MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien
Degree Program Embedded Systems

Analysis and Processing of Smart City Data

By: Martin Pahr, BSc
Student Number: 1820297001

Supervisor 1: FH-Prof. Mag. DI Dr. Friedrich Praus
Supervisor 2: Christopher Chung, M.Arch.

Vienna, October 26, 2020

FH University of Applied Sciences TECHNIKUM WIEN

Declaration of Authenticity

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz/ Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.”

Place, Date

Signature

Kurzfassung

Das Thema der vorliegenden Masterarbeit handelt von den immensen Datenmengen, welche in einer Smart City durch Sensoren entstehen. Diese Daten müssen von Rechnern oder Mikrocontrollern interpretiert und verarbeitet werden, um danach verschiedenste Aktoren anzusteuern. Generell beschreibt diese Arbeit, was eine Smart City definiert und in welchen Bereichen noch Handlungsbedarf herrscht, um das Leben in der Stadt angenehmer zu machen. Als praktisches Beispiel wurde eine Simulation zusammen mit einer Smartphone-App entwickelt, welche zeigt, wie Daten auf dem Smartphone gesammelt werden und danach autonom fahrende Busse die effizienteste Route durch den Stadtverkehr berechnen. Durch die Simulation soll gezeigt werden, dass mithilfe von Technologien, wie dem Internet der Dinge, der Verkehr auf den Straßen in einer Smart City ressourcenschonender, aber vor allem sicherer werden kann. Mithilfe des CAD-Programms Rhino und dessen Erweiterung Grasshopper konnte eine realitätsgetreue Nachbildung von autonom fahrenden Bussen erzeugt werden.

Durch die Erfüllung der an diese Arbeit gestellten Anforderungen konnte gezeigt werden, dass computerunterstützte Systeme bei der Datenverarbeitung in einer Smart City helfen können, da diese lesbar aber vor allem interpretierbar gemacht werden.

Schlagwörter: Smart City, Big Data, Internet der Dinge, Autonome Busse

Abstract

The topic of the present Master's Thesis is about the immense amount of data, which are generated through sensors within a Smart City. These data need to be interpreted and processed by PCs or microcontrollers in a way that different actors can be controlled afterwards. In general, this Thesis describes how a Smart City is defined and in which areas actions are needed in the future to make life in a city more comfortable. As a practical example, a simulation together with a smartphone application was developed, which shows how data on a smartphone are collected and autonomously driving shuttles are calculating the most efficient way through the city traffic on their own, afterwards. The purpose of this simulation was to illustrate how technologies, such as the Internet of Things, could improve the traffic within a Smart City and make it more resource efficient as well as safer. With the help of the CAD software Rhino and its plugin Grasshopper, a realistic copy of autonomously driving shuttles was developed.

The fulfilment of the Thesis' requirements reflects that the use of Embedded Systems could help with the data processing within a Smart City, particularly to make it readable as well as interpretable.

Keywords: Smart City, Big Data, Internet of Things, Autonomous Shuttles

Acknowledgements

With these lines I would like to thank my family. The graduation of this Master's program would not have been possible without you.

In addition, I would like to say thank you to the supervisor of my home university, Mr. Friedrich Praus and the supervisor at UM, Mr. Christopher Chung. Thank you so much for your continuous support during the last month.

Table of Contents

1 Introduction

Technology has already become an essential part of the daily life of human beings. That fact is supported by the growing number of weekly smartphone and computer users for example in Austria in the year 2018 [55]. The use of social networks or online games is and will be growing in the next few years even more. This continuing growth can also be seen in the development of today's cities. City planners and researchers are doing their best to be ready for the increasing number of residents with the use of technology.

These 'smart' cities have different core applications, such as Smart Waste Management, Smart Building Automation or Smart Security systems which make a resident's quality of life better [56].

This Master thesis concerns with the data which is generated in a Smart City. Therefore, the collection and processing of data in a city is analysed. In addition, as part of a project, a simulation is developed where Autonomous Shuttles pick up and drop off a random number of passengers. This simulation is realized in the software *Rhinoceros 3D* (abbr.: Rhino) with the plug in *Grasshopper* from Robert McNeel and Associates. Grasshopper allows to write code in the programming language Python and to use 3D drawings as input for Python scripts. So, the drawings can be animated, and special effects can be implemented (e.g. changing the colour or size of an object).

1.1 Motivation

Approximately 180,000 people move to cities every day and cities have to prepare for that. So, the cities must become more efficient and of course more sustainable than they already are, they have to become intelligent or in other words 'smart'. The implementation of new technologies, like Internet of Things, could be the solution [1]. Smart Cities are on the focus of researchers since the last couple of years (see [2], [11], [25]) and different technologies will find their way into such an environment, what leads to immense amounts of data. That is why the following question comes up: How can raw data that is provided by Smart City sensors be analysed and processed and specific actors be controlled as a result of the analyzation?

As part of the 4th and final semester of the Master's degree course 'Embedded Systems' at the University of Applied Sciences Technikum Wien each student has to write a Master thesis. The topic 'Analysis and processing of Smart City data' was chosen because the author is highly interested in helping to develop technology for future needs.

This Master thesis was written in cooperation with the University of Miami in the United States of America, where the author did an on-site research about Smart Cities at the School of Architecture’s Responsive Architecture and Design lab (RAD), particularly about one application of it, Smart Mobility.

This thesis provides the reader with information about Smart Cities, Big Data and the Internet of Things. Additionally, the author has developed a simulation about autonomously driving shuttles within a Smart City, together with a smartphone application to call such a shuttle. During the 3rd semester, every student has to do an Embedded Systems Project, which normally corresponds with the Master’s Thesis in the 4th semester. Noteworthy is that the author got the chance to do research about Smart Cities in the United States of America. That is the reason why a different topic other than the description of the Embedded Systems Project was chosen.

1.2 Definition of task

The main reason of travelling to the US and doing research was to create a simulation about one Smart City application, Smart Mobility. A computer software should simulate how autonomously driving shuttles pick up and drop off passengers at certain stops. A detailed description of the simulation requirements can be found in Chapter 2.

In addition to the simulation, a detailed research about intelligent cities, the so-called Smart Cities, should have been done. So, this thesis contains a theoretical study about Smart Cities and a use-case of a Smart City application. As the topic Smart Cities together with Big Data and the Internet of Things is very wide ([11], [12], [13], [14], [5], [15]), this thesis mainly focusses on parts which are essential for the implementation of the Smart Mobility simulation.

The research project is separated into two parts, a theoretical research and a practical part, where the simulation is implemented:

Part 1: Theoretical research

The theoretical part includes research for journals and books in the online sources of Springer, Springer Open, ScienceDirect, OECD, IEEE, Google Scholar and Google Books as well as an Internet research with the following searching criteria: ‘Smart Cities’, ‘Big Data in Smart Cities’, ‘Internet of Things’, ‘Smart City sensors’, ‘Security in Smart Cities’ and ‘Smart City infrastructure’. The results can be found in Chapter 3.

Part 2: Development of a Smart Mobility simulation

The author of the present thesis developed an algorithm in the scripting language Python, which serves as possible use case within Zenciti, the University of Miami's concept of a Smart City (see Chapter 3.5.7). This use case belongs to the service Smart Mobility within a Smart City. Chapter 5 presents the results where a closer look at the algorithm and different scenarios is taken. Additionally, the author has created an iOS app that could be a part of the later described ZBook application (see Section 5.4). This app simulates the process of calling an autonomous shuttle within Zenciti.

1.3 Structure

The thesis starts with an introduction in Chapter 1, where further the main keywords of the thesis are defined. In Chapter 2, the functional and non-functional requirements of the Smart Mobility simulation are presented. Chapter 3 includes the theoretical part of this thesis, where the State-of-the-Art is presented to the reader. This chapter informs about the key characteristics and the infrastructure of a Smart City. Additionally, the two main parts of the research question are annotated with practical examples and concepts of Smart Cities. A further point within this chapter is a short introduction to self-driving cars is given to the reader to make a transition to the developed simulation. Chapter **Error! Reference source not found.** includes an architectural approach where the first ideas of the simulation are presented to the reader. In Chapter 5 this Smart Mobility simulation is presented in detail. Within this chapter, the reader gets informed how the requirements of Chapter 2 were implemented. The next chapter (Chapter 6: Discussion) discusses how the Smart Mobility simulation can be implemented into a Smart City. Therefore, the process of collecting and processing data is considered. In addition, a connection between the selected topic and the simulation is presented. A conclusion in Chapter 7 sums up the results of this research project followed by possible improvements of the presented simulation as part of a future work.

1.4 Definitions

This subchapter should give the reader an overview of the main technologies which are described within this thesis. Big Data as well as the Internet of Things are two essential parts of realizing a future Smart City.

1.4.1 Smart City

In general, a Smart City uses technology to improve the conditions of living in a city [2]. The vision of city planners is to make living for everyone better than it is nowadays according to the exchange of a huge amount of data about energy, waste and recycling, etc. Technology development does not stop, what leads to decreased costs on computation power and further to a change in speed how the data of a city can be analysed and processed [1]. A Smart City enhances the live of citizens, especially by using data which improves housing or transportation, for example. Unfortunately, there is still a lack, especially when thinking of security. Moreover, it is not easy to trace where the Smart City data comes from because the sources (i.e. sensors) are distributed over a wide range [3].

A Smart City is often named as a system of systems because it contains different platforms which share information together [4]. Such platforms are in general a variety of embedded and intelligent systems and could decrease the costs of Internet of Things applications a lot in the future. The main aspect in a Smart City is focussed on the sensors, which provide helpful information of temperature or the pollution in a city, to serve as examples. These sensors are then processed by different city units. So, a sensor which is detecting fire is directly connected to the Fire Department. Therefore, different scenarios could be detected by different sensors (see Figure 1) [5].

Figure 1: A variety of different accidents in a Smart City [5]. A Smart City where sensors react to different events and contact different departments as a result.

1.4.2 Big Data

The use of today's electronical devices changed the way of how people communicate together, for example by using social media. Even the devices communicate with other devices, what leads to the creation of massive amounts of data, also known as Big Data. The analyzation of these data is also identified as big data analytics and goes hand in hand with the further development of Smart Cities because a city only needs the useful information of the vast amount of produced data [4].

In addition, the four Vs Volume, Velocity, Variety and Veracity identify this data. Volume means that the produced data is larger than gigabytes or terabytes. Velocity describes the amount of data which is collected by companies such as Facebook. Variety is the complexity of the collected data and Veracity tells how reliable the data is, or in other words, if the data from sensors is trustworthy [4].

If this Big Data is correctly analysed and processed, living in a city can be improved a lot. Residents can get real-time information of traffic scenarios which happened on their way to the office or back home. This means that car drivers get informed if a road is blocked and automatically receive the fastest route shown on the car's GPS in real-time. Furthermore, the nearest available parking lot is sent to the smartphone or the car's GPS. In addition, the health of residents can be improved by getting informed about the pollution in the air. When toxic gases are in the air, the residents get informed and authorities can then decide what the citizens should do and what is best for them [6].

1.4.3 Internet of Things

There are expectations that by 2020 approximately 20 billion smart devices are in operation all around the globe. Those devices could be wearables, cars as well as TVs [7]. All these devices connected to the Internet create huge amounts of data that help to provide information for analytics, for example. The connection of these devices to the Internet is called the Internet of Things (IoT). When talking about the architecture of such an IoT device it is important to state that it does not only communicate with the Internet but also with other sensors or actuators [8]. IoT devices operate in a variety of different fields, like transportation or safety. So, especially the maintenance of those devices easily becomes a challenge because of the huge number of available devices, nowadays. Developers must deal with questions, such as how the devices could be installed or updated [9].

Years ago, a computer network consisted of devices, such as Personal Computers, which were linked together. This network enabled the communication of a people, what is known as human-to-human (H2H) communication. When now thinking about different devices which are interconnected over an IoT network, there does not need to be a H2H communication anymore and the result will be a machine-to-machine (M2M) communication [10].

2 Requirements

Different requirements of the Smart Mobility simulation had to be defined in advance. This chapter lists the functional and non-functional requirements. Weekly meetings with the client of the project, the supervisor and the author took place during the development of this thesis to keep the requirements up to date.

2.1 Functional requirements

In general, the listed functional requirements within this chapter describe what the product (simulation and smartphone application) shall fulfil after the final implementation. Furthermore, the requirements outline what benefits the product brings to a Smart City.

FR1: A simulation of autonomously driving shuttles shall be created, where the shuttles pick up and drop off passengers at certain stops.

FR2: The shuttles shall decide on their own (after a computational calculation) how the passengers get to their wanted drop off station in the fastest and most efficient way.

FR3: The user of the simulation shall be able to 'play' with the inputs of the simulation. This means, that the number of shuttles as well as the number of passengers, who are waiting at the bus stops, can be varied.

FR4: The shuttles have three colours (Red, Green and Blue), which indicate where the shuttle goes. So, the colours have basically the same effect such as numbers on the display (e.g. Bus 100 drives to Stop 2).

2.2 Non-functional requirements

The non-functional requirements list features which were used for the implementation process. These features have not been essential for the realization of the simulation.

NFR1: The simulation shall be created in the 3D drawing software Rhino with the plug-in Grasshopper. The used programming language must be Python.

NFR2: The simulation shall be created for a later use in classes to illustrate which projects can be realized in Rhino next to its functionality as a CAD software.

NFR3: The process of calling a shuttle shall be kept simple.

The author of this thesis decided to create a smartphone application with Apple's developing software XCode. The used programming language is Swift.

3 State of the art & Related work

Embedded devices within a Smart City can intercommunicate with other devices over the so-called Internet of Things. The big advantage is that the devices communicate wirelessly in most cases [16][17].

Big Data and the Internet of Things are two commonly used terms related to Smart Cities, where Big Data is the data which is collected from smartphones or sensors that are located in a Smart City and the Internet of Things is responsible for the distribution of the collected data. Different areas within a Smart City can be improved according to the information which comes from electronic devices (i.e. sensors) [18]. The Internet of Things provides city officials or those who are responsible for the maintenance of a city with an unpredictable amount of data. During the last few decades, the environment in cities has changed and city officials do their best to make it more sustainable as well as efficient by taking care of necessary resources, such as water. The Internet of Things is a combination of hardware and software parts. The hardware, such as standalone sensors or sensors in smart devices (e.g. smartphones) in combination with software components, which are able to process the collected information and save it on cloud servers for the processing, makes a city 'smart' and helps citizens to make their living more comfortable than it is, furthermore. Citizens as well as visitors provide officials with data by using the offered services within a Smart City. In addition, Smart Cities contain smart traffic lights or systems which are able to monitor the waste management (e.g. intelligent garbage cans). There are different core systems, like business, education or the public security which are enhanced using technology. One reason, why city officials may distance from using such technology may be the high costs of those systems. The implementation of a machine-to-machine (M2M) interaction is known as the already described Internet of Things and can be used to save costs, compared to already finished complete systems [1].

Today's researchers are working on solutions to implement a Wireless Sensor Network (WSN) into a Smart City. In such a network, sensors are used to monitor the daily life of residents, what is per definition the already mentioned IoT (see Chapter 1.4.3). Especially the development of embedded platforms, which are mandatory to process data of Smart Cities, could provide the needed computational power to deal with Big Data and IoT [28]. This chapter describes which techniques are deployed by researchers and developers for making cities 'smart'. The results are presented within this chapter.

