Cartography M.Sc.

Master thesis

Spatio-temporal public transport accessibility analysis and benchmarking in an interactive WebGIS

Majk Shkurti

Technical University of Munich

Technische Universität Wien – Vienna University of Technology

Technische Universität Dresden

University of Twente – ITC

2022

Spatiotemporal public transport accessibility analyses and benchmarking in an interactive WebGIS

submitted for the academic degree of Master of Science (M.Sc)
conducted at the Department of Aerospace and Geodesy
Technische Universität München

Author:	Majk Shkurti
Study Course:	Cartography M.Sc
1st Supervisor:	Liqiu Meng, Prof. Dr. (TUM)
2nd Supervisor:	Sebastian Seisenberger, M.Sc (TUM)
Reviewer:	Karl Rehrl, Dr. (Salzburg Research)
Chair of the Thesis Assessment Board:	Liqiu Meng, Prof. Dr. (TUM)
Date of submission:	10.10.2022

Statement of Authorship

Herewith, I declare that I am the sole author of the submitted Master's thesis entitled:

"Spatiotemporal public transport accessibility analyses and benchmarking in an interactive WebGIS"

I have fully referenced the ideas and work of others, whether published or unpublished
Literal or analogous citations are clearly marked as such.

Munich, 10.10.2022 Majk Shkurti

Acknowledgments

First and foremost, I would like to express my deep and sincere gratitude to Elias Pajares for inspiring, motivating, and giving me incredible support during my studies and walking me through the thesis process. Without him, I would not have been able to accomplish this study.

Secondly, I would also like to thank my supervisor M.Sc. Sebastian Seisenberger for his continuous feedback and for guiding me in the right direction.

A special thanks go to all the professionals who attended my workshop and usability testing. Their comments and ideas were very valuable and significantly helped me to have a better overview from a different perspective.

I would like to take this opportunity to also thank my friends who helped me throughout this difficult journey and for the beautiful memories that I will remember forever.

Finally, I would like to thank my family for believing in me and for the unwavering support and encouragement they gave me during my studies.

Abstract

Assessing public transport using the concept of accessibility is of rising importance in the planning practice. Meanwhile, the ever-increasing technological progress in GIS, web technologies, and the rising availability of data facilitate fast innovation. Accordingly, analyses that previously were done using conventional desktop software can be realized through comparatively easy web tools. The accessibility instrument GOAT is positioned within this scope, though to date, mainly analyses for active mobility are possible.

Therefore, this master thesis aims to study the integration of accessibility analyses for public transport into the tool. Within this process, suitable accessibility measures should be identified that can be integrated into a WebGIS. A suitable pre-selection of indicators was assessed based on the criteria of institutionalization in practice, data availability, technical complexity, and indicator complexity. In the second step, the pre-selected indicators were discussed in an online workshop with experts from planning practice. Finally, three indicators were selected as the most promising ones and were implemented into the GOAT tool.

The technical development made us of the novel routing algorithm (R5) and grid-based data structure that allow dynamic rendering of indicators such as isochrones. Furthermore, the development made use of diverse frontend and backend technologies. Finally, it is concluded that the realized development is a first step to bringing elaborated accessibility analyses for public transport into a WebGIS and finally into practice. However, there is still the need to test further with practice and iteratively optimize the solution.

Kurzfassung

Die Bewertung des öffentlichen Verkehrs anhand des Konzepts der Erreichbarkeit gewinnt in der Planungspraxis zunehmend an Bedeutung. Der stetige technologische Fortschritt bei GIS und Web-Technologien sowie die zunehmende Verfügbarkeit von Daten ermöglichen schnelle Innovationen. So können Analysen, die bisher mit herkömmlicher Desktop-Software durchgeführt wurden, mit vergleichsweise einfachen Web-Tools realisiert werden. Das Erreichbarkeitsinstrument GOAT ist in diesem Umfeld zu verorten, wobei bisher hauptsächlich Analysen für die aktive Mobilität möglich waren.

Ziel dieser Masterarbeit ist es daher, die Integration von Erreichbarkeitsanalysen für den öffentlichen Verkehr in das Werkzeug zu untersuchen. Dabei sollen geeignete Erreichbarkeitsmaßnahmen identifiziert werden, die in ein WebGIS integriert werden können. Eine geeignete Vorauswahl von Indikatoren wurde anhand der Kriterien Institutionalisierung in der Praxis, Datenverfügbarkeit, technischer Aufwand und Komplexität der Indikatoren bewertet. Im zweiten Schritt wurden die vorausgewählten Indikatoren in einem Online-Workshop mit Experten aus der Planungspraxis diskutiert. Schließlich wurden drei Indikatoren als die vielversprechendsten ausgewählt und in das Tool GOAT implementiert.

Bei der technischen Entwicklung wurden der neuartige Routing-Algorithmus (R5) und die grid-basierte Datenstrukturen verwendet, die eine dynamische Darstellung von Indikatoren wie Isochronen im Frontend ermöglichen. Außerdem wurden bei der Entwicklung verschiedene Frontend- und Backend-Technologien eingesetzt. Abschließend wird festgestellt, dass die realisierte Entwicklung ein erster Schritt ist, um ausgefeilte Erreichbarkeitsanalysen für den öffentlichen Verkehr in ein WebGIS und schließlich in die Praxis zu bringen. Es besteht jedoch noch die Notwendigkeit, die Lösung weiter in der Praxis zu testen und iterativ zu optimieren.

Contents

1. Introduction

1.1. Background and motivation

Attractive public transport is essential to transition towards sustainable and carbon-neutral mobility in urban areas. A high public transport quality unifies a wide range of benefits. Public transport can be considered energy and space efficient compared to private motorized transport [1]. Furthermore, attractive public transport can foster economic growth [2]. Convinced by this wide range of benefits, many cities across the globe want to expand their public transport network and foster intermodal integration with other modes of transportation.

To measure the quality of existing and the effects of public transport expansions, the concept of accessibility plays a major role. Hansen [3] defined accessibility as *"the potential of opportunities for interaction"*. Specifically for public transportation, a wide range of indicators can be used to model and visualize accessibility. In accessibility research, accessibility indicators are usually referred to as accessibility measures [4], and a high diversity exists. Within this thesis, both accessibility measures and just indicators will be used.

This study differentiates between indicators that measure accessibility to public transport stops and accessibility by public transport. Accessibility is usually modeled using GIS methods and different routing algorithms. The relevant accessibility measures vary in complexity, computation time, and practical relevance [5]. While indicators like travel time isochrones are widely understood, others are more complicated and might be black boxes to non-experts.

Tools to model accessibility are usually referred to as accessibility instruments. Usually, they are based on different GIS technology [6]. With the progress of modern web technology, there is an increasing number of accessibility instruments developed as WebGIS applications [7]. Despite the fast technological development, implementing accessibility analyses into a WebGIS remains difficult. Long computation times of many indicators and the appropriate visualization of complex geospatial data are still challenging. Furthermore, the spatio-temporal dynamics of public transport require flexible visualization. In this scope there is seen particular need first to identify relevant accessibility measures for the planning practice and integrate them into a WebGIS environment.

1.2. Research objectives

This applied research aims to work on the described open challenges. Therefore, the development will be carried out using the open-source accessibility instrument Geo Open Accessibility Tool (GOAT) [8]. There are seen in particular the following research objectives.

Identify suitable benchmarks
The high diversity of accessibility measures in the public transport field makes it a challenge to identify the most relevant ones, while considering the technical feasibility of integrating them in a web-based GIS environment. Therefore, the first aim is to provide an overview of frequently used accessibility measures in Germany and other European countries. In the next step the accessibility measures will be classified using the criteria: institutionalization, data availability, technical complexity, and indicator complexity. As institutionalization, it is understood whether the indicator is already used in formal planning processes. For data availability studies whether it is easy to find needed data to apply the indicator in practice. With technical complexity, it is understood whether it is feasible to integrate the indicator in a web environment from a software development perspective. Within indicator complexity, it will be studied whether the indicator is easy to understand by the foreseen user group.

Develop backend framework for interactive public transport benchmark analyses
After suitable indicators are identified the backend logic for their computation of them will be developed within an isolated backend environment. Accordingly, the analyses can be used within the tool GOAT but also beyond in other applications and purposes. Therefore, it is aimed to combine different backend technology and use the novel routing algorithm R5. The different accessibility measures will be summarized in a backend framework that will provide public transport benchmark analyses. Furthermore, the proposed solution should be expandable for new features

such as scenario building (e.g., new public transport line) and new indicators.

Integration of public transport benchmarks into the dynamic user interface
The current user interface of GOAT focuses on analyses on the local scale and the transport modes of walking and cycling. The implementation of public transport benchmarks should be flexible to allow different analysis types that integrate into the existing tool architecture and user flow. A particular challenge is the temporal dynamics of public transport; therefore a solution is needed that does include the computation of indicators at various times of day and weekdays. Furthermore, it is aimed to increase the interactivity and granularity of the calculation. In most of the existing implementations, the isochrone is visualized as arbitrary polygons created in the backend using algorithms like concave or convex hulls, therefore the calculation intervals are fixed and cannot be modified in the frontend of the calculation without redoing the calculation. So if the user prefers computing another interval a new calculation, including eventually long calculation times needs to be triggered. Therefore, it aims to find a method to create the isochrone shape in the client of the application based on the routing results processed in the backend.

Scope Limitations of the study
One core limitation of the study is that it focuses on existing accessibility measures from the European context. Meanwhile, some of them might be used in a similar form in other parts of the world. Another limitation is that the study uses the tool GOAT as an example for the implementation of the indicators. Accordingly, the findings cannot be generalized in all cases. However, it is aimed to develop the components in a flexible architecture so they can be integrated into other accessibility instruments. Accordingly, the open source tool GOAT should be seen as a technical framework to test the realized innovations.

In the scope of this research project in particular the following research questions should be answered.

RQ 1: Which spatio-temporal public transport benchmarks can integrate into a web-based accessibility instrument considering practice relevance, technical feasibility and data availability?

RQ 2: How to compute and visualize spatio-temporal public transport benchmarks in a web-based accessibility instrument?

1.3. Thesis structure

The master thesis consists of five chapters. It starts with Chapter 1 that provides an overview of the background and gives insights into the authors' motivation. Furthermore, the research objectives and research questions are formulated. Chapter 2 gives an overview of the state-of-the-art in accessibility research and briefly describes existing routing algorithms.

In Chapter 3 an overview of the methodology being used in the study is presented. It starts with describing the tool GOAT, and the methodological and technical environment for the development carried out. Afterward, the data used in this study and the most critical data preparation workflows are presented. In the following, the methods used for software development are presented. Finally, the chapter presents the methods for involving experts and assessing the studied accessibility measures.

Within Chapter,4 the core results of the master thesis are presented. It will be started by explaining the assessment of the different accessibility measures. This part will be followed by the presentation of the developed software architecture. The most extensive part of the results will be the presentation of the development of the integrated accessibility measures. The last Chapter 5 will summarize the core conclusions and give an outlook on potential future developments.

2. Theoretical Background

2.1. Accessibility theory

The concept of accessibility was defined for the first time by [3] as the *"potential of opportunities for interaction"*. More recently, Brömmelstroet, Curtis, Larsson, and Milakis [9] defined accessibility as *"an expression of the potential number of relevant activities that are located within acceptable reach"*. Furthermore, K. Geurs and Wee [10] defined four accessibility components, namely the transport, land-use, temporal and individual components.

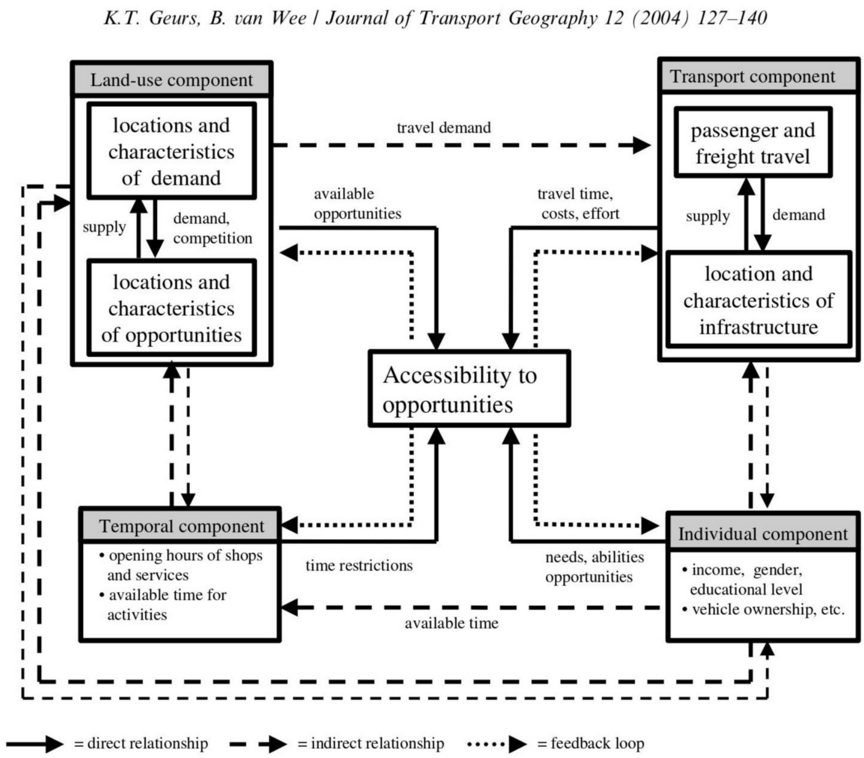


Figure 2.1.: Four accessibility components
[10]

As seen in Figure 2.1 all four components define the accessibility to opportunities. Accessibility can be translated by indicators usually named accessibility measures [10]. While literature suggests incorporating all four accessibility comments into an accessibility measure [10], frequently only the land-use and transport components are included. Accessibility measures are usually included in an accessibility instrument, which is capable of measuring and visualizing accessibility. Usually, accessibility instruments are GIS-based and were defined by [6] like the following:

> "Accessibility instruments (...) are a type of planning support system (PSS) designed to support integrated land-use transport analysis and planning (...)"

Despite the potential benefits of using accessibility instruments, they are frequently not used in practice now [5, 11]. One core challenge of this implementation gap is described as the *"disconnect between the worlds of instrument developers (who aim for scientific rigor and have an abstract understanding of the planning problem and process) and potential users (who aim for direct relevance, with the complexity of the real world as a starting point, and often are antagonistic towards sophisticated external technologies)"*.

