Cross-border and Cross-domain Integration of 3D Content in a European Geospatially Enabled Ecosystem

Lassi Lehto, Jaakko Kähkönen
Department of Geoinformatics and Cartography
Finnish Geospatial Research Institute
Masala, Finland
e-mail: lassi.lehto@nls.fi, jaakko.kahkonen@nls.fi

Jari Reini[a], Timo Aarnio[a], Roope Tervo[b]
National Land Survey of Finland[a]
Finnish Meteorological Institute[b]
Helsinki, Finland
e-mail: jari.reini@nls.fi, timo.aarnio@nls.fi,
roope.tervo@fmi.fi

Abstract—**The Geospatially Enabled Ecosystem for Europe (GeoE3) project has as the main goal to develop use case-oriented, cross-border and cross-domain geospatial services conforming to the second-generation interface standards. The five-nation project focusses on cross-domain integration of content by applying dynamic, service level methods for joining climate and statistical data with geospatial features. Important climate parameters have been successfully integrated with building data to support the envisaged use cases. Various technologies are being tested for providing seamless, cross-border access to geospatial resources in context of the new Open Geospatial Consortium's OGC API family of service interface standards. 3D aspects of geospatial data are also being considered, both from content encoding and service interface point of view. Both pre-created 3D data sources and dynamic, on-the-fly techniques are applied in the 3D content provision. The project aims at supporting renewable energy- and smart city-related applications.**

Keywords-geospatial ecosystem; cross-border; cross-domain; climate; OGC API; 3D; solar energy; smart city.

I. INTRODUCTION

In an interoperable co-operative computing environment, every participating actor should gain benefits. This is the main guiding principle in the spatial data ecosystems currently being developed. A major action, Geospatially Enabled Ecosystem for Europe (GeoE3), is working to establish an ecosystem of use case-oriented geospatial services in five European countries [1]. The action is part of the Connecting Europe Facility (CEF) programme advancing European transport, energy and digital infrastructure development [2]. The project runs for three years and involves five national mapping and cadastral agencies, together with a few other governmental organizations and private companies [3]. The main goals of the project include adaptation of modern geospatial APIs (Application Programming Interfaces) to establish cross-border services for renewable energy and urbanization related use cases, use of dynamic service-level mechanisms for cross-domain content integration, and promotion of the developed approaches through extensive online innovation and education programme.

GeoE3 consortium membership includes the national mapping or cadastral agency from Finland, Norway, Estonia, The Netherlands and Spain. Furthermore, the consortium includes the national meteorological and statistical agency of Finland: Finnish Meteorological Institute and Statistics Finland, respectively. Private companies include Spatineo from Finland and Aventi Intelligent Communication from Norway. Open Geospatial Consortium (OGC) Europe participates in the project as a representative of the standardization community. National Land Survey of Finland coordinates the project that will be finalized by Sep 2023 [4].

The main use cases identified by GeoE3 include renewable energy applications, specifically in the context of buildings construction and use. In particular, this involves solar energy potential, based on solar panels on rooftops. Wind energy and heating/cooling facilities of the buildings are considered too. The other use cases deal with traffic applications, in particular efficient use of electric cars, and with various urbanization challenges. The last one is considering urban expansion efficiency employing the United Nation's Sustainable Development Goal (SDG) indicator 11.3.1 'Ratio of land consumption rate to population growth rate' [5].

The paper is organized as follows. In Section II, the envisioned service architecture of the GeoE3 platform is described. Section III discusses various mechanisms for content integration. In Section IV, some novel ideas concerning OGC API Features functionality are discussed. Section V describes results of the coverage data integration experiments. Section VI presents the initial considerations of the project on dealing with 3D geodata and Section VII describes the first results achieved. The paper ends with conclusions in Section VIII.

II. SERVICE ARCHITECTURE

The GeoE3 project aims at establishing a set of use case - oriented services that enable content integration both across national borders and domain boundaries. The idea is to set up an integration layer on top of national services. The integration layer provides modern service interfaces to the client side and accommodates various categories of source services on the country level. The GeoE3 services are based on the OGC API family of second-generation, internationally standardized service interfaces [6]. These include services like OGC API Features [7], OGC API Coverages [8] and OGC API Records [9]. Internally the GeoE3 platform will apply at least OGC

