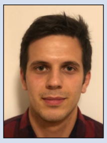

Labian Gashi

Student	Labian Gashi
Examiner	Prof. Stefan F. Keller
Subject Area	Software and Systems

A Minimal Feature and Tile Server written in Python

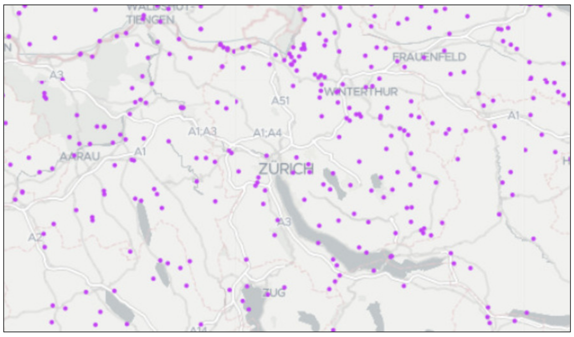

Fig. 1. Screenshot of Castle Map showing violet dots represented as vector and raster tile data in front of a base map
Own presentment

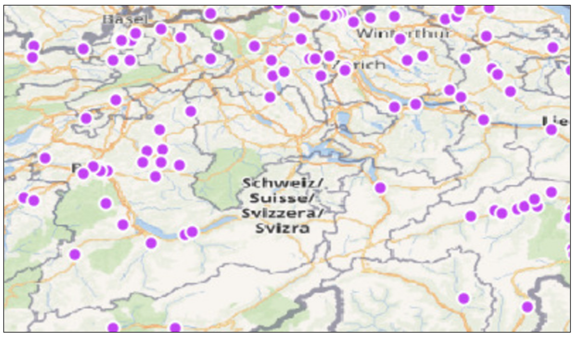

Fig. 2. QGIS's experimental WFS3 (OAPIF) vector layer communicating with the API
Own presentment

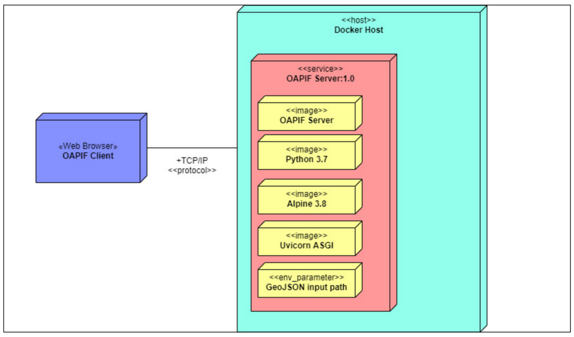

Fig. 3. UML Deployment Diagram describing the deployment architecture of own implementation
Own presentment

Introduction: Display services are an important standard in the geospatial domain. The Open Geospatial Consortium (OGC) released serveral APIs to deal with feature (vector) and raster data.

This project is an application that serves as an API for web applications that deal with geospatial data and especially for clients which support the "OGC API - Features" (OAPIF) standard.

The goal of this thesis project is to implement a service in Python which is minimally compliant to the OAPIF standard.

In order to speed up the display of features, another API has to be implemented that conforms to another OGC standard, called Web Map Tile Service (WMTS) which returns raster tiles. The input to the service can be a large GeoJSON file. Typical URL parameters of the APIs are a zoom level and a coordinate position.

The implementation has to be showcased with an existing web application, called "Castle Map" (see Fig. 1). This is an interactive web map of castles in Switzerland and neighbouring countries. It combines data from OpenStreetMap, Wikidata, Wikimedia Commons and Wikipedia.

In addition, the implementation is required to serve as a service for the desktop application QGIS (see Fig. 2).

Approach / Technology: The implemented application uses at it's core the Geospatial Data Abstraction Library (GDAL) for reading and writing raster and vector geospatial data formats. GDAL has a Python wrapper and includes an OAPIF driver.

The application first loads one or more GeoJSON files (called layers) into memory as input which contain the main feature collections. It then processes this data in order to give the required API responses.

The application is containerized in a Docker container, which contains an image consisting of: Alpine 3.8 which is a Linux distribution OS, Python 3.7, which is the main programming language for this application and Uvicorn, the light-weight server used to run the application (See Fig. 3). The application was also unit tested using "pytest" which is a Python unit testing framework.

Result: All requirements mentioned in first section could be fulfilled with the implementation: The service works together with The Castle Map as well as with QGIS version 3.12.

In order to test the implementation, the command line interface program "ogrinfo" has been used which is part of the GDAL open source package. ogrinfo lists information about a OGC compliant service. This test lead to program adaptions and ensured an even better degree of compliancy, especially of the new OAPIF standard.

In the future, this application might be benchmarked for performance, comparing it to the existing API that the Castle Map showcase uses in order to assert which one delivers better performance.

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz