Meteo Italian SupercompuTing poRtAL

Deliverable

D4.3 Requirements for open data re-use

Deliverable Lead:	Dedagroup
Deliverable due date	30/03/2019
Status	FINAL
Version	V1.0

his project has received funding from *European Commission*
Connecting Europe Facility 2014-2020
AGREEMENT No INEA/CEF/ICT/A2017/1567101

Document Control Page

Title	D4.3 Requirements for open data re-use
Creator	Martina Forconi (DEDAGROUP)
Publisher	Mistral Consortium
Contributors	Martina Forconi (DEDAGROUP), Stefano Pezzi (DEDAGROUP), Enrico Minguzzi(ARPAE Emilia Romagna). As Annexes are reported the resulting from the legal advice with Simone Aliprandi.
Type	Report
Language	en-GB and it for Annexes
Rights	copyright "Mistral Consortium"
Audience	☒ public ☐ restricted
Requested deadline	

Contents

Executive Summary

The goal of Mistral project is to facilitate and foster the re-use of datasets by the Italian meteorological community, as well as by its cross-area communities including private providers and citizens.

This deliverable is aimed at identifying and analysing the potentialities and the constraints related to the data use and re-use, considering Legal aspects and Technical requirements for open data re-use.

In order to facilitate data use and re-use, Mistral offers a set of custom services to discover and donwload data. Moreover, prototypes of standard services will be investigated.

1 Legal aspects

1.1 Background

The problem of copyright in databases is relatively recent, and a stable framework has not yet been defined. Data licenses are essential to allow the re-use of a dataset: failure to specify a licence puts both provider and user in a legal grey zone, and may hamper investments and development on the dataset.

Mistral platform gathers data from different providers, and wishes to re-distribute them in an homogeneous framework, allowing both a widespread access to data and the possibility to implement commercial added value services. Data licences are therefore a key element of the infrastructure, and a great effort has been devoted to this subject during the project.

Mistral data providers have different objectives and different data policies; some of them distribute data without specifying a licence, while others have already adopted a licence, and may be reluctant to change it. License policy must provide a common ground for these different needs, and safeguard both data providers and users:

- data providers must be sure that they will not be liable for misuse of their data, once they lose control on data distribution;
- users that intend to set up commercial services must clearly know what they can and can not do with the data, and have a stable and reliable legal framework to develop their products and services.

1.2 Licensing of Mistral data

In order to investigate the problems related to the use of data with particular reference to data distribution licenses, with a subcontracting foreseen in the project budget of Arpae Emilia Romagna, a consultancy was carried out with a lawyer expert in the data licensing problem.

The consultancy lasted several months during which meetings were held between Arpae and the consultant, some of which with the participation of several partners. Important meetings were held to clarify aspects related to the problem of open licenses for both data and software.

The different types of licenses were analyzed in order to have a picture as complete as possible of the emerging issues in terms of managing incoming and outgoing licenses for both software and data.

Licenses that have been proposed and adopted worldwide belong to separate groups, such as “Creative Commons” (CC) and “Open Database License” (ODBL). Each group contains several licenses with different levels of permissions, but licenses coming from different groups are difficult to compare, since they use different language definitions and address different issues.

The greatest compatibility issues arise from the “Share-Alike” clauses, that require that derived products keep the same licence of the original data. As a consequence, two data with different licenses can not be simultaneously used to build a derived product: it is at most possible to show the data together (“mere aggregation”), but they must be always kept physically distinct.

Therfore, on the advice of the consultant, there was an orientation towards licenses with the only "attribution" clause such as the CC by license. Specifically, the CC-BY-4.0 licence has been chosen as a reference, since it fully meets the objectives of the project. Moreover, this license has been recently adopted by several Italian and European institutions: ECMWF uses it for the distribution of all non-relatime data, while talian government has adopted it for the National Open Data Portal, and recommend its use in the “Italian Guidelines for the Exploitation of Public Information Assets”.

The difference between being a simple collector of already existing data or being a dataset maker has also emerged as essential for Mistral.

The consultant gave indications for both of these different perspectives which may also coexist. In cases where Mistral collects existing data, the essential thing to work on, is to prepare information on the license and data source on the platform, in a way that is easily accessible by the user.

Instead, in cases where Mistral produces data deriving, for example, from a postprocessing, it is necessary to take into account the compatibility between different open licenses and consequently it was necessary to program the platform for the correct management of data from incompatible licenses.

Mistral open data strategy was therefore focused on two opjectives:

- the infrastructure must permit to keep data with different licences independent. This has required a great technological effort, but was mandatory to allow the platform to accept data with “share alike” licenses.

- Mistral has recommended, but did not oblige, data providers to adopt the CC BY 4.0 licence. This was also a great challenge, but it has been rather successful, and a great majority of data provieders have adopted this licence.

All the four reports produced by the legal consultancy are placed as annexes to this Deliverable 4.3 (see Annexes list on par. ANNEXES pag. 20, the reports are in Italian Language).

Two of these reports focuses on the problem of data licences, in detail:

- the ANNEX 2 *Proprietà intellettuale e licenze: inquadramento generale* was drawn up by the consultant based on the documentation of the Mistral project and what emerged from the meetings with the project managers and partners.
- the ANNEX 3 *Corretta gestione di alcuni scenari concreti e possibili soluzioni* was realized by the consultant for the purpose of providing operational information on the most probable concrete scenarios that the Mistral Project will face in the field of copyright and data licensing. This report was realized on the theoretical framework arguments already expressed in report of ANNEX 2 but analyzing specific use cases of the Mistral project and the data that was expected to be included in the platform.

2 Technical requirements for open data re-use

In order to facilitate data use and re-use, Mistral offers a set of custom services to discover and donwload data. Moreover, prototypes of standard services will be investigated.

Standards and Services background

In this section we will briefly describe the standards and services adopted or use as prototyped on the Mistral platform.

2.1.1 INSPIRE and Open Geospatial Consortium

The European Directive INSPIRE aims at establishing a spatial data infrastructure to ensure interoperability between databases and facilitate the dissemination, availability, use and reuse of geographical information in Europe.

Introduced in 2007, the EU INSPIRE Directive aimed to improve environmental policy decisions in Europe by introducing general rules to establish an infrastructure for spatial information.

It combines a legal and a technical framework for all EU Member States, to make relevant spatial data accessible for further reuse.

In particular, this means that data shall be discoverable and interoperable through the implementation of a common set of standards, data models and network (web-based) services.

Among the standards considered there are mainly those of ISO / TC211 and the Open Geospatial Consortium.

The Open Geospatial Consortium (OGC) is an international consortium of more than 530 businesses, government agencies, research organizations, and universities driven to make geospatial (location) information and services FAIR - Findable, Accessible, Interoperable, and Reusable.

The OGC enables the sharing of spatial data across EU member states and the successful implementation of INSPIRE by providing the necessary standards for geospatial data.

To ensure the interoperability of the spatial data infrastructures (SDI) of the Member States, the INSPIRE Directive required that common Implementing Rules (IR) were adopted regarding:

1. Data: Conceptual model based on international GI and domain standards: ISO 19100, OGC, WMO, FAO, IUGS, ...
2. Metadata: based on ISO 19115/19
3. Services: based on international GI standards: OGC WMS, WMTS, WFS, CSW, ...

