

GeoE3 Service Architecture



Co-financed by the Connecting Europe
Facility of the European Union

2021-11-05



CONFIDENTIAL DRAFT – NOT FOR FURTHER CIRCULATION

Revision History

Version	Status	Author	Modifications
1.0	Updated	Mikko Visa	Note

Distribution

Version	Status	Recipient

Table of Contents

Revision History.....	2
Distribution.....	2
Table of Contents.....	3
Figures.....	4
1. Introduction	4
2. Service functionalities	5
2.1. Cross-border implementation	5
2.2. Data service: OGC API Features.....	5
2.3. Data Service: Environmental Data Retrieval	6
2.4. Data Service: OGC API Coverages	6
2.5. Data Service: OGC API Processes.....	6
2.6. Data Catalog Service: CS-W or OGC API Records (optional).....	7
2.7. Supporting Service: Georeferenced Table Joining Service.....	7
2.8. Supporting Service: Automated data quality validation	8
2.9. Supporting Service: Automated data and service quality validation	8
2.10. Supporting Service: Traffic data	9
2.11. Pictures.....	10

1. Introduction

This document describes the envisaged service architecture for the ecosystem of services being built in the GeoE3 (Geospatially Enabled Ecosystem for Europe) project. The architecture aims at a modular structure, relying on OGC API family of services. While data are provided from numerous distributed services, the client is provided with a single point of access. The service architecture is presented in figure 1.

Metadata is provided from the European Data Portal catalog service, retrieved via CS-W from national sources. Optionally an OGC API Records service might be tested as an early adaptor for demonstrational purposes.

In case of non-open data, the authentication is done with a separate APIKEY authenticator. The requests are authenticated both in European cross border implementation and in data services. Ideally this should be avoided, however.

Where applicable, the services should support a change-only data provision based on, for example, INSPIRE lifespan tags. However, such functionality is not reasonable for example with constantly updating meteorological data.

2. Service functionalities

2.1. Cross-border implementation

Cross-border implementation works as a single point of access for the clients. It works as a proxy and cascades the requests to the backend services based on the requested data and area. The target is to provide only a single service for each data theme. The selected approach is to treat datasets provided by the participating countries as data collections inside a single OGC API service endpoint. The so-called cross-collection queries are to be supported, too. This would enable the clients to see a given GeoE3-supported content theme as a single seamless data resource. The necessary schema transformations are carried out by the Cross-border implementation.

Provided interface:	OGC API Features OGC EDR (Environmental Data Retrieval) OGC API Coverages (optional)
Dependencies:	Underlying services
Provided data:	Assumption is all data in the project.

2.2. Data service: OGC API Features

Provides access to the underlying data. Ideally, provide data directly from the database / data storage, but may also work as a wrapper to legacy services, such as WFS 1.X or WFS 2.0. All services shall support GeoJSON conformance class. Filtering on dataset properties is not required in the project. Required CRS support is to be decided during the project.

Provided interface:	OGC API Features
Dependencies:	Data bases / data storage Legacy services
Provided data:	Assumption is all data in the project.
Standard:	https://ogcapi.ogc.org/features/

2.3. Data Service: Environmental Data Retrieval

Provides access to the meteorological data. Essentially, the service extracts data from a multidimensional data cube to the requested point / trajectory / corridor / area.

Provided interface:	OGC EDR (item Location Area Trajectory)
Dependencies:	Data storage
Provided data:	Meteorological data.
Standard:	https://ogcapi.ogc.org/edr/

2.4. Data Service: OGC API Coverages

Provides DEM and other gridded data from data storage. Ideally, provide data directly from the database / data storage, but may also work as a wrapper from legacy services, such as WCS 1.X or WCS 2.0.1. GeoTIFF and CoverageJSON are to be supported as data formats.

Provided interface:	OGC API Coverages
Dependencies:	Data storage
Provided data:	DEM and possibly other gridded data as necessary.
Standard:	https://ogcapi.ogc.org/coverages/

2.5. Data Service: OGC API Processes

The service provides a standard API to carry out on-demand calculations (such as “time in daylight”). The service initiates the process, which might employ for instance existing OGC API services. In case of synchronous requests, the result is responded straight away from the service. In case of asynchronous processing, the results are saved to a temporary datastore and served via a separate data service.

Provided interface:	OGC API Processes
Dependencies:	Data services Internal Data Storage Processing platform

Provided data: TBD during project as necessary.

Standard: <https://ogcapi.ogc.org/processes/>

2.6. Data Catalog Service: CS-W or OGC API Records (optional)

Provides required metadata. The metadata are pushed to the service by data providers or, where applicable, the data are synchronized from existing catalogs.

The metadata is synchronized also to the European data portal. Ideally the data portal synchronizes the catalog from an CS-W or OGC API Records service. If the portal is not capable to fetch the data from the OGC API Records service, either DCAT Application profile extension need to be developed to the OGC API Records service, or a separate DCAT service must be established and the metadata be updated to two separate services.

The clients may consume the metadata from European data portal via user interface or CSW API or from OGC API Records via API.

The metadata records are translated, where applicable, to European languages employing the eTranslation service.

