



API4INSPIRE

D2 - Deployment strategies for standard-based APIs



Contents

Tables	4
Figures	5
1. Definitions and Acronyms	7
1.1. Acronyms.....	7
1.2. Definitions	8
1.3. Standards.....	8
1.4. Supporting Documents.....	9
2. Introduction.....	11
3. Data Providers	12
3.1. Selection Criteria	12
3.2. Data Nests	13
3.2.1. Airy Austria	13
3.2.2. Urban Data Platform Hamburg	14
3.2.3. Franco-Germanic Flow	15
4. Deployment Methodology	16
4.1. GeoServer Deployment	18
4.1.1. OGC API – Features in GeoServer	19
4.1.2. App-Schema.....	20
4.1.3. New guidelines for modern standards	21
4.1.4. Preconfigured GeoServer Black Boxes in Docker	23
4.2. OGC API – Features Development - OGC API Simple	24
4.3. LD-Proxy	26
4.4. FROST Deployment.....	29

D2 - Deployment strategies for standard-based APIs

4.4.1.	FROST-Maintained Database.....	30
4.4.2.	Custom Database.....	30
4.4.3.	Deployment Options	30
5.	Data Provider Specific Deployment.....	32
5.1.	Austrian Meteorological Agency – ZAMG (AT).....	32
5.2.	Austro Control – ACG (AT).....	32
5.3.	Umweltbundesamt - UBA (AT) & European Environment Agency - EEA (EU)	33
5.4.	City of Hamburg – CH (DE)	34
5.5.	French Geological Survey – BRGM (FR).....	35
5.6.	French Office for Biodiversity (OFB) – INSIDE - environmental information systems research center (FR)	36
5.7.	Environment Agency Baden-Württemberg – LUBW (DE)	37
5.8.	Summary Overview	38
6.	Conclusions.....	39
7.	Next Steps.....	40
8.	Annex 2.A Mapping INSPIRE AQD to SensorThings API	41
8.1.	Austrian Inspire Air Quality Data.....	41
8.2.	EEA Air Quality Data	47

Tables

Table 1: Overview Data Providers and Datasets	38
Table 2: STA Datastream Mapping Austrian Air Quality	42
Table 3: STA Sensor Mapping Austrian Air Quality	43

D2 - Deployment strategies for standard-based APIs

Table 4: STA FoI Mapping Austrian Air Quality	44
Table 5: STA Thing/Location Mapping Austrian Air Quality	45
Table 6: STA Observation Mapping Austrian Air Quality	45
Table 7: STA Observation Mapping EEA Air Quality	47
Table 8: STA Observation Metadata Mapping EEA Air Quality	48

Figures

Figure 1: Deployment Methodology	16
Figure 2: GeoServer New Smart App-Schema Data Source	22
Figure 3: OGC API - Simple Interface	25
Figure 4: LD-Proxy - add new service	26
Figure 5: LD-Proxy - show feature types	27
Figure 6: LD-Proxy - feature attributes	28
Figure 7: LD-Proxy - schema mapping	28
Figure 8: LD-Proxy - TN-A Data	29

D2 - Deployment strategies for standard-based APIs

1. Definitions and Acronyms

1.1. Acronyms

API: Application Programming Interface

CSV: Comma Separated Values

DB: Database

desktop GIS: Geographic Information System running on a desktop computer

DWBP: W3C Data on the Web Best Practices

ELISE: European Location Interoperability Solutions for e-Government

FROST: Fraunhofer Opensource SensorThings-Server

GeoJSON: An open standard format designed for representing geographical features.

GIS: Geographic Information System

HTML: HyperText Markup Language

INSPIRE: Infrastructure for Spatial Information in the European Community

INSPIRE MIF: INSPIRE maintenance and implementation framework

INSPIRE MIG: INSPIRE maintenance and implementation group

JSON: JavaScript Object Notation

MS: European Union Member State

OGC: Open Geospatial Consortium

OGD: Open Government Data

OWS: OGC Web Services

REST: Representational state transfer

SF-X: Simple Features Level X

SOAP: Simple Object Access Protocol

D2 - Deployment strategies for standard-based APIs

SDK: Software Development Kit

SDWBP: W3C Spatial Data on the Web Best Practices

STA: SensorThings API

SWG (OGC): Standards Working Group

W3C: World Wide Web Consortium

WFS2: Web Feature Service version 2

XML: Extensible Markup Language

1.2. Definitions

Administrative Stakeholders: Those responsible for providing resources for the provision and use of APIs

Data Providers: Organisations that make data available

Data Users: People or organisations that use data made available by data providers

Degrees of Freedom: directions in which recommendations can lead to concrete impacts and improvements

Evaluation Criteria: individual criteria derived from the Five Level Evaluation Model by Moilanen

Evaluation Types: methods of interrogating stakeholders: Heuristic Expert Evaluation, Peer Review, Interview, Questionnaire

Five Level Evaluation Model: API evaluation model proposed by Jarkko Moilanen and extended in D1

Peer Review: evaluation approach for API development evaluation defined by Farooq and described in D1

Stakeholder Perspectives: different stakeholder roles: development, deployment, usage

1.3. Standards

The following standards are referenced throughout this document:

- OGC® 07-036r1: OpenGIS® Geography Markup Language (GML) Encoding Standard
- OGC® 09-025r2: OGC® Web Feature Service 2.0 Interface Standard – With Corrigendum
- OGC® 10-100r3: Geography Markup Language (GML) simple features profile (with Corrigendum) (2.0) - profile

D2 - Deployment strategies for standard-based APIs

- OGC® 12-006: OGC® Sensor Observation Service Interface Standard
- OGC® 14-055r2: OGC OWS Context GeoJSON Encoding Standard
- OGC® 15-078r6: OGC SensorThings API Part 1: Sensing v1.0
- OGC® 16-032r2: OGC WaterML 2: Part 4 – GroundWaterML 2 (GWML2)
- OGC® 17-069r3: OGC API - Features - Part 1: Core
- OGC® 18-088r1: OGC SensorThings API Part 1: Sensing v1.1 (in public review)

1.4. Supporting Documents

- Good Practice: INSPIRE download services based on OGC API - Features: <https://github.com/INSPIRE-MIF/gp-ogc-api-features>
- OGC Web API Guidelines: <https://github.com/opengeospatial/OGC-Web-API-Guidelines>
- OGC API Hackathon: <https://www.opengeospatial.org/projects/initiatives/oapihackathon19>
- GDAL Support for OGC API - Features: <https://gdal.org/drivers/vector/oapif.html>
- Simple Features GeoJSON Java: <https://github.com/ngageoint/simple-features-geojson-java>
- W3C Data on the Web Best Practices: <https://www.w3.org/TR/dwbp/>
- W3C Spatial Data on the Web Best Practices: <https://www.w3.org/TR/sdw-bp/>
- W3C Semantic Sensor Network Ontology: <https://www.w3.org/TR/vocab-ssn/>

D2 - Deployment strategies for standard-based APIs

2. Introduction

This deliverable reports on the results of Task 2 specified in the Technical Specification of the project. The project partners together with the data providers have researched deployment strategies of the chosen OGC APIs to be evaluated as described in detail in the previous Deliverable 1 “Methodology for evaluation of standard based APIs”. Fraunhofer IOSB together with DataCove e.U. have developed a methodology for the successful deployment of new standards, whereby exemplary use cases guide the process of data mapping and query optimization until the required functionality is met. We built on this successful strategy, making minor modifications as required.

Different strategies were required depending on the maturity of the individual data providers. In some cases, fully INSPIRE compliant WFS2 services have long been deployed, so the endpoint need only be upgraded to OGC API - Features; should project resources suffice, alternative options for provision of OGC API - Features based on existing WFS2 deployments (i.e. LD-Proxy) were explored in order to determine if functionality and performance criteria could be considered equivalent. In other cases, while there are some basic OWS services in operation, these may not be fully aligned with the requirements ensuing from INSPIRE and the existing mapping may need to be extended and the configuration adjusted accordingly if the mapping differences are significant. On the other end of the spectrum, data sources were leveraged, which has not yet been available. In these cases, a basic harmonization process was performed, focusing on making the data available through the new APIs first, and implementing INSPIRE requirements where it makes sense.

For the steps #5 to #7, commonly utilized open source software has been applied, enabling a smooth transition to the new APIs and wide-scale uptake. Specifically: GeoServer as OGC API - Features implementation, and FROST with the STA 1.1 extension as required INSPIRE as SensorThings API implementation.

3. Data Providers

During the preparatory stage of the API4INSPIRE project, a set of data providers was selected to provide interesting data sources for experimentation and evaluation of the new APIs as well as to provide feedback on both on the effort required for provision of the new APIs as well as for insights as to the usability of these services.

3.1. Selection Criteria

The following criteria guided our selection process:

- Co-location of complementary datasets enabling Use Cases integrating multiple sources across data providers and Member States
- Balance of different INSPIRE Themes and data models
- Balance between OGC API - Features and OGC SensorThings API
- Wide range of development levels at the different data providers

The core criterion for identifying potential data providers lay in the suitability of the available data for implementation of realistic use cases. Co-location of datasets spatial scope was an essential criterion to ensure that data sources can be evaluated not only by themselves, but also pertaining to how well they can be integrated with data from other API sources.

Pertaining to the balance between types of data being provided as well as API specifications being utilized, care has been taken to complement highly dynamic sources typically provided via OGC SensorThings API with more static spatial sources provided via OGC API - Features. For selected data sources, the same data holdings will be exposed via alternate technologies, providing users with the possibility of direct comparison of API functionality.