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

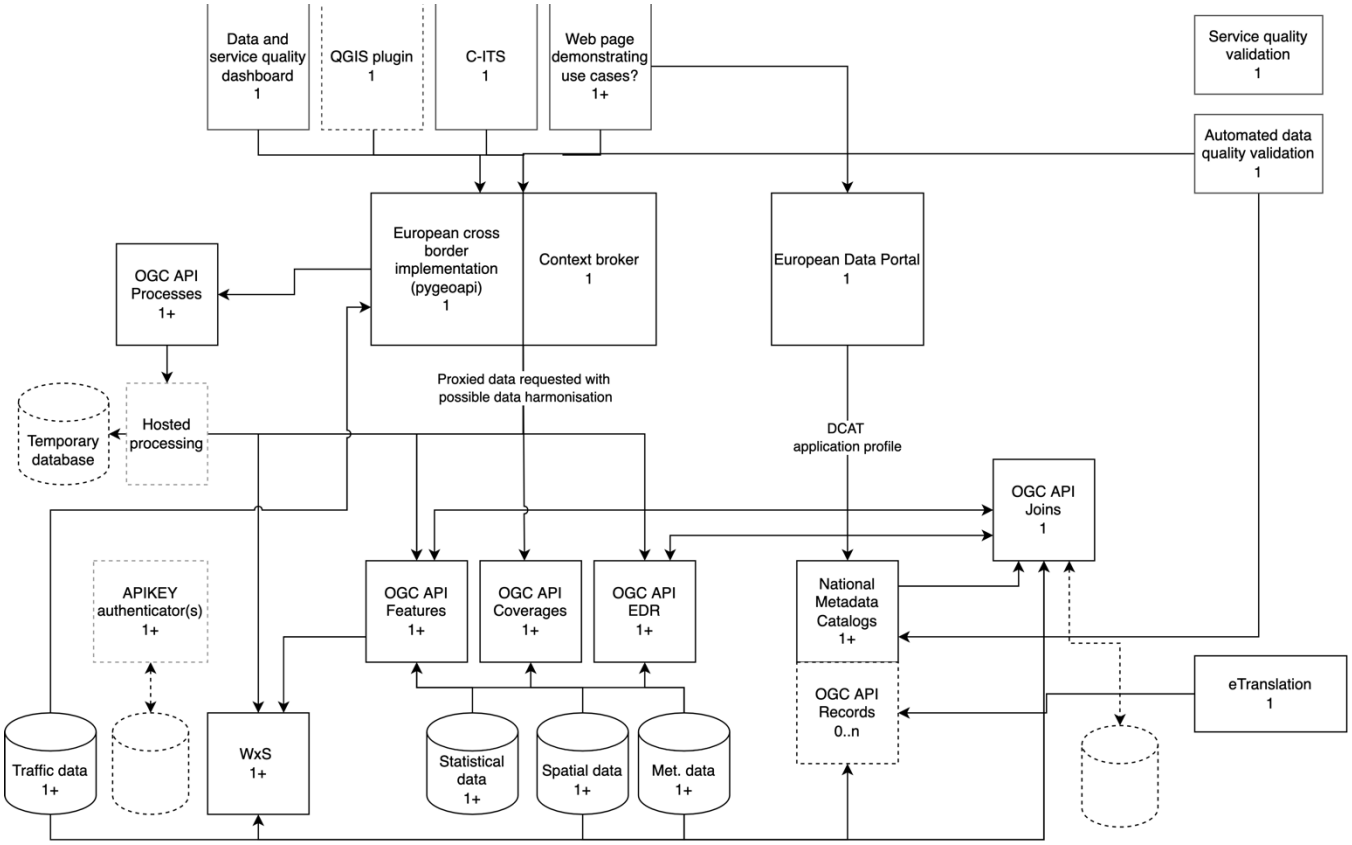

Figure 1. The GeoE3 service architecture.

API Processes [10] for introducing analysis-based content layers to the clients and OGC API Joins [11] for content integration across domain boundaries. OGC API Environment Data Retrieval (EDR) [12] interface will be used for requesting meteorological data, particularly for supporting transport-related applications.

The illustration in Fig. 1 shows the GeoE3 service architecture. The WxS box in the left bottom corner represents all the country level data resources that will mostly be accessed through service access points conforming to the legacy OGC standard interfaces, like Web Feature Service (WFS), Web Coverage Service (WCS), and the OGC API family of services. The databases at the bottom emphasize the GeoE3 goal of integrating spatial data with statistical and other tabular data sets. All the data resources are made available via modern interface standards and are integrated in cross-border manner on the GeoE3 integration platform (European cross-border implementation in Fig. 1).

Metadata records of all GeoE3 services will be harvested from the national metadata catalogues into the European Data Portal (EDP) and also be available from the OGC API Records service interface maintained on the GeoE3 platform. The CEF eTranslation [13] building block tool will be used to provide translations between the national language and the target language. It provides machine-to-machine use via a Web service interface for translating text and documents in the multilingual European context. The eTranslation Web service is asynchronous. In other words, the client sends a translation request and is notified later, when the text snippet or document is translated. The client needs to expose a call-back URL, which will receive a notification that the translation job in question has been completed. Translated metadata documents will be published via OGC API - Records interface.

An implementation of the OGC API Joins standard is an essential component of the GeoE3 service architecture. It will be responsible for joining tabular data sets with geospatial features. The resulting combined resource is then made accessible to client applications via an OGC API Features service.

The GeoE3 integration platform is a cloud service-based solution for aggregated access to all content becoming available in the context of the project. Django is used as the Web application framework on the platform [14]. The main geospatial software package used on the platform for the provision of OGC API compliant services is a product called pygeoapi [15]. pygeoapi is a Python library supporting OGC API Features and partially supporting OGC API Coverages, OGC API Records and OGC API Processes interfaces. pygeoapi is the reference implementation for the OGC API Features specification.

In the GeoE3 project, pygeoapi has been tailored in many ways. First of all, it has been adapted to be used in the Django framework. Map visualisation library originally used in the

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

product, Leaflet, has been replaced by more feature-rich OpenLayers [16]. Fundamental changes have been made to pygeoapi's handling of OGC API Features HTML-formatted output (see Section IV). OGC API Coverages implementation has been enhanced to support Web-based viewing of the coverage data. This involves for example support for PNG images and the 'scale-size' parameter in the coverage request. A set of content provider plugins for various different data sources have also been developed. Some work has been invested in developing more advanced functionalities for the Web user interface, for example to support 3D viewing of content (see Section VII).

III. CONTENT INTEGRATION

A. Content Integration with OGC API Features

An essential component on the GeoE3 integration layer is an OGC API Features service implementation. The successor for the OGC's WFS interface specification, OGC API Features follows the principles of the REST (Representational State Transfer) service architecture. The output of the service consists of individual geospatial features, organized as a GeoJSON -encoded feature collection [17], geometries expressed in WGS84 Coordinate Reference System (CRS). The OGC API Features service instance offers a set of data collections, accessible from the given service address paths. This provides a natural setting for organizing individual countries' data sets as collections inside a single OGC API Features instance. For instance, the buildings data sets of Finland, Norway and Spain could be organized as follows.

- /geoe3/buildings/collections/buildings_FI/items
- /geoe3/buildings/collections/buildings_NO/items
- /geoe3/buildings/collections/buildings_ES/items

The service implementation adopted in the GeoE3 platform, pygeoapi, provides a flexible plugin architecture for content provision and formatting. The off-the-shelf library supports, via OGR Simple Features Library [18] and rasterio [19] provider plugins, a vast array of vector and raster source formats and services. For example, WFS is supported as source service. OGR provider also supports OGC API Features service access, thus enabling cascading service approaches. Specific project-related needs can be accommodated by developing a tailored plugin. As the plugin architecture is based on Python language, all the Python-based geospatial libraries can be easily utilized. The set of already available preliminary GeoE3 components include plugins for WFS versions 1.1.0 and 2.0.0 with basic schema transformation capabilities, a plugin for WCS version 1.0.0, a plugin for a remote OGC API Features service and a plugin for accessing raster data from a virtual file (VRT) -based data storage.

Schema transformations form an essential part of the functionality of a vector data provider plugin. The principles of the so-called alternative encoding INSPIRE (Infrastructure for Spatial Information in Europe) schema [20] is used as the approach for the common target schema in GeoE3. So far the focus has mostly been on the content theme 'Buildings'. Schema transformations have been developed from two national schemas (Finland and The Netherlands), from two legacy INSPIRE schemas (Norway and Spain), and from one INSPIRE-like schema (Estonia) to the common target schema. The transformation also involves translating the data elements from the GML (Geography Markup Language) encoding to the GeoJSON encoding, in case of four countries. The property names of the current version of the target schema for theme 'Buildings' are presented in Table 1.