Provided interface: CS-W or OGC API Records

Dependencies: Data storage
Legacy services

Provided data: Metadata

Standard: <https://www.ogc.org/standards/cat>
<http://docs.ogc.org/DRAFTS/20-004.html>

2.7. Supporting Service: Georeferenced Table Joining Service

Georeferenced Table Joining Service (to be called OGC API Joins) is used to combine statistical data with georeferenced location data. The data may be uploaded during the request or fetched from OGC API Features, Environmental Data Retrieval and statistical services, such as PX Web API. The results are published via a separate OGC API Service. TJS service requires a dedicated internal database for its operations and storing the intermediate results.

Provided interface:	-
Dependencies:	Data Services OGC API Records Internal database
Provided data:	TBD during project as necessary.
Standard:	https://portal.ogc.org/files/?artifact_id=40095

2.8. Supporting Service: Automated data quality validation

Automated data quality validation service fetches the data from the cross-border implementation/context broker and evaluates it based on a set of predefined rules. The result is then updated to the metadata into the OGC API Records service.

Provided interface:	-
Dependencies:	Data Services OGC API Records Cross-border implementation / Context broker
Provided data:	TBD
Standard:	-

2.9. Supporting Service: Automated data and service quality validation

Shows data and service quality via UI. Support for OGC API family to be built in Spatineo Monitor service during the project.

Provided interface:	-
Dependencies:	Automated Data Quality Validation
	Service Validation
	OGC API Records
Provided data:	-
Standard:	-

2.10. Supporting Service: Traffic data

Traffic data should be available to the cross-border implementation. However not available via OGC API's but rather directly from different services (FI: Digitraffic).

Provided interface:	TBD
Dependencies:	-
Provided data:	TBD
Standard:	DATEXII

2.11. Pictures

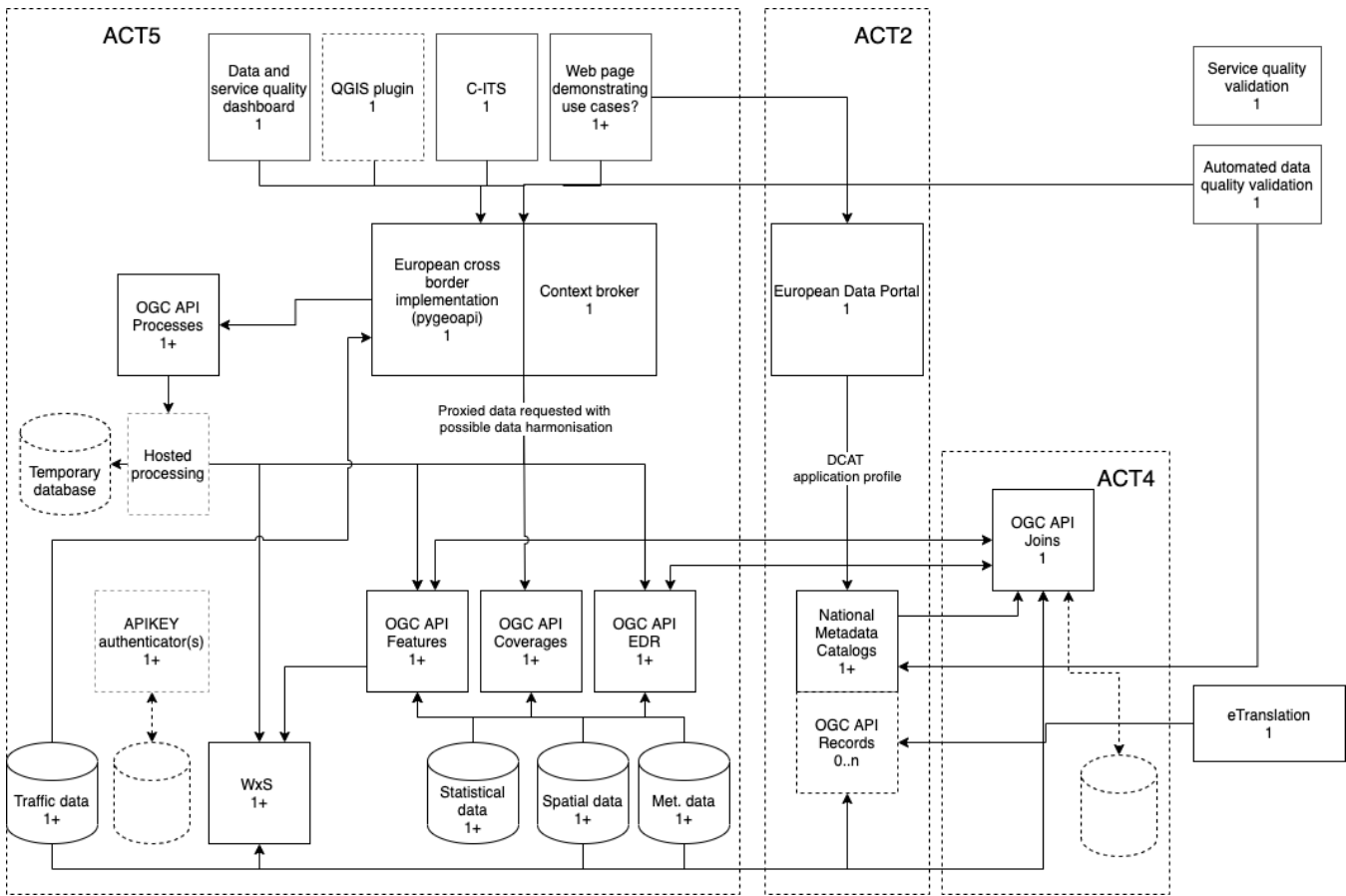


Figure 1: Service architecture