Web-Based Geothermal Energy Potential Mapping and Analysis for Berlin

Estimating the shallow geothermal energy potential of Berlin

Tim Kröger

August 10, 2024

Master of Science

Matrikelnummer: 0468029

Technische Universität Berlin
Institut für Geodäsie und Geoinformationstechnik

Erstprüfender: Prof. Dr. Kada
Martin.kada@tu-berlin.de

Zweitprüfender: Prof. Dr. Fernandez-Steeger
Fernandez-steeger@tu-berlin.de

A Master thesis in cooperation with the GASAG-Solution-Plus

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit eigenständig ohne Hilfe Dritter und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe. Alle Stellen die den benutzten Quellen und Hilfsmitteln unverändert oder sinngemäß entnommen sind, habe ich als solche kenntlich gemacht.

Sofern generische KI-Tools verwendet wurden, habe ich Produktnamen, Hersteller, die jeweils verwendete Softwareversion und die jeweiligen Einsatzzwecke (z.B. sprachliche Überprüfung und Verbesserung der Texte, systematische Recherche) benannt. Ich verantworte die Auswahl, die Übernahme und sämtliche Ergebnisse des von mir verwendeten KI-generierten Outputs vollumfänglich selbst.

Die Satzung zur Sicherung guter wissenschaftlicher Praxis an der TU Berlin vom 8. März 2017. https://www.static.tu.berlin/fileadmin/www/10000060/FSC/Promotion____Habilitation/Dokumente/Grundsaetze_gute_wissenschaftliche_Praxis_2017.pdf
habe ich zur Kenntnis genommen.

Ich erkläre weiterhin, dass ich die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Berlin, den August 10, 2024

Tim Kröger

Tim Kröger

Masterarbeit von Tim Kröger

Eine Masterarbeit an der Technischen Universität Berlin

Titel:

Web-Based Geothermal Energy Potential Mapping and Analysis for Berlin

Schätzung des oberflächennahen geothermischen Energiepotenzials von Berlin

Institut für Geodäsie und Geoinformationstechnik
Dr. Löwner & Prof. Dr. Kada

Institut für Angewandte Geowissenschaften
Prof. Dr. Fernandez-Steeger

In Zusammenarbeit mit der GASAG Solution Plus
Dr. Alexander Meeder

Zusammenfassung

Die Nutzung der oberflächennahen Geothermie ist ein Schlüsselfaktor der Energiewende Deutschlands. Allerdings ist die digitale Infrastruktur, die diese Entwicklung unterstützt, bisher erst dabei zu entstehen. In dieser Arbeit wurde eine Software entwickelt, welche eine Erstabschätzung des geothermischen Potenzials im Raum Berlin auf der Basis öffentlich zugänglicher Daten ermöglicht. Die entwickelte Software ist in der Lage flurstücksgenau ein theoretisches Erdwärmesondenfeld zu modellieren, die über ein Jahr entnehmbare Wärme abzuschätzen und die Wirtschaftlichkeit des Modells zu bewerten.

Die Anwendung wurde verwendet, um zu ermitteln, welche Flächen in Berlin für Erdwärmesonden genutzt werden können und welches technisch theoretisch nutzbare Wärmepotenzial daraus gewonnen werden kann. Diese Analyse wurde für verschiedene Szenarien ausgelegt, um zu ermitteln, welcher Anteil des Wärmebedarfs Berlins durch oberflächennahe Geothermie versorgt werden kann. Die Analysen zeigen, dass ∼34% des gesamten Berliner Wärmebedarfs durch den alleinigen Einsatz von Erdwärmesonden gedeckt werden könnten. Der Heizwärmebedarf von Wohnungen allein könnte zu ∼46% gedeckt werden. Es wird ebenfalls aufgezeigt, dass eine Änderung in den Vorschriften, welche die Nutzung landeseigener öffentlicher Flächen erlauben würde, die Deckung des gesamten Wärmebedarfs Berlins auf ∼59% erhöhen kann. In diesem modellierten Szenario könnte somit der Großteil des Berliner Wärmebedarfs, bei entsprechender anteiliger saisonaler Regeneration des Untergrundes, allein durch die Nutzung oberflächennahenr Geothermie gedeckt werden.

Die entwickelte Anwendung ist öffentlich online zugänglich [1]. Der Quellcode des Projekts ist ebenfalls öffentlich auf GitHub [2] verfügbar und kann unter der MIT Lizenz verwendet werden [3].

Master Thesis by Tim Kröger

A Master Thesis at the Technische Universität Berlin

Titel:

Web-Based Geothermal Energy Potential Mapping and Analysis for Berlin

Estimating the shallow geothermal energy potential of Berlin

Institute of Geodesy and Geoinformation Science

Dr. Löwner & Prof. Dr. Kada

Institute of Applied Geosciences

Prof. Dr. Fernandez-Steeger

In cooperation with the GASAG Solution Plus

Dr. Alexander Meeder

Abstract

The utilisation of near-surface geothermal energy is a key factor in the energy transition currently undergoing in Germany. Yet to date, the digital infrastructure supporting this development is just beginning to emerge. In this thesis publicly available data was used to construct an application which can provide an initial estimate of the geothermal potential in the area of Berlin. It is capable of detecting land parcels for requested coordinates, perform a theoretical modelled setup of borehole heat exchangers for a ground source heating system, estimate the heat that can be extracted over a year and evaluate the economic efficiency of the model.

The application was then used to determine which areas of Berlin can be utilized for ground source heat pump systems and which theoretical heat potential can be extracted from them. This analysis was performed for different scenarios and related to different estimates of the heat demand of Berlin. The evaluation shows that this theoretical potential can cover ∼34% of all the heat demand of Berlin. Only the residential area alone can be utilized to cover ∼46% of Berlin's domestic heating. It was also shown that a change in regulations, which would allow the use of state own public areas, would increase the coverage of all of Berlin's heat demand to ∼59%. In this modelled scenario, the majority of Berlin's heating demand could be covered by the use of near-surface geothermal energy alone, without the utilization of other technologies. The developed application is publicly accessible online [1]. The source code of the project is also publicly available on GitHub [2] and the code provided there can be used under the MIT licence [3].

Contents

List of Tables

List of Figures

List of Equations

List of Abbreviations and Symbols

API	Application programming interface
BHE	Borehole heat exchanger
ENUM	Enumeration type - A set of named constants of the underlying integral numeric type
GeoJSON	A GeoJSON is an extension of the JSON format used to encode geographic data structures
GSHP	Ground source heat pumps
HASH	A mathematical algorithm converting input data into a fixed-size string of bytes, crucial for data integrity and security in computing
HTTPS	Hypertext Transfer Protocol Secure
JSON	JSON (JavaScript Object Notation) is a lightweight data-interchange format that is built for standardized communication between different systems
RAM	Random-access memory
SenSBW	Senate Department for Urban Development, Building and Housing of Berlin
SRID	Spatial reference identifier
TW	Terawatt
URI	Uniform Resource Identifier
UTM-33N	European Terrestrial Reference System 1989 / Universal Transverse Mercator zone 33N
WCS	Web-Coverage-Services
WFS	Web-Feature-Services
WMS	Web-Map-Services

1. Introduction

The development of new technological advancements has drastically expanded the possibilities of available resources for energy production. Progress has been made in the area of domestic heating in particular [4]. Through the improvements in geothermal systems and heat pumps, which are able to heat or cool fluids, geothermal applications are becoming more attractive as a way to produce heat energy. These ground source heat pumps (GSHP) systems focus on space heating and cooling, by exploiting stable ground and groundwater temperatures at shallow depths (10-200m) [5].

With climate change as a real threat to our future, geothermal systems support a transformation from fossil fuels to a solution that minimises the effect of heating on global warming. By using the available ground temperature as a base, the energy consumption can be reduced by 30%-40% in contrast to electrical heating [6]. In correspondence the air pollution emissions can be reduced up to 44% compared to air source heat pumps and 72% compared to electrical resistance heating [5]. Ground source heat pumps offer right now the most energy efficient way to provide heating and cooling in many applications, as they can act as renewable energy sources when properly regenerated [6]. Therefore, geothermal energy offers enormous potential for the future heating and cooling of buildings [7].

Three main systems can be used to generate geothermal energy in shallow depths. The first one are Geothermal collectors. These consist of horizontal pipe structures which are located close to the surface and cover therefore a wide but not depth area. The second system is a BHE field, which follows the same principle, but with a vertical setup. As the third system, geothermal wells can be utilized to access ground water directly and deliver it to the heat pump [8]. This system uses the heat of underground water reservoirs, but is therefore more elaborate to plan and construct. All these system use a GSHP for the heat exchanging.

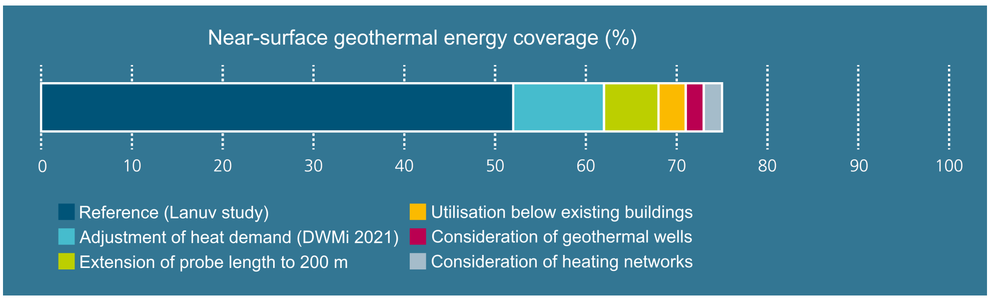


Figure 1: Heat extraction coverage by ground source heat pumps in the area of NRW. Translated figure from the "Roadmap Oberflächennahe Geothermie" [9] by the Fraunhofer Institute.

The Fraunhofer Institute also came to the conclusion that around 75% of the heating demand of all of North Rhine-Westphalia (NRW) can be covered by GSHP systems [9], as shown in Figure 1. From these used systems, borehole heat exchanger (BHE) are often the best suited of these systems [7]. However, this value estimated by Fraunhofer institute must be considered with caution as it includes mostly rural areas with plenty of open space, where this system can be installed more easily. Borehole Heat exchanger work by exchanging heat between the ground and the refrigerant fluid, where the ground temperature is used as the base temperature. This is done by a network of borehole heat exchangers. The BHEs are pipe constructs, in which a fluid circulates as a heat transfer medium, which absorbs the underground heat and transports it to the surface [10]. The system takes advantage of the fact that ground temperatures are fairly stable throughout the year [5]. Ground temperatures of 10–15°C, which are normally considered as cold, contain useful heat that is continuously replenished by the sun [6]. By just applying a little more energy, an electric heat pump can raise the source temperatures to the needed temperature level for heating [6]. The ground temperature given in the first 100m is not subject to thermal energy of the inner core of the earth but almost exclusively renewed by the air temperature and solar energy. The technology relies on the fact that the ground temperature is stable to create heat deposits. In winter the ground temperature is higher than the air temperature, which makes it more efficient to exchange heat with the ground [5]. This also cools the ground slowly. In summer the ground temperature is lower and can be used for cooling purposes. This, on the other hand, warms the soil again for use in the following winter [6]. When no active cooling exists, the heat of the building on warm days can still be fed into the ground to regenerate it. Because ground source heat pumps do not require thermal anomalies, they are suitable for many regions, even if no ordinary geothermal conditions are given [5]. Especially countries that are not known for geothermal resources have ranked high in geothermal utilization in shallow depth due to the use of geothermal heat pumps [5]. To find possible areas where geothermal systems can be installed, applications are on the rise which can identify these possible areas from the far. They use broadly available input data to make an estimation without the need to take measurements on the site.

1.1. Current Evaluation of GSHP

The evaluation of a ground source heat pump system is highly complex. It can be divided into three parts that make up the analysis. The technical parameters of the BHE, the geophysical parameters and the heat usage of the consumer.

The technical parameters cover multiple factors like power consumption, flow rate, thermal conductivity of the piping material and the composition of the refrigerant fluid mixture [11]. All these factors have a significant influence on the heat extraction. For example, differences in pipe diameter in conjunction with flow rate can lead to a change from laminar to turbulent flow, which in itself has a major impact on the performance. Because of the complexity of the system and the influence of many factors on each other, simulations are often used to evaluate a BHE field. One common software used is the "Earth Energy Designer" [12].

The second part of the analysis focuses on the geophysics of the subsurface. This involves constructing a detailed underground model and discretizing the relevant parameters. This is done to capture all necessary details. The system is also highly complex and consists of multiple factors that influence each other. These factors include the underground temperature, the thermal conductivity of the ground, the presents of ground water, its temperature, hydraulic conductivity (permeability), porosity, water and matrix tempera-

ture, the flow rate of the ground water and the geological sections present at the drilling spot. To determine these parameters, field investigations are conducted rather than relying solely on software. For analysation, software like “Earth Energy Designer” [12] or “Groundwater Energy Designer” [13] are used. These can model BHE but can’t provide the numeric consideration of the underground parameters. Because of that they are used in combination with numerical software, such as the finite element software Feflow [14] and Petrel [15] to validate the data and calibrate the model.
The last part is the annual heat consumption and utilization of the user. This factor influences the technical side. For example, higher consumption can lead to a change in the flow rate in the BHE, and a possible regeneration of the system in non-heating periods also has a significant influence on the ground temperature.
In addition, a variety of input data is required. These simulations are therefore not optimised for a quick initial assessment and require prior knowledge in order to be used correctly. While this is the most accurate way to evaluate a GSHP system, this procedure is time consuming. Therefore, simulations are mostly only performed when interest is present from a property owner. In order to obtain a quick initial assessment of the geothermal potential, web-based applications are on the rise.

1.1.1. Related Web-applications

Applications for a fast web-based assessment of the feasibility of a geothermal system and its potential already exist to support the first planning steps. But these applications are right now not available for every federal state of Germany. The most developed version of these are the “Geothermieportal LBGR Brandenburg” of Brandenburg [16], the “Geothermie viewer NRW” of NRW [17], the viewer of the “Landesamt für Geologie, Rohstoffe und Bergbau (LGRB)” from Baden-Württemberg [18], the “Energie-Atlas” from Bavaria [19] and the “Geothermal Information System (GeotIS)” from the Leibniz Institute for Applied Geophysics (LIAG) [20]. The first four of them are restricted to the area of their federal state. GeotIS is currently on its way to cover multiple federal states, stating in the north of Germany. This tool can currently determine shallow ground geothermal energy analyses for Mecklenburg-Vorpommern and Schleswig-Holstein and deep geothermal analyses for the whole of germany.
The given nature of the federal states of germany leads to the development of multiple applications in this field, which all try to achieve the same result. Another reason for this is that data formatting and availability changes between federal states and therefore make a development of a software that covers the entirety of Germany very resource consuming.

The already mentioned existing applications often base their efficiency evaluation just on one single parameter, the thermal conductivity. This is an important key parameter, but the solely use of one factor can result in an insufficient evaluation. Only the “Energie-Atlas” provides more detailed information to the heat that can be extracted in a `10x10m` raster. For the identification of restricted spaces, the tools focus on water protection- and hydrogeologically sensitive areas. Although these are important to consider, a lot of restrictions are left out. Non of the existing applications therefore provide the possibility to safely detect if a geothermal system can actual be build on a chosen area or not. An unique advantage the “Geothermie viewer NRW” offers is that it allows to virtually simulate a technical setup. The application can be provided with data about the sensor, the fluid composition, the expected power profile of the end-user and a thermal response test from the chosen area to calculate an even more accurate estimation. But to perform this, extended information are needed. An advantage from the GeotIS system is that it can estimate the heat demand from buildings and can even categorize them into different sectors. This can be helpful to determine where GSHP systems are needed. Currently none of these systems can model BHE configuration or estimate the potential for a given land parcel or area.

1.2. Objectives

To overcome these limitations a web application for the geothermal energy estimation and GSHP planning was developed in conjunction with this thesis for the area of Berlin. It provides a more in depth analyses than the other currently existing tools. The evaluation is performed in a land parcel resolution rather than a single point, as in most cases this is the area of interest. It can calculate the area that is usable for drilling and performs a theoretical modelling of a BHE constellation on the property. With the modelled BHEs the developed tool can perform an estimation of the potential heat extraction and a rating of the economic efficiency. The application shows that an initial assessment is possible without the use of computational and time consuming simulations. With the source code publicly available, the tool can be used as a basis for further development in this sector. The second objective of this thesis is to use the developed application to estimate the heat coverage that GSHP systems can provide to cover Berlins heat consumption. This is done in order to improve the data availability, because so far only rough estimations are available of how much of the heat consumption can be covered by the use of GSHP systems in the area of Berlin [10]. The area of Berlin offers hereby a particularly valuable observation, because it has, as a city, a high heat demand that needs to be covered, paired with limited areas that are available for BHE installation. Until now, the generation of heat in Berlin has been predominantly dominated by fossil fuels. According to the latest figures from 2021, around 43% of homes in Berlin are heated with district heating, 37% with natural gas and 16% with heating oil. The fuel used to generate district heating is made up of: 61% natural gas, 16% stone coal, 11% renewable energies, 11% other and 1% natural oils [21]. The simulations performed in this thesis can therefore show to which degree GSHP systems can replace this heavy fossil fuel usage.

2. Methodology and Data

This thesis consists of two major segments. The first segment covers the methodology and the second the concrete implementation of the application, which was build in conjunction with this thesis. The methodology is covered in this chapter and the implementation in chapter 3 Implementation. The methodology provides a formal description of the structure of the application and the methodology of the scientific evaluation of GSHP systems for the area of Berlin. The application structure can hereby be further split into the data acquirement and actualisation, which is covered in chapter 2.1 - 2.1.1 and the geothermal assessment of the tool, which is covered in chapter 2.2 - 2.7. The methodology of the scientific evaluation of Berlin is explained in chapter 2.8.

The application is designed in such a way that it always acquires the latest available data and makes a geothermal estimation based on it. These functionalities are provided over multiple application programming interfaces (API), which are hosted by a server. APIs are a standardized communication interface that allows different software applications to interact and share data seamlessly. They therefore allow everyone to utilize the geothermal assessment developed. The application then sends the result that corresponds to the request back over to the API. In correspondence to the server, a website was also developed, which allows an easy communication with the API by the use of a web map as a user interface.

2.1. Source Data

The data used for this thesis is exclusively open-source data. Most of the used data is provided over the website Fis-Broker. The data is provided over the website Fis-Broker [22] which is a geo-portal managed by the "Senate Department for Urban Development, Building and Housing" (SenSBW) of Berlin.
All data is therefore used under the licence BY-2.0 from Germany [23]. For this thesis, the given data was integrated into a tool that was developed and used for scientific evaluation. The source data underwent modification and editing during this process.

The used data consists exclusively of geo-data, which is provided by Web-Feature-Services (WFS) and Web-Coverage-Services (WCS).
Web-Map-Services (WMS) were not usable for calculations because they can only supply pixel information and are not developed for exact measurements. All data used can be found in Table 1.

ALKIS Berlin Landparcels [24]	Data from the official real estate cadastral information system (ALKIS) - The parcel is a clearly geometrically defined part of the earth's surface and is a booking unit in the real estate cadastre.
Geothermal potential - specific extraction capacity [40-100m | 1,800 - 2,400 h/a] [25]	Factual data for the geothermal potential distribution of Berlin with the distribution of the specific extraction capacity in W/m for boreholes with a depth of 40 to 100m and annual operating hours of 1,800 to 2,400 for heat pumps. (1,800 applies for heating only and 2,400 for heating operation with hot water preparation).
Geothermal potential - specific thermal conductivity [40-100m] [26]	Factual data for the geothermal distribution of the specific thermal conductivity in W/mK up to a depth to 100m, depending on the geological layer sequence at the location.