TABLE 1. GEOE3 TARGET SCHEMA

inspireId_localId
inspireId_namespace
inspireId_versionId
beginLifespanVersion
endLifespanVersion
externalReference_informationSystem
externalReference_informationSystemName
externalReference_reference
conditionOfConstruction
dateOfConstruction
currentUse_currentUse
currentUse_currentUse_href
currentUse_percentage
address_adminUnit_name_[lang]
address_postCode
address_thoroughfare_name_[lang]
address_locator_designator_addressNumber
cadastralParcel_label
officialArea_value
floorDescription_floorArea
heightAboveGround_value
heightAboveGround_status
heightAboveGround_status_href
volume
numberOfFloorsAboveGround
numberOfBuildingUnits
numberOfDwellings
heatingSystem
heatingSource
materialOfStructure
constructionMethod
materialOfFacade
geometry2D_referenceGeometry
geometry2D_horizontalGeometryReference
geometry2D_horizontalGeometryReference_href
geometry2D_verticalGeometryReference
geometry2D_verticalGeometryReference_href

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The INSPIRE-defined model simplification rules applied in the schema definition process include General Flattening, Associated Component Soft Type, Simple Codelist Reference and Simple Geographic Name [21]. Two property names in the schema cannot be derived from INSPIRE data models: 'volume' and 'constructionMethod'. These names can be taken as proposals for a further extension of INSPIRE schemas.

Another important consideration in geospatial data integration is the use of Coordinate Reference Systems. According to the GeoJSON specification, the CRS to be used in GeoJSON-compliant files is WGS84 with axis order: longitude, latitude (identifier: urn:ogc:def:crs:OGC::CRS84). Consequently, all queries to the GeoE3 integration platform's OGC API Features services must be expressed in this CRS. However, all the source data sets of the OGC API Coverages services are stored in the national CRSs and queries are thus expected in these CRSs. In the GeoE3 project a decision has been made to use Pseudo Mercator (identifier: urn:ogc:def:crs:EPSG::3857) as the common CRS on the client side. OpenLayers, the map application library used in the current user interface, is responsible for carrying out the required CRS transformations to and from the client side CRS.

B. *Content Integration with OGC API Processes*

OGC API Processes is a new resource-based process definition standard. The standard defines how the client program can start the process, how the process inputs are provided, and where the results of the process are stored.

Computational tasks are wrapped inside the process and the results are provided to the client application. Processes can be for example geometry buffering or routing from point A to point B. The process can be either synchronous, giving the results immediately, or asynchronous, in which case the service informs the client, when the results are ready and where they can be found.

The GeoE3 project has developed a process, in which the 3D geometry of the building is dynamically generated. In addition, a process is being developed to calculate the average amount of solar energy in a building area. This dynamically created information can then be integrated back to the original features.

The OGC API Processes interface standard follows the same principles as the other OGC API standards (e.g., OGC API Features). The path structure of the GeoE3 OGC API Processes services for theme 'Buildings' might be as follows.

- /geoe3/buildings (The service landing page)
- /geoe3/buildings/conformance (Conformance classes describing what service can do)
- /geoe3/buildings/processes (The list of processes in the service
- /geoe3/buildings/processes/extrude-building (Process to create 3D buildings from the building footprint)
- /geoe3/buildings/processes/solarenergy-potential (Solar energy potential calculator)

The first application of OGC API Processes -based functionality in the GeoE3 project is the dynamic creation of LOD1 (LOD: Level of Detail) category 3D models of buildings. The on-the-fly computation has been implemented on the GeoE3 integration platform and is launched automatically in case only part of the buildings is available as pre-created LOD2 models (in case of Finland). If 3D buildings do not exist at all, the on-the-fly computation is applied always (in case of Norway).

The computation uses as input the 2D building footprints, the Digital Terrain Model (DTM) and Digital Surface Model (DSM) elevation values from within the building polygon. When the user selects an individual building in the client application, the corresponding feature data is first requested from the GeoE3 OGC API Features service using the buildings feature identifier. The 2D geometry is extracted from the received feature and used as input for requests to retrieve DTM and DSM values contained within the building outline polygon. The lowest DTM value is used to determine the elevation of the building's floor. An average of the DSM values is used for elevation of the roof. To minimise the possible deviation caused by nearby trees, only lower half of the DSM values are used. Finally, a CityJSON encoding of the LOD1 building model is generated using the building's footprint polygon and the computed elevations as input. An example of the resulting 3D view is shown in Fig. 2. As another example, in Fig. 3 dynamically created LOD1 building models are shown with bluish colour, together with buildings in genuine, pre-created LOD2 models with red coloured roofs.

C. *Content Integration with OGC API Joins*

OGC API Joins provides a method for joining spatial and tabular data. The standard is currently in draft status in OGC and is going to be used in GeoE3. It allows integration of tabular statistical data to the platform.

To carry out the join, the service needs the spatial and tabular data and a common identifier between them. Several columns of the tabular data can be joined at once. Both the spatial and tabular data can either be configured to be available from the server or they can be uploaded on demand. Spatial data can thus be delivered via OGC Features API services and tabular data, e.g., via PxWeb API [22] compliant services.

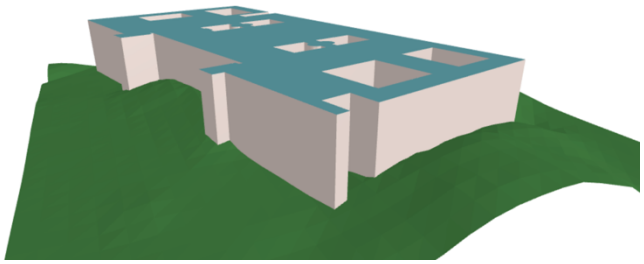

Figure 2. An example of a LOD1 model of a building (the Finnish House of Parliament), created in an on-the-fly OGC API Processes -based computation. Shown together with DTM.

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Figure 3. Buildings with on-the-fly computed LOD1 models (bluish roofs), shown together with buildings represented with a pre-created LOD2 model (red coloured roofs) in the 3D viewing component (in the right bottom corner).

The goal of the project is to evaluate the fitness for purpose of the OGC API Joins standard in the use cases identified in the project. An OGC API Joins instance will be set up and configured, along with a browser-based end user interface. While OGC API Joins can be demonstrated and used with a user interface, it can be more valuable when used in the background as a part of an automated process. An obvious benefit of joining data is that spatial data need not be stored in several places.

The output of OGC API Joins has several options. The combined data can be downloaded for further use in several spatial formats, such as GeoJSON. Furthermore, results from the joining operation can be calculated from the data. This includes information about successful joins, unsuccessful joins and extra data rows. In the project this derived data will be visualized in the user interface as a report for the user.