2.2. Routing algorithms

There is an existing high variety of routing algorithms that are frequently used to calculate the shortest distances between one or several locations. A common approach when finding the shortest path is using the Dijkstra algorithm, which can be applied to all graphs represented through nodes and edges. The way that Dijkstra works is by selecting the unvisited node that has the lowest distance, calculating the distance through it to each unvisited neighbor, and updating the neighbor's distance if smaller. However, there are several disadvantages to this approach when applied to public transport networks, mostly related to the different data structures that public transport networks have. By definition, public transport routing is the process of determining the lowest travel costs between potential places of interest. The travel cost is typically represented by the travel time, and it is affected by different factors such as time of the day, travel modes, or network path types. [12]
As mentioned above, while many routing algorithms like Dijkstra can be used for a single Origin-Destination (O-D) pair, this is not feasible for calculating public

transport isochrone on a transit network that includes multiple (O-D) without compromising the computation time. To overcome this issue, the core algorithm used for public transport isochrone routing is RAPTOR, a round-based public transit router that, when employing frequency-based GTFS data, enables a systematic and efficient sampling of multiple simulated schedules. [13] . Unlike other graph algorithms, this algorithm processes each route (e.g., rail) in rounds. It starts by searching from a list of access stops and arrival times. All trips from these stops are explored for each round, so transfer from all reached stops. This algorithm allows further optimization by applying different rules and using multiple computation cores, making it very efficient for dynamic isochrone calculation. Conveyal R5 engine works on or Multi-criteria Range Raptor (McRR), an extended version of the base RAPTOR algorithm that iterates over minutes of the selected time window and considers more criteria besides arrival time and transfers [13].

2.3. Web map based visualization

Web mapping and Geographic Information Systems (GIS) have developed quickly in recent years. Several decades ago, only desktop applications mainly implemented GIS, which stored data in files using proprietary formats. It can be said that this technology was oriented toward experts. However, in the last decade, this situation has changed. As a result, many technologies, such as relational data and web applications, have adopted GIS components to provide geographic information to a broader number of users. Online maps are now standardized using OGC services and are fully supported in all browsers through technologies such as WebGL, Canvas2D, SVG, and CSS. These technologies are the foundation of web mapping and allow the creation or rendering of geospatial data on the web. The web map components consist of four core elements:

- Javascript map objects
- Base Layers
- Thematic Layer or System Layers
- Interaction

These elements are usually utilized through mapping libraries that abstract the complexity and provide a workspace that facilitates the development workflow. Another important aspect is providing geospatial data to the layers mentioned above. Nowadays, it is feasible to provide the data by using different formats, which can include:

- Programmable Objects (GeoJSON, PBF, XML)
- Database tables (PostGIS, MongoDB)
- Geospatial Web services OGC (WMS, WFS, WMTS, WPS)
- File Based (Shapefile, GeoJSON, KML, GML)

The selection of the data format depends on the requirements of the web map application. Regarding this project, the objective is to build an interactive WebGIS with dynamic data, in which data is provided in programmable objects served through custom Web Services. This part is described in more detail in chapter 4.

3. Methodology

3.1. GOAT - Geo Open Accessibility Tool

The accessibility instrument GOAT was started by the Chair of Urban Structure and Transport Planning at TUM and is actively developed by Plan4Better GmbH. GOAT can be understood as a web-based accessibility instrument. It targets users from local governments such as municipalities and counties, consultancies, and public transport operators. The tool can be understood as a specialized application that positions between an easy-to-use WebGIS and a fully featured desktop GIS [7]. The GOAT mainly focuses on modeling local accessibility for walking and cycling transport modes. GOAT is mainly focusing on modeling planning scenarios such as modification of transport infrastructure or changes in land-use. The tool was developed together with planning practice in a co-creative environment [14].

The extension realized in this thesis is integrated into the project GOAT 3.0, which is an applied research project of the partners Plan4Better GmbH, TUM, Leibniz-Institut für ökologische Raumentwicklung, PSU - Prof. Schalker UmweltConsult GmbH and Münchner Verkehrs- und Tarifverbund.

Technically the tool is a classical server-client application (see Figure 3.1), which uses a spatially-enabled PostgreSQL/PostGIS database as the backbone. As API, the framework FastAPI is written in Python. The application makes use of different data geospatial data. The source code of the GOAT is managed open source using the GNU General Public License v3.0 [8]. The current technical architecture will be extended as sketched in Section 3.3 and 4.

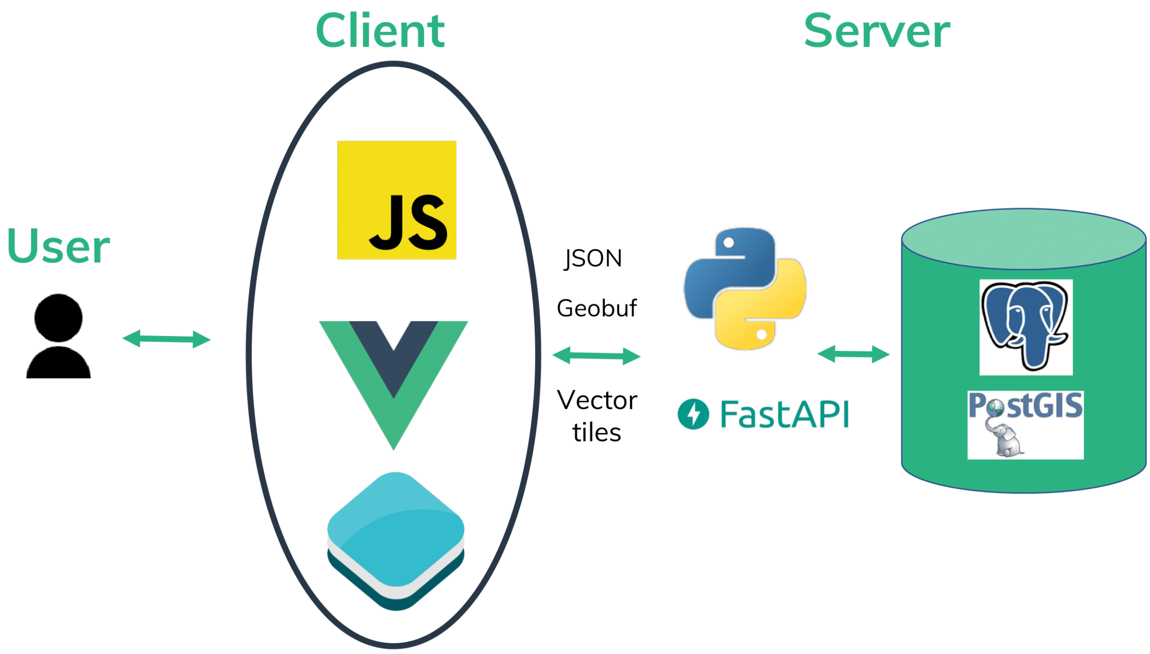


Figure 3.1.: GOAT system architecture
[15]

3.2. Data used

Data is considered an essential aspect of public transport analysis and a crucial factor in the outcome of the results. The data processing pipeline consists of three consecutive steps:

Identification and prioritization
The data types required for calculating indicators are categorized into *spatial* and *non-spatial* data.
From the results of the literature review on the indicators and their specification on the methodology of calculation, the data requirements are organized in a table and complemented with metadata for the provider, reference year, and other details, as shown in Fig. 3.1. The first step is the identification of the data-sets that are required in all indicators, such as GTFS and amenities (Points of Interest), which are named core data-sets. The second step is prioritization based on the assumption that not

all data is equally relevant for all indicators. For each data-set, a weight is assigned, representing the order for the collection phase.

Table 3.1.: Core data-sets used

Use	**Official Name**	**Data Provider**	**Reference Year**
Points of Interest, Network	OpenStreetMap raw files [16]	OpenStreetMap contributors	2022
Points of Interest	Fused POIs P4B [15]	Plan4Better GmbH	2021
Areas of Interest	ATKIS Basis-DLM [17]	Landesamt für Digitalisierung, Breitband (German UND?) Vermessung Bayern	2020
Population	Disaggregated population P4B [15]	Plan4Better GmbH	2021
Public transport timetables, stations	GTFS (NeTEx) [18]	Patrick Brosi	2022

Collection

Data collection is the process of gathering data and analyzing data from various sources. Different data collection methods exist, and choosing the correct method varies considering type (spatial or non-spatial) and other factors. For the amenity data-set, the collection is handled through a combination of data sources by using OSM and data extraction from third-party sources like websites or documents using web scrapers. The population is disaggregated from 3D building models in CityGML. The GTFS (Public transport timetables, stations) is provided under open access by a data provider which transforms the data from the German HAFAS system. As the data collection method varies it is essential to carefully assess the five traits of data quality (*accuracy, completeness, reliability, relevance, and timeliness*) for each data-set.

Cleaning

Data cleaning is the processing of removing or fixing incorrect or not relevant records from a data-set and is the final step in the data processing pipeline. To clean the data, the following techniques are used:

- Remove duplicates
  When data is merged from different sources (e.g., amenities), often, entries are duplicated. For cleaning the data, the validation is conducted using tools for detecting duplicates, such as PostGIS topology functions for geospatial data and other custom scripts.

- Remove irrelevant data
  The collected data often has irrelevant information, which can slow down or give different results during analysis. The dispensable entries are defined and filtered out during the database import step to remove the irrelevant data from the analysis. For example, in the GTFS data, selected agencies that provide on-demand services are excluded for a more accurate model.

- Convert data types
  The data types for numerical and text values need to be standardized and converted during the cleaning process. The numbers appearing as text in the data-set are converted to numerals, date types are stored in the correct format, and geometries for geospatial data are correctly defined with the respective reference system and geometry type.

- Fix errors
  Spelling checks, typos and other inconsistencies in the data are also removed.

- Missing values
  The missing values are either removed or completed if the information is available from other sources.

3.3. Software development

The process of software development is a structure imposed on the development of a software product. It is a set of activities and steps to find predictable, repeatable processes that improve productivity and quality [19]. The software development life-cycle includes five fundamental activities, as shown in Fig.3.2

1. Requirement Analysis
   It is the most critical stage in the SDLC as it defines the technical requirements for the project implementation. The planning stage is performed by domain

experts in the industry and other actors, such as customer requirements, sales departments, and different surveys. The result of this phase is the Software Requirement Specification (SRS) or SRS document. This step includes the expert involvement explained in workshop details in 3.4

2. Design
   Based on the SRS, the design step involves software architects proposing and documenting the product architecture in Design Document Specification (DDS). Usually, there is more than one design variant, and based on different factors such as budget, risk, design, and time, only one of the designs is selected as the final for the product.

3. Implementation
   At this stage, the product is developed and brings the idea or concept to the market. Developers follow the guidelines and use different programming languages and frameworks depending on the software.

4. Testing
   The testing phase involves users that perform product testing, which is the procedure of measuring application properties and finding issues and performance problems or new feature suggestions.

5. User Feedback and Evaluation
   It is essential to report back the outcome of the testing phase and consider the feedback in the next release cycle.

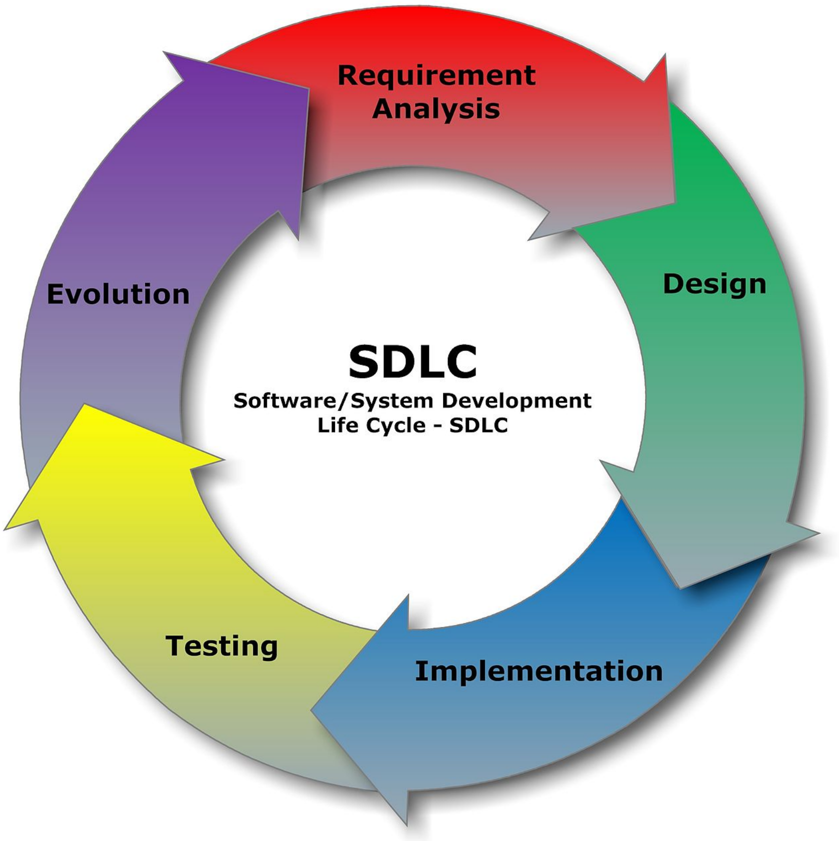


Figure 3.2.: SDLC
Source: User:Cliffydcw (Wikimedia Commons)

3.3.1. Environment

A Software Development Environment (SDE) consists of hardware and software tools developers use to create software systems. The IDE (Integrated Development Environment) is software that provides developers with a code editor, compiler, and debugger under a single interface. There are four environments considered in the SDLC implementation workflow:

1. Development Environment
   It provides a workspace for developers to program tasks related to software development. The workspace comprises the IDE and the distributed version control system (GitHub). This source code hosting service provides a platform

for task management, bug tracking and other features needed for collaboration.

2. Testing Environment
   The testing environment helps to identify and fix bugs that can affect the user experience. In a test environment, the software system is deployed in a test server through CI/CD pipeline where users can access and report the issues in GitHub. In addition, automated testing is also executed in this step.

3. Staging Environment
   It is nearly identical to the production environment and is used to test all the functionality under actual conditions before the final deployment.

4. Production Environment
   It is the environment where the end user can see, experience, and interact with the new features. The application is deployed in production using docker container and Kubernetes a container orchestration system for automating software deployment, scaling, and management in production.

IDE

It offers various features such as code auto-completion, syntax highlighting, and typo detection. The IDE used for this project is VSCode which provides all the features mentioned above, and it has full support for the languages JavaScript and Python. In addition, VSCode allows using docker containers as a development environment. It was used to run the API application, tools, libraries and run-times needed with the code-base. It also aids in continuous integration and deployment pipeline (CI/CD).

Prototype

It is a critical step in the SDLC workflow. For communicating ideas, the prototype was designed in an iterative process with the experts and existing users of GOAT. The tool used for the prototype design of the User Interface (UI) was Figma which offers a collaborative web application with features that help to brainstorm, design, and accelerate the iteration phase.