In order to assure that the evaluation being performed pertains to a wide spectrum of potential data providers, data providers were also selected to reflect the wide range of stakeholder types described within the stakeholder analysis performed within “D1 - Methodology for evaluation of standard based APIs”. The usage types detailed in section “3.1.3 Usage Spectrum” not only illustrate the status quo pertaining to data usage, they also provide the starting point upon which the APIs will be deployed. Data providers have been selected to reflect the following existing usage and access types:

- Direct access to data source (DB)
- File-based data Sources
- Access to previous SOA+XML style OGC Web Services (OWS) with simple Features
- Access to previous SOA+XML style OGC Web Services (OWS) with complex Features
- Access to (non-standard) REST-Based APIs

3.2. Data Nests

As mentioned above, care was taken to identify complementary sets of colocated data, allowing users to integrate the data made available from the APIs provided for the creation of applications. Based on this requirement together with the other selection criteria described, six different data providers across Europe maintaining relevant data sources were contacted and cooperation agreements reached. These agreements allow us to provide the following Data Nests:

3.2.1. Airy Austria

In Austria, we have two quite advanced data providers integrated within API4INSPIRE, providing both spatial data via various OGC service types as well as dynamic measurement data, allowing users to integrate over both API types. In addition, air quality data has been added in an ad-hoc manner, initially from the Austrian Environment Agency (Umweltbundesamt), but then rapidly extended across central Europa via access to data from the European Environment Agency.

- **Austro Control (ACG)**, responsible for air traffic control within Austria, has long had nearly all datasets relevant for the INSPIRE Theme Transport Networks - Air online via WFS in strict accordance with the INSPIRE data specifications. In addition, they have defined a solid URI based identifier scheme, allowing for direct resolution of the identifier URI to the underlying data object. Within API4INSPIRE, an OGC API endpoint will be configured in parallel to the WFS endpoint; additionally, the stand-alone OGC API - Features implementation for provision of simple features corresponding to SF-0 will be deployed on this data source. Thus, participants will be able to experiment with diverse provision options and evaluate the strengths and weaknesses of each of the three different systems serving the same dataset.
- **Zentralanstalt für Meteorologie und Geodynamik (ZAMG)**, the Austrian agency for meteorology and geodynamics is responsible for the continuous measurement of meteorological parameters, as well as the provision of this data to the World Meteorological Organization (WMO). They are currently finalizing services for INSPIRE compliant WFS provision of this data, and will add an additional OGC API - Features endpoint to this configuration. In addition, the same dataset will also be provided in parallel via SensorThings API, underscoring the strengths of sensor centric services for the provision of dynamic measurement data.
- **Ad-hoc Austrian Air-Quality API** - while not formally planned, this additional source has been triggered by the Corona virus outbreak and the relationship with air quality. As near-real-time Austrian air-quality data was already available in accordance with INSPIRE and European reporting requirements via existing OGC WFS and SOS services, it was a fairly simple task to harvest these endpoints, transform the data as required and insert into a SensorThings API instance. This endpoint is being dynamically extended as additional data sources become available from sources such as the European Environment Agency. Details are available at <http://datacove.eu/ad-hoc-air-quality/>

D2 - Deployment strategies for standard-based APIs

Potential Use Cases in this context pertain to linking locations from air transport networks with data stemming from meteorological and air quality monitoring stations.

3.2.2. Urban Data Platform Hamburg

The Urban Data Platform of Hamburg Germany is supporting us through provision of their Smart City Sensors - deployed on SensorThings API. These range from car charging stations over the bike stations of “StadtRad” to various data of the Energy Campus of Hamburg University of Applied Science. A small selection:

Charging Stations:

[https://iot.hamburg.de/v1.0/Things?\\$filter=substringof\(%27Lade%27,name\)&\\$expand=Locations,Datastreams/Observations\(\\$orderby=phenomenonTime%20desc;\\$top=1\)&\\$count=true](https://iot.hamburg.de/v1.0/Things?$filter=substringof(%27Lade%27,name)&$expand=Locations,Datastreams/Observations($orderby=phenomenonTime%20desc;$top=1)&$count=true)

StadtRad Locations and available Bikes:

[https://iot.hamburg.de/v1.0/Things?\\$filter=properties/serviceName%20eq%20%27STA%20StadtRad%27&\\$expand=Locations\(\\$select=location\),Datastreams\(\\$expand=Observations\(\\$select=phenomenonTime,result;\\$orderby=phenomenonTime%20desc;\\$top=1\);\\$filter=properties/type%20eq%20%27eq%202\)&\\$count=true](https://iot.hamburg.de/v1.0/Things?$filter=properties/serviceName%20eq%20%27STA%20StadtRad%27&$expand=Locations($select=location),Datastreams($expand=Observations($select=phenomenonTime,result;$orderby=phenomenonTime%20desc;$top=1);$filter=properties/type%20eq%20%27eq%202)&$count=true)

Data from the Energy Campus:

[https://iot.hamburg.de/v1.0/Things?\\$filter=name%20eq%20%27Energie%20Campus%20Hamburg%20University%20of%20Applied%20Sciences%27&\\$expand=Locations,Datastreams](https://iot.hamburg.de/v1.0/Things?$filter=name%20eq%20%27Energie%20Campus%20Hamburg%20University%20of%20Applied%20Sciences%27&$expand=Locations,Datastreams)

In addition, the City of Hamburg maintains over 400 spatial datasets, many already available online in various formats. As the Smart City Sensors would be very interesting in combination with routing information (and while INSPIRE TN-RO is available, there are no routing algorithms to date), we propose to complement the APIs provided with data stemming from OpenStreetMap (OSM)¹.

Conclusion: Lots of data on the bottom, cool applications on the top, and some juicy APIs in between.

Potential Use Cases in this context pertain to Smart City applications, e.g. providing routing information to the nearest charging station. In addition, users can experiment with concepts stemming from building management systems utilizing the data from the Energy Campus.

¹ <https://ec.europa.eu/jrc/en/publication/comparing-inspire-and-openstreetmap-data-how-make-most-out-two-worlds>

3.2.3. Franco-Germanic Flow

For our third data nest, we've pulled together data from both the German and French sides of the Rhine River, integrating data from the German Bundesland of Baden-Württemberg (LUBW) with that stemming from the French Geological Survey (BRGM) and the French Office for Biodiversity (OFB) (via its environmental information systems research center - INSIDE). This data covers spatial sources ranging from the basic river network information covered by the INSPIRE Theme Hydrography with additional river features supplied by the INSPIRE Theme Transport Networks - Water over water measurement stations provided in accordance with the INSPIRE Theme Environmental Monitoring Facilities to known flood risk zones provided under the INSPIRE Theme Natural Risk Zones, that will be exposed via OGC API - Features. This data will be complemented by dynamic data provided via SensorThings pertaining to both water quality and quantity.

Groundwater data will also be provided in the form of Hydrogeological units, their monitoring facilities served by OGC API - Features and their associated raw quantity observations provided by SensorThings API.

One interesting aspect of this combination of datasets will pertain to the overlaps in data maintained by different MS, with the French River Networks and Aquifers extending into Germany. A further duplication will be created pertaining to Environmental Monitoring Facilities, as we aim to provide these via both the OGC API - Features as well as SensorThings API; as far as possible these parallel datasets will provide cross references.

Potential Use Cases in this context pertain mostly to flooding within the Rhine catchment area, including both surface and ground water, but can also extend to navigability of French water transport networks based on water levels.

4. Deployment Methodology

Fraunhofer IOSB together with DataCove have developed a seven-step methodology for the successful deployment of new standards:

1. Definition of exemplary use cases
2. Mapping of required data to the data model of the standard
3. Derive queries from use cases aligned with query functionality of the API
4. Define expected performance of these queries
5. Either convert and import the data into a server/database which implements the API or create the necessary configuration files for accessing existing data sources
 - a. If required, configure the service to the data model to be provided
6. Execute and measure the specified queries
7. Optimize queries where necessary

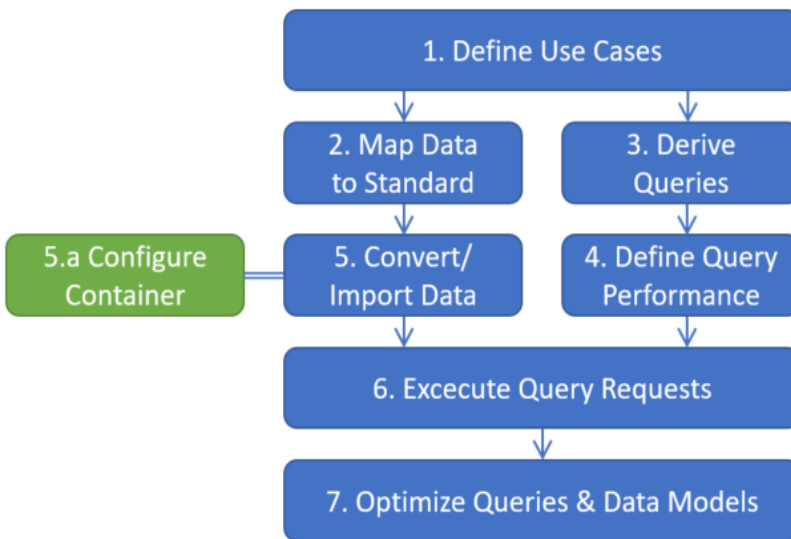


Figure 1: Deployment Methodology

While this methodology was initially defined for the deployment and benchmarking of SensorThings API, only minor modifications are required to extend this successful methodology towards provision of OGC API - Features. Modifications of this strategy will be required depending on the maturity of the individual data providers. The following list shows the current state-of-play among our data providers, as well as the deployment strategy selected:

D2 - Deployment strategies for standard-based APIs

1. Fully INSPIRE compliant services have been deployed [ZAMG, ACG, UBA]
 - a. GeoServer in cooperating data providers: In this case, the existing deployment can be maintained, only the OGC API - Features extension needs to be added to the GeoServer Deployment.
 - b. Deegree or other servers: for sources where we cannot easily add an additional OGC API - Features endpoint, conversion tools such as Id-proxy will be utilized.
 - c. For the Austrian Air Quality service, we selected an alternative route, harvesting the available data from both the WFS2 and SOS2 endpoints and inserting into a SensorThings Server.
2. Spatial data is available within the data providers organization, but not or only partially exposed in a non-standardized manner. For this purpose, we have created reusable theme specific configurations for GeoServer in the form of preconfigured docker images. In these cases, a basic harmonization process will be performed, focusing on making the data available through the new APIs first, and implementing INSPIRE requirements where it makes sense [BRGM, OFB]
 - a. Where possible, GeoServer with the OGC API - Features extension will be deployed within the data providers infrastructure.
 - b. Where this is not possible such deployments will be hosted within the project
3. SensorThings endpoints are available, but not fully aligned with INSPIRE requirements [BRGM, CH]
 - a. Where possible, the additional data elements required by INSPIRE will be added to the SensorThings deployments in cooperation with data providers staff.
 - b. Should provision of full INSPIRE requirements not be possible, these endpoints will still be included, whereby these will allow a comparison of usability between base SensorThings vs. SensorThings including the INSPIRE extensions.
4. Dynamic data is available within the data providers organization, but not or only partially exposed in a non-standardized manner [BRGM, OFB, LUBW, EEA]
 - a. Where possible, the FROST SensorThings implementation will be deployed within the data providers infrastructure.
 - b. Where this is not possible such deployments will be hosted within the project

In addition, a standalone OGC API - Features implementation, only supporting provision of simple features in accordance with the OGC SF-0 profile, is being developed within this project, providing insights into difficulties encountered by developers not familiar with the various underlying standards. This implementation will be deployed on selected datasets in addition to the options described above, allowing users to experiment with the strengths and weaknesses of such a simplified system.