IV. OGC API Features Rethought

Some of the OGC API Features concepts have been rethought in the development of the preliminary GeoE3 platform. For instance, the HTML-formatted feature browsing in the OGC API Features development is generally understood as a tabular presentation of feature properties, browsed through the feature collection page by page. When there are potentially millions of features in the national databases, and when the order of features presented is arbitrary, the browsing process often becomes unfeasible. The feature browsing in the GeoE3 OGC API Features is returned back to the traditional spatially organized, map-based browsing. A plain background map is first shown to the user. The individual vector features are requested and presented only when the user zooms in deep enough.

Another new concept introduced in the GeoE3 project is the idea of treating the HTML-formatted visualization of an individual spatial feature as an application-specific dashboard of the feature [23]. Into this dashboard, a wide set of feature properties and visualizations can be collected by the GeoE3 integration platform. For example, a sun energy-related building dashboard could contain all the relevant building attributes - like the area, volume and heating information, and the DSM-based sun exposure analysis from the building's area - useful for solar energy analyses. Furthermore, climate-related information could be integrated to the same dashboard on the GeoE3 platform by accessing appropriate values from a meteorological service. The dashboard could be seen as a shop window for the available content, and could later be downloaded into a digital analysis process in form of the corresponding JSON-formatted data set. An example of a prototypical building dashboard for sun energy-energy related applications is shown in Fig. 4. OGC API Features browsing by map view is shown on the left. On the right, a detail window of the selected feature is shown as 2D and 3D representations. Further on, the full set of feature attributes is displayed, together with a detailed 2D map with orthophoto as the background layer. The third user interface presents the building in the full-size 3D viewing module.

The sun exposure computation used in the GeoE3 feature dashboard is based on the geomorphometric analysis function 'TimeInDaylight' of WhiteBoxTools [24]. The calculation is based on the use of the DSM values around the building. In the server side the process is currently organized as a cascading Web Map Service (WMS). The process providing the source image retrieves the required part of the DSM data

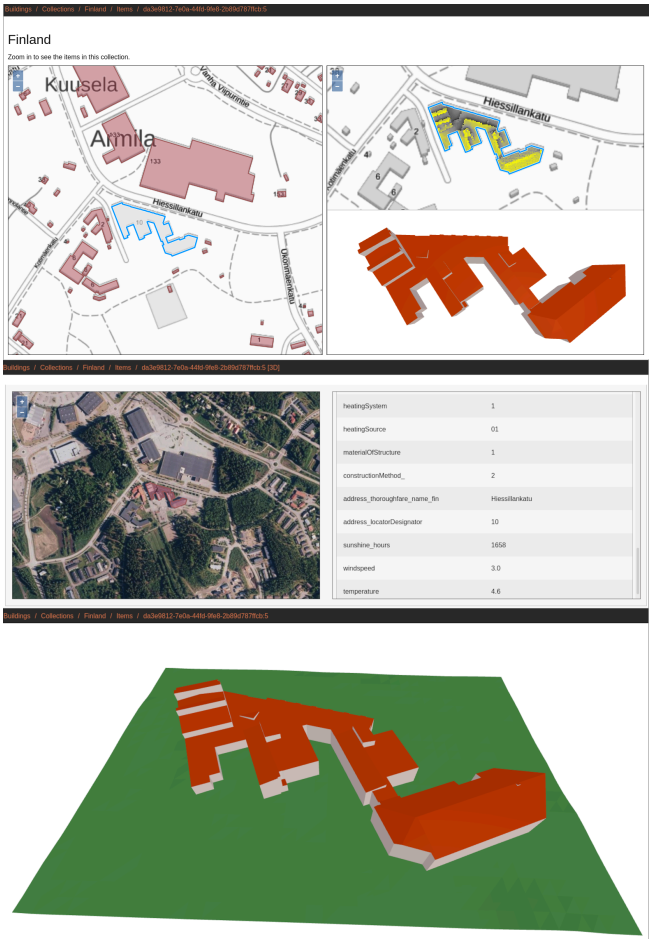

Figure 4. Preliminary OGC API Features html output from the GeoE3 platform as a building's renewable energy dashboard.

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

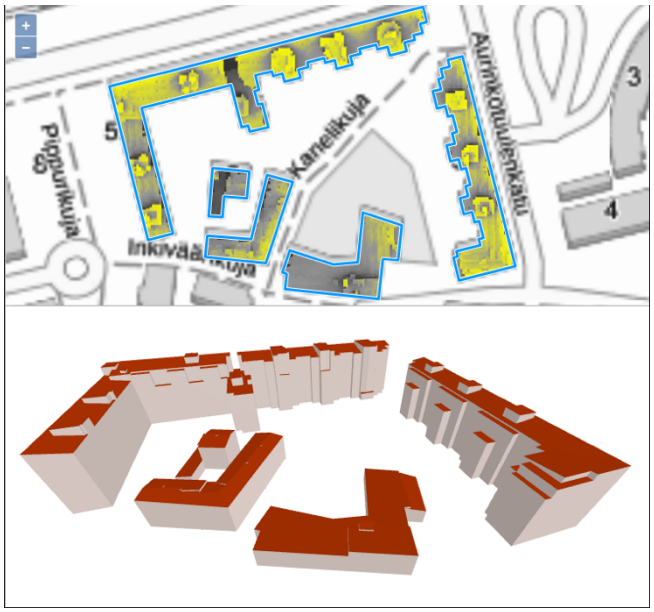

Figure 5. The result of the sunlight exposure analysis of the selected group of buildings (top), together with a LOD2 3D model viewing component (bottom).

from the GeoE3 OGC API Coverages service (see Section V). The resulting exposure information is formatted as TIFF image, which is then converted by the WMS service to a PNG image for viewing in the Web client environment. The display of the analysis results is presented in Fig. 5. In future the analysis is planned to be implemented and further developed as a process inside the OGC API Processes service implementation of the GeoE3 platform (see Section IIIB).

V. COVERAGE DATA

The GeoE3 project is also involving coverage data in its content integration development. This is somewhat novel approach, as most of the previous projects aiming at geodata harmonization and integration have been focusing on vector-formatted data sets. The first coverage data sets being dealt with in the GeoE3 project include themes like elevation and climate.

A. Elevation data

Given the envisaged use case on solar energy potential of a building rooftop, DSM becomes an important source data set. Based on the DSM values, the form of the roof can be automatically derived. The amount of sun exposure can be also evaluated, based on the DSM of the surrounding areas. DTM is important for determining the absolute elevation of the building's floor level, which is needed for instance in volume calculations and integrated visualizations.