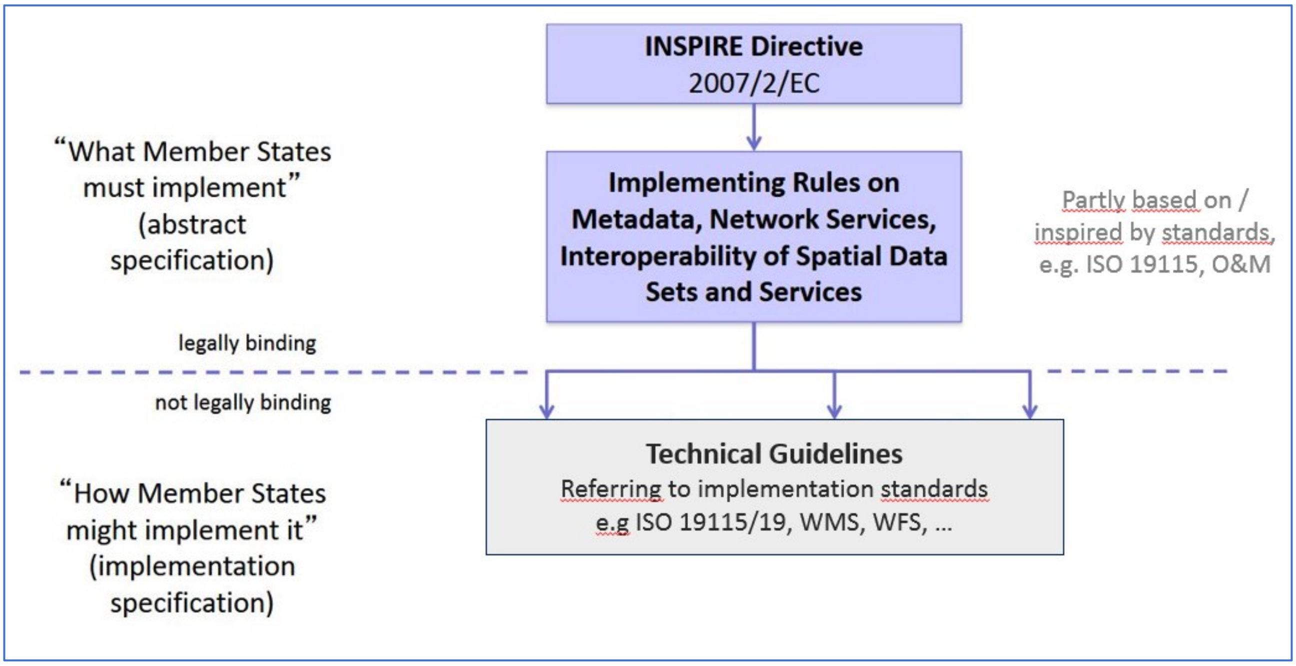

FIGURE 1 - INSPIRE IMPLEMENTING RULES VS. TECHNICAL GUIDELINES[1]

The directive classifies data in different topics with specific requirements.The weather service is one ofthe topics of Annex III of the Directive. It is the subject of a specification document at the European level named “Data Specification on Atmospheric Conditions - Meteorological Geographical Features – Technical Guidelines”.

The meteorological data sets are inherently different from many other traditional geospatial data sets: The data volumes gathered from weather measurements stations and using remote sensing instruments like weather radars and satellites, or produced by numerical weather forecast models are huge and continuously changing.

[1] http://tinyurl.com/krh329m

Weather forecast data have specific characteristics compared to other geographical data, in particular regarding time issue, vertical extent, coordinate systems, refresh rate and data volumes that can often be very high.

In geomatics, a web service offers a service that will enable remote management of data. Several types of services covering different functional areas can be identified:
- discovery services (catalogue services);
- view services, i.e. the simple display of maps (WMS/WMST protocols);
- download services of data (WCS, WFS and STA protocols)

Two examples of Meteorological Open Data Services and INSPIRE directive implementation are given below:

- Meteo France: https://donneespubliques.meteofrance.fr/?fond=geoservices&id_dossier=14
- Finnish Meteorological Institute: https://en.ilmatieteenlaitos.fi/open-data-manual

2.1.2 DCAT-AP_IT

The DCAT Application Profile for data portals in Europe (DCAT-AP) is a specification based on the Data Catalogue Vocabulary (DCAT) developed by W3C.

This application profile is a specification for metadata records to meet the specific application needs of *data portals in Europe* while providing semantic interoperability with other applications on the basis of reuse of established controlled vocabularies (e.g. EuroVoc) and mappings to existing metadata vocabularies (e.g. Dublin Core, SDMX, INSPIRE metadata, etc.).

Also, DCAT-AP provides a common specification for describing public sector datasets in Europe to enable the exchange of descriptions of datasets among data portals. DCAT-AP allows:

- *Data catalogues* to describe their dataset collections using a standardised description, while keeping their own system for documenting and storing them.
- *Content aggregators*, such as the European Data Portal, to aggregate such descriptions into a single point of access.
- *Data consumers* to more easily find datasets through a single point of access.

The DCAT-AP_IT is the Italian Extension of DCAT-AP, defined by AGID, that adds additional constraints to DCAT-AP, whose first version was published in April 2017 (https://www.dati.gov.it/content/dcat-ap-it-v10-profilo-italiano-dcat-ap-0)

2.1.3 OGC:WMS - Web Mapping Service