3.1 Smart City characteristics

The authors of [29] list 6 key characteristics which are present in a Smart City:

- Smart Economy
- Smart Environment
- Smart Governance
- Smart Living
- Smart Mobility
- Smart People

Smart Economy within a Smart City offers the possibility of trading in international markets, of working on innovations or creating trademarks. Economy itself is a very important factor in a city because of its association with wealth concerning the goods production, for example. The fact that more people are moving to cities every day will automatically increase the product demand and boost a city's economy [29].

It is a fact that people need to take more care about the environment by reducing CO2 emissions or better handling the recycling of waste. A **Smart Environment** initiative takes care of a city's carbon footprint by switching to hybrid cars or electrical cars. In addition, resources, such as water, can be used to prevent using air condition systems in huge buildings. Therefore, it must be highlighted that resources of the earth can and should be used more sustainable [29].

The organization and the goals, which cities want to reach, can be variably. This form of self-organization is called **Smart Governance** and can lead to a better decision-making and a governance which is transparent [29].

Smart Living means that tourist attractions, safety of a resident, or the implementation of facilities for education play an important role in the decision of city officials [29].

Smart Mobility is an essential characteristic of a Smart City because people need to get to work somehow, either by bus, by a subway, or by bicycles like in Barcelona. The use of real-time data is more and more becoming a trend in huge cities to help people getting informed about upcoming arrivals of trains or busses in the most comfortable way (e.g. smartphone application) [29].

Within a Smart City, **Smart People** help to collect data, for example by using specific apps on their smartphone. Furthermore, the knowledge of residents can help to develop new technologies, what makes the educational background even more important [29].

3.2 Infrastructure

Today citizens trust in the aspect that a city is working, what means the infrastructure within a city helps to live and to work. There is a difference between technical (Mobility, Energy, Recycling, etc.) and social (Health, Education, etc.) infrastructure. Etezadzadeh [19] writes about the consequences of missing infrastructure within a city:

- Contaminated water
- Increase of daily traffic congestions
- Air pollution
- Increase of lease rental charges
- Etc.

About 40 percent of the world's population was using the Internet in 2014 and predictions estimated that 50 percent are using the technology in 2017 according to the growing investments in telecommunication services [20]. Statistics say that about 4.54 billion people used the Internet in 2020 [21]. Therefore, the infrastructure in a city must work as effective as possible [20].

Dameri and Rosenthal-Sabroux [22] name effectiveness, environment consideration and innovation as the most important drivers for smartness within a Smart City. Effectiveness is defined as the way of how effective public and private services work for citizens in this context. Furthermore, a Smart City takes care of its environment itself by trying to decrease impacts on water or air pollution, to serve as examples. A Smart City shall use technology to improve life quality of citizens (i.e. Innovation).

A smart city needs a variety of different network architectures (classes), such as personal area network (PAN), local area network (LAN) or wide area network (WAN). Each application uses different protocols for communication, like illustrated in Table 1 [23]:

Table 1: Network architectures and protocols used in a Smart City [23].

Network Architecture	Network protocol	Protocol name	Area	Application in a Smart City
PAN	IEEE 802.15.4 IEEE 801.15.1	Zigbee Bluetooth	Short range Communication	e.g. Smart Buildings
LAN	IEEE 802.11	Wi-Fi	Long range communication	e.g. Smart Water Network Smart Transportation
WAN	IEEE 802.16	WiMAX	Wide range communication	e.g. Unmanned aerial vehicle (UAV)

Smart Buildings use interconnected sensors as well as intelligent devices to make buildings more efficient. The building is able to communicate with its electronic network to generate forecasts about the demand of the building's users (e.g. water, gas, etc.) [24].

A **Smart Water Network** is very important in a Smart City because the soil's temperature and moisture are monitored the whole time and actors are controlled as a result of the analyzation. The sensors and actors are connected to a network which enables the function to store the results for example in a cloud. Authorized people can then make decisions as information of time and weather are also stored in the cloud [23].

Smart Transportation is a system which monitors the traffic or the conditions of roads. Car drivers can be provided with alternative routes to their destination [25]. Such a system can include for example bike-sharing systems as well where officials are able to control when a bike is used and where the people drive with it [25].

In addition to Table 1, there exist Long Range Wide Area Networks (LoRaWAN), which enable the wireless communication of IoT devices within ranges from up to 40 km in rural environments. The city of St. Gallen in Switzerland decided to build such a network providing a connection within the whole city in 2016 to improve Smart Parking, Smart Mobility, Smart Metering as well as Smart Lightning. In some cases, LoRaWAN can be used in the Active and Assisted Living (AAL) section or the detection of noise with corresponding sensors too [26].

It is very important to consider that there already exists technology which enables a better living for everyone in big cities of today's society: the public means of transport. Their use reduces the CO2 emission what makes them more sustainable and efficient compared to cars [27].

Niedbal [27] says that future Smart Cities also have bicycle rental services because bicycles are good for the nature and environment too. Furthermore, they are the fastest available mean of transport in a city, nowadays. In addition, On-Demand cars offer the possibility to extend the public transport or turn it into Smart Transportation, as those cars are able to calculate the most efficient way through the city for every new ride on their own. It is said that a car parks approximately 96% of the day what makes it highly inefficient in its use. On-Demand cars (i.e. smart vehicles) can reduce the parking time to a minimum.

3.3 Collection of data in Smart Cities

One key factor of data in Smart Cities is to simplify the process of deciding at the right moment. This assumes, that sensor data or data from emergency buttons as well as surveillance cameras is collected and processed afterwards, everything in real-time. When looking at a Smart Home, all the data which is collected with sensors (e.g. electricity or water) is sent to a central system that manages the supply of resources. Therefore, every household only gets the resources it really needs, what makes a Smart City a lot more sustainable compared to usual cities. In addition, the parking situation on every parking spot can be detected in real-time as sensors measure for example how many spots are free at a specific time of the day. According to this information, city authorities can decide if new parking spots are needed in a city. Furthermore, Smart Transportation plays a key role in a Smart City. If a car accident on a road happens, different departments can be contacted in real-time to rescue citizens much faster than today [6].

When collecting data in a Smart City, it is very important to figure out which data is reliable. To serve as an example, data from different cars could mean that data is generated from different manufacturer algorithms, what can lead to a variance. Speaking of Big Data automatically implies that there are huge amounts of data generated and that the data quality is of very high importance. Incorrect or non-standardized data could easily lead to wrong decisions of authorized people in the health-sector or of Autonomous Vehicles on the road. Developers are also faced with the problem of combining the data of a variety of different sources. Three embedded platforms of different manufacturers use variable data structures, for example, what requires processes such as data mining or data diagnosis to combine the information to a so-called high-quality information. Additionally, Big Data must be filtered to give citizens only the information which they need or to provide customers with information about specific products in advance. A sophisticated visualization would then improve the acceptance by a Smart City's resident or visitor and help local governments as well [30].

A smart city is not a vision of the future anymore, it has already become reality. Many cities all over the planet implement community fiber for a faster Internet connection or municipal-wide Wi-Fi to reduce costs for surfing through the World Wide Web (WWW). It does not matter which application in a Smart City is highlighted, the focus lies on data that is generated. This data is generated from sensors which are located in a variety of different locations in a city. Citizens are also helping to generate data by using mobile devices. All the generated information is made available by the so-called Information and communications technology (ICT), which uses IoT to collect such a huge amount of data. This information is then used to improve and simplify the making of decisions in a Smart City by authorized people. Big Data in Smart Cities consists basically of three layers, such as illustrated in Figure 2 [31].

Figure 2: Three layers of Big Data in a Smart City [31]. The first layer lists the devices which are used to generate data. The raw data of this collected information is then processed by different applications. The highest level includes citizens as well as companies which are necessary to implement the changes according to the available information.

ICT is one of the most important services in a Smart City because hardware, software and cloud services can be processed all together. When using sensors for short distances, the well-known, Bluetooth can be used. Wired data transmission is then used for longer distances. ICT works with a centrally located hubs, which are able to connect a huge variety of different devices [32].

There are already huge amounts of data collected in today's cities, but this data needs to be processed in a way that could improve parking, traffic flow, security or waste management. In Nanjing, China, officials installed more than 1 million sensors in taxis, buses and private cars to collect information of the daily traffic and improve routes for drivers, all without the need of building new roads. In Italy, the rail operator Trenitalia equipped trains with sensors to get informed about a train's condition in real-time. The city of Los Angeles is installing interconnected LEDs in streetlights, which inform officials about the status of each light in real-time. Those are only three examples of "smart" cities. There are more cities which are trying to implement or have already implemented smart devices in the United States, for example. Most of these data is stored but forgotten afterwards, unfortunately [31].

According to physical changes in nature, sensors will send data to a manager, or a centrally located hub, where the data is processed afterwards. This enables real-time decision making about a device’s maintenance. To serve as an example, when a bridge is overloaded, the manager could be informed. Furthermore, sensors which detect the vibration of a building can help to reduce risk for humans inside the building when the wind is very high outside. Therefore, the design as well as the maintenance of a city’s buildings and constructions can be improved by using sensors and collecting essential data. Those sensors are embedded as a process of constructing the building or they can be placed in near distance. So, the sensors must be designed in a way that they take no damage in case of a security event and must provide data which is interpretable by authorized people. This means that managers or authorized people must be able to read the collected information and react to it as soon as possible in case of an emergency for example. The sensors as well as vehicles or smartphones will generate amounts of terabytes or even more soon and a city has to prepare its infrastructure for such a vast amount of data [33].

Sensors react to and monitor the physical, chemical or biological happenings in a Smart City. Those happenings are transformed into a signal which is transferred to a more intelligent electronical device. There are antennas which are positioned outside of a building or device whereas embedded sensors are placed inside and can measure the inner temperature of a circuit, for example, what makes the decision which type of sensor should be equipped essential for further decisions [33].

It is very important to provide a Wireless Sensor Network (WSN) in a Smart City. Commonly, sensors were directly wired to a computer or the Internet and the data was then stored on a server. A WSN provides the ideal benefit of locating sensors all around the city and costs could be reduced as well. WSNs have a base station to which all the data is sent via a dynamic path. This path changes when for example one or more nodes are defect and another path leads to the base station faster instead. The whole network must be tolerant against failures, what means that one defect sensor could not lead to a failure of the whole system [33].

There are several different sensors for specific kinds of applications available nowadays. Embedded sensors in a building can detect if a resident goes into the bathroom and does not come out for a couple of hours according to a fall. In the bathroom, a smart toilet, equipped with a heated seat or the ability to play music, is already available today. Other sensors can monitor the sleep and further the health according to the bloodstream. There is also the idea of a smart mirror, which shows weather information or the news and checks the skin for potential diseases furthermore [34].
In addition, there exists another technology to collect information about a passenger’s daily usage of the public transport within huge cities: the smart cards. These cards are necessary to enter the bus, the subway or a ferry, what means, that passengers pay the fees to make use of the public transport within a city.

Furthermore, smart travel cards log the usage of passengers, which therefore can be analysed by data scientists or urban planners. The advantage of those smart cards is the protection of user privacy, what means that there is no way of gathering information about why a specific route has been chosen by the passenger [53].

The growing number of tourists in big cities makes public transport even more essential and necessary. Starting in the 1990s, the described smart cards become popular for the use of public transport. One popular example of a smart travel card is the Oyster Card in London. The card automatically collects information about when a public mean of transport has been entered or left and further the timestamp when money has been added to the card [54].

There are several different benefits of using smart travel cards for public transport, like contactless paying of fees, but these cards can easily get lost, stolen and used by other passengers instead, afterwards. Therefore, the presented application in addition to the Smart Mobility simulation in Chapter 5 can help to make travelling more personal. Users can get personalized routes or information about when the public mean of transport is used by many passengers. All the information as well as the payment process is shown and can be done on the smartphone. Therefore, there is no need for smart travel cards anymore within a Smart City. A user can call an autonomously driving shuttle and the shuttle automatically knows who the passenger is and where he or she typically wants to go, for example to work at 6:30 a.m.

3.4 Processing of data in Smart Cities

The University of Guadalajara has created a Living Lab to analyse data and develop prototypes for the IoT. This type of Living Lab serves as an example for scalability, modularity and security within a Smart City [35]. The purpose of a Living Lab for a Smart City is the connection of academia, government and industry to develop solutions with focus on the citizens [14]. The collected data in a Smart City is considered to be reused by machines or interpreted by humans. The problem is that the information from sensors is raw data, which is hard to interpret, even for a machine. University of Guadalajara's Living Lab uses Arduino microcontrollers connected to sensors and actuators, which communicate with a server, using the MQ Telemetry Transport (MQTT) protocol. The collected raw data from sensors is then stored in a database in form of a JSON document. Villanueva-Rosales et al. found a possible way to make raw data better readable by humans and machines as well using the JSON-LD standard [35].

Big Data used in Smart Cities requires technologies, like the MongoDB (database) or techniques, such as data mining or machine learning, to process the collected high volume

of information in time. The data comes from different sensors, what makes it hard to combine to make a reliable decision of officials as a result of the analyzation [36].
The technique multi-sensor data fusion is important and could be the solution to get information from different systems at the same time. This can include citizen information about energy, transport or mobility coming from multiple sensors integrated in devices like smartphones, computers or even cars. All the sensors or devices are interconnected through a wireless communication network. Such a wireless network could also come from a satellite (GPS) or from a mobile phone's network [36].

Big Data analytics is an essential part of a Smart City and there are already different platforms available, which provide scalable and reliable software solutions as well as the corresponding hardware, such as named by Bibri and Krogstie [36]: Hadoop MapReduce, IBM Infosphere Streams, Stratosphere, Spark.

The platforms support techniques like mining, pre-processing or the very important visualization to make the information better readable. The benefit of these platforms is that they work well with systems that are distributed all over a Smart City [36].

Cloud computing is another model which makes Big Data analytics possible because of the shared computing resources, such as memory or network bandwidth. Therefore, the only device a citizen, an official or authorized person needs is one that can display what is stored on the cloud's servers (e.g. smartphone, tablet, etc.). Reusability, cooperation or interoperability as well as device independence are only some of the many advantages, which are offered by the model of cloud computing. By using for example solar panels to power the necessary devices, the energy consumption is sustainable, furthermore. A middleware infrastructure, what links the kernel of an operating system with the applications, is needed to enable the interconnection and seamless cooperation of different hardware and software systems. Commonly known algorithms for data processing are not as powerful as they need to be for data mining or machine learning regarding Big Data. So, the already existing algorithms need to be improved in a way to handle these datasets, perhaps with programs for artificial intelligence. The interconnected sensors must communicate independent of their location, their infrastructure or the network which connects them to the Internet. Another problem is the difficulty of trusting the data collected from sensors in a Smart City and to control an urban system afterwards. The system or application itself must be able to interpret what the collected information means or in other words, it must react according to a different context [36].

3.5 Examples and concepts of Smart Cities

Developers all around the globe are working with an immense effort on making today's cities 'smart'. The following chapter gives six examples where Smart Cities are not some kind of Utopia anymore and where a future vision turned into reality.

3.5.1 Vienna

Vienna, the capital city of Austria, already started to plan for a more sustainable city in 2011 [37]. A few years later, in 2019, the government of Vienna agreed to a strategy which contains different goals to make the city more liveable, by reducing the CO2 emissions or the energy consumption, for example, until 2050 [38]. To realize these goals, different projects of a variety of city applications are developed (e.g. traffic lights which automatically detect if someone wants to cross the street or drones which monitor PV systems and wind parks etc.) [39].