The OGC API Coverages service interface is implemented in the GeoE3 integration platform's coverage-related services. The OGC API Coverages is currently available as a draft specification and is thus not yet widely implemented. However, the pygeoapi software package, used in GeoE3 integration platform, provides preliminary support for the specification. In the project the OGC API Coverages functionality of pygeoapi has been extended in various ways. Provider plugins have been developed to support access to WCS -based source services. Support for PNG output has been added to support visualization in Web clients. An important addition is the possibility to scale the output grid to the resolution appropriate for the client's needs. This is achieved by supporting the 'scale-size' parameter in the coverage query.

As with the OGC API Feature case, explained in Section IV, the Web based access to the OGC API Coverages service interface has been extended in the GeoE3 project. The goal is to facilitate easy viewing of the coverage contents in the Web user interface. To enable this, support for HTML formatted output has been added to pygeoapi. The initial output for the coverage content request in HTML format is an HTML page with an empty map window. The client application can then first perform a request for a background map and then send back a query for a PNG-formatted coverage. The initial view covers the full extent of the coverage. The background map will then enable the user to zoom in and pan the map appropriately to the location the user is interested in. Finally, the user can download the part of the coverage visible in the map view in desired coverage format (GeoTIFF and CoverageJSON [25] being the formats available at the moment).

Data integration across borders is one of the main goals of the GeoE3 project. In case of coverage data, this is particularly interesting, as little work has previously been done in this area. Coverage integration on the GeoE3 integration platform is based on the experimental implementation of cross-collection query as defined in the OGC API Features specification. In this approach the query has a new 'collections' parameter, by which a list a relevant collection names can be given. All the other parameters are applied in all the coverages listed in the 'collections' parameter. Thus, if the bounding box of the query overlaps a national border and the countries on both sides of the query are included into the 'collections' parameter value, the result would be a grid, in which the both coverages are included. As an example, the following query to the GeoE3 DSM Coverages interface will produce the result shown in Fig. 6:

```
/geoe3/dsm/search?f=png&collec-
tions=DSM_NO,DSM_FI&subset=x(1225341:3876849),y(993
8802:11883241)&scale-size=x(1500),y(1100)
```

Production of the integrated image in Fig. 6 involves quite a lot of processing on the server side. The incoming query's bounding box, expressed in the Pseudo Mercator CRS (EPSG:3857) must first be transformed to the CRS of source data sets. When the requested grid is received from the GeoE3 platform's corresponding OGC API Coverages service, the grid must be reprojected to the target CRS, i.e., Pseudo Mercator. Once both countries' grids have been queried, they will be merged into a single result grid, considering the nodata areas present in the source grids. If the request is for a visible image, the grid needs to be masked for nodata areas and formatted in PNG. Each result image is also stretched to full radiometric scale for the best possible visual exploration.

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Figure 6. Response image for a cross-border (Finland-Norway) request for DSM utilising cross-collection query capabilities of the OGC API Coverages interface (experimental).

B. *Climate data*

Climate-related information was selected as one of the main target areas of the project, based on the requirements of the envisaged use cases. The most interesting climate parameters include sunshine hours, wind conditions and mean temperatures.

It was decided to approach the climate parameters as coverage data, although the source information is typically available as point values of weather observation stations. From the GeoE3 integration platform the climate data is thus available via an OGC API Coverages service endpoint. The first implementation, available currently only for the area of Finland, is based on the idea of cached grid data that is computed from the original observation points.

When a request for climate coverage data comes into the integration platform, the process will first check, whether the cached GeoTIFF file for the requested climate parameter exists on the platform. If the file is not found, a request is made to the OGC API Features service of the Finnish Meteorological Institute (FMI) for the corresponding point data. The data set is requested with a bounding box covering the whole country. From the received value point set, a gridded representation is derived using the SciPy library's [26] 'griddata' function. The process is first run with the method 'nearest' to cover the whole request area. Then the same is carried out with the method 'linear', which only covers the area spanned with the observation points. Subsequently, the two grids are combined, so that the grid points calculated with 'linear' take precedence. Finally, the grid is cropped with the boundary of the country to get the final result, which is then saved on the platform as a GeoTIFF file. The quality of the resulting grid is heavily dependent on the number and layout of the observation stations in the area. Best results have so far been achieved with temperature data.

When further queries come in, for instance as the user zooms in the map view, the cached GeoTIFF file is accessed to compute the result. This significantly speeds up processing of the queries. A functionality has been added to the Web user interface for displaying values of the visualized parameter when the mouse is hovered over the map. These queries are carried out as OGC API Coverages requests with zero-size bounding box and are performed on the cached GeoTIFF file on the server.

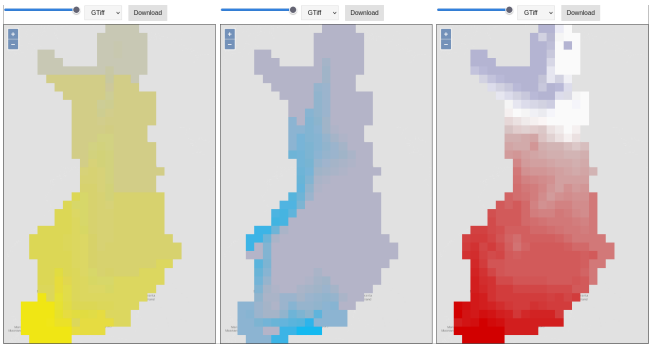

Figure 7. Climate coverages displayed in the GeoE3 Web user interface: sunshine hours (left), windspeed (middle) and mean temperature (right). All parameters are averages over 30 years.

The climate grid is visualized in the Web user interface with colored raster, requested from the OGC API Coverages service as a PNG image. Examples of the user interface are presented in Fig. 7. The real grid data can be downloaded from the viewed area in GeoTIFF and CoverageJSON formats.

VI. Preliminary 3D Work

The GeoE3 project introduces the idea of an HTML-formatted application-specific dashboard, especially in the context of building construction and use with renewable energy applications, such as solar energy potential. While computing the energy potential, many data sources and service interfaces can be utilized, depending on the computing tasks in hand. Using DTMs, DSMs and building footprints, one can automatically compute a 2D raster layer that indicates rooftops, suitable for solar energy production. This raster layer can then be portrayed together with the DSM and the orthophoto for 3D visual exploration, even without proper 3D models of the buildings.

In the exemplary visualization shown in Fig. 8, locations suitable for solar panels are presented with yellow colour. The visualization is a result of a parameterized on-the-fly process, in which the DSM slope and aspect values are used, together with the building footprints, and the height difference between the DSM and the DTM, to compute the areas well exposed to sun energy. Fig. 8 shows a three.js-based [27] Web application that requests the analysis result layer and the DSM from OGC-compatible service interfaces and drapes the raster on top of the DSM.

