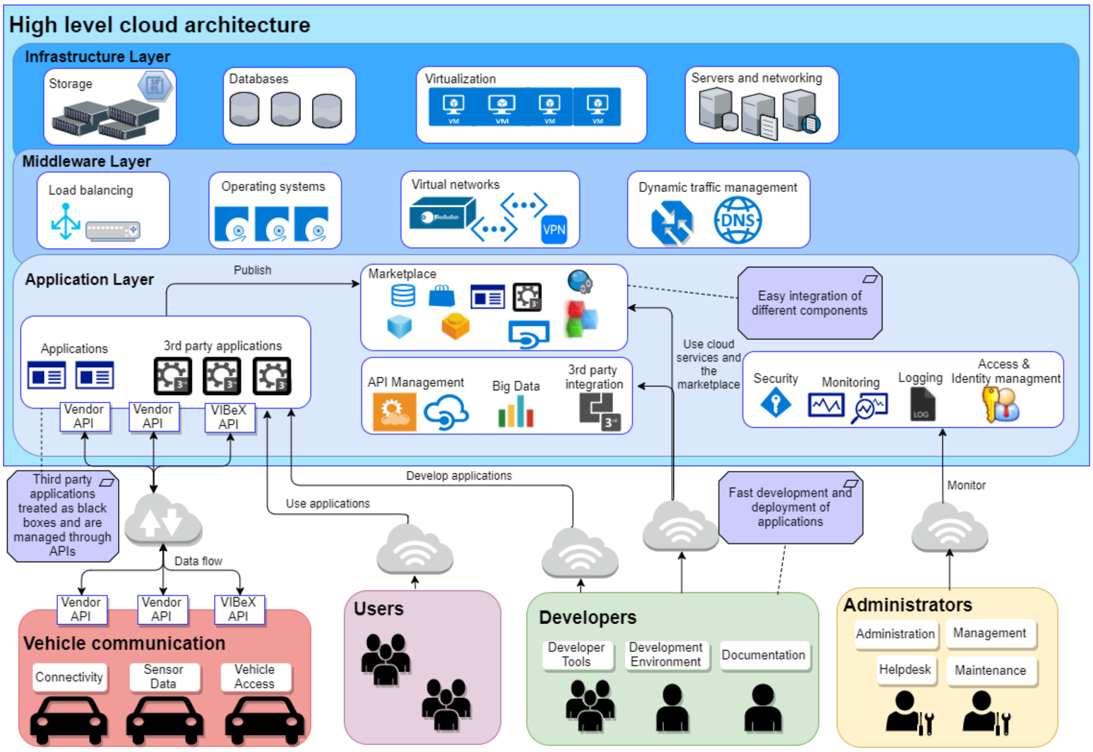

Figure C.6: High-level architecture

C.3.6 Current status - October 2017

The potential of such a platform immediately attracted interested from inside and outside the university. As mentioned above, it is highly important to host the system in the TU/e context and several activities were performed to ensure that.

On the one hand, Microsoft was approached after thorough analysis in order to facilitate with the first stages of setting up the platform. The outcome of the meetings was very positive and Microsoft is very interested in forming a partnership with the university. In this way, the university can also gain experience on how to work with such platforms and start initiating projects into achieving full functionality.

On the other hand, it is desirable that the university adopts officially the platform. Towards achieving that goal, the relevant people were approached (Carlo van de Weijer, Maarten Steinbuch, Bert-Jan Woertman). At this moment, there is a real proposal for the Department of Computer Science and Mathematics to officially adopt the platform and integrate it with the department's and consequently the university's activities. The outcome of this proposal has yet to be seen. However, other departments of the university such as the Electrical Engineering department can also be approached as potential owners of the platform.

In any case, there is a real chance and proposition at this moment to really engage in smart technologies with an open and vendor agnostic mindset. It is important to note as well, that several external stakeholders have expressed interest in the platform already, either from the industry or the authorities.

