Interreg North Sea

Co-funded by the European Union

ShareDiMobiHub

Inventory of shared mobility hubs in the Province of Utrecht

DELIVERABLE 2.14

20-12-2023

Gido Stoop (Hogeschool Utrecht)

Summary sheet

Project Name	**ShareDiMobiHub**
Title of the document	
Deliverable	D 2.14
Work Package	2
Programme	Interreg NSR
Coordinator	Province of Utrecht
Website	https://www.interregnorthsea.eu/sharedimobihub
Author	Gido Stoop
Status	Public
Dissemination level	Public
Reviewed by	
Submission date	
Starting date	September 2022
Number of months	36

Project partners

Organisation	Abbreviation	Country
Province of Utrecht	ProvU	Netherlands
Capital Region of Denmark	CRD	Denmark
Vestfold and Telemark county	VTFK	Norway
Subpartner: Statens vegvesen	SVV	Norway
Subpartner: Tønsberg kommune	TK	Norway
Subpartner: Porsgrunn municipality	PK	Norway
Subpartner: Skien municipality	SK	Norway
Promotion of Operation Links with Integrated Services	POLIS	Belgium
City of Amsterdam	AMS	Netherlands
City of Leuven	LEU	Belgium
University of Antwerp	UAntw	Belgium
Transport Authority for the Amsterdam Region	VRA	Netherlands
Mpact	Mpact	Belgium
Autodelen.net	Auto	Belgium
City of Rotterdam	ROT	Netherlands
Hamburg University of Applied Sciences	HAW	Germany
University of Applied Sciences Utrecht	HU	Netherlands

Document history

Version	Date	Organisation	Main area of changes	Comments
0.1	24-06-2023	HU	Initial version	
1.0	20-12-2023	HU	Added limitations paragraph, policy recommendations and other minor changes to figures	

Table of Contents

List of figures

List of tables

1. Abstract

This report focuses on shared-mobility providers, shared-mobility vehicles and mobility hubs in the Province of Utrecht. The aim is to gain an understanding of the current state of shared mobility in the Province of Utrecht. Data was gathered directly from service-providers and used to create an inventory of shared mobility offers. Data was collected on shared mobility options such as cars, bicycles, mopeds, cargo bikes and vans. Public transportation stops were also used in the spatial analysis performed for identifying hubs. The data was reviewed and validated through a combination of fieldwork and desk research. The collected data was then plotted and analysed using GIS. The inventory of shared-mobility vehicles showed that cars are the most prevalent shared modality, followed by (e-)bikes and mopeds. Shared-mobility could be found throughout the province, although the level of urbanization did influence the available modality types. The hub-analysis reveals that a total of 100 hubs are identified, of which 80 have at least one form of shared-mobility options. The bus-car hub were the most common. The hubs are distributed throughout the province, with a concentration around the city of Utrecht.

2. Introduction

Meanwhile, shared mobility has become a well-established concept. The province of Utrecht also recognizes opportunities to expand shared mobility in both urban and rural areas. However, what is the current state of affairs? Do we truly have an understanding of the (seemingly) rampant proliferation of shared mobility services being deployed? As a partner in the Interreg project ShareDiMobiHub, the University of Applied Sciences Utrecht conducted a systematic inventory of shared mobility vehicles and hubs in the province of Utrecht.

As of now, the appointment of shared mobility services is performed on a municipal level. To gain insight into the current state of affairs regarding shared mobility services, the municipalities should share this information with other government bodies like the province of Utrecht. However, unfortunately this is not always the case and the registration of the available services in each of the municipalities is not uniform. Moreover, they don't always have all of the data themselves, but rather in the databases of the shared mobility providers.

The province of Utrecht aims to create a plan for rolling out shared mobility services that will operate across municipal borders. This is still uncommon, as shared mobility is provided on a municipal level and providers are therefore forced to keep their vehicles within these borders. In order to collect all the data needed to create a plan that transcends municipal borders, a general approach to data collection has to be taken that can be performed throughout the province. In the end it is desirable that this approach can be applied in any province. In this research, data from public data sources is collected and analysed regarding its ability to give an insight into the state of affairs.

Finally, this research is focussed on identifying mobility hubs. Even if these hubs were previously not labelled as such. Hubs are a place where travellers can easily transfer between (shared-mobility-)modalities. Therefore, a hub is a place where at least two different type of modalities are within acceptable walking distance of each other. The hubs are a key feature in the development of MaaS (Mobility as a Service), which aim to provide, among others, an integrated system that can provide multi-modal travel advice and means of booking and paying these services.

This research is followed up with research on the identification of potential new hub places, and the development of a data dashboard, that can provide information to municipalities and other stakeholders regarding the effects of shared mobility.

All research questions that The University of Applied Sciences Utrecht aims to answer for the project are listed below. The questions in bold relate to the inventory and location of hubs and therefore relate to this document:

- **What can be learned from the current supply and use of shared mobility and mobility hubs?**
- What are success factors for the successful roll-out of shared mobility and mobility hubs?
- How can the best locations for shared mobility hubs be identified?
- How can data on shared mobility/hubs be structured to be used as a decision support tool?
- How can the impact of shared mobility/hub on multiple societal goals be measured best?

To answer this question, it is necessary to have knowledge on the current state of affairs regarding shared mobility and mobility hubs. Therefore, this report answers the questions:

- What is the current inventory of shared mobility providers and shared mobility vehicles in the Province of Utrecht?
- What is the current inventory of shared mobility hubs in the Province of Utrecht?

3. Methodology

This methodology starts with a description of the case study in this research. This is followed by an explanation of the research design, outlining the overall structure. Special attention is given to the data collection methods employed and analyzing their suitability for the research, keeping in mind, completeness, relevance, quality and limitations.

3.1 Case study

To gain an understanding in both the urban and rural shared mobility situation for the ShareDiMobiHub project, a case study approach was employed for the Province of Utrecht. The province is the smallest in area of the Netherlands, but is 5th out of 12 in terms of total population. The province is responsible for securing the reachability of all it's 26 municipalities. The province is also responsible for the public transport system in the region, as they have the role as grantor. The province of Utrecht has been chosen fort his case study, as an increase in mobility and a growing need for flexibility for travelers is expected. These developments ask for the connection of multiple mobility networks to one multimodal regional (and potentially national) mobility system.

3.2 Research design

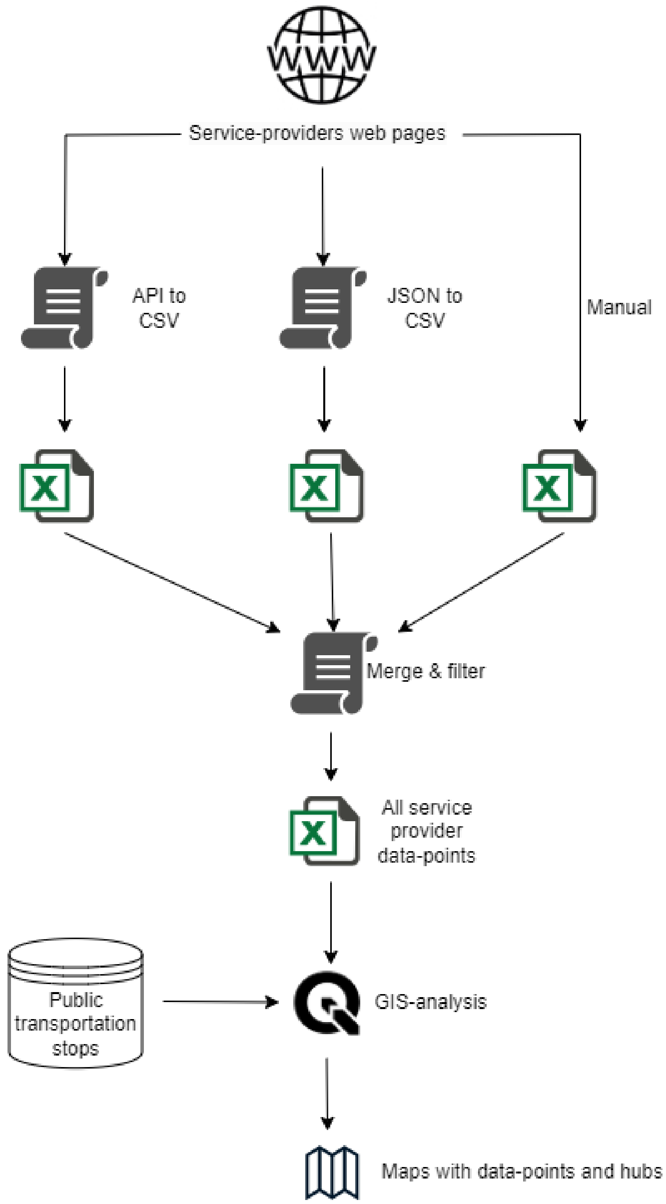
As mentioned in the introduction, the research data will be acquired from public data-sources. This approach has been chosen, as shared-mobility data is usually collected on a municipal level, but this data is not collected uniformly by all municipalities, is often incomplete and hard to come by. Therefore, public data-sources are chosen and its possibilities and limitations will be evaluated in this report. Another benefit of using public data-sources, is that this research will give guidelines for a general executable approach that can be used by others when answers to similar research questions are sought.

In this report, the data used is for the research thoroughly discussed. As data was collected from a multitude of sources, much steps have been taken to homogenise the data. This is of high importance since characteristics of the data used, can influence the way the inventory can be interpreted.

Finally, a quantitative analysis will be used to determine the existing (shared-)mobility hubs in the province of Utrecht. Geographical Information Systems (GIS) are used to conduct spatial analysis on the collected data. In this spatial analysis, the spatial relationship between different types of modalities are analysed and clustered. I.a. locations are identified where multiple forms of transportation are within close proximity of each other.

Figure 1 shows the conceptual model used for this research. In the top row are the public data sources used. Following the arrows, it displays the way in which the raw data is saved to a generalized file format, which is merged, filtered and cleaned. Then GIS are used to perform spatial analysis and identify hub locations and classify them based on the size of the hub. For this step, additional data has been obtained from an open data source on public transportation stops.

Figure 1 Conceptual model of the hub-analysis



3.3 Data Collection Methods

3.3.1 Data from service-providers

As can be seen in figure 1, the data sources collected can be divided in roughly two categories. Firstly, the shared-mobility data-points, and secondly, the public transport stops. Since this case study is focused on the Province of Utrecht, an attempt has been made to include all forms and providers of both shared-mobility services and public transportation services in the province. A thorough search has been conducted, and since shared-mobility services must be easily findable for successful operation, and later cross-checked with exploratory research performed by students and an expert from the Province. Table 1 shows all the service providers, the data source from where the data is obtained, the transport mode the service providers offer, the method for extracting the data, and finally the vehicle return type. Regarding the vehicle return types for the various services, there are four different types in the Province of Utrecht:

1) The Fixed Base, which means that the rented vehicle must be returned to the place where it was initially taken from.
2) Back-to-many (or Base-to-Base), which means that the rented vehicle must be returned to any of the dedicated bases the service provider offers.
3) Free Floating, which means that the rented vehicle can be taken from anywhere within the zone(s) and must also be returned to anywhere in the zone. The zones are usually very extensive and aim to cover much of the area within the city limits.
4) Zone, which is in principle the same as Free Floating, although in this case, the zone only consists of one specific area of not more than a few streets or a small neighborhood.

There is also a fifth service type which is being disregarded in this report. This service is based on neighborhood cooperation, in which the (mostly) shared-cars are the joint responsibility of the participants in the neighborhood. However, this also means that the shard-vehicle is not accessible to those who don't participate. Therefore this service type is not taken into consideration in this inventory. An example of a service-provider in the province of Utrecht is MobiGo

Table 1 Shared mobility service provider characteristics and data extraction method

Service provider	Data source	Modality	Extraction method	Return type
OV-fiets	OVfietsenbeschikbaar.nl	Bicycle	API	Fixed Base
MoveYou	CROW	Bicycle	API	Back-to-many
Donkey Republic	CROW	Bicycle	API	Back-to-many
GO Sharing	CROW	Moped	API	Free Floating
Check	CROW	Moped	API	Free Floating
Tier	CROW	(E-)Bike	API	Free Floating
Bird	CROW	(E-)Bike	API	Free Floating
Dott	CROW	Bicycle	API	Free Floating
Cargoroo	Cargoroo	Cargo bike	Web scraping	Fixed Base
Keobike	Keobike	Bicycle	Web scraping	Back-to-many
Greenwheels	GreenWheels	Car	Extract API-response from source	Fixed Base
MyWheels	MyWheels	Car	Extract API-response from source	Fixed Base or Zone
SnappCar	SnappCar	Car	API	Fixed Base
WeDriveSolar	WeDriveSolar	(E-)Car	Download	Fixed Base

JustGo	JustGo	Car	Manual extraction	Fixed Base
SixtSharing	SixtSharing	Car	Manual extraction	Fixed Base
Hely	Hely	Car	Manual extraction	Fixed Base
Kav2Go	Kav2Go	Van	Manual extraction	Fixed Base

Not all services providers follow the same business model, and likewise the data is not homogenous. During this research, choices had to be made about which services to include, and which services to exclude from the analysis. In the following sections, the data, the business model of the providers and the limitations of the data are discussed.

3.3.2 OV-Fiets

OV-Fiets is a bicycle rental scheme of the national railway company of the Netherlands (NS). It is a return to base system in which bicycles can be rented against a daily fee. The bicycles are rented using the public transportation card. OV-Fiets is usually, but not always, situated near train stations. Most locations offer standard bikes, but at some of the bigger stations, e-bikes are also offered.