One of the goals of the GeoE3 project is to make use of 3D vector data. For instance, the project coordinator, National Land Survey of Finland, has made first 3D building models available from limited test areas. The production process is based on the use of the building footprints and DSM, and aims at constructing LOD2 category building geometries.

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

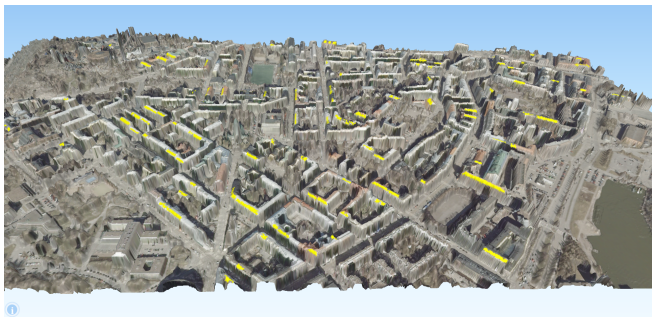

Figure 8. Visual exploration of potential solar panel locations on rooftops. Analysis is based on the use of DSM slope and aspect values, height differences between the DSM and the DTM, and building footprints.

A widely accepted, production-level service interface specification for 3D models does not exist yet. A possible workaround for this situation is to store the 3D geometries into a 3D City DB database [28] and export them as a CityGML [29] or CityJSON [30] -formatted file on demand.

At this moment, CityJSON format is the most interesting proposal for the 3D vector representation. CityJSON is a JSON-based subset of the OGC's CityGML data model and is designed for storage and transfer of 3D city models. CityJSON has been standardized by the OGC as an official OGC Community Standard in Oct 2021. Current version of the CityJSON standard is available at the CityJSON development site.

There are not yet freely available, browser-based applications for visualization of CityGML content. However, JSON processing is widely supported in browsers, and CityJSON can naturally benefit from this too [31]. There are development initiatives for providing an easy-to-use CityJSON-supporting software component for Web applications. On the left side in Fig. 9, an example is presented of browser-based visualization displaying NLS Finland buildings in CityJSON format utilizing a software component called Ninja [32]. On the right side in Fig. 9, the same features are presented using the 3D map view of the QGIS application [33], imported with the help of the QGIS plugin for CityJSON.

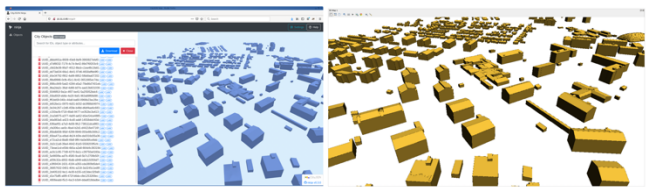

Figure 9. Visualization of 3D vector buildings in browser using Ninja (on the left) and in the 3D map view of the QGIS application (on the right).

VII. 3D Experiments

In the GeoE3 project, a substantial amount of work has been devoted on developing functionalities for 3D representation and visualization of buildings. Due to varying level of maturity and techniques applied in participating countries, a set of different approaches has been used to achieve a reasonably consistent provision of 3D buildings for all the countries.

In the case of Finland, Estonia and The Netherlands, pre-existing LOD2 category 3D models were available as downloadable files. They are encoded in various formats (database dump, CityGML, CityJSON) and were downloaded and imported into a database on the GeoE3 integration platform. In case of Norway, no pre-existing 3D data set could be found and thus an on-the-fly process for producing 3D models in real time was developed (described in Section III B). The case of Spain is different from the others as Spain already has a service endpoint available providing access to 3D representation of the country's buildings encoded in Keyhole Markup Language (KML). This KML-encoded 3D building model is produced by an on-the-fly process using floor plans as the source information. Consequently, the resulting models are in LOD1 category. A proxy service was developed on the GeoE3 integration platform to carry out the required KML to CityJSON transformation. A schematic illustration of the 3D buildings provision framework of the GeoE3 project is shown in Fig. 10.

As a final result of the development, 3D representation of buildings in all the participating countries is available on the GeoE3 platform, consistently encoded in CityJSON, and retrievable using the building's feature identifier as the key. The GeoE3 Web user interface currently utilizes a 3D visualization component 'ThreeJsViewer' developed by the University of Delft [34]. Some representative examples are shown in Fig. 11.

VIII. Conclusions and Future Work

GeoE3 project has as its main goal the development of a geospatially enabled ecosystem of use case-oriented services conforming to modern, second-generation interface standards. The project is tackling the challenge of cross-border and cross-domain integration of content employing a flexible, state-of-the-art integration platform, based on OGC API supporting Python library called pygeoapi. The plugin architecture of pygeoapi facilitates development of tailored components for connecting the platform to various different country level source services.

The overall service architecture of the GeoE3 project can be seen as a three-tier architecture with country level services forming the bottom data layer. The GeoE3 integration platform with advanced capabilities for dynamic content integration and use case-specific adaptation of the resulting data sets works as the middle processing layer. Finally, end user applications form the third architecture layer.

An application-specific output from an OGC API Features service can be provided in the form of an html-formatted dashboard of the feature in focus. Based on the user needs, an appropriate set of information items can be collected and put together by the GeoE3 platform, and the resulting data set be formatted in HTML as a shop window for potential users. The same data set could finally be requested as a JSON-encoded package, to be fed into external analysis processes.