Geothermal potential - Restricted areas [27]	Factual data on restrictions on the use of geothermal energy (restricted areas). The utilisation is not permitted in water protection areas. In areas with increased salt content in the groundwater, artesian groundwater and in areas of high rupel clay layers, the utilisation is permitted with restrictions.
Expected highest groundwater level (zeHGW) [28]	Factual data on the expected highest groundwater level (zeHGW) in the Berlin glacial valley and Panketal. The zeHGW describes the highest groundwater level that can occur in an area as a result of weather conditions without artificial intervention.
Groundwater levels of the main aquifer 2020 [29]	Groundwater levels of the main aquifer from May 2020.
Groundwater levels of the Panketal aquifer [30]	Groundwater levels of the Panketal aquifer from May 2020.
Groundwater quality [Ammonium, Boron, Chlorine, Electric conductivity, Potassium, Orthophosphate, Sulfate] [31]	Groundwater exposure and quality.
Groundwater measuring points [32]	Location of the groundwater measuring points owned by the Senate Department for Mobility, Transport, Climate Protection, Environment and the “Berliner Wasserbetriebe”.
Groundwater temperature 20 - 100m below ground surface [33]	Distribution of the groundwater temperature at 20 to 100m below ground level. Both current temperature measurements from 2020 and older measurements from 2015-2019 were taken into account. Measurements prior to 2020 were extrapolated using an area-specific adjustment coefficient.
Water protection areas [34]	An overview of Berlin’s water protection areas.
Holstein layer [35]	Clay layer, separating the groundwater layers.
ALKIS Berlin utilisation [36]	Data from the official real estate cadastre about the utilisation of areas.
Tree population Berlin [37]	Factual data of planted trees with information on tree species, address, year of planting and height.

Table 1: Used source data

2.1.1. Data acquirement and actualisation

To provide a fast estimation by the developed application all the mentioned data is retrieved from its original source (SenSBW) and saved as a local copy. This is done by API requests, which are pre-written, to access specific datafiles. After downloading the data, it gets pre-processed to extract the data correctly and to sort it. This is done separately for the geometry and parameter information contained in the file. The parameter

information needs hereby a more in-depth processing, because naming schemas needs to be handled correctly. The standardized data then get stored on a database, where the information is accessible for any further computations. When storing spatial data on the database, all information is spatially indexed, which improves the search time.
The second step of the data acquirement is to ensure that the locally stored data is always up to date. To do this the source data gets checked at a regular basis and compared to the locally stored data. When the source data differs from the stored file, the old data gets replaced by the new information. With the help of these regular updates, the application works always consistently on the latest available data.
The detailed layout and operations performed are shown in chapter 3.2.

2.2. BHE simulation base

To get an estimate of the heat that can be utilized by a borehole heat exchanger a simulation needs to be performed. To do this in the reference frame of a web assessment, the default GSHP configuration of the SenSBW for the area of Berlin [38] is used, which is evalauted for a defined technical setup and the geophysical parameters of the observed location. The parameter of the simulation are shown in Table 2.

With these assumptions and the usage of location dependent parameters like the thermal conductivity of the underground, heat extractions for different depths were estimated for the entirety of Berlin. This model is then used as a reference to estimate all heat extractions. It has to be noted, that this setup is not always the best setup, which can be deployed, but it serves as an average configuration to estimate the potential heat extraction. The double U-tube system is commonly used and the assumption of `2400` heat pump operating hours represent the average use for domestic heating without hot water preparation.
This simulation represents, with its defined conditions, a conservative estimation and is therefore expected to underestimate the total heat extraction of a BHE. The reason for this is that, for example, the average ground temperature is a fixed variable that is set to `9°C`. In reality the ground temperature is location dependent and has a range from `9.5°-12.5°` in the area of Berlin. The lower temperature boundary of the fluid with `1.5°C` is also quite high and can often be as low as `-3°C` in a practical use case. Lastly the groundwater flow is not considered, which can provide a significant improvement to the heat extraction. This setup is also simulated for a configuration of two BHEs. If several BHEs are present, the resulting heat extraction must be adjusted to take the cross-reference of the BHEs to each other into account.
This representative setup is used for all further estimations of the heat extractions of BHEs.

2.3. Area determination

With the reference simulation provided, it is possible to evaluate a GSHP system for a requested area. To do this the first step is to determine which part of the drilling area can actually be used to deploy BHEs. The area suited for geothermal drilling is referred to as *Usable-Area* and the area where it is prohibited as *Restricted-Area*. Only one of these

Parameter	Value
Average ground temperature	9 °C
Probe arrangement	2 probes of 100 m length each, 6 m distance between the probes
Borehole diameter	180 mm
Flow rate per probe	0.5 l/min (lower limit for turbulent flow in the fluid)
Probe type	Double-U, PE DN 32 PN 10
Center distance	0.07 m
Backfill thermal conductivity	1.5 W/(m*K)
Refrigerant	Monoethylene glycol 25%
Borehole resistance	According to the above construction
Simulation period	25 years
Annual performance factor	4.3 (according to the guidelines of the "Bundesamtes für Wirtschaft und Ausfuhrkontrolle", BAFA)
Lower temperature boundary condition of the fluid	1.5 °C
Ground temperature constant	Equal to the average temperature in Berlin (9 °C)
Specific extraction rate calculation	For heating without hot water preparation with 2400 full load hours of the heat pump per year (annual operating hours)
Groundwater flow	Not considered

Table 2: Ground source heat pump system simulation after the SenSBW [38]

needs to be determined in the calculation, because the other area is just the inverse, cut by the land parcel area. In the case of this thesis the *Usable-Area* is calculated directly and the *Restricted-Area* as its inverse.

For the determination of the *Usable-Area* it needs to be known which objects must be avoided and what the distance is, that must be maintained to these objects. Each federal state has its own legislation in this area, which must be taken into account. For Berlin, which is the area of interest of this thesis, all rulings for a ground source heat pump installation are given in the "Guide for geothermal energy" by the Senate Department for the environment, transport and climate protection of Berlin [39].
The leaflet covers multiple rules, but in the step of area determination only the rulings that affect the change in size of the area are regarded. The verification of whether the area can be used for geothermal purposes at all or whether it can be systematically ruled out, for example due to a water protection area, will only take place in further steps.

The first rule is the *Land-parcel-distance*. Every BHE needs to have at least a `3m` distance

to the boundary of its land parcel. If an agreement with the neighbours for a joint setup is possible, then this can also be `3m` to the combined boundary of the land parcels.
The second rule is that every BHE location must be at least `2m` away from every already existing building.
Besides that, trees must be accounted too, but no official ruling exists for their distance. For this thesis a spacing of `4m` was chosen. There are also most likely smaller objects that need to be avoided, but these three are the most important and the only ones that get accounted for in this thesis and in the application built in conjunction with this thesis. It is assumed that the land parcel, building and tree distances make up the bulk of the restriction area and create therefore a good enough approximation.

With this information given the calculation is straight forward.
Firstly the exterior ring of the land parcel polygon is extracted and then buffered by `3m`. For the buildings the multi-polygons are buffered by their respective `2m`. Then all point positions of the trees are buffered by `4m`. All three buffered areas then get subtracted from the original polygon of the land parcel.
The result of this operation is a geometry collection that can hold any geometry type. To clean this up all geometries besides polygons and also polygons smaller than $1m^2$ are omitted afterwards. The reason for this is that they are considered too small for practical realisation. The area calculation as a formula can be seen in Equation 1. All units in the formulas are given in meter.

$$\begin{aligned} UsableArea = LandParcelArea - Buffer(ExteriorRing(LandParcelArea),\ 3) \\ - Buffer(Buildings,\ 2) - Buffer(Trees,\ 4) \end{aligned} \tag{1}$$

With the *Usable-Area* given, the *Restricted-Area* can also be calculated, like shown in Equation 2.

$$RestrictiveArea = Clip(LandParcelArea, Inverse(UsableArea)) \tag{2}$$

An example calculation from a spatial viewpoint can be seen in Figure 2.
With the *Usable-Area* now determined the next step is to acquire all spatial information that intersects with the *Usable-Area* and returning them over the API. For more detailed information, the *Usable-Area* can now be used to perform borehole heat exchanger modelling. Restrictions like water protection zones are not included in the area determination. The area calculated is only based on distance regulations. This is due to the fact that an *Usable-Area* must first be defined in order to then determine whether a restricted zone exists for this very area, by acquiring the spatial intersection information.

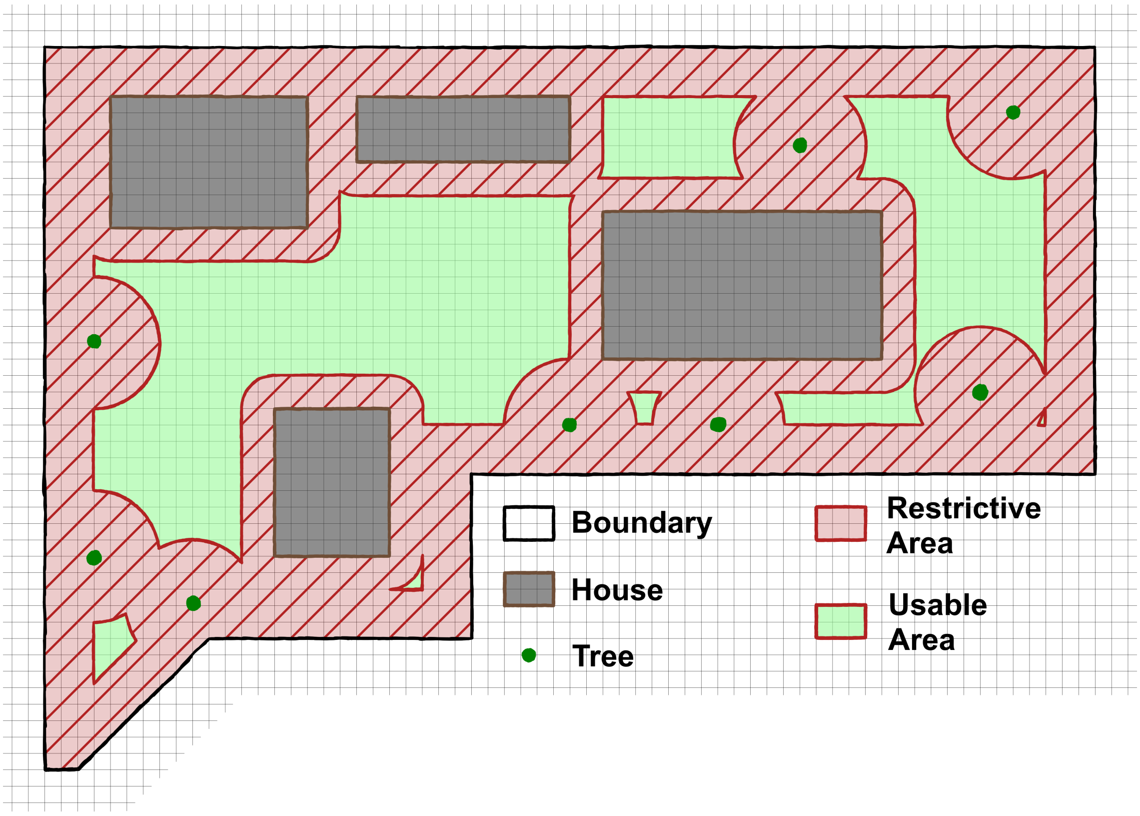


Figure 2: Visualisation of the area determination with the defined conditions.

2.4. Automatic borehole heat exchanger modelling

With the *Usable-Area* acquired from the last step, an example setup of borehole heat exchanger allocations can be simulated. For this an algorithm is needed that can model BHE positions for an arbitrary multi-polygon geometry. To do this optimised the goal of the algorithm should be to maximize the amount of BHEs that can be placed inside the *Usable-Area*.
As the area determination provides for an area that is already law confirmed, only one further rule needs to be taken into account. BHEs need to have a distance of at least `6m` to each other. Based on the spacing circle this creates, it is more efficient to place borehole heat exchangers on edges or even corners, because then most of the blocked area lies on the *Restricted-Area* and does not use up space of the *Usable-Area*.

By determining which algorithms fit the problem the best, circle packing algorithms come to mind [40]. These seem to fit, because the modelling acts like circle packing. But the spacing circles can overlap in the case of the BHE modelling which is not considered in circle packing. To find a point-based approach that distributes points in the correct manner, a Poisson Disk Sampling can be used [41], which "[...] produces points that are tightly-packed, but no closer to each other than a specified minimum distance" [42]. The practical application is known as Bridson's Algorithm [43], which is time optimised. These algorithms are universally applicable for different areas and can even be used in meshes [44]. Unfortunately, the Poisson Disk Sampling only creates a uniform distribution, but not a maximal distribution of points. This comes quite close to the wanted functionality, but does not satisfy the criteria. To achieve this, a different version must be used that was published by Mohamed S. Ebeida et al. [45], which they called *Efficient Maximal Poisson-Disk Sampling*. This Poisson-Disk Sampling on the other hand is highly efficient in computation speed and maximises point placement mathematically perfect. A working version is also available since March 2023 [46], written in $C++$.

With all these advantages the *Efficient Maximal Poisson-Disk Sampling* still does not provide a complete solution. The algorithm is designed to create a perfect configuration of points inside any polygon, but this does not work for a multi-polygon setup, which consists of multiple not connecting polygons, like shown in Figure 2. The key problem is that performing an operation on all polygons individually and aggregating the solution provides not the same result as performing the entire process for all of them at once. Reason for this is that the boundary spacing of points in one polygon can reach into the next polygon. Therefore they can all influence each other. Ignoring this problem can consequently create law violating configurations. Random samples of land parcels in the study area show that this violation is not a rare occurrence, which is why it must be addressed.
To solve this, two solutions are possible. Either to extend the *Efficient Maximal Poisson-Disk Sampling* algorithm to spatial multi-polygon data or to develop an own algorithm. Due to the nature of the already existing algorithm, an extension and integration was considered to be less realisable than creating a new one, that can be built into the application. This is due to the fact that the "Efficient Maximum Poisson Disc Sampling" algorithm has no concept of spatial relationships.

Based on this state an own algorithm was developed in this thesis. The target function of the algorithm optimizes for a borehole heat exchanger configuration in which each BHE with its spacing area covers the smallest proportion of the *Usable-Area*. This leads inevitable to a configuration with the maximum number of BHEs. At the current state it is not mathematically proven that this algorithm is optimal, but practical tests show a highly optimal positioning. The input of this algorithm is the prior calculated *Usable-Area*.

The workflow of this algorithm can be described as follows:

Algorithm - Area minimising point positioning in multi-polygon structures:

Initialization: The input multi-polygon of interest (*Usable-Area*) is defined as *Remaining-Area*.

1. Discarding of small polygons: Check all individual polygons inside the *Remaining-Area* for size of area. Discarded any polygons that cover an area of $< 1m^2$. Check in this step also if points or poly-lines were created. If yes, discard them also.

2. Checking the remaining area: Check if the set of *Remaining-Area* is empty or null. If this is the case, all BHE positions have been modelled and the algorithm can be terminated. Otherwise continue.

3. Center-point calculation: Determine the centroid of the *Remaining-Area*.

4. Extraction of the boundary ring: Extract the linear ring of the *Remaining-Area*.

5. Extraction of vertices: Extract all vertices which form the linear ring.

6. Locating starting position: Calculate all distances from the centroid to all vertices. Rank them by distance. Select all points with the largest distance. These points then make up the list of *Possible-Points*.

7. Checking possible positions: Iterate through all points in the list of *Possible-Points*.
   (a) Point buffer: Buffer the points by the defined spacing area (`6m`).
   (b) Determine area usage: Calculate the intersection area of the point buffer and the *Remaining-Area*.
   (c) Check area usage: When the point is the first point checked, or its intersection area is smaller than the *Area-Usage* of a prior point, set its intersection area as the new *Area-Usage* and mark the point as the *Best-Point*. Save also the corresponding buffer of the *Best-Point*.
   (d) Repeat: Repeat this until all points were checked.

8. Final points: Append the *Best-Point* to the list of *Final-Points*.

9. Determine new possible points: Calculate the intersection of the buffer of the *Best-Point* and the linear ring of the *Remaining-Area*. All intersections make up a new list of *Possible-Points*.

10. Update the remaining area: Subtract the area of the *Best-Point Buffer* from the *Remaining-Area*, discard again small polygons and extract a new linear ring of the *Remaining-Area*.

11. Repeat and terminate: Check if the list of *Possible-Points* contains points. When no intersection occurs go back to Step (1). Otherwise jump to step (7).

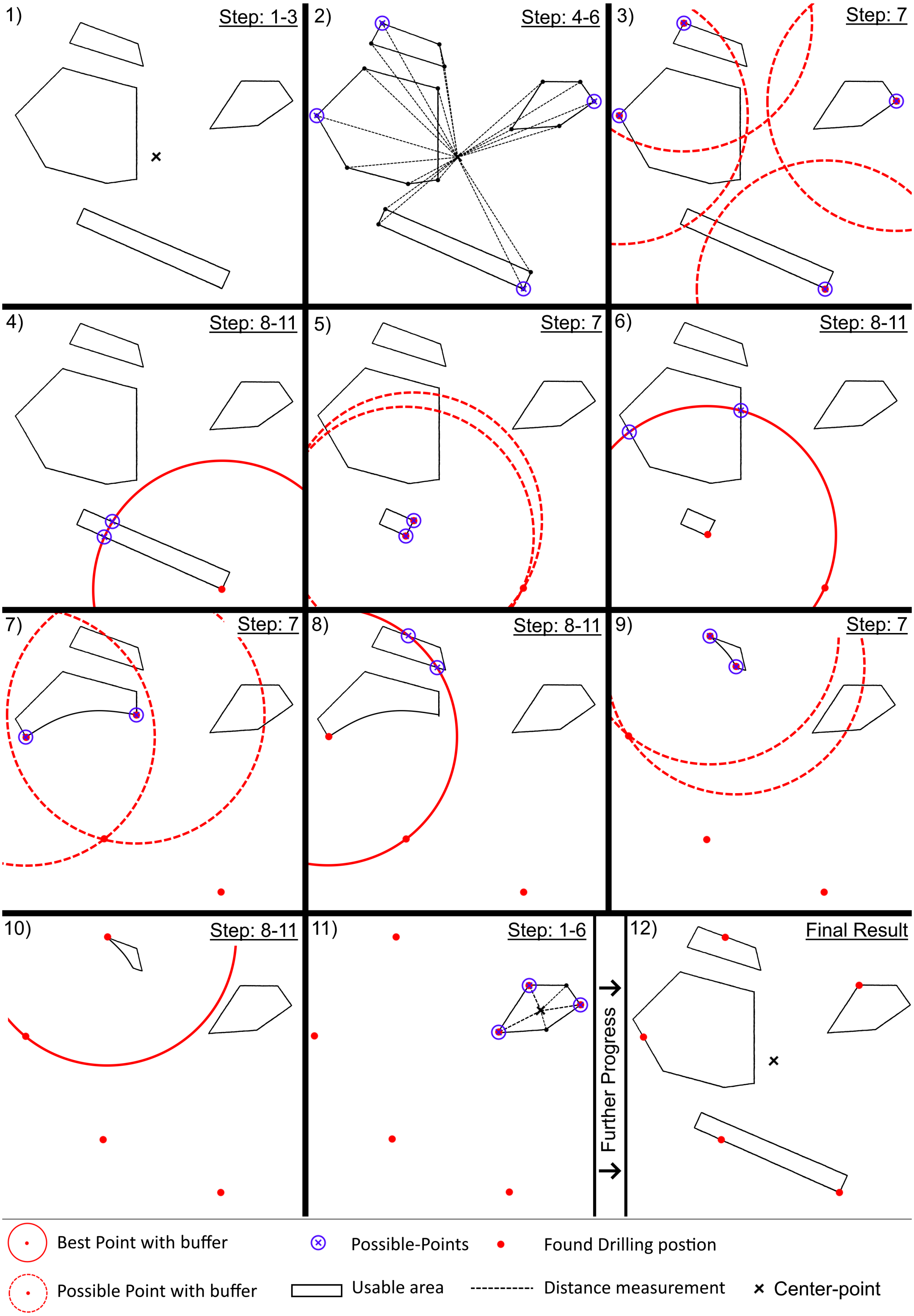


Figure 3: Stepwise visualisation of the area minimising point positioning algorithm for multi-polygon structures. The area used is the *Usable-Area* prior computed.