Data on OV-fietsen has been obtained from ovfietsenbeschikbaar.nl/locaties. The data is shown in a map on their website, but the raw data-points can also be extracted from this map. Simply open the inspect tool in the browser and go from ‘Network ‘-> ‘map-locations’ -> ‘Response’. Below is a small snippet of the JSON response.


```
{…}
beschikbaar"},{"location":"Aalten","
tableurl":"\/locatie\/atn001","lat":"51.92161","lng":"6.57929","text":
"3                                                          fietsen
beschikbaar"},{"location":"Alkmaar","url":"\/locatie\/amr001","lat":"5
2.63753","lng":"4.73985","text":"29 fietsen beschikbaar
{…}
```


Using the python file OV-fiets_JSON_to_CSV.py (Appendix A) the JSON file was converted to a csv file with longitude and latitude columns.

An evaluation of the OV-Fiets data can be found in table 2.

Table 2 Data evaluation OV-Fiets

Data Reliability and Source	Data from this source is based on data from OpenOV, which part of Stichting OpenGeo , which provides open geodata.
Data Completeness	The data-source has data for all of the Netherlands, and therefore also covers all of the Province of Utrecht.

Data Quality	The coordinates seem to correspond with the actual OV-fiets locations.
Data Relevance	OV-fiets is the most well-known and biggest 'fixed base' shared-bicycle provider in the Netherlands.
Privacy and Legal	Open data provided by the organization 'Open Geodata'
Limitations	The data only provides information on the total number of available bikes. It does not show the total size of the fleet. It is possible to obtain this information, but for this research, we assume that every fleet has more than one bicycle. The data-points represent the rental location and not the bikes itself, this means that it is not possible to retrieve usage data from the data used for this analysis.

3.3.3 CROW dashboard

The CROW dashboard for shared mobility (CROW, 2023) has got data points of the locations of 19 different shared-mobility providers. The CROW's raw datapoints can be obtained with a GET request to their API (Appendix B). For this case study, only the ones that are operating in the province of Utrecht were selected and which have data that is hard to retrieve via other sources. These are 6 total and can be found in the table below.

Table 3 CROW dashboard service providers used for analysis

Provider	Modality	**Is the data correct?**
Donkey Republic	Bicycle	Dashboard does not correspond with service website
Check	Moped	Have to install the app to check
Tier	(E-)Bicycle	Have to install the app to check
Bird	(E-)Bicycle	Have to install the app to check
Dott	Bicycle	Have to install the app to check
MoveYou	Bicycle	Dashboard correspond with service website

An evaluation of the CROW data can be found in table 4.

Table 4 Data evaluation CROW

Data Reliability and Source	CROW is a renowned Dutch knowledge platform on infrastructure, public space and transportation
Data Completeness	CROW has included a wide range of shared mobility providers on their platform. Especially the data on shared (cargo-)bikes and mopeds are extensive. Data on shared cars is limited to a few providers.
Data Quality	The data-points are refreshed every 15 minutes. Due to arrangements with the service providers, the data is warped for about 100m in random directions, so the coordinates of the data-points are not entirely accurate.

Data Relevance	It is one of the most important sources for the inventory of data. It is the most complete source for acquiring shared mobility data-points, especially for free floating vehicles.
Privacy and Legal	CROW has granted us permission to use their public-API, but the data needs to be handled carefully to not breach agreements with the data-providers. So the guarantee of privacy needs to be considered when using the data.
Limitations	- The API should not be called too often, to not overload the CROW-server. - Some of the service providers that use a free floating scheme, also have some standard form of 'fixed bases' in a few occasions. The data does not distinguishes between these types. And therefore it is hard to determine the fixed locations from this data-source.

<u>MoveYou:</u>

The website does not mention their bicycles. However, a blog post does mention that they have taken over all activities of GoAbout since October 2021 (MoveYou, 2021). The bases are listed on the GoAbout website (along with information on whether and how many bicycles are available at each base). These correspond to the markers labeled on the CROW dashboard. After conducting thorough fieldwork (which involved walking to the bicycle parking at Heidelberg 7), I have confirmed that the bicycles still bear the GoAbout name.

<u>Donkey Republic</u>

According to the Donkey Republic website, they offer bicycles in Amsterdam, Rotterdam, The Hague, and Dordrecht within the Netherlands (Donkey Republic, 2023a). However, when looking at the Donkey Republic map, it appears that they also operate in Maarn, Doorn, and Amerongen (Donkey Republic, 2023b). Lastly, the CROW's Shared Mobility Dashboard indicates that at the time of writing (20-04-23, 15:00), there is 1 bicycle in Utrecht and 4 bicycles in Leerdam. Therefore, the overview provided by Donkey Republic is quite unclear. A group of students from the HU that are doing research on the topic, have confirmed that Donkey Republic is indeed active in Maarn, Doorn and Amerongen.

3.3.4 Cargoroo

Cargoroo is a company that provides electrical cargo bikes through a mobile application. The costs are calculated by total time used. Subscriptions are also available, which reduce the hourly tariff of the cargo bike (Cargoroo, 2023). Cargoroo bikes have their own fixed bases, sometimes indicated by road marking on the sidewalk.

Cargoroo data is displayed on a map on their website. The coordinates of their locations are embedded in the HTML-script of https://cargoroo.nl/cargoroo-cities/ and can therefore be extracted using a Python-script. This script was also used to convert the data to CSV (Appendix C).

An evaluation of the Cargoroo data can be found in table 5.

Table 5 Data evaluation Cargoroo

Data Reliability and Source	The data was obtained directly from the Cargoroo website and should therefore be a reliable source for data on Cargoroo cargo bike locations.
Data Completeness	The data is obtained from the Cargoroo website, and it is therefore assumed that the data is complete and up-to-date.
Data Quality	Cargoroo cargo bikes have their own dedicated parking spot, therefore the data-points are very accurate as to where the cargo bike would usually stand.
Data Relevance	Cargoroo is currently the only provider of cargo bikes in the Province of Utrecht. They currently only operate in the city of Utrecht.
Privacy and Legal	The data is publicly available on their website in the form of coordinates.
Limitations	The data is not real-time, so it does not represent the location of the cargo bike, but rather the location of the pick-up and drop-off point. This also means that it is not possible to retrieve usage data.

3.3.5 Keobike

Keobike is a company that offers clients back-to-many bicycle rental services. Just like OV-Fiets, it is a service provided by public transportation companies, namely Keolis and Syntus. It can be hired at the bases using the app for an hourly, or daily rate (Keobike, n.d.). Keobikes can be found hanging upright on so called ‘carrousels’.

For acquiring the data of the locations of the Keobike bases, the same method was used as for Cargoroo. The coordinates were embedded in the HTML of their home page and can were extracted and formatted in a CSV-file using a Python script (Appendix D).

An evaluation of the Keobike data can be found in table 6.

Table 6 Data evaluation Keobike

Data Reliability and Source	The data was obtained directly from the Keobike website and should therefore be a reliable source for data on Keobike bicycle locations.
Data Completeness	The data is obtained from the Keobike website, and it is therefore assumed that the data is complete and up-to-date.
Data Quality	Keobikes have their own dedicated ‘carrousels’ (back-to-many), therefore the data-points are very accurate as to where the bikes can be found.
Data Relevance	Keobike is the biggest provider of back-to-many bicycle rental options in the Province of Utrecht and can be found in both Rural and Urban areas.

Privacy and Legal	The data is publicly available on their website in the form of coordinates.
Limitations	The data is not real-time, so it does not represent the location of the bicycle, but rather the location of the pick-up and drop-off point. This also means that it is not possible to retrieve usage data.

3.3.6 GreenWheels

GreenWheels is a company that provides shared cars that are accessible to everyone who registers through their application. The cost for the user is a function of the duration of the rental and the distance travelled. GreenWheels cars usually have their own dedicated parking spaces with signage with either one of multiple cars. The signage used differs per municipality (figure 2). GreenWheels has both electric and non-electric cars (GreenWheels, 2023).

Figure 2 Signage for GreenWheels in the municipality of Utrecht

Greenwheels' car locations are displayed on a map on their website. Unfortunately, they request their API using the POST method, meaning the data cannot be requested from unauthorized sources. However, it is possible to get the raw data-points the same way as the OV-fiets points. Namely via 'inspect' -> 'network' -> 'graphql' -> 'response'. Finally, the Greenwheels_JSON_to_CSV.py (Appendix E) script was used to convert the data to a table.

An evaluation of the GreenWheels data can be found in table 7.

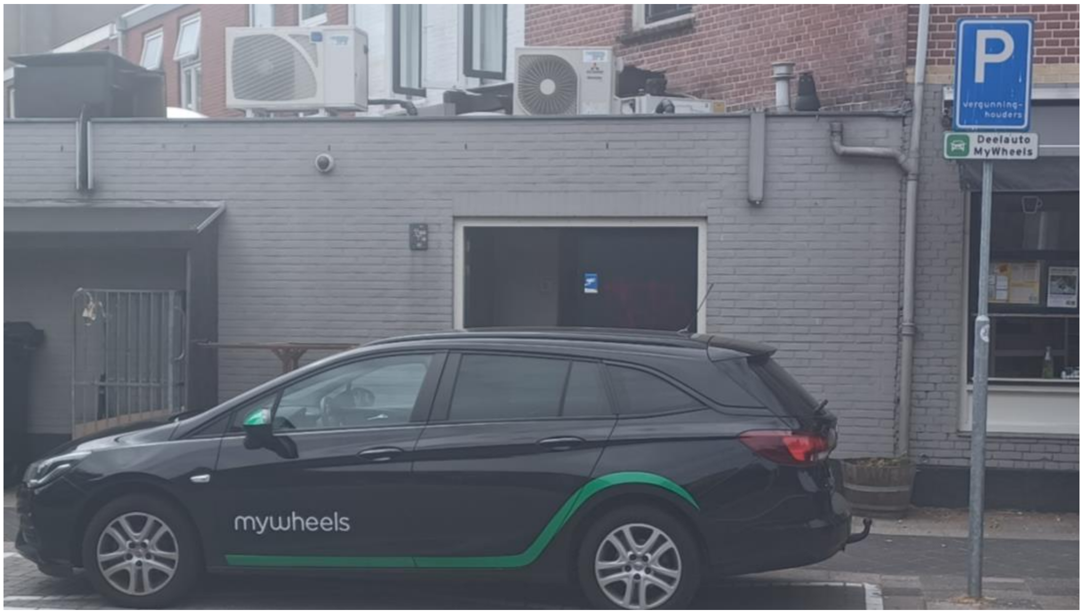
Table 7 Data evaluation GreenWheels

Data Reliability and Source	The data was obtained directly from the GreenWheels website and should therefore be a reliable source for data on GreenWheels car locations.
Data Completeness	The data is obtained from the GreenWheels website, and it is therefore assumed that the data is complete and up-to-date.
Data Quality	GreenWheels cars have their own dedicated parking spot, therefore the data-points are very accurate as to where the car would usually stand.
Data Relevance	GreenWheels has more than 2600 cars and is therefore one of the bigger shared car providers in the Netherlands.
Privacy and Legal	The data also contains information on an address associated with the coordinate information and license plate information. This data has been left out, as it is not relevant for this research and might conflict with privacy standards and ethics.
Limitations	The data is not real-time, so it does not represent the location of the car, but rather the location of the pick-up and drop-off point. This also means that it is not possible to retrieve usage data.

3.3.7 MyWheels

MyWheels is similar in concept to GreenWheels. It is a company that provides shared cars that are accessible to everyone who registers through their application. The cost for the user is a function of the duration of the rental and the distance travelled. MyWheels cars usually have their own dedicated parking spaces. MyWheels also has various locations, in which there are multiple MyWheels cars in adjacent parking spots. MyWheels has 2500 cars, of which 1327 are electric (GreenWheels, 2023). These cars are usually signed with the text “Deelauto MyWheels” (figure 3).

Figure 3 Signage for MyWheels in the municipality of Utrecht



MyWheels' API is also protected via the POST method. The same method applies as for the GreenWheels locations: 'inspect' -> 'network' -> '/api' -> 'response'. To convert the data to a table, MyWheels_JSON_to_CSV.py was used (Appendix F).

An evaluation of the MyWheels data can be found in table 8.

Table 8 Data evaluation MyWheels

Data Reliability and Source	The data was obtained directly from the MyWheels website and should therefore be a reliable source for data on MyWheels car locations.
Data Completeness	The data is obtained from the MyWheels website, and it is therefore assumed that the data is complete and up-to-date.
Data Quality	MyWheels has two types of parking types. In some cases, the car needs to be parked in a zone, other times, it has a fixed location. In the case of the zone, the precise location of the car is unknown
Data Relevance	MyWheels has more than 2600 cars and is therefore one of the bigger shared car providers in the Netherlands.
Privacy and Legal	The data is publicly available on their website and does not same to contain any privacy-sensitive data.
Limitations	MyWheels is present in some 'hub' locations that were designed by the municipalities, however in the data, the distinction between isolated cars and clustered cars are not made, further spatial analysis need to be performed to get this information.

MyWheels has locations with only one car, but also locations with multiple cars. However, the data does not display any relation between multiple cars in the same location. Therefore a buffer analysis was performed to add this spatial relation. A buffer radius was used of 15 meters, to cluster all the cars within

15 meters distance of one another. 15 meters was chosen, as this should encapsulate all adjacent parking spaces.

3.3.8 SnappCar

SnappCar is a peer-to-peer car sharing service. Meaning that the cars are owned by individuals that applied some modifications to their cars in order to meet the requirement for enrollment in the SnappCar scheme. SnappCars can only be rented for full days (i.e. 1 day, 2 days, 3 days etc.) (SnappCar, 2023). As the cars are owned by individuals, the type of parking can also vary. Some cars might have a private dedicated parking spot, while other cars might have to be parked in a certain zone.