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

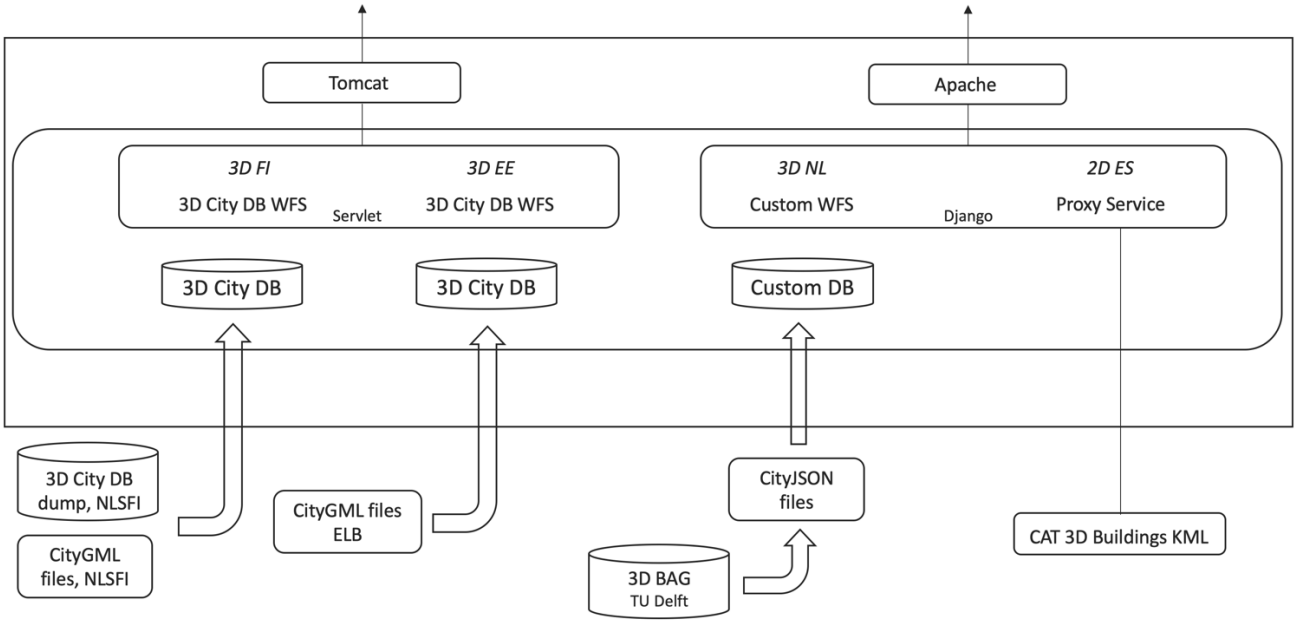

Figure 10. The 3D buildings data provision framework of the GeoE3 project. On-the-fly processes for creating LOD1 models are not included. (FI: Finland, EE: Estonia, NL: The Netherlands, ES: Spain, NLSFI: National Land Survey of Finland, ELB: Estonian Land Board, TU Delft: Delft University of Technology, CAT: General Directorate of Cadastre, Spain).

One of the main results of the GeoE3 work so far is the availability of 3D buildings from all the participating countries. The mechanisms behind the service interface vary a lot from country to country, but as the final end result 3D buildings are accessible by feature ID queries via a WFS conforming service in CityJSON encoding. A real-time process has been developed for generating LOD1 models where LOD2 building models are not available. The process is based on the building's 2D footprint and DTM and DSM coverages within the footprint and provides CityJSON encoded LOD1 models via an OGC API Processes -compliant service implementation.

First climate-related data sets have also become available on the GeoE3 integration platform. These provide 30-year averages of sunshine hours, wind speed and mean temperature as coverage type content from an OGC API Coverages service.

The project will continue developing the described functionalities over the coming one year and half and aims at wide adoption of the developed services by actors working in the identified use case domains.

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

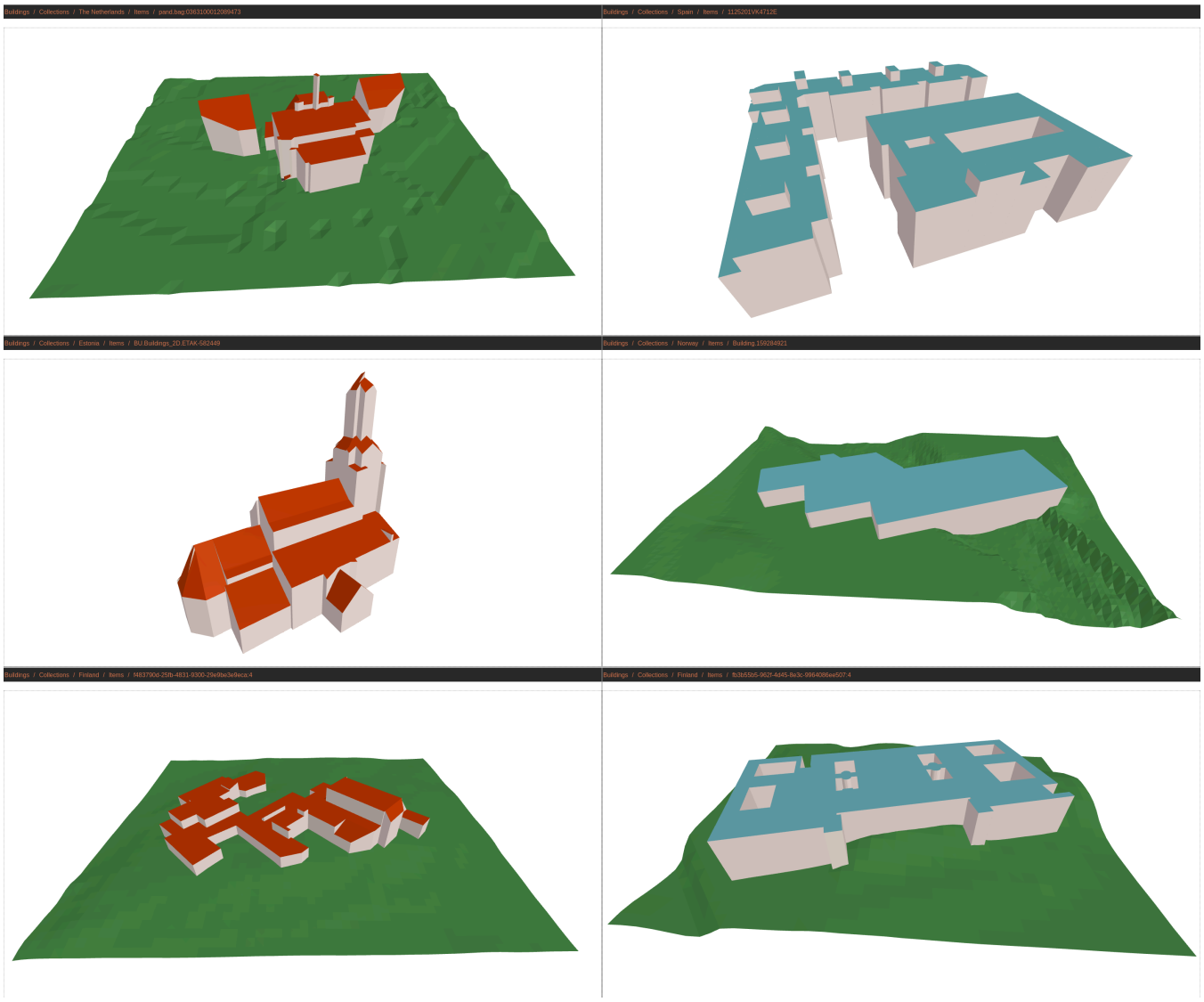

Figure 11. Examples of 3D buildings from GeoE3 participating countries. The Netherlands (top left), Spain (top right), Estonia (middle, left), Norway (middle right) and Finland (bottom). The buildings from Spain, Norway and Finland (bottom right) presented as dynamically created LOD1 models. DTM missing from Spain (building model not in absolute elevation) and Estonia (DTM data not yet available).