In Figure 3 is the described algorithm is stepwise depicted. The list of *Final-Points* generated by the execution contains the final set of positions where the borehole heat exchangers can be placed. Areas $< 1m^2$ are discarded, because they are not practically usable. Drilling in these small areas is in the most cases not feasible. But it must be mentioned that in theory an area of $< 1cm^2$ can still be used, even though the BHEs have a diameter of $< 20cm$, since these dimensions have already been taken into account during the creation of the *Usable-Area*.
The entirety of the code of the algorithm, as well as the complete code base of the entire application is, publicly available on GitHub [2].
With a possible borehole heat exchanger setup now simulated, the exact BHE locations can be used to carry out BHE-based analysis.

2.5. BHE-based analysis

With the BHE positions simulated, a more detailed analysis can be performed. In contrast to before, where the geothermal factors were determined by their intersection with the land parcel polygon, the information can now be received for every modelled BHE. This has multiple advantages.

If there are several values of a certain factor present in a polygon, the intersection of the geometry of a parameter and the land parcel polygon can result in a value range. For this reason, the ratio in which each value of the parameter is present cannot be known. Only if they occur at all. When for example a water protection zone slightly intersects with the area in question, the whole polygon gets marked as intersecting with a water protection zone. In practice, this means that the property cannot be used.
When checking the intersection with the modelled borehole heat exchanger points on the other hand, value ranges are not present, because the simulated BHEs get treated as geometric points, which have no area. This would mean that in the same example from before where the water protection zone slightly intersects the restriction area for which the BHE was modelled, only a few BHEs would show the presets of this intersection. Based on this view, it can therefore be shown that the area can still be utilised, at least in part.
In all of that it must be made clear that the difference between a polygon and a multi-point perspective is more pronounced for larger areas and less relevant for smaller areas. A borehole heat exchanger based analyses of a backyard of $20m^2$, does often not provide better information than a simple polygon intersection.

The second advantage of this sample-based analysis is, that the specific heat extraction capabilities that can be provided by a plot of land, can now be calculated. The heat extraction is given by the source data as a specific heat extraction rate W/m. With the knowledge of all the BHEs and their depth, heat extraction over a year can be calculated like shown in Equation 3.

$$HeatExtractionPerYear(d, e, h) = \sum_{i=1}^{p} d_i \cdot e_i \cdot h_i \tag{3}$$

Where:

- d = Maximal drilling depth in `meter` of all BHEs.
- e = Extraction value given as `Watt/meter` (W/m) of all BHEs.
- h = Annual operating `hours` of the heat pump.

The extraction values were estimated from simulations of the SenSBW. These values reference a setup of two double-U BHEs with a spacing of `6m`, that operates over `25` years [38]. The values used are given as heat extraction in `W/m` by `2,400` annual operating hours for the corresponding depth. `2,400` annual operating hours are assumed, as these are the average hours for heating in Germany [47].

The heat extraction values, determined by Equation 3, are therefore only correct for the reference setup of two BHEs. When more BHEs are present at the same location the heat extraction decreases because of cross influence. The influence of BHEs on each other is a highly complex function which is dependent on many factors. Because not all of them can be known with the limited data set given, a rough approximation is needed. To achieve this the orienting model calculations from the City of Zurich Building Department [48] were used to model the losses in the heat withdrawal by an increase of BHEs. Based on the simulations a logarithmic equation was modelled in Python to describe the relationship, which is given as Equation 4. A visualisation of the value distribution can be seen in Figure 15 [Appendix]. The cross influence reaches its equilibrium at 71 BHEs, where the values does not further decrease.

$$BHECrossInfluence(n) = \begin{cases} \dfrac{159.863 - 12.5247 \cdot \ln(100.761 \cdot n - 141.675)}{100} & \text{if } n < 70, \\ 0.49 & \text{if } n \geq 71 \end{cases} \tag{4}$$

Where:

- **n** = Number of BHEs.

With the cross influence given, the adjusted heat extraction can be calculated with Equation 5.

$$\begin{aligned} AdjustedHeatExtractionPerYear(d, e, h, n) = {} & HeatExtractionPerYear(d, e, h) \\ & \cdot BHECrossInfluence(n) \end{aligned} \tag{5}$$

Based on these adjusted values, the heating capacity of the modelled BHE setup can be determined.

If the heat consumption of the users were also known, the coverage of the entire system could be calculated. Without that only the modelled maximum heating capacity can be estimated. For a more economic evaluation a rating can be performed for the modelled setup.

2.6. Rating of the geothermal potential

To get a complete evaluation of the area of interest an own rating needs to be performed. Because of this a *Geothermal-Potential-Rating* was designed to get a better assessment. The target of this function is to determine how efficient and therefore economic a ground source heat pumps system for the area in question is. To do this the calculation has two major goals.

Checking if a BHE can be placed at a specific position and determining how efficient the extraction for that position is. The whole evaluation is therefore based on the modelled BHEs and is carried out for each individual BHE. To then get an overview for the complete area in question, the rating of the BHEs is summarised.

Symbol / Name	**Unit / Return**	**Description**
w_p	*Boolean*	Value indicating if the area is water protected
d	*Meter*	Denotes the max drilling depth of the BHE
λ_{g}	W/mK	Denotes the value of thermal conductivity of the underground
T	*Degrees in Celsius (°C)*	Denotes a variable of the underground temperature
$Threshold(w_p, d, T, \lambda_{\mathrm{g}})$	*Boolean*	Checks the boundary conditions. Returns 1 if all the boundaries of the conditions are met, otherwise it returns 0
$DepthFactor(d)$	$Numeric(0-10)$	Represents the drilling depth factor determined by the given conditions
$ThermalConFactor(\lambda_{\mathrm{g}})$	$Numeric(0-10)$	Represents the thermal conductivity factor determined by the given conditions
$UnderGroundTempFactor(T)$	$Numeric(0-10)$	Represents the underground temperature factor determined by the given conditions
$WeightedRating(d, T, \lambda_{\mathrm{g}})$	$Numeric(0-10)$	Creates the rating of a BHE, before checking the threshold
$Rating(w_p, d, T, \lambda_{\mathrm{g}})$	$Numeric(0-10)$	Creates the final rating of a BHE

Table 3: Values and parameters of the rating equations

The first part of the equation is to determine if a BHE fulfils all threshold values. The list of descriptions for elements used in this and all other following equations of this chapter

can be found in Table 3.
The threshold values define conditions in which it is no longer economical to drill a borehole heat exchanger or just not possible. This threshold acts in addition to the *Restricted-Area*. The *Restricted-Area* only checks for legally defined minimum distances. The threshold checking function creates an output boolean integer, which is either 1 or 0. When it returns 1 the BHE can be used. When it returns 0 the BHE cannot be drilled, or the surrounding conditions are to uneconomical to drill the BHE.
The first parameter to be checked is whether the BHE is located within a water protection zone. When this is the case, the BHE can not be drilled after current legislations of the "Grundwasserverordnung" of the land of Berlin [49].
The second value in the equation is the drilling depth (d). After the same legislation the drilling for borehole heat exchanger is limited to a depth of `100m` in the area of Berlin. But this depth can not always be reached. In the case of the presence of specific geological layers the depth can be reduced to `under 100m`. In the practical use case BHEs rarely get drilled shallower than `40m` and BHEs with a max depth smaller than `30m` are in nearly all cases not economical [47]. Therefore all BHEs with a depth smaller than `30m` get discarded.
The next parameter in question is the underground temperature. The underground temperature is the most essential parameter, because it provides the base temperature for the heat pump, which directly affects its efficiency. From practical evaluation underground temperatures of `<8°C`, were defined as too inefficient to deploy a BHE there.
The last factor is the thermal conductivity of the ground surrounding the borehole heat exchanger. This factor defines how fast heat can be transferred through the ground. Thermal conductivities of `<1.6`, which mostly are found in clay dominated deposits, were defined, after practical experience, as to low for actual usage [47].
With all these values given, the threshold can be checked by the Equation 6. When any of the values does not fulfil its conditions, the complete BHE is rendered not feasible, which is indicated by a return value of `0`.

$$Threshold(w_p, d, T, \lambda_g) = \begin{cases} 0 & \text{if } w_p = 1 \\ & \text{ or } d < 30 \\ & \text{ or } T < 8 \\ & \text{ or } \lambda_g < 1.6 \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

The second part of rating borehole heat exchangers consists of determining how efficient the BHEs are. To achieve this, every factor that can influence the BHE gets interpreted separately, before summed up with the other factors. The factors determined in the different equations always return a value between `0` and `10`. For the summation these individual ratings then get weighted to calculate the weighted rating.

The first factor is the *DepthFactor*. It rates how the total length of a BHE is assessed for the analysis. The borehole length can influence the rating of the BHE in two ways. On one side, the heat extraction generated by three `33m` BHEs is lower than the extraction of one `100m` long BHE. The reason for this is not the rise in temperature with further depth, which is not present at this depth, but the thermal influence of the BHEs on each other. On the other side is the economic factor of installing the BHE for the user. The mentioned three BHEs cost more in sum to drill than the single BHE, even though combined the depth is nearly the same. The factor describing the relation can be seen in Equation 7. A visualisation of the value distribution can be seen in Figure 12 [Appendix]. The equation is linear, because all the mentioned factors act linear. The function is capped by `100m`. BHEs deeper than `100m` get the full rating, but no further increase. The BHE depth is currently capped for the aforementioned reason that the drilling depth in Berlin is limited to `100m`.

$$DepthFactor(d) = \begin{cases} \frac{(d-30)}{7} & \text{if } 30 \leq d \leq 100 \\ 10 & \text{if } d > 100 \\ 0 & \text{if } d < 30 \end{cases} \tag{7}$$

The second factor is the *ThermalConFactor*, which is a rating of the thermal conductivity of the underground. The underground is hereby the sediment surrounding the borehole heat exchanger. The thermal conductivity is the result of different soil types and grain sizes which affect how fast the ground can transfer heat. The rating of this factor is shown in Equation 8. A visualisation of the value distribution can be seen in Figure 13 [Appendix]. The assessment curve was modelled after simulations with changing thermal conductivity with the "Earth Energy Designer"[12] in cooperation with Gasag-Solution-Plus [50]. The curve has a logarithmic nature, causing changes in the lower values to be more significant than in the higher values. The reason for this is that heat transfer is an important parameter, but it needs already existing heat to be effective. Therefore it yields diminishing returns in performance gain. On the other side, increases in the lower values in the thermal conductivity cause major performance improvements, because some thermal conductivity is always necessary to effectively transport heat through the ground.

$$ThermalConFactor(\lambda_g) = \begin{cases} 6 \cdot \ln(\lambda_g - 1.320569) + 7.65 & \text{if } 1.6 \leq \lambda_g \leq 2.8 \\ 10 & \text{if } \lambda_g > 2.8 \\ 0 & \text{if } \lambda_g < 1.6 \end{cases} \tag{8}$$

The last factor is the *UnderGroundTempFactor*. This factor rates the impact of the underground temperature acting on the efficiency of the BHE. More heat is hereby nearly always an advantage, because less energy is needed to raise the temperature of the heat pump. The rating of this factor is given by Equation 10, which is derived from the COP (Coefficient of Performance) calculation formula, seen in Equation 9 for geothermal heating [51]. For this the COP value was scaled to `0-10` for the respective numeric interval. The quality grade is the ratio between the real COP value and the maximum COP value of a lossless heat pump [51]. A visualisation of the value distribution can be seen in Figure 14 [Appendix]. Because of this, the equation has a slightly exponential nature, which causes increases in higher temperature to be more impactful than in lower. The formula was derived on the assumption that the heating temperature of the consumer household is `45°C` on average.

$$COP = \eta \cdot \frac{T_N}{T_N - T_U} \tag{9}$$

where:

- η = Quality grade ($\eta = 0.45$ for BHE [51])
- T_U = Temperature of the heat source
- T_N = Heating temperature of the consumer household

$$UnderGroundTempFactor(T) = \begin{cases} \frac{64 \cdot (T-8)}{45-T} & \text{if } 8 \leq T \leq 13, \\ 10 & \text{if } T > 13, \\ 0 & \text{if } T < 8. \end{cases} \tag{10}$$

With all the different factors given, the weighted rating can be computed with the Equation 11. To combine the various factors, it needs to be known which factor has which amount of influence on the system. These values were given by a sensitivity analysis of shallow geothermal systems [52]. For this the relative distribution of the first-order Sobol' indices of the parameters were used to determine the percentage share.
This percentages should be treated with caution as the source uses different assumptions regarding the drilling depth d, themal conductivity of the underground λ_g and the underground temperature T.

$$\text{WeightedRating}(d, \lambda_\mathrm{g}, T) = \sum_{i=1}^{3} w_i \cdot f_i \tag{11}$$

where:

$$\begin{aligned} w_1 &= 24.70\%, \quad f_1 = \text{DepthFactor}(d) \\ w_2 &= 18.20\%, \quad f_2 = \text{ThermalConFactor}(\lambda_\mathrm{g}) \\ w_3 &= 57.10\%, \quad f_3 = \text{UnderGroundTempFactor}(T) \end{aligned} \tag{12}$$

With all the ratings calculated, the final check with the threshold values can be performed. When all thresholds are fulfilled, the rating becomes the value of the weighted rating. When not, the rating becomes `0`, because at least one of the parameters is too low to make the BHE feasible. The formula for the final rating is given in Equation 13. In the practical realisation of the application built in conjunction to this thesis, the threshold values are checked first and further calculations are cancelled if the conditions are not met, in order to save computing time.

$$\text{Rating}(w_p, d, \lambda_\text{g}, T) = \begin{cases} \text{WeightedRating}(d, \lambda_\text{g}, T) & \text{if Threshold}(w_p, d, T, \lambda_\text{g}) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

All these equations calculate the rating of a single borehole heat exchanger. To get an overview of the whole area, the rating is iterated over all BHE positions that were automatically modelled by the algorithm mentioned. The list of ratings then get sent back over the API, together with the rest of the sample based analyses in the from of a report. Which rating can be interpreted as good or bad must be determined through further use of the tool and expert knowledge.

2.7. API connection

Having carried out all the calculations and modelling performed by the developed application, a communication interface is required. For this the mentioned application programming interfaces (API) are used. These allow anyone to use the application throughout the web and enables integration into other applications. The input of these APIs varies depending on the specific API, but currently all of them require a list of coordinates and a definition of the spatial reference frame they are given in. The output that is sent back to the user is on the other side standardized. For this purpose, an own data format was developed that holds all crucial information for geothermal purposes. Part of it are the calculation results as values and the spatial data, corresponding to the requested area. The format is a “report” file, which is independent of any prior data formats and defines therefore its own structure. These reports can be received over multiple APIs, but are always provided in the same standardized structure. An API that returns only a subsection of all possible calculations therefore creates a report that only holds values in the respected fields and is empty for all other values. The reason behind this standardized format is to enable an easier incorporation in applications and communication. In the current application setup of this master thesis the API request gets executed by an independent static website, which plots the spatial data on a web-map and summarizes the values calculated. With this structure the functionality of the application can be used by anyone. At present, no documentation of API communication is publicly available, which will have to be provided in the near future. More detailed information to the API endpoints and the creation of the report file can be found in chapter 3.3 Report Creation.

2.8. Scientific evaluation

With the application developed in conjunction with this thesis, a complete overview of the technical and surface near geothermal potential of Berlin can be established. By iteratively determining the potential of each modelled BHE in Berlin and aggregating the results, it is possible to estimate Berlin’s potential. The first step is to determine which land parcels can actually be used realistically.
To do this the “Utilisation of areas” data for Berlin [53] from the SenSBW was used. This data set provides a utilisation description for every area of Berlin. Since it depends on the perspective which areas are considered realistic, three different scenarios were created,

that cover varying initial conditions.

Residential scenario:
Only covers the area marked as "**Residential-Area**" as the area of interest. This parameter represents only areas containing living spaces and not service districts.

Industrial and commercial scenario:
This scenario covers areas marked as "**Residential-Area**", "**Industrial-Area**" and "**Commercial-Area**" as the area of interest. It therefore represents the currently available area.

Public area scenario:
This last scenario covers areas marked as **Residential-Area**, "**Industrial-Area**", "**Commercial-Area**", "**Path-Area**", "**Sport free time and Relaxation-Area**", "**Town square-Area**", "**Area with a special functional character**" and "**Mixed use Area**".
The "Path-Area" represents only small paths between buildings and not streets. These are mostly plain dirt paths. The "Sport free time and Relaxation-Area" includes areas like public sport areas, parks, large public areas like the "Tempelhofer-Feld" and allotments. The "area with a special functional character "represents all area owned by the state. This can be the area containing kindergartens, public schools and buildings for public authorities of all kinds. The "Mixed use Area" is an area in residential, industrial, commercial or public use that could not clearly be matched to one of these types.

These scenarios were then used to determine different geothermal potentials for the area of Berlin. Firstly all in the scenario defined utilization areas were selected and then united to one single multi-polygon, which defines the *Area of interest*. This ensures a better matching. It was then checked for each land parcel in Berlin whether it was covered by the *Area of interest*. Land parcels that only intersect the area without a complete coverage were discarded. Only the land parcels that satisfied this condition were further used. All selected polygons of the land parcels were then united to a multi-polygon, labelled as *Selected-Area*. This is done to minimise the area lost due to the land parcels boundaries. Therefore, all scenarios assume that all owners of land parcels in a sub-area cooperate. When this is not the case, every owner needs to fulfil the distance regulation explained in chapter 2.3 Area determination, of its own land parcel area. It is assumed that the amount of BHEs that can be deployed would decrease in this case, but not by much, because the BHE spacing distance still needs to be considered. The resulting areas are in the most cases district blocks, which are split apart by streets, forests and other infrastructural areas, like shown in Figure 4.

With the *Selected-Areas* now determined all polygons can iteratively be processed by the means of the *Detailed-Report*. However, as the *Selected-Areas* do not contain any information of their own and are solely relevant for the purpose of the parcel boundaries defined for the calculation, only the modelled BHEs and their assessment are used.
For all the modelled BHEs the heat extraction gets calculated by the Equation 5 for ev-

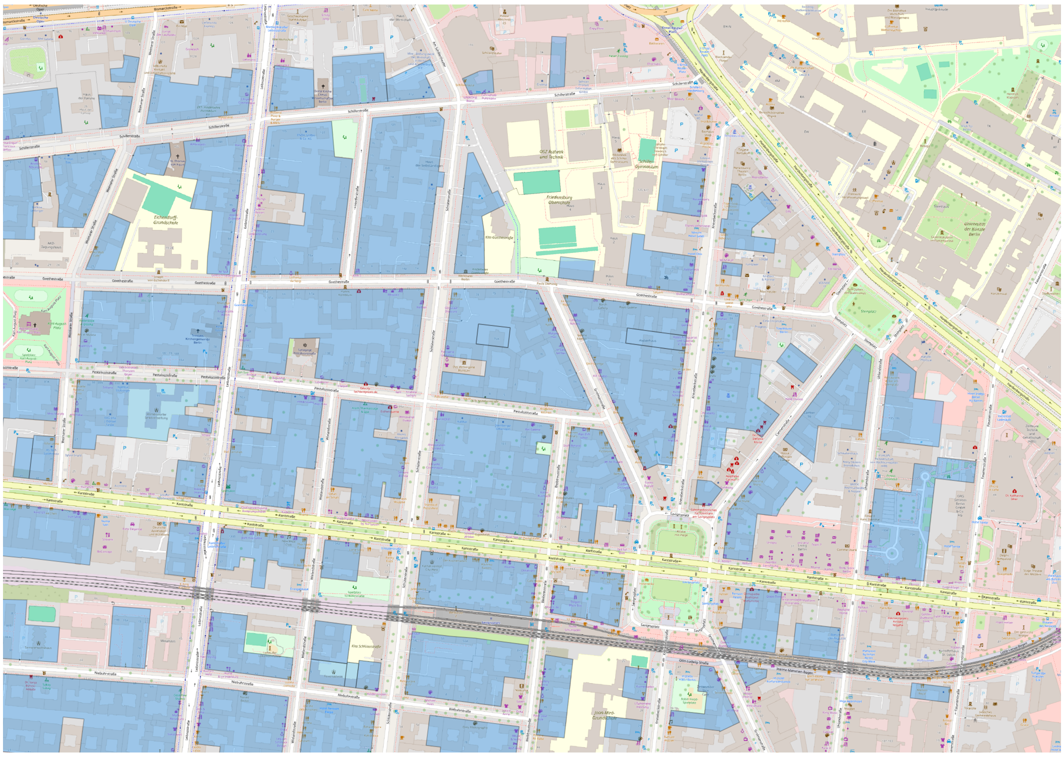


Figure 4: Visualisation of the united land parcel polygons after the utilisation defined by the *Residential-scenario*. Polygons were plotted in the QGIS [54]. The background was provided by OpenStreetMaps [55].

ery single BHE. For the various scenarios, these calculations were carried out for $\sim$3.4 to $\sim$8.1 million BHEs per scenario.

