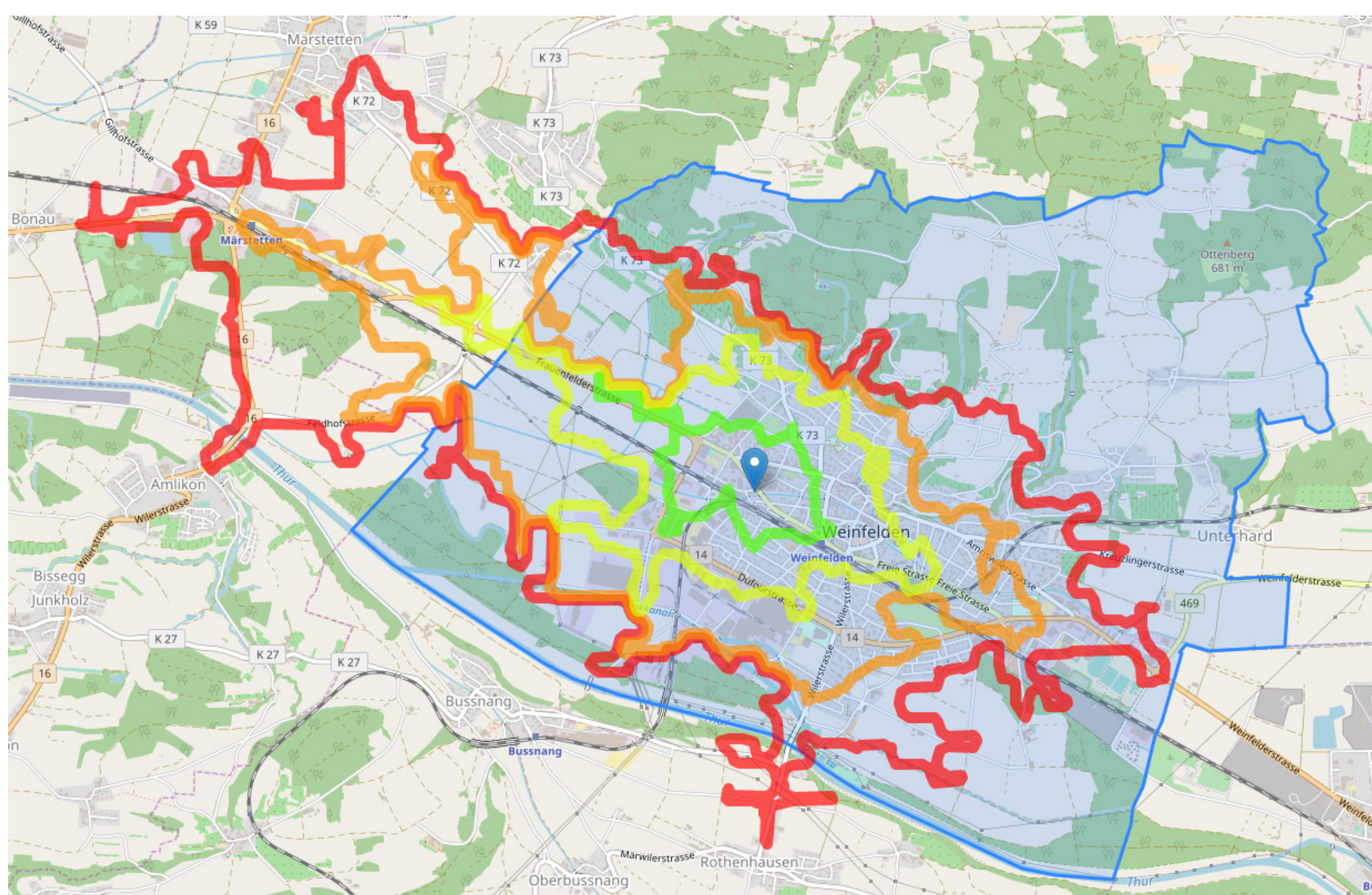


Accessibility analysis for emergency service vehicles

In many countries there are guidelines that define the maximum time in which emergency services have to arrive at the scene. This time is called response time. It consists of several phases. One of these phases, the travel time, defines the duration that is scheduled for driving from the fire or rescue station to the scene. With the help of accessibility analyses it is possible to compute the area that is reachable within the response time. A frontend was developed to allow the user a simple control of the execution of accessibility analyses.



Accessibility analysis for Weinfelden (Switzerland)

To perform an accessibility analysis the user only has to choose the starting point (for example a fire or rescue station). In this case the analysis is performed by using the default settings. If necessary the maximum travel time, the vehicle and other settings can be easily customized. For each minute a separate isochrone is computed and then the isochrones are drawn in a color gradient from green to red.

It is possible to store the result of an accessibility analysis in a GeoJSON file as well as to import GeoJSON files and display them. The GeoJSON files of several accessibility analyses can be merged together. This is for instance helpful if a city has more than one fire or rescue station.