3.5.2 Singapore

The state of the art in Singapore is that people, who are smoking in zones where it is actually forbidden, can be detected regarding the use of sensors. Furthermore, other sensors allow to detect if people throw trash out of very high buildings' windows. In addition, the city of Singapore works with an open data platform, what means that the data is made available to the public. Smart Transportation is another important topic in Singapore, as city officials are investing in sensors for roads and intelligent traffic lights [40]. The city of Singapore worked together with the nation to implement smartness into Transport, Business productivity or Public sector services, etc. Therefore, they developed a service where data can be shared and where it is made publicly available. In addition, the city of Singapore introduced a strategy for cyber security as well as a Cyber Security Lab which is hosted by the University of Singapore [41].

3.5.3 Barcelona

The city of Barcelona, Spain, installed many sensors which help to manage the everyday traffic on the roads. Furthermore, sensors which collect information about noise and the quality of the air improve the comfort of citizens. The majority of Barcelona's public spaces offer free Wi-Fi for citizens as well as visitors. Officials of Barcelona made the data, which is collected from sensors, available on the WWW [42]. After introducing smart technologies into city services, Barcelona was able to save $58 million on water and $37 million according to the use of smart lighting. The idea and further implementation to make the city smart was supported by stakeholders, like universities or business companies [41].

3.5.4 New York City

New York City has a population of approximately 8.5 million people, what implicates a huge water consumption. The use of water of every household can be monitored, the city implemented an Automated Meter Reading device according to which households as well as officials benefit from. Additionally, smart bins collect information about how many waste is inside the bin and how the garbage disposal works [43].

3.5.5 University of Guadalajara

The University of Guadalajara (UDG) in Mexico built a Living Lab for Smart Cities at their campus starting at the end of 2014 to encourage solutions regarding challenges which cities in the future have to face [14]. The main purpose of UDG's Living Lab is to implement projects that are scalable, interoperable, modular, resilient and secure [14]. A Living Lab consists of three main technologies, such as illustrated in Figure 3:

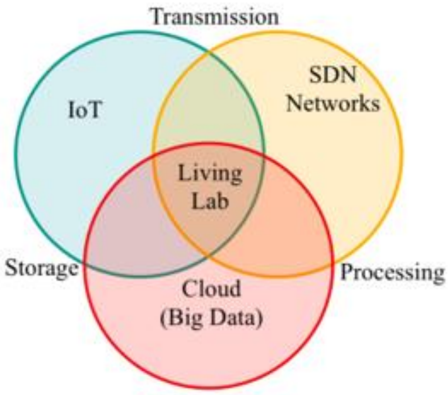

Figure 3: Main technologies of UDG's Living Lab for a Smart City [14]. Software Defined Networks (SDN) are important because the used network technologies for processing the data must somehow be managed by administrators. The Internet of Things is an essential part of this Living Lab as it makes the transmission of data between sensors and actuators possible. In addition, a cloud service is mandatory to process and store the data (Big Data).

UDG uses the MQTT protocol to enable the communication between sensors and actuators connected to Raspberry Pi and Arduino controllers. The collected sensor information is then stored in a service, named MongoDB, which is basically a database, in the JSON format. With an already existing optical fiber network, the data of the following sensor prototypes were sent across the Living Lab: temperature, air quality, light conditions, etc. Sensors will monitor the water or energy consumption soon as well. Additionally, some improvements will be done, so that the devices will be powered by a Power over Ethernet (POE) connection and will be able to do a restart by the device itself [14].

3.5.6 Imperial College London

The Imperial College in London, United Kingdom, has a Living Lab, such as described in the previous section. It is named London Living Lab (L3) and serves as an environment for partners of the college to test their developments around three places: Enfield, Brixton and Hyde Park. At Enfield, lower cost sensors collect information about the air quality, where at Brixton factors like mobility, preventative health or public safety are monitored. At Hyde Park, this Living Lab helps Royal Parks to improve the quality of the water or to monitor the noise and light pollution, to serve as examples [44].

3.5.7 University of Miami

The University of Miami's School of Architecture with its Responsive Architecture and Design (RAD) Laboratory is convinced that computational power, or in other words embedded technology, can be part of every building [45]. The RAD lab works on a specific project with the aim of building a Smart City, named Zenciti, in Yucatan, Mexico. In Zenciti, services, such as the Internet of Things, Augmented Reality or Telepresence help improving the daily life in the city with interactive techniques, like illustrated in Figure 4.

Figure 4: Example of Zenciti's Zbook [46]. Zbook is a platform, what can be customized by citizens to give authorizations to grant access to neighborhoods to visitors, for example. Additionally, information about the weather is shown directly to the user, who gets different interfaces to communicate with the application.

The purpose of realizing the concept of Zenciti is to serve as a laboratory where ideas of developers can be tested in real life [47]. Within Zenciti, traffic patterns, transportation or the energy consumption will be monitored and provide helpful information not only to officials but also to citizens [48].

3.6 Autonomous vehicles within a Smart City

Self-driving cars do not need a driver in the cockpit at the highest of 6 levels of automation which are defined from the Society of Automotive Engineers (SAE). Level 0 means that a driver fully controls the car, whereas level 6 means, that no driver is needed anymore. Of course, different challenges have to be accomplished and different tests have to be passed until such a self-driving car can replace today's cars [49].

Those cars can be used in the cities of the future, the Smart Cities. They can work as taxis and help older residents or visitors to do shopping, drive to the hospital or just to visit friends or relatives. These taxis can be cars for passengers up to two or even four people, but they can also be manufactured as shuttles (autonomously driving busses). The University of Miami's concept of a Smart City is Zenciti, which is described in Chapter 3.5.7. Within Zenciti, shuttles for up to 8 passengers are driving autonomously through the city and pick up and drop off passengers at certain stops.

An elevator can be compared to an autonomously driving shuttle because it transports people from one location to another, either in a building such as elevators do or on the road such as shuttles do. Today, elevators use grouping mechanisms in their elevator algorithm to be the most efficient when going up or down. Grouping in a building can be explained with an example: If a building has 10 floors, and a passenger wants to drive from floor 1 to 8 and calls an elevator car before another passenger wants to go from the 3^{rd} to the 5^{th} floor, the car even stops at floor 3 to pick up the second passenger because the car must drive to floor 3 and 5, nevertheless.

This way of grouping can also be used in the algorithm for self-driving shuttles within a Smart City, to save time for passengers and be sustainable as well as efficient. So, a shuttle has basically the same challenge to transport people from one stop to another. The concept of transportation is the same, but the technology is different.

4 Approach for building a future Smart City

The previous chapter described how data are collected and processed within a Smart City. From the author's point of view, building a Smart City seems time-consuming and expensive at the first moment, but important resources can be saved when investing in more technology. This chapter compares two types of cities:

- A city without technological implementations
- An intelligent (i.e. smart) city

Additionally, a possible method of saving money before building a Smart City is presented.

A Smart City has six key characteristics which are explained in Chapter 3.1 in more detail. All characteristics provide a Smart City with data, like illustrated in the Figure below.

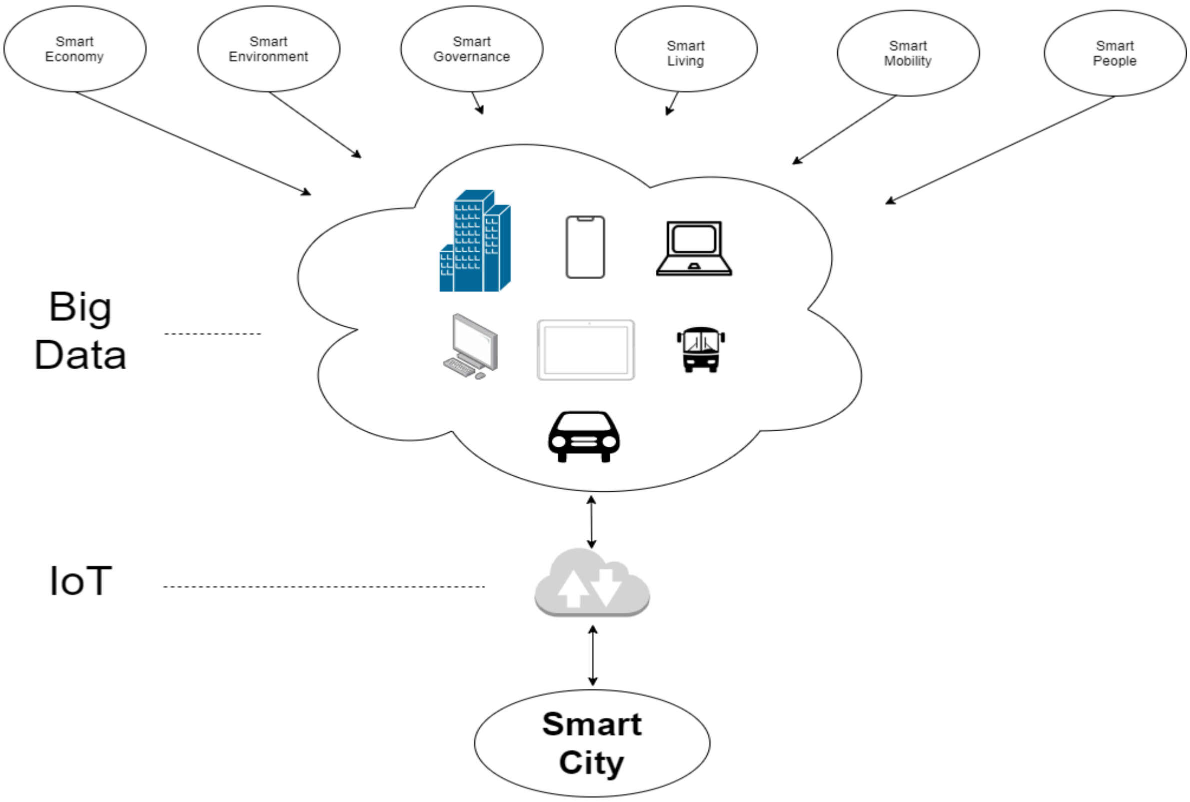

Figure 5: Block diagram of a Smart City. The six items at the top of this figure represent the key characteristics of a Smart City. All characteristics have one essential attribute together, they transmit and receive data via smartphones, tablet, computers, notebooks or even buildings (Smart Home) and vehicles. This data is called Big Data. The Internet of Things is the technological transmitting part of the data. Thus, all the collected data is transferred to departments within a Smart City, which then analyze and process the information to control actors afterwards or present the information to residents.

The presented block diagram illustrates the different dependencies within a Smart City and further shows how the city collects data. The Smart City itself communicates via the Internet of Things with a variety of smart devices, which belong to the residents.

Today’s cities do already have smart implementations, like animated advertisement posters or information displays at bus stops, but most of them are not interconnected to each other and they do not communicate wirelessly. Furthermore, they do not have the six key characteristics implemented. The Table below compares the cities of today with future Smart Cities and informs about the approach of intelligent cities to make the life of people who are living in a city better.

Table 2: Comparison of today‘s cities with future Smart Cities

Implementation	City of today	Smart City	Benefit / Saved Resources
Smart Water Network	No	Yes	Reduce water and energy consumption
Smart Buildings	No	Yes	Reduce energy consumption
Smart Shuttles	No	Yes	Reduce CO_2 emissions
Smart Education	No	Yes	Enable home schooling in virtual classes
Smart Sensors	No	Yes	Detect crimes and contact officials in real-time
Smart Applications	Yes	Yes	Reduce trash
Smart Homes	Yes	Yes	Reduce energy consumption

Smart Water Networks help to reduce the water and further the energy consumption by monitoring the amount of a household’s water. Regarding this information, every household only gets the needed amount of water (see Chapter 3.2).

Smart Buildings are able to communicate with other buildings. Embedded sensors in the building monitor the energy consumption and send this data via a cloud service to the energy provider. The provider can then control how much energy a building needs.

Smart Shuttles have no driver on board, they drive autonomously within a Smart City. These shuttles can calculate the most effective route through the city traffic with their on-board computer. All these Smart Shuttles drive with an electrical engine, so there are no exhaust fumes. Therefore, the CO_2 emission can be reduced.

Virtual classes can take place all around the globe and students from Austria can easily attend classes in Australia or the United States, for example. This can possibly improve the worldwide teaching system.

Smart Sensors, which are equipped at different spots within a Smart City can make life of residents safer by detecting crimes in real-time and contacting the police, for example, as fast as possible.

Smart Applications on the intelligent devices, like a smartwatch, can improve the waste management within a city. To serve as an example, bus tickets do not have to be printed on paper, they simply appear on the smartwatch or the smartphone.

Smart Homes reduce the energy consumption of a household by controlling different actors, such as LED lamps, in the most efficient way. Sensors detect that the sun is shining and automatically send this information to a hub which controls the specific actors afterwards.

Table 2 illustrates different applications within a Smart City to save and reduce the most important resources, but a lot of money has to be invested in the technology to develop and implement these applications.

4.1 Save resources before building a future Smart City

This subchapter describes several possibilities to save resources before a Smart City is built. Each key characteristics of a Smart City is different, so a variety of aspects need to be considered before developing technology or investing resources, such as time or money, in the realization of one. The technology of today helps developers as well as researchers to use tools which have not existed in the past. Here, four ways of realizing key characteristics of a Smart City are explained in detail. In order to keep this explanation clear, the author decided to focus just on one characteristic and tools to save resources: Smart Mobility. The four tools are the ones as follows:

- Tool 1: Miniature setup of autonomously driving shuttles
- Tool 2: Real setup of autonomously driving shuttles
- Tool 3: Integration of an Embedded System into existing shuttles
- Tool 4: Development of a computer simulation about autonomously driving shuttles

Compared to existing shuttles, autonomously driving ones do not have a driver in the cockpit anymore (see Chapter 3.6). Thus, there is an immense amount of data which has to be received and transmitted from and to other participants of the road traffic. In addition, necessary calculations about the next steps (e.g. overtaking, pick up, drop off, etc.) have to be done in real time.

Tool 1: Miniature setup of autonomously driving shuttles

A miniature setup of autonomously driving shuttles is one opportunity to save costs before implementing new technology into an existing city. This setup can be realized by different parties within a research unit or company (i.e. developers, craftsmen). Miniature setup means that the shuttles can be represented by robotic vehicles in different sizes on a desk, for example.

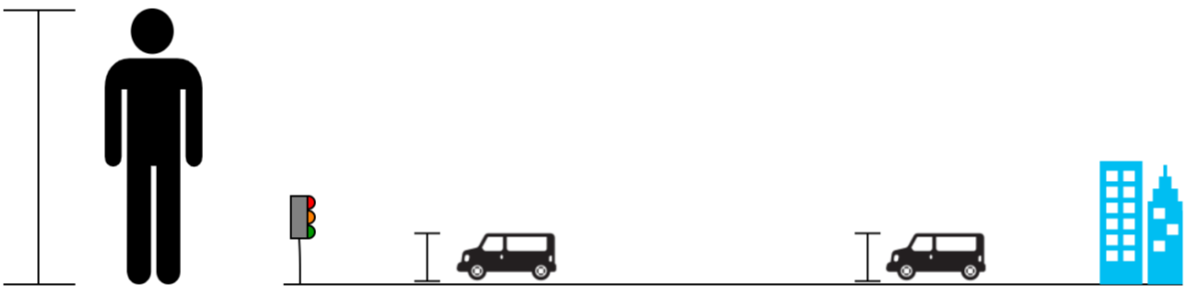

Figure 6: Miniature setup of autonomously driving shuttles. The above figure illustrates the different sizes of a human and the setup. The shuttles could drive independently of human interactions within the setup and important information about their behavior can be collected before tests in a real environment take place.