For the number of BHEs, which are needed to estimate the cross influence part in Equation 5, the median amount of BHEs in the *Selected-Areas* of all scenarios were determined. The number of BHEs are in all three scenarios $\geq$ 71, where the Equation 4 reaches its equilibrium. Therefore all heat extractions where adjusted by the maximum cross influence factor of 0.49. *Selected-Areas* with a BHE number $<$ 71 also exist, but they were scaled by the same cross influence. The reason for this is that *Selected-Areas* are quite close to each other and often just split apart by a single street. This distance should in the most cases not be large enough to completely cancel out the influence. Because all these different influences cannot be accounted for accurately, the worst case scenario is assumed for the small areas as well, which results in a more conservative estimation.

Lastly 20% of the total heat extraction is deducted. This represents an assumed approximated area loss to small obstacles, like underground pipe structures, non listed trees and other minor objects that hinder drilling. This deduction also accounts for the loss of areas that cannot be used, because they are not reachable by the drilling machinery. The heat extraction from all BHEs is then summed up to determine the potential heat that can be extracted in the Berlin area based on the scenarios.

3. Implementation

This chapter covers the most important software implementations of the application built. The application can be divided into two parts. The `Front-end`, which covers all the code that is executed on the client side and the `Back-end`, where the code is executed on the server side. Depending on the design, the main calculation of the application can be performed on one of the two sides or split between the two. In the case of this application developed in accordance with this thesis, all important calculations are done on the server side. The reason for this is that the server side was developed to host multiple application programming interfaces. The advantage of an API is that it can be used not only for the purpose for which it was originally designed, but can also easily be integrated into other applications. Based on this design choice the `Front-end` only displays the calculated results and provides a user-friendly interface to communicate with the API. No calculations and analysis are therefore performed on the `Front-end` side. Because of this, the `Front-end` application side is not covered in this thesis, but only the server side structure.

The complete code corresponding to the thesis is available on GitHub [2]. A complete implementation diagram of the implementation structure of the server is shown in Figure 16 [Appendix] and the database tables created for the application and their schematic in Table 8 [Appendix] and Table 9 [Appendix]. The implementation mostly covers the data retrieval and the report creation for the API. All formulas that are explained in the chapter 2.2 - 2.6 are also part of the application, but not explained in the following implementation part. The reason for this is that the methodology of area determination, automatic borehole heat exchanger modeling, BHE-based analysis and assessment of geothermal potential does not differ significantly from the respective implementation. All these steps were implemented based on the prior described formulas and algorithms. They are therefore not discussed here again. The chapter 2.8 Scientific evaluation is also detailed in its own chapter and needs no further explanation of its implementation.

This chapter therefore covers the fundamental structure and functions that need further explanations in their implementation.

3.1. Working Environment and Frameworks

Programming languages:

The software development on the server side of this thesis was mainly implemented in the programming language C# [56]. Reasons for choosing this language are an already known familiarity with the environment, the advantages of using an object-oriented programming language, precompilability, good documentation, fast execution and an easy deployment on Linux servers with the help of the Blazor Framework. `Front-end` development which displays the webmap and handles the API communication also required the use of HTML [57], CSS [58] and JavaScript [59].

IDE:

The integrated development environment (IDE) "Visual Studio" [60] and "Visual Studio Code" [61] were used, which are both software products of Microsoft. Because C# is also strongly supported by Microsoft and its .NET environment, these IDE's offer the most support for the programming language. Visual Studio also offers an integrated GitHub support, which simplifies keeping an up-to-date Git repository.

.NET:

The .NET framework [62] was used, which is a high-performance application platform from Microsoft, that is needed to incorporate many other tools and frameworks that are used in the application. C# was specifically developed to work with the .NET framework and the framework is needed to run "Visual Studio Code".

Database Management (pgAdmin):

As a relational database management system, the open source administration and development platform pgAdmin [63] was used. Reasons are its open source nature and the advantage that pgAdmin supports geo-data with the extension "postgis". This allows fast calculations with geo-data which is an essential condition for this project.

Entity Framework Core:

Entity Framework Core (EF Core) [64] was used to map object-related database schemes to the .NET environment. It was developed by Microsoft as an open-source lightweight tool. It allows to create a scaffold of the scheme of every table inside a database, that holds corresponding objects inside the .NET environment. These objects allow an easy interaction with the database by effectively bridging the gap between a relational database and an object-oriented application. The created objects are also secured against SQL injections.

Blazor framework:

The Blazor framework [65] was used to run C# and the .NET environment inside web applications, instead of JavaScript. It can execute .NET code directly inside the browser. This offers the advantage that the same programming language is used on both the client and the server side.

Server:

In the project the application was deployed to a server, which runs the API and website. For the server environment the Linux distribution Ubuntu 22.04 LTS is used. Within the Linux distribution, the application and the database are executed in their own Docker containers. Communication with the different containers inside the server was provided by the use of Nginx, which is a reverse proxy.

Docker:

Docker [62] was used to create isolated environments inside the Linux distribution. These environments share the same host kernel but run independent from each other. They can communicate by an internal IP and a PORT, as if they are on different machines. One container was used to run the pgAdmin on a Linux distribution and another one to run a .NET environment, which itself runs the developed application.

Leaflet:

Leaflet [66] is a JavaScript library that was used in the `Front-end` for displaying spatial data on an interactive map within the webpage. These web maps allow to display the calculated results from the server spatially and also offer an easy way to determine input coordinates from the user.

NetTopologySuite:

The NetTopologySuite [67] was used, because it provided a GIS solution inside the .NET environment. It is capable of performing spatial operations. This allows to perform spatial calculation with C# inside the application directly.

3.2. Data Insertion and Preprocessing

The first step, which is mandatory to all further analysis and implementation, is to gather and pre-process all the required data. This chapter explains therefore all the steps from data accumulation to the insertion of data into a database.

All the data that is used in this project was retrieved from the web and stored locally. Even though the data is in theory always available online, there is a huge difference in the response time between both methods, which is important for the web based real time application services built in conjunction with this thesis. For this reason a cloning of the data to a local database is needed.

This data, as seen in Table 1, exists either in the form of a WFS and WCS, which are provided over a geo-server, or are stored locally by a manual download. Data available over these two channels is provided as geo-data. Geo-data is a general description of data, which defines a data format which includes spatial information and metadata corresponding to the geometric structure. The specific format of geo-data that was used in the framework of this project is the GeoJSON format [68], which is an extension of the broadly known lightweight data-interchange format JSON.
Metadata describing the WFS, WCS, local file paths, the type of data and further additional parameters needed for the data gathering is stored in a database table, which can be seen in Table 8 [Appendix]. The reason for storing the data in a database table is that the overwhelming majority of the data consists of WFS. Therefore the URI (Uniform Resource Identifier) connection request needs to be known and stored in some kind of data structure.

With the connection information available through the given table, the data is retrieved as a serialised JSON, which is the string representation of a JSON data structure, by a GET request, which is a "Hypertext Transfer Protocol" (HTTP) method used by clients to retrieve data from a specified server resource. This JSON can then, when cashed in its entirety, be deserialized into a corresponding JSON structure of the used programming language.

The JSON elements then get extended to spatial data, which is then projected into a chosen reference system, which is followed by an insertion into a database table containing the spatial elements, which can be seen in Table 9 [Appendix]. This table also contains the related parameter information as a JSON file. From this point on all data used in the project is stored locally and can therefore be accessed quickly and without the dependence on other servers.

3.2.1. Data retrieval

Because the data used in this thesis is not inherently given, a process is needed to gather the wanted source data. Like mentioned before the data is copied from its source location, changed and locally stored. The reasons for this are plenty. The geo-data needed for the current state of the project is in most part stored in databases reachable over API services, which are managed by the SenSBW. This is not a rarity, but the common way to distribute geo-data throughout the web. When the application implemented in conjunction to this thesis sees further development and needs to access different data providers, its very likely that the data will also be provided within a geo-server structure. Therefore it is inevitable to adept to the status quo.
The mentioned geo-server API structure, which is shown as in Figure 5, is great for the accessing of data, but not in a real time setting. To improve this, the data must first be downloaded.

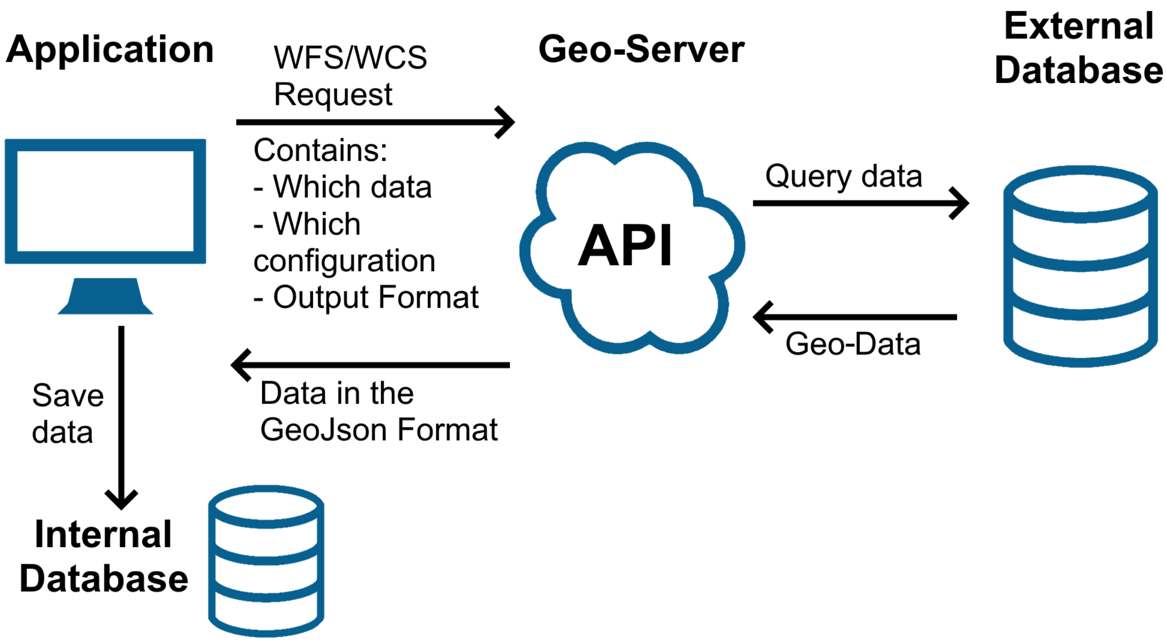


Figure 5: API request structure

The first reason the data needs to be downloaded is that even when considering that the data does not get fetched (retrieval of data from an API) in its entirety, because an on the fly interpretation is possible, in the average case, still half the data needs to be downloaded, to find a specific entry. The source files used for this project (GeoJson) contain multiple vectors that hold spatial information for the area of Berlin, which are stored regularly in over 300MB in file size. Even with a good download speed of 20 MB

per second, it takes an average of 7.5 seconds to find a single entry in a file, which is in terms of modern computing an abysmal performance. This performance then even declines further when considering that several data files need to be searched in single query.
The second reason for downloading the data is that it can be stored in a database in later steps, which increases the query speed. This is the case because both mediums have a different time complexity, which is a factor described by the Big-O notation [69]. The Big-O notation expresses the upper bound on the time complexity of an algorithm, indicating how its performance scales with the size of the input data set. This time complexity can hereby also be used to specify the search time. In the serialised JSON structure the time complexity is just linear $O(n)$, while data properly stored in a database and spatial indexed has a logarithmic complexity $O(log(n))$. This indicates that it takes, especially for larger data sets, less time to find entries in a well managed database structure than in a string file. This of course presupposes that data needs to be downloaded, inserted into the databased and indexed first, but this only needs to be done once with a local copy and is therefore faster.
The third major reason is that locally stored data enables independence from third parties. In this case, data utilisation is possible even if the server on which the source data is stored is down.
Taking all these factors into account, it was decided to download a copy of the data and process it locally, even though all the data is in theory always available online.

With this decision made, the first major task of gathering the data is the fetching of the source data over the available APIs and the automatisation of this procedure.
The heart of the process is a database table holding all important connection information, shown in short form in Table 4 or in detail in Table 8 [Appendix].

ID	**Type of Data**	**Geometry Type**	**SRID**
ID for every item	Type the data got assigned to. Describes which parameter the data holds/structure	Class of the geometry type	Number representing the spatial reference system the data is given
Last Update	**Last Ping**	**Hash**	**Set Request**
Timestamp of the last update of the data	Timestamp of the last connection to the geoserver of the source data	Hash representation of the serialised geojson	URI defining the request structure

Table 4: Geothermal Parameter Table

This table will be referenced as the *geothermal parameter* table for further explanations. Every row in this table contains all information for one API request and the corresponding metadata belonging to that request. The most important information in the table, is the URI of the GET request itself. The URI structure looks like seen in Table 5.
The GET request defines which data is wanted and in which format the data is provided. Multiple formats are often available, which reach from raster maps to vector data. One

URI part	Component
https://fbinter.stadt-berlin.de/fb/wfs/data/senstadt/s_poly_entzugspot2400_100	Address of the server, over which the geo-server can be reached.
?service=wfs	Definition of the service type which is used.
&version=2.0.0	Version of the WFS protocol being used.
&request=GetFeature	Request type (operation to be performed).
&typeNames=fis:s_poly_entzugspot 2400_100	Name of the feature type to be retrieve.
&outputFormat=application/json	Format in which the output is provided.

Table 5: Example structure of a GET request (WFS)

of this services that provides data for calculations and is mainly used in conjunction with this thesis is mentioned the web feature service, which contains vector data. In this case the specific type of data that gets requested also needs to be defined, because multiple types can be present. A request contains also an output element that defines that the data will be already provided as a GeoJSON file, which saves work when converting the data.
This means a complete communication with an API can be initiated by the GET request. All GET requests saved in the table were created manually after researching the fitting source data. This API communication can also be semi-automated with a single reference to the more broadly defined GetCapabilities services, but this was not done in this project.

The second important column in the *geothermal parameter* table is the type of data. The column holds enums (enumeration types), which represent sets of data, which holds common information. For instance APIs that provide point information of tree positions are represented by the same enum. This is crucial information for operations with the data, because every data set can have a different structure and holds therefore different attributes. This type, which also was defined manually, is utilised for two tasks.
Firstly, to identify the structure of the GeoJSON file when importing the data in order to convert it correctly into the database structure in later steps.
Secondly, it contains easily accessible information to querying the data for a specific type of entry.

The next important information in the table is the column that holds the SRID values. The SRID is a spatial reference identifier which is also known under the name of an EPSG number. These numbers represent different coordinate projection systems, that are needed for handling the geo-data. In this thesis, which only covers data from Berlin, nearly all the data is given in the coordinate reference system of "*European Terrestrial Reference System 1989 / Universal Transverse Mercator zone 33N*" [70] or UTM-33N in short. The Reason for this is that the projection system proved to be quite accurate for this part of the globe. The SRID gets read from the data while importing, followed by its injection in the *geothermal parameter* table.

With the catching of SRID values nearly all steps of the data retrieval are done. It is known where the data is stored, what its associated GET request is, which class of types it belongs to and in which projection system it is given.
But the downloading of the data, while necessary to solve a majority of issues and increase effectivity, also introduces a new problem. When copying data from the source server, only a copy of the current state is created. This means that changes in the source data do not update the local configuration. This would not be a large problem if the underlying data does not change rapidly. But official land parcel data and building footprints, which are used in this project, get updated in cycles of around two weeks. With this in mind an automatic update structure is necessary.

This process is shown in short form in Figure 6 or in detail in the upper section of Figure 16 [Appendix]. It starts with the initial call of the update database event. This event can be invoked by two possible means. The first one is a direct manual invocation, which is hereby labelled as "Admin input". This represents a call by the command line or a hidden button, not reachable for common users. This is also the current setup in the application built in conjunction with this thesis. A better configuration that should be aimed for in the future is the implementation of a function which automatically gets executed at a specific point in time. Methods with this functionality are also known as "Cron-Jobs". This function can then reliably be executed in early morning hours, when data traffic is low.

When this event is now invoked it triggers the retrieval of the data. The data fetcher, query's the GET request from the geothermal parameter table in a local database and iteratively request the data stored on the geo-servers. The files sent back from the server then get fetched by the HTTPS request and stored temporary in RAM (Random-access memory). Data retrieved this way is a GeoJSON that is stored in its serialised form, which is the string representation of the data that needs further preprocessing before it can be stored in a database.

Because the source data can be quite enormous, it would be an unnecessary operation to download and then overwrite the already saved data every time. It is way more efficient to only update the data which changed in contrast to its previous state. For this comparison hash values are used. These values can be created by a hash function, which transforms input data of arbitrary length into a fixed size string or numbers with a defined range, which ensures uniqueness and facilitates the retrieval and verification of data [71]. In contrast to the whole string of the serialised GeoJSON, a hash value can easily be stored in just 4 bytes and can be matched faster than a operation of the whole string.
The hash function is used to create a hash for a serialised GeoJSON file which was fetched. This function creates a nearly unique hash value for every input file. When the input string of the GeoJSON should just change by one character, the hash number changes significantly. Therefore, two datafiles that are completely identical result in the

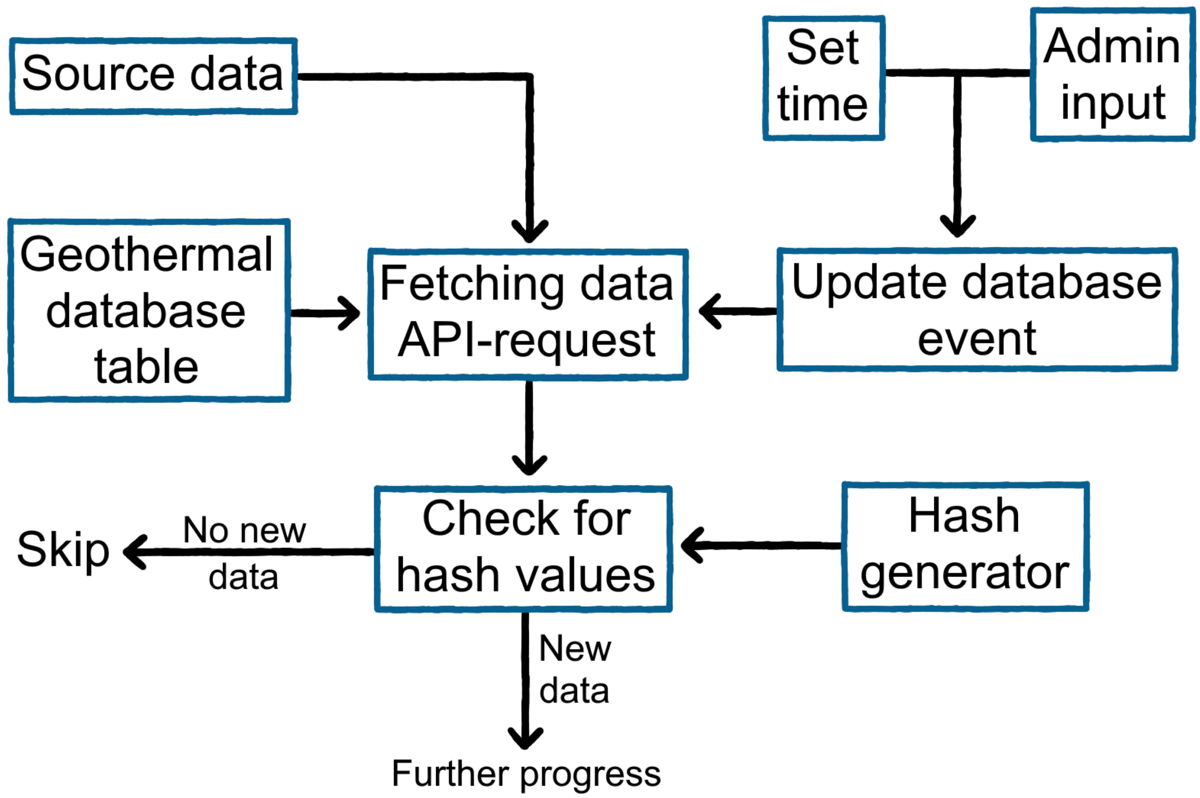


Figure 6: Automatic data retrieval and actualisation

creation of the same value.
With this principle the data gets an initial hash value when inserted into the database for the first time. When then at a later point the update event is triggered, a new hash value can be created from the retrieved data and compared with the stored hash. When they are equal, the data set is identical and therefore does not need to get updated. When this happens, the following steps get omitted and the next data set gets retrieved. But when the hash differs, the data has changed and needs to be updated.