The extent to which individual data provider staff will be integrated in the deployment work will be strongly dependent on the level of expertise available at that data provider. In the case of highly technical data providers such as Austro Control or BRGM, we will be working hand-in-hand with data provider staff on upgrading their existing deployments to the new APIs. In addition, staff from these institutions will participate in the heuristic expert evaluation as well as the peer reviews. For other data providers such as the French Biodiversity Office OFB, far more support will be required; in those cases we will need to provide at least initial configurations together with documentation on how to extend this core

D2 - Deployment strategies for standard-based APIs

configuration, providing immediate feedback on the quality of the training and guidance materials being provided under D5.

Initial data analysis and mapping will be done on systems of IOSB or GeoSolutions, while the final deployment used for evaluation is done on the systems of the data provider. If the data provider cannot provide infrastructure to deploy the services required in this project, the services will be deployed on a Kubernetes cluster at Fraunhofer IOSB and maintained at least one year past contract finalization.

Although the actual deployments in the project will be done using FROST-Server and GeoServer, much of the deployment methodology, like the data mapping procedures and container deployment technology, is not implementation specific. The evaluation will foremost focus on these general aspects, implementation specific aspects will be clearly flagged as such.

The technologies listed here are described in greater detail in the sections below.

4.1. GeoServer Deployment

GeoServer is a free and open source project designed for geospatial data interoperability and is an implementation of several OGC standards such as WFS, WMS, WCS and more recently the OGC API, which includes OGC API - Features. GeoServer is built on top of the GeoTools library and integrates natively with GeoWebCache, providing a flexible and easy to use tile cache mechanism. GeoWebCache implements several standards including the WMTS service.

In GeoServer, modularity and flexibility are first class citizens, with the default package caring for most common needs and extensions supporting a variety of additional functionality. The former includes the most common data stores (e.g. Shapefile, GeoPackage, PostgreSQL and GeoTIFF) and OGC services (e.g. WFS, WMS and WMTS).

GeoServer offers a large amount of extension points allowing anyone to add new functionality in a modular way. Some of those extensions are contributed back to the GeoServer Community, becoming official GeoServer plugins: at present around eighty plug-ins are available for GeoServer. These plug-ins extend GeoServer in a variety of ways: adding support for new data sources, adding new services or extending existing ones, adding new security methods, and so on.

Plug-ins can be divided into two main categories: community modules and extensions. Community modules are generally considered experimental in nature and can be undergoing significant development. Once a community module development is stable and has proven to be useful it can be promoted to extension. To become an extension a community module must have an official maintainer, pass the code quality requirements and have official documentation. If an extension is judged to be important enough or is commonly used, it can be promoted to a GeoServer core module.

Three GeoServer plugins will be particularly relevant for this project: INSPIRE extension, OGC API community module and App-Schema extension. The INSPIRE extension allows GeoServer to be compliant

D2 - Deployment strategies for standard-based APIs

with a number of INSPIRE Services, for example, the View Service and Download Service. When this extension is present, new UI elements allowing the configuration of extra metadata and INSPIRE specific options are available; relevant OGC services outputs will also be modified in accordance with the INSPIRE specifications. In addition, we will experiment with the JSON-LD plugin, as this allows for stronger linkages between features and better indexing of data by search engines crawlers which should enhance data reuse.

The OGC API community module exposes a variety of services based on the new OGC API commons and the OGC API – Features 1.0 service, delivering feature data. The OGC API – Features 1.0 is thus a replacement for the existing WFS services, although GeoServer will still support the existing WFS services (1.0.0, 1.1.0 and 2.0.0); it will also be possible to use them alongside with the OGC API – Features.

GeoServer core only allows the publishing of features compliant with level SF-0 of OGC Simple Features profile, the App-Schema extension adds to GeoServer the capability of publishing features compliant with level SF-1 and SF-2. This extension brings also the possibility of building features by defining the mappings between a data source, typically a relational database, and a target GML schema. For most of the use cases App-Schema can use the database as is, although for certain corner cases SQL views must be created prior to configuration.

It is worth mentioning that although several data source integrations exist with App-Schema, the most mature and feature rich one is the PostgreSQL \ PostGIS integration. Hence PostgreSQL \ PostGIS will be the target data source for the GeoServer deployments in the context of this project.

4.1.1. OGC API – Features in GeoServer

OGC is defining a new set of services to replace the currently used OWS, that date back 20 years. These new services are collectively identified as OGC APIs, and are based on different concepts compared to the old OWS based services, in particular:

- They are Web APIs, sometimes referred to as RESTful services;
- Each service is described by an OpenAPI document;
- Each of them is geared towards JSON based representations of resources, at least for meta information, while allowing other representations as extensions;
- HTML representation is also promoted as a way to make the service easier to learn and access from a browser;
- Each service has a minimal core, and numerous optional extensions to add functionality.

For example, OGC API – Features 1.0 core does not mandate any output format (GeoJSON being recommended), only supports CRS84, provides limited filtering abilities (time, space, single attribute equality), and requires no schema for the data being published. Extra coordinate systems support, full-fledged filtering and transactions are currently being developed as extensions and will be delivered to the public sometime in the future.

D2 - Deployment strategies for standard-based APIs

The service is based on a small set of resources:

- /, the landing page, providing access to all other resources
- An OpenAPI service description, which is linked from the landing page, and does not have a fixed location, though many implementations place it at /api
- /conformance, a set of conformance classes, a list of URIs identifying the capabilities of the service
- /collections, providing a list of available collections (previously known as feature types)
- /collections/{collectionId}, providing a description of the single collection
- /collections/{collectionId}/items, providing access to the actual data, often in GeoJSON format, although no formats are mandated by the specification
- /collections/{collectionId}/items/{itemId}, providing access to a single feature

The GeoServer implementation provides full core support; all resources are available either as JSON or HTML. A minimal CQL filtering extension has been added, based on the work done at the 2019 Washington STAC and OGC API sprint, while support for the CRS extension is still not present, but incoming in the next few months. As the specification for OGC API Part 3 Common Query Language is still very much a work-in-progress with results not expected until the final quarter of 2020, filter functionality will not be covered within the evaluation of OGC API.

4.1.2. App-Schema

GeoServer distinguishes between Simple Features, features whose properties may contain only values of XML simple types, and Complex Features, features whose properties may contain values of XML complex types and link other features. The former roughly corresponds to compliance level SF-0 of OGC's Simple Features profile (OGC® 10-100r3), while the latter corresponds to level SF-1. By default, GeoServer only supports simple features. In the most common use case, GeoServer is connected to a data source such as a relational database and each table containing geographic data is automatically mapped to a simple feature type.

Things become more complicated when the physical data model is more complex. For example, if multiple relational database tables are required for a single layer, simple features are often not enough; thus, we must create complex features, which support nested objects (list of objects containing other objects or other lists of objects). Publishing complex features requires more complex configuration work from the user, at least a target GML schema and App-Schema mappings between the data source and the target GML schema must be provided.

App-Schema will produce a stream of complex features that will be used by the GML and GeoJSON encoders, which are used by WFS and OGC API – Features, to produce the respective responses. The produced GML responses will match the target GML schema that was used to create the App-Schema mappings. The GeoJSON responses will be produced based on a set of rules defined based on the original target GML. It is worth mentioning that even if we are only interested in the GeoJSON responses, the App-Schema mappings still need to be defined between a data source and target GML schema.

App-Schema will also be utilized to interpret queries performed against the published complex features. In most situations, App-Schema will be able to interpret the query and translate it to one or more SQL queries that will be sent to the database to retrieve the necessary data. In certain cases, App-Schema will not be able to translate the query to SQL; in those situations, all the data will be retrieved from the database and the filter executed in memory. It should be noted that this requires extra memory consumption and will add a significant performance penalty.

4.1.3. New guidelines for modern standards

The new guidelines based on modern standards, GeoJSON and OGC API – Features, that are being proposed within the INSPIRE community will be considered during this work, foremost:

- [Alternative encodings for INSPIRE, the GeoJSON encoding;](#)
- [Setting up an INSPIRE Download service based on the OGC API-Features standard.](#)

The implications of these new guidelines' implementations are not only purely technical and raise concerns regarding the reusability of the modeling and implementation efforts already made by INSPIRE implementers.

Ideally it should be possible to migrate existing implementations to these new API's, leveraging on the investments already made. At the same time, new implementations should be free to adopt these new guidelines without being tied to the complexity of previous data models. This raises a few technical challenges that will be investigated during this project.

The implementability of those guidelines will also be evaluated, using GeoServer as a sandbox. We already foreseen a few enhancements that may be done to GeoServer to simplify the publishing of complex data models.