SnappCar does not have a map, however, the locations of the cars can be queried through their API. Since the requests are limited to giving only 10 cars per requests, the querying all cars in the province of Utrecht may take a few minutes. The script for querying the data is called SnappCar_API_to_CSV (Appendix G).

An evaluation of the SnappCar data can be found in table 9.

Table 9 Data evaluation SnappCar

Data Reliability and Source	The data was obtained directly from the SnappCar website and should therefore be a reliable source for data on SnappCar car locations.
Data Completeness	The data is obtained from the SnappCar website, and it is therefore assumed that the data is complete and up-to-date. All vehicles were queried using a range that covers the whole province of Utrecht.
Data Quality	The SnappCar-website only mentions the street in which the standard spot of the vehicle is, not the exact location. The coordinates used for the data-points are based on these street names and therefore the coordinates might not correspond exactly with the location of the vehicle's parking spot.
Data Relevance	Even though they are peer-to-peer, they are too big to leave out of the inventory. They have over 13.000 cars in the Netherlands.
Privacy and Legal	The data was collected by running a query over their public-API. Collecting all data requires multiple API-requests, a small time-lag has been applied between these requests to not overload their server.
Limitations	SnappCar is a peer-to-peer car sharing platform, which means that the company actually doesn't own any of the vehicles and don't have control over where the vehicles are placed. Therefore, they are not involved with the municipalities for actively creating hubs.

3.3.9 We Drive Solar

We Drive Solar is a subscription based car sharing service. In which the user pays a monthly fee in order to rent the car. With this subscription, users can use the cars from We Drive Solar and charge them with the exclusive We Drive Solar charging locations, as the company only offer electrical cars. We Drive Solar cars are situated in a fixed base with one or more cars (We Drive Solar, 2023).

We Drive Solar has a useful map on their website showing all the locations of We Drive Solar cars and charging stations. These can be downloaded as a .kml file from Google Maps. Then, only the cars were loaded into QGIS and exported to a CSV file. Finally, two columns were added in Excel: one with the header 'service', with the value 'We Drive Solar' for all rows, and another column with the header 'modality', with the value 'car'. This is necessary to merge everything into one large CSV file. The script used for merging can be found in Appendix H.

An evaluation of the We Drive Solar data can be found in table 10.

Table 10 Data evaluation We Drive Solar

Data Reliability and Source	The data was obtained directly from the We Drive Solar website and should therefore be a reliable source for data on We Drive Solar car locations.
Data Completeness	The data is obtained from the We Drive Solar website, and it is therefore assumed that the data is complete and up-to-date.
Data Quality	The data-point coordinates correspond with the location of the Standard parking spot of the cars.
Data Relevance	We Drive Solar works with dedicated parking places and is therefore eligible for the hub-location analysis that will be performed in this research.
Privacy and Legal	The data-points can simply be downloaded from their website for free.
Limitations	Using We Drive Solar requires a subscription. Therefore it might be less suitable when trying to incorporate it in MaaS.

3.3.10 Other small providers

Finally, there are some other small providers which only have a few vehicles in the province of Utrecht. These are: JustGo with 5 cars, SixtShare with one base of cars, Hely with one base of cars that is publicly accessible, and Kav2Go with 7 vans. The locations of these providers has been manually added to CSV files (JustGo, n.d.; SixtShare, n.d., Hely, 2022; Kav2Go, 2023).

An evaluation of the data of small service providers can be found in table 11.

Table 11 Data evaluation other small service providers

Data Reliability and Source	The data-points were obtained by either finding addresses of the locations on their website, or from a map with points on their website. Therefore, it is assumed that the points are correct and up-to-date.
Data Completeness	In the case of SixtShare, it has one data-point a whole fleet of cars. However, it is not entirely clear how many cars are in their fleet. Hely has the same issue. For the other providers: the vehicles have their own fixed parking spot.

Data Quality	The location of the SixtShare fleet is somewhere in a parking garage, the coordinates are not exact. In all other cases, the coordinates are exact.
Data Relevance	Small providers are included for completeness. SixtShare and Hely are even more relevant, as they operate with multiple cars on a fixed base return type.
Privacy and Legal	Data-points have been generated manually by using the data available on the websites of the providers.
Limitations	These providers total only a hand-full of cars. Hely operates only with a subscription model.

3.3.11 Public transportation:

The public transportation stops were obtained from a public layer from University of Groningen on the ArcGIS online platform. This layer is based on GTFS data from OpenOV from October 2021. The public transportation stops in the file also contain data on which lines halt at each stop.

The OV-network of 2023 was directly obtained from OpenOV (2021). It is a shapefile of the network of all of the Netherlands. This layer will only be used for the visualization of the network. The actual information on the relationship between the stops and the network is obtained from OSM. The network is also used to check they line up with the crowd-sourced data from OSM.

An evaluation of the 'public transportation stop' data can be found in table 12.

Table 12 Data evaluation public transportation stops

Data Reliability and Source	Data from this source is based on data from OpenOV, which part of Stichting OpenGeo , which provides open geodata. It was modified and published by the University of Groningen, which is a renowned University in the Netherlands.
Data Completeness	It is very hard to verify it's completeness, so it is a matter of trusting the sources. In general, it seems to be complete and corresponds with OpenStreetMap.
Data Quality	The locations of the bus-stops seem to correspond with the actual bus stop locations. For some stops, the route information was checked and was deemed correct.
Data Relevance	The data is from 2023, so some things might have changed since then. However, for the purposes of this research, the data is still relevant.
Privacy and Legal	OpenOV provides open geo-data, and the University of Groningen has also publicized the modified layer as open data.
Limitations	The data for measuring the popularity of an public transport stop is done by using the total number of routes on the stop. Ideally, number of passengers were used, but this data was not found or available.

3.4 Validation of the data

In order to validate the data-points, a combination of field-work and desk research have been conducted. Some randomly selected data-points from most of the service providers have been validated using Google Maps imagery, to check whether the vehicle was present on the correct place. In cases where this was not possible (vehicle was temporarily gone, imagery was too old etc.), some data-points have been checked by looking in a neighborhood in the city of Utrecht.

3.5 Data analysis

After the data collection, the data was merged in a single CSV-file (Appendix I) and visualized in a map using GIS-software. This CSV-file contains the name of the service provider, the location of each of the vehicles, the vehicle-type, and the return-type, and the number of vehicles per data-point. Finally, the data-points were filtered to only be in the province of Utrecht.

To improve the reproducibility of the research, a small shell script has been written, linking some automatic script and providing some instructions for acquiring up-to-date data for performing this research. This script is however currently for the Province of Utrecht and its service providers, other provinces might have other service providers and thus miss data when using this tool.

3.5.1 Analysis of Hubs

Hubs are places where two or more (shared-)modalities are in close proximity of each other. Therefore, spatial analysis is needed to analyze the spatial relationship between the data-points and identify hubs. Since hubs are defined by the number of different (shared-)modalities, this analysis will not focus on the inner-modality spatial relationships, but rather the inter-modality spatial relationships. For example, the spatial relationship between two cars is not taken into account, but the spatial relationship between a train and a bicycle is.

In the definition, the proximity parameter is only defined to be having to be 'close'. According to literature, the acceptable walking distance between transfer stops is about 3-5 minutes. Which is about 250 to 400 meters at an average walking speed (Duran-Rodas et al., 2022). However, using these values in high density areas made seemingly unrelated modal options into identified hubs. 150 meters seems to be the best parameter for proximity, as it makes spatial relationships between modal options within the same general area, but not on the other side of the building blocks.

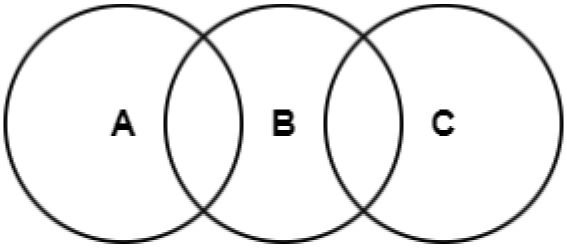
Not all shared-mobility modes have a role in forming mobility hubs. The shared mobility hub is a fixed place where shared mobility options can be expected to be available most of the time. Expectation is very important in this definition, for example, one cannot expect free floating devices to be present in the fixed location of the shared-mobility hub, therefore they are not taken into account in the analysis. Likewise, bases with only one car are also disregarded, as it cannot be realistically expected that this car will be available most of the time. Locations with multiple cars in one base are taken into consideration in the analysis. In table 13, an overview has been made of all the data-points that have been used in the hub-analysis.

Table 13 Service provider data-points used in hub-analysis

Service provider	Modality	Return type	Used in hub analysis?	Why not?
OV-fiets	Bicycle	Fixed Base	✓	
MoveYou	Bicycle	Back-to-many	✓	
Donkey Republic	Bicycle	Back-to-many	✓	
GO Sharing	Moped	Free Floating	✗	Free Floating devices don't use fixed hubs
Check	Moped	Free Floating	✗	Free Floating devices don't use fixed hubs
Tier	(E-)Bike	Free Floating	✗	Free Floating devices don't use fixed hubs
Bird	(E-)Bike	Free Floating	✗	Free Floating devices don't use fixed hubs
Dott	Bicycle	Free Floating	✗	Free Floating devices don't use fixed hubs
Cargoroo	Cargo bike	Fixed Base	✓	
Keobike	Bicycle	Back-to-many	✓	
Greenwheels	Car	Fixed Base	✓*	
MyWheels	Car	Fixed Base or Zone	✓*	
SnappCar	Car	Fixed Base	✗	As SnappCar is a peer-to-peer platform, all cars are singular entities. Due to this system, the cars are also not placed, but are rather in the parking spot that the owner of the car happens to use.
WeDriveSolar	(E-)Car	Fixed Base	✓*	
JustGo	Car	Fixed Base	✓*	
SixtSharing	Car	Fixed Base	✓	
Hely	Car	Fixed Base	✓	
Kav2Go	Van	Fixed Base	✓*	
*Only locations with multiple cars of the same service provider are used in the analysis				

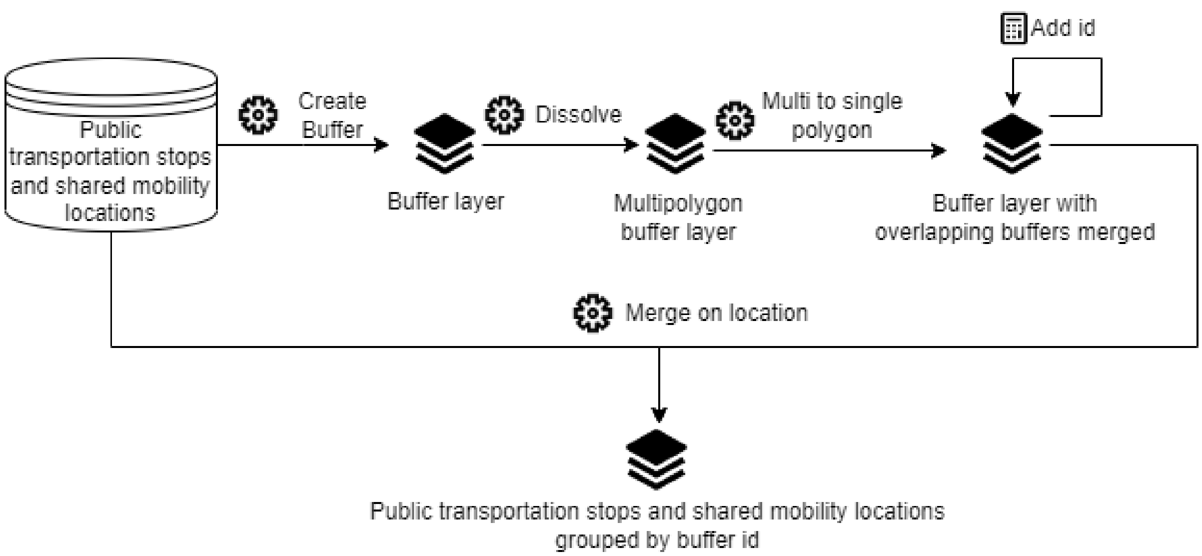
For the analysis, every data point will get a buffer with a radius of 75 meters. If the buffer overlap, this means that the datapoints are within 150 meters proximity of each other. The cluster are created by grouping all data-points with a direct or indirect spatial relationship. Figure 4 shows an example of how this clustering works. Even though circle A and C are not directly connected, they are connected through B and are therefore all part of cluster Z. In set theory it can be denoted as: $A \cap B \neq \emptyset \wedge B \cap C \neq \emptyset \Rightarrow \{A, B, C\} = Z$. Each cluster retains the data of each of the elements of the cluster.

Figure 4 Conceptual model of clustering method used in the hub-analysis



The full analysis was made using QGIS and was streamlined using its Model Designer. The full model can be found in Appendix J. A simplified version is shown in figure 5.

Figure 5 Visual representation of the model used for hub analysis in QGIS



After all clusters were made, each cluster would be would be analysed based on the characteristics of elements. If all elements had the same mode, they were discarded. If they have two or more different modalities, they are regarded as a hub.

4. Results

After performing the data collection and the analysis, several maps and tables were made. In this chapter, attention will be focused on the province as a whole, as well as some parts of the province in particular, like the biggest cities in the province, Utrecht and Amersfoort. The results section is split into two parts. The first part concerns the inventory of shared-mobility vehicles, and the second part of the results focusses on the identified shared-mobility hubs.