All records of this procedure are then saved in the *geothermal parameter* table. Besides the hash value, a timestamp of the last update and last ping also get saved in the table. Last ping stands for the last time data could get fetched from an API and last update for the last time the data set was renewed locally. Both entries help with diagnostics and maintenance. When the last ping holds the current date, but the last update has an old date, the data was not updated, but the underlying API is still reachable. On the other side, when the last ping is also out of date, it can be assumed that the API itself is not reachable anymore reachable.

In theory the entire hash comparison could be neglected by just checking the GET capabilities of the WFS holding the data, which provides metadata for the request in question. But unfortunately a timestamp of the date is not part of the given metadata. Because of this, all data from APIs that are used need to be fetched first in their entirety and then be compared, because it is unclear where in the file a change may have occured. For that reason fetching the data still needs time to resolve, even when no data was updated.

With the retrieval and updating resolved, the data needs to be transferred to the local database to be used in all further processes. However, as the retrieved file is only a serialised JSON file at this stage, the data must be converted into proper geodata and its parameters before.

3.2.2. Conversion and insertion

All the data that is fetched from API services needs to be properly inserted into a local database, to ensure that it can be accessed easily and fast.
To achieve this, all information that holds the actual geometric structure and no meta-data information needs to be inserted into a second table, which is shown in short form in Table 6 or in detail in Table 9 [Appendix].
This table will be referenced as the *geo data* table for further explanations.

ID	**Parameter key**	**Geom**	**Parameter**
Id for the database	Foreign reference key from the geothermal_parameter database	Geometry vector data stored in a binary format	JSON file containing all the parameter information

Table 6: Geo data Table

The *geo data* table holds, besides its own and the foreign key of the *geothermal parameter* table, just two columns. One stores geometry information and the other parameter information associated with the geometry. Inserting data into any of these segments presents its own challenge that needs to be overcome.

The previous step of data importing left of with a GeoJSON file which was retrieved and stored in RAM temporarily. These files are at this point given in their serialised form. For all further processes these files need to be transformed into their deserialized structure. This structure is different for every programming language, because the deserialization is adapted to the used environment. But it is mostly some form of list type.
The challenging part hereby is, that for deserializing a JSON file, the target structure needs to be known. It is possible to perform a deserialization without a preknown structure, but this approach creates an unstandardised JSON element. This is why a common deserialized structure is needed.
To do this properly the geometry and its parameters are split into different configurations, deserialized independent and then later reunited in the database, like shown in Figure 7.
Both, the deserialization and insertion into the database start as a combined step, because they share the same file. But shortly after, they get preprocessed in different ways.
The deserialization of the geometry mostly focuses on creating the right geometry types paired with an indexing, while the parameter section needs to handle the standardization of different input files.

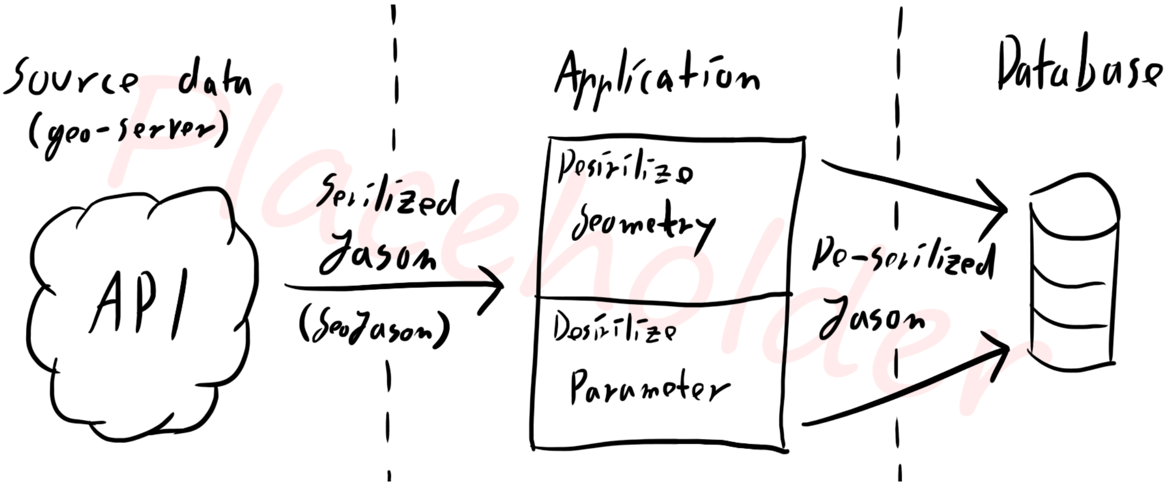


Figure 7: Placeholder: Data serialization

Geometry:

The geometry needs to be deserialized and cannot be used directly, because the geometry information needs to be preprocessed. The geometry information is in its core structure a vector of coordinates. Depending on the type and structure associated to these coordinates, the geometry they represent can be determined. Because the parameter information now gets split apart from the geometry, the geometry needs to be redefined and inserted into the database. This is also done, to enable a mixed storage of all geodata inside the database.
In an alternative way the GeoJson file could also be directly sent to the database and deserialized there with a spatial extension. This way was not chosen, because in this case a prior created table needs to exist, which is adapted to the structure of the original file. But it is desired to adept the file to a standardized structure. This could be avoided by only inserting the geometry, but even in this case the geometry must be split from the attributes, given in the GeoJSON and then being again serialised as pure coordinate information to just be deserialized inside the database again. This is also technically possible, but it does not offer a clearly better solution for inserting data. For all these reasons the geometry gets deserialised and preprocessed in the application level and then sent to the database.

Because the geometry is fetched in its source form, its coordinates can be set in any arbitrary coordinate reference system. Since a fast calculation in a real-time application should be strived for, all geometries are projected into a common spatial reference system. Without this shared projection, the data would need to be projected into a different spatial reference system in every calculation where they differ, which costs too much resources. For the target projection system was the UTM-33N chosen, because it is optimised for an area of which Berlin is a part of.
Fortunately, in the current configuration, the data is already provided in this exact reference system, which is why no projection is required. But when new data gets inserted in the future with a different reference frame, a projection is unavoidable to perform.

The second challenge is that it is not always perfectly clear to which geometry type the actual geometry belongs to. The geometry type is given as metadata included in the GeoJSON file, which can be read from there, like the geometry. Unfortunately the geometry does not always perfectly match the type description. This happens only rarely and it is not known why this is the case. Because of the mismatching information, it has proven easier to look at the coordinate structure of the data and determine the type of the geometry from there. For the current implementation the information is also stored in the *geothermal parameter* table, which can overwrite the metadata information.

With the geometry types known the data can now properly be deserialized. The information is needed because the structure in which the coordinates are given differs for every geometry type. Some contain lists of elements, others lists of lists of elements, and these elements are also different themselves. The dilemma now is that, on the one hand, a singular deserialisation structure is desired, which minimises complexity and redundancy in the implementation, but on the other hand, different structures are required for the various coordinate formats.
This problem was solved by redefining the different coordinates in the serialised JSON state and then deserializing all of them with the same procedure.

With that step the geometry information is imported in the local structure of the programming language. But to insert it into the database, the structure needs to be changed a second time to a format the database can work with. For that the NetTopologySuite framework in C# is used. This framework allows a creation of geometry data, which can be used in all sorts of proper spatial calculations.

This geometry data then gets sent to the database, based on its scaffolded structure. In the case of an update where old data is already present, which needs to be replaced, the outdated data gets deleted in the same request. To avoid loss of data and therefore keep data integrity, transactions are used. Transactions ensure that outdated data is only deleted after all new data has been successfully transmitted to the database. This is quite important, because otherwise a loss of connection can cause deletion of data, without a corresponding replacement. With this transaction the geometry part of the data gets successfully inserted into the database.

Finally, after all spatial data is inserted, all geometries are spatially indexed. This is done by a special kind of tree structure, named the R-tree, in which spatial data can be organized. Like all binary indexing methods, the R-Tree allows a fast retrieval of entries [72]. In contrast to non-indexed data, the improvement in performance through indexing is significant and considerably increases the reaction speed in a real-time application. The indexes of entries get replaced every time a portion of the data gets updated, to ensure it always covers every element.

Parameter:

The deserialization and insertion of the parameters on the other side, has its own chal-

lenges. The fundamental problem of the deserialization of the parameter information is that every instance of geo-data holds its complete own naming scheme. The fundamental problem of the deserialization of the parameter information is that every instance of geo-data holds its complete own naming scheme. This leads to the problem, that just from a data perspective it is completely unknown which attribute holds which information. For a user this is not as much of a problem. As a human operator the meaning behind measuring depths with possible names like ”100m”, ”x100”, ”ValueOnehundred”, ”MaxDepth100” seem quite obvious, but this is inconsistent naming of attributes. By using data from only one source, like in the current setup, the problem is not fatal and can be ignored initially. When in future development the point is reached where different sources need to be combined, the naming problem needs to be solved. This can occur by a possible expansion of the Brandenburg data set. A desirable solution is to define a standardized set of names an attribute can have and then create a dictionary that redirects all possible names to the standardized naming scheme.
Because of time constraints and how limited the source data set is, this problem is, like mentioned, firstly ignored. In the current setup the attributes are adapted to the structure they were given in. With that the attributes will be deserialized into the corresponding file structure.

Here arises a similar problem the spatial data also has. Different files, even when standardized, have different attributes. Data that for instance covers the groundwater temperature has an attribute named “waterTemp”. On the other side a different file that describes the drilling depth can have a parameter with the name “depth”. But the drilling depth will definitely not have an additional parameter for “waterTemp”. This leads to the problem that, if a minimal data structure is required, each file must be deserialised differently. To accomplish this, the deserialization would need to be branched into sub deserializations, which holds the fitting structure. This issue seems minor but causes a possible inflation in code complexity, because it presupposes that every type of data has its own table inside the database. To understand why this is a possible problem and how it can be solved, it is necessary to take a look at the database structure in which the data is stored.

Given through the usage of the administration and development platform pgAdmin, which is in its core a relational database management system, all data is stored in the mentioned relational database format. This data model enables very quick queries to select and calculate data, but comes with the cost of a flexible data structure.
But to overcome this problem a flexible data structure is unavoidable, which supports changes in the format. With this in consideration, the deserialisation of the attribute file was first adapted. For the target structure, one single deserialization format was defined. It covers all the parameters the different data types can have in their entirety. This means referring to the example from before, the file contains a parameter for “waterTemp” and “depth”, even though they can’t occur in the same datafile. This allows that every file, regardless of its type, can be deserialized into one common structure. The only downside is that the structure is mostly empty, and only contains data in some spots. For the project it was decided that this is the better option, because it saves many hours of redefining data structures and database tables.

But now the problem cascades to the database level. A possible solution is to create a large database table, which contains a column for every attribute that the data can hold. In this case the table has a lot of missing entries. This is an acceptable exchange that can be made for simplicity, but it does not solve the problem of the adding and changing of parameters. To solve this, a column was defined, which can hold parameters as a JSON file, like shown in short in Table 6.
This column can hold a JSON file of any structure. For the implementation, this allows the complete JSON structure to be just defined and changed inside the application and simply sent to the database to be saved there.
This column acts like a non-relational database nested inside a relational database, what provides the best of both worlds. Data can be stored in a single column and only needs to be changed in the application level. All changes in there can just be directly transferred to the database without any structural changes. PgAdmin even supports the possibility to query for entries inside the JSON file, which makes it truly act like a natural non-relational database with all features for this purpose.
With all information now gathered, the application can start with the report creation.

3.3. Report creation

The main goal of the application side of this thesis is to create a report, which provides a fast and easy overview of all the geothermal information of an area. The report is hereby not a text, but a JSON file, which holds all the information. To prepare the report, it must be checked whether an area can be used at all and, if this is possible, a summary of the local conditions will be obtained.
To access this data two different APIs exist, which are provided by the application created in conjunction with this thesis and can generate a report. How the request and return of the APIs looks like is shown in Figure 16 [Appendix].
The two different reports that exist are a *Thin-Report* and a *Detailed-Report*. The *Thin-Report* is optimised for a fast response time and the *Detailed-Report* for maximum information. All information of the *Thin-Report* is also included in the *Detailed-Report*.
The input given by the GET request to the APIs is in both cases a list of coordinates doubles and an integer referencing the SRID of these coordinates.

<u>Thin-Report:</u>

When retrieving information, in this case for planning a GSHP setup, there is always the question of the right perspective from which to view the data. Because the input data is a list of coordinates, it would be possible to return data exactly matching the position of these coordinates. This can be helpful in some cases, but in most situations land parcels are considered for the planning of ground source heat pumps. Reason for this is that the actual planning and building of these systems are bound to a land parcel or a number of land parcels from property owners.

Based on this, the first task that the application performs when it receives the GET request is to convert the specified coordinates into the correct geometry. This can be done by the usage of the NetTopologySuite framework in C#. The created points then get concatenated to a multi-point geometry and projected from their source spatial reference system, given by the SRID element, to the spatial reference system in use, which is the UTM-33N.

From there, all data in the database gets filtered for polygons which hold information of the land parcel areas by a boolean test for intersection with the created multi-point geometry from the request. All polygons that pass this check are then the area of interest. Right now, only a coherent area is acceptable, because of the current application structure. A list of coordinates that would result in a multi-polygon is therefore rejected. If this is desired, the request can simply be made for both areas separately. This whole workflow is also depicted in Figure 8.

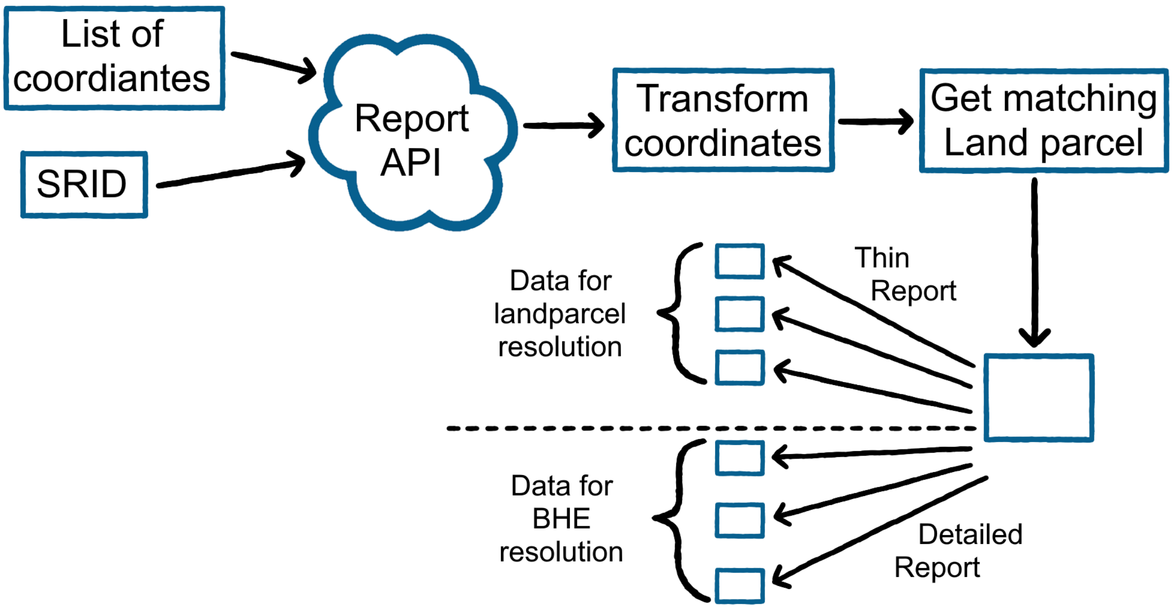


Figure 8: Report creation

With the spatial information of the area of interest retrieved, all the major calculations can be performed. In the case of the *Thin-Report*, only the polygon, built from the input coordinates, is considered as the initial geometry in these calculations.

The first part of these calculations is to determine the actual *Usable-Area*, which is covered in chapter 2.3 Area determination. With the *Usable-Area* then given, all the geometry factors that are present can be summarised. All information about intersecting polygons, nearest lines and nearest points is queried for this purpose. How this can look like in the application can be seen in Figure 9.

The major usage of this *Thin-Report* is to spot inevitable limitations like water protection areas or strong depth regulations. In these cases, it can be seen that a more detailed analysis is not needed, because it is already not possible to install a ground source heat pump in this area.

The report then gets converted into a serialised JSON and send back to the requester. The JSON is in this case a file which contains all information gathered and the geo-data of the areas in the GeoJSON format [68].

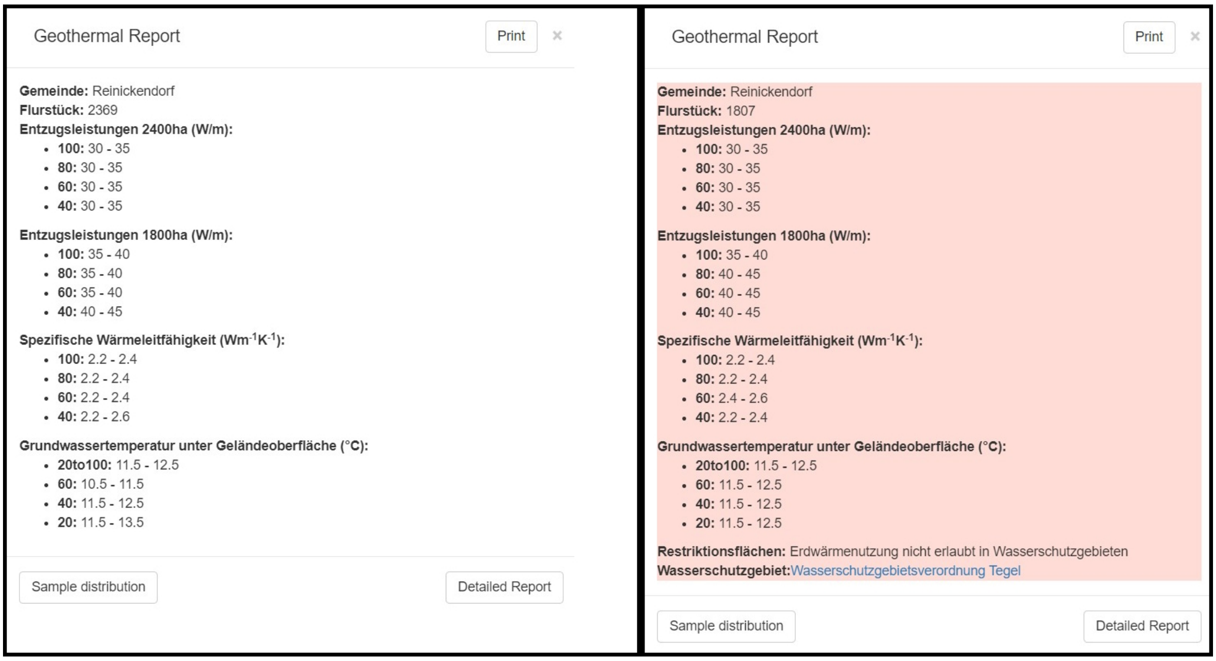


Figure 9: Thin-Report shown in the frontend application. Limiting factors like water protected areas get only shown when present. The left report covers a land parcel without clear restrictions. The right report covers a land parcel which lies in a water protected zone.