3.3.2. Technologies

Data layer

The initial starting point of the implementation is the Data layer. It contains a relational database running on a PostgreSQL instance using AWS (Amazon Web Services) RDS cloud provider with a PostGIS extension. AWS RDS is a distributed relational database service that is simple to set up, operate, and scalable based on the demand of application resources. It also includes a MongoDB instance for the external system R5. This layer also contains various third-party data services for seeding and updating the data, which are not shown in the diagram as they are out of scope for this thesis. The main software stack used in the data layer is

PostgreSQL

It is a data management system and a relational database that allows for the expansion and application of open standards. The primary function is safely saving data and returning it as responses to SQL requests made by other programs. In the case of this application, it is used for storing and managing all geographical and non-geographical information

(Version: 14, License: PostgreSQL)

PostGIS

It is an extension installed on top of the PostgreSQL database and adds support for geographic data by making it possible to read geometries (points, lines, polygons, raster, etc.), adding many GIS functionalities, and allowing their usage through SQL language

(Version: 3.1, License: Global Public License (GPL))

MongoDB
The NoSQL database program uses JSON-like objects to store the data. In the case of this application, it is used for storing and managing information for R5, such as regions (or study area) information, GTFS, OSM network bundle status, public transport scenarios and other metadata.
(Version: 4, License: SSPL)

Logic layer

The Logic layer or application layer contains the application core logic. It consists of a Web-Api Service written in Python using the FastAPI web framework and various libraries for analyzing public transport networks and geospatial data. The logic for the indicators analysis and application business logic is written in this layer and exposed as endpoints in which third-party applications can communicate and interact using the REST interface. These services are implemented as part of existing GOAT endpoints and are used primarily through a Single-Page Application (SPA) client (GOAT interface). In addition, an instance of the Conveyal R5 engine is set up to handle requests for public transport routing analysis which is exposed through the same API. The main software stack used in the logic layer is

FASTAPI

It is a Web Framework for building high-performance RESTful API with Python 3.6 +. It uses various libraries for validating, serializing and de-serializing data. The library has full support for asynchronous programming and can automatically generate endpoint documentation using OpenAPI specification.
(Version: 0.74, License: MIT)

SQLAlchemy

It is an Object-Relational Mapper (ORM) for Python. It provides an abstraction interface for interacting with different database types using Python object collections.
(Version: 1.4.23, License: MIT)

Conveyal R5
R5 (Rapid Realistic Routing on Real-world and Reimagined networks) is a routing engine focused on trip analysis for public transport. It supports time windows by considering different travel paths of the user as opposed to single-trip passengers facing journey planning. It allows scenario creation and evaluation of cumulative opportunities accessibility indicators. For this project, R5 is the main engine for calculating one-to-many and many-to-many public transport travel-time calculations, which are further enhanced with complementary data from the GOAT database.
(Version: 6.4, License: MIT)

Presentation layer
The presentation layer is a communication layer where users can interact with the application layer through a user interface. This layer is a dynamic environment in which users can visualize the geospatial data of the indicators and interact with the endpoints through UI (user interface) components. These components are implemented in a Single-Page Application (SPA) Client and are automatically enabled for all study areas where GTFS data is available. This is the system's entry point for the user, which provides an interactive web map interface. The application is developed using the Javascript programming language, framework Vue.js and other third-party utility libraries. The SPA communicates with API using standard HTTPS operations (Create, Read, Update, Delete), and it can accept JSON (text) or PBF (binary) results. The software stack used in this layer is

Openlayers
It is a Javascript library for displaying maps on the Web, as well as manipulating vector data by using the WebGL technology of browsers. It gives the possibility of using APIs for building WebGIS dynamic like "Google Maps" or "Bing Maps."
(Version: 6.5, License: BSD 2-Clause)

Vue
It is a Javascript library (framework) that facilitates the development of graphical interfaces of Web applications using an HTML syntax that allows binding DOM and instance data.
(Version: 2.6.6, License: MIT)

Vuetify
It is a material design component framework for Vue with many components for building user interfaces. These components are directly used in the application without the need for programming.
(Version: 2.0.18, License: MIT)

3.4. Expert involvement

This master thesis is developed in collaboration with the software startup Plan4Better, which develops the accessibility instrument GOAT. The development though is carried out in close collaboration with partners such as the Technical University of Munich (TUM). Therefore, this project had the chance to engage with experts from both institutions. Within the scope of the applied research project GOAT 3.0, the author had the chance to engage in project meetings and present preliminary results to individuals from practice and research. That first results could be discussed, and feedback is obtained, that was considered in the development.

A focus with TUM in the exchange was on the selection and the scientific soundness of the accessibility measures. Therefore, researchers from TUM provided a list of accessibility measures that was used as the basis of the list presented in Table 4.1. The exchange the researchers from TUM usually happened as part of the supervision meetings.

The exchange with software developers and transport planners from Plan4Better was shaped by the technical exchange and the experience the team has when communicating with customers from municipalities and public transport authorities. As the author is himself part of the team the exchange happened continuously and using different means. Accordingly, things were discussed face-to-face, in video calls, or case of software development-related aspects using the platform GitHub.

To obtain expert feedback outside the described environment, a workshop with practice partners was organized. The workshop was contained in a remote setting using Microsoft Teams and was recorded. It took approximately 1.5 hours and occurred on the 12th of June, 2022. In total, six external experts were involved. The experts were transport and regional planners from Verkehrsverbund Rhein Neckar (VRN), Münchner Verkehrsgesellschaft (MVG), the City of Augsburg, and Metropolregion Rhein-Neckar. The core questions that were asked to the experts were:

How practical are the indicators in terms of data availability, comprehensibility, technical complexity and visualization?

In order to obtain the feedback the workshop was structured like shown in Table 3.2.

3.5. Assessment accessibility measures

Based on an overview of accessibility measures provided by TUM - Chair of Urban Structure and Transport planning, there were selected five indicators falling in the category public transport station accessibility, and four indicators falling in the category of accessibility by public transport. This pre-selection was made based on the discussion with experts from TUM and Plan4Better. In the following the indicators were assessed based on the following criteria:

Institutionalization
It is examined whether the indicator is already used in formal planning processes by stakeholders like public transport authorities or municipalities. A higher ranking of institutionalization could indicate that there is high practical relevance. There was given a higher ranking if the indicator is already institutionalized in Germany, then in another country.

Data Availability
Each accessibility measure is assessed whether it is hard to obtain the needed data. Factors included in the assessment are the number of needed data-sets and the ease of finding the data in the German context. Data that is generally published as open data is ranked as highly available. However, proprietary data or data such as POIs ,which are often only available when combining different data sources, are rated as hardly available.

Technical complexity
Within the category technical complexity, it is studied whether the indicator is hard to implement and compute within a WebGIS environment. As reference, the tool GOAT is taken. Accordingly, it is studied whether complex algorithms or long calculation times are needed to implement the indicator.

Indicator Complexity
As indicator complexity, it is studied whether the indicator involves different aspects of accessibility, such as the different accessibility components. Therefore it is also a degree whether the indicator aim to picture accessibility and mobility behavior as realistic as possible. Typically complex accessibility measures are composed of

several sub-indicators. As a consequence, complex indicators are also usually harder to understand by individuals outside of the accessibility modeling community.

The classification was done based on existing literature and if available open source code repositories. Furthermore, the exchange with the planning experts was incorporated into the assessment. Despite following including different criteria, the assessment is firmly based on the authors' perception. The lack of a scientifically more sound assessment framework is clearly a limitation of the classification.

Agenda item	Purpose
Welcome and round of introductions (2:00-2:05 PM)	Get familiar with each other and understand the motivation and responsibilities of the expert within his organization
Overview of indicators in local transport planning (2:05-2:15 PM)	Show the literature review summary on existing accessibility measures to prepare the discussion in the next step.
Presentation of selected indicators (2:15-2:25 PM)	Show more in detail the indicators that were identified as the most promising ones in the literature review and internal discussions.
Interactive presentation of GOAT (2:25-2:35 PM)	Present the functionality of GOAT to facilitate participants understanding the context of the development and the foreseen integration into the platform.
Discussion of the strengths, weaknesses and practicality of selected indicators using Miro (2:35-3:15 PM)	Critically discuss the potential of each indicator using the online collaboration platform Miro. The participants were asked to leave sticky notes with comments and rank more relevant indicators using dot voting. Each participant did this on an individual basis and afterwards each of them explained the comments and justified the rating.
Open exchange and discussion (3:15-3:30 PM)	Finally, the results were summarized and the participants had the chance to express additional feedback and ideas.

Table 3.2.: Agenda expert workshop

4. Results and Discussion

4.1. Accessibility measure assessment and selection

There is a wide range of accessibility measures being used for the accessibility analysis of public transport. Furthermore, they can highly vary in their purpose. While some are officially used by public transport authorities, others are mostly used in academia so far. As suggested within this thesis project the indicators are divided into two groups:

1. **Public transport station accessibility**
   Indicators that measure accessibility to public transport stations. They can include the assessment of timetable departures at the stop level to incorporate the station attractiveness.

2. **Accessibility by public transport**.
   Accessibility measure that model travel-times using the public transport network, which consists of stations, lines and schedules. They involve routing analyses such as as one-to-many or many-to-many.

In table 4.1 an overview of the assessed accessibility measures is provided.

Name	Institution-alization	Data Availability	Technical complexity	Indicator Complexity	References
Public transport station accessibility					
Station departure count	●●●	●●	●●	●	[20]
Catchment areas (Buffer)	●●●	●●●	●	●	[21]
Walking and cycling catchment area (Isochrone)	●●	●●	●●	●●	[7]
ÖV-Güteklassen	●●	●●	●●	●●	[22, 23]
PTAL	●●	●●	●●●	●●●	[24, 25]
Accessibility by public transport					
Intermodal catchment area (isochrone)	●●	●●	●●	●●	[26]
Travel Time to selected destination	●●	●●	●●	●	[27]
Cumulative opportunities heatmap	●●	●●	●●	●●	[5]
Gravity-based accessibility heatmap	●	●●	●●●	●●●	[5]

●●● High ●● Medium ● Low (–) No Rating

Table 4.1.: Public Transport Accessibility Indicators (Overview)

It has to be underlined that there is not a perfect accessibility measure neither to assess public transport station accessibility nor accessibility by public transport.

Though, in the context of this study the perfect indicator would have been one that is relatively easy to implement, while being institutionalized and of moderate indicator complexity. In the following it was decided to chose the station departure count and ÖV-Güteklasse as accessibility measure to be integrated into GOAT within the field of public transport station accessibility. Walking and cycling catchment areas were seen also as very relevant, though they are already implemented in GOAT. In particular, in the case of the ÖV-Güteklassen the high value of this indicator was confirmed by high dot voting during the workshop (see Appendix A.1). The official use of ÖV-Güteklassen in the spatial planning of Switzerland was another argument when favoring the accessibility measure.

For the accessibility by public transport indicators the intermodal catchment area (isochrone) was seen as indicator to be integrated first as it is relatively easy to understand and was perceived to bring added value to the experts from practice (see Figure A.2). Meanwhile, the gravity-based accessibility heatmap can be considered as the most complex indicator with comparatively little institutionalization. Therefore, it is suggested to start with the implementation of the isochrone also because it might be basis for the implementation of the other indicators.

4.2. Developed software architecture

The overview of the software architecture is modeled in the following diagrams using the C4 model, an "abstraction-first" approach to diagramming software architecture. The first diagram 4.1 shows the top-level context, which represents the overview of internal and external systems and how users interact. In this case, users interact with the GOAT system through HTTPS protocol to conduct accessibility analysis and visualize geospatial data. GOAT system uses HTTP protocol to communicate with the R5 external system to use its internal services.

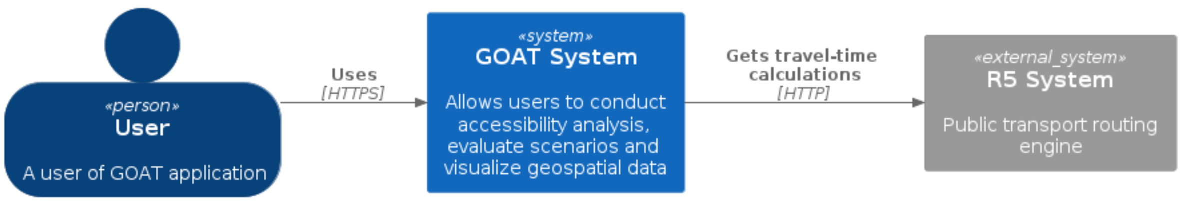


Figure 4.1.: Top Level Context Diagram (C4 Model - Level 1)

A detailed overview of each system is given in the container diagram 4.2. This diagram is the starting point in defining the software system architecture. It shows the internal services, APIs, applications, databases, and how they interact. Each application or service is visualized with a container box, and their interactions are shown at a high level. The diagram 4.2 shows the internal services and databases for GOAT and R5 systems, the technologies used and the communication protocols.

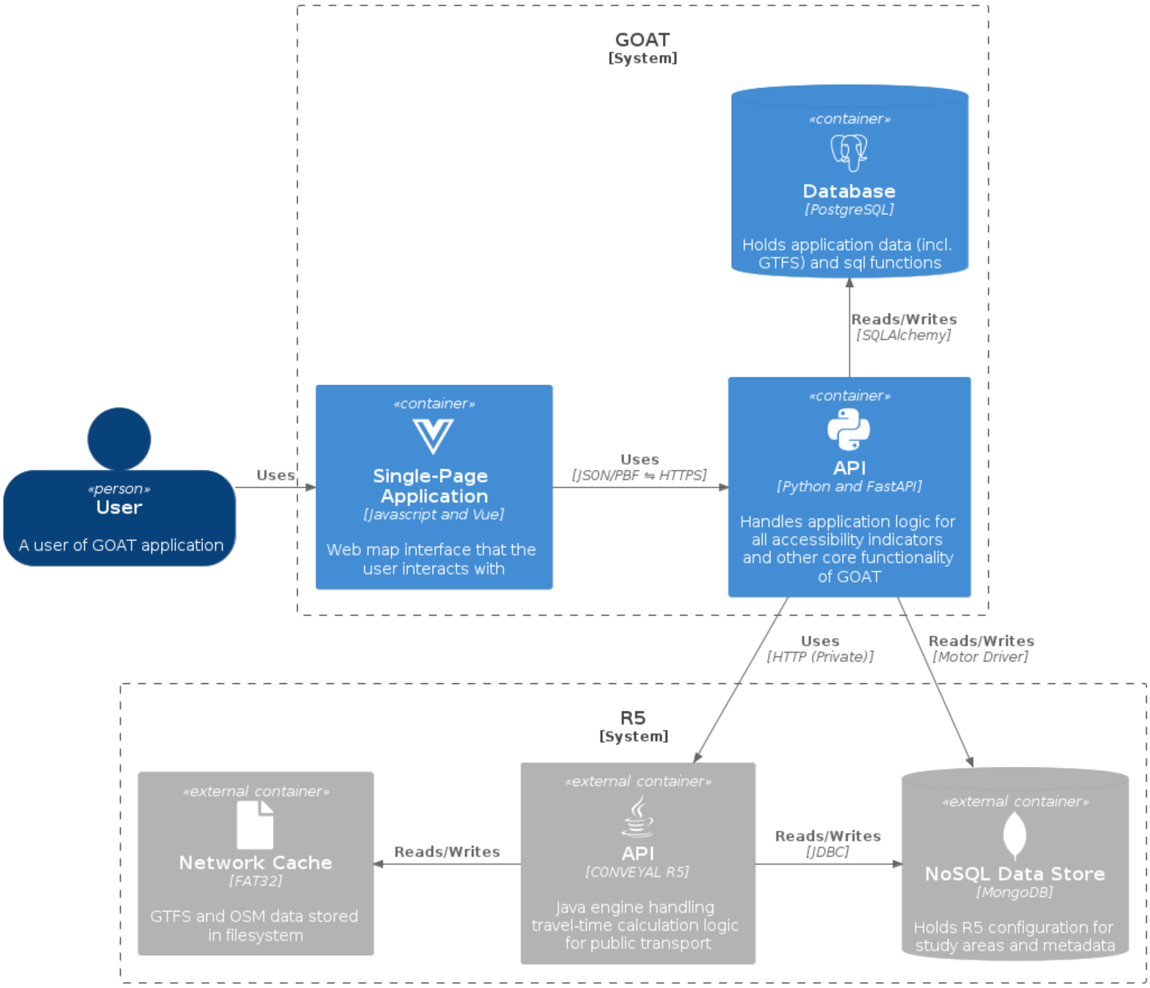


Figure 4.2.: Container Diagram (C4 Model - Level 2)

The container diagram 4.2 is based on a three-tier architecture style that consists of three layers (data layer, logic layer and presentation layer). The advantage of this architecture is that each layer is an isolated environment and can be developed independently of other layers, can be scaled using micro-service architecture, and reduces maintenance complexity in large projects. The implementation is adapted to the existing GOAT architecture, which runs in a cloud environment.