4.1 Inventory of shared-mobility vehicles

In table 14 an overview is made of all the data-points in the Province of Utrecht. A total of 3.404 data-points have been collected. Out of all the data-points, 1.767 data-points are related to cars. This suggests that cars are the most prevalent or widely recorded modality in the province. (E-)bicycles also have a significant presence with 1.306 data-points, though slightly lower when compared to cars. Mopeds are third with 293 data-points, all of which are belonging to Check and are located in the municipality of Amersfoort. There are 34 cargo bikes, all belonging to Cargoroo and only present in the municipality of Utrecht. Finally, the data includes 7 data-points, indicating a comparatively minimal presence in the dataset. Please note that data-points is not equivalent to number of vehicles, as data-points indicate the presence of one or more vehicles in the location.

Table 14 Number of data-points per modality

Modality	Data-point count
Car	1.767
(E-)Bicycle	1.306
Moped	293
Cargo Bike	34
Van	7
Total	3.404

In the table below (15) a overview is given of the total number of data-points used per provider. Tier has the highest count with 1.117 data-points, which also correspond to 1.117 vehicles. This suggests that a substantial amount of shared-vehicles are related to Tier. SnappCar is the second largest provider of data-points, with 783 records. Even though it has a significant presence in the data-set, the visibility of the service provider is low, due to the peer-to-peer business model it uses. Therefore, most cars are not branded. After SnappCar, GreenWheels and MyWheels have an almost equal amount of cars in the province. MoveYou seems to be the biggest provider of shared bicycles with a total of 118 data-points, although it can be assumed that OV-Fiets has higher total number of bikes, since they have vastly more vehicles per data-point. Unfortunately, the exact number is unknown.

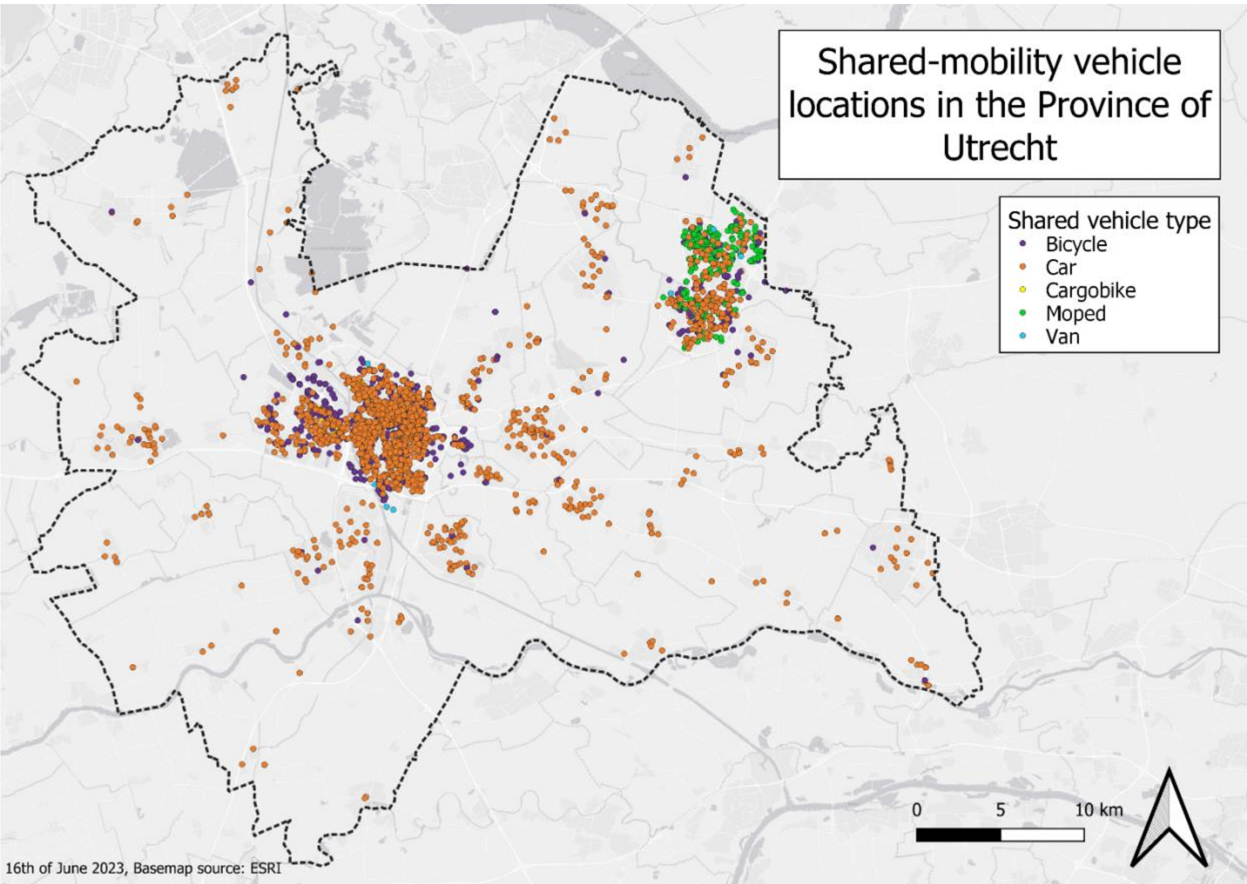
Table 15 Number of data-points per service provider

Service provider	Modality	Data-points count	Vehicles per data-point
Tier	(E-)bicycle	1117	1
SnappCar	Car	783	1
GreenWheels	Car	433	≥1
MyWheels	Car	400	≥1
Check	Moped	293	1
WeDriveSolar	Car	146	≥1
MoveYou	Bicycle	118	1
OV-fiets	Bicycle	37	>1
Cargoroo	Cargo bike	34	≥1
Keobike	Bicycle	15	>1
Dott	Bicycle	9	1
Kav2Go	Van	7	≥1
Donkey Republic	Bicycle	7	1
JustGo	Car	3	≥1
Bird	(E-)bicycle	2	1

SixtSharing	Car	1	>1
Hely	Car	1	>1

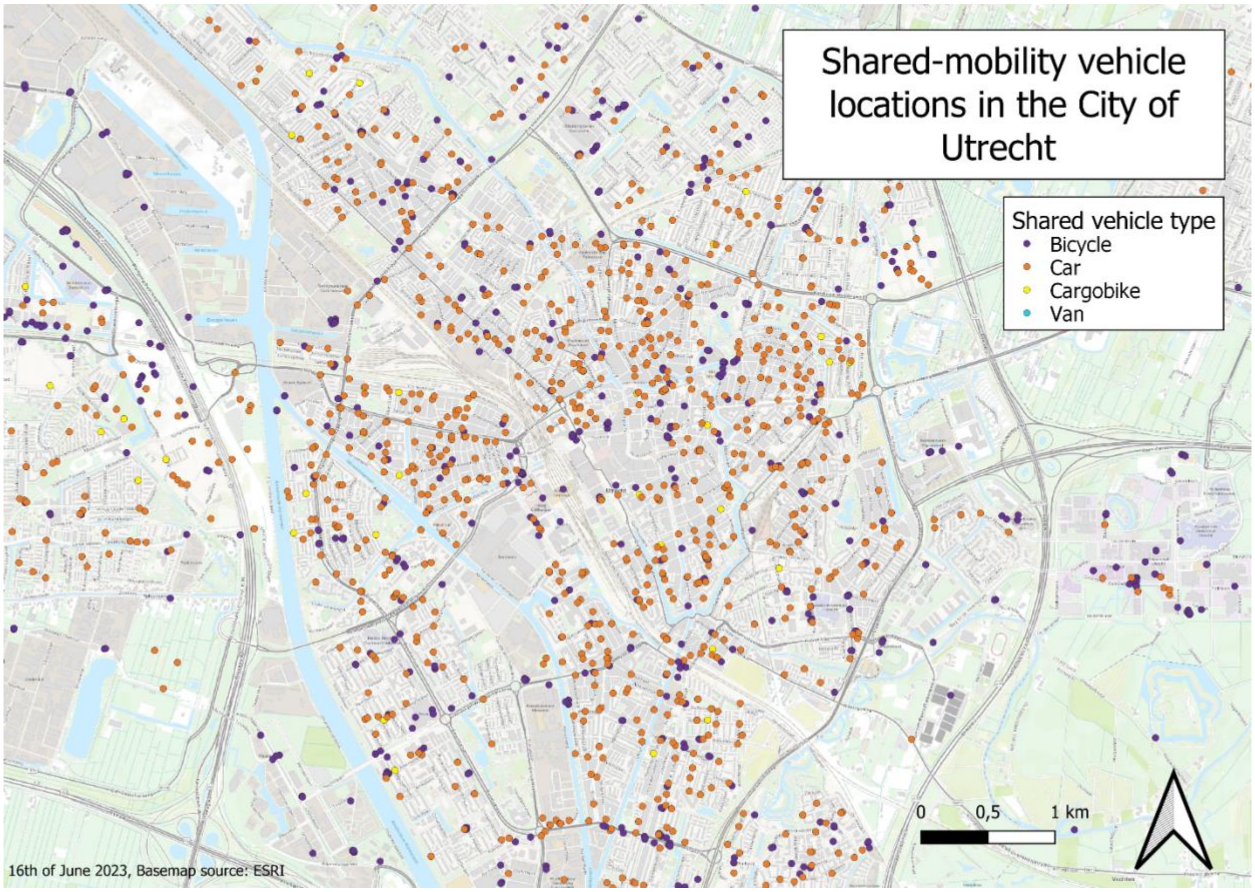
When looking at figure 6. it becomes clear that shared mobility is present in both urban- and rural areas, although the modalities in rural areas and smaller municipalities are mostly shared cars, and to a lesser extent, (e-)bikes. The majority of shared vehicles and the largest selection of unique modalities can be found in the two most urbanized municipalities of the province in terms of population, namely Utrecht and Amersfoort.

Figure 6 Shared-mobility vehicle locations in the Province of Utrecht



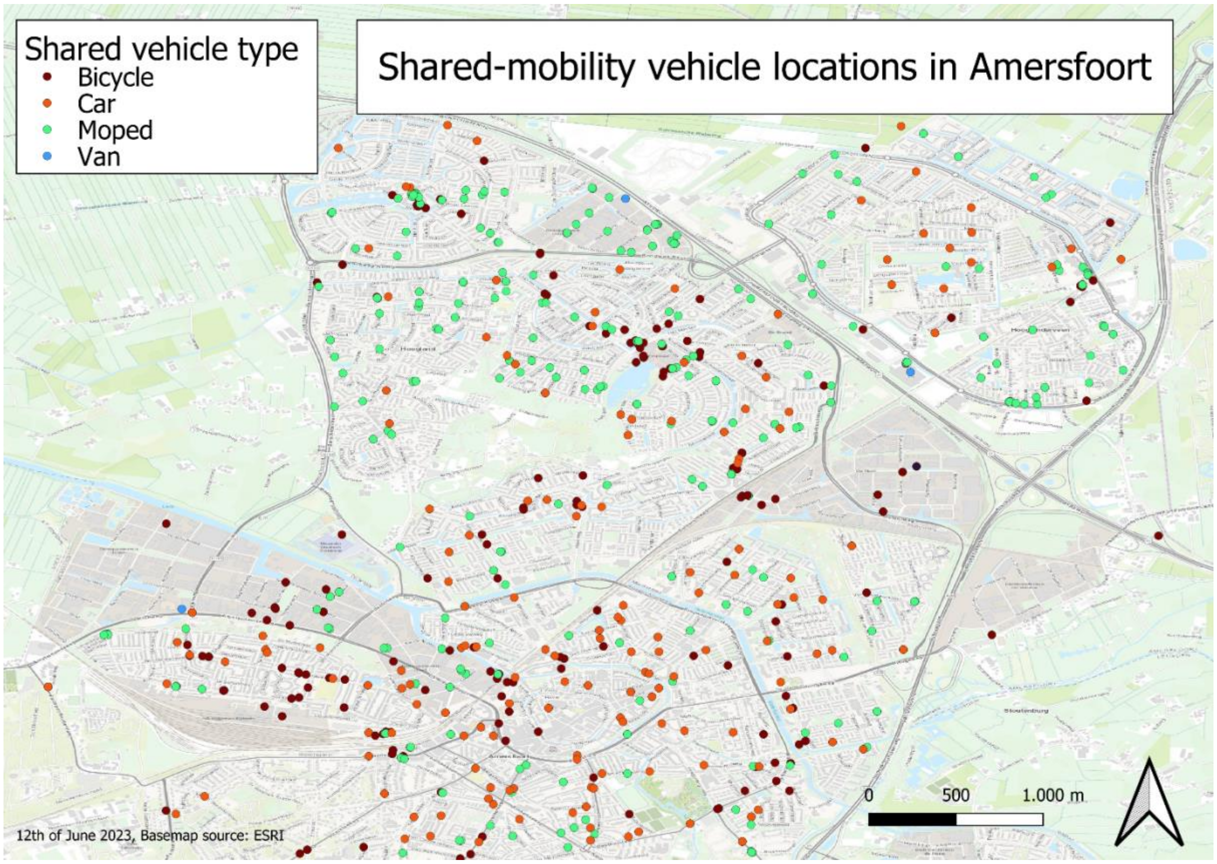
In the city of Utrecht, the shared-mobility options seem to be dominated by cars, bicycles and cargo bikes (Figure 7). Since 2022, Utrecht has removed all shared mopeds from its streets, and therefore they are not present on this map. Notable is the dispersion of the shared-mobility vehicles, as it seems like the vehicles are quite evenly distributed along the city, with some exceptions for pedestrian-, industrial-, and corporate areas.