The setup itself can be connected to a computer and exchange data via the Internet of Things. The computer is a centrally hub in this application, which is located in a computation center within a Smart City. So, the necessary calculations of the shuttle's next move can either be done on the hub or within the shuttle's onboard computer.

This way of communication can be compared to the technology of sensors within a Smart City. Sensors cannot execute complex algorithms, they only provide computers with information. The information has to be sent to an external computer, which then analyzes and processes the collected information. Additionally, developing costs can be saved when doing the calculations on an external computer because embedding an onboard computer into every shuttle of the miniature setup can get very expensive later.

Tool 2: Real setup of autonomously driving shuttles

The second presented tool is about a real setup of autonomously driving shuttles. These shuttles drive independently of a driver on a real street. This tool is the most expensive one, because the shuttles must work and behave like other participants on the road do.

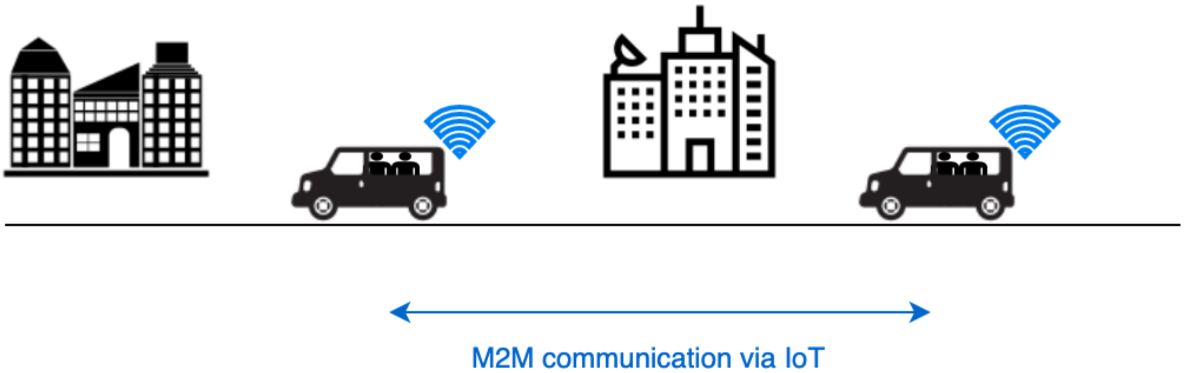

Figure 7: Real setup of autonomously driving shuttles within a city. Here, a Machine-to-Machine communication (M2M) via the Internet of Things takes place. Important information about where one shuttle is currently driving, the distance to other vehicles in the environment or the status of the road can be shared together.

The setup in Figure 7 illustrates that each of the autonomously driving shuttles must be connected to the IoT. In order to realize this requirement, each shuttle needs an onboard computer which collects the information of the shuttle's sensors (e.g. proximity sensor, speed sensor, light sensor, etc.). Compared to Tool 1 and Figure 6, a fault within the system of the vehicle can have dramatically consequences and thousands of testing hours have to be invested into the development in advance. Several questions could come up after an accident with a self-driving shuttle, including the question who is responsible for the fault.

Tool 3: Integration of an Embedded System into existing shuttles

Tool 3 presents the idea of equipping already existing busses or shuttles with a computer or an Embedded System. Compared to Tool 1 and Tool 2, these vehicles need to be connected to the Internet of Things as well. Sensors could be located on the road and within the shuttle to collect important information about the next move.

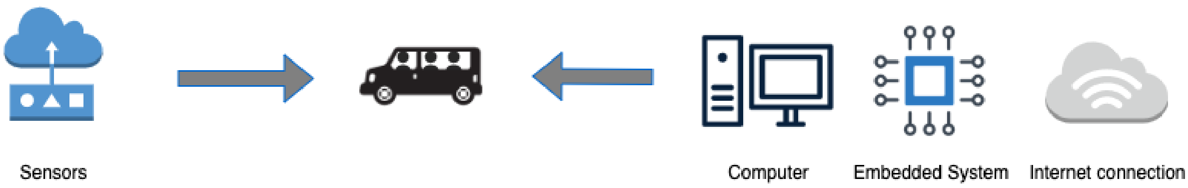

Figure 8: Integration of an Embedded System into an existing shuttle. This method will save costs because the vehicle is already built and the only necessary steps include the placement of sensors and a computer which controls the actors of the shuttle (i.e. wheels, brakes, lights, etc.). This computer can then be connected to the Internet in order to share the information which is provided by the sensors of the shuttle.

Tool 3 is a very interesting one because technology from the past is connected to technology of today. Vehicles were invented decades ago and the Internet of Things together with the computer are the technology of the future which has already begun. This means that self-driving cars or shuttles are not some type of Science-Fiction anymore. Tests with those vehicles do already take place nowadays.

Tool 4: Development of a computer simulation about autonomously driving shuttles

Tool 4 is the one where developing costs can be saved in advance. The main requirement of a computer simulation of self-driving shuttles within a Smart City is the processor power of the computer where the simulation is developed. Other important requirements can be defined in another process (see Chapter 2).

One idea of this Thesis was to implement a simulation of a Smart City key characteristic to illustrate how resources can be saved and the life of residents will be improved after building an intelligent city. Therefore, the author developed a Python simulation together with a smartphone application to give an example of how an IoT device can communicate wirelessly with the Smart City, in that specific case with autonomously driving shuttles. The following figure will give an overview of the simulation.

Figure 9: Smart Mobility simulation block diagram. People can easily communicate each shuttle, which is currently driving on the road via the Internet of Things. Data from the smartphone is collected via the installed smartphone application which is connected to the Internet as well.

Figure 6 shows that there is a bidirectional communication between the shuttle and the Internet of Things as well as between the people and the Internet of Things. Bidirectional communication means that a data flow is produced in both directions. Thus, data is received and transmitted from both participants of the communication.

A simulation of a Smart City characteristics is possible with the performance of today's computers. Thus, different other aspects of an intelligent city can be inspected and discussed in advance, like illustrated by the following ideas:

- Smart Economy
  - Simulate the improvements of paying with smart devices, such as smartphones or smartwatches

- Smart Environment
  - Simulate the energy consumption within a Smart City

- Smart Governance
  - Simulate the effect of a political decision in advance

- Smart Living
  - Simulate the detection of earthquakes or other natural catastrophes

- Smart Mobility
  - See Chapter 5 with the requirements of Chapter 2

- Smart People
  - Simulate the improvements when using intelligent devices

5 Implementation of a Smart Mobility simulation

A simulation for a Smart Mobility system within a Smart City as well as an application for the Apple operating system iOS was developed simultaneously to the creation of this thesis. This simulation fulfils the requirements which are defined in Chapter 2 and is separated in two version. The first version of the project is more abstract than version 2, where drawings of people and shuttle should give the user of the simulation a more realistic feeling. In this section, a closer look at both versions of the simulation and a developed smartphone app is taken.

5.1 Smart Mobility simulation – Version 1

Version 1 of this simulation was developed with focus on the functionality. This means, that rectangles represent shuttles, whereas circles represent passengers. The street is just a simple combination of polylines.

5.1.1 Introduction

The goal of this simulation was the development of one and even more autonomous shuttles with the software *Rhinoceros V6* and its plugin *Grasshopper* from Robert McNeel and Associates. The main purpose of *Rhinoceros V6* (abbr.: *Rhino*) is the creation of 3D models for architecture, but the plugin allows the user to write code in the programming language Python. Therefore, Grasshopper offers a specific rhinoscriptsyntax library which already has some very useful functions implemented to draw lines, draw rectangles or circles, etc. A very helpful documentation for this library can be found online [57].

First, the street, where the shuttles drive and the stops, where passengers are picked up and dropped off had to be drawn, as shown in Figure 10.

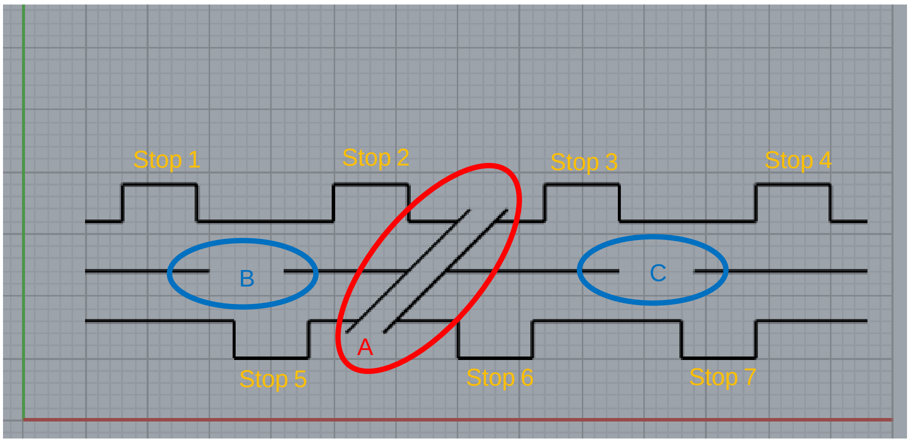

Figure 10: Street and stops for the autonomous shuttles. The red and green line illustrate the X and Y axis of a coordinate systems with the point (0/0) in the bottom left corner. Rhino works with vector points, what means with X-, Y- and Z-coordinates. Figure 5 illustrates only the bird view of the simulation, what is sufficient for the purpose presented here. The street has 7 stops (see Stop 1 to Stop 7 in orange) as well as a street separator (see **A** in red) to illustrate that the street is much longer, and the shuttles do not only drive within the range of these seven stops. **B** and **C** illustrate the so-called U turns of the street which are described in detail later in this thesis.

There could be several different shuttles drive on the same road, either in the left or right direction. The number of shuttles can be adapted each time the simulation is started, such as seen in the next figure.

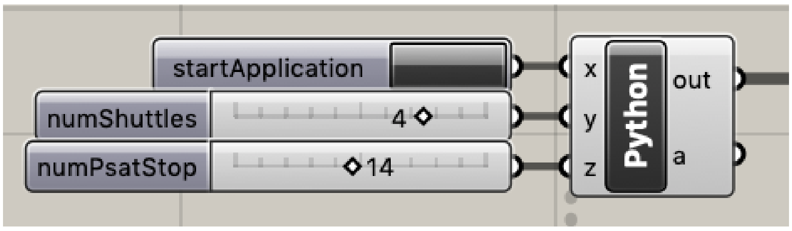

Figure 11: User input of the shuttle simulation. For better testing and modularity purposes the user can decide how many autonomous shuttles are driving on the road (i.e. numShuttles). Here a number between 1 to 5 shuttles was chosen from the author according to the limited number of stops (see Figure 5). In addition, the user can decide how many passengers are already waiting at the seven stops (i.e. numPsatStop). Therefore, another slider was implemented which represents a numerical value between 0 and 30. With the button 'startApplication' the user is able to start the simulation from the beginning with the chosen number of autonomous shuttle driving on the road and the number of passengers at the stops. The Python block illustrated here opens the Python editor when double clicking it. The inputs '**x**', '**y**' and '**z**' are necessary for the already described functionalities. The output 'out' can be connected to a control panel which serves as some sort of console window where print commands or error messages are displayed. The second output 'a' can be connected to other Python blocks or can be used for further a further processing mechanism (e.g. creating a point).

It is very important to mention that Python is a script language. To enable motion in the simulation (i.e. shuttle drives from A to B) the script must be called several times. Therefore, a timer is necessary which recalls the script every amount of time. Here, a time interval of 20 ms was chosen, what means that the Python script is executed every 20 ms from the beginning. The simulation is only running when the timer is enabled.

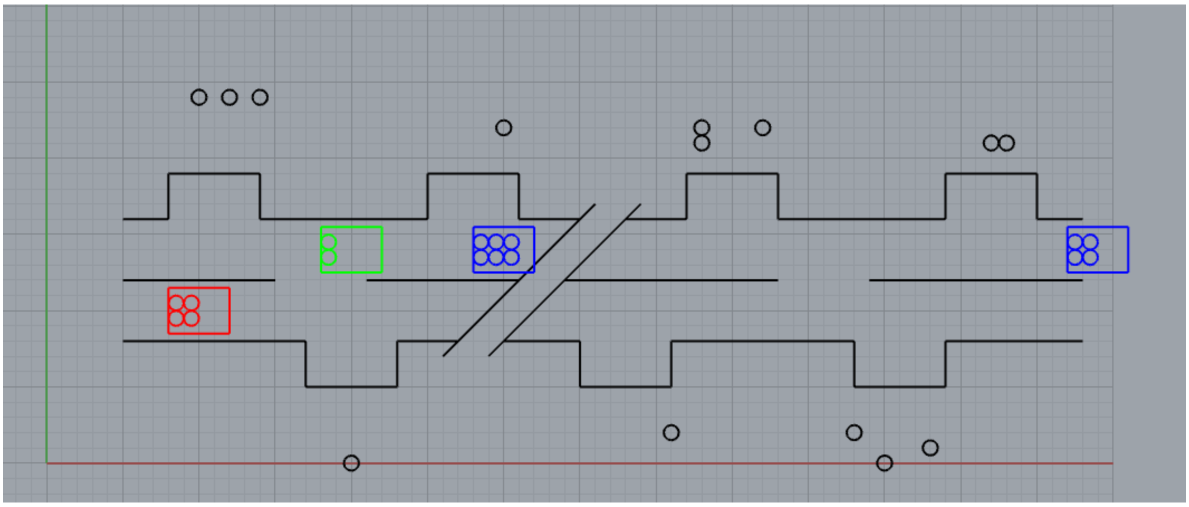

Figure 12: Start of the shuttle simulation. According to the settings of Figure 6 and after the press of the button 'startApplication' the corresponding number of shuttles and passengers is illustrated in the Rhino document. The rectangles represent the shuttles, where all the circles are the passengers.

As illustrated in the figure above, the shuttles are already colored after the simulation has started. The colors are described in in Figure 13.