As mentioned above, the indicator calculation logic is implemented inside the API container. Therefore a detailed diagram is shown in 4.3 which describes the main internal elements (components) in more detail. Each indicator in the API container is defined as a separate endpoint and operates using the CRUD component. OpenAPI documentation for the indicators is automatically generated and visualized in a dedicated user interface using Swagger UI, allowing testing outside of the SPA as shown in figure 4.4.

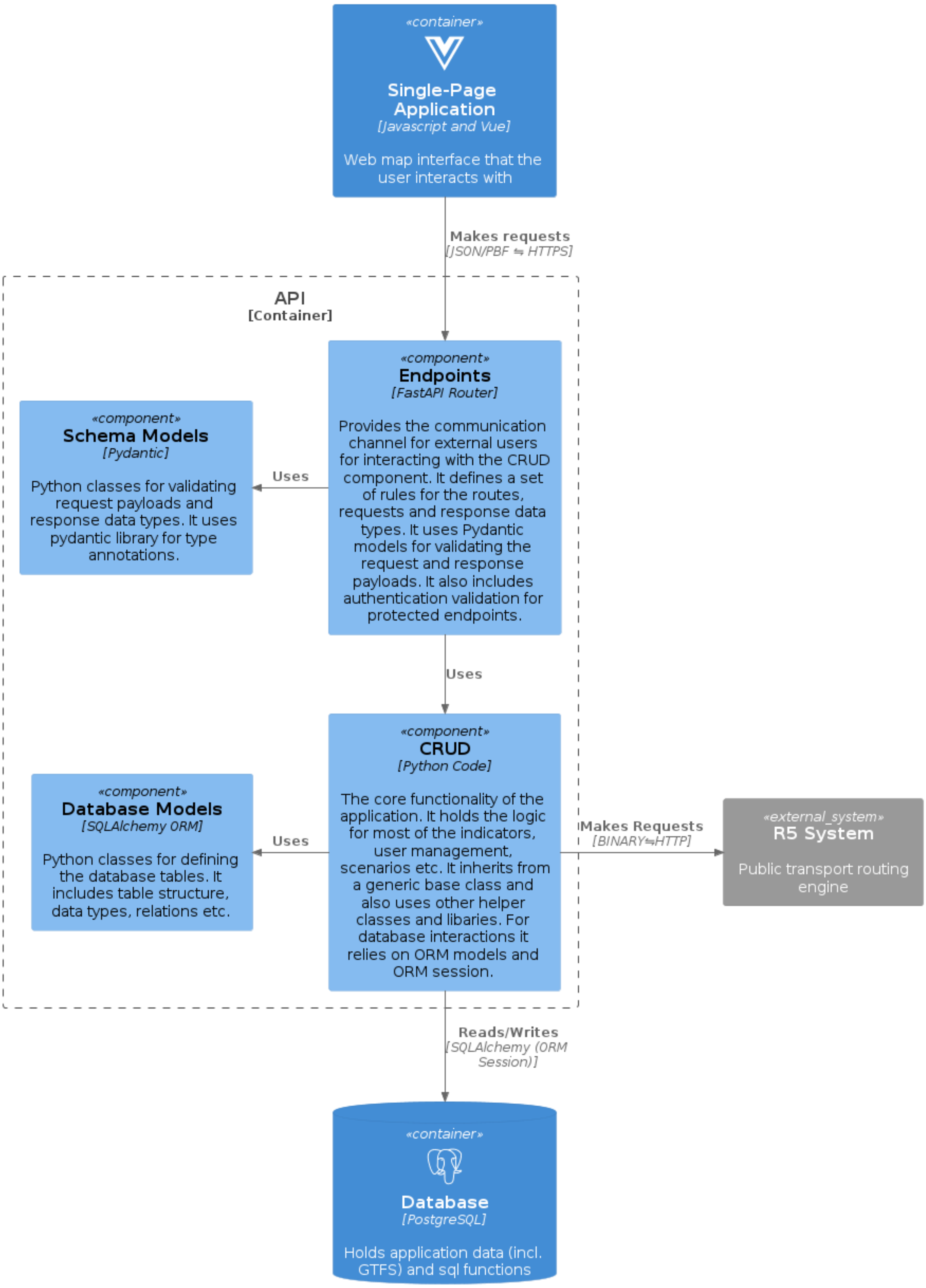


Figure 4.3.: Component Diagram for API (C4 Model - Level 3)

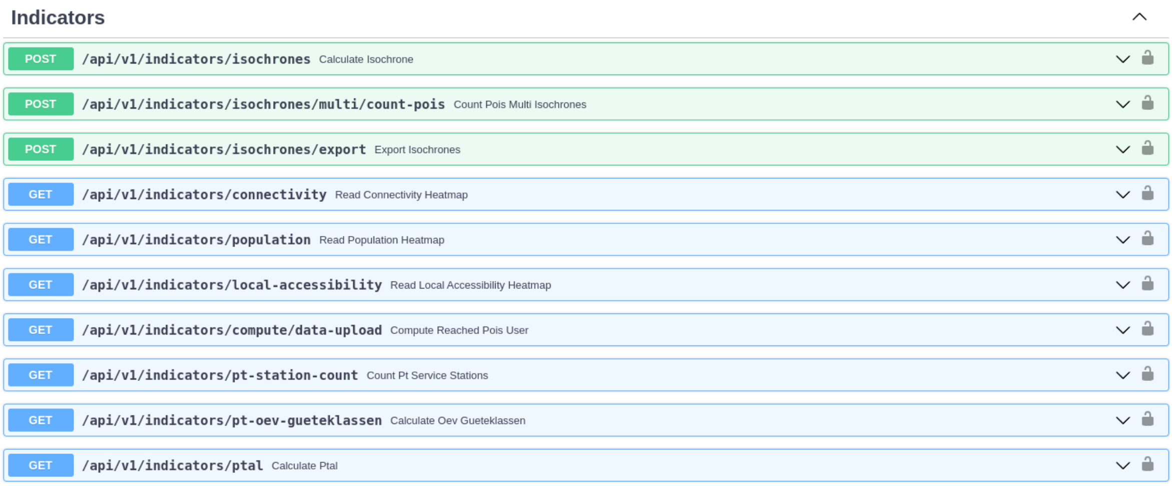


Figure 4.4.: Indicators Endpoints (SWAGGER UI)

4.3. Development accessibility measures

4.3.1. Public Transport isochrone

Isochrones or travel-time map is an indicator that provides the user with an easily understandable visualization of places accessible from a selected location within a specified time. The area's borders falling within the time threshold are commonly known as isochrones. The isochrones can be computed for modes such as walking, cycling, car and public transport from one or multiple starting points. Each mode includes input factors such as speed, infrastructure network and traveling time. In this thesis, the focus will be on the implementation of public transport isochrone. This indicator is prevalent in the planning practice, and can be used to answer various planning questions such as:

- How many points of interest can be reached within a time window using public transport services?
- How many people can the public transport services reach on a specific date and time within a travel time limit. ?
- Which areas can be accessed for a selected weekday (e.g., Monday) and time window (e.g., 7:00 AM - 9:00 AM) using a combination of different public transport services (rail, bus, tram.)

Data

Compared to other isochrone modes, public transport isochrone has an increased complexity. It includes a temporal dimension and requires a combination of different data types. The data-set used for calculating public transport isochrone is the routable transport network which includes scheduled public transport services, the region's streets, sidewalks and bikeways. This dataset is a bundle from a GTFS and OSM processing procedure created through the R5 engine. The data-set is used in the R5 external system, loaded into memory for high-performance reading, and stored as a custom binary file format for later reuse. The isochrone calculation result is intersected with complementary data for the points of interest and population, which are fetched from the GOAT database. An overview of the data pipeline is shown in the diagram 4.5.

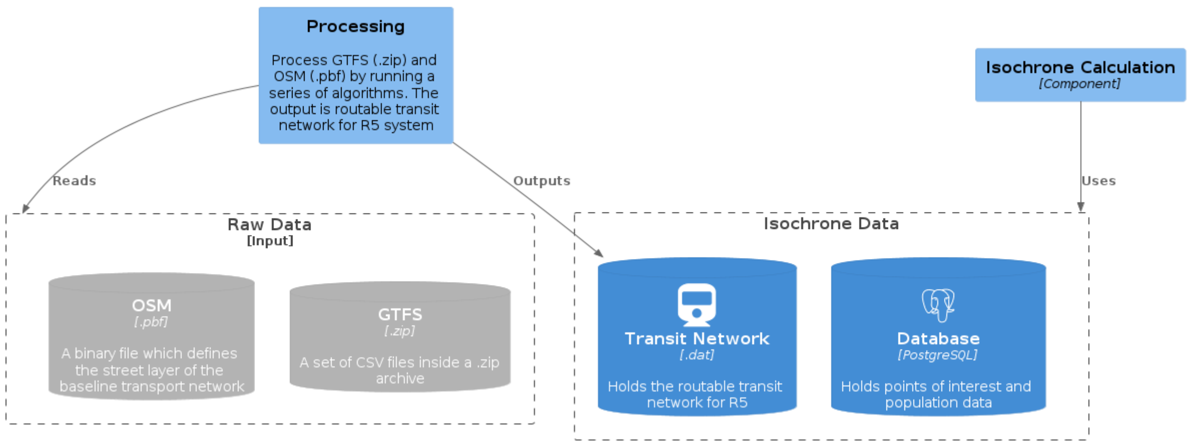
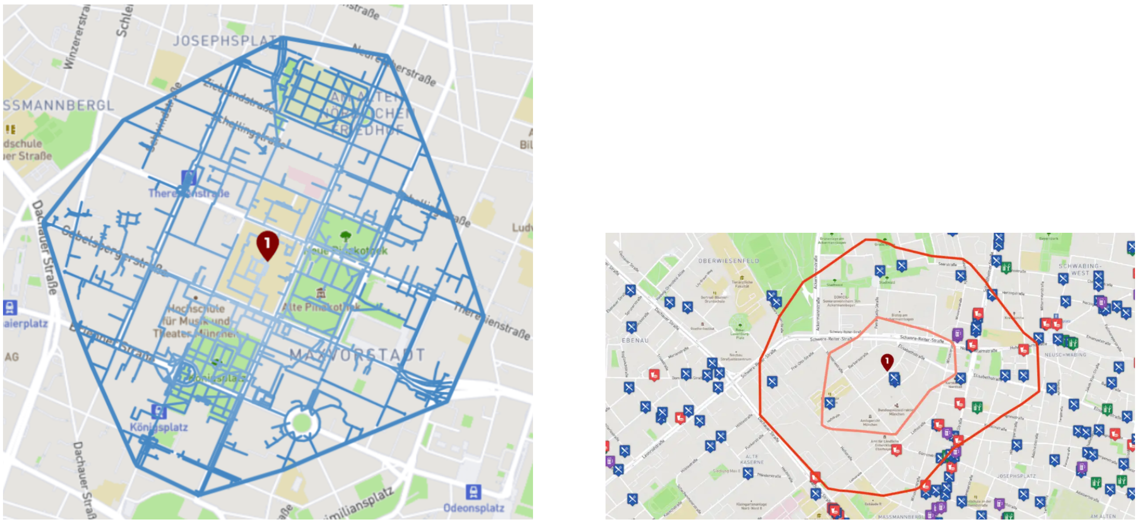


Figure 4.5.: Isochrone Data

Results

Isochrone results are usually visualized using arbitrary polygons created using alpha shapes using a concave hull algorithm by fitting boundary polygons around a point data-set created from the road network. However, this approach does not always give granular information for the reached area that is not very close to the boundary. In a 10-minute isochrone calculated using this technique, it is assumed that all the points inside the boundary are reached within this time limit. Although it does not give any information on how far a location is from the starting point. To overcome this limitation, isochrones are often calculated using multiple steps, e.g., every 2 minutes until the 10-minute limit is reached Fig. 4.6.



(a) Single Step (b) Multiple Step

Figure 4.6.: Isochrone Visualization - (Concave Hull)
Source: plan4better.de

In small areas and when the travel time limit is not long, using multiple steps is sufficient. This usually is the case for travel modes such as walking and cycling. For public transport isochrone, the minimum travel-time taken for accessibility planning is much larger and varies between thirty minutes to two hours which results in significantly larger covered areas. Therefore multiple isochrone steps in this scenario will give results in which visualization and the user experience are not very pleasant to the end user. To overcome this problem, Conveyal introduced a new method of isochrone visualization for public transport by representing the origins and destinations on a regular grid. This is known as the Modifiable Areal Unit Problem (MAUP), a statistic effect in which samples for a given area are aggregated into other spatial units. In this approach, the result of the isochrone calculation, which contains the node costs of the transit network, is aggregated and represented into a grid in which one cell is one unit step in Web Mercator projection. Therefore the resolution of the results depends on the Web Mercator zoom level. High zoom levels will return high-resolution results but also come with the penalty of computation time. The cell width also varies depending on latitude, longitude and the selected zoom level. Table. 4.2.

Zoom	Cell Width (Equator)	Cell Height (45 degree)
9	306 m	216 m
10	153 m	108 m
11	76 m	54 m

Table 4.2.: Web Mercator Zoom levels used for public transport isochrone

The R5 routing returns a binary encoded data structure that contains the aggregated travel times for each grid cell that intersects with the routed network. In addition, the result is complemented with additional data for the grid starting point (northwest), width and height. A simplified version of this grid is shown in Fig. 4.7.