Figure 7 Shared mobility vehicles in Utrecht



The situation in Amersfoort is quite different to the situation in the city of Utrecht (Figure 8). Amersfoort does not have cargo bikes, however, they do have mopeds. Interestingly, these moped can be found throughout the city, but the seem to dominate the neighborhoods further away from the city center. In general, the density of vehicles in Amersfoort also seems to be much lower than in Utrecht. Which is to be expected, as it is a smaller city.

Figure 8 Shared-mobility vehicle locations in Amersfoort



In table 16, the modalities have been split into sub-categories of 'return after rental'-type. Most of the data-points are related to the 'fixed base' model, predominantly consisting of cars. The other way around, most of the cars also use a 'fixed base' model. Only a small number of the fixed base data-points are related to (E-)bikes, cargo bikes, and vans.

In contrast, the 'free floating' model reveals a substantial number of data points primarily attributed to (E-)bikes. Furthermore, there is a notable presence of data points for mopeds in this category. The 'zone' model Is only used for cars, and the 'back-to-many' model currently only involves the movement of bicycles between designated bases.

Table 16 Total number of data-points per return type and modality

Return after rental type	Modality	Data-points count
Fixed Base	**Total**	**1628**
	Car	1550
	(E-)Bicycle	37
	Cargo bike	34
	Van	7
Free floating	**Total**	**1421**
	(E-)Bicycle	1128
	Moped	293
Zone	**Total**	**217**

	Car	217
Back-to-many	**Total**	**140**
	(E-)Bicycle	140

Again, when we look at the municipality of Utrecht and Amersfoort, we can see differences (figure 9 and figure 10). In Utrecht, the 'fixed base' is more dominant as opposed to Amersfoort, where there are more 'free floating' vehicles. This can be attributed to the mopeds that can be found throughout the city of Amersfoort. Due to the dominance of cars in more rural areas, the 'return after rental'-types here are dominated by the 'fixed base' model.

Figure 9 Return model of shared-mobility vehicles in the City of Utrecht

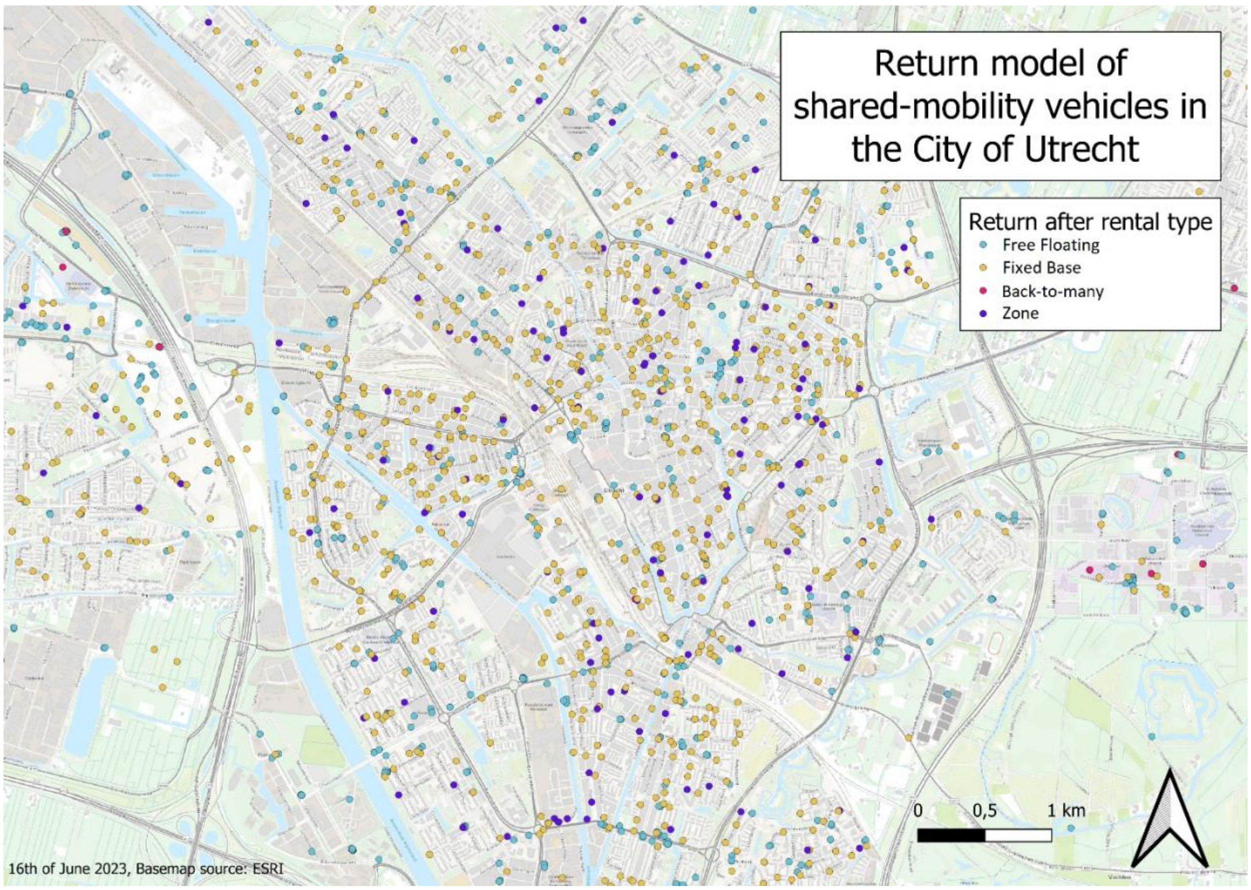
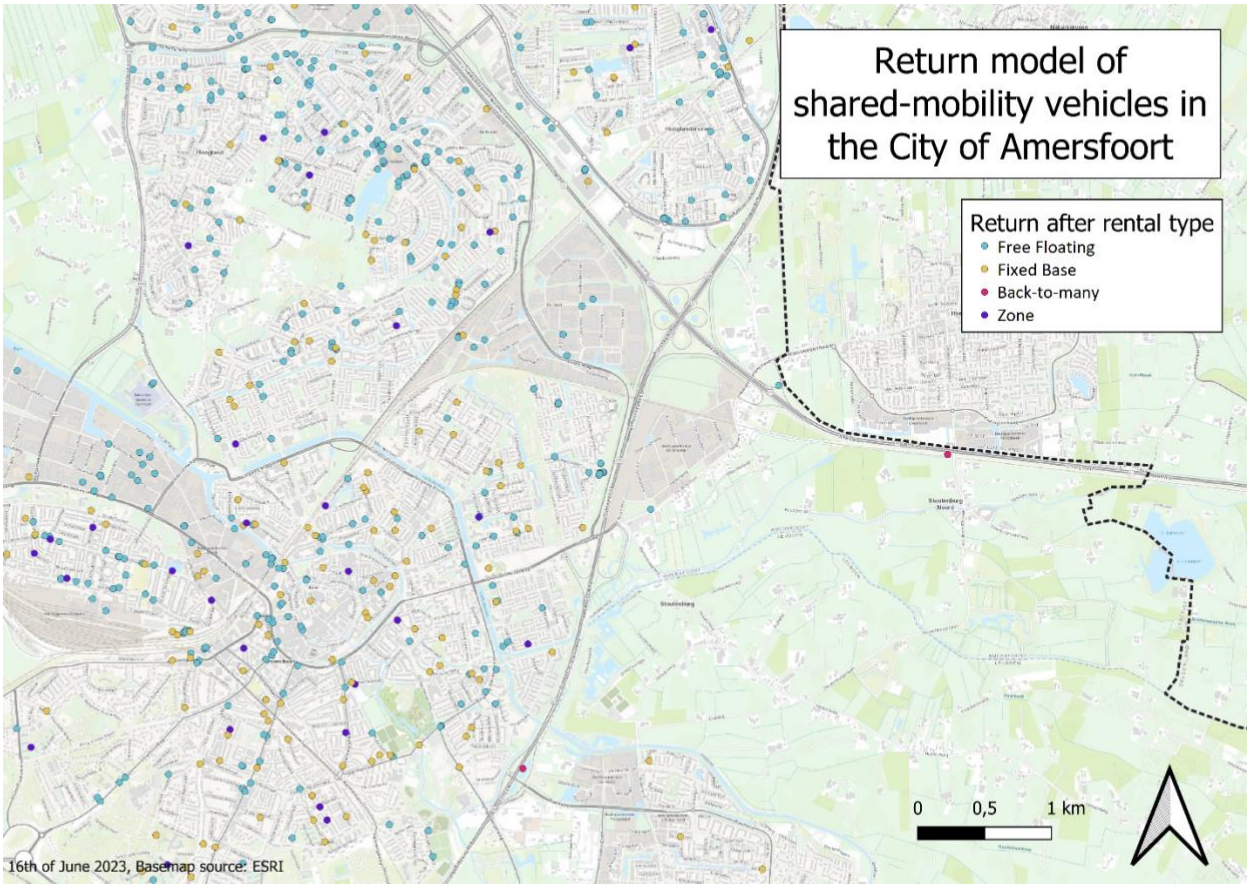


Figure 10 Return model of shared-mobility vehicles in the City of Amersfoort



4.2 Inventory of hubs

After performing the analysis, a total number of 100 hubs have been identified in the Province of Utrecht. The frequency table 17 shows the distribution of hub sizes in terms of numbers of unique modalities it holds. The are only two hubs with 5 unique modalities, furthermore, as the number of unique modalities decreases, the frequency increases.

Table 17 Distribution of hub sizes as of the total number of hubs

Number of unique modalities in hubs	Number of hubs
5	2
4	4
3	24
2	70
Total	**100**

The composition of the hubs can also vary. In table 18 the compositions and frequencies are depicted. The most common hub is the bus–car hub. Although the bicycle-car, bicycle-train-bus, and tram-bus hubs are also very common. 20 of the 100 identified hubs only have public-transportation options, meaning that in 80% of the cases there is at least one form of shared-mobility that is not public transport.

Table 18 Modality combinations and frequency of the identified hubs

Car	Bicycle	Moped	Cargo bike	Van	Train	Tram	Bus	Frequency
✓							✓	22
	✓						✓	17
	✓				✓		✓	17
						✓	✓	16
✓			✓				✓	8
	✓					✓	✓	5
					✓		✓	4
✓	✓				✓		✓	4
✓			✓					2
✓	✓				✓	✓	✓	2
	✓				✓			2
✓	✓				✓			1

Figure 11 shows a map of the Province of Utrecht and the locations of the mobility hubs. On the map, it shows that many of the mobility hubs are centered around the city of Utrecht, although some are also present in the smaller, more rural areas in the Province. Overall, mobility hubs can be found throughout the province.