C.4 What do we want to be - Recommendation

To capitalize on all the above information, a strategy needs to be created taking advantage of the strengths and opportunities but also try to mitigate the threats as well as improve the weaknesses. A clear vision needs to be formulated regarding the future of the region in the smart era and how VIBe can contribute.

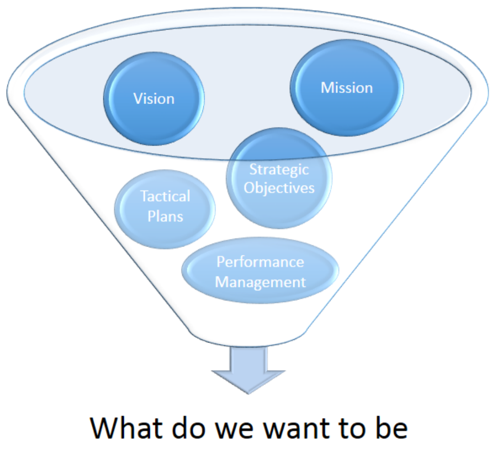

Figure C.7: Strategy management

C.4.1 Vision Statement

VIBe was initiated with the purpose to promote open architectures and standardization, utilizing EV technology. This innovative mindset should remain a core element in VIBe. However, it is clear by now that addressing only one domain is not enough. The realization that the smart technologies should be approached as a whole is needed.
Therefore, VIBe should represent the long term vision that the region needs. It is important to inspire the region to see the bigger picture and how to transform their businesses (regardless of the domain) to be smarter and in this way, unravel new business opportunities.
The Why of VIBe, the purpose, should be to help Brainport come together, adopt a collaborative spirit, accept open architectures and standards and promote innovation in all domains. The How of VIBe is to start from smart mobility and connected electric vehicles as a means to show what it means to be smart today using smart platforms designed to enable innovation in an open and fast way.

C.4.2 Mission Statement

The vision statement can be the foundation of the mission of VIBe and in general the region in the connected car and smart domains context. More specifically, the following questions can shape the mission statement:

- ***Where do we want to be in 5 years?***
  As a region, we want to show leadership and initiative in the smart domain. On one hand,

we need to reach a common understanding and form agreements internally to face the smart tsunami together.
On the other hand, we want Brainport to be used by every major stakeholder (OEMs, authorities, suppliers) as a testing environment for the technologies that will shape our future. We want to be considered innovative (EV technology, ITS initiatives, automated driving,etc), resourceful (past experiment data, lessons learned, history), cooperative (open mindset in regulations and policies), proactive (ensuring the required technology is in place) and most importantly, united (authorities, companies, knowledge institutes all together). We want to be at the front of the innovation and not trying to keep up.
Ideally, VIBe will have facilitated towards the creation of smart applications on several domains using open standards and multi-vendor platforms. Mobility, will be the leading sector, demonstrating the added value of the applications and the infrastructure behind them. In this way, VIBe can also attract more members which will provide opportunities to create multi-domain solutions.

- ***What do we do?***
  On the regional level, we accelerate the establishment of the necessary smart infrastructure and encourage the local industry to initiate more pilot projects not only in ITS domain but in smart mobility in general. The goal would be to have in place a smart platform, where everyone can develop smart applications easily. This is the necessary extension from supplying an open API to supporting an open, smart, multi-vendor platform.

- ***How do we do it?***
  VIBe and the Technical University of Eindhoven are already discussing the establishment of an open multi-vendor platform that will enable rapid prototyping of smart mobility applications. Similarly, this can be extended or connected with similar projects in other smart sectors to form a unique smart city profile. By providing such infrastructure, we acquire a vast amount of data and access to most of European and international projects that can provide valuable information regarding the deployment of these systems. In addition, in this way an acceleration in policy establishment can be achieved. Using the VIBeX API along with platforms similar to the aforementioned TU/e smart platform, showcases of applications can be created fast which is the first step in understanding what it means to be part of the smart ecosystem.
  Finally, rethinking the way of approaching investments and projects at the authorities level is necessary, as the people with the right vision and drive need to be involved.