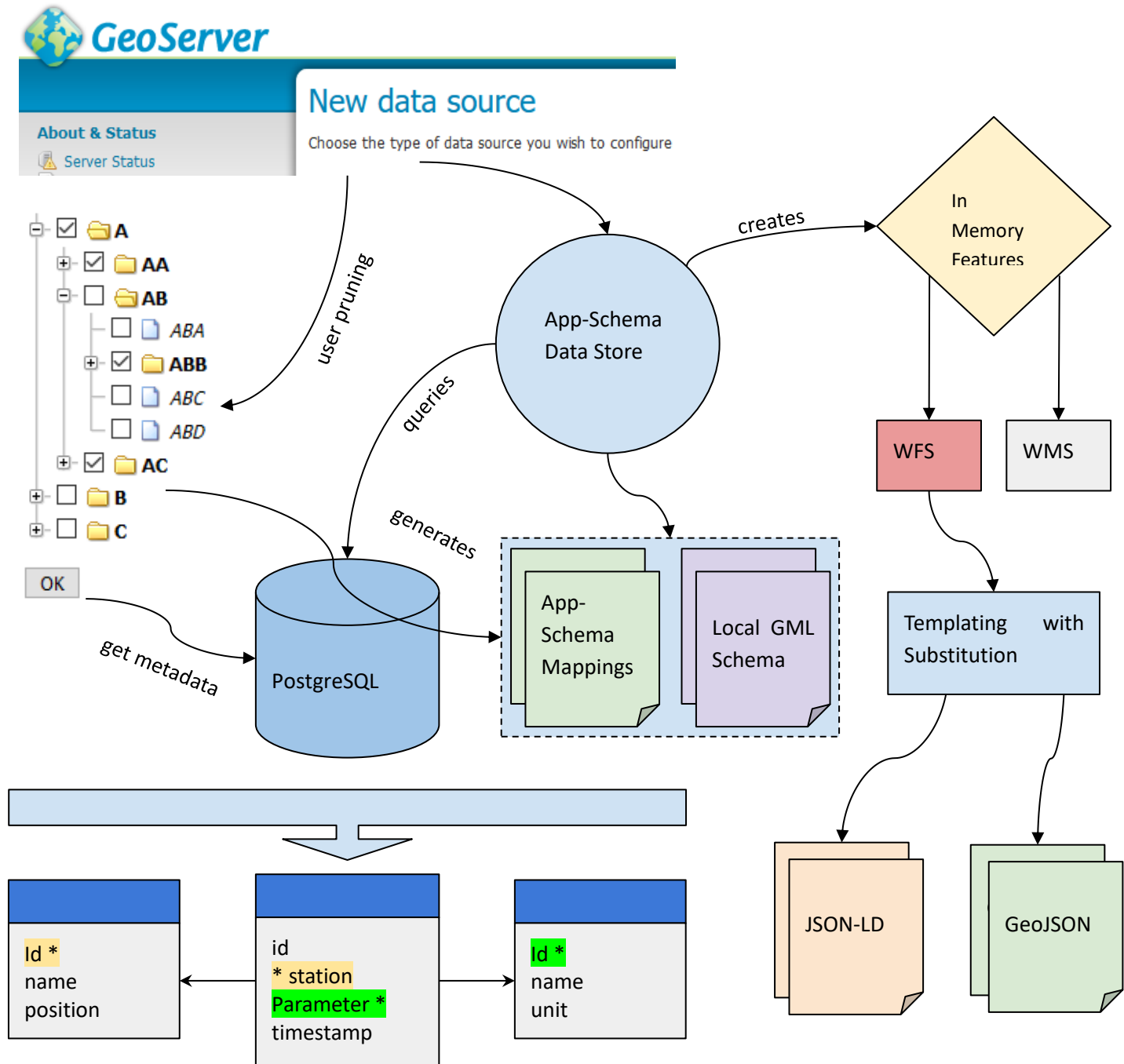


Figure 2: GeoServer New Smart App-Schema Data Source

The idea is to create a new Smart App-Schema data source in GeoServer that will be able to read a database structure, starting from a root table, and automatically generate the App-Schema mappings and target

D2 - Deployment strategies for standard-based APIs

local GML schema² required by App-Schema. The user will be prompted to perform some basic pruning, excluding relations or columns that are not relevant.

The Smart App-Schema data store will be responsible to query the database and build the complex features whose structure will follow the automatically generated GML schema, which was generated based on the database structure. The store will also be responsible to interpret users queries and correctly delegate them, when possible, to the database.

We also plan to extend the current [JSON-LD templating mechanism](#) to support GeoJSON, allowing users to customize the GeoJSON output using a *what you see is what you get* approach. This will significantly simplify the publishing of complex data and provide a very flexible customization for the GeoJSON output without a performance penalty.

4.1.4. Preconfigured GeoServer Black Boxes in Docker

In typical GeoServer deployments, the GeoServer instance is configured to access the data provider's native database source via the App-Schema configuration files. While this is surely the best mode of deployment pertaining to efficient data provision of up-to-date data, the downside is that these configurations can rarely be reused as they strongly reflect the data providers in-house infrastructure.

In order to allow for rapid deployment, we have had success in the utilization of preconfigured GeoServer Black Boxes. Under this deployment paradigm, the source database accessed by GeoServer is structured in close alignment with the final GML output format; an App Schema configuration provided that allows GeoServer to provide these FeatureTypes. This configuration is then packaged for simple deployment via docker.

The data to be exposed can then be provided to this preconfigured GeoServer Black Box in two manners:

- Import: The required data can be transformed and imported directly into the foreseen database tables. The advantage of this option is that the GeoServer deployment can be hosted on separate infrastructure, which has security advantages; the disadvantage of this option pertains to highly dynamic data, as this would require regular updates to the system.
- Views: As an alternative to data import, the required data can also be made available in the form of database views that mimic the structure of the database tables foreseen in the GeoServer Black Box configuration. Depending on the size of the dataset to be provided, one may need to utilize materialized views for performance criteria. The advantage of this option is that dynamic data is

² The generated target local GML schema is not meant to be used for data exchange or to be public, it should be seen as an internal configuration file for App-Schema.

D2 - Deployment strategies for standard-based APIs

automatically updated; the disadvantage of this option pertains to large datasets that could lead to performance issues with the views as well as security considerations.

Alternatively, these preconfigured GeoServer Black Boxes can also be used as a starting point for deployment directly linked with a data provider's data sources. While the modification of the App Schema configuration does require a degree of technical experience, it is far simpler than having to start from scratch as most of the difficult aspects of the configuration have already been correctly performed.

4.2. OGC API – Features Development - OGC API Simple

In order to ascertain the complexity of developing an OGC API - Features conformant API on top of existing data holdings, a stand-alone development of the OGC API will be implemented. As this exercise should reflect the issues encountered by non-experts attempting to create correct APIs based on the existing standards and guidance documents, we have started a collaboration with a vocational informatics school in Vienna. A team of interested students has been created, and implementation work initiated based on the following simple specification requirements:

- Deploy on an existing geospatial database
- Manage connections to geospatial databases based on PostGIS and Spatialite DB
- Expose columns as properties in SF-0
- Select the relevant geometry column and expose that as Geometry
- Map table name to collection name if desired
- Map column names to properties names if desired
- Hide or show tables and/or columns if desired
- Provide all through a user interface, configuration details stored as json config files
- Additional collection types can be created through SQL statements
- Filtering functionality (limit, bbox, feature properties and combinations)
- Display configured collections in a simple map view
- Display collections and items as HTML and JSON as far as reasonable
- Docker-enabled installation

D2 - Deployment strategies for standard-based APIs

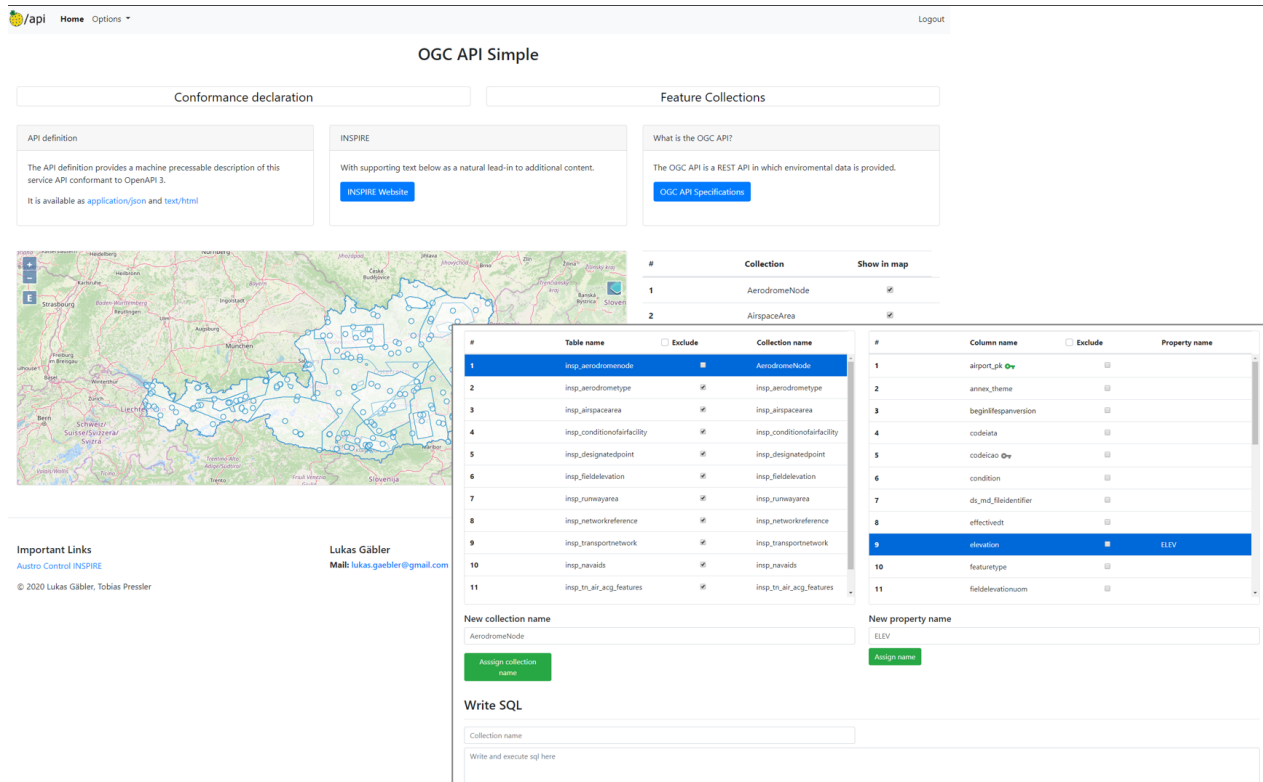


Figure 3: OGC API - Simple Interface

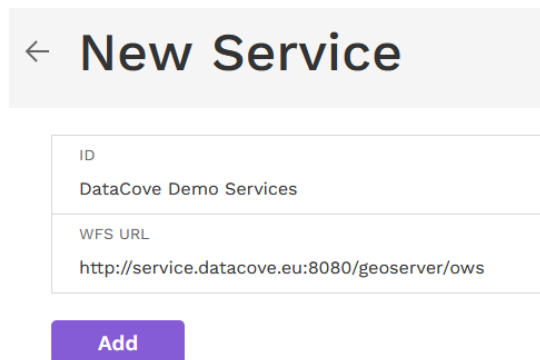
The application will be a Java Servlet and will be developed and tested using the Apache Tomcat web server. The backend of the OGC API Simple will be developed as a Java Spring application and the frontend will use the TypeScript based web framework Angular (v.8). The web interface shown in Figure 3: OGC API - Simple Interface will get the data to display from the Java Spring REST API which will be implemented according to the OGC API Core specification and some more REST paths (e.g.: for the authentication).

This simple OGC API demonstrator will be created in an iterative manner. In an initial step, the collection name will correspond to the name of the table in the database, while the individual attribute names will be the same as the columns of the selected table. In further iterations, it will be possible to configure these names as required. All source code is available via the project GitHub under the OGCAPI Simple folder.

For encoding, this implementation will rely on simple feature encoding standards and only provide GeoJSON and JSON encoded data. At present, this implementation is limited to PostGIS and Spatialite databases. Austro Control has kindly provided dumps of their repositories for this development work. While this work will foremost concentrate on Austro Control Air Transport Data, other data sources will also be demonstrated.

4.3. LD-Proxy

In order to leverage existing WFS2 implementations not stemming from GeoServer deployments as well as GeoServer deployments where we cannot include the OGC API extension, we will provide an instance of LD-Proxy that can perform on-the-fly conversion. Adding services to LD-Proxy is a very simple undertaking, all one needs to provide in order to configure a new service interface is a name and the URL



← New Service

ID	DataCove Demo Services
WFS URL	http://service.datacove.eu:8080/geoserver/ows

Add

Figure 4: LD-Proxy - add new service

of the WFS2 Endpoint.

After the Service Endpoint has been added to the LD-Proxy instance, a list of the feature types available from that end point are displayed.

The first time one selects a feature type, LD-Proxy requests information on the feature types being provided using the WFS DescribeFeatureType request. Please note that the functionality of this service

D2 - Deployment strategies for standard-based APIs

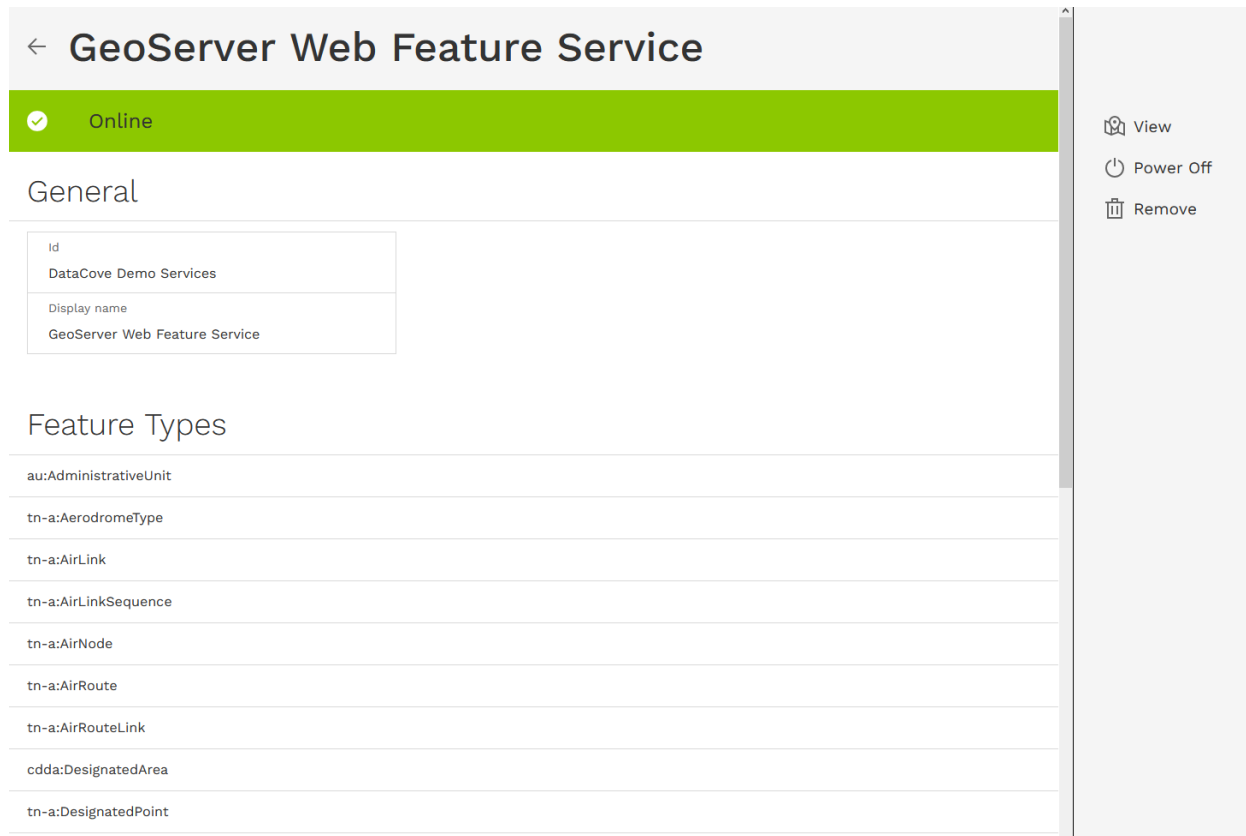


Figure 5: LD-Proxy - show feature types

request is essential for LD-Proxy to function, some WFS servers unfortunately do not correctly support this.

Once LD-Proxy has retrieved all relevant information on the feature types being provided by the underlying WFS2, the structure of the feature types can be displayed. In the example in Figure 6, the attributes for the type tn-a:AirNode are displayed.

←

tn-a:AirNode

General

Id

AirNode

Display name

tn-a:AirNode

Mapping

tn-a:AirNode

☐ @gml:id

☐ net:beginLifespanVersion

net:inspireId

base:Identifier

☐ base:localId

☐ base:namespace

☐ base:versionId

☐ net:endLifespanVersion

☐ net:geometry

tn:geographicalName

gn:GeographicalName

☐ gn:language

Figure 6: LD-Proxy - feature attributes

Where applicable, information on the data type of the individual attributes are mapped to common vocabularies such as schema.org. This is of special relevance when accessing the available data in JSON-LD format.

LD Type

http://schema.org/Place

Figure 7: LD-Proxy - schema mapping

Once the LD-Proxy endpoint has been configured, one can then proceed to view or access the data in GML, GeoJSON or JSON-LD format.

28 / 48

tn-a:DesignatedPoint

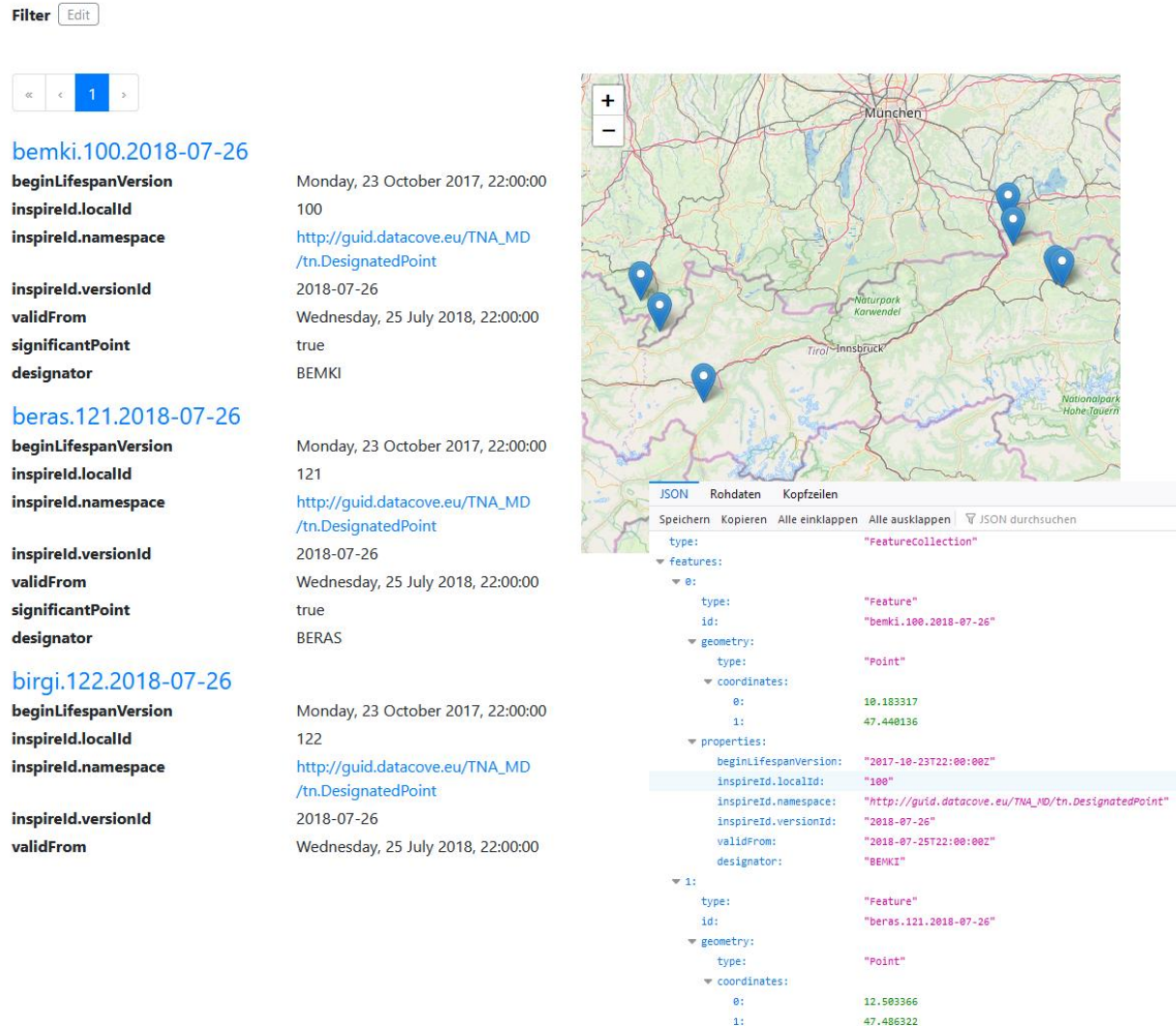


Figure 8: LD-Proxy - TN-A Data

4.4. FROST Deployment

For data storage, FROST uses the PostgreSQL database server with PostGIS geospatial extension, whereby the data model must be aligned with the underlying data model put forth in the STA standard. Additional attributes as required by INSPIRE can be supplied in extended properties sections, as described in

“Extending INSPIRE to the Internet of Things through SensorThings API”³. These extended properties sections are available in the 1.1 version of the STA standard, which FROST already implements.

4.4.1. FROST-Maintained Database

By default, FROST creates and maintains its own tables in the database, and users do not need to deal with the details of these tables besides adding use-case specific indices to speed up specific queries required for the use case. When using FROST (or any other SensorThings API server implementation) in this way, all interactions with the data can happen through the API itself. New entities are added and updates are made through the API, and data is read through the API.

4.4.2. Custom Database

Is it also possible to not have FROST maintain its own database tables, but instead mimic those tables using database views and have FROST operate on those views. The big advantage of this approach is that any existing applications and procedures can remain in place. Any new data automatically appears in the SensorThings API interface and does not need to be duplicated.

For most deployments, we have been successful by creating database views on the source data aligned with the database tables expected by FROST; however, in the case of massive data holdings, it has been shown necessary, for performance purposes, to create a physical copy of the source data, either using materialized views or transfer the data to FROST database tables.

4.4.3. Deployment Options

The FROST server software is deployed separately from the database, and can be deployed in several ways:

- As java “war” file that can be deployed on an existing Tomcat or other servlet container. This option is ideal for organisations that already have other services running on Tomcat and thus have experience administering Tomcat.
- Using the docker image, on any docker environment. The FROST Docker images allow for an easy way to deploy and update FROST. In a Docker environment on a cluster it is possible to set up automatic horizontal scaling.

³ Extending INSPIRE to the Internet of Things through SensorThings API, <https://www.mdpi.com/2076-3263/8/6/221>

D2 - Deployment strategies for standard-based APIs

- Using the Helm⁴ package manager on a Kubernetes cluster. This option is the easiest for those organisations that use Kubernetes to manage their Docker environment.

⁴ <https://helm.sh/>

5. Data Provider Specific Deployment

Within the API4INSPIRE project, we have brought together a set of different data providers with a wide spectrum of levels of expertise and organizational infrastructure. This allows us to illustrate various deployment options for both OGC API - Features as well as SensorThings API. At the one end of this spectrum, we find data providers such as Austro Control, who have long fulfilled their INSPIRE Requirements pertaining to provision of GML encoded data via WFS utilizing GeoServer and only needed to add the OGC API - Features extension to their existing infrastructure, or the City of Hamburg with a rich smart sensor landscape long deployed on SensorThings API. At the other end of the spectrum are data providers who have to date relied on internal or national data structures and formats, where the necessary data sources must first be identified, mapped to INSPIRE requirements, the data transformed as required by the data models, and finally the necessary software for the selected API deployed. Going beyond this foreseen spectrum of data provision options, based on current events triggered by the Corona virus crisis, we've started leveraging existing non-harmonized data sources on *in situ* air quality measurements from both national and European data sources, transforming and providing this data via SensorThings API.

In the following sections, we detail the deployment methodology to be utilized for which datasets stemming from which data providers.

5.1. Austrian Meteorological Agency – ZAMG (AT)

The Austrian Meteorological Agency has over a century of experience in the collection and administration of meteorological data. Thus, this institution has a wide range of expertise at its disposal pertaining to all aspects of meteorological data management and provision. Information on meteorological measurement stations as well as the data ensuing from these stations is currently available in the form of a WFS2 deployment utilizing GeoServer. This endpoint will be extended in cooperation with ZAMG staff to also provide an OGC API endpoint. In addition, SensorThings API will be deployed at ZAMG.

- Meteorological Measurement Stations [EF]: SensorThings API and OGC API - Features
- Meteorological Data [AC/MF]: SensorThings API and OGC API - Features

5.2. Austro Control – ACG (AT)

Austro Control (German: Österreichische Gesellschaft für Zivilluftfahrt) is the air navigation services provider that controls Austrian airspace. Its location and jurisdiction is Vienna, with the physical offices also being located in Vienna. Austro Control is the official and certified air navigation service provider for Austrian airspace. In this role, Austro Control manages all data pertaining to air traffic networks in Austria, and currently provides this data via an INSPIRE compliant endpoint running on GeoServer. This endpoint has been extended by Austro Control staff to also support OGC API - Features. In addition, Austro Control is sponsoring the development of a simplified OGC API - Features implementation that supports the

D2 - Deployment strategies for standard-based APIs

provision of data conforming with the SF-0 simple features specification; further information and source code is available via the project GitHub.

- Air Traffic Network [TN-A]: Currently available as WFS2 and OGC API - Features on GeoServer. In addition, this data will be made available via the simple features implementation of OGC API - Features.

While the WFS2 and OGC API - Features endpoints will continue to be hosted by Austro Control via their corporate site, the simple features implementation of OGC API - Features will be hosted by Fraunhofer IOSB.

Currently available endpoints:

WFS2: <https://sdigeo-free.austrocontrol.at/geoserver/tn-a/wfs?service=WFS&version=2.0.0&request=GetCapabilities>

OGC API: <https://inspire.austrocontrol.at/ogcapi/ogc/features>

5.3. Umweltbundesamt - UBA (AT) & European Environment Agency - EEA (EU)

The provision of air quality data was not initially foreseen within the API4INSPIRE project. However, due to recent events pertaining to the spread of Covid-19, this data source has been added in an ad-hoc manner without integration of the data providers. This was possible in part due to the fact that the data has already been harmonized due to the European Air Quality Directive (AQD) data formats, that were created through the extension of the INSPIRE Environmental Monitoring Facilities Theme.

In an initial step, air quality data was harvested from the near-real-time services offered by Umweltbundesamt, the Austrian Environment Agency, that provides monitoring stations via WFS2 and near-real-time measurement data from these stations via OGC SOS. This data was inserted into a SensorThings API server via native SensorThings API requests, the endpoint has been freely published.

The importer runs in two stages. First it queries the WFS to find all the Stations (Things), Processes (Sensors), ObservedProperties, SamplingPoints (Datastreams), etc. and creates the entries in the SensorThings service. This needs to be done only once, but it can be done again when the source data changes, since the importer is smart enough to update data that has already been imported. The second step is to import the actual Observations from the SOS. On an hourly basis it fetches, for each Datastream, the new Observations since the last Observation that was imported. On a daily basis it fetches all Observations of the last three days, to check if any Observations were changed after they had been imported.

Based on increasing interest by other European Union Member States, we then turned to the sources offered by the European Environment Agency. This task proved to be the bigger challenge; while the

D2 - Deployment strategies for standard-based APIs

concepts from the AQD data formats have been maintained, the structure foreseen by the data models was removed, with the resulting data only available in not-standardized csv formats. While this did cause some issues in the import process, they could finally be resolved.

Since the source data are published in a different manner, the importer also works slightly differently. For the EEA data, the importer starts from the CSV files containing the Observations. When importing an Observation, the importer checks if the Datastream and other metadata entities already exist for this Observation. If not, the importer downloads the CSV file containing the station metadata for all stations in Europe, and uses this data to generate the missing metadata entities.

The total number of Observations, for the ObservedProperties SO₂, PM_{2.5}, PM₁₀, NO, O₃, CO, CO₂, NO₂ and 'NO_x as NO₂' in the countries of Austria, Switzerland, Germany, Italy and France, for the time period of 2018 to now is about 100 million. These are grouped into 8185 Datastreams, for 2117 measurement stations (Things).

At present, we are in negotiations with other national data providers interested in adopting SensorThings API for direct provision of air quality data from their respective institutions.

Currently available endpoints:

STA: <https://airquality-frost.docker01.ilt-dmz.iosb.fraunhofer.de/v1.1>

Viewer: <https://wg-brgm.docker01.ilt-dmz.iosb.fraunhofer.de/servlet/is/121/>

5.4. City of Hamburg – CH (DE)

The City of Hamburg has long seen the potential of Smart City technology, and long been involved in diverse Smart City initiatives, successively extending their smart sensor infrastructure to an ever widening usage area. These efforts have gained the city accolades such as being Germany's smartest city. While formally much of this data goes beyond the formal INSPIRE specifications, we believe that it nicely illustrates further potential of such a consistent European SDI merging into an Urban Data Platform and highlights the challenges of integration between the formally specified core of INSPIRE with additional data sources.

- Smart City Sensors [EF++]: A great deal of real-time information pertaining to mobility within Hamburg is already available via SensorThings API. This includes data on the availability of charging stations and data stemming from the StadtRad CityBike initiative.
- Energy Campus [EF++]: while stemming more from building management, the data from the Energy Campus of the Hamburg University of Applied Sciences provides a wide field for experimentation with the SensorThings API.
- Road Networks [TN-R]: This data is currently available via an INSPIRE compliant WFS2 from the regional authorities. While we'd initially hoped to make this data available via LD-Proxy this has not been possible due to issues in the underlying WFS configuration. As an alternative to this

D2 - Deployment strategies for standard-based APIs

source, we are proposing the use of data from OpenStreetMap (OSM), which has the added benefit of providing routing functionality that will be beneficial to the planned use cases.

Currently available endpoints:

STA: <https://iot.hamburg.de/v1.0>

Viewer: <https://wg-brgm.docker01.ilt-dmz.iosb.fraunhofer.de/servlet/is/110/>

5.5. French Geological Survey – BRGM (FR)

BRGM has long been involved in pushing the envelope pertaining to the possibilities of environmental data provision. They are intensely involved in various OGC initiatives pertaining to the exploration of novel data provision methodologies as well as providing valuable input to the core standardization activities within the OGC. BRGM staff was integral in the creation of relevant INSPIRE data specifications, and this organization has provided funding for many of the tools the INSPIRE community is coming to rely upon pertaining to access to and processing of complex spatial features. Based on this expertise, they will also be providing support towards the French Office for Biodiversity, that does not have such strong resources at its disposal, but hosts complementary datasets.