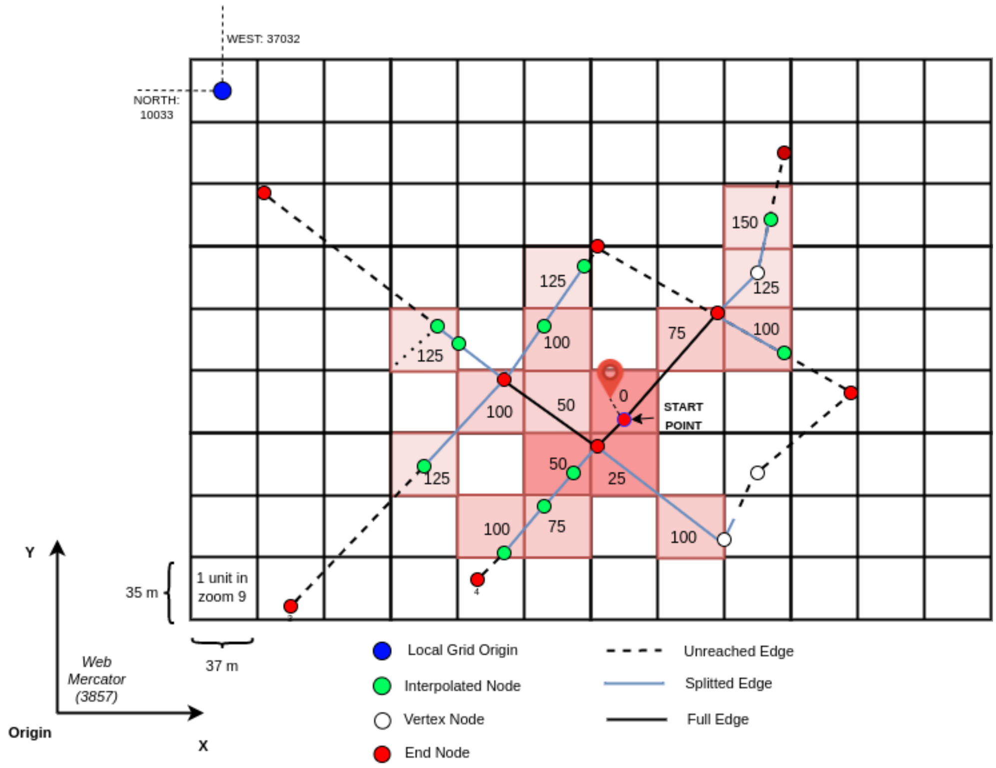
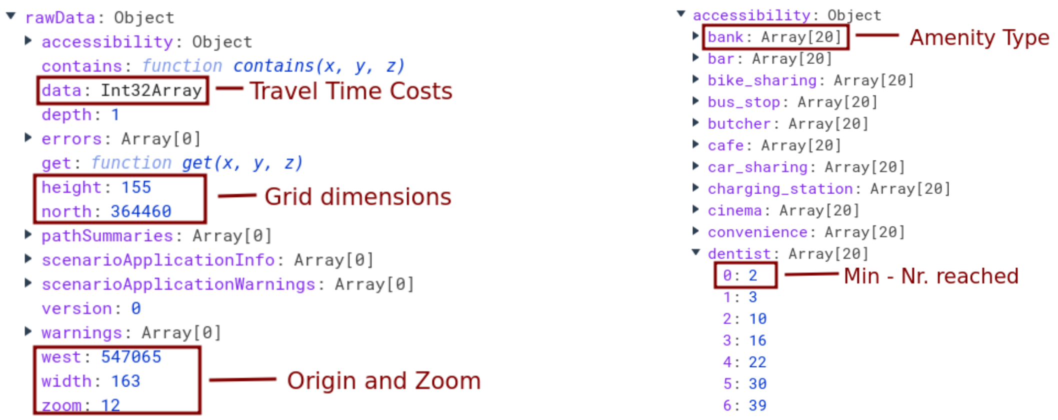


Figure 4.7.: Grid Interpolation with Travel Time Costs

The data structure of the grid returned from the R5 engine is shown in Fig. 4.8. It contains the travel-time costs in a 2-dimensional (2D) array, and also the necessary information for reconstructing the grid without the need to store geometries in the response.



(a) Data (b) Amenities Reached

Figure 4.8.: Grid Structure

However, for the dynamic isochrone, the data in the initial response from R5 do not give any information on the reached amenities, such as point of interest and population. To complement the returned data, the grid is intersected with the amenity data from the GOAT database in a second step, Fig. 4.5. As a result, the grid will contain accessibility information for all the amenities for each minute of the travel-time window. The endpoint for the isochrone is shown in the Swagger UI in Fig.4.9

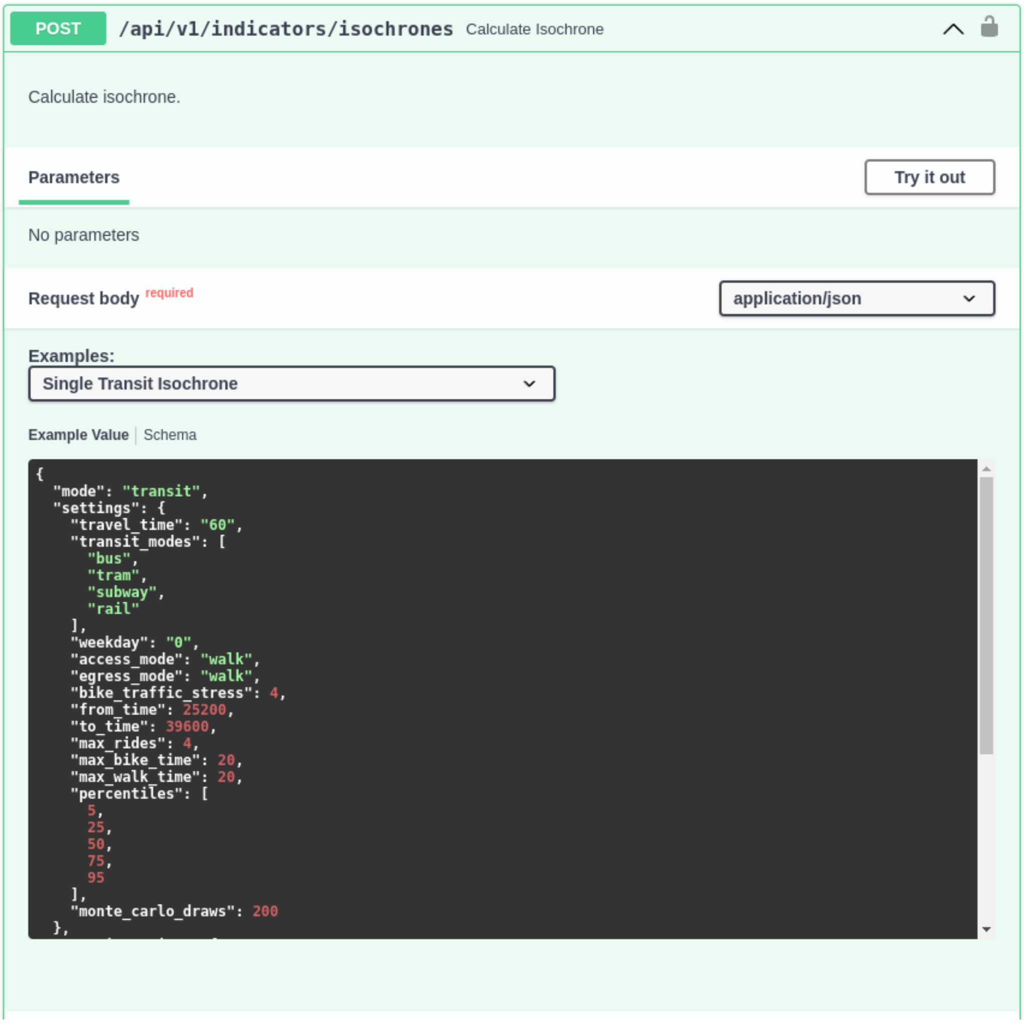


Figure 4.9.: Isochrone API - (Swagger)

Visualization

For visualizing the data structure given in 4.8, it should transform to a standard geospatial data structure such as GeoJSON. This is achieved using *"Marching squares"*, a computer graphics algorithm that can generate two-dimensional contour lines from a rectangular array of values [28]. An illustration for 2D image processing is shown in Fig. 4.10. The implemented logic is similar even though it is not applied in raster pixel. In the scope of this thesis, this algorithm is implemented in JavaScript (inspired by Conveyal UI) for visualization with Openlayers. In addition, a new implementation is written in Python for the API container, which is used for spatial operations with the database when finding the reached amenities.

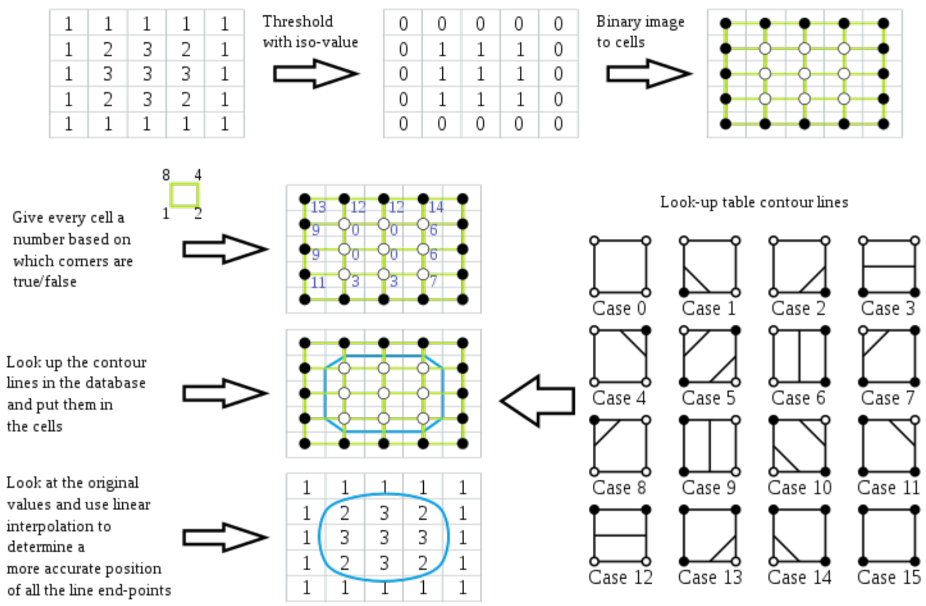


Figure 4.10.: Marching Square Algorithm
Source: User:Nicoguaro (Wikimedia Commons)

User Interface

Components for public transport isochrone are implemented as part of the GOAT application. As shown in Fig.4.11 it has three sections on the left side panel, which include configuration (1), calculation button (2), results (4), and map interactivity (3). The configuration section contains dropdowns for the routing type (*Transit*), the weekday (*Monday - Sunday*), From Time - To Time (*analysis time window*), Access Mode (*How user access the station, e.g., Walk, Bicycle or Car*), Egress Mode (*How user changes* *the station*) and Transit Modes (*e.g. Tram, Rail, Bus or Rail*). The calculation button activates the map interaction expects the user to click on the map, and select the starting point. The map click event triggers the isochrone calculation event, and sends the request to the API with the payload containing starting point coordinate and options configuration. The calculation is added in the results section (4), automatically activated, and shown on the map.

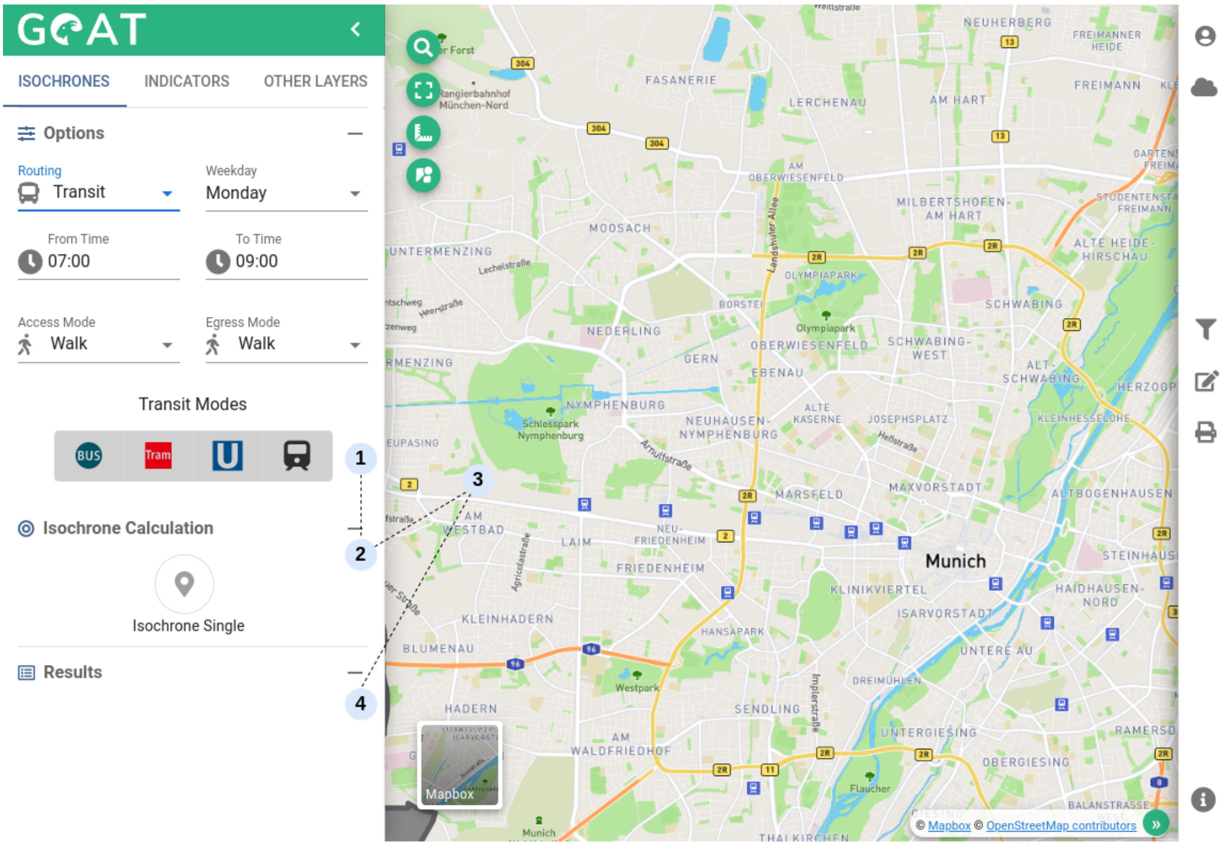


Figure 4.11.: User Interface (Flow)

The result user interface contains three main components Fig. 4.12 . The result card (C), isochrone window (B), and amenity selection tree (D). The result card has interactivity for the visibility of the calculation. It displays information on the options used for the request as well as the address of the starting point. Users can select a maximum of two calculations at the same time, which will activate the comparison mode. The calculation data of the selected isochrones is then displayed on the isochrone window. The window contains a time slider element that can re-render the isochrone shape based on the travel-time the user selects. The slider change event will trigger the marching square contour line algorithm, and generate a new shape of the isochrone. In addition to the slider control, the population and amenity data is also represented in this component using a table, line chart Fig.4.13, or pie chart. Depending on the scenario user can switch between different visuals. Table data is used when the user wants specific attention at the selected time, while a line chart gives an overview of the amenity values throughout the total travel-time. To know the exact travel-time at a specific location user can hover in the isochrone surface on the map and get the information in a tooltip element following the cursor. In addition to the interactive elements, users can export the isochrone results into three different formats, GeoJSON, Shapefile, or XLXS, to visualize in third-party applications such as QGIS or ArcGIS.