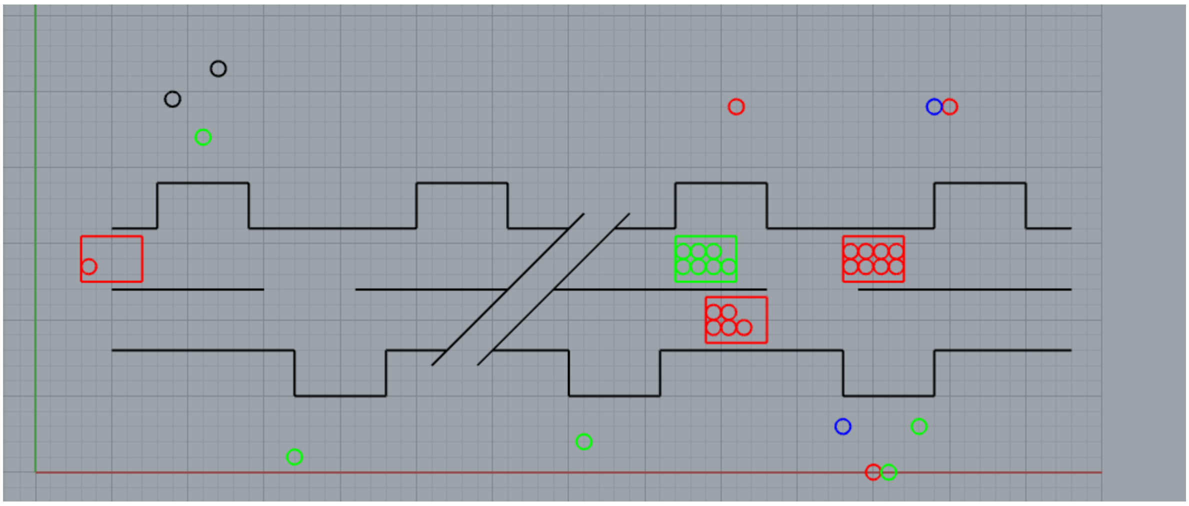

Figure 13: First seconds of simulation. After one of the autonomous shuttles passes a specific point of the street, the passengers at the stops get a color too. The color represents which shuttle picks up whom (e.g. the red shuttle collects the red colored circles). Within this simulation, the three basic colors red, green, blue as well as the color of the street (i.e. no color) are available. If the circles do not have a color anymore means that they were currently dropped off by the autonomous shuttle. They leave the station automatically. The decision of the shuttle's color is made randomly at this simulation.

In the real world, what means in Zenciti, the shuttle automatically detects where passengers want to go. The app ZBook is the reason why the shuttles know the passenger's current location and the wanted destination. The user (i.e. passenger) can decide that he or she wants to go to Stop 7. The shuttles then intercommunicate and decide how the passenger arrives at Stop 7 in the fastest way. Therefore, the shuttles calculate the fastest route on their own, what makes the use of a variety of different sensors necessary.

It is further possible to make use of the already mentioned U turns. Those turns are essential because a shuttle does not always have to complete the whole road in the same direction. That is why some passengers may have called the shuttle and want to go to a specific stop which is in the other direction. Therefore, the shuttle abridges at the U turn and is ways faster at the pickup stop than without such a turn. The U turn can also be used performed when some passengers want to leave the shuttle. In other words, this U turn can be used in different ways to make the ride of the shuttle as efficient as possible.

5.1.2 ‘Smart’ scenarios

This section describes which scenarios can make the autonomous shuttle ‘smart’ in detail, including the already described U turn. Therefore, some sort of case study is done below.

Shuttle calculates the fastest route

Smart autonomous shuttles can decide how to get to the final stop as fast as possible by calculating the route on their own. To achieve this case, the shuttle has to know where passengers, who are waiting at a stop, want to go by communicating with the ZBook application. So, the shuttle knows the fastest route even before the passenger or passengers have entered the shuttle’s entrance. The shuttle then sends a message to the smartphone and a message appears within the ZBook app, what tells which shuttle to be taken and when it arrives.

Shuttle makes a U turn when driving

Each autonomous shuttle in Zenciti can make a U turn when driving to the next stop, such as shown in the next figure.

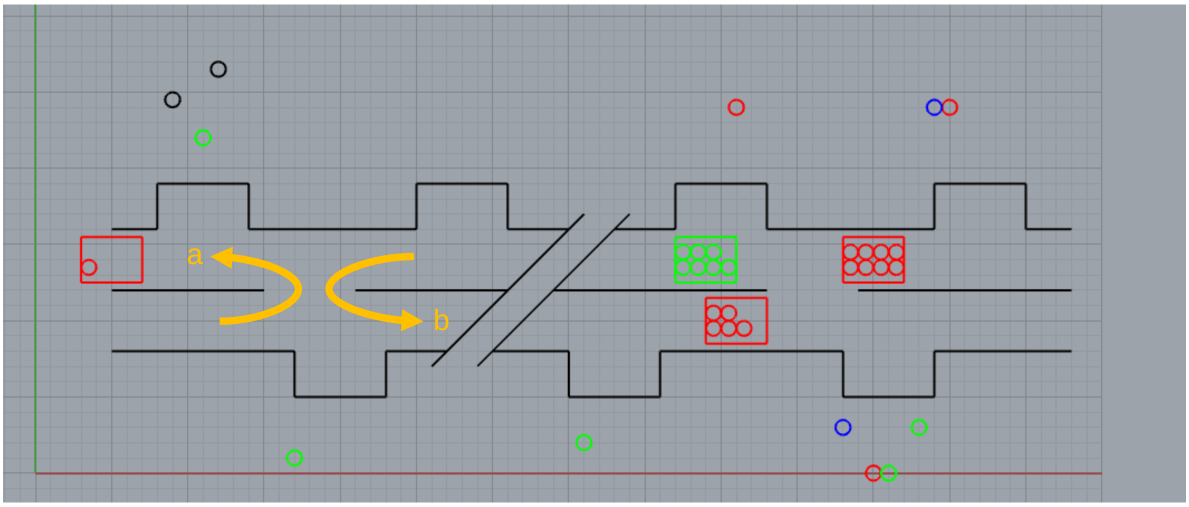

Figure 14: The U turn. The shuttle can make a U turn in both directions, as illustrated by **a** and **b**. To enable this function, the vehicle checks where the passenger(s), who are waiting at specific stops, want to go. When the shuttle then realizes that no one from the other stops enters this vehicle, it performs a U turn to abridge the route and therefore shorten the transportation time for everyone who has already entered the shuttle before.

It must be mentioned that this scenario can be performed at every U turn which is built into the road.

Shuttle picks up people who are going in the same direction

Another important scenario comes with the color of the shuttle, which can be red, green or blue. The color does not mean that the shuttle is going to the same stop every time because the final stop or even the next stop is calculated depending on the user input in the ZBook application. The app tells the passenger or passengers which shuttle has to be taken, what means either the red, green or blue one. The shuttle automatically knows where the other shuttles on the road are driving, enabling the function of picking up passengers who normally have to take, for example, the red shuttle. When the route to the next stop or even stops of the green shuttle is shorter, the vehicle is able to pick up the passengers waiting for the red shuttle and drives them to their wanted drop-off station. This makes the autonomous shuttles in Zenciti much more efficient compared to shuttles which are going to the same stops every day. So, the shuttles make some sort of intelligent clustering.

Shuttle communicates with the other shuttles on the road

In order to prevent a crash of the shuttle driving on the road, each shuttle has to know where the others are driving at the moment. Therefore, the author of this thesis has created an algorithm which automatically slows down shuttles to prevent an accident and to take care of the passengers in the vehicle. This algorithm is explained in Section 5.4.

Shuttle communicates with the road

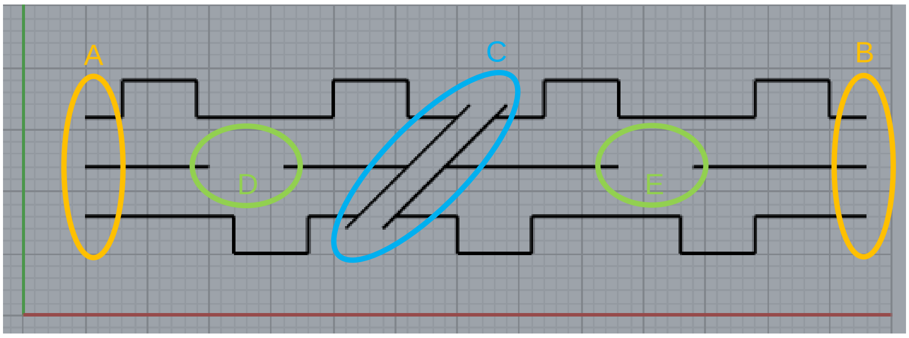

Figure 15: Communication of the shuttles with the road. There are several different points of the road, which the shuttles exactly have to know. For the purpose of this simulation, a road with seven stops is illustrated. When one shuttle passes the either the point **A** or **B**, the shuttle is deleted and a new one is created at a randomly chosen starting point. In addition, the shuttles have to know where the street separator (here **C**) is located. The shuttles do not just cross this point, the are hidden for a

moment and shown again when they have passed the street separator. Finally, the shuttle has to know where the position for a U turn is (see **D** and **E**).

Such as described in Figure 10, the shuttle must know at which point of the road it is driving. In this simulation, the x, y and z position of every shuttle is stored in a Python list, which contains elements of a class.

5.2 Smart Mobility simulation – Version 2

The second version of the developed Smart Mobility simulation is implemented with drawings of more realistic shuttles and passengers as well as a more realistic road. Therefore, the algorithm had to be adapted at some points in a way that the drawings work as placeholders and other drawings can be implemented too. This makes the simulation reusable by other developers.

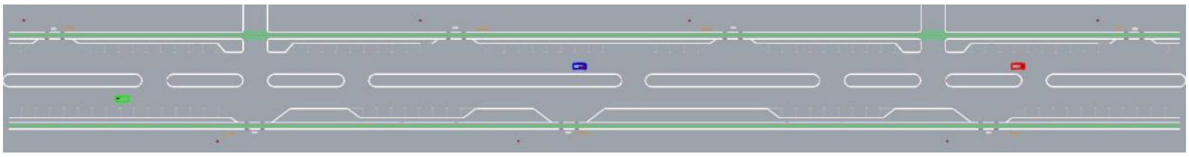

Figure 16: Final implementation of the simulation. The here presented figure shows three shuttles in the colours Red, Green and Blue driving on the road. Compared to version 1 of the simulation, there are seven stops again. 7 people are waiting at the stops for an arriving shuttle.

The author and developer extended the Grasshopper implementation of version 1 with the drawings and some specific elements.

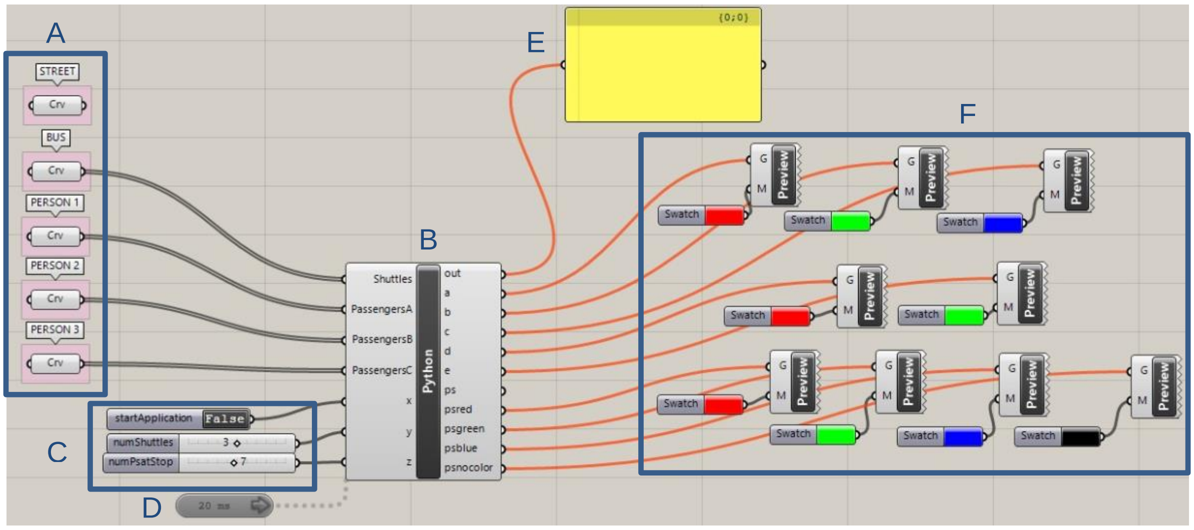

Figure 17: Grasshopper elements of version 2. Within this figure, the elements can be seen as separate ones, but together a working system is formed. Element **A** includes the drawings of the street, the shuttle (bus) and two waiting passengers as well as one passenger, which illustrates a person, who is sitting in the shuttle. Element **B** is the Python interface. A double-click opens the editor where the developer writes the code. Element **C** has the user input embedded. The user is able to distinguish between 1 and 5 shuttles, which are driving on the street. Furthermore, the user can choose between 0 and 15 passengers, who are waiting at the bus stops. The toggle startApplication starts the simulation. **D** is the 20 ms timer, which animates the simulation by calling the complete Python script (**B**) every 20 ms. **E** is an output panel, which is necessary for the developing process. Grasshopper does not support debugging of the code. So, the developer must use print commands which show their output at this panel. The elements in **F** are necessary to color the drawings of **A**.

Section C of Figure 17 illustrates that a user of the simulation can vary between how many shuttles are waiting at the stops and how many shuttles drive on the road.

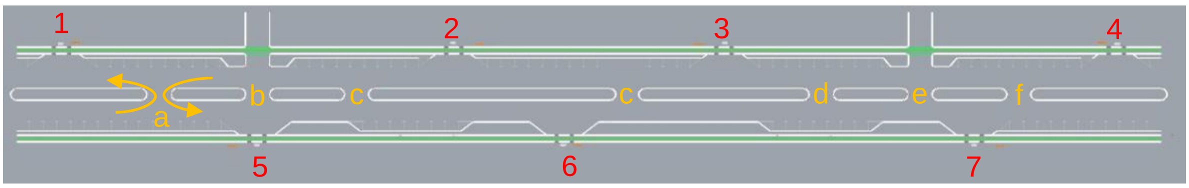

Figure 18: Drawing of the street. The street of version 2 is basically the same as the one in version 1. The only difference is the number of U turns in the second version. **1 to 7** illustrate the bus stops where the shuttles stop, and passengers can enter or leave the shuttle. The letters **a to f** represent the U turns, which can be used by shuttles driving either from right to the left or from left to the right. Therefore, the shuttle communicates with the passenger's smart device to decide if another shuttle is already going to the wanted drop-off station. This makes the shuttles the most efficient and saves time for the passengers.

The shuttles can calculate the shortest distance for every passenger and no driver is needed for that purpose. The U turns are used to save time while driving on the road.

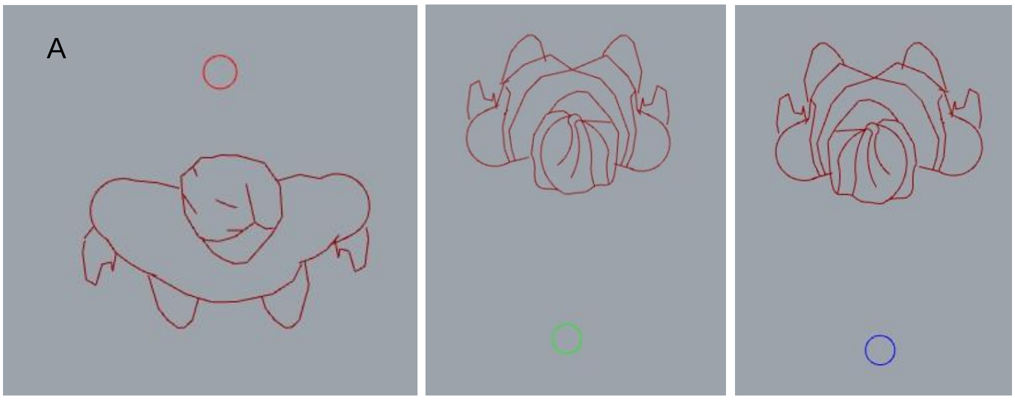

Figure 19: Passengers with colored circles. Here, the passengers, who are waiting at the bus stops are illustrated. The red, green or blue circles above the passengers represent the shuttle, which has to be entered by the passenger to get to the wanted drop-off station. So, the passenger on the left with the red circle in picture **A** must wait for the red shuttle to arrive at the stop. The circles are either on the upper or lower side of the passengers. A circle above the passenger indicates that the passenger is waiting at stop 1 to 4. A circle below the passenger shows the user of the simulation that the passenger is waiting at stop 5 to 7.