The *Thin-Report* therefore only returns the summarised information and no rating. To get a more comprehensive analysis a *Detailed-Report* is needed.

<u>Detailed-Report:</u>

The *Detailed-Report*, shown in Figure 10 in the application view, contains all the same information as the *Thin-Report* and starts also in the same manner. The only difference is that additional operations are undertaken. But in contrast to before, all this further information is not anymore related to the *Usable-Area*, but modelled BHE configurations. The modelling of the BHEs will be discussed in detail in chapter 2.4 Automatic borehole heat exchanger modelling. With the simulated positions gathered, data can then be retrieved in a borehole heat exchanger resolution. The BHE resolution enables a more versatile view, which is also more realistic on the basis of a modelled structure. How the geothermal potential is estimated in this resolution is covered in chapter 2.5 BHE-based analysis.
With all this information accessible, a rating of the geothermal potential can be done. This rating can show quickly how suitable the selected area with the modelled configuration is. This allows even laymen to estimate how good a land parcel can be used for geothermal purposes. The rating is explained in more details in chapter 2.6 Rating of the geothermal potential.
With all these calculations resolved the report gets returned by the API, as in the case of the *Thin-Report*.

These Reports build the core functionality of the application. All data is summed up in the report structure and every geometry is given in the GeoJSON format, which is

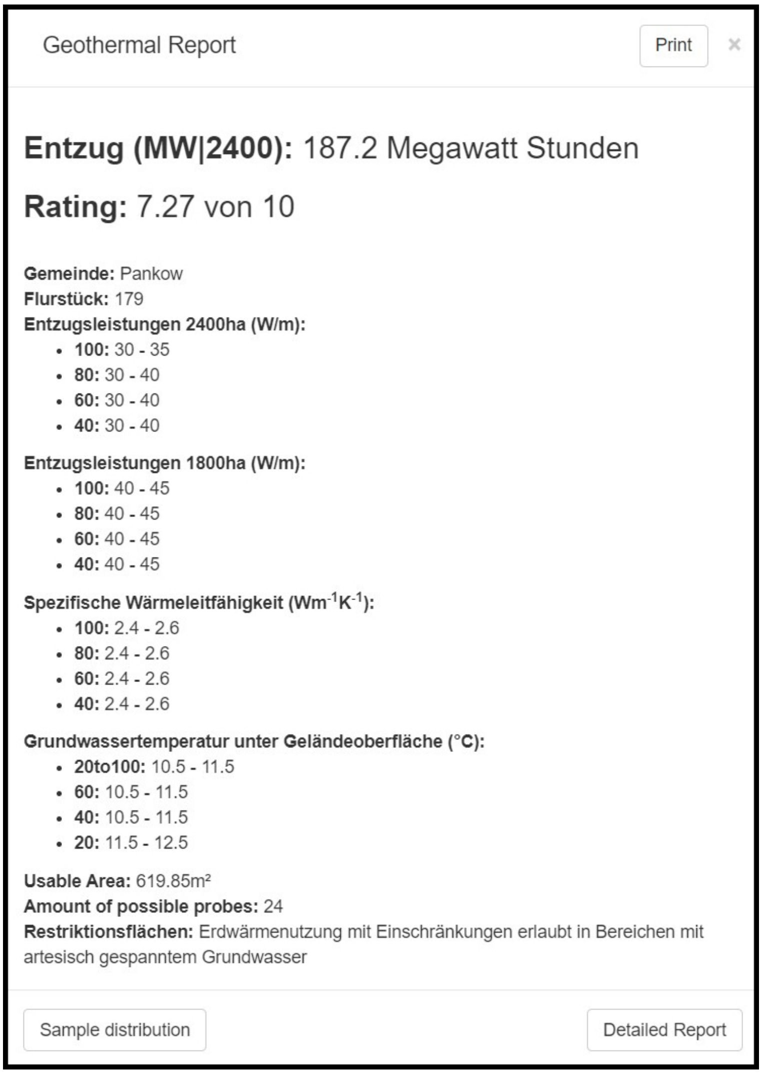


Figure 10: Detailed-Report shown in the frontend application. It includes additional information to the extraction value, the custom rating and the potions of the modelled BHE (Not shown in this graphic)

included as an own entry in the reports. The entire evaluation is thus contained in a single JSON file.
All the major calculations that are carried out in both reports have only been abbreviated here and need their own chapters to cover them in their entirety. The first one of them, which is contained in both reports, is the determination of the *Usable-Area*.

4. Results

The results are presented on the basis of the three different scenarios defined in chapter 2.8 Scientific evaluation. To estimate the coverage of Berlin's heating demand, various heating consumptions are assumed. The first of these is Berlin's heat requirements in `2020`, which the Fraunhofer Institute estimates at `36.7 terawatts (TW)` [10]. This is the latest known total heat demand of the whole area of Berlin and therefore representative for the current consumption. The second considered consumption is the heat demand of Berlin in `2035`, which was prognosticated to shrink by `4.55 TW` to `32.15 TW`, by the same study of the Fraunhofer institute [10]. Finally, the heat demand of the residential units alone was considered, which according to the Berlin-Brandenburg statistical office was 19.14 TW for 2021 [73]. All values of the results are given in Table 7 and their coverage of the heat demand of the defined scenarios is depicted in Figure 11.

	Residential scenario	**Industrial and commercial scenario**	**Public area scenario**
BHE modelled			
Total amount	3,350,967	4,774,225	8,163,872
Usable area			
Total (Km^2)	89.73	137.18	252.99
Percentage share of Berlin's area	10.07%	15.40%	28.40%
Heat extraction share of BHEs in percent			
< 30 (W/m)	4.58%	3.90%	4.38%
$30 - 35$ (W/m)	69.00%	69.57%	68.12%
> 35 (W/m)	26.42%	26.53%	27.50%
Total heat extraction in TW (Terrawatt)	8.76	12.47	21.60
Coverage by heat demand			
Berlin-2020 (36.70 TW)	23.87%	33.97%	58.86%
Berlin-2035 (32.15 TW)	27.25%	38.78%	67.19%
Housing only (19.14 TW)	45.78%	65.15%	100% *(112.87%)*

Table 7: Table of the total heat extraction and their coverage share for the given scenarios.

As can be seen in the table, all scenarios cover a different total area of Berlin. The *Usable-Area* referenced in Table 7 is hereby the already *Usable-Area* determined by the area determination. This area therefore represents the amount of space that can be utilised for borehole heat exchanger deployment for a ground source heat pumps system in accordance to the current legislations for the given scenarios. This area is also provided as the percentage share of Berlin's total area.
The heat extraction in W/m, determined by the SenSBW [38], was tracked for every BHE modelled for each scenario. These values only differ slightly between the different scenarios. They show that BHEs with an extraction of $< 30(W/m)$ account for ~3.9% - 4.6% of the modelled BHEs by 2,400 operating hours. The majority of the modelled BHEs have with ~68.1% - 69% occurrents, an average heat extraction of $30 - 35(W/m)$. High performance heat extraction values of $> 35(W/m)$ was determined for ~26.4% - 27.5% of the modelled BHEs. For a more detailed analysis, the changes in the various scenarios must be considered.

For all the different scenarios, the heat extraction values were used in combination with

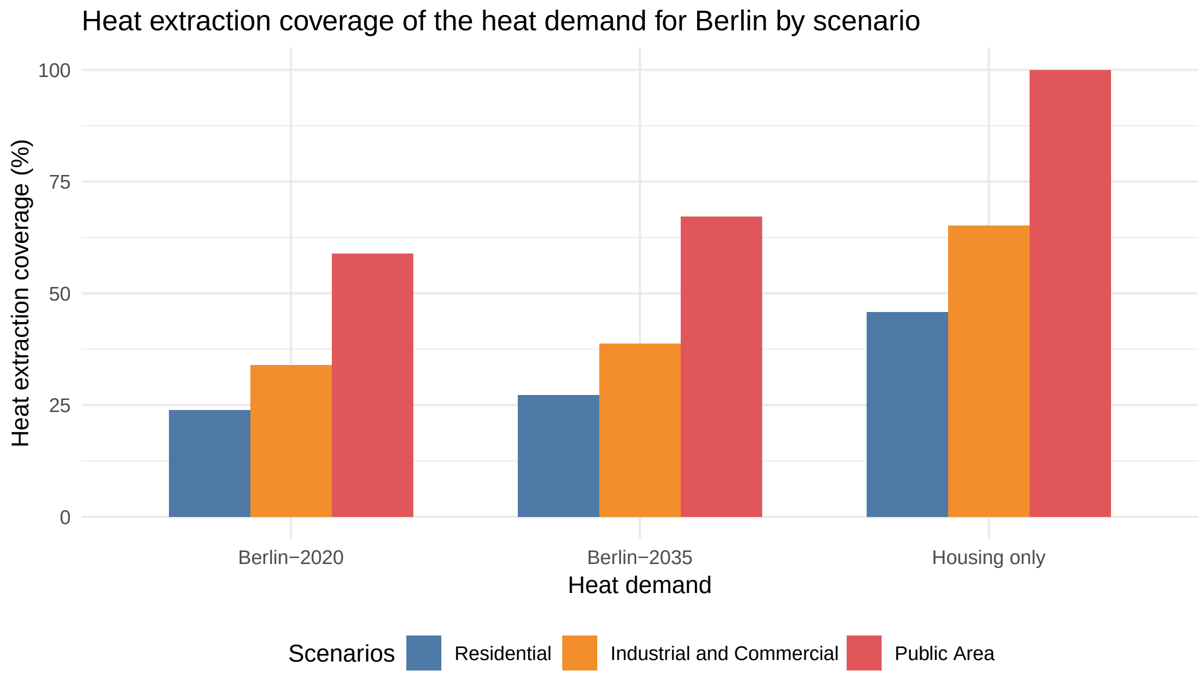


Figure 11: Total heat extraction that can be covered for every sector by the different BHE deployment scenarios. Values are given in Table 7.

the maximum depth at the site for each modelled BHE to determine the total heat extraction as the sum of all the BHEs. The calculated total heat is then compared to the different possible heat demand cases.

Residential scenario:
The area that can be used for BHE deployment and therefore GSHP setups in the *Industrial and commercial scenario* covers a region of 89.73 km^2, which is 10.07% of the area of Berlin. In this area 3,350,967 BHEs were modelled by the automatic BHE modelling. The total theoretical heat that can be extracted from the basis of the scenario is 8.76 TW. The geothermal heat energy generated under the theoretical complete utilization of the area can cover 23.87% of the current heat demand of Berlin, 27.25% of the estimated heat demand of Berlin in 2035 and 45.78% of all current domestic heating.

Industrial and commercial scenario:
The area that can be used for BHE deployment and therefore GSHP setups in the *Public area scenario* covers a region of 137.18 km^2, which is 15.40% of the area of Berlin. In this area 4,774,225 BHEs were modelled by the automatic BHE modelling. The total theoretical heat that can be extracted from the basis of the scenario is 12.47 TW. The geothermal heat energy generated under the theoretical complete utilization of the area can cover 33.97% of the current heat demand of Berlin, 38.78% of the estimated heat demand of Berlin in 2035 and 65.15% of all current domestic heating.

Public area scenario:
The area that can be used for BHE deployment and therefore GSHP setups in the *Residential-scenario* covers a region of 252.99 km^2, which is 28.40% of the area of Berlin.

In this area 8,163,872 BHEs were modelled by the automatic BHE modelling. The total theoretical heat that can be extracted from the basis of the scenario is 21.60 TW. The geothermal heat energy generated under the theoretical complete utilization of the area can cover 58.86% of the current heat demand of Berlin, 67.19% of the estimated heat demand of Berlin in 2035 and 100% of all current domestic heating.

5. Discussion

The calculation results for the entire city of Berlin show that geothermal heat pumps can cover a large part of Berlin's heating requirements. One of the most important numbers here is that with the current heat consumption, almost half of the domestic heating (45,78%) can be supplied by GSHP's, located on the residential area. As this area surrounds the living spaces, it is in close proximity to the end users and can therefore be utilised without long connection pipes that cause heat loss. This could introduce a massive reduction of primary energy consumption like natural gas for the housing sector.
Looking at Berlin as a whole, around a third (33.97%) of the total heat consumption could be covered by the use of all available area, under the current legislations. Although this is less than the residential sector, it can still cover a large portion of the heating requirements of the entire city. These two numbers are also expected to increase, as it is assumed that the energy-efficient renovation of the building sector will reduce heat consumption [10], which results in an even higher percentage coverage of the heat demand.
This is the potential under current legislations. One possible scenario for Berlin could also be a reformation of the restrictions for the use of public space for local residents. This can for example mean, that the area of a nearby park or town square can be utilised for the good of the community. For a more concrete scenario the area of the "*Tempelhofer-Feld*" could be used do deploy BHEs which cover the heat demand of the surrounding living spaces. Many of these public areas exist in Berlin, which could improve the coverage of the heat demand massively, when they could be utilised. Currently the usage of these public areas is decided for every individual case separately. A general legislation could improve their utilisation greatly. In this scenario the usage of the pavement is also missing, which could further increase the coverage. With the entire public area available, all the domestic heating currently needed in Berlin could be covered by near-surface geothermal energy alone.

All the estimated heat extraction calculated for the area of Berlin is the theoretical heat extraction. It is therefore the heat that can potentially be extracted from all the areas available, depending on the scenario. But in a real use case it is essential that the energy is produced closely to the end-user. This saves pipe infrastructure and prevents heat loss over large distances. All the estimations did not account for this factor. It is to be assumed that the location uncertainty has only a small effect for the Residential area, which is by its nature in close proximity to the end-user, and a larger influence on the public area scenario. Large amounts of heat can be extracted from these public areas, but it is not clear if enough consumers exist, that demand these quantities of energy near the location. The energy generated in this scenario should therefore be viewed critically.
All the modelling of BHEs was also performed under the minimal spacing distance of

6m. This distance represents the structure of the most coverage. In practical use cases, these distances can also be larger. 7m are regularly used and the spacing can also increase further for larger areas. The reason for this is the in most cases the cost of the whole GSHP system, which is more economically, when the BHEs are split apart further [47]. This is also the case because every property owner often only covers the heating consumption of his own property. This can lead to a situation where a property owner with a large area for the use of BHEs only installs a few BHEs to meet his heating requirements, while his neighbours cannot install BHEs because they do not have the space. In the calculations done for this thesis full cooperation between all property owners were assumed. This is in reality often not the case, because many landowners do not want to provide the underground of their land to support their neighbours heating demands. This is only the case in communes, which may act as one entity. This problem of shared space can also further decrease the total heat extraction.

On the other hand, there are factors that led to an underestimation of the total heat extraction. Firstly, the heat extraction of the SenSBW used for these calculations was based on poor initial conditions and was estimated without taking regeneration from cooling in warmer periods into account. Considering this potential would increase the heat extraction and therefore the rise in the total heat that can be extracted.
The other major factor is that no new construction of buildings was assumed. When a building is newly constructed the regulations for building distances do not apply and BHEs can be placed under the building, before the foundation is set. This hugely affects the heat that can be extracted, because more BHEs can be deployed. In 2022 alone 295,300 new living spaces were build [74]. This factor is so essential that most of the GSHP systems are currently deployed and planned for new constructions in daily operations. This is also due to new legal requirements, but it is still has a huge advantage to utilize more space for drilling. Area won through new constructions also greatly increases the total heat extraction.

6. Conclusion

After an estimation from the Fraunhofer institute 12.8% of the heating energy demand in Berlin can be provided by the use of Ground source heat pumps till 2035 [10]. Although this is a major first step, this thesis shows that the potential is far greater. It was calculated that it is theoretically possible to extract enough heat by the use of ground source heat pumps alone to cover a third of Berlin's heat demand with the right now available areas for BHE drilling. This number does not even account for new construction of buildings, which would only further increase the supply, since the space under buildings could also be utilised for BHE deployment during construction. Considering only the heat demand of the housing units in Berlin, nearly half of their heat demand (45.78%) could be covered by GSHP systems. It also can be shown that a change in legislations, which allow the use of state owned public areas, could significantly increase the heat extraction for the entire city. After this scenario it would be possible to cover the complete domestic heat demand, or over half of Berlin currents heat demand (58.68%).

Regarding these outcomes, the potential of near-surface geothermal energy should not be underestimated in consideration of the energy transition that Germany is currently facing. The development could also further be supported by a change in legislations regarding the BHE drilling depth, which could further increase the heat extraction. Right now, all BHE drilling is limited to `100m` by the "*Grundwasserverordnung*" of the land Berlin [49]. A different model that allows deeper drilling when geologically feasible is possible, like proven by the state of Brandenburg [75]. With these changes, shallow ground geothermal heat could cover the majority of Berlin's heat demand in the future. With the great potential in efficiency and cost of decentralized heating options, like the discussed ground source heat pumps, but also similar technologies such as deep geothermal heat extraction, river water heat pumps, solar thermal energy, waste water heat pumps, waste heat from data centres, industrial waste heat, Power-to-Heat and other technologies, consideration should be taken of further expanding the decentralised heating infrastructure and moving away from district heating.

On the application side, the development of software that enables a quick remote estimation of the geothermal potential for a location is only just beginning to emerge. The applications existing today do not completely fulfil the expectations that are often placed on them. All applications beside the one developed in conjunction with this thesis only return data for a specific point coordinate and do not model the BHE placement for a potential GSHP setup. Additionally, they mostly only check roughly for protected areas. Some of these challenges were overcome in this thesis, others still need to be solved in the future.

The major problem in the development of these tools is the non existing standardisation in the format of the source data. Berlin's data is mostly provided by the SenSBW over the Fis-Broker [22] platform. Data from other federal states differs greatly. This leads to increased complexity for every application that tries to estimate the geothermal potential of more than one federal state. This is only reinforced by the fact that every federal state develops its own application. A cooperation for a nation wide geothermal software could overcome redundancy and provide assistance for GSHP deployment on a national scale.

It is reasonable to assume that these applications do not only provide a benefit for planning purposes, but also for normal citizens, by enabling everyone do get a quick first initial estimation of the geothermal potential of their property. A lack of information usually prevents the decision to invest in a GSHP setup. Making this information accessible to all citizens in an easily understandable format is therefore also an essential part of the future of the expansion of renewable energies.

7. Future Development

The application developed in conjunction with this thesis already provides a good and quick assessment of the geothermal potential for an area, but can still be improved in many ways.