- ***Whom do we do it for?***
  These are the necessary steps for the region to establish a profitable, innovative but mainly sustainable future in the smart domain context. The private sector companies, the authorities and the research institutes need to jump on the smart wagon as soon as possible. As explained in the beginning of the document, there are already cities that take serious steps towards a long term vision in a unified and collaborative way. Failing to follow a similar approach will result in isolated efforts without a common vision.

- ***What value do we bring?***
  As mentioned above, the region will bring the needed leadership in implementation of innovative, smart projects. In addition, for the region to be able to do that, a common consensus is needed among the stakeholders. VIBe can create that value by being the bridge between the authorities, the industry and TU/e.

C.4.3 Strategic Objectives

From all the above, it is necessary to derive more specific, strategic objectives:

For the region:

- Organize smart leadership at authority and university level
- Form agreements and alliances among the three main sectors (private sector, authorities, universities)
- Initiate more smart projects, starting from mobility and connected car technology but extending the vision to all smart domains
- Ensure the establishment of open, multi-vendor smart platforms (such as the TU/e Smart Mobility Platform)
- Smart investments with clear goals and following a clear vision
- Support new promising startups with funding (Lightyear, Amber Mobility, etc)

For VIBe:

- Be the vision that the region needs
- Adapt the vision to include all smart domains towards the establishment of the smart city through mainly addressing smart mobility and its connection with other domains.
- Promote open mentality and fight vendor lock-ins
- Take the lead to facilitate closer collaboration between the authorities, the industry and the university
- Maintain and promote the EV mentality but support all types of vehicles

C.4.4 Tactical Plans

The question now is how to realize these objectives and what are the steps and guidelines to follow. Regarding the VIBe objectives:

- Evaluation of the current members and partners regarding support of the vision of VIBe
- Effort to include partners from different domains (cloud providers, authorities, smart energy, smart city, etc)
- Create showcases of smart multi-vendor, mobility applications using open standards (VIBeX API)
- Help TU/e adopt in the smart era by proposing (or investing in) applications to be built on the TU/e Smart Platform
- Integrate the feedback from the use of the VIBeX API into a new unified version

C.4.5 Performance Management

It is highly important to have an agile approach in the new smart era. Developments are happening so fast that in order to keep up, regular refinements on the strategy and planning should be scheduled. To make that measurable, at least every 6 months the members of VIBe should meet and adjust the next steps. Close monitoring of the developments regarding open platforms, the VIBeX API and applications developed is needed. In order to be successful in the connected car era, agility and proactive behavior play an important role.

C.5 Conclusion

We think that, the smart tsunami is moving fast and is on our doorstep. It is crucial to make the region more aware of these development and to identify the new business models that arise. The region has a solid foundation to be successful and a pioneer. What is missing is a common consensus, the realization that joined forces by the industry, the authorities and the knowledge institutes will be a great benefit for all parties involved and especially for the region.

VIBe can be the linking component of that effort, promoting an open multi-vendor platform in the university context to inspire more innovation and at the same time avoiding wasted time and capital amongst competitors in their effort to be beat everyone else.

Now is the time to step up and take the leadership having in mind a clear vision on the impact of the smart revolution. Raising awareness regarding smart technologies is definitely not achieved only through presentations or conferences. Showcases and prototype platforms should be created in order to inspire a community. And Brainport is waiting to be inspired.

D Technical content

This appendix provides more details regarding technical aspects of the project. More specifically, the full list of requirements is included as well as the specifications.