The simulation is animated with a 20 ms timer, which calls the written Python script in an interval of 20 milliseconds. This leads to a motion of the shuttle as well as the passengers, who are waiting at the stops. To illustrate that people do not stand still the whole time and to make the simulation more realistic compared to the real life, a movePassengers() function was implemented by the developer. The passengers are moving randomly up, down, left or right on the screen until the shuttle arrives at the stop.

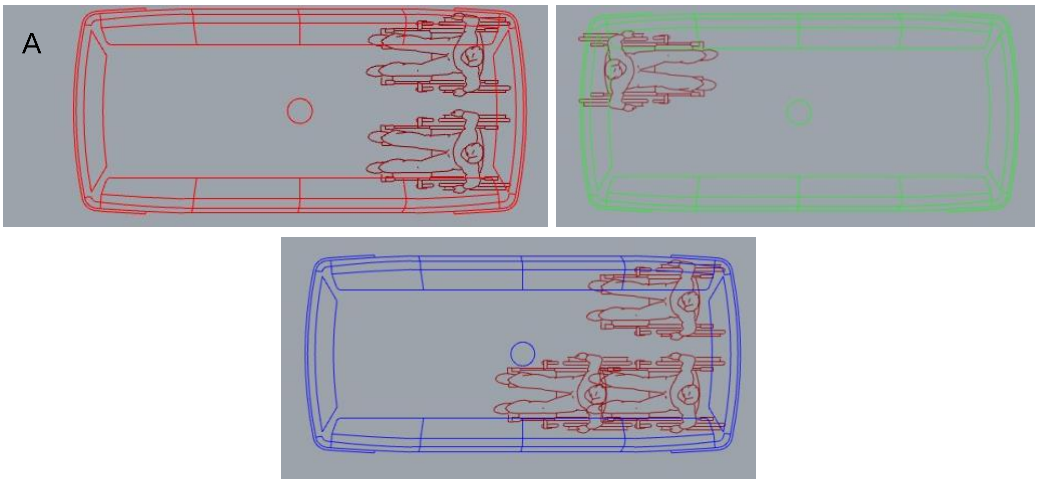

Figure 20: Passengers, who are driving in the shuttle. The passengers can take either a red, green or blue shuttle, what is illustrated with the above figures. Each shuttle has a maximum capacity of eight passengers and the shuttle knows how much people have already entered or left the vehicle. Picture A shows two passengers driving in direction 0 (right to left).

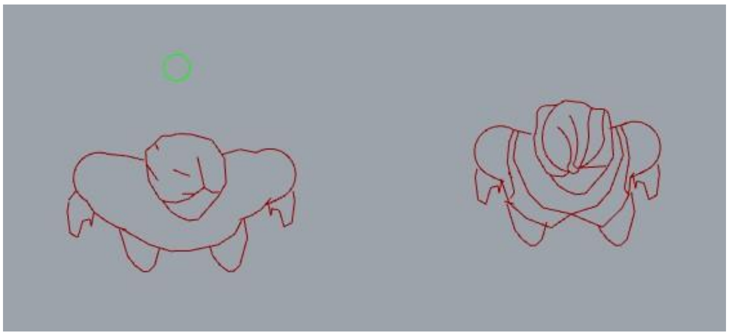

Figure 21: Two passengers looking at the stop. Here, the passenger on the left with the above green circle is waiting for a shuttle to arrive, while the person on the right has recently left a shuttle. This is indicated that the person on the right does not have a colored circle above.

Two functions have been implemented in the code to either pick-up or drop-off passengers at a specific stop. The pickupPassengers() function checks if the passenger has already assigned a shuttle and compares the color of the passenger's circle. So, a green shuttle picks up the passengers with a green circle and so on. The dropoffPassengers() function assigns the shuttle number 200 to the passenger, what indicates that he or she does not wait for a shuttle to arrive anymore.

Such as already described, the algorithms of both versions of the simulation are basically the same with the main difference that version 2 can be reused easier by another developer as part of a future work on this topic, to serve as an example. The shuttles, passengers and the street are represented by drawings, which are placeholders now, what means that other drawings (e.g. cars, trains, etc.) can also be used if there is a need to.

5.3 Algorithm

This subsection describes the finished algorithm of the Smart Mobility simulation in detail. Some graphics help shall help the reader to get a clearer overview of the code.

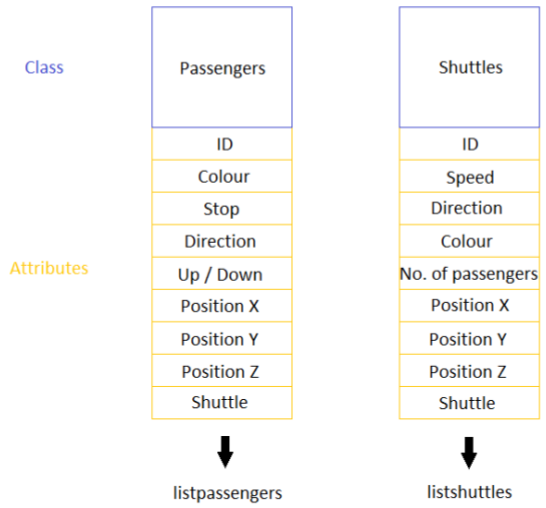

Figure 22: Class definition of simulation. The class ‘passengers’ shown in this figure helps creating some imaginary passengers, which are either waiting at the bus stops or already sitting in the shuttle. The second class ‘shuttles’ contains the information which is necessary to create a shuttle at a given position.

Both classes contain the three variables Position X, Position Y and Position Z. These variables indicate the current position of either a shuttle or a passenger and represent points in the X, Y and Z axis. Each created object (circle, rectangle, point, etc.) has an identification, which is called Guid in Rhino. This Guid is represented through the variable ID here and is unique during the script execution. The variable Colour indicates the object’s color (red, green or blue). There is a difference between the shuttle variable of both classes. In the Passengers class, the shuttle variable states if the passenger is waiting for a shuttle at a stop (Shuttle = 100), if the passenger leaves the bus stop (shuttle = 200) or if the passenger sits in the shuttle (Shuttle = 0, 1, 2, 3, 4, representing every shuttle). In the class Shuttles, the shuttle variable is a number between 0 and the number of shuttles driving on the road. This is necessary because the objects of both classes are created as a part of two lists, one for the passengers and one for the shuttles. So, the first shuttle has the number Shuttle = 0 and is the first element of the shuttles list, simultaneously, to serve as an example. Furthermore, both classes contain a variable direction, which is 0 if the shuttle drives from left to right and 1 the other way. In the Passengers class the direction variable is necessary for the movement of the passengers at the bus stop, what creates a more realistic simulation.

In addition, the variable up in the Passengers class enables the up and down movement of the imaginary passengers at each stop.

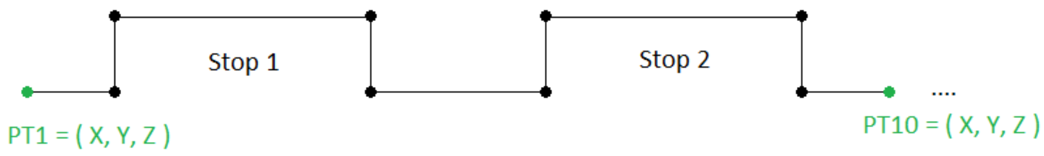

Figure 23: Creation of the street for the simulation. To make the simulation more realistic, a street was created, which consists of multiple lines (see Figure 5) containing a start as well as an end point. Each point has a X, Y and Z value. The function 'rs.AddPolyline(…)' then creates a line with the coordinates of the given points.

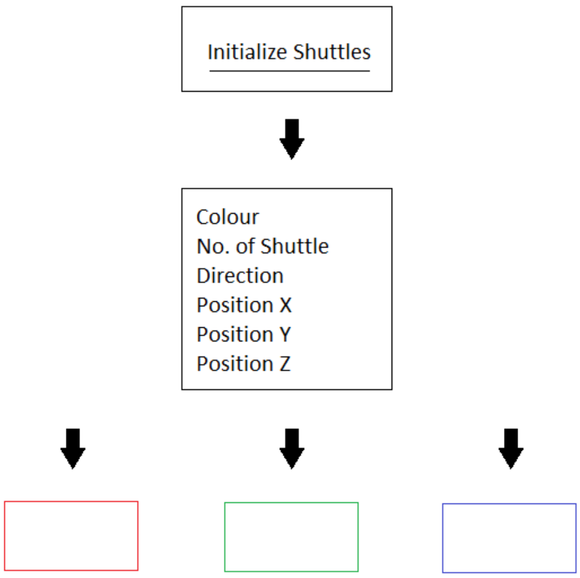

Figure 24: Function to create an autonomous shuttle. An autonomously driving shuttle is generated each time this function is called. The function returns an adjusted list, which contains objects of the class 'shuttles' (see Figure 11).

First, a rectangle of the size 4.0 x 3.0 units is generated, returning the already described Guid of the object. The color of this rectangle depends on the parameter colour, which is passed through when the function is called. The function 'rs.ObjectColor(…)' is able to change the color of an object according to the passed parameters, which represent the RGB colors. So, the call 'rs.ObjectColor(listshuttles[numsh].id, (255,0,0))' changes the color of the previously created rectangle to Red. After the creation, the shuttle (i.e. rectangle) is automatically moved to the correct position with the function 'rs.MoveObject(…)'. It is important to mention that every newly created object is located at the point (X = 0, Y = 0, Z = 0) without the 'rs.MoveObject(…)' function.

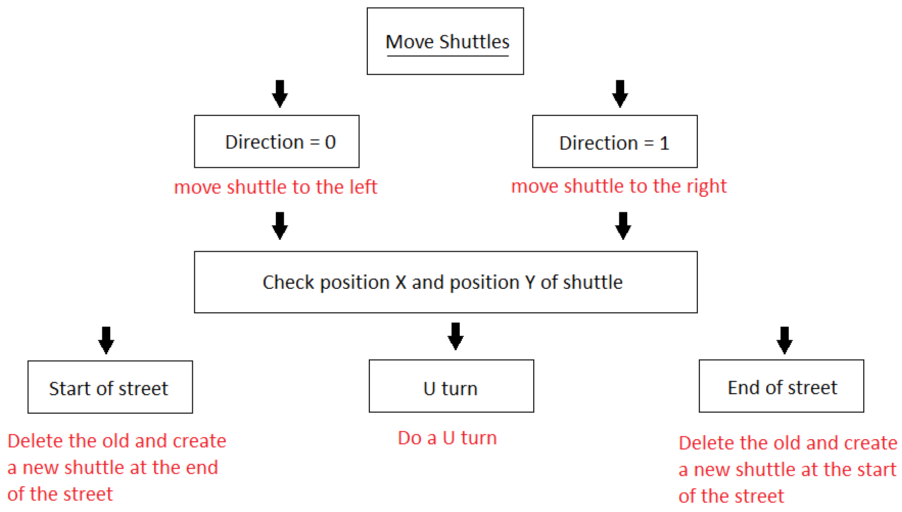

Figure 25: Function to animate the shuttles driving on the road. The autonomous shuttle is moved to a new position every time this function is called, depending on the direction, the current position and the speed.

The algorithm for both directions is basically the same, but some coordinates are different, and the speed is added to the current position of the shuttle when it drives from right to left, where the speed is subtracted in the other direction.

The function 'moveShuttles(…)' first checks if the shuttle has to drive to the left (direction = 0) or to the right (direction = 1). The speed is added or subtracted to the current position of both the shuttle as well as the passengers sitting in the shuttle, afterwards. When the vehicle reaches the position 0.0, it is deleted and a new shuttle is automatically generated, enabling that the simulation runs infinitely until the user stops it by disabling the timer which executes the Python script every 20 ms. For the performance of a U turn, the shuttle first checks which shuttle the passengers at the next stops are waiting for. Therefore, the function 'checkColors(…)' was created, which increments a counter each time a passengers waits for a red, a green, or blue shuttle.

The implementation of the 'moveShuttles(…)' function is the algorithm where the shuttle automatically calculates the fastest and most efficient way on its own.

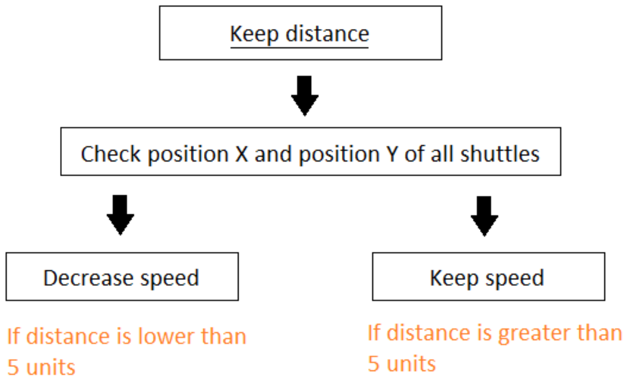

Figure 26: Keep a distance to the other shuttles on the road. This function is essential for the simulation as it automatically checks the position of the other shuttles driving on the road at the moment and reduces the speed of the shuttle behind until a distance of 5 units is reached.

This function starts with a separation of shuttles which are driving in one or another direction. The hence created list is then sorted regarding the x position of every shuttle. In a next step, the speed of the shuttle which is driving behind another shuttle is reduced until a distance of 5 units is reached again. As a last step, the list is linked together again and returned to the calling function in the script.

When two shuttles have a correct distance between each other, the speed is kept or increased until it reaches the initial value.

5.4 Smartphone application

The author’s aim of the smartphone application, which is presented here, is to enable the function to call an autonomously driving shuttle. In order to create such an app, the program *XCode* from Apple Inc. together with the programming language Swift is used.

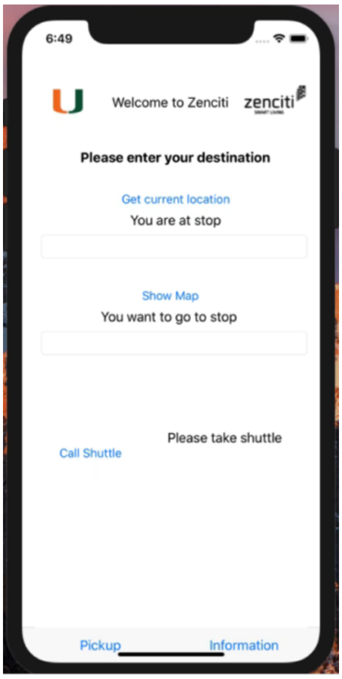

Figure 27: Start window of smartphone app. At the start window, the user is able to get his or her current location (i.e. button ‚Get current location‘), select the stop where he or she wants to go (i.e. text field below ‚You want to go to stop‘) and to call a shuttle (i.e. button ‚Call Shuttle‘). In addition, the information when the called shuttle arrives can be shown (i.e. button ‚Information‘). When pressing the button ‚Pickup‘, the start window is shown again.