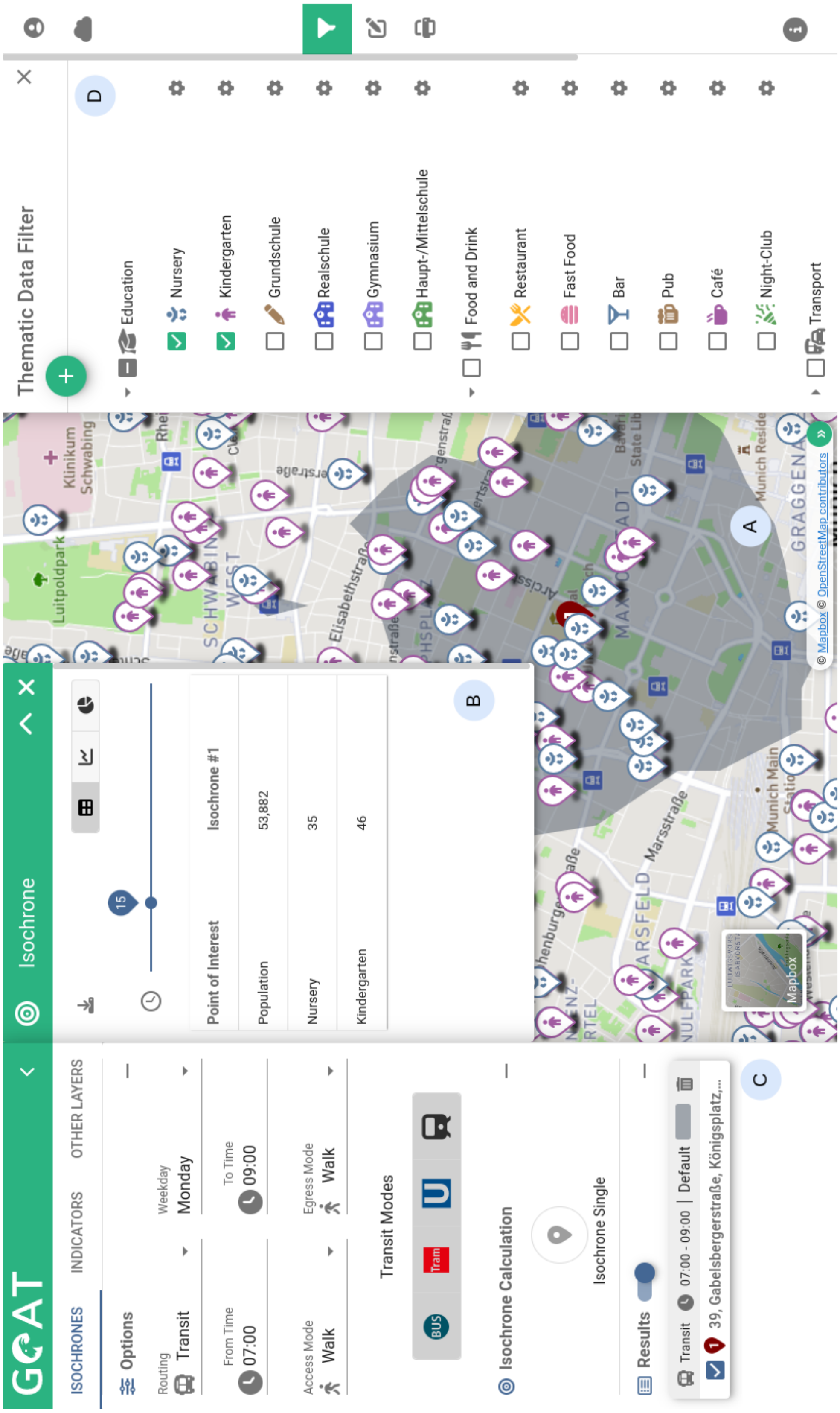


Figure 4.12.: Results - Table

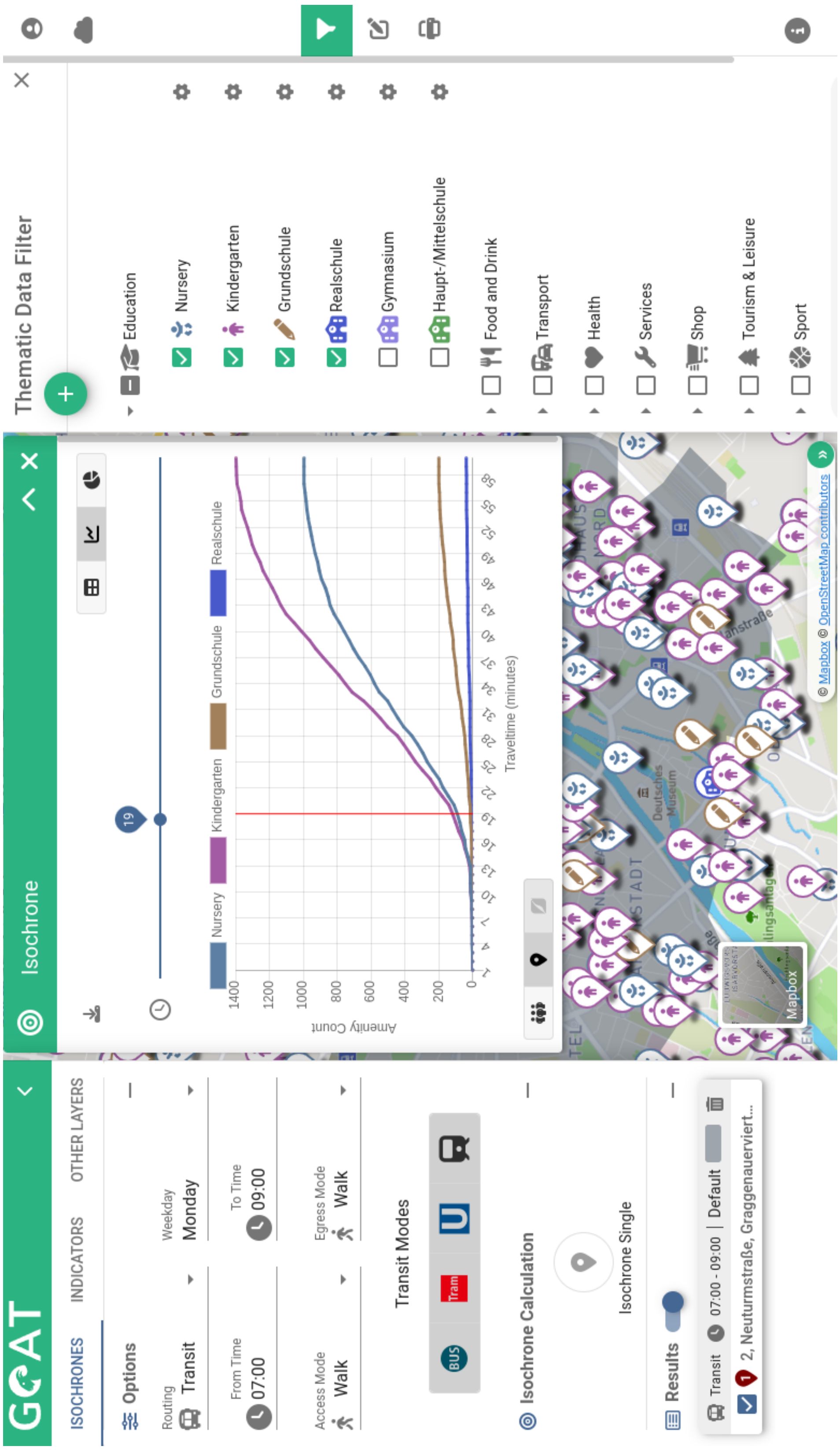


Figure 4.13.: Results - Amenity Line Chart

4.3.2. ÖV Güteklassen

Public transport quality classes (*ÖV Güteklassen*) is a public transport indicator for measuring how well the locations perform as attractive development areas. The specification is created by the Swiss Federal Office for Spatial Development (ARE), with the first version in 1993 and the latest revision in 2011. Since then, it has been considered an essential instrument in formal planning processes in Switzerland. In addition, the Swiss model served as an inspiration for use in Austria (including Voralberg). The institutionalization of the indicator in German-speaking countries and the comprehensible and, at the same time, differentiated calculation methodology are essential advantages of the public transport quality classes. The indicator was adapted to the conditions in Germany and implemented in GOAT. The calculation period has been made more flexible so the indicator can be calculated for any day of the week and time of day. In the Swiss version of the indicator, the quality classes are usually calculated for departures on weekdays between 6 a.m. and 8 p.m. The calculation methodology implemented in GOAT is explained in more detail below.

Data

The calculation is based on the GTFS, which is imported into the PostgreSQL database of GOAT and covers all of Germany. The data reading performance is optimized using additional database indexes on stop times, and services such as on-demand buses are excluded from the calculation to have a more accurate result.

Implementation

The first step is finding the number of departures for each station for the public transport modes (bus, tram, rail, metro). The departure sum is divided by two to avoid counting twice for outward and return directions. In the next step, the frequencies are calculated. Based on the specification in Table 4.3, each station is assigned a number from I - VII to determine the transport category. This table is dynamic, and it changes based on the available services in the area. In Table 4.3, the specification includes only rail for metro and regional trains, tram, and bus services.

Interval	**Transport Category**		
	Rail	**Tram**	**Bus**
< 5 min	I	I	II
5 – 10 min	I	II	III
10 – 20 min	II	III	IV
20 – 40 min	III	IV	V
40 – 60 min	IV	V	VI
60 – 120 min	V	VI	VII

Table 4.3.: Identification of the station category

After identifying the station category from each station, the last step is to calculate buffers for area classification. The class is determined by a combination of the distance to the station that can be covered on foot and the station category. The distance to the station is defined in the specification Table. 4.4 and it ranges from 0 to 1000 meters, divided in four groups. The buffers create several radii that are brought together from each station and then merged, which results in a single geometry (multipolygon) for each class. The classes represent the quality of the area underneath:

- Class A: High-quality accessibility
- Class B: Good quality accessibility
- Class C: Average quality accessibility
- Class D: Not-so-good accessibility
- Class E: Bad accessibility
- Class F: Worst accessibility

Station Category	Distance to station			
	< 300 m	**300 – 500 m**	**501 – 750 m**	**751 – 1000m**
I	Class A	Class A	Class B	Class C
II	Class A	Class B	Class C	Class D
III	Class B	Class C	Class D	Class E
IV	Class C	Class D	Class E	Class F
V	Class D	Class E	Class F	-
VI	Class E	Class F	-	-
VII	Class F	-	-	-

Table 4.4.: Station Category Classification

Like isochrone, the ÖV Güteklassen implementation is done in the Python API and exposed as a FastAPI endpoint and the configuration for the specification is sent as a payload in JSON format, as shown in the Swagger UI in Fig. 4.14. The endpoint gives more flexibility for the users to change the station configuration according to their requirements and provides the possibility to use the indicator on other applications outside GOAT. The *station_config* object is optional, and by default, the API will use the settings described above. Payload also include *start_time*, *end_time* defined in seconds from the midnight and the weekday from 1-7 *(Monday - Sunday)*.

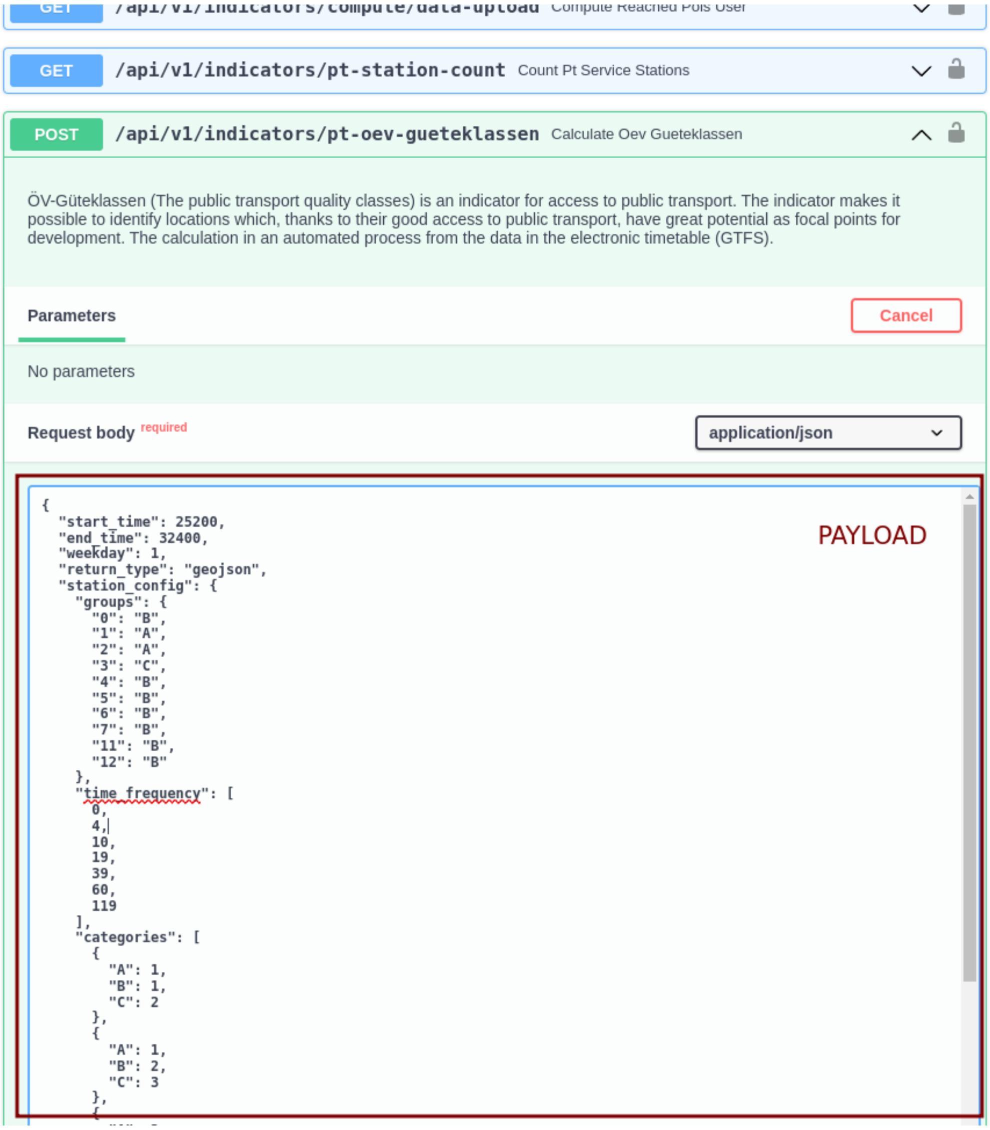


Figure 4.14.: ÖV Güteklassen API - (Swagger)

User Interface

The ÖV Güteklassen indicator is added as a layer in the indicators section in the GOAT UI, as shown in Fig.4.15. In the left side panel, the user can control the layer's visibility using the checkbox and the layer's transparency using the slider. When activated, another component for the time and weekday selection becomes active on the map. The user can use this component to change the time window and the day of the week, which triggers a new request on the server and recompute the indicator. In addition, users can change the map style for each class or click on the map to get more information. The default map style for this layer is based on the style from the *geo.admin.ch (Swiss Geoportal)* and adjusted slightly to meet the map style of other indicators that show similar metrics.