BRGM is the French Geological survey but its activities go way beyond the “traditional” aspects of Geology and Mineral Resources. Its Risk division has an important activity and BRGM IT division hosts the French national silo for risks (<https://www.georisques.gouv.fr/>).

Its Environmental & Ecotechnologies division provides expertise on Groundwater but also on Ecotechnologies, Soils, etc. Associated with this, the BRGM IT division hosts many data portals of the French Water Information System but also for the national polluted soils Information System on behalf of the French ministry of environment.

Thus, the available data sources pertain to geological structures as well as the subsurface aspects of water such as the location of aquifers and the measurement of groundwater quality and quantity. The pertinent INSPIRE Themes for this data are Geology (Hydrogeology package and its equivalent in the international OGC standard GroundWaterML2 - OGC 16-032r2 - where BRGM and JRC pushed the necessary elements for INSPIRE compliance) for the spatial feature types, Environmental Monitoring Facilities for the complementary groundwater quality and quantity measurements.

For the provision of OGC API - Features, we will start with preconfigured GeoServer Black Boxes in Docker, whereby there are plans to successively modify the configuration to directly access the native data sources available within BRGM.

For the provision of dynamic measurement data via SensorThings API, BRGM started in 2018 with porting its infrastructure collecting raw Groundwater observation from GPRS sensors and exposing it according to

D2 - Deployment strategies for standard-based APIs

O&M from OGC:SOS to OGC SensorThings API. This activity is directly contributing to the API4INSPIRE project.

Within API4INSPIRE, the following datasets will be made available via the following APIs:

- Groundwater Quantity [EF]: both OGC API - Features and SensorThings API
- Hydrogeological Unit [GE & or OGC:GroundWaterML2]: OGC API - Features
- Flood Risk [NZ or AM]: OGC API - Features; candidate for SF-0 encoding

Currently available endpoints:

STA: <https://sensorthings.brgm-rec.fr/SensorThingsGroundWater/v1.0/>

Viewer: <https://wg-brgm.docker01.ilt-dmz.iosb.fraunhofer.de/servlet/is/101/>

5.6. French Office for Biodiversity (OFB) – INSIDE - environmental information systems research center (FR)

The French Office for Biodiversity is a very new institution, founded on 1 January 2020 to protect and restore biodiversity in metropolitan France and overseas territories. It merges the French Agency for Biodiversity with the French national agency for wildlife. The public institution is under the tutelage of the French Environment and Agriculture ministers. As a result it is in charge of three main French Information Systems: the Water Information System, the Marine Information System, the Biodiversity Information System

The Agency has 5 complementary missions:

- making available knowledge of, and research and expertise on species, and their uses
- environmental and wildlife health policing
- support for public policies
- assisting protected area managers
- supporting stakeholders and mobilising civil society

Jointly with BRGM, OFB set up the environmental information systems research center called 'pole INSIDE'. It aims to push standards, practices and an identified reference open-source implementations up to a maturity level that enables them to be deployed in production by the three Information Systems driven by OFB as well as its partners (BRGM, IFREMER, OFB, MNHN,...).

Its activity and roadmap can be openly accessed through its GitHub (<https://github.com/INSIDE-information-systems/Roadmap/projects/2>).

D2 - Deployment strategies for standard-based APIs

“INSIDE” teams members will support the provision of OFB data to the API4INSPIRE project.

For the provision of OGC API - Features, we will start with preconfigured GeoServer Black Boxes in Docker, whereby there are plans to successively modify the configuration to directly access the native data sources available within OFB.

For the provision of dynamic measurement data via SensorThings API:

- Surface water quantity : a new ST API instance is being set up on top of the national flows (XML Sandre based). It will be fed using ST API REST & MQTT,
- Surface water quality : in 2018-2019 INSIDE research center already worked with DataCove and Fraunhofer IOSB on exposing the national water quality database according to ST API (https://github.com/INSIDE-information-systems/SensorThingsAPI/tree/master/Surface_Water_Quality_Bigdatavolume_Benchmarking). This work will contribute to API4INSPIRE

For the purpose of this work, the following datasets will be made available via the following APIs:

- Surface Water Quality & Quantity [EF & OM]: both OGC API - Features and SensorThings API
- Hydrological Networks [HY]: OGC API - Features
- Water Transport Networks [TN-W]: OGC API - Features; possible candidate for SF-0 encoding

Currently available endpoints:

STA Surface Quantity: <https://sensorthings.brgm-rec.fr/SensorThingsFlood/v1.0>

STA Surface Quality: <https://sensorthings-wq.brgm-rec.fr/FROST-Server/v1.0>

Viewer: <https://wg-brgm.docker01.ilt-dmz.iosb.fraunhofer.de/servlet/is/101/>

5.7. Environment Agency Baden-Württemberg – LUBW (DE)

The Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (LUBW) or Baden-Württemberg State Institute for the Environment, Survey and Nature Conservation is a central institution of the German federal state of Baden-Württemberg, whose legal responsibilities lie in the fields of environmental conservation, health and safety and consumer protection and thus they support the state authorities at a technical level in nature conservation and radiation protection. Surface Water Quality & Quantity [EF]: SensorThings API

- Hydrological Network [HY]: SensorThings API and OGC API - Features

D2 - Deployment strategies for standard-based APIs

- Surface Water Quality [EF & OM]: both OGC API - Features and SensorThings API

For the provision of the spatial data pertaining to the hydrological network in Baden-Württemberg we will utilize a preconfigured GeoServer Black Box in Docker, while additionally making this information available via SensorThings API. The dynamic Surface Water Quality & Quantity data will be provided via SensorThings API. As Fraunhofer IOSB is responsible for the administration and management of LUBW's water resources, the APIs from this data provider will be configured and deployed on Fraunhofer IOSB Infrastructure.

The data is available in an internal expert system used to manage everything water-related that the LUBW is responsible for. The system consists of a database and a fat-client used to access this database. To export the relevant data from this expert system into a server implementing the SensorThings API, the fat-client has been extended with export functionality. This export function first checks if the required metadata entities (Things, ObservedProperties, etc.) exist, and creates them if they are missing, or updates them if they are out of date. Then the exporter checks which Observations need to be exported or updated.

Currently available endpoints:

STA: <https://lubw-frost.docker01.ilt-dmz.iosb.fraunhofer.de/v1.1>

Viewer: <https://wg-brgm.docker01.ilt-dmz.iosb.fraunhofer.de/servlet/is/114/>

5.8. Summary Overview

The following table provides a summary overview of all data providers with available datasets.

	Features	Observations	INSPIRE Themes
ZAMG	Not public WFS2 endpoint	Not public WFS2 endpoint	AC/MF
ACG	Public WFS2 endpoint	NA	TN-A
UBA	Public WFS2 endpoint	Public SOS endpoint	EF & OM
EEA	Public CSV access	Public CSV access	EF & OM
CH	Public WFS2 endpoints	Public STA endpoint	TN-R, Smart City
BRGM	Public WFS2 endpoints	Public SOS & STA endpoints	EF, GE & or OGC:GroundWaterML2, NZ or AM
OFB	Internal DB, non public	Internal DB, non public	EF & OM, HY, TN-W
LUBW	Internal DB, non public	Internal DB, non public	EF, HY

Table 1: Overview Data Providers and Datasets

6. Conclusions

In the sections above, we have described the methodologies we have defined for the deployment and configuration of both OGC API - Features as well as SensorThings API on a wide variety of stakeholder infrastructure. These deployment plans pertain to both existing software systems as well as to new developments being triggered by the API4INSPIRE Project such as the OGC API Simple currently being developed. In addition, both the GeoServer system for the provision of OGC API - Features as well as the FROST SensorThings API Server are being extended as additional requirements emerge.

Pertaining to the SensorThings API we have made rapid progress in endpoint deployment, with many of the planned endpoints already operational. This was in part due to preexisting work with our data providers, but also due to the power of standardized data models as illustrated in our ad-hoc air quality activity. In parallel to the work on API4INSPIRE, we will also monitor related activities pertaining to STA such as the newly emerged COVID-STA available at <http://193.196.138.56:8080/STACOVID/>. One major advantage of the FROST SensorThings API implementation is the utilization of JSON-Blobs for the storage of all additional properties. Where traditional servers and RDBMS require the creation of a dedicated database schema and subsequent configuration of the server to this structure, FROST allows full access to properties stored in the JSON-Blob format without additional configuration.

Provision of OGC API - Features is more challenging as this standard is much fresher, and software still in the process of development. Necessary extensions to GeoServer for simple provision of OGC API - Features have been agreed upon as described in the sections above, and will be implemented within the next weeks and months. While this development work progresses, we will continue with the configuration of App Schema based on existing GML Schemas for provision of complex data sources. Once the new Smart App-Schema data store has been implemented, we will commence testing this new development for simpler provision; some selected endpoints will be reimplemented with this technology in order to allow for comparison of effort between the two alternative configuration approaches. In addition, our newly developed OGC API - Simple will be deployed on several endpoints to allow us to explore the potential of SF-0 encoding for provision of at least simplified datasets.

7. Next Steps

A challenge in the next weeks and months will be ensuring enough end user engagement as more and more planned face-to-face meetings and events are being postponed or cancelled. This is one of the reasons we've widened the scope of provided data with activities such as the Ad-hoc Air-Quality API or the COVID-STA in order to best leverage current events and engage potential communities. In addition, we are currently exploring virtualization methods, providing alternatives to physical meetings in the form of Webinars and other virtual events and challenges. This work will be continued in the next weeks and months in order to assure a good outcome to API4INSPIRE regardless.

8. Annex 2.A Mapping INSPIRE AQD to SensorThings API

8.1. Austrian Inspire Air Quality Data

The alignment between the data being harvested from the Austrian INSPIRE air quality services and SensorThings API was quite simple due to the similarities in the underlying data models, as is shown in the following tables. In addition, we made use of the mapping between INSPIRE and SensorThings available from the publication Extending INSPIRE to the Internet of Things through SensorThings API - doi:10.3390/geosciences8060221

The necessary information for the description of the observed properties was taken directly from the corresponding EEA Vocabulary.