D.1 Platform requirements and specifications

D.1.1 Requirements

Requirement	Description	Rationale	Stakeholder	Use Case
FR01	The system must provide comprehensive developer tools and environments	The smart multivendor platform should provide all the necessary developer tools and environments (SDKs, etc) to facilitate the rapid development of applications from developers of the platform. Similarly to website or smartphone application development, the effort needed to create a new component should be minimized.	Data service developer Infrastructure administrator	
FR02	The system must provide heterogeneous systems support and be hardware and cloud-provider agnostic	Not only should the smart mobility platform leverage the latest hardware, virtualization and software solutions, but it should also support a data center's existing infrastructure. While many of the early movers based their solutions on commodity and open source solutions, larger service providers and enterprises have requirements around both commodity and proprietary systems when building out their clouds. Furthermore, the solution should not be hardware dependent and it should be able to support all new technologies.	Data service developer Data service provider Infrastructure developer	Different vendors can use their existing systems to communicate with the platform
FR03	The system must provide service management and maintenance tools	It is important that administrators have a simple tool for defining and metering service offerings. A service offering is a quantified set of services and applications that end users can consume through the provider. The service management functionality should tie into the broader offering repository such that defined services can be quickly and easily deployed and managed by the end user. Finally, there should be clear guidelines regarding the maintenance of the toolbox.	Infrastructure administrator	

FR04	The system should provide flexibility and access control mechanisms	When storing data in the cloud, having a flexible set of access control mechanisms makes it easier to comply with your security policies and helps ensure only authorized access to your data.	Infrastructure administrator Data user	
FR05	The system could provide business-driven configurability	Cloud apps can be configurable, so the organization is freed from costly customizations and business people can configure processes that meet the specific needs of the organization. That said, you should be able to choose from among multiple types of configurations.	Data user	
FR06	The system should support true multi-tenancy	Multi-tenancy eliminates many of the problems created by the traditional software licensing and upgrade models. Multi-tenancy ensures that every customer is on the same version of the software.	Data user Data service developer	In case of updates or upgrades to the cloud platform, all connected applications are updated
FR07	The system should handle regularly delivered, vendor managed updates	A cloud application is a single version of software that is regularly updated, often several times a year. Change every four years is not a sustainable model (and never really was) as the industry is constantly evolving and innovating.	Data user Data service developer	
FR08	The system should support cellular standards and connectivity protocols	The platform should support cellular standards and protocols for communication, connectivity, messaging, etc.	Data service developer Data service provider Data provider	Different vendors can use their existing systems to communicate with the platform
NFR01	The system must support seamless on-demand integration of components	Cloud applications should be built from the ground up to lower the cost, time, and risk of integrating them with existing on-premise and on-demand applications. As a customer you should expect your cloud provider to offer an integration platform and tools, a strong partner ecosystem, and generally whatever assistance you require for pain-free integrations.	Data service developer Data service provider Infrastructure developer	A company can use its existing modules to create a new application.
NFR02	The system should provide high reliability (95%) and availability (95%) and basic security services	Whether the cloud serves as a test bed for developers prototyping new services and applications or it is running the latest version of an application, users expect it to be functioning every minute of every day.	Data user Infrastructure administrator Infrastructure developer	Applications should have constant connection with the platform, handling vehicle data reliably

NFR03	The system must support fast deployment functionality (ideally deployment in a few days)	Since cloud applications don't require investments and installation of hardware and software, you should be able to get them running and productive in a fraction of the time compared with on-premise software.	Data service developer Infrastructure developer	A TU/e student can create a smart mobility application in a few days for his/her project.
NFR04	The system must be scalable and elastic	It is important for the platform to be scalable and elastic with respect to computational resources in order to enable fast growth and expansion of the capabilities of the platform.	Data service developer Infrastructure developer Infrastructure administrator	With the addtion of more applications, the hardware resources are scaled (CPU, memory, storage, etc) to meet the demands
NFR05	The system should account for dynamic workload and resource management	In order for a cloud platform to be truly on-demand and elastic while consistently able to meet consumer service level agreements (SLAs), the cloud must be workload- and resource- aware. Cloud computing raises the level of abstraction to make all components of the data center virtualized, not just compute and memory. Once abstracted and deployed, it is critical that management solutions have the ability to create policies around workload and data management to ensure that maximum efficiency and performance is delivered to the system running in the cloud.	Data service developer Infrastructure developer	The High Tech Campus startups join the platform to create prototypes. That would mean rapid increase of the workload and the resources needed.
NFR06	The system could support easy data extraction and portability	The platform should be able to extract a variety of data from the car and in addition, it should enable effortless porting of data in and out of the platform.	Data service developer Data service provider	
NFR07	The system should provide data privacy and security mechanisms	A cloud application provider should be able to offer world-class security and data privacy better than its customers can do on their own, and at no additional cost. That includes physical, network, application, and data-level security, as well as full back-up and disaster recovery.	Data user Data provider Data service developer Data service provider	