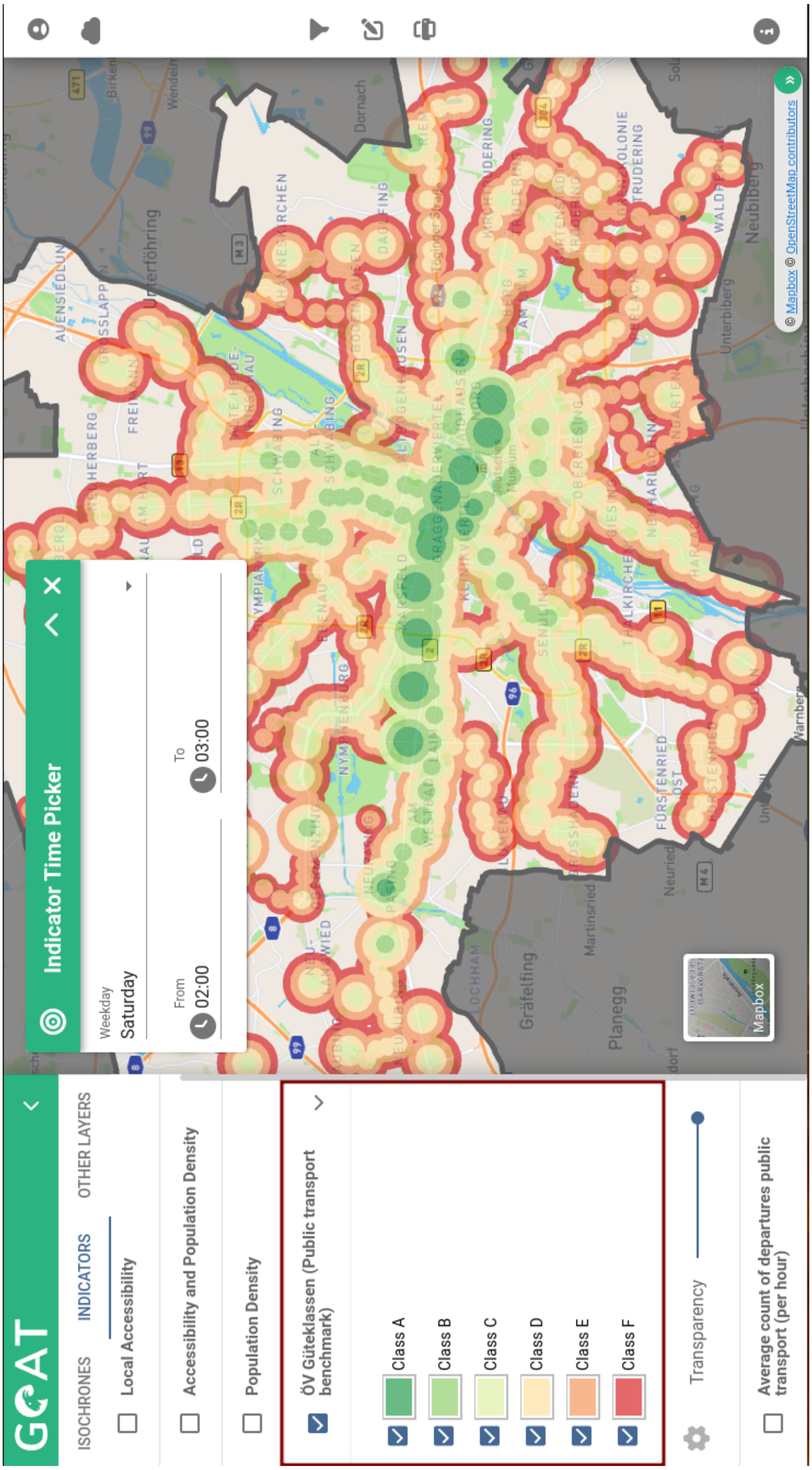


Figure 4.15.: User Interface - ÖV Güteklassen

4.3.3. Station departure count

It is an indicator showing the average count of departures for public transport per hour. This indicator is used as the base for the ÖV Güteklassen but can also be used independently as a simple public transport metric for displaying information on a station level. It provides an overview of how well a station performs at a defined time window and day, which is relevant information for planning authorities.

Data

Same as ÖV Güteklassen, this indicator reads GTFS data from the GOAT database. Tables used for the calculation are *stop_times*, *stop_times_optimized*, *stops*, and *calendar_dates*. Also, on-demand service lines are excluded.

Implementation

The core logic of this indicator is written as a SQL function inside the PostgreSQL database, which is called and executed from the Python API. The endpoint is exposed using FASTAPI as shown in the Swagger UI in Fig. 4.17. It takes several query parameters such as *start_time*, *end_time*, *weekday*, *study_area_id* and *return_type*. These parameters are the same as in ÖV Güteklassen endpoint except for the station configuration. The result with the default *return_type* is a GeoJSON file with features for every station, and information on the number of departures (trips) for the selected time window. The number of departures in the response is defined as the total, and the average per hour is calculated during the visualization step by dividing by the number of hours. The API endpoint also supports the return type in Geobuf format, which returns a compressed binary response which makes it faster to transfer from server to client. This response type is often used for study areas with many station features and is decoded in the client.

User Interface

The indicator is added as a layer in the indicators section and has the same controls as ÖV Güteklassen for visibility and transparency settings. When the layer is activated, the indicator time picker component is shown on the map with the default value set on *Monday 07:00 - 09:00*. The map style for this layer is implemented using pie charts as a multivariate map, shown in Fig. 4.17. The service types are represented with different colors based on the configuration of the study area. The radius of the pie chart is defined by the total number of departures and is limited from a range of 5-pixel to 20-pixel.

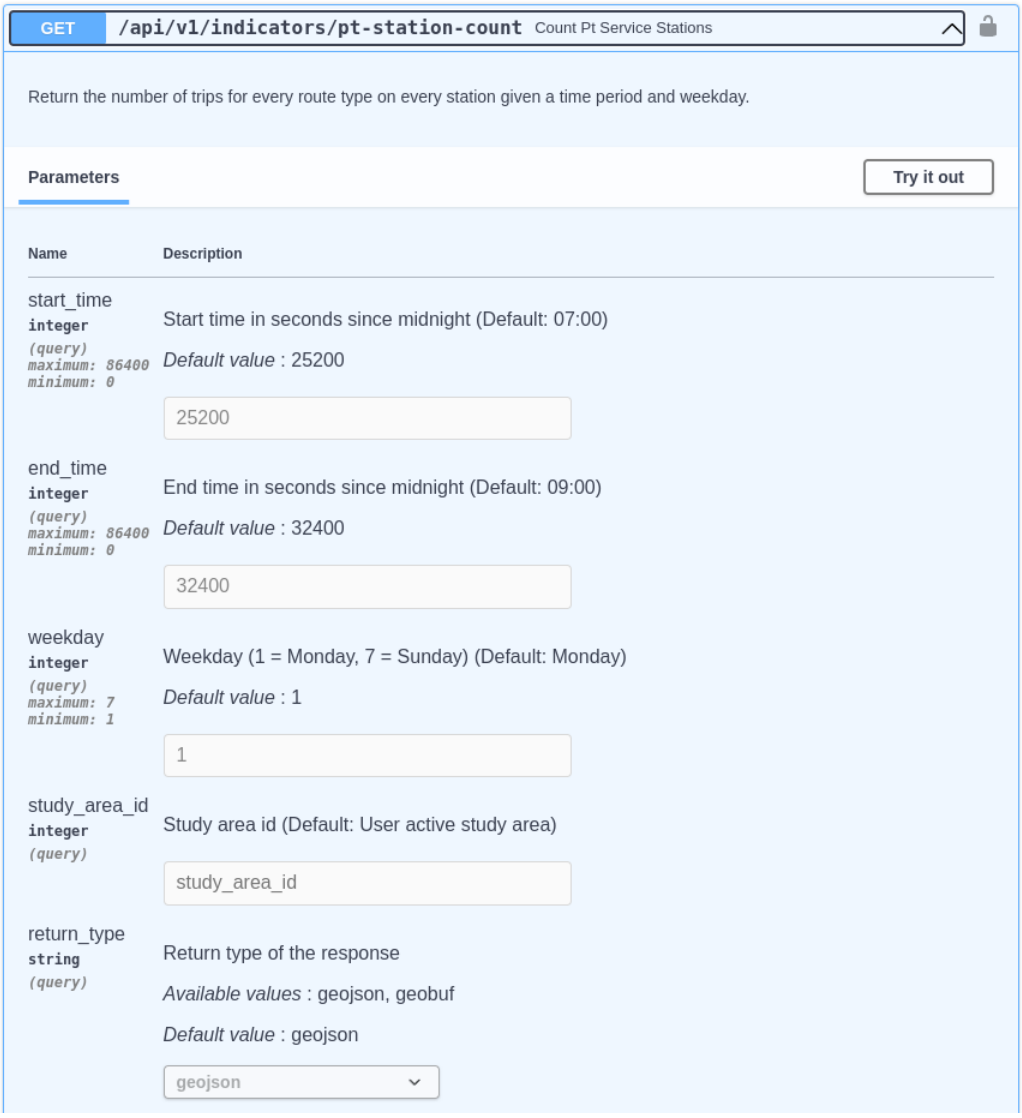


Figure 4.16.: Station Count API - (Swagger)

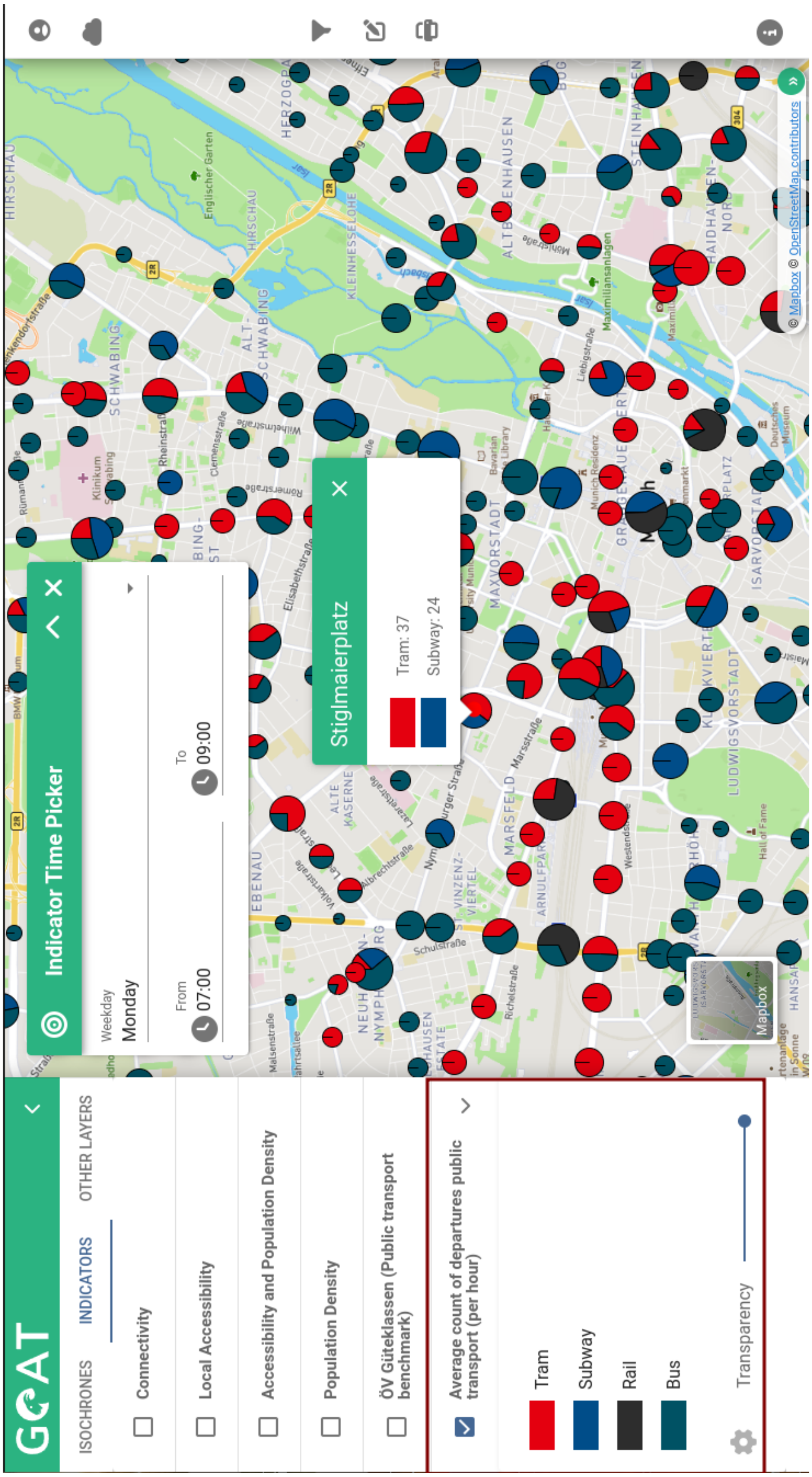


Figure 4.17.: User Interface - Station Count

5. Outlook and conclusion

5.1. Outlook

Even though this thesis project aimed to implement all the indicators selected from the workshop, some of the indicators are still a work in progress and not completely finalized within the timeline. From the outcome of the workshop results, the most important accessibility measures are the "Dynamic Public Transport Isochrone", "Station departure count" and "ÖV-Güteklassen," which are already implemented and explained in the results section. The core limitation preventing the implementation of the other indicators is the complexity of the data requirement and computation optimization to have indicators in a web-based environment, from which users expect fast analyses without long calculation times. A more in-depth study is needed to find the best way to implement and maintain the same quality as already existing indicators in performance and UI/UX. The following steps are suggested for the next phase.

Usability tests

Having more detailed feedback for each indicator is crucial and improving further map visualization and UI/UX. It is suggested continuously stay in the SDLC iteration. Usability testing using the "Unmoderated remote usability testing" method will be conducted as the first step before implementing the other indicators. Furthermore, it is suggested to work with techniques such as eye-tracking to understand the user-flow better.

Practice application of accessibility measures

It is a validation step in which analyses are tested in real-world scenarios outside the usability tests. This step is suggested to be carried out in collaboration with partners when working on real-world planning documents. This step could demonstrate whether the analysis is actually of added value. Based on the outcome of this step, necessary actions are taken to improve or reconsider the implementation. As this method gives a more profound understanding from a different perspective, the information can be used to have a better strategy when implementing accessibility measures in various study areas. For example, some indicators might be relevant in the German context, while others are more relevant in other countries.