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

ACKNOWLEDGMENT

The GeoE3 project is co-financed by the Connecting Europe Facility (CEF) of the European Union with the grant agreement number INEA/CEF/ICT/A2019/2063390.

The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

REFERENCES

[1] L. Lehto, J. Kähkönen, J. Reini, T. Aarnio, and R. Tervo, Cross-border and Cross-domain Integration of Content in a European Geospatially Enabled Ecosystem. *GEOProcessing 2021, the Thirteenth International Conference on Advanced Geographic Information Systems, Applications, and Services*, Jul 18-22, 2021, Nice, France, pp. 1-5. ISBN: 978-1-61208-871-6

[2] CEF, Connecting Europe Facility, Home Page. [Online]. Available from: https://ec.europa.eu/inea/en/connecting-europe-facility, 2021, [retrieved: May, 2022]

[3] GeoE3, Geospatially Enabled Ecosystem for Europe, GeoE3 Home Page. [Online]. Available from: https://geoe3.eu, 2021, [retrieved: May, 2022]

[4] NLS, National Land Survey of Finland Home Page. [Online]. Available from: https://www.maanmittauslaitos.fi/en, 2021, [retrieved: May, 2022]

[5] United Nations, SDG Indicators, Goal 11. [Online]. Available from: https://unstats.un.org/sdgs/metadata/?Text=&Goal=&Target=11.3, 2021, [retrieved: May, 2022]

[6] OGC, OGC API Home Page. [Online]. Available from: https://ogcapi.ogc.org, 2021, [retrieved: May, 2022]

[7] OGC, OGC API – Features Home Page. [Online]. Available from: https://ogcapi.ogc.org/features/, 2022, [retrieved: May, 2022]

[8] OGC, OGC API – Coverages Home Page. [Online]. Available from: https://ogcapi.ogc.org/coverages/, 2022, [retrieved: May, 2022]

[9] OGC, OGC API – Records Home Page. [Online]. Available from: https://ogcapi.ogc.org/records/, 2022, [retrieved: May, 2022]

[10] OGC, OGC API – Processes Home Page. [Online]. Available from: https://ogcapi.ogc.org/processes/, 2022, [retrieved: May, 2022]

[11] OGC, OGC API – Joins, Draft. [Online]. Available from: https://github.com/opengeospatial/tjs/blob/master/drafts/OGC%20API%20-%20Joins%20-%20Part%201%20Core.pdf, 2022, [retrieved: May, 2022]

[12] OGC, OGC API – Environmental Data Retrieval (EDR) Home Page. [Online]. Available from https://ogcapi.ogc.org/edr/, 2022, [retrieved: May, 2022]

[13]] CEF, CEF Digital - eTranslation. [Online]. Available from: https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation, 2022, [retrieved: May, 2022]

[14] Django, Django Home Page. [Online]. Available from https://www.djangoproject.com, 2022, [retrieved: May, 2022]

[15] T. Kralidis, pygeoapi Home Page. [Online]. Available from: https://pygeoapi.io, 2021, [retrieved: May, 2022]

[16] OpenLayers, OpenLayers Home Page. [Online]. Available from: https://openlayers.org, 2022, [retrieved: May, 2022]

[17] GeoJSON, GeoJSON Home Page. [Online]. Available from: https://geojson.org, 2021, [retrieved: May, 2022]

[18] GDAL, GDAL/OGR Home Page. [Online]. Available from: https://gdal.org, 2021, [retrieved: May, 2022]

[19] Mapbox, rasterio Home Page. [Online]. Available from: https://rasterio.readthedocs.io/en/latest/, 2021, [retrieved: May, 2022]

[20] INSPIRE, INSPIRE UML-to-GeoJSON encoding rule. [Online]. Available from: https://github.com/INSPIRE-MIF/2017.2/blob/master/GeoJSON/geojson-encoding-rule.md, 2019, [retrieved: May, 2022]

[21] INSPIRE-MIF, INSPIRE Alternative Encoding Model Simplification Rules. [Online]. Available from: https://github.com/INSPIRE-MIF/2017.2/blob/master/model-transformations/TransformationRules.md, 2020, [retrieved: May, 2022]

[22] M. Nordberg, PxWeb API description. [Online] Available from: https://pxnet2.stat.fi/API-description_SCB.pdf, 2020, [retrieved: May, 2022]

[23] L. Lehto and J. Kähkönen, 2021. OGC API Features HTML-output as a Feature Dashboard, *Abstr. Int. Cartogr. Assoc.*, 3, 176, https://doi.org/10.5194/ica-abs-3-176-2021, 2021.

[24] WhiteBoxTools, WhiteBoxTools Home Page. [Online]. Available from: https://www.whiteboxgeo.com/geospatial-software/, 2022, [retrieved: May, 2022]

[25] CoverageJSON, CoverageJSON Home Page. [Online]. Available from: https://covjson.org, 2022, [retrieved: May, 2022]

[26] SciPy, SciPy Home Page. [Online]. Available from: https://scipy.org, 2021, [retrieved: May, 2022]

[27] three.js, three.js Home Page. [Online]. Available from: https://threejs.org, 2021, [retrieved: May, 2022]

[28] 3DCityDB, 3D City DB Home Page. [Online]. Available from: https://www.3dcitydb.org/3dcitydb/, 2022, [retrieved: May, 2022]

[29] OGC, CityGML Home Page. [Online]. Available from: https://www.ogc.org/standards/citygml, 2012, [retrieved: May, 2022]

[30] CityJSON, CityJSON Home Page. [Online]. Available from: https://www.cityjson.org, 2021, [retrieved: May, 2022]

[31] J. Virtanen, et al., "Near real-time semantic view analysis of 3D city models in web browser", *ISPRS International Journal of Geo-Information*, vol. 10, no. 3, pp. 1-23, https://www.mdpi.com/2220-9964/10/3/138, 2021

[32] S. Vitalis et al. CITYJSON + WEB = NINJA, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, VI-4/W1-2020, pp. 167–173, https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-167-2020, 2020.

[33] QGIS, QGIS Home Page. [Online]. Available from: https://www.qgis.org/en/site/, 2021, [retrieved: May, 2022]

[34] TUDelft, TU Delft 3D Geoinformation Home Page. [Online]. Available from: https://3d.bk.tudelft.nl, 2022, [retrieved: May, 2022]

2022, © Copyright by authors, Published under agreement with IARIA - www.iaria.org