STA Class	STA Attribute	AQD Field
DATASTREAMS		
DATASTREAMS	ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPoint/@gml:id
DATASTREAMS	PHENOMENON_TIME_START	/wfs:FeatureCollection/wfs:member[91]/aqd:AQD_SamplingPoint/ef:operationalActivityPeriod[3]/ef:OperationalActivityPeriod/ef:activityTime/gml:TimePeriod/gml:beginPosition
DATASTREAMS	PHENOMENON_TIME_END	/wfs:FeatureCollection/wfs:member[91]/aqd:AQD_SamplingPoint/ef:operationalActivityPeriod[3]/ef:OperationalActivityPeriod/ef:activityTime/gml:TimePeriod/gml:endPosition
DATASTREAMS	PROPERTIES/processType	"http://inspire.ec.europa.eu/codeList/ProcessTypeValue/process"
DATASTREAMS	PROPERTIES/resultNature	"http://inspire.ec.europa.eu/codeList/ResultNatureValue/primary"
OBSERVATIONS	FEATURE_ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPoint/ef:observingCapability/ef:ObservingCapability/ef:featureOfInterest/@xlink:href
DATASTREAMS	OBS_PROPERTY_ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPoint/ef:observingCapability/ef:ObservingCapability/ef:observedProperty/@xlink:href

D2 - Deployment strategies for standard-based APIs

DATASTREAMS	SENSOR_ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPoint/ef:observingCapability/ef:ObservingCapability/ef:procedure/@xlink:href
DATASTREAMS	PROPERTIES/metadata	"http://luft.umweltbundesamt.at/inspire/wfs?service=WFS&version=2.0.0&request=GetFeature&typeName=aqd:AQD_SamplingPoint"
DATASTREAMS	THING_ID	/wfs:FeatureCollection/wfs:member/aqd:AQD_SamplingPoint/ef:broader/@xlink:href

Table 2: STA Datastream Mapping Austrian Air Quality

STA-Class	STA-Attribute	AQD Field
SENSORS	ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/@gml:id
SENSORS	PROPERTIES/namespace	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:inspireId/base:Identifier/base:namespace
SENSORS	NAME	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/aqd:measurementEquipment/aqd:MeasurementEquipment/aqd:equipment/@xlink:href
SENSORS	PROPERTIES/measurementtype	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/aqd:measurementType/@xlink:href
SENSORS	PROPERTIES/method	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/aqd:measurementMethod/aqd:MeasurementMethod/aqd:measurementMethod/@xlink:href
SENSORS	DESCRIPTION	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/aqd:equivalenceDemonstration/aqd:EquivalenceDemonstration/aqd:demonstrationReport
SENSORS	PROPERTIES/responsibleParty	
SENSORS	PROPERTIES/responsibleParty/individualName	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:individualName/gmd:LocalisedCharacterString

D2 - Deployment strategies for standard-based APIs

SENSORS	PROPERTIES/responsibleParty/organisationName	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:organisationName/gmd:LocalisedCharacterString
SENSORS	PROPERTIES/responsibleParty/adminUnit	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:contact/base2:Contact/base2:address/ad:AddressRepresentation/ad:adminUnit/gn:GeographicalName/gn:spelling/gn:SpellingOfName/gn:text
SENSORS	PROPERTIES/responsibleParty/locatorDesignator	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:contact/base2:Contact/base2:address/ad:AddressRepresentation/ad:locatorDesignator
SENSORS	PROPERTIES/responsibleParty/postCode	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:contact/base2:Contact/base2:address/ad:AddressRepresentation/ad:postCode
SENSORS	PROPERTIES/responsibleParty/electronicMailAddress	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:contact/base2:Contact/base2:electronicMailAddress
SENSORS	PROPERTIES/responsibleParty/telephoneVoice	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:contact/base2:Contact/base2:telephoneVoice
SENSORS	PROPERTIES/responsibleParty/website	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_SamplingPointProcess/ompr:responsibleParty/base2:RelatedParty/base2:contact/base2:Contact/base2:website
SENSORS	PROPERTIES/metadata	"http://luft.umweltbundesamt.at/inspire/wfs?service=WFS&version=2.0.0&request=GetFeature&typeName=aqd:AQD_SamplingPointProcess"

Table 3: STA Sensor Mapping Austrian Air Quality

D2 - Deployment strategies for standard-based APIs

STA-Class	STA-Attribute	AQD Field
FEATURES	ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Sample/@gml:id
FEATURES	FEATURE	
FEATURES	GEOM	
FEATURES	GEOM	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Sample/sams:shape/gml:Point/@srsName
FEATURES	GEOM	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Sample/sams:shape/gml:Point/gml:pos
FEATURES	PROPERTIES/metad ata	"http://luft.umweltbundesamt.at/inspire/wfs?service=WFS&version=2.0.0&request=GetFeature&typeName=aqd: AQD_Sample"

Table 4: STA FoI Mapping Austrian Air Quality

STA- Class	STA-Attribute	AQD Field
THINGS	ID	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/@gml:id
	PROPERTIES/namespac e	
THINGS	e	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:inspireId/base:Identifier/base:namespace
THINGS	NAME	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:name
	PROPERTIES/mediaMo nitored	
THINGS	nitored	"http://inspire.ec.europa.eu/codelist/MediaValue/air"
	PROPERTIES/measure mentRegime	
THINGS	mentRegime	"http://inspire.ec.europa.eu/codelist/M MeasurementRegimeValue/continuousDataCollection"
THINGS	PROPERTIES/mobile	"FALSE"

D2 - Deployment strategies for standard-based APIs

THINGS	PROPERTIES/beginTime	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:operationalActivityPeriod/ef:OperationalActivityPeriod/ef:activityTime/gml:TimePeriod/gml:beginPosition
THINGS	PROPERTIES/endTime	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:operationalActivityPeriod/ef:OperationalActivityPeriod/ef:activityTime/gml:TimePeriod/gml:endPosition
THINGS	DESCRIPTION	
THINGS	PROPERTIES/metadata	http://luft.umweltbundesamt.at/inspire/wfs?service=WFS&version=2.0.0&request=GetFeature&typeName=aqd:AQD_Station
LOCATIONS	GEOM	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:geometry/gml:Point/@srsDimension
LOCATIONS	GEOM	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:geometry/gml:Point/@srsName
LOCATIONS	GEOM	/wfs:FeatureCollection/wfs:member[1]/aqd:AQD_Station/ef:geometry/gml:Point/gml:pos

Table 5: STA Thing/Location Mapping Austrian Air Quality

STA-Class	STA-Attribute	AQD Field
DATASTREAMS	SENSOR_ID	/sos:GetObservationResponse/sos:observationData/om:OM_Observation/om:procedure/@xlink:href
DATASTREAMS	OBS_PROPERTY_ID	/sos:GetObservationResponse/sos:observationData/om:OM_Observation/om:observedProperty/@xlink:href
OBSERVATIONS	FEATURE_ID	/sos:GetObservationResponse/sos:observationData/om:OM_Observation/om:featureOfInterest/@xlink:href
OBSERVATIONS	RESULT_NUMBER	/sos:GetObservationResponse/sos:observationData/om:OM_Observation/om:result/swe:DataArray
OBSERVATIONS	DATASTREAM	/sos:GetObservationResponse/sos:observationData/om:OM_Observation/om:parameter[2]/om:NamedValue/om:value

Table 6: STA Observation Mapping Austrian Air Quality

D2 - Deployment strategies for standard-based APIs

8.2. EEA Air Quality Data

The EEA makes CSV files with the air quality measurements available. To find the CSV files, one has to make a GET request to the following URL:

https://fme.discomap.eea.europa.eu/fmedatastreaming/AirQualityDownload/AQData_Extract.fmw?CountryCode=<CountryCode>&Pollutant=<PollutantCode>&Year_from=<StartYear>&Year_to=<EndYear>&Source=All&Output=TEXT&TimeCoverage=Year

The result of this request is a text file with one URL per line. These URLs point to the CSV files containing the actual observations, and are of the following scheme.

Source URL:
<https://ereporting.blob.core.windows.net/downloadservice/<CountyCode> <PollutantCode> <stationId> <Year> timeseries.csv>

Column:	Mapped to:
Concentration	Observation/result
DatetimeBegin	Observation/phenomenonTime (start of interval)
DatetimeEnd	Observation/phenomenonTime (end of interval)
AirQualityStation	Thing/name
AirPollutant	ObservedProperty/name

Table 7: STA Observation Mapping EEA Air Quality

To find the correct Datastream, the following query is sent to the SensorThings service:

`v1.1/Datastreams?$filter=Thing/properties/localId eq '<AirQualityStation>' and ObservedProperty/name eq '<AirPollutant>'`

If this query does not return a Datastream, a new Datastream is created, and possibly a new Thing, Location, Sensor and ObservedProperty. The data required for these entities is read from the station metadata file.

The station metadata can be downloaded for all stations in Europe in one file, containing 58842 records.

Source URL: http://discomap.eea.europa.eu/map/fme/metadata/PanEuropean_metadata.csv

Column:	Mapped to:
---------	------------

D2 - Deployment strategies for standard-based APIs

AirQualityStation	Thing, Location	name properties/localId
Countrycode	Thing, Location, Datastream, Sensor	properties/countryCode
Namespace	Thing, Location, Datastream, Sensor	properties/namespace
AirQualityStationArea	Thing, Location	properties/mediaMonitored
ObservationDateBegin	Thing	properties/beginTime
ObservationDateEnd	Thing	properties/endTime
SamplingProces	Sensor	name properties/localId
MeasurementEquipment	Sensor	metadata
AirPollutantCode	ObservedProperty	definition properties/localId
SamplingPoint	Datastream	name properties/localId
<Source URL>	Thing, Location, Datastream, Sensor	properties/metadata
http://dd.eionet.europa.eu	Thing, Location, Datastream, Sensor	properties/owner
http://inspire.ec.europa.eu /codelist/MediaValue/air	Thing	properties/mediaMonitored

Table 8: STA Observation Metadata Mapping EEA Air Quality