COMMON DATA MODEL FOR THE MICROSERVICES OF A RADIOPROPAGATION TOOL

Article Authors:
Adrian Valledor - Marcos Barranquero - Juan Casado - Josefa Gómez- Albdelhamid Tayebi

Presenter:
Adrián Valledor - Computer Science Dept.
Universidad de Alcalá - Madrid, Spain
e-mail: adrian.valledor@uah.es

The presenter is Adrián Valledor

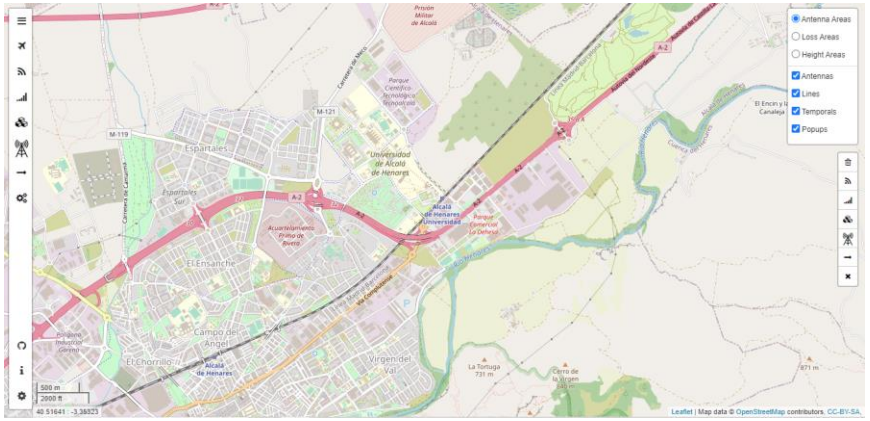
- He is a information systems engineer.
- This year, he is studying a Master's Degree in Agile Software Development for the web.
- Also, at this moment, he is working with the Computer Science Department in Universidad de Alcalá de Henares.

Table of Contents.

Introduction.

In recent years, there has been a need to **migrate from the old monolithic systems to current microservices models**.

The microservices can be independent and therefore they may not share the same specifications. In this case, **the improvement of a radio propagation simulation tool is being developed.**



State of Art.

GraphQL

- GraphQL is a query-based language.
- It provides a complete description of the data in its API, allowing users to request what they need.

Rest APIs

- It is a set of architectural principles that fits the specific needs of each application as defined by Dr. Roy Fielding.
- This provides a high level of flexibility and freedom for development of a microservices architecture.

Although tools exist that facilitate sharing the same data model, dependencies are costly and can complicate the architecture and communications between servers.

```
1
2   FROM node:latest
3
4   ARG PORT
5   ARG ACTION
6   ARG ROOT=/app
7
8   ENV ACTION=${ACTION}
9   ENV PORT=${PORT}
10
11  RUN mkdir ${ROOT}
12  WORKDIR ${ROOT}
13
14  COPY package.json ${ROOT}
15  COPY index.js ${ROOT}
16  COPY load.js ${ROOT}
17  COPY ${ACTION}.js ${ROOT}
18  RUN npm install
19
20  EXPOSE ${PORT}
21  CMD npm start
```

Microservices with Docker.

Another solution related to microservices is the **use of Docker**.

This small development in Docker it is possible to load the microservices components and code dynamically. That allows to have the same interfaces and code for all of the microservices, and loading only the part of the code relevant to that microservice encapsulated in one container. In the following case, it is used with servers of different types depending on whether they are necessary or not.

Solution Specifications.

In the **back-end**, the microservices are responsible for providing the necessary data both to show the users of the application and to carry out internal operations.

There is a **front-end** in charge of representing graphically in the browser the data collected from the servers in the form of geometries representing that data.

Solution Specifications.

As a **solution to the data model,** it is decided to create a common data type that will be used transversally across the different microservices of the application and that has been defined as ”Specification”.

This Specification is based on the common properties that all geometry is considered to have geometry being a fundamental structure in the application.