Essential features would be the generation of a detailed report in text form, the usage of groundwater flow rate and the consideration of regeneration. The report that is given back to the user is a JSON file that gets displayed in the frontend as a summary. It would also be desirable to create a PDF file explaining the results in text form and incorporate it into the report. The verbal description can help to clarify the impact of the values to non-experts. This text based report could be created by the use of LaTeX [76] and text building blocks. Another useful feature would be to estimate the groundwater flow rate and to incorporate its influence into the estimations. With the flow rates given, regeneration can also be estimated. This would make it possible to model different scenarios in which the user does or does not regenerate the GSHP system in warmer months. To do this, more detailed data needs to be incorporated into the database. The source data to perform these calculations is already available.

With this given, extendibility is the next important subject of future development. Even though the application is not bound to Berlin, structural changes are required to enable the integration of different areas and data sources. Firstly, an own definition of parameters and values needs to be defined, which are independent of Berlin's data structure. With that a geothermal JSON file format can be created that incorporates all relevant parameters for a geothermal estimation of an area. Their defined structure can be a superscript from the right now available data, to cover all the cases that can be relevant. With this, data from other regions could be more easily inserted into the tool and generated reports could be better integrated into other applications.

Lastly the speed of the calculation can be improved. For this the *Efficient Maximal Poisson-Disk Sampling* [45] could be used, by adapting it to the multi polygon format. This algorithm is faster and provides the mathematically proven best configuration of points. On the database side the *geo data* table can be split into multiple tables, to improve query speed. The speed is hereby improved, because empty entries are avoided by this structure. With this change the non-relational database part, represented by the JSON column inside the *geo data* table is not needed anymore. The complete theoretical geothermal JSON file format should be hereby known before defining the tables, to have a shared structure.

The application developed shows already how a web-based energy geothermal mapping can work for the area of Berlin. With the changes mentioned implemented and an extension to other federal states possible, applications like the developed application can be a great tool for shallow ground geothermal planning and scientific evaluations. In order to achieve this, a nationwide application is being strived for instead of several federal solutions.

8. References

[1] Tim Kröger. GERMAG: Geothermal Energy Resource Mapping and Analysis for Germany. https://tkroeger.com/germag/. Accessed on 2024-08-09.

[2] Tim Kröger. GERMAG-GitHub: Geothermal Energy Resource Mapping and Analysis for Germany. https://github.com/TimKroeger13/GERMAG. Accessed on 2024-06-19.

[3] MIT License Contributors. The MIT License (MIT). https://mit-license.org/. Accessed on 2024-08-09.

[4] Chijindu Ikechukwu Igwe. Geothermal energy: A review. International Journal of Engineering Research & Technology (IJERT), 10(3):655–661, 2021.

[5] Mobark M Osman and Harun Sevinc. Adaptation of climate-responsive building design strategies and resilience to climate change in the hot/arid region of khartoum, sudan. Sustainable Cities and Society, 47:101429, 2019.

[6] Abdeen Mustafa Omer. Ground-source heat pumps systems and applications. Renewable and sustainable energy reviews, 12(2):344–371, 2008.

[7] IW Johnston, GA Narsilio, and S Colls. Emerging geothermal energy technologies. KSCE Journal of Civil Engineering, 15:643–653, 2011.

[8] Ingrid Stober and Kurt Bucher. Geothermie. Springer Spektrum, Berlin, Heidelberg, 3rd edition, 2020.

[9] Holger Born, Rolf Bracke, Timm Eicker, and Michael Rath. Roadmap oberflächennahe geothermie. Unter Mitarbeit von Fraunhofer-Gesellschaft, 2022.

[10] Robert Egelkamp, Lina Wett, and Anna Cadenbach (neé Kallert). Potenzialstudie klimaneutrale waermeversorgung berlin 2035. Politische wissenschaftliche Studie, 2021.

[11] Onder Ozgener and Arif Hepbasli. Modeling and performance evaluation of ground source (geothermal) heat pump systems. Energy and Buildings, 39(1):66–75, 2007.

[12] Building Physics. Eed – earth energy designer. https://buildingphysics.com/eed-2/. Accessed on 2024-07-05.

[13] AFRY. Afry groundwater energy designer. https://afry.com/en/service/afry-groundwater-energy-designer. Accessed on 2024-08-09.

[14] DHI Group. Feflow subsurface flow and transport modelling. https://www.dhigroup.com/technologies/mikepoweredbydhi/feflow. Accessed on 2024-08-09.

[15] SLB. Petrel subsurface software. https://www.slb.com/products-and-services/delivering-digital-at-scale/software/petrel-subsurface-software/petrel. Accessed on 2024-08-09.

[16] Landesamt für Bergbau, Geologie und Rohstoffe Brandenburg (LBGR). Geothermieportal. https://geo.brandenburg.de/?page=Geothermieportal, 2024. Access date: July 16, 2024.

[17] Geologischer Dienst Nordrhein-Westfalen (GD NRW). Geothermie in NRW. https://www.geothermie.nrw.de/, 2024. Access date: July 16, 2024.

[18] Landesamt für Geologie, Rohstoffe und Bergbau Baden-Württemberg (LGRB). Informationssystem oberflächennahe Geothermie Baden-Württemberg (ISONG). https://isong.lgrb-bw.de/, 2024. Access date: July 16, 2024.

[19] Bayerisches Staatsministerium für Wirtschaft, Landesentwicklung und Energie. Geothermal energy map - energie-atlas bayern. https://www.karten.energieatlas.bayern.de/start/?c=677751,5422939&z=8&l=atkis,cdc6588d-dda7-4a69-afaa-9c365797fce6&l_o=1,0.8&t=geothermie. Accessed on 2024-08-09.

[20] Leibniz Institute for Applied Geophysics (LIAG). Geothermal Information System (GeotIS). https://www.geotis.de/geotisapp/geotis.php, 2024. Access date: July 16, 2024.

[21] Investitionsbank Berlin. Berlin fokus: Geothermie als wichtiger baustein der wärmewende. https://www.ibb.de/media/dokumente/publikationen/volkswirtschaftliche-publikationen/berlin-fokus/ausgaben-2023/berlin-fokus-geothermie.pdf. Accessed on 2024-07-09.

[22] Building Senate Department for Urban Development and Housing. Fis-broker. https://fbinter.stadt-berlin.de/fb/index.jsp. Accessed on 2024-05-21.

[23] Building Senate Department for Urban Development and Housing. Data licence germany – attribution – version 2.0. https://www.govdata.de/dl-de/by-2-0. Accessed on 2024-05-21.

[24] Building Senate Department for Urban Development and Housing. Alkis berlin landparcels. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/22e633ff-a3cc-3ac7-b599-91588f77c5c3. Accessed on 2024-05-21.

[25] Building Senate Department for Urban Development and Housing. Geothermal potential - specific extraction capacity [40-100m | 1800 - 2400 h/a]]. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/4dd3fe5f-7f09-3a2b-a325-861827d07141, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/29744740-8f97-3226-9a5b-d6ff9de4260e, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/6a8644ef-d9ce-3b8f-8763-9d89d90806e7, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/ac8fece3-7c04-39c1-b641-f712f6995686, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/

ec627696-4cac-332b-9cfc-0ad6c44dbbaf, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/8cdd851f-7ccd-3adb-86b6-6241bbb0c691, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/13f53afe-ad6f-3677-bbc8-bce26833c491, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/18e63de1-ffb1-3b1e-be87-c59a8a493bf9. Accessed on 2024-05-21.

[26] Building Senate Department for Urban Development and Housing. Geothermal potential - specific thermal conductivity [40-100m]. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/4a7bceec-0815-3a2c-9182-e8b678f3fe8c, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/6d816e71-f6dc-39be-98b7-a7ce3de996d4, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/4a2eeed2-d4e3-376a-95b8-fc647450c4c1, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/f1db97ff-10e3-3f41-a9a6-9aea6692b679. Accessed on 2024-05-21.

[27] Building Senate Department for Urban Development and Housing. Geothermal potential - restricted areas. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/342649d0-73ce-3f68-a386-420ae4182657. Accessed on 2024-05-21.

[28] Building Senate Department for Urban Development and Housing. Expected highest groundwater level (zehgw). https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/3551afd8-0e52-3048-ae6a-d0660e291f61. Accessed on 2024-05-21.

[29] Building Senate Department for Urban Development and Housing. Groundwater levels of the main aquifer 2020. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/0fba17cb-6a5a-3db1-8797-5a6b668241df. Accessed on 2024-05-21.

[30] Building Senate Department for Urban Development and Housing. Groundwater levels of the panketal aquifer. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/c1722af5-ddf1-3639-bc51-2553030c9d20. Accessed on 2024-05-21.

[31] Building Senate Department for Urban Development and Housing. Groundwater quality. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/e093b460-6948-338c-bb46-ba7308e634cf, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/24be547a-79c7-31ec-bbbf-0c860aee7aff, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/73116f3c-9c70-3d20-8403-edc7a995cd2f, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/

49aa05b8-2e72-3249-b836-fe0e0daeec3a, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/d6133e90-9340-3847-9686-36e5cc558f08, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/e1bcccd2-fd57-378c-b7ca-e2490c90671c, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/b051ea94-24ac-3078-8c83-e59e78cb189d. Accessed on 2024-05-21.

[32] Building Senate Department for Urban Development and Housing. Groundwater measuring points. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/2b515158-7d5a-32aa-a7af-bdb7c5efa5f1. Accessed on 2024-05-21.

[33] Building Senate Department for Urban Development and Housing. Water protection areas. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/7f453d7b-0454-320e-adea-ba93821af4c6, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/b89bfe25-6eaf-3a03-a593-a728c569ba46, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/1b2641f2-fd86-3814-9045-2856c9655331, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/d4cf6485-cd27-3ed0-ac0c-c536c7a8c7d3, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/b31a00a8-5461-33c0-ae74-0570c3d2ee32, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/7bdfdbed-7697-3107-96e4-03abf4f66dd3. Accessed on 2024-05-21.

[34] Building Senate Department for Urban Development and Housing. Water protection areas. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/ba6769e8-1dae-37ba-aacf-b19ee9a81e11. Accessed on 2024-05-21.

[35] Maximilian Frick, Judith Bott, Magdalena Scheck-Wenderoth, Mauro Cacace, Nasrin Haacke, and Michael Schneider. 3D geological model of Berlin - Germany. GFZ Data Services, 2020. https://doi.org/10.5880/GFZ.4.5.2020.005.

[36] Building Senate Department for Urban Development and Housing. Alkis berlin utilisation. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/0ee77a1d-0dd2-3806-8b6e-8a1c959acbf3. Accessed on 2024-05-21.

[37] Building Senate Department for Urban Development and Housing. Tree population berlin. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/0f682ebc-2b6c-3502-a233-97cac76d1762, https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/98ff39da-d9e7-3764-bbf8-e10ca6eefddf. Accessed on 2024-05-21.

[38] Building Senate Department for Urban Development and Housing. Fis-broker. https://fbinter.stadt-berlin.de/fb_daten/beschreibung/umweltatlas/klang_geo_entzug2400_100_2017.html. Accessed on 2024-07-08.

[39] Verkehr und Klimaschutz Senatsverwaltung für Umwelt. Leitfaden erwärmenutzung in berlin. Technical report, Senatsverwaltung für Umwelt, Verkehr und Klimaschutz, 2020.

[40] Charles R Collins and Kenneth Stephenson. A circle packing algorithm. Computational Geometry, 25(3):233–256, 2003.

[41] Nick Roux. Poisson disk sampling: Bridson's algorithm. https://sighack.com/post/poisson-disk-sampling-bridsons-algorithm. Accessed on 2024-06-19.

[42] Jason Davies. Poisson disc sampling. https://www.jasondavies.com/poisson-disc/. Accessed on 2024-06-19.

[43] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. SIGGRAPH sketches, 10(1):1, 2007.

[44] Bo Geng, HuiJuan Zhang, Heng Wang, and GuoPing Wang. Approximate poisson disk sampling on mesh. Science China Information Sciences, 56:1–12, 2013.

[45] Mohamed S Ebeida, Andrew A Davidson, Anjul Patney, Patrick M Knupp, Scott A Mitchell, and John D Owens. Efficient maximal poisson-disk sampling. ACM Transactions on Graphics (TOG), 30(4):1–12, 2011.

[46] SZU Advanced Technology. Efficient maximal poisson-disk sampling. https://github.com/SZU-AdvTech-2022/210-Efficient-Maximal-Poisson-disk-Sampling. Accessed on 2024-06-19.

[47] GASAG Solution Plus. Personal communication with GASAG Solution Plus. solution@gasag.de.

[48] Hochbaudepartement Stadt Zürich. Grenzabstände von erdwärmesonden, 2017. Accessed on 2024-07-09.

[49] Verkehr und Klimaschutz Senatsverwaltung für Umwelt. Schutz und benutzung des grundwassers. https://www.berlin.de/sen/uvk/umwelt/wasser-und-geologie/grundwasserbenutzung/schutz-und-benutzung/#Grundwasserschutz. Accessed on 2024-06-25.

[50] EUREF Campus. Gasag solution plus gmbh. https://euref.de/en/entry/gasag-solution-plus-gmbh-2/. Accessed on 2024-06-26.

[51] Martin Zogg. Waermepumpen. Geothermie - die Energie des 21. Jahrhunderts. Zertifikatslehrgang ETH, 2009.

[52] Simon Richter, Katrin Lubashevsky, Jakob Randow, Steve Henker, Jörg Buchwald, and Anke Bucher. Global sensitivity analysis and uncertainty quantification for design parameters of shallow geothermal systems. Geothermal Energy, 12(1):8, 2024.

[53] Building Senate Department for Urban Development and Housing. Utilisation of areas. https://gdi.berlin.de/geonetwork/srv/ger/catalog.search#/metadata/0ee77a1d-0dd2-3806-8b6e-8a1c959acbf3. Accessed on 2024-07-06.

[54] QGIS Development Team. Qgis geographic information system. https://qgis.org. Accessed on 2024-07-06.

[55] OpenStreetMap contributors. Openstreetmap. https://www.openstreetmap.org. Accessed: 2024-07-06.

[56] Microsoft. C# Documentation. https://learn.microsoft.com/en-us/dotnet/csharp/, 2024. Accessed: July 17, 2024.

[57] Mozilla. HyperText Markup Language. https://developer.mozilla.org/en-US/docs/Web/HTML, 2024. Accessed: July 17, 2024.

[58] Mozilla. Cascading Style Sheets. https://developer.mozilla.org/en-US/docs/Web/CSS, 2024. Accessed: July 17, 2024.

[59] Mozilla. JavaScript. https://developer.mozilla.org/en-US/docs/Web/JavaScript, 2024. Accessed: July 17, 2024.

[60] Microsoft. Visual Studio. https://visualstudio.microsoft.com/, 2024. Accessed: July 17, 2024.

[61] Microsoft. Visual Studio Code. https://code.visualstudio.com/, 2024. Accessed: July 17, 2024.

[62] Microsoft. .NET. https://dotnet.microsoft.com/en-us/, 2024. Accessed: July 17, 2024.

[63] The pgAdmin Development Team. pgAdmin. https://www.pgadmin.org/, 2024. Accessed: July 17, 2024.

[64] Microsoft. Entity Framework Core. https://learn.microsoft.com/en-us/ef/core/, 2024. Accessed: July 17, 2024.

[65] Microsoft. Blazor. https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor, 2024. Accessed: July 17, 2024.

[66] Volodymyr Agafonkin. Leaflet. https://leafletjs.com/examples.html, 2024. Accessed: July 17, 2024.

[67] FObermaier. NetTopologySuite. https://github.com/NetTopologySuite/NetTopologySuite, 2024. Accessed: July 17, 2024.

[68] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Stefan Hagen, and Tim Schaub. The geojson format. Technical report, Internet Engineering Task Force, 2016.

[69] P Danziger. Big o notation. Source internet: http://www. scs. ryerson. ca/mth110/Handouts/PD/bigO. pdf, Retrieve: April, 1(1):6, 2010.

[70] EPSG Geodetic Parameter Dataset. Etrs89 / utm zone 33n. https://epsg.io/25833. Accessed on 2024-06-10.

[71] Ivan Bjerre Damgård. A design principle for hash functions. In Conference on the Theory and Application of Cryptology, pages 416–427. Springer, 1989.

[72] Lakshmi Balasubramanian and M Sugumaran. A state-of-art in r-tree variants for spatial indexing. International Journal of Computer Applications, 42(20):35–41, 2012.

[73] Amt für Statistik Berlin-Brandenburg. Energie- und co2-bilanz in berlin 2021. https://www.statistik-berlin-brandenburg.de, 2023. Statistical Report E IV 4 - j / 21. Publication frequency: annually. Published in November 2023.

[74] Statistisches Bundesamt. Statistics on building completions. https://www.destatis.de/DE/Presse/Pressemitteilungen/2023/05/PD23_199_31121.html, 2023. Press release, published on 23 May 2023.

[75] Landesamt für Bergbau, Geologie und Rohstoffe Brandenburg (LBGR). Merkblatt Geothermie: Geologie und Genehmigungen. https://lbgr.brandenburg.de/sixcms/media.php/9/MerkblattGeothermie_GeologieGenehmigungen_Aug2023.pdf, 2023. Publication date: August 2023.

[76] LaTeX Project. The LaTeX Project Website. https://www.latex-project.org/, 2024. Accessed: July 15, 2024.

9. Acknowledgments

Your name could appear here

10. Appendix

10.1. Rating equation plots

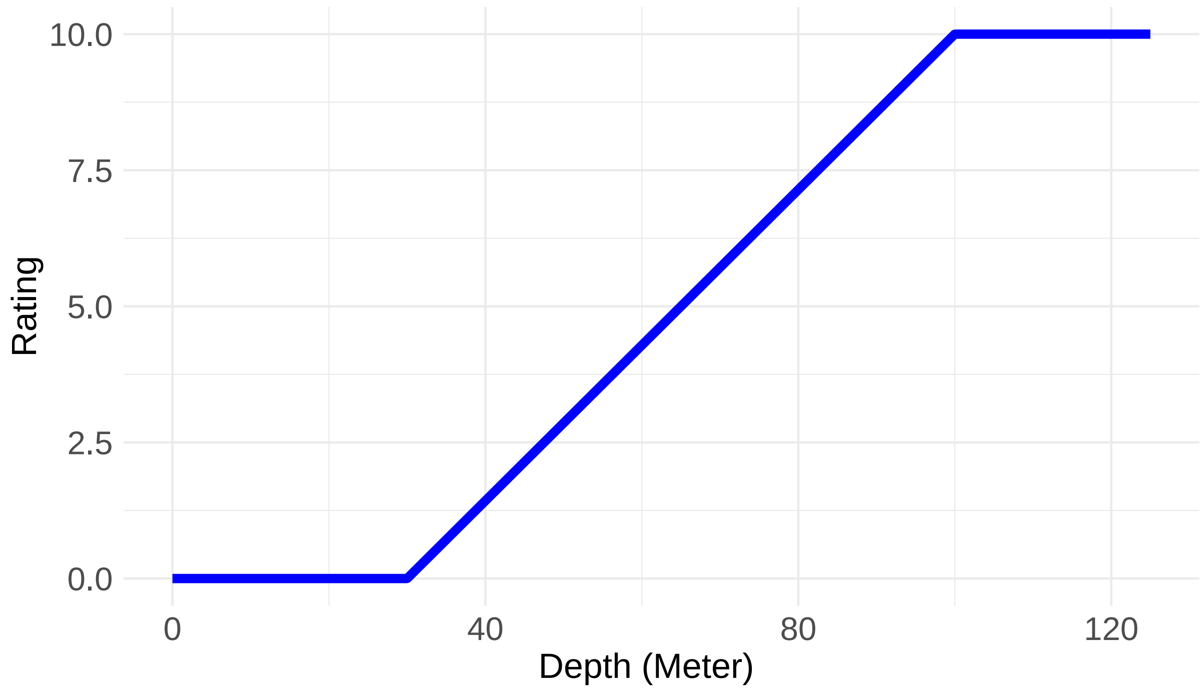


Figure 12: *DepthFactor* value distribution: Visualisation of the rating of different input values based on Equation 7. Depth is given in meter.