Figure 11 Mobility hubs in the Province of Utrecht

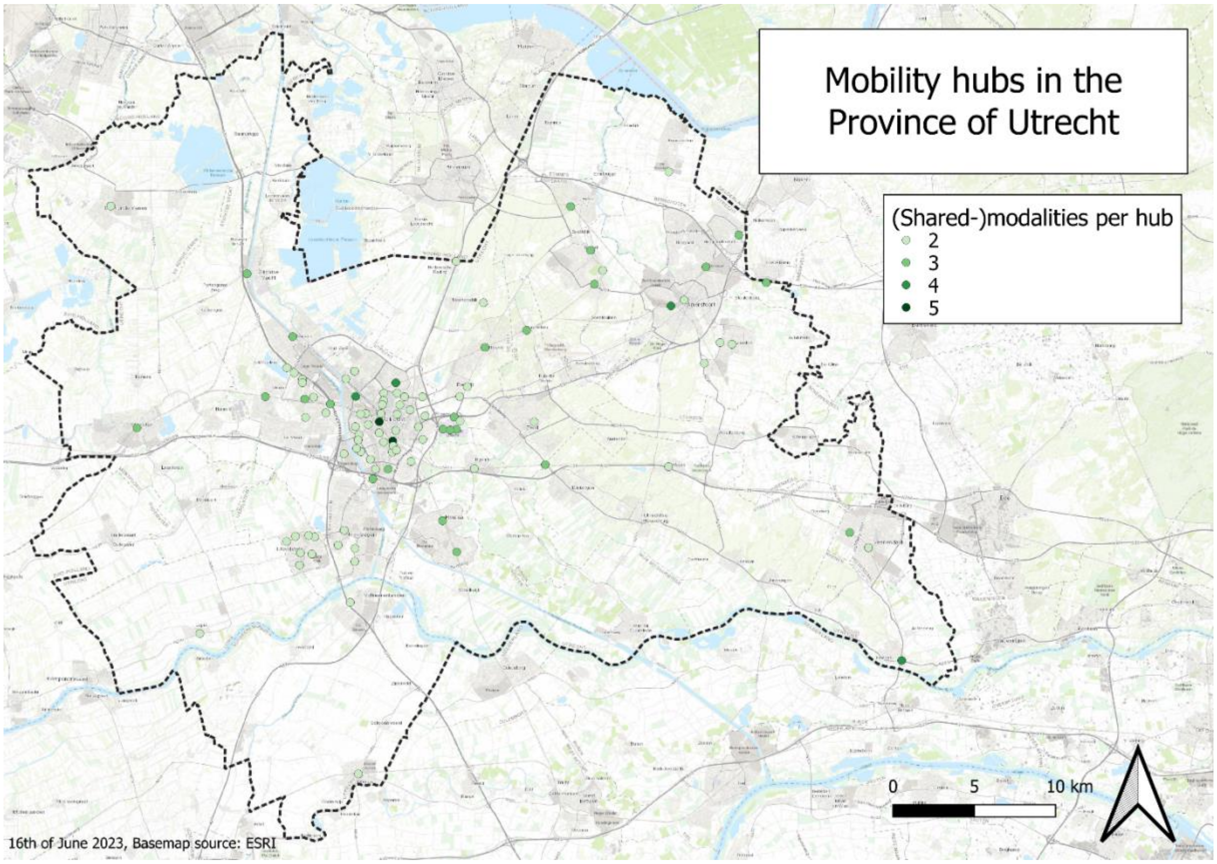
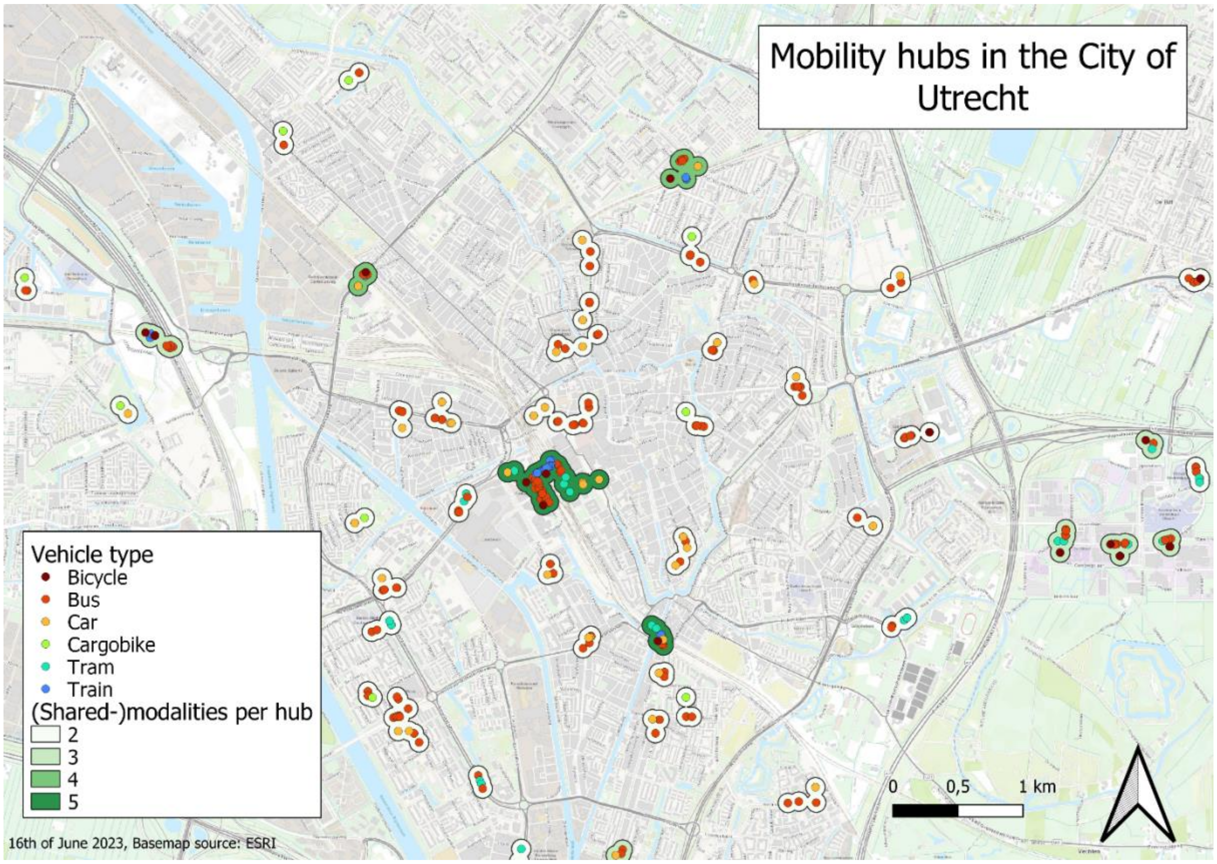


Figure 12 shows the two biggest mobility hubs in the province and its surrounding area. The mobility hubs are both centered around a train station, namely Utrecht Central Station and Utrecht Vaartsche Rijn. The station of Zuilen and Overvecht also have some of the bigger hubs. Other hubs that can be seen in the city center are also located in popular areas, such as Vredenburg, Neude, Stadsschouwburg and Ledig Erf.

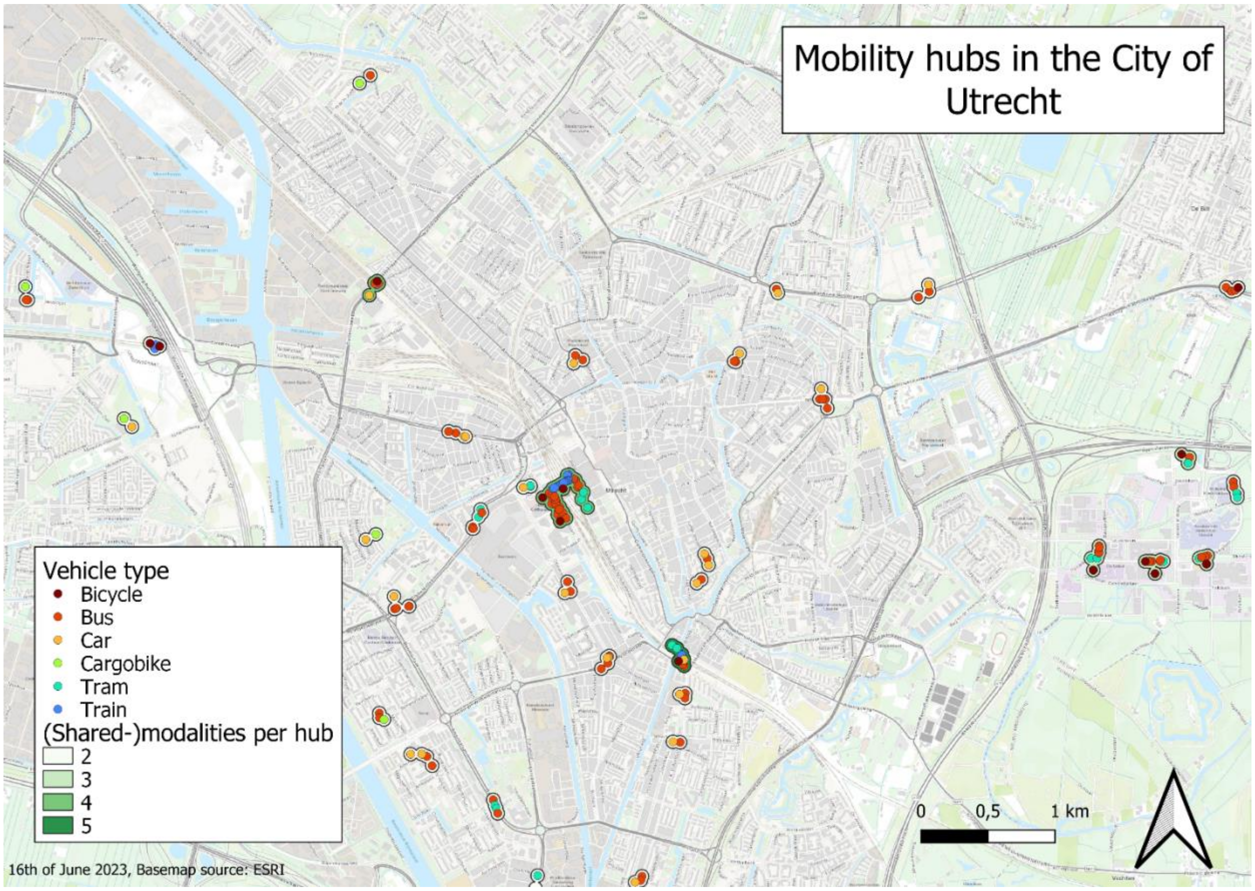
Figure 12 Mobility hubs in the City of Utrecht



4.2.1 Inventory of hubs with alternate scenarios

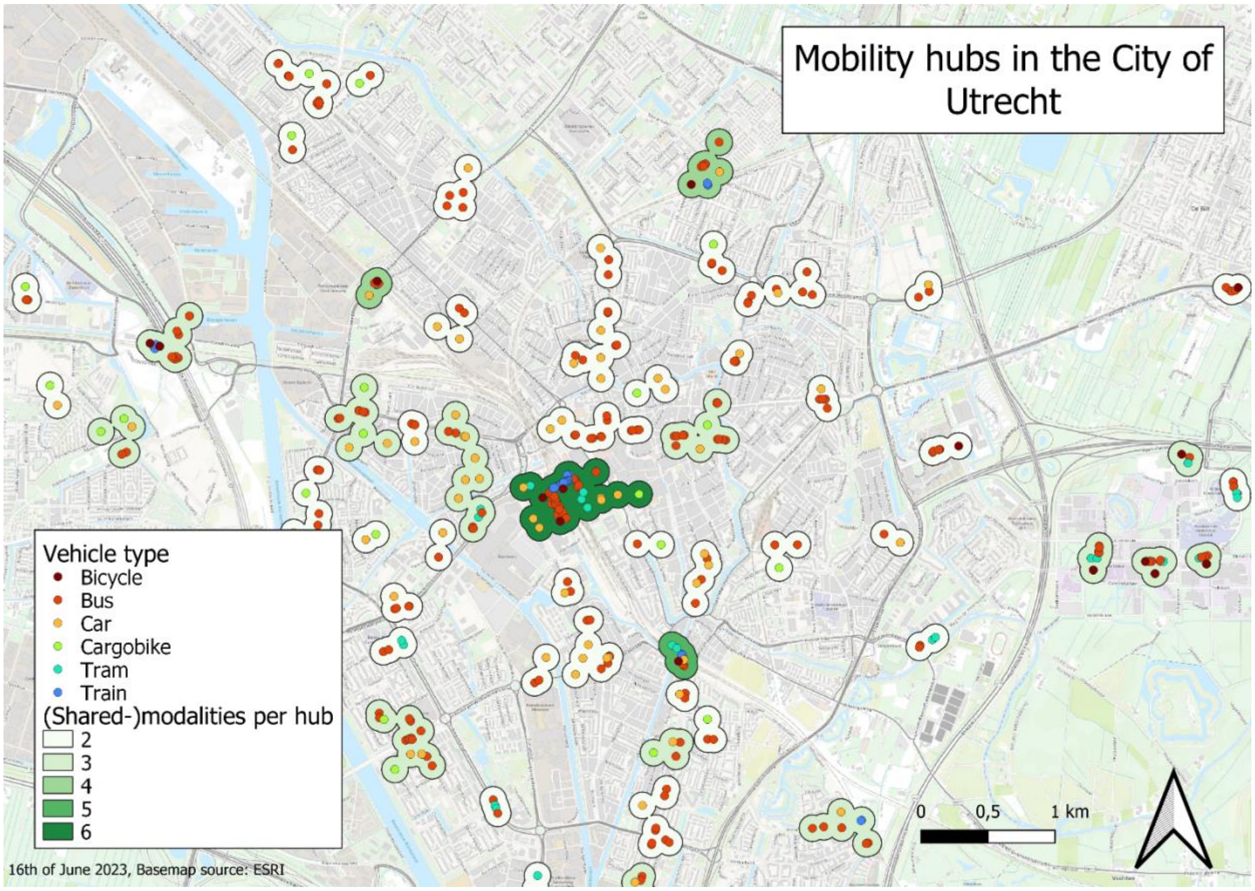
To gain an understanding of the influence of the chosen walkability, two alternative scenarios have been executed. First, instead of an acceptable walkability distance of 150 meters, a distance of 100 meters was used. As a consequence, the total number of hubs in the Province of Utrecht dropped from 100 to only 85. The situation in the city of Utrecht can be seen in figures 13. One of the most notable changes in this scenario is the dissapearance of one of the biggest mobility hubs in the city, namely the hub of Utrecht Overvecht, which had 4 shared-modalities in the previous scenario. Another effect is the split of the hub at Utrecht Centraal, which has become two separate hubs. Other than that, mostly small hubs have not been identified in this scenario compared to the original scenario.