The Specification will have a similar structure to JSON type files and is defined by the type of geometry it represents, as well as the coordinates of the points that compose it.


```
{"type":"Feature","geometry":{"type":"Polygon","coordinates":
[[-3.3555140044189393,40.521139148981185],[-3.3555140044189393,40.50728043088608],
[-3.3327857067429636,40.50728043088608],[-3.3327857067429636,40.521139148981185]]},"properties":
{"subType":"Rectangle"},"specificationTypes":"RectangleSpecification"}
```

Advantages and Disadvantages.

Advantages

- Use of a common data model.
- No external dependencies.
- Easy to maintain structure, change tolerant.
- It provides a simple overview of the tool.

Advantages and Disadvantages.

Disadvantages

- It does not have the possibility of being reusable in other developments.
- It is a too concrete design, focused on the actual tool.

Conclusion.

- It can be seen how beneficial it is for a structure such as this tool the importance of maintaining an architecture through microservices.
- Elimination of external dependencies.
- It adds modularization on the project, providing scalability and load balance between front-end and back-end.
- Along with this, current alternatives to this data model have been seen, but they do not fit with the situation of the tool being developed, either because of its size or because of future dependencies.

Future work.

- The improvement of the user interface.
- The relationship of this with the different libraries that allow the representation of maps in the browser.
- Make use of these specifications designed in the data model.

Future work.

- This improvement will be based on the use of the OpenLayers library, allowing the map to represent different operations in a browser.
- The react framework will improve the frontend, offering the possibility for the user to add the data for representing the designed specifications.
- The data will be kept accessible throughout the entire structure of the tool.

References.

[1] J. Thönes, "Microservices," in IEEE Software, vol. 32, no. 1, pp. 116- 116, Jan.-Feb. 2015, doi: 10.1109/MS.2015.11.

[2] A. Tayebi, J. Gomez, F. Saez de Adana, O. Gutierrez, M. Fernandez de Sevilla, "Development of a Web-Based Simulation Tool to Estimate the Path Loss in Outdoor Environments using OpenStreetMaps [Wireless Corner]," IEEE Antennas and Propagation Magazine, vol. 61, no. 1, pp. 123-129, Feb. 2019, doi: 10.1109/MAP.2018.2883088.

[3] F. Saez De Adana, J. Gómez, A. Tayebi, J. Casado, "Applications of Geographic Information Systems for Wireless Network Planning", Artech, 2020.

[4] GraphQL — A query language for your API.[Online]. Available from: http://www.Graphql.org. June, 2022.

[5] R. Fielding and R. Taylor, "Principled design of the modern Web architecture". ACM transactions on Internet technology, 2(2), pp.115–150, 2002, doi: 10.1145/514183.514185.

[6] J. Gómez, A. Tayebi, J. Casado, 2020, "On the use of Websockets to maintain temporal states in stateless applications", in The 15th International Conference on Internet and Web Applications and Services ICIW 2020, pp. 21-24.

[7] X. Wan, X. Guan, T. Wang, G. Bai, B. Choi, "Application deployment using Microservice and Docker containers: Framework and optimization". Journal of Network and Computer Applications, 119, pp.97-109, 2018, doi: 10.1016/j.jnca.2018.07.003.

[8] O. Zabala-Romero, E. Chassignet, J. Zabala-Hidalgo, P. Velissariou, H. Pandav, A. Meyer-Baese, "OWGIS 2.0: Open source Java application that builds web GIS interfaces for desktop and mobile devices", SIGSPATIAL'14: Proceedings of the 22sn ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 311-320, doi:10.1145/2666310.2666381.

[9] C. M. Novac, O. C. Novac, R. M. Sferle, M. I. Gordan, G. Bujdoso, C. M. Dindelegan, "Comparative study of some applications made in the Vue.js and React.js frameworks". 16th International Conference on Engineering of Modern Electric Systems (EMES), 2021, pp. 1-4, doi: 10.1109/EMES52337.2021.9484149.

Thanks for your attention.