

Vector tile cache connecting effective spatial communication and geospatial AI



by SHARON CHAWANJI

The principle goal of this thesis is to examine the possibility of using cached vector tiles as a geospatial framework in the Table Joining service (TJS): for communicating spatial information. TJS is an OGC Web Service that define how tabular data can be integrated with a geographic framework [1].

BACKGROUND

Linked open data (LOD) has explored globally. Datasets such as social media data and open government data are being published as LOD. By applying Geospatial AI, concealed knowledge can be extracted from LOD and visualised on maps. LOD can be accessed via SPARQL endpoints; afterwards it can be converted to manageable data formats like CSV. Often the data is embedded with geographic identifiers (e.g. area codes) which can be linked to geographic features packaged in vector tiles. TJS is used to integrate the CSV data with vector tiles, thus creating thematic maps that can be visualized on web GIS applications [2]. Vector tiles can be cached easily, and they are considered the most efficient and optimum format of delivering geospatial information on the Web [3]. However, the use of vector tile caches in TJS has not well understood.

RESEARCH OBJECTIVES

1. Examine the possibility of using **vector tile caches** as a geospatial framework for TJS.
2. Develop a TJS API prototype that can join cached vector tiles with attribute data.
3. Develop a web map application for displaying results of vector tiles joined with attribute data using the TJS API.

METHODS

To provide a proof of concept a prototype is built based on a three-layer software architecture (see Fig.1). The three layers are: data layer, application layer and client layer.

PROTOTYPE IMPLEMENTATION

TJS API is implemented in the application layer. The API provides an interface to join attribute data in CSV format with cached vector tiles encoded in GeoJSON format. The CSV and GeoJSON are converted into