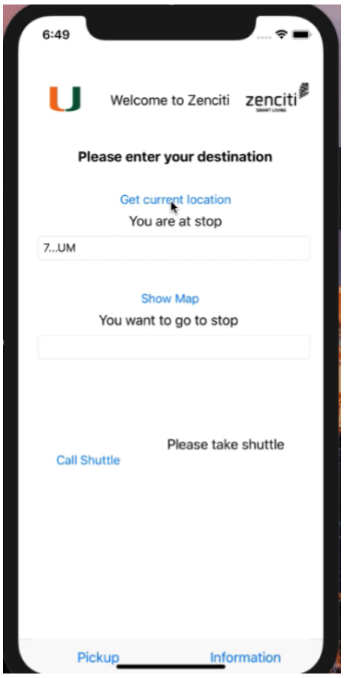

Figure 28: Get current location. When a user presses the button ‚Get current location‘, the algorithms provides a randomly created number between 1 and 7 representing the stop where the user is waiting for a shuttle at the moment. This function can be extended so that the smartphone uses GPS data to automatically detect the nearest stop.

Figure 29: Map of Zenciti. In addition to the function to a passenger’s current location, a user is able to show the map and to zoom in or out. Each yellow dot is representing a stop. This function can be extended so that the user can press one of the yellow dots and the algorithm automatically calculates the fastest route.

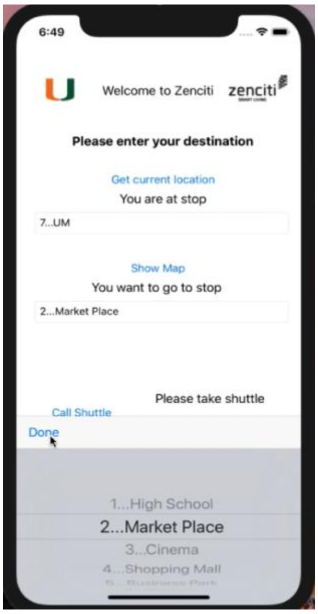

Figure 30: Choose the stop. If a user taps inside the text field of the stop to which he or she wants to go, a dropdown menu opens at the bottom of the smartphone screen. Here, the user can choose, where the shuttle should drop off him or her. A press of the button ‚Done‘ takes this setting.

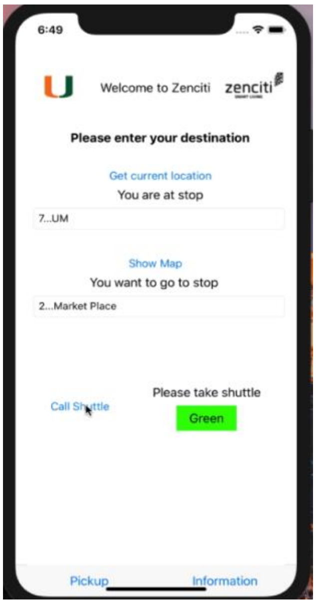

Figure 31: Call a shuttle. When both, the current location and the drop off station are chosen, a shuttle can be called with a press of ‚Call Shuttle‘. Afterwards, the algorithm randomly presents either a red, green or blue rectangle indicating which shuttle the user has to take. This function can be extended so that the application automatically communicates with each shuttle on the road and calculates which one must be taken to get to the drop off station in the fastest way.

Figure 32: A timer which represents the waiting time until a shuttle arrives. If a user presses the button ‚Information', a timer appears which indicates how long the passenger must wait for the arrival of the called shuttle.

Figure 33: Arrival of the called shuttle. When the timer stops at 0 minutes and 0 seconds, a message on the screen appears which gives somebody the notice that the called shuttle has arrived.

This smartphone app is kept very simple to make the call of a shuttle as easy as possible. The aim of this project was to develop a video which shows the interconnection between both, the smartphone app and the simulation with Rhino.

To give the reader a more detailed overview of the code, some parts are described in the following part of this thesis.

The program XCode provides a GUI with several different features to create a running application with some clicks. The developer can distinguish on which device, so either a tablet or a smartphone, the application shall be simulated.

Compared to the programming language C, each program in Swift starts with the declaration of variables.

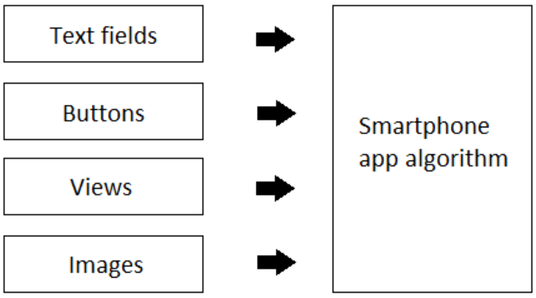

Figure 34: Connect elements to code. This figure is a simple illustration of the first step, which has to be done after the implementation of elements, such as text fields or images. Within XCode, a user can simply draw a connection between the element on the left to the code on the right and the software automatically detects the basic attributes of it.

Different text fields are necessary to output some text on the screen of the smartphone. Buttons require the input of a user and specific events are called, afterwards. The views in this application were necessary to switch between different screens on the smartphone. The main view shows the text fields and buttons to enable the function of calling a shuttle. Another view shows the map and the third view implements the waiting time until the next shuttle arrives at a specific stop. The developer used different images (e.g. header and geographical map) to make the application more realistic.

When the button ‘Show Map’ is pressed, a geographical map of Zenciti appears on the device screen. Such as already described, three views **viewPickup**, **viewInformation** and **viewMap** have been added to the code, therefore. A button press simply hides two of three views and the map automatically appears afterwards.

If all the settings are entered by the smartphone user and the button ‘Call Shuttle’ is pressed, the application automatically chooses a shuttle which picks up the passenger(s). Such as illustrated in Figure 20, a label appears that shows the colour of the shuttle which the passenger must take to get to the wanted location.

For testing purposes, the shuttle is again chosen with the use a random number generator and set to number ‘2’ to indicate that a green shuttle is arriving. Furthermore, the press of the button ‘Call Shuttle’ starts a timer, which invokes a function **timerAction** every second. In the function **timerAction**, two text fields show the actual value of minutes and seconds that have to pass until the called shuttle arrives. When the time has passed and the shuttle arrives, a sentence on the screen appears, which informs the app user with the sentence ‘Your shuttle has arrived!’.

6 Discussion

A Smart City has six different key characteristics (see Chapter 3.1). As part of one of these characteristics a simulation as well as a smartphone application was developed in order to make this topic not only conceptual. This simulation is a good example of a use-case which can be implemented in a real Smart City one day.

FR1 was fulfilled successfully because a simulation, where passengers can be picked up and dropped off at certain stops, was developed. In addition, the shuttles are intelligent enough to calculate the most efficient route on their own. Thus, FR2 is fulfilled as well. One negative point is that the shuttles do not always work correct in version 2 (see Chapter 5.2) of the simulation. They do not recognize that some passengers are waiting at other stops and do a U turn instead, what leads to driving in a circular path sometimes. Noteworthy is, that this problem does not exist in version 1 (see Chapter 5.1).

Figure 6 and Figure 17.C illustrate that FR3 has been successfully implemented too. Users of the simulation can vary the inputs of the simulation and decide how many shuttles drive on the road as well as how many passengers are waiting at the bus stops. This procedure works well in both versions of the simulation. The only difference is that the action which triggers the start of the simulation is a button in version 1 and a Boolean Toggle in version 2. Every time the button in version 1 is pressed, the simulation starts from the beginning, whereas version 2 runs as long as the Boolean Toggle is in the True state.

FR4 is successfully implemented in both versions. In version 1, the Red, Green and Blue shuttles are represented by rectangles. In version 2, the shuttles are represented through more realistic drawings of busses without a driver (i.e. autonomously driving shuttles).

Both versions were implemented in the CAD software Rhino with its plugin Grasshopper and the simulation is ready to be used in classes for teaching purposes in its current state. Thus, NFR1 and NFR2 have successfully been realized.

The last non-functional requirement NFR3 was implemented with use the use of Apple's development software XCode. A smartphone application, where the process of calling a shuttle is shown, was developed (see Chapter 5.4).

During the development process of the Smart Mobility simulation, the author had to deal with different problems. Especially version 2 needed much more computational resources than expected. It is highly recommended to use a PC with enough random access memory (RAM) as well as an up-to-date graphics card. Regarding the timer to move the shuttles from one point to another, the Python script had to be recalled every 20 ms (see Chapter 5.1.1). This leaded to performance issues sometimes and the PC worked with a high percentage of workload. Thus, version 2 did not run fluently at some points.

The described algorithm of Chapter 5 represents a simulation for Smart Mobility, which is a key characteristic of a Smart City. Data from smartphones are wirelessly sent to the shuttle, which calculates the most effective transportation route for the passenger, afterwards.

The fact that the use of smartphones is said to be growing even more in the near future [50] supports the author's statement that this intelligent device provides the best platform to collect data for Smart Mobility. When a user connects to a network (Wi-Fi or cellular) and further to the internet by using specific applications with his or her smartphone, massive amounts of data are automatically collected (e.g. user location via GPS). A Smart Mobility app can be programmed in a way that users automatically get the location of the nearest stop of autonomously driving shuttles within a Smart City (see author's project described in Section 5). This collected data comes as raw data to the end device and can be processed to make it readable by humans then. The data can be made interpretable by humans, but it must not because the devices can already communicate with other devices (i.e. Internet of Things), what means that the collected data can be automatically forwarded. The benefit of this technology is the fast processing time of those devices which communicate within the IoT, what makes an application faster, even for the end user (e.g. smartphone user).

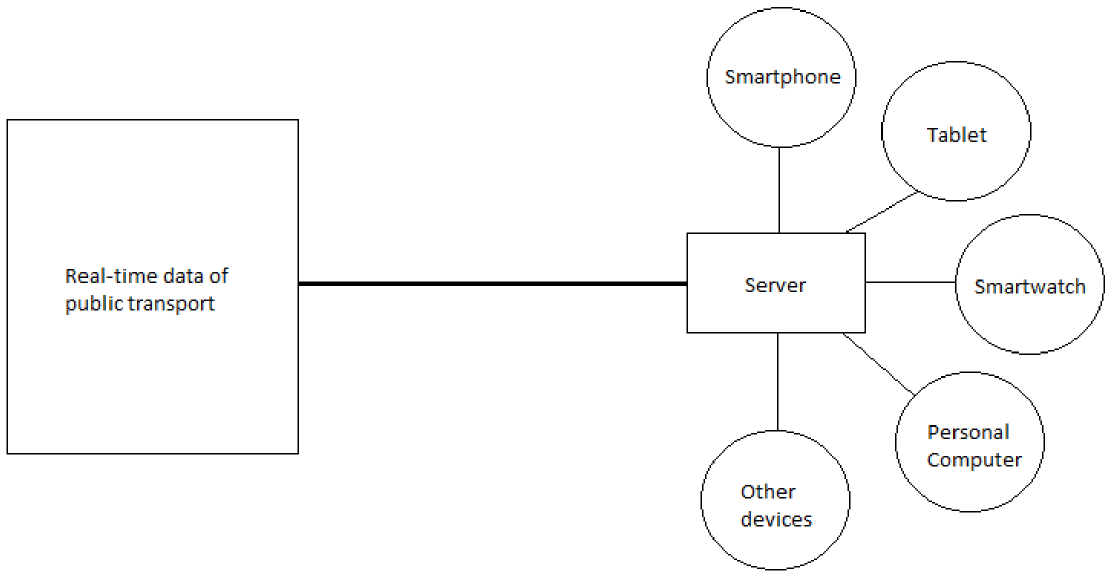

Figure 35: Example of data collection and processing. One example where data is collected from different sensors (i.e. location of public means of transport) is shown here. The data is collected in real-time and send to centrally located servers as a next step. Those servers interpret the collected information and send it to end user devices, such as smartphones, tablets, etc. As part of this process, the collected raw data information is processed, and the end device makes it readable for humans (for example by presenting it on the screen within an app).

This is just one of thousands or even millions of examples where data (i.e. digital information) is collected and further processed. Within a Smart City, this process goes in one hand with the placement of sensors in multiple different locations. These sensors are essential for the collection of data which is then sent to servers or hubs via different network architectures. There, the data is processed and forwarded to end user devices, such as the smart shuttles of Section 5 or even smart displays which are located at each pick-up or drop-off stop which are served by the shuttles.

The above-mentioned displays are already located in some cities nowadays, like described in [51].

This Master's Thesis informs about the current research on Smart Cities and shows how self-driving shuttles could help to make a city, particularly, the life of residents, safer and environmentally friendlier. It is just a simulation that shows what is possible in the near future. Collecting data (Big Data), distributing the information wirelessly (IoT) and controlling actors (autonomously driving shuttles) afterwards could be realized within a Smart City one day. Several examples illustrate the benefits of making today's cities 'smart'. The use of the described autonomously driving shuttles is one of it:

Reduce the CO2 emissions by taking advantage of autonomously driving shuttles

Chapter 5 of the here presented thesis describes a simulation of autonomously driving shuttles in a Smart City. These shuttles can reduce the CO2 emissions dramatically according to their prevention of using a fuel-operated motor and switching to a fully electrical engine instead. In addition, these vehicles operate in the most effective way by calculating the fastest transportation route for passengers. Factors, such as the traffic density play an important role here.

7 Conclusion

Chapter 7 includes Table 3 and Table 4 which highlight the benefits of a Smart City and the presented Smart Mobility simulation.

Table 3: Benefits of a Smart City to the six key characteristics.

#	Smart City Application	Implementation	Description
1	Smart Environment	Smart City	Sensors collect data to inform about possible upcoming events (e.g. earthquake) – see Section 1.4.1
2	Smart Environment	Smart City	Sensors help to reduce a household's water consumption by using a Smart Water Network – see Section 3.2
3	Smart Environment	Smart City	IoT devices communicate together and provide useful information to city officials – see Section 1.4.3
4	Smart Living	Smart City	Sensors help to improve security by informing official departments (e.g. police) about unusual events
5	Smart Mobility	Smart City	Smart shuttles can transport passengers autonomously
6	Smart People	Smart City	Applications, which run on intelligent devices (e.g. smartphones, tablets, etc.) help doing daily activities, such as working, doing sports or shopping

Table 3 explains four of the six key characteristics with the help of some examples. Sensors have to extend the infrastructure within a Smart City to control specific actors after the collected information has been processed. Those sensors can help to detect natural catastrophes, like earthquakes, in advance. Other sensors could help to save resources, like energy or water, whereas sensors to detect crimes can contact official departments in real time about unusual events. The Internet of Things plays a key role in the construction of a Smart City because the collected sensor information must be sent to a centrally located computer and to the actors afterwards.

Smart devices, such as smartphones or wearables do and further will improve the daily life of citizens by sharing important information about traffic congestion, the weather or the personal health.

In order to help realizing the construction of a future Smart City, the author presented a method of saving resources before the construction process starts. Doing a computer simulation is a cheap and easy method to collect information in an early phase of the construction process. A simulation about autonomously driving shuttles, representing the Smart Mobility key characteristic, was created. The algorithm of this simulation could help developers of 'real' smart shuttles in the future.

Table 4: Benefits of the Smart Mobility simulation.

#	Smart City Application	Implementation	Description
1	Smart Environment	Research project	Autonomously driving shuttles reduce the CO_2 emissions by calculating the most effective route
2	Smart Living	Research project	Autonomously driving shuttles reduce the stress of commuting during the week
3	Smart Living	Research project	Autonomously driving shuttles reduce the accidents caused by humans on the road
4	Smart Mobility	Research project	Autonomously driving shuttles use the infrastructure of a Smart City
5	Smart Mobility	Research project	Smartphone app simplifies the process of calling an autonomous shuttle

Table 4 is a summary of the resources that can be saved by developing and constructing autonomously driving shuttles for the purpose of transporting passengers in a Smart City.