Integration of scenario building functionality
The tool GOAT is known for its scenario-building capabilities. Though, the current implementation of the public transport indicators does not allow the assessment of scenarios such as new public transport infrastructure. This feature quickly reveals the impacts of decision-making; from the discussions with the experts from practice, it was realized that scenario building is a very desired feature. The data model within the implementation was designed from the beginning to facilitate scenario building in the future. The complexity, therefore, is mostly in optimizing the codebase to be more scaleable in work with different routing graphs, which would be needed for developing scenarios in a multi-user scenario.

Thresholds and calibration of accessibility measures
When developing and using the different accessibility measures, it was realized that some of the indicators, such as the "ÖV-Güteklassen," are very sensitive towards the calibration with the settings. Accordingly, slight changes in the settings can heavily influence the results; the same is true for the "Dynamic Public Transport Isochrone," which can be computed using different departure probabilities. The challenge is that the indicators are often not standardized by a formal authority. Therefore, it is suggested to test different settings using methods such as sensitivity analyses. By that, the variations of the results can be explored and documented.

5.2. Conclusion

With the new development in web technologies, public transport analysis nowadays can be provided in an easy and lightweight web interface that hides the complex calculation workflows which usually come when accomplished using conventional methods. This thesis project attempted to identify a set of indicators most relevant in the planning practice and integrate them into a web tool that brings the data-driven analysis closer to experts. Even though the solution was implemented and tested in a

German context, the aim was to provide a standardized way of using the indicators. Furthermore, it was tested which indicators are requested by the planning practice. It was realized that the suggested approach was favored, and experts contended a high demand for such a solution.

Despite the high efforts invested in this thesis, there are mainly two significant limitations. First, the assessment of the most suitable indicators was not clearly following a structured and scientifically sound assessment method. Instead, a multi-criteria assessment was favored that incorporated findings from the literature and own interpretations. Another major limitation was that there were not conducted enough workshops and usability tests with potential users of the analyses. As the development effort was underestimated, there was not enough time to still test the results with users. Therefore, the results were not independently tested but only examined by a small group of experts at Plan4Better. Despite the legitimate criticism of the presented work, the realized development is of high novelty and maturity. The feedback from the experts showed that the recognized development has the potential to support real-world planning problems and facilitate more individuals in performing data-driven analyses in public transport planning.

A. Appendix

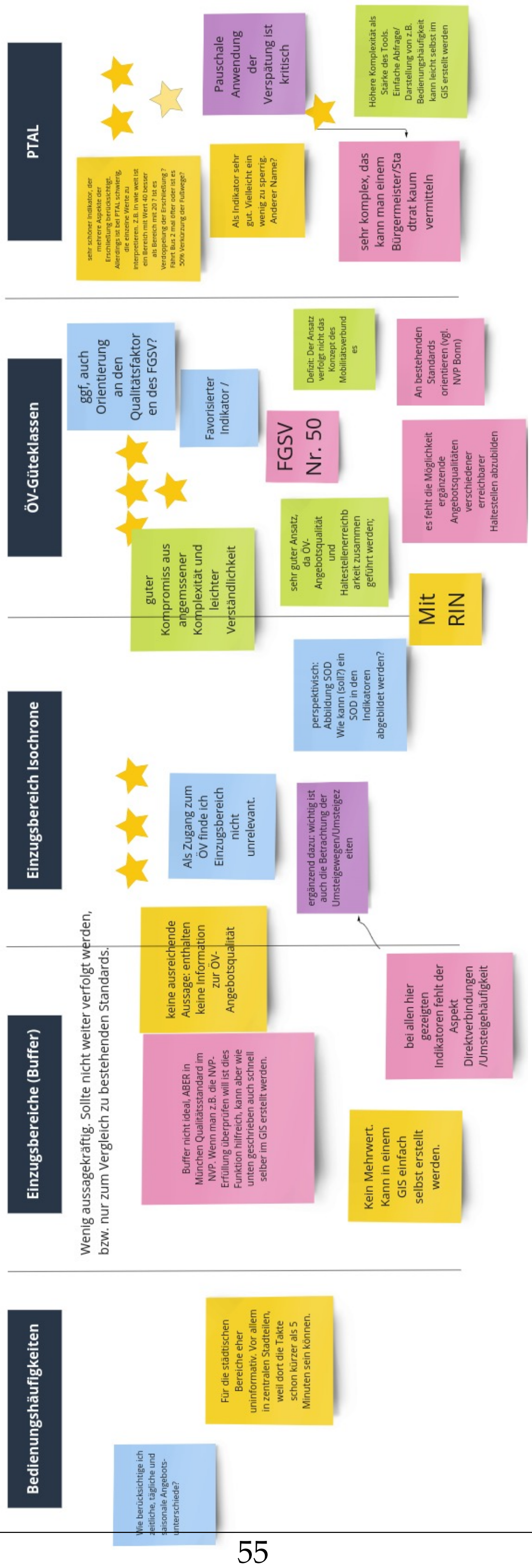


Figure A.1.: Summary discussion accessibility to public transport

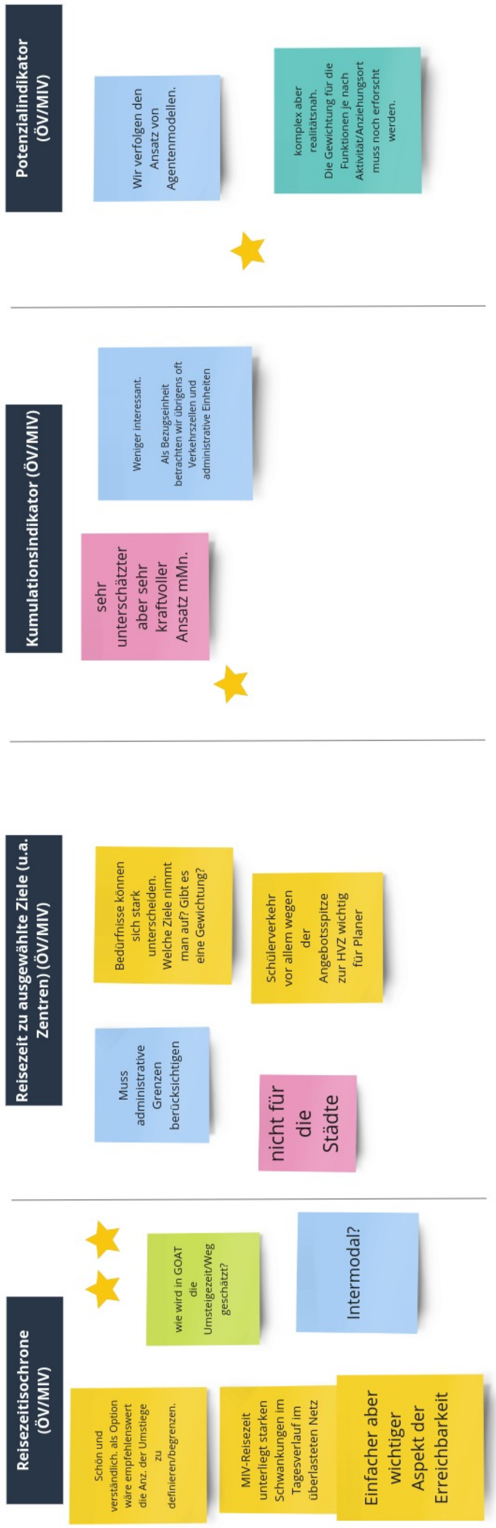


Figure A.2.: Summary discussion accessibility by public transport

List of Figures

List of Tables

Bibliography

[1] P. Schiller and J. Kenworthy. *An Introduction to Sustainable Transportation: Policy, Planning and Implementation*. en. 2017. URL: https://www.routledge.com/An-Introduction-to-Sustainable-Transportation-Policy-Planning-and-Implementation/Schiller-Kenworthy/p/book/9781138185487 (visited on 09/23/2022).

[2] D. Banister and Y. Berechman. "Transport investment and the promotion of economic growth". en. In: *Journal of Transport Geography* 9.3 (Sept. 2001), pp. 209–218. ISSN: 09666923. DOI: 10.1016/S0966-6923(01)00013-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0966692301000138 (visited on 09/23/2022).

[3] W. Hansen. "Accessibility & Residential Growth". PhD thesis. Massachusets Institute of Technology (MIT), June 1959.

[4] K. T. Geurs and B. van Wee. "Accessibility evaluation of land-use and transport strategies: review and research directions". en. In: *Journal of Transport Geography* 12.2 (June 2004), pp. 127–140. ISSN: 09666923. DOI: 10.1016/j.jtrangeo.2003.10.005. URL: http://linkinghub.elsevier.com/retrieve/pii/S0966692303000607 (visited on 11/27/2018).

[5] A. Hull, L. Bertolini, and C. Silva. *Accessibility Instruments for Planning Practice*. en. OCLC: 1025996720. Edinburgh: Heriot-Watt University ; Porto : Departament of Geography Planning and International Development Studies Faculty of Engineering of Oporto University ; Amsterdam : 2012. ISBN: 978-989-20-3187-3.

[6] E. Papa, C. Silva, M. te Brömmelstroet, and A. Hull. "Accessibility instruments for planning practice: A review of European experiences". English. In: *The Journal of Transport and Land Use* 9.3 (July 2015), pp. 57–75. DOI: 10.5198/jtlu.2015.585.

[7] E. Pajares, B. Büttner, U. Jehle, A. Nichols, and G. Wulfhorst. "Accessibility by proximity: Addressing the lack of interactive accessibility instruments for active mobility". en. In: *Journal of Transport Geography* 93 (May 2021), p. 103080. ISSN: 09666923. DOI: 10.1016/j.jtrangeo.2021.103080. URL: https://linkinghub.elsevier.com/retrieve/pii/S0966692321001332 (visited on 07/02/2021).

[8] GOAT-Community. *Github-Account GOAT-Community*. en. 2022. URL: https://github.com/goat-community (visited on 08/11/2021).

[9] M. te Brömmelstroet, C. Curtis, A. Larsson, and D. Milakis. "Strengths and weaknesses of accessibility instruments in planning practice: technological rules based on experiential workshops". In: *European Planning Studies* 24.6 (Mar. 2016), p. 21. DOI: 10.1080/09654313.2015.1135231.

[10] K. Geurs and B. van Wee. "Accessibility evaluation of land-use and transport strategies: review and research directions". en. In: *Journal of Transport Geography* 12.2 (June 2004), pp. 127–140. ISSN: 09666923. DOI: 10.1016/j.jtrangeo.2003.10.005. URL: http://linkinghub.elsevier.com/retrieve/pii/S0966692303000607 (visited on 11/27/2018).

[11] M. te Brömmelstroet, C. Silva, and L. Bertolini. *Assessing usability of accessibility instruments*. en. OCLC: 883953178. Amsterdam: COST office ; 2014. ISBN: 978-90-90-28212-1.

[12] D. Levinson and D. King. "Transport Access Manual: A Guide for Measuring Connection between People and Places". en. In: *Committee of the Transport Access Manual, University of Sydney* (Jan. 2020), p. 229. URL: https://hdl.handle.net/2123/23733.

[13] D. Delling, T. Pajor, and R. F. Werneck. "Round-Based Public Transit Routing". en. In: (), p. 11.

[14] E. Pajares. "Assessment of the usefulness of the accessibility instrument GOAT for the planning practice". en. In: (2022), p. 16.

[15] Plan4better GmbH. *Nachhaltige Städte und Regionen planen | Plan4Better*. German. 2022. URL: https://plan4better.de/ (visited on 09/16/2022).

[16] Geofabrik GmbH. *GEOFABRIK // Home*. 2022. URL: https://www.geofabrik.de/ (visited on 09/16/2022).

[17] Landesamt für Digitalisierung Breitband und Vermessung. *Bayerische Vermessungsverwaltung - Produkte - ATKIS Basis-DLM*. 2022. URL: https://www.ldbv.bayern.de/produkte/atkis-basis-dlm.html (visited on 06/26/2021).

[18] GTFS.DE. *GTFS.DE - GTFS für Deutschland*. de. 2022. URL: https://gtfs.de/ (visited on 04/19/2022).

[19] *Introduction to Software Engineering/Process/Life Cycle - Wikibooks, open books for an open world*. [Online; accessed 10. Oct. 2022]. Sept. 2022. URL: https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Life_Cycle#/media/File:SDLC_-_Software_Development_Life_Cycle.jpg.

[20] FGSV, ed. *Hinweise für die Qualitätssicherung im ÖPNV*. ger. Ausg. 2006. FGSV 145. Köln: FGSV, 2006. ISBN: 978-3-937356-78-5.

[21] Bayerisches Staatsministerium für Wirtschaft, Infrakstruktur, Verkehr und Technologie. *Leitlinie zur Nahverkehrsplanung in Bayern*. Tech. rep.

[22] Bundesamt für Raumentwicklung ARE. *ÖV-Güteklassen Berechnungsmethodik ARE*. de. Grundlagenbericht. 2022, p. 10. URL: https://www.are.admin.ch/dam/are/de/dokumente/verkehr/oev-gueteklassen-berechnungsmethodikare.pdf.download.pdf/oev-gueteklassen-berechnungsmethodikare.pdf.

[23] H. Hiess. *Entwicklung eines Umsetzungskonzeptes für österreichweite ÖV-Güteklassen*. Tech. rep. 2017, p. 35. URL: https://www.oerok.gv.at/fileadmin/user_upload/Bilder/2.Reiter-Raum_u._Region/1.OEREK/OEREK_2011/PS_RO_Verkehr/OeV-G%C3%BCteklassen_Bericht_Final_2017-04-12.pdf.

[24] Transport for London. *Measuring Public Transport Accessibility Levels*. 2020. URL: https://s3-eu-west-1.amazonaws.com/londondatastore-upload/PTAL-methodology.pdf.

[25] S. Cooper, P. Wright, and R. Ball. *Transport for London*. 2009.

[26] R. H. M. Pereira, M. Saraiva, D. Herszenhut, C. K. V. Braga, and M. W. Conway. "r5r: Rapid Realistic Routing on Multimodal Transport Networks with R ^{5} in R". en. In: *Findings* (Mar. 2021). ISSN: 2652-8800. DOI: 10.32866/001c.21262. URL: https://findingspress.org/article/21262-r5r-rapid-realistic-routing-on-multimodal-transport-networks-with-r-5-in-r (visited on 10/10/2022).

[27] FGSV. *Richtlinien für integrierte Netzgestaltung (RIN)*. FGSV. Köln: FGSV-Verl, 2008.

[28] J. F. de Queiroz Neto, E. M. d. Santos, and C. A. Vidal. "MSKDE - Using Marching Squares to Quickly Make High Quality Crime Hotspot Maps". en. In: *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. Sao Paulo, Brazil: IEEE, Oct. 2016, pp. 305–312. ISBN: 978-1-5090-3568-7. DOI: 10.1109/SIBGRAPI.2016.049. URL: https://ieeexplore.ieee.org/document/7813048/ (visited on 10/01/2022).