NFR08	The system could have visibility and reporting mechanisms	The need to manage cloud services from a performance, service level, and reporting perspective becomes paramount to the success of the deployment of the service. Without strong visibility and reporting mechanisms the management system performance becomes increasingly difficult. Data center operations have the requirement of having real-time visibility and reporting capabilities within the cloud environment,	Infrastructure administrator	
BR01	The system should be hosted in the context of TU/e	The platform needs to be at a neutral ground in order to avoid vendor lockins. TU/e is the most suitable place to host such a platform which can be beneficial for the university too.	Infrastructure administrator Infrastructure developer	
BR02	The system must support open development and APIs	It is necessary for the smart multivendor platform to support open APIs and standards. An open mentality is desired as an end result and therefore, the application programming interfaces should support open development.	Data service developer Data provider Data service provider	The latest VIBeX API (v1.5) will be one of the supported APIs.
BR03	The system must support multi-vendor functionality	The platform should be able to support components from different vendors, as well as providing the integration functionalities needed. New stakeholders should be able to integrate their own components and use the	Data service developer Data service provider	
BR04	The system should host a marketplace of different components and applications	It is desirable that the platform serves as a marketplace of different components and applications that enable the developers to create unique prototypes. The different vendor components that are hosted in the marketplace are treated as black boxes. However, it is important to have defined interfaces (APIs) to facilitate fast integration of new components.	Infrastructure developer Infrastructure administrator Data service developer Data user Data provider	A developer uses an API from Company A to create an application that communicates with a vehicle from Company B

Figure D.1: Full list of requirements

BR05	The system must enable the seamless creation of showcases of smart mobility applications	The main goal of the platform is to enable applications to be developed and deployed easily. The system should be able to host several showcases of applications without time-consuming restrictions	Data user Data service developer Infrastructure developer	TU/e student competition/ integration with existing compan applications.
BR06	The system should support experimental and commercial development	On one hand, the goal is to open new opportunities for experimental applications for showcase. On the other hand, it should also be possible for a company to use the platform and create their own commercially-oriented prototype	Data user Data service developer	

Figure D.2: Full list of requirements

D.1.2 Specifications

Specification	Requirement	Description	Proximity
FS01	FR01:The system must provide comprehensive developer tools and environments	Microsoft PaaS provides all the needed developer tools and environments such as Visual Studio. In addition, more than 5 programming languages are supported.	Prototype
FS02	FR02: The system must provide heterogeneous systems support and be hardware and cloud-provider agnostic	Using the Container service, any application or component developed will be hardware and cloud provider agnostic. At the same time, most of the popular OS are supported by Microsoft Azure.	Prototype
FS03	FR03: The must provide service management and maintenance tools	Microsoft Azure services include numerous management and maintenance tools (Azure Management Portal, API Management, System Center, PowerShell Command-Lets).	Prototype
FS04	FR04: The system should provide flexibility and access control mechanisms	Microsoft Azure services include access control mechanisms (Microsoft Azure Access Control Service, Active Directory, Azure Active Directory Authentication Extensions)	Prototype
FS05	FR05: The system could provide business-driven configurability	Partnering with Microsoft will open access to a lot of configurable services. Effort is needed in the long term to properly configure these services.	Long term
FS06	FR06: The system should support true multi-tenancy	The administrators of the platform will have the necessary tools for the multi-tenancy functionality via different Microsoft Azure services.	Long term
FS07	FR07: The system should handle regularly delivered, vendor managed updates	Multi-tenancy will be handled via Microsoft Azure scheduled updates and with guaranteed functionality during the downtime.	Prototype
FS08	FR08: The system should support cellular standards and connectivity protocols	Different APIs and connectivity modules will be integrated in the platform and will be available for developers. Eventually, developed components by users of the platform will provide extra functionalities.	Long term
NFS01	NFR01: The system must support seamless on-demand integration of components	Existing applications or components can migrate to the platform easily by using automated processes, which will be developed in the future. Microsoft Azure via the cointainer services and through the support of different programming languages, can facilitate to that direction.	Long term