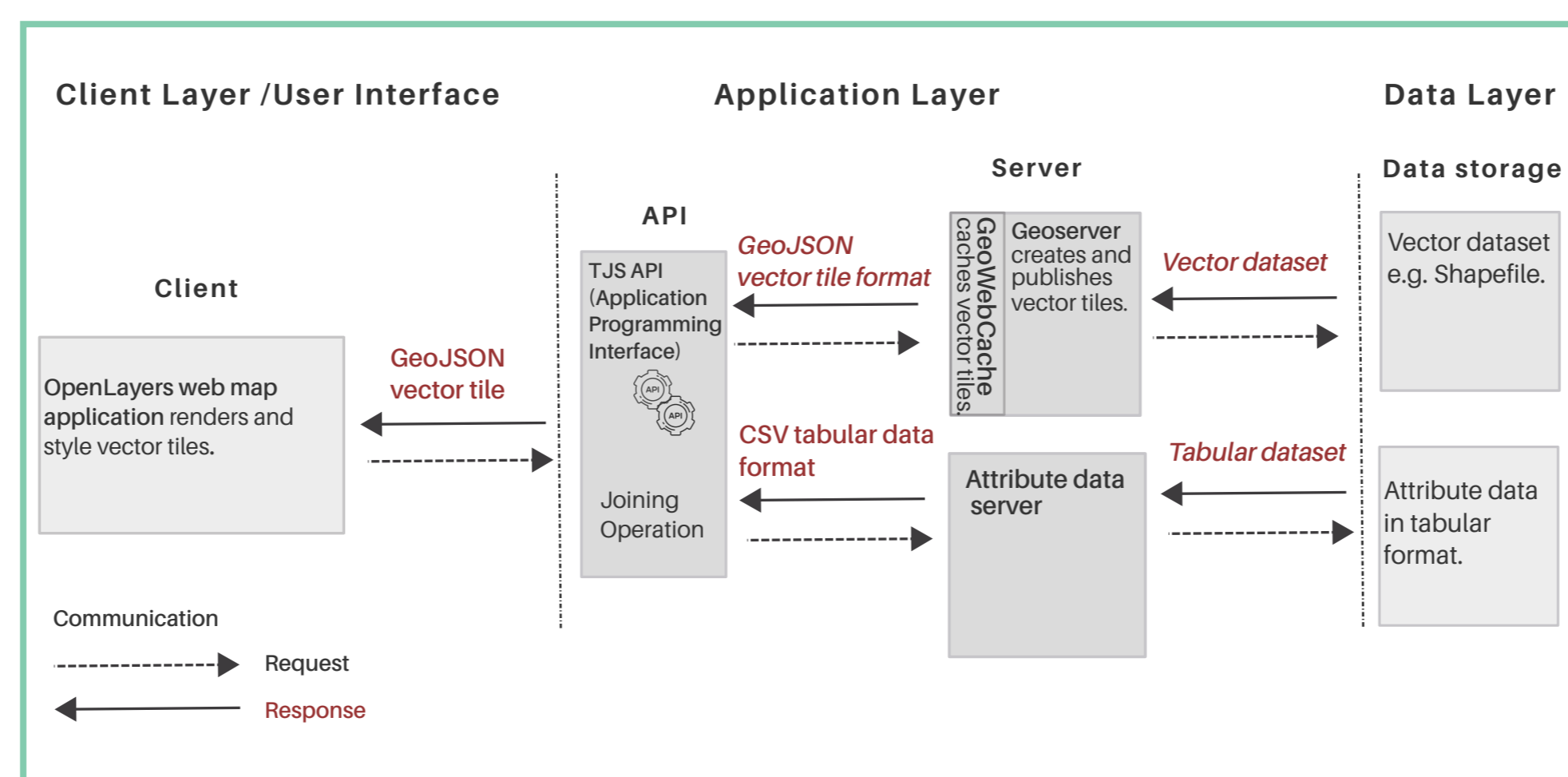


Fig. 1. Three-layer software architecture.