Reducing CO2 emissions or helping to reduce accidents on the road are some of the benefits of autonomously driving shuttles. A smartphone application could improve the process of calling a shuttle to a specific bus stop and further inform the shuttle where a passenger wants to go.

Autonomously driving shuttles are in contact with the infrastructure of a Smart City constantly (e.g. traffic lights). They communicate with other shuttles on the road and do not need the input of a driver in the cockpit anymore.

8 Future work

The developed research project (Smart Mobility Simulation and Smartphone application – see Chapter 5) which was developed from the author of the here presented thesis is working well to illustrate how the use and process of Big Data can be combined with the Internet of Things technology within a Smart City:

- Big Data: User data is collected with the smartphone (i.e. current location, wanted destination)
- Internet of Things: Autonomously driving shuttles communicate together about their current position on the road
- Smart City: The infrastructure within a Smart City (sensors, which provide information about traffic congestion or GPS sensors) helps to calculate the most effective route of the shuttle

The project can still be extended to make the shuttles ‘smarter’ and to make use of a Smart City’s technology. Within Table 5, possible improvements are presented to the reader.

Table 5: Possible improvements of the presented research project.

#	Smart City Application	Description
1	Smart Mobility	Use real-time GPS data of autonomously driving shuttles to shorten the transport time for the passengers
2	Smart Mobility	Use real-time GPS data to automatically detect the nearest pick-up / drop-off station
3	Smart Mobility	Use real-time GPS data to improve the location detection on the map within the smartphone application (see Figure 18)
4	Smart People	Evaluate the collected data of the smartphone app (e.g. collect information about travelling times to suggest the most effective route)
5	Smart People	Evaluate the collected data of the smartphone app (e.g. automatically call a shuttle when arriving at the stop to shorten the waiting time)

Bibliography

[1] Rassia S. Th., Pardalos P. M., Suzuki L. R., *Smart City Networks: Through the Internet of Things*, Chapter: *Smart Cities IoT: Enablers and Technology Road Map*, pp. 167-190, Cham, Springer, 2017

[2] Rassia S. Th., Pardalos P. M., Dameri R. P., Ricciardi F., *Smart City Networks: Through the Internet of Things*, Chapter: *Leveraging Smart City Projects for Benefitting Citizens: The Role of ICTs*, pp. 111-128, Cham, Springer, 2017

[3] Sun M., Zhang J., *Research on the application of block chain big data platform in the construction of new smart city for low carbon emission and green environment*, Journal: *Computer Communications*, Volume 149, pp. 332-342, Elsevier, 2019

[4] Osman A. M. S., *A novel big data analytics framework for smart cities*, Journal: *Future Generation Computer Systems*, Volume 91, pp. 620-633, Elsevier, 2018

[5] Costa D. G., de Oliveira F. P., *A prioritization approach for optimization of multiple concurrent sensing applications in smart cities*, Journal: *Future Generation Computer Systems*, Volume 108, pp. 228-243, Elsevier, 2020

[6] Rathore M. M., Paul A., Hong W.-H., Seo H. C., Awan I., Saeed S., *Exploiting IoT and big data analytics: Defining Smart Digital City using real-time urban data*, Journal: *Sustainable Cities and Society*, Volume 40, pp. 600-610, Elsevier, 2018

[7] Goad D., Collins A. T., Gal U., *Privacy and the Internet of Things – An experiment in discrete choice*, Journal: *Information & Management*, 103292, Elsevier, 2020

[8] Nord J. H., Koohang A., Paliszkiewicz J., *The Internet of Things: Review and theoretical framework*, Journal: *Expert Systems with Applications*, Volume 133, pp. 97-108, Elsevier, 2019

[9] Kamran M., Khan H. U., Nisar W., Farooq M., Rehman S.-U., *Blockchain and Internet of Things: A bibliometric study*, Journal: *Computers and Electrical Engineering*, Volume 81, 106525, Elsevier, 2019

[10] Song H., Srinivasan R., Sookoor T., Jeschke S., *Smart Cities: Foundations, Principles, and Applications*, Chapter: *Internet of Things (IoT)*, pp. 104, New Jersey, Wiley, 2017

[11] Etezadzadeh C., *Smart City – Stadt der Zukunft? Die Smart City 2.0 als lebenswerte Stadt und Zukunftsmarkt*, Wiesbaden, Springer Vieweg, Springer, 2015

[12] Etezadzadeh C., *Smart City – Made in Germany Die Smart-City-Bewegung als Treiber einer gesellschaftlichen Transformation*, Wiesbaden, Springer Vieweg, Springer, 2020

[13] Mardacany E., *Smart Cities Characteristics: Importance of Built Environment Components*, Journal: *IET Conference of Future Intelligent Cities*, London, IEEE, 2014

[14] Larios V. M., Gomez L., Mora O. B., Maciel R., Villanueva-Rosales N., *Living Labs for Smart Cities: A use case in Guadalajara City to foster innovation and develop citizen-centered solutions*, Journal: 2*016 IEEE International Smart Cities Conference (ISC2)*, IEEE, 2016

[15] Ahlgren B., Hidell M., Ngai E. C.-H., *Internet of Things for Smart Cities: Interoperability and Open Data*, Journal: *IEEE Internet Computing*, Volume 20, Issue 6, pp. 52-56, IEEE, 2016

[16] McClellan S., Jimenez J. A., Koutitas G., Lehr T., *Smart cities: Applications, technologies, standards and driving factors*, Chapter 1, pp. 5, Springer, 2018

[17] McClellan S., Jimenez J. A., Koutitas G., Zou Y., Jolly B., Li R., Wang M., Kaur R., *Smart cities: Applications, technologies, standards and driving factors*, Chapter 5, pp. 75, Springer, 2018

[18] Al Nuaimi E., Al Neyadi H., Mohamed N., Al-Jaroodi J., *Applications of big data to smart cities*, Journal: *Journal of Internet Services and Applications*, pp. 3-4, Springer, 2015

[19] Etezadzadeh Ch., *Smart City – Stadt der Zukunft? Die Smart City 2.0 als lebenswerte Stadt und Zukunftsmarkt*, Chapter: *Funktionsfähigkeit*, pp. 11-17, Springer Vieweg, Springer, Wiesbaden, 2015

[20] Etezadzadeh Ch., *Smart City – Stadt der Zukunft? Die Smart City 2.0 als lebenswerte Stadt und Zukunftsmarkt*, Chapter: *Digitalisierung*, pp. 39-47, Springer Vieweg, Springer, Wiesbaden, 2015

[21] Capital – So digital ist die Welt 2020 [Online], Available: https://www.capital.de/leben/so-digital-ist-die-welt-2020, Last access: September 13, 2020

[22] Dameri R. P., Rosenthal-Sabroux C., *Smart City – How to Create Public and Economic Value with High Technology in Urban Space*, Chapter: *Smart City and Value Creation*, pp. 1-12, Springer International Publishing Switzerland, Springer, Cham, Heidelberg, New York, Dordrecht, London, 2014

[23] Jawhar I., Mohamed N., Al-Jaroodi J., *Networking and Communication for Smart City Systems*, Journal: *2017 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, IEEE, 2017

[24] Hernandez L., Baladron C., Aguiar J. M., Carro B., Sanchez-Esguevillas A. J., Llore J., Massana J., *A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings*, Journal: *IEEE Communications Surveys & Tutorials*, Volume 16, Issue 3, IEEE, 2014

[25] Bibri S. E., *The anatomy of the data-driven smart sustainable city: instrumentation, datafication, computerization and related applications*, Journal: *Journal of Big Data*, Article number: 59, Springer Open, 2019

[26] Meier A., Portmann E., *Smart City – Strategie, Governance und Projekte*, Chapter: *LoRa-Netzwerke als Basis-Infrastruktur für Smart Cities*, pp. 8-9, Springer Vieweg, Springer, Wiesbaden, 2016

[27] Etezadzadeh Ch., Niedbal M., *Smart City – Made in Germany*, Chapter: *Mobilität – Bewegung und Bewegungsfreiheit*, pp. 471-473, Springer Vieweg, Springer, Wiesbaden, 2020

[28] Chamoso P., González-Briones A., De La Prieta F., Venyagamoorthy G. K., Corchado J. M., *Smart city as a distributed platform: Toward a system for citizen-oriented management*, Journal: *Computer Communications*, Volume 152, pp. 323-332, Elsevier, 2020

[29] Waleed E., Alagan A., *Internet of Things for Smart Cities – Technologies, Big Data and Security*, Chapter: *Internet of Things for Smart Cities: Overview and Key Challenges*, pp. 1-15, Springer, Cham, 2019

[30] Chiehyeon L., Kwang-Jae K., Maglio P. P., *Smart cities with big data: Reference models, challenges, and considerations*, Journal: *Cities*, Volume 82, pp. 86-99, Elsevier, 2018

[31] Medium – Big Data & Smart Cities: How can we prepare for them? [Online], Available: https://medium.com/dataseries/big-data-and-smart-cities-why-we-need-them-now-a194b2498fb1, Last access: September 18, 2020

[32] Interesting Engineering – In What Ways Data Collection in Smart Cities Is Threatening? [Online], Available: https://interestingengineering.com/in-what-ways-data-collection-in-smart-cities-is-threatening, Last access: September 18, 2020

[33] McClellan St., Jimenez J. A., Koutitas G., Schemmel P. J., Schemmel J. J., Humphries E. D., *Smart Cities: Applications, Technologies, Standards, and Driving Factors*, Chapter: *Technology-Enhanced Infrastructure*, pp. 209-233, Springer, Cham, 2018

[34] Colistra J., *The Evolving Architecture of Smart Cities*, Journal: *2018 IEEE International Smart Cities Conference (ISC2)*, IEEE, 2018

[35] Villanueva-Rosales N., Garnica-Chavira L., Larios V. M., Gómez L., Aceves E., *Semantic-enhanced Living Labs for better interoperability of Smart Cities solutions*, Journal: *2016 IEEE International Smart Cities Conference (ISC2)*, IEEE, 2016

[36] Bibri S. E., Krogstie J., *The core enabling technologies of big data analytics and context-aware computing for smart sustainable cities: a review and synthesis*, Journal: *Journal of Big Data*, Article number: 38, Springer Open, 2017

[37] Stadt Wien – Smart City – Historie [Online], Available: https://smartcity.wien.gv.at/site/der-wiener-weg/historie/, Last access: September 18, 2020

[38] Stadt Wien – Smart City – Rahmenstrategie [Online], Available: https://smartcity.wien.gv.at/site/der-wiener-weg/rahmenstrategie/, Last access: September 18, 2020

[39] Stadt Wien – Smart City – Projekte [Online], Available: https://smartcity.wien.gv.at/site/projekte/, Last access: September 18, 2020

[40] IoT World Today – The World's 5 Smartest Cities [Online], Available: https://www.iotworldtoday.com/2016/05/18/world-s-5-smartest-cities/, Last access: September 18, 2020

[41] Vitunskaite M., He Y., Brandstetter Th., Janicke H., *Smart cities and cyber security: Are we there yet? A comparative study on the role of standards, third party risk management and security ownership*, Journal: *Computers & Security*, Volume 83, pp. 313-331, Elsevier, 2019

[42] IoT World Today – The World's 5 Smartest Cities – Page 2 [Online], Available: https://www.iotworldtoday.com/2016/05/18/world-s-5-smartest-cities/2/, Last access: September 18, 2020

[43] Technologymagazine – Top 10 smart cities in the world [Online], Available: https://www.gigabitmagazine.com/big-data/top-10-smart-cities-world, Last access: September 18, 2020

[44] Imperial College London – London Living Labs (L3) [Online], Available: https://wp.doc.ic.ac.uk/aese/project/london-living-labs-l3/, Last access: September 18, 2020

[45] RAD lab – About [Online], Available: http://rad-um.com, Last access: September 18, 2020

[46] RAD lab – Projects – Zenciti [Online], Available: http://rad-um.com/projects/zenciti, Last access: September 18, 2020

[47] ARCH2O – A New Smart City in the Yucatan Peninsula will be built by the help of University of Miami School of Architecture [Online], Available: https://www.arch2o.com/new-smart-city-yucatan-peninsula-will-built-help-university-miami-school-architecture/, Last access: September 18, 2020

[48] Archinect News – University of Miami School of Architecture helps to create a new Smart City in the Yucatan Peninsula [Online], Available: https://archinect.com/news/article/149966504/university-of-miami-school-of-architecture-helps-to-create-a-new-smart-city-in-the-yucatan-peninsula, Last access: September 18, 2020

[49] Synopsys – What is an Autonomous Car? [Online], Available: https://www.synopsys.com/automotive/what-is-autonomous-car.html, Last access: September 18, 2020

[50] Statista – Number of smartphone users worldwide from 2016 to 2021 [Online], Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, Last access: September 18, 2020

[51] Dziekan K., Kottenhoff K., *Dynamic at-stop real-time information displays for public transport: effects on customers*, Journal: *Transportation Research Part A: Policy and Practice*, Volume 41, Issue 6, pp. 489-501, Elsevier, 2007

[52] Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M., *Internet of Things for Smart Cities*, Journal: *IEEE Internet of Things Journal*, Volume 1, Issue 1, pp. 22-32, IEEE, 2014

[53] Briand A.-S., Côme E., Trépanier M., Oukhellou L., *Analyzing year-to-year changes in public transport passenger behaviour using smart card data*, Journal: *Transportation Research Part C: Emerging Technologies*, Volume 79, pp. 274-289, Elsevier, 2017

[54] Gutiérrez A., Domènech A., Zaragozi B., Miravet D., *Profiling tourists' use of public transport through smart travel card data*, Journal: *Journal of Transport Geography*, Volume 88, 102820, Elsevier, 2020

[55] Statista – Welche dieser Online-Aktivitäten üben Sie mindestens wöchentlich mit einem Smartphone/ mit einem Computer aus? [Online], Available: https://de.statista.com/statistik/daten/studie/461297/umfrage/umfrage-zur-nutzung-von-angeboten-auf-mobilen-endgeraeten-in-oesterreich/, Last access: October 11, 2020

[56] Axis Communications – Smart City – Was bedeutet das eigentlich? [Online], Available: https://www.axis.com/blog/secure-insights-de/was-ist-eine-smart-city/, Last access: October 11, 2020

[57] Rhino Online Help – Rhinoscriptsyntax [Online], Available: https://developer.rhino3d.com/api/RhinoScriptSyntax/, Last access: October 12, 2020

List of Figures

List of Tables

List of Abbreviations

AAL	Active and Assisted Living
BAS	Building Automation System
CAD	Computer Aided Design
CCTV	Closed Circuit Television
ECU	Electronic Control Unit
GPS	Global Positioning System
H2H	Human to human interaction
HVAC	Heating, Ventilation and Air Conditioning
ICT	Information and Communications Technology
IoT	Internet of Things
LoRaWAN	Long Range Wide Area Networks
M2M	Machine to machine interaction
OS	Operating System
PC	Personal Computer
RAD	Responsive Architecture and Design lab
RAM	Random Access Memory
UDG	University of Guadalajara
UDP	Unified Diagnostic Services
UM	University of Miami
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WSN	Wireless Sensor Networks
WWW	World Wide Web