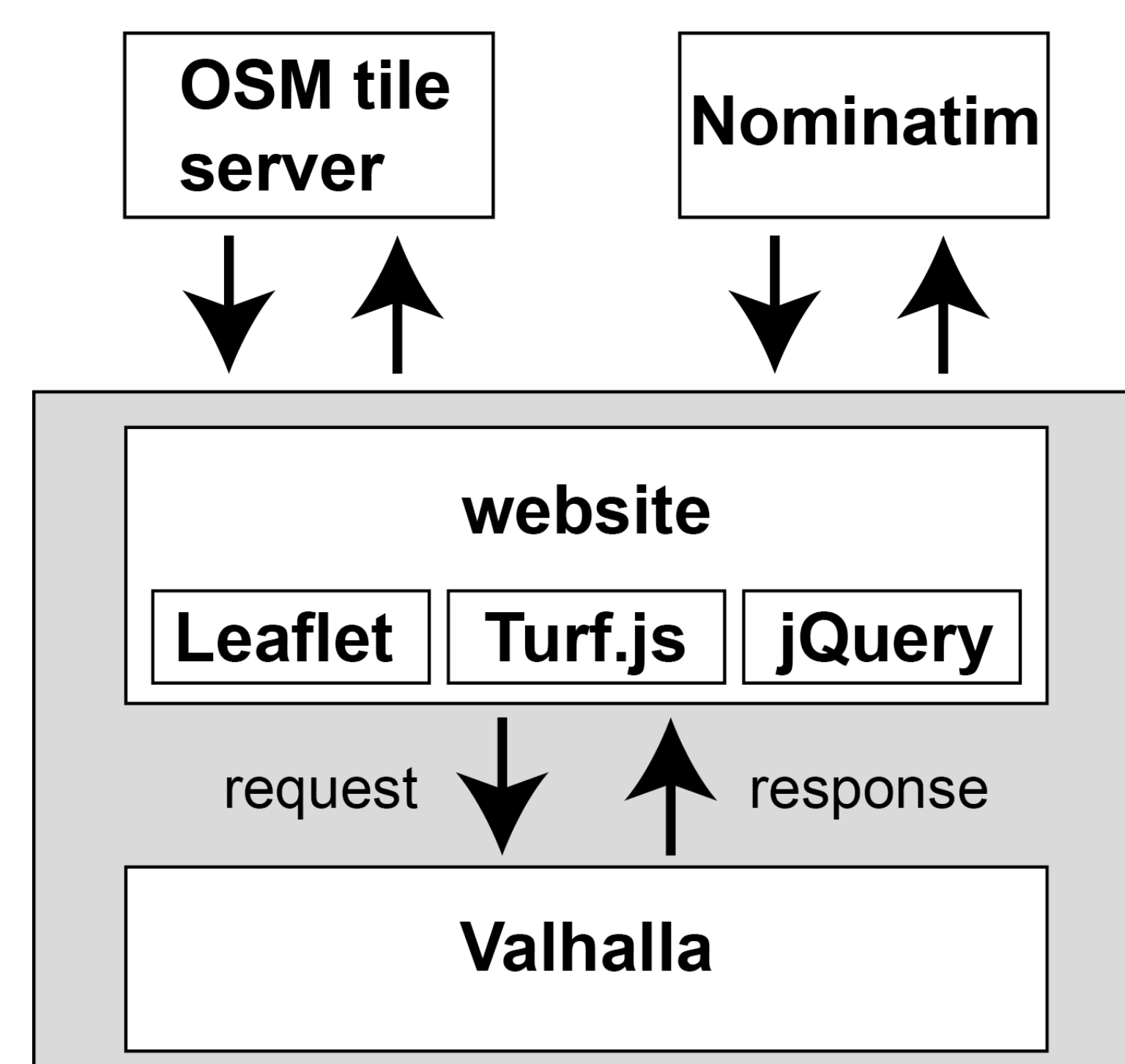
By adding the border of the alarm response area it can

easily be seen whether the complete alarm response area is reachable within the required travel time or not. Alarm response areas almost always correspond with the borders of districts, cities, or quarters.

Only open source software (Valhalla, Nominatim), libraries (Leaflet, Turf.js, jQuery) and data (OpenStreetMap) are used to perform the analysis, because they can be adapted to the personal needs and are free to use.

Core of the project is a website that can be used to control all functionalities. Leaflet is used to display the map tiles of the OpenStreetMap tile server as well as the computed isochrones, the markers for the starting locations, and the borders at the website. Turf.js is needed to merge the GeoJSON files and jQuery allows to simplify and reduce the code.

Accessibility analyses are performed by a local installation of the routing engine Valhalla and the resulting isochrones are displayed on a base map. The OpenStreetMap tile server provides this base map. The Nominatim service enables the geocoding functionality and the search for the borders of cities.



The architecture of the project

The code of the website can be found at:
<https://github.com/BFutterer/Masterthesis>.