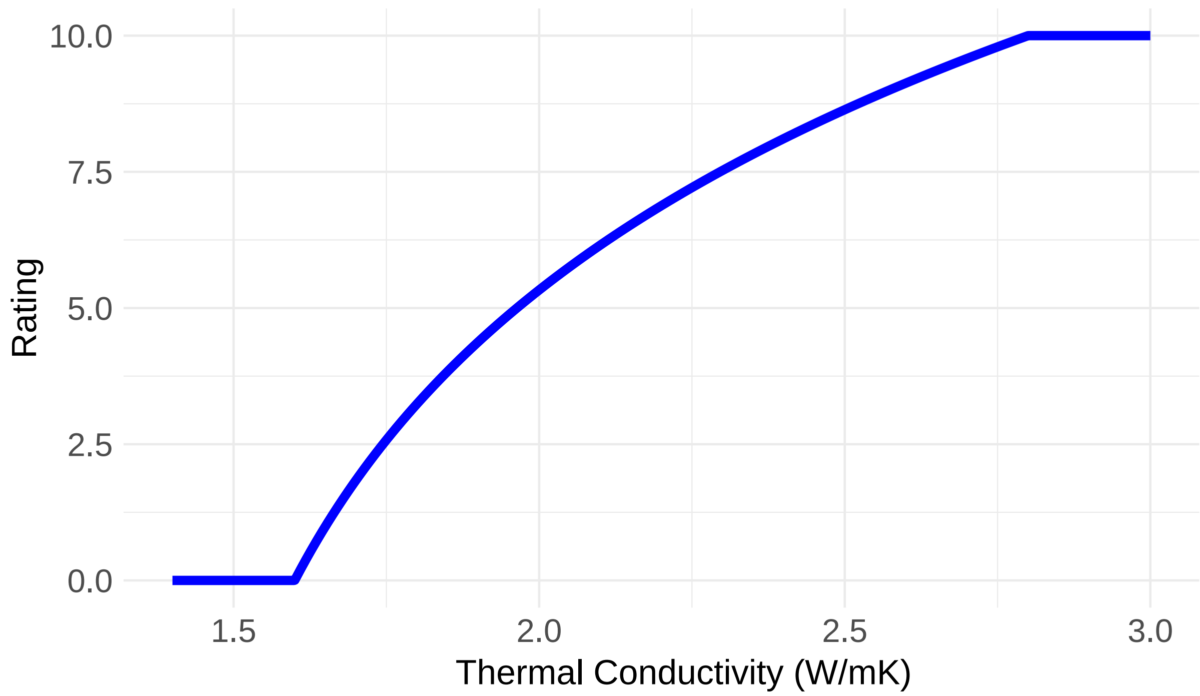


Figure 13: *ThermalConFactor* value distribution: Visualisation of the rating of different input values based on Equation 8. Thermal conductivity is given in W/mK.

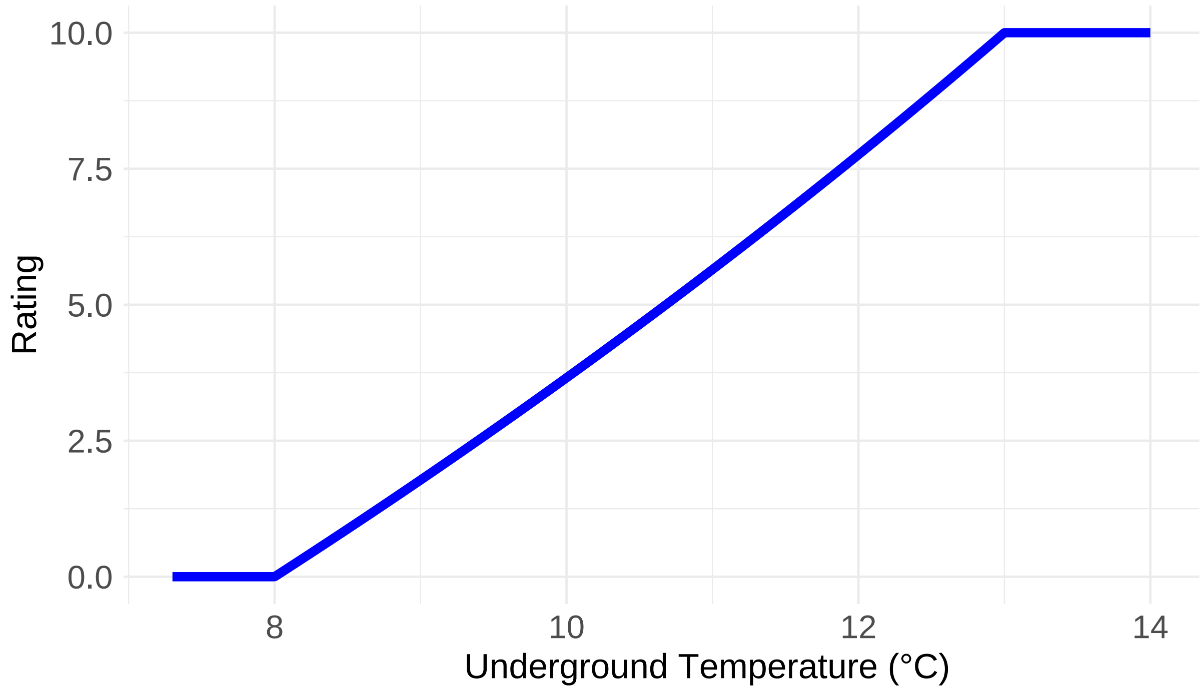


Figure 14: *UnderGroundTempFactor* value distribution: Visualisation of the rating of different input values based on Equation 10. Temperature is given in Degrees in Celsius (°C).

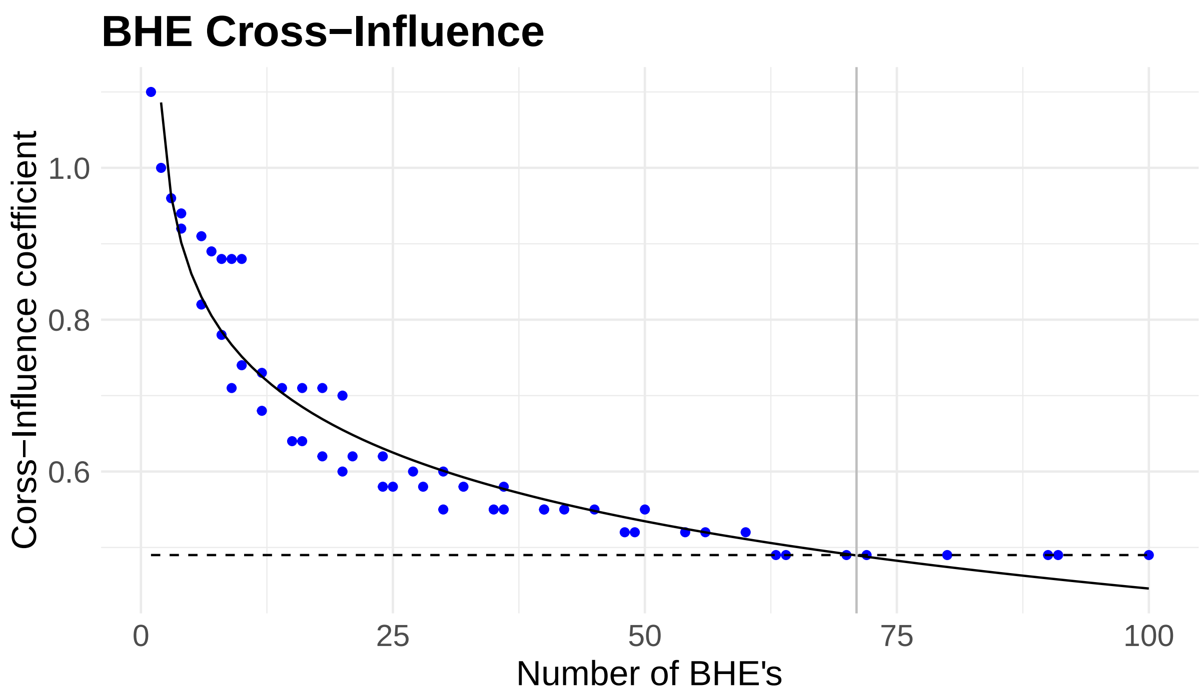


Figure 15: Cross influence value distribution. Visualisation of Equation 4. The cross influence after `71` BHEs reaches its equilibrium at a value of `0.49`. The Equation is based on modulations of the City of Zurich, Building Department [48].

10.2. Database Tables

Presentation structure of the rows of the following tables:

1. Column Name
2. Data Type
3. Example data entry

Table 8: Database Table - geothermal_parameter

id	typeofdata	area	range
Primary Key [Integer]	Enum [land_parcels, geo_poten_restrict, main_water_lines, groundwater_surface_distance, ground_water_height_main, ground_water_height_pankow, water_ammonium, water_bor, water_chlor, water_kalium, water_sulfat, water_ortho_phosphat, electrical_con, mean_water_temp_20to100, mean_water_temp_20, mean_water_temp_40, mean_water_temp_60, water_protec_areas, expe_max_groundwater_hight, geo_poten_100m_with_2400ha, geo_poten_100m_with_1800ha, geo_poten_80m_with_2400ha, geo_poten_80m_with_1800ha, geo_poten_60m_with_2400ha, geo_poten_60m_with_1800ha, geo_poten_40m_with_2400ha, geo_poten_40m_with_1800ha, thermal_con_40, thermal_con_60, thermal_con_80, thermal_con_100, groundwater_measuring_points, building_surfaces, depth_restrictions, area_usage, tree_points	Enum [berlin]	Enum [near_range, far_range]
1	*GeoPoten_100m_2400ha*	*Berlin*	*NearRange*

geometry_type	getrequest	service	srid	last_update	last_ping	hash
Enum [point, polygon, polyline, raster, multipolygon]	String	Enum [restrictive, efficiency]	Integer	timestamp	timestamp	Bigint
multipolygon	`https://fbinter.stadt-berlin.de/fb/wfs/data/...`	*Restrictive*	*25833*	*2023-03-09 04:01:12 GMT*	*2024-15-08 04:01:02 GMT*	*-215*

Table 9: Database Table - geo_data

id	parameter_key	geom	parameter
Primary Key [Integer]	Foreign Key [Integer]	geom	json
9	*2*	*88423D0023..*	*Eigentuemer: Land_Berlin, Xcoord: 402619.11, Ycoord: 5812111.53,Nh4: 0E-11, Lf: 1101.33, Cl: 0E-11, ...*

Geothermal_parameter - Table:

Holds the Metadata and API connnection information.

- id
  A unique identifier
- typeofdata
  Holds the information to which substructure the input data belongs. Different types are associated which different information that a specific entry can hold.
- area
  Name of the area in which the data was collected. For the instance of this thesis, this is only Berlin, but can be extended in the future. In future cases it can be used to shorten the search by filtering data for requested location initially over the area parameter, before filtering geometrically for the explicit geometry.
- range
  Variable to differentiate between surface near geothermal energy up to 100m and geothermal energy deeper than 100m. For this thesis this is also limited to surface near geothermal energy, but can also be extended in the future.
- geometry_type
  The definition of the geometric primitives of the reference source data. This information should be also given in the requested data of the GeoJSON's, but is sometimes incorrect in the case of polygons and multi-polygons, for which a manual correction is right now still needed.
- getrequest
  A string of the get-request of the WFS or WCS of the source data (API connection). The get request string already contains the full request to query the data as a GeoJSON from the given geo-servers.
- service
  Variable to indicate for which calculations category the data is utilized.
- srid
  Id of the Spatial reference system of the source data. When added data to the database it will be projected to the reference system 25833 and stored accordingly to this projection. This helps saving time in the calculation and unifies the geometry information.
- last update
  Timestamp of the last modification of this object.
- Last ping
  Timestamp of the last suspenseful interaction with the WFS or WCS.

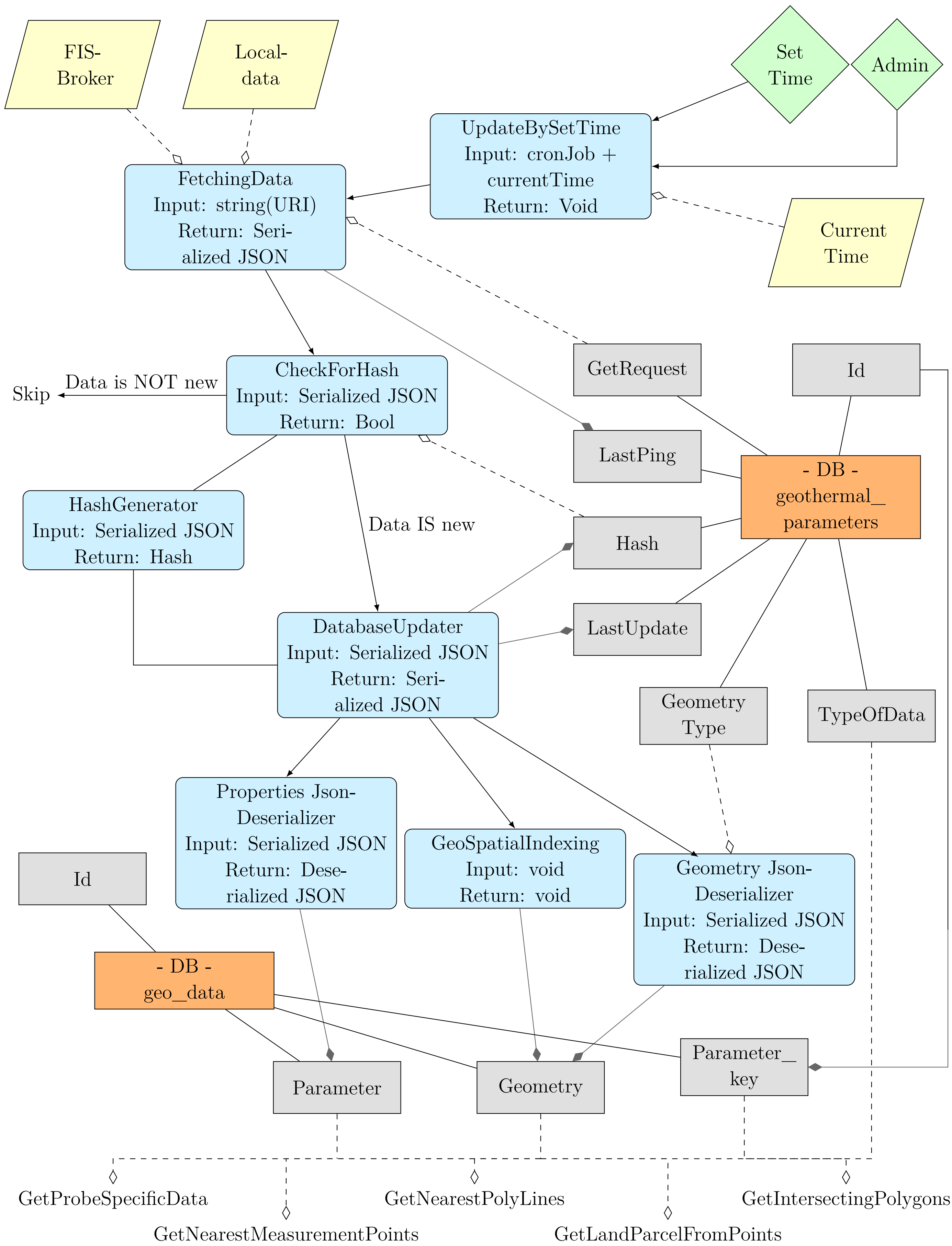
Geo_data - Table:

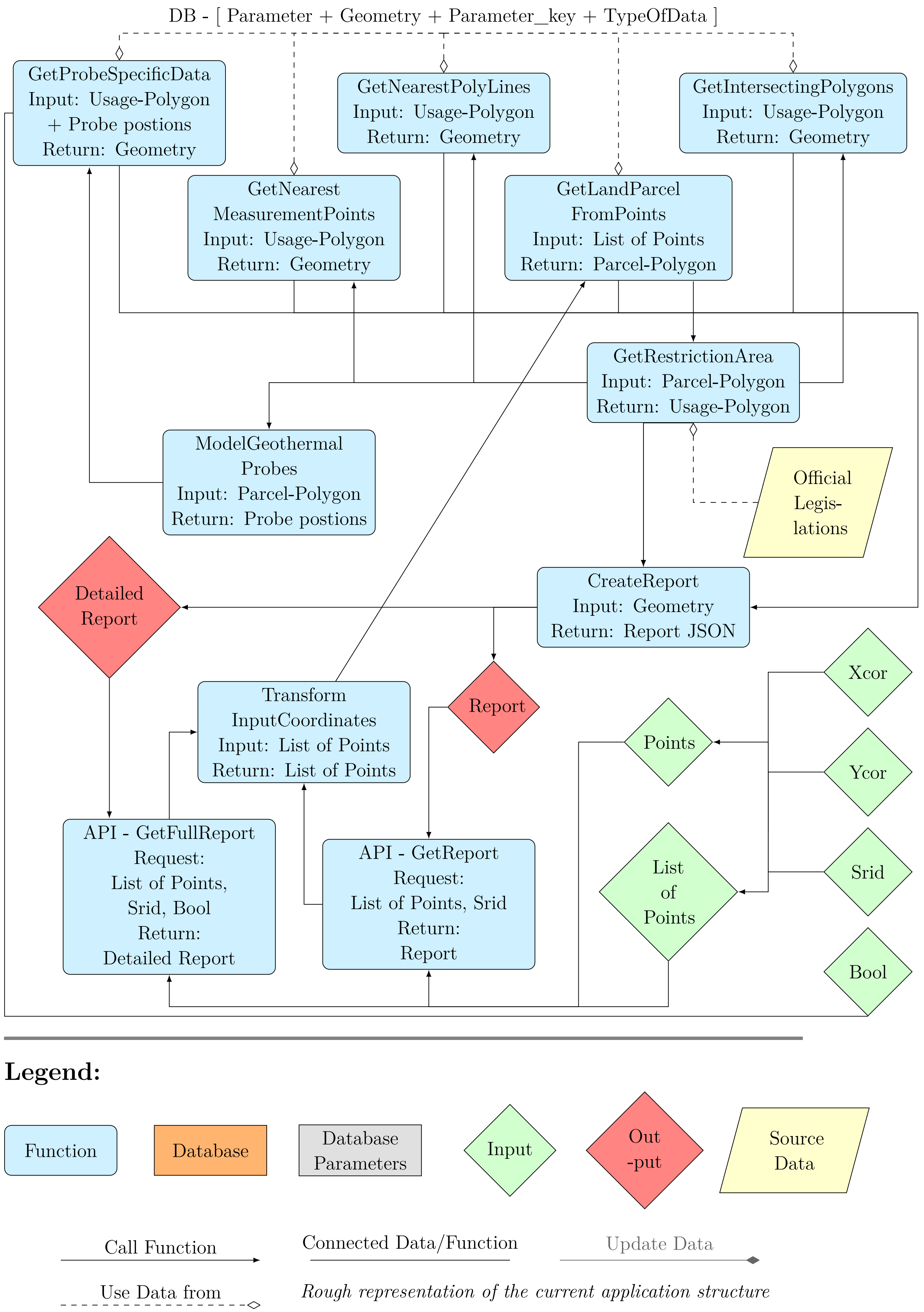
Holds the individual geometry and corresponding attributes values, which are associated with the geometry.

- id
  A unique identifier
- parameter_key
  The foreign key corresponding to the primary key of the geothermal_parameter table.
- geom
  Local saved copy of the geometry in pgAdmin geo-data format.
- parameter
  JSON which holds all to the geometry corresponding properties. The JSON contains all entries that are possible. Attributes that are not given in the structure of an entry are present as null values.

10.3. Implementation diagram - Server

Figure 16: Structure diagram - Application layout server





Rough representation of the current application structure