Figure 13 Mobility hubs in the City of Utrecht using an acceptable walking distance of 100 meters



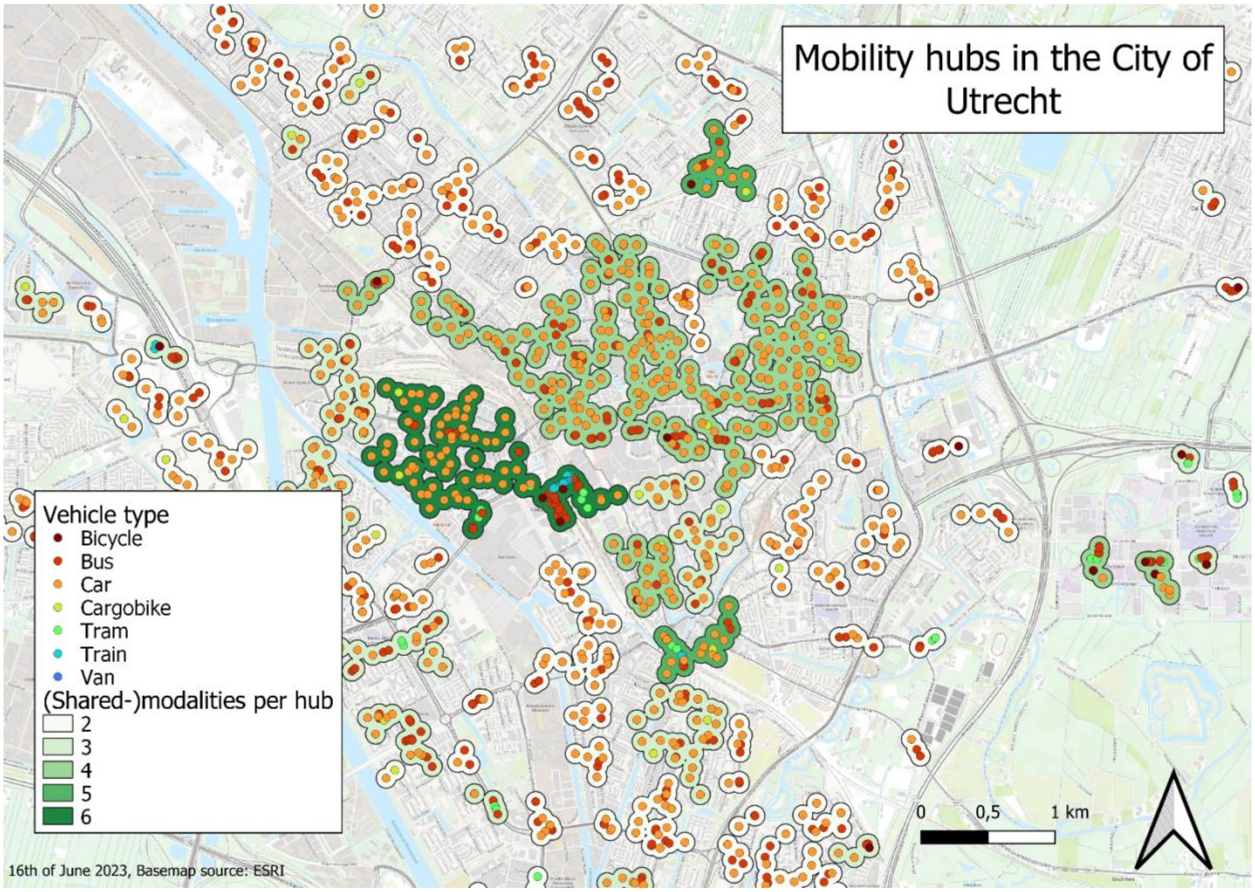
When using an acceptable walkability distance of 200m, 112 hubs have been identified in the Province of Utrecht. Besides an increase in the total number of hubs, some hubs have also grown significantly and even merged with hubs identified in the '150 meter scenario' (figure 14). Due to the method use, in some cases, modalities within the same hub are almost a kilometre apart. After comparing both the alternative scenario's, using 150 meters as an acceptable walking distance has been deemed the most effective.

Figure 14 Mobility hubs in the City of Utrecht using an acceptable walking distance of 200 meters



In order to validate the choices mentioned previously, regarding the filtering of car bases, the hub analysis has also been performed including all cars, and not just car bases with multiple cars. For this scenaria, an acceptable walking distance of 150 meters has been used. Figure 15 shows how this affects the formation of hubs. Because of the dispersion of cars, hubs are more often linked to one another, creating much more and much bigger hubs. However, this challenges the idea of a hub, as distances within hubs now exceed desirable walking distances.

Figure 15 Mobility hubs in the City of Utrecht including car bases with only one car



4.3 Limitations

A limitation to this analysis is the disregarding of free floating devices. Even though they make up a significant portion of the total shared-vehicle inventory, taking them into consideration for hub analysis has been proven difficult and have therefore been left out. If they could somehow be included in the analysis, it would give a more complete picture. Furthermore, the analysis only considers the total number of unique modalities, but does not take into account the tot number of vehicles or the number of possible public transport lines in the hub. Another limitation is that tram and bus are considered as being two distinct modalities, although it can be argued that both serve the same purpose as a regional public transportation provider. Finally, a definition have been given to determine the return models of various service providers (free floating, back-to-many, etc.). However, not all return models fit strictly into one category and might have some properties of multiple return model. For the purposes of this research, these service providers have been classified in the most suitable category.

Moreover, In this research, the analysis has been performed from a multimodal transportation perspective. In which it is assumed that all shared-mobility services can be used as a first- and last-mile solution. In practice, it is unclear whether this is also the case, especially considering cars or cargo bikes.

5. Conclusion

The results section provides the necessary findings to answer the research questions about the inventory of shared-mobility and mobility hubs in the Province of Utrecht. In the first section, an overview of the shared mobility vehicles in the province was provided, showing that cars are the most prevalent shared modality, followed by (e-)bikes and mopeds. Shared-mobility could be found throughout the province, although the level of urbanization did influence the available modality types. In the second part, results of the hub-analysis are discussed and reveal that a total of 100 hubs are identified, of which 80 have at least one form of shared-mobility options. The bus-car hub were the most common. The results illustrate the distribution and location of mobility hubs throughout the province, with a concentration around the city of Utrecht and some hubs in rural centers. Overall, the findings shed light on the landscape of shared mobility in the province, providing valuable insights for future research and urban planning initiatives.

As a follow-up on this research, and as part of the ShareDiMobiHub-project, research will be conducted on potential hub locations in the Province of Utrecht. In which, among others, results from this research will be used to identify the most suitable location for the expansion or creation of hub locations using a GIS-analysis approach. This document will then also be used to see if the identified most suitable hub locations are in line with the current state of affairs regarding shared mobility options.

6. Policy recommendations

Establishing shared mobility hubs in the Province of Utrecht is a vital part of its smart mobility strategy. These hubs should be strategically placed in key areas to fill existing gaps in the Provinces transportation network. This document can be utilized to get a better understanding of the spatial spread of shared mobility vehicles in the Province and can be a starting point for a strategy to fill the existing gaps in the transportation network. Improving the offer of shared mobility vehicles and establishing shared mobility hubs can provide a more accessible and inclusive region.

The analysis also identified shared mobility hubs which are not yet officially labelled as such. To improve the visibility and accessibility of these clustered shared mobility vehicles, an official hub location with uniform mobility hub signage should be established. This can be achieved through a synergy in public-private partnerships between the municipalities and shared mobility providers.

Finally, this document and the corresponding data-files should be consulted when evaluating potential hub locations. In the evaluation, information can be found about the modalities already present in the area and possible partnerships with the providers of these shared-vehicles. Consequently, it can also be used to see which shared-vehicle modes are still missing.

7. References

Cargoroo (2023). Hoe werkt de elektrische buurtbakfiets van Cargoroo. Retrieved on 21 June 2023 from https://cargoroo.nl/hoe-werkt-de-elektrische-buurtbakfiets-van-cargoroo/

CROW (2023). Dashboard Deelmobilitiet: Een project van CROW. Retrieved on 21 June 2023 from https://dashboarddeelmobiliteit.nl/

Donkey Republic (2023a). Explore the cities we are in. Retrieved on 21 June 2023 from https://www.donkey.bike/cities/

Donkey Republic (2023b). Never be without a bike: Donkey Republic. Retrieved on 21 June 2023 from https://www.donkey.bike/

Duran-Rodas, D., Navarro-Ávalos, F., Hall, J., Nichols, A., Büttner, B., Baguet, J. & Susilo, Y. (27 August, 2022). Guidelines for the integration of mobility hub into the urban space. SmartHubs Delivrable 3.1. Retrieved on 21 June 2023 from smartmobilityhubs.eu/_files/ugd/c54b12_c50885556b2d49108e90d0c70c777c07.pdf

GreenWheels (2023). Zo werkt autodelen met GreenWheels. Retrieved on 21 June 2023 from https://www.greenwheels.nl/zo-werkt-t

Hely (2022). Hely in Utrecht. Retrieved on 21 June 2023 from https://hely.com/utrecht?lng=en

JustGo (n.d.). JustGo in Utrecht. Retrieved on 21 June 2023 from https://www.justgo.nl/provincies/utrecht

Kav2Go (2023). Busje huren bij onze Pick up Points in heel Nederland. Retrieved on 21 June 2023 from https://kav2go.nl/

KeoBike (n.d.). Home: KeoBike. Retrieved on 21 June 2023 from https://www.keobike.nl/

MoveYou (October 22, 2021). GoAbout acquired by MoveYou. Retrieved on 21 June 2023 from https://moveyou.com/goabout-overgenomen-door-moveyou/

MyWheels (2023). MyWheels. Retrieved on 21 June 2023 from https://www.mywheels.nl/

OpenOV (2021). Index of /haltes/. Retrieved on 21 June 2023 from http://data.openov.nl/haltes/

Sixt (n.d.). Deelauto in Utrech: SIXT Share. Retrieved on 21 June 2023 from https://www.sixt.nl/share/deelauto/utrecht/#/?sxamc=Google|Search&fir=1&gclid=CjwKCAjwv8qkBhAnEiwAkY-ahgKi8mNUAxgGrKB2QuaBdni6ntSB74kbqW3tSW0KkgFZe5690lbXXBoCBowQAvD_BwE&exactag_uk=0f508c02e51141d2ae75e9e9667621aa

SnappCar (2023). How does SnappCar work? Learn all about car sharing. Retrieved on 21 June 2023 from https://www.snappcar.nl/hoe-werkt-het

We Drive Solar (2023). Hét Duurzame alternatief voor je eigen auto. Retrieved on 21 June 2023 from https://www.wedrivesolar.nl/

8. Appendices

Appendix A: OV_Fiets_API_to_CSV.py

```
#Somewhat static data, so no need to use a get request, but might do in the
future.
import json
import csv
from datetime import date
import requests

Date = date.today().strftime("%d-%m-%Y")
response = requests.get("https://ovfietsbeschikbaar.nl/data/map-locations")
data = response.json()

OVFiets = data

with open(f'JSON_API_Responses/{Date}/OV_Fiets_{Date}.json', 'w') as f:
    json.dump(data, f)

data_file = open(f"CSV_vehicle_locations/{Date}/OV_fiets_locations_{Date}.csv",
'w', newline='')

csv_writer = csv.writer(data_file)

count = 0

for location in OVFiets:
    if count == 0:

        header = ["service",
"modality","location","bike_count","longitude","latitude"]
        csv_writer.writerow(header)
        count += 1

    FietsLocation = location["location"]
    longitude = location["lng"]
    latitude = location["lat"]
    bike_count = location["text"].split(' ', 1)[0]
    csv_writer.writerow(["OVFiets", "bicycle", FietsLocation, bike_count,
longitude, latitude])

data_file.close()
```

Appendix B: CROW_API_to_CSV.py

```
import requests
import csv
import datetime
import json


# Posting a get request to deelfietsendashboard API with current time
Date = datetime.date.today().strftime("%d-%m-%Y")
now = datetime.datetime.now().strftime("%Y-%m-%dT%XZ")
response = requests.get(f"https://api.deelfietsdashboard.nl/dashboard-
api/public/vehicles_in_public_space?zone_ids=51233&timestamp={now}")
data = response.json()

CROW = data['vehicles_in_public_space']

with open(f'JSON_API_Responses/{Date}/CROW_{Date}.json', 'w') as f:
    json.dump(data, f)

data_file = open(f"CSV_vehicle_locations/{Date}/CROW_locations_{Date}.csv", 'w',
newline='')

csv_writer = csv.writer(data_file)

# Counter variable used for writing headers to the CSV file
count = 0

for location in CROW:
    if count == 0:
        # Writing headers of CSV file
        header = ['form_factor', 'latitude', 'longitude', 'system_id']
        csv_writer.writerow(header)
        count += 1

    # Writing data of CSV file
    form_factor = location['form_factor']
    latitude = location['location']['latitude']
    longitude = location['location']['longitude']
    system_id = location['system_id']
    if system_id == "keobike" or system_id == "cargoroo":
        continue
    csv_writer.writerow([form_factor, latitude, longitude, system_id])

data_file.close()
```

Appendix C: Cargoroo_API_to_CSV.py

```
import re
import csv
from datetime import date
import requests

Date = date.today().strftime("%d-%m-%Y")

response = requests.get("https://cargoroo.nl/cargoroo-cities/")
data = response.text

latitudes = re.findall('data-lat="\d+\.\d+', data)
longitudes = re.findall('data-lng="\d+\.\d+', data)

lat_list = []
for latitude in latitudes:
    latitude_temp = latitude.strip('data-lat="')
    lat_list.append(float(latitude_temp))
lng_list = []
for longitude in longitudes:
    longitude_temp = longitude.strip('data-lng="')
    lng_list.append(float(longitude_temp))

data_file = open(f"CSV_vehicle_locations/{Date}/Cargoroo_locations_{Date}.csv",
'w', newline = '')
csv_writer = csv.writer(data_file)

header = ['service', 'modality', 'latitude', 'longitude']
csv_writer.writerow(header)

for (latitude, longitude) in zip(lat_list, lng_list):
    csv_writer.writerow(['Cargoroo', 'cargobike', latitude, longitude])

data_file.close()
```

Appendix D: Keobike_API_to_CSV.py

```
import re
import csv
from datetime import date
import requests

Date = date.today().strftime("%d-%m-%Y")

response = requests.get("https://reizen.keolis.nl/nl/keobike/vind-een-fiets")
```

```
data = response.text

latitudes = re.findall('"latitude":\d+\.\d+', data)
longitudes = re.findall('"longitude":\d+\.\d+', data)

lat_list = []
for latitude in latitudes:
    latitude_temp = latitude.strip('"latitude": ')
    lat_list.append(float(latitude_temp))
lng_list = []
for longitude in longitudes:
    longitude_temp = longitude.strip('"longitude": ')
    lng_list.append(float(longitude_temp))

data_file = open(f"CSV_vehicle_locations/{Date}/Keobike_locations_{Date}.csv",
'w', newline = '')
csv_writer = csv.writer(data_file)

header = ['service', 'modality', 'latitude', 'longitude']
csv_writer.writerow(header)

for (latitude, longitude) in zip(lat_list, lng_list):
    csv_writer.writerow(['Keobike', 'bicycle', latitude, longitude])

data_file.close()
```