NFS02	NFR02: The system should provide high reliability (95%) and availability (95%) and basic security services	Microsoft guarantees 99.95% reliability and availability (through geo-replication services and Service Level Agreements or SLAs) and several options for security.	Prototype
NFS03	NFR03: The system must support fast deployment functionality (ideally deployment in a few days)	Microsoft PaaS model incorporates tools and services for fast deployment of new applications and easy integration of existing ones.	Prototype
NFS04	NFR04: The system must be scalable and elastic	Using Microsoft Azure's public services, it guarantees automatic scalability and elasticity.	Prototype
NFS05	NFR05: The system should account for dynamic workload and resource management	Microsoft Azure public services in combination with specific PaaS models, can provide premade load balancers and dynamic workload functionality.	Prototype
NFS06	NFR06: The system could support easy data extraction and portability	With more vendors using the platform, it will be possible to choose among a variate of data extraction components.	Long term
NFS07	NFR07: The system should provide data privacy and security	Microsoft Azure privacy and security options are ensuring privacy and security of data.	Long term
NFS08	NFR08: The system could have visibility and reporting mechanisms	Microsoft Azure provides a lot of services for monitoring and reporting functionalities.	Long term
BS01	BR01: The system should be hosted in the context of TU/e	TU/e will be the owner and administrator of the platform. Not necessarily responsible for maintanance, as Microsoft public services take care of maintenance.	Long term
BS02	BS02: The system must support open development and APIs	One of the first APIs will be the VIBeX API, which is open. Also, the developer tools from Microsoft are already free and open to use.	Prototype
BS03	BR03: The system must support multi-vendor functionality	Different companies will be invited to create new applications and use existing components.	Long term

BS04	**BR04: The system should host a marketplace of different components and applications**	**The creation of new applications that can be shared with others, will create the marketplace. Eventually, companies can also put components there for a price.**	**Long term**
BS05	**BR05: The system must enable the seamless creation of showcases of smart mobility applications**	**By enabling open development and use of existing components and developer tools, a lot of applications can be created quickly. For the prototype, a simple application will be developed as a showcase**	**Prototype**
BS06	**BR06: The system should be meant for innovation rather than development of commercial products.**	**The platform is intented to enable development and prototypes rather than create market ready products.**	**Long term**

Figure D.3: Full list of specifications

About the author

A.Krivas received his B.Sc degree and M.Eng in Electrical and Computer Engineering at the University of Patras, Greece. Currently, he is pursuing his Professional Doctorate in Engineering (PDEng) degree in Automotive Systems Design at Eindhoven University of Technology. A.Krivas has participated in several projects in the automotive field.
Some of these projects regard an ADAS Design featuring lane and car detection for TomTom, an accurate in-car 3D reference for automated driving using PreScan for TASS International, as well as design and proof of concept of an electric-assisted waste heat recovery system using preview information for DAF Trucks.
His master graduation topic was Study and Design of an On-Board Diagnostics System (OBD-II) for vehicles (CAN compatible). His PDEng final graduation project is about Cloud Connected Vehicles in collaboration with the VIBe initiative.

Where innovation starts