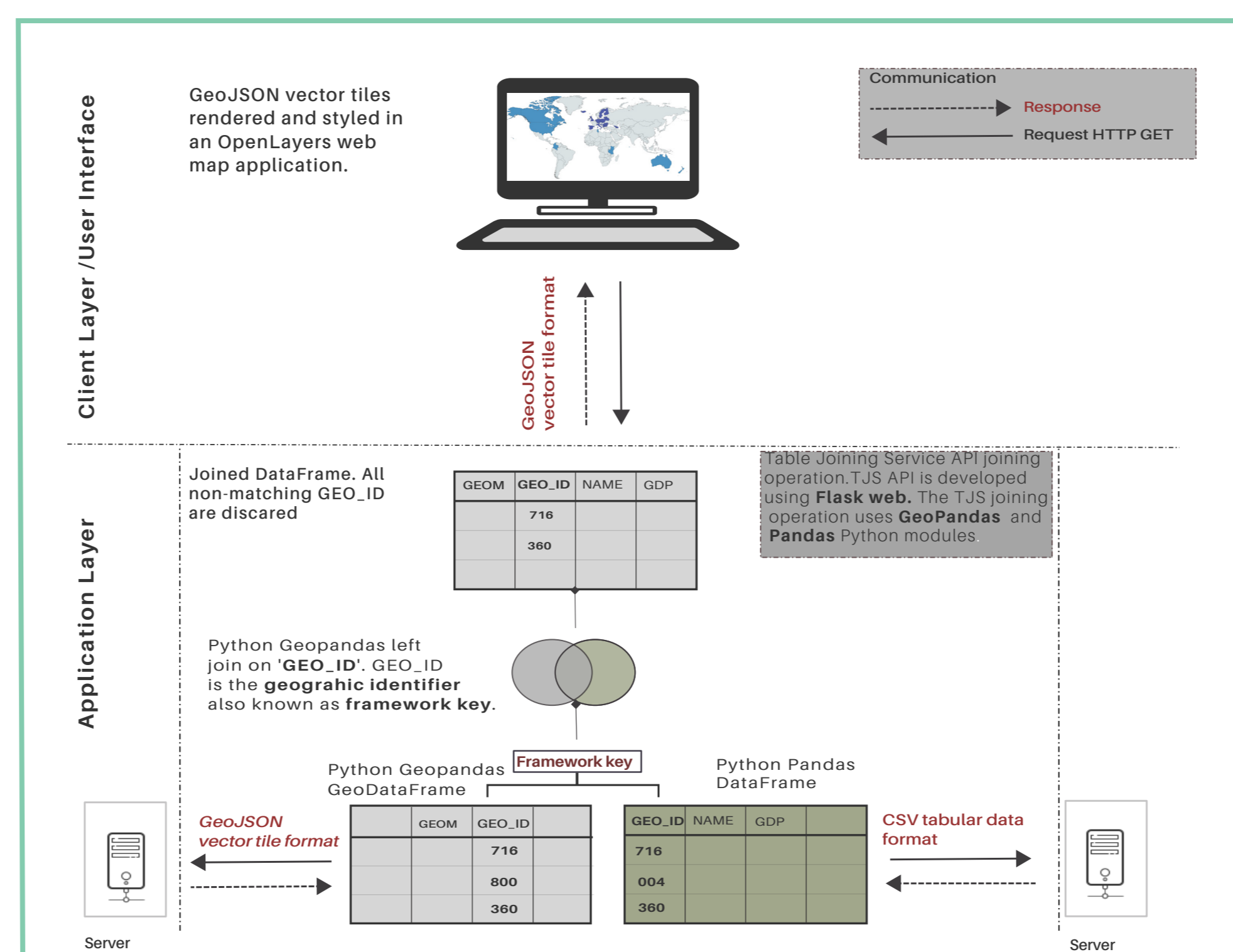


Fig. 2. Table Joining Service API joining operation using Python modules.

dataframes; so that they can be joined based on common framework keys/geographic identifiers (GID). The joined dataframe is converted back to GeoJSON vector tiles. An OpenLayers (OL) web application, in the client layer, renders and styles the vector tiles from the TJS API (see Fig.2).

DISCUSSION AND CONCLUSION

The prerequisite of TJS are attribute data, framework data and GIDs. GIDs are important for the successful merging of attribute data with the vector tiles. In this research standardised ISO 3166-1 numeric codes were used as GIDs. It was crucial for all geographic features in the vector tiles to contain unique GIDs. In the joining operation, unmatched records were discarded which disrupted the rendering of vector tiles on the OL web application. The solution was to make sure there were corresponding records of GIDs in attribute data even for no data values.

The software architecture used enables geospatial data and attribute data to be managed autonomously. The TJS API provided a RESTful interface for joining the datasets together.

The study proves that it is possible to use cached vector tiles as a geospatial framework for TJS.

THESIS CONDUCTED AT

Research Division Cartography
Department of Geodesy and
Geoinformation
Technische Universität Wien



THESIS ASSESSMENT BOARD

Chair Professor: Univ.Prof. Mag.rer.nat.
Dr.rer.nat. Georg Gartner, TU Wien

Supervisor: Dipl.-Ing. Dr. Markus Jobst,
TU Wien

Reviewer: M.Sc. Mathias Gröbe, TU
Dresden

YEAR

2020

KEYWORDS

Vector tile cache, Table Joining
Service, SPARQL endpoint

LINK TO TJS API PROTOTYPE

<https://github.com/schawanji/TJS-Server>

REFERENCES

- [1] Grothe, M., Brentjens, T., 2013. Joining tabular and geographic data – Merits and possibilities of the Table Joining Service 53
- [2] Hong, J.H., Lin, S.Y., 2005. Web-based thematic map service in openGIS environment.
- [3] Ingensand, J. (Ed.), 2019. OGC Testbed-13: Vector Tiles Engineering Report 163.