Appendix E: GreenWheels_JSON_to_CSV.py

```
#Greenwheels API uses protected POST request, not possible to send request from
remote server. Instead, get locations JSON from "inspect" -> "network" ->
"graphql" response page

import json
import csv
import datetime

Date = datetime.date.today().strftime("%d-%m-%Y")

# Opening JSON file and loading the data into the variable data
with open(f"JSON_API_Responses/{Date}/GreenWheels_{Date}.json") as json_file:
    data = json.load(json_file)

GreenWheels = data['data']['locations']
```

```
data_file =
open(f"CSV_vehicle_locations/{Date}/GreenWheels_locations_{Date}.csv", 'w',
newline='')

csv_writer = csv.writer(data_file)

# Counter variable used for writing headers to the CSV file
count = 0

for location in GreenWheels:
    if count == 0:

        # Writing headers of CSV file
        header = ['service','modality', 'id', 'city', 'longitude', 'latitude',
'car_count']
        csv_writer.writerow(header)
        count += 1

    # Writing data of CSV file
    id = location['id']
    city = location['city']['name']
    longitude = location['geoPoint']['lng']
    latitude = location['geoPoint']['lat']
    car_count = len(location['cars'])
    csv_writer.writerow(["GreenWheels", "car", id, city, longitude, latitude,
car_count])

data_file.close()
```

Appendix F: MyWheels_JSON_to_CSV.py

```
#MyWheels API uses protected POST request, not possible to send request from
remote server. Instead, get locations JSON from "inspect" -> "network" -> "/api"
response page

import json
import csv
import datetime

Date = datetime.date.today().strftime("%d-%m-%Y")

# Opening JSON file and loading the data into the variable data
with open(f"JSON_API_Responses/{Date}/MyWheels_{Date}.json") as json_file:
    data = json.load(json_file)
```

```
MyWheels = data['result']['results']

data_file = open(f"CSV_vehicle_locations/{Date}/MyWheels_locations_{Date}.csv",
'w', newline='')

csv_writer = csv.writer(data_file)

# Counter variable used for writing headers to the CSV file
count = 0

for location in MyWheels:
    if count == 0:

        # Writing headers of CSV file
        header = ['service','modality', 'id', 'city', 'longitude', 'latitude',
'parkingType']
        csv_writer.writerow(header)
        count += 1

    # Writing data of CSV file
    id = location['resource']['id']
    city = location['resource']['city']
    longitude = location['resource']['longitude']
    latitude = location['resource']['latitude']
    parkingType = location['resource']['parkingType']
    csv_writer.writerow(["MyWheels", "car", id, city, longitude, latitude,
parkingType])

data_file.close()
```

Appendix G: SnappCar_API_to_CSV.py

```
import requests
import csv
from datetime import date
from geopy.geocoders import Nominatim

#Set initial values for query
Date = date.today().strftime("%d-%m-%Y")
QueryLocation = "lat=52.09198321198723&lng=5.120756611230562" #Query location
(Neude = 52.09, 5.12)
Distance = "38000" #In meters (Prov Utrecht radius is around 50km)

#Initialize variables
Offset = 0
```

```
json_string = ""
count = 0
data_file = open(f"CSV_vehicle_locations/{Date}/Snappcar_locations_{Date}.csv",
'w', newline = '')
csv_writer = csv.writer(data_file)
geolocator = Nominatim(user_agent="Geolocator")


#While loop, as every requests queries a max of 10 results
while True:
    print("Running... this one might take a while...")
    response =
requests.get(f"https://api.snappcar.nl/v2/search/query?{QueryLocation}&fuzzy=fals
e&instant-bookable=false&max-
distance={Distance}&sort=recommended&order=desc&country=NL&show-
sums=true&limit=10&snapprankBucket=A&offset={Offset}")
    if response.status_code == 200:
        Offset += 10
        json_dict = response.json()
        query_results = json_dict['results']
        for result in query_results:
            if count == 0:

                # Writing headers of CSV file
                header = ['service', 'modality', 'id', 'city', 'street',
'latitude', 'longitude', 'country_code']
                csv_writer.writerow(header)
                count += 1

            # Writing data of CSV file
            id = result['ci']
            city = result['car']['address']['city']
            if city == "Amsterdam" or city == "Almere" or city == "Hilversum" or
city == "Gouda" or city == "Dordrecht":
                continue
            street = result['car']['address']['street']
            location = geolocator.geocode(f"{street}, {city}")
            if location == None:
                continue
            lat = location.latitude
            lng = location.longitude
            country_code = result['car']['address']['countryCode']
            csv_writer.writerow(["SnappCar", "car", id, city, street, lat, lng,
country_code])
    else:
```

```
        break

data_file.close()
```

Appendix H: WeDriveSolar_KML_to_CSV.py

```
from pykml import parser
import csv
from datetime import date

Date = date.today().strftime("%d-%m-%Y")

# Open the KML file and parse it
with open(f'JSON_API_Responses/{Date}/WeDriveSolar_{Date}.kml', 'r') as f:
    doc = parser.parse(f).getroot()

data_file =
open(f"CSV_vehicle_locations/{Date}/WeDriveSolar_locations_{Date}.csv", 'w',
newline = '')
writer = csv.writer(data_file)
header = ['service', 'modality', 'car_count', 'longitude', 'latitude', 'z']
writer.writerows([header])
for pm in doc.Document.Folder.Placemark:
    coords = pm.Point.coordinates.text.strip()
    CSVRows = coords.split(",")
    CSVRows.insert(0, 'WeDriveSolar')
    CSVRows.insert(1, 'car')
    try:
        description = pm.description.text.strip().lower()
        car_count = description.count('zoe') + description.count('tesla') +
description.count('kona') + description.count('2x') + description.count('ioniq')
+ description.count('meerdere')
        if car_count == 0:
            car_count = 1
    except:
        car_count = 1
    CSVRows.insert(2, car_count)
    writer.writerows([CSVRows])
```

Appendix I: Merge_CSV.py

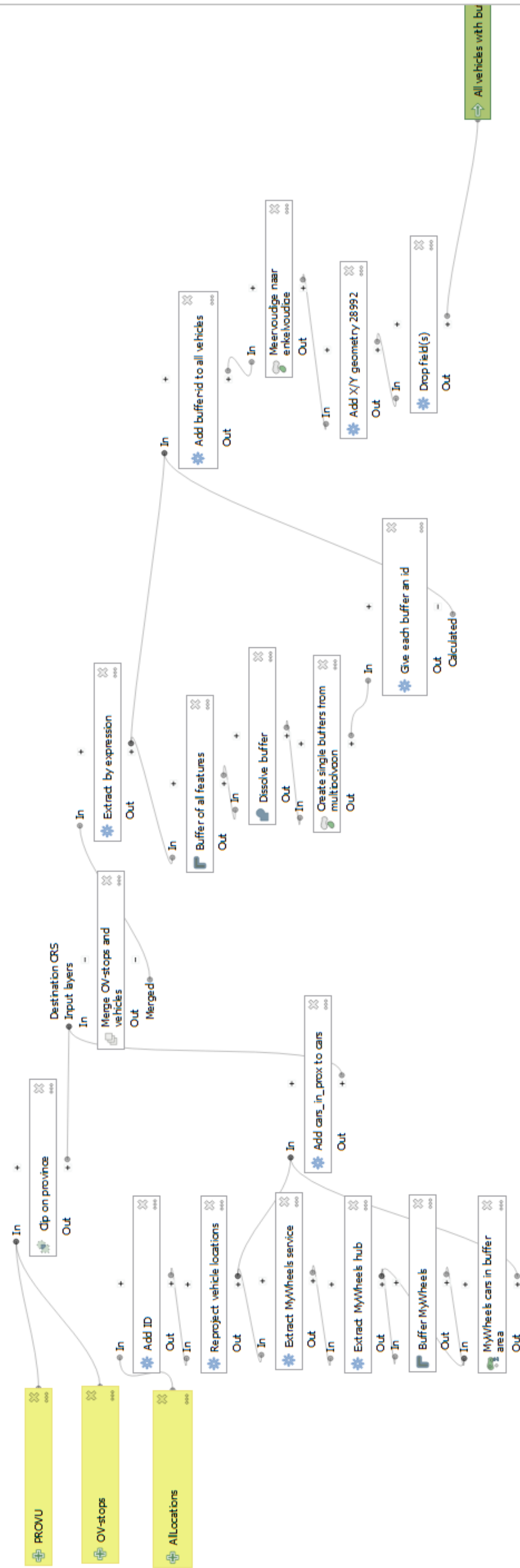
```
import pandas as pd
import os
import re
import numpy as np
```

```
from RayCastingAlgorithm import contains
from PROVU_Coords import PROVU_Coords
from datetime import date

class Point:
    def __init__(self, x, y):
        """
        A point specified by (x,y) coordinates in the cartesian plane
        """
        self.x = x
        self.y = y

Date = date.today().strftime("%d-%m-%Y")

directory = f'CSV_vehicle_locations/{Date}/temp'

# Regex: detect csv file with format YYYY-mm-dd in name
DateFormat = "((?:19|20)\\d\\d)-(0?[1-9]|1[012])-([12][0-9]|3[01]|0?[1-9]).csv"

# Create an empty DataFrame to store the merged data
merged_df = pd.DataFrame()

# Loop through each CSV file in the directory and merge the data
for file in os.listdir(directory):
    if file.endswith(f"{Date}.csv"):
        print(f"Merging {file}")
        # Read the CSV file
        df = pd.read_csv(os.path.join(directory, file), encoding_errors =
'ignore', on_bad_lines='skip')
        # Add a column with the name of the service provider
        provider = re.search(r'^[^_]+', file).group()
        df['data_provider'] = provider
        # Make longitude and latitude columns universal across loaded df's
        if 'lng' and 'lat'in df:
            df.rename(columns={'lng': 'longitude', 'lat': 'latitude'},
inplace=True)
        if 'X' and 'Y' in df:
            df.rename(columns={'X': 'longitude', 'Y': 'latitude'}, inplace=True)
        if 'x' and 'y' in df:
            df.rename(columns={'x': 'longitude', 'y': 'latitude'}, inplace=True)
        if 'system_id' in df:
            df.rename(columns={'system_id': 'service'}, inplace=True)
        if 'form_factor' in df:
            df.rename(columns={'form_factor': 'modality'}, inplace=True)
```

```
        #Add hubtype based on service value
        conditionsHubType = [
        (df['service'] == "bird") | (df['service'] == "check") | (df['service']
== "dott") | (df['service'] == "gosharing") | (df['service'] == "tier") |
(df['service'] == "cargoroo"),
        (df['service'] == "Cargoroo") | (df['service'] == "GreenWheels") |
(df['service'] == "JustGo") | (df['service'] == "MyWheels") | (df['service'] ==
"OVFiets") | (df['service'] == "SnappCar") | (df['service'] == "WeDriveSolar") |
(df['service'] == "Hely") | (df['service'] == "SixtSharing") | (df['service'] ==
"Kav2Go"),
        (df['service'] == "donkey") | (df['service'] == "keobike") |
(df['service'] == "moveyou") | (df['service'] == "Keobike")]

        valuesHubType = ['freefloating', 'hub', 'hubtohub']

        df['hubtype'] = np.select(conditionsHubType, valuesHubType)
        if "MyWheels" in file:
            df['hubtype'] = np.where((df['service'] == 'MyWheels') &
(df['parkingType'] == "zone"), 'zone', df['hubtype'])

        #Add vehicle_count based on service
        conditionsVehicleCount = [
        (df['service'] == "bird") | (df['service'] == "check") | (df['service']
== "dott") | (df['service'] == "gosharing") | (df['service'] == "tier") |
(df['service'] == "cargoroo") | (df['service'] == "Cargoroo") | (df['service'] ==
"MyWheels") | (df['service'] == "SnappCar") | (df['service'] == "donkey") |
(df['service'] == "keobike"),
        (df['service'] == "Hely") | (df['service'] == "SixtSharing") |
(df['service'] == "moveyou") | (df['service'] == "Keobike")
        ]

        valuesVehicleCount = ['1', '>1']

        df['vehicle_count'] = np.select(conditionsVehicleCount,
valuesVehicleCount)

        if "WeDriveSolar" in file or "GreenWheels" in file or "JustGo" in file or
"Kav2Go" in file:
            df['vehicle_count'] = df['car_count']
        elif "OV_fiets" in file:
            df['vehicle_count'] = df['bike_count']

        #Add Boolean statement wheter it is in ProvUtrecht
        for index, row in df.iterrows():
```

```
            df.at[index, 'inUtrecht'] = contains(PROVU_Coords,
Point(df['longitude'][index], df['latitude'][index]))

        # Merge the data with the existing data
        merged_df = pd.concat([merged_df,
df.loc[:,["service","data_provider","longitude","latitude","modality","hubtype","
inUtrecht","vehicle_count"]]])
        merged_df_Utrecht = merged_df[(merged_df['inUtrecht'] == True)]
        merged_df_Utrecht = merged_df_Utrecht.drop(['inUtrecht'], axis=1)

# Write the merged data to a new CSV file
merged_df.to_csv(f'{directory}/all_vehicle_locations.csv')
merged_df_Utrecht.to_csv(f'{directory}/all_vehicle_locations_Utrecht.csv')
```

Appendix J: Model used for defining hubs in QGIS with Model Designer



9. The ShareDiMobiHub Consortium

The consortium of ShareDiMobiHub consists of 13 partners and 4 subpartners with multidisciplinary and complementary competencies. This includes European cities and regions, universities, network partners and transport operators.

For further information please visit https://www.interregnorthsea.eu/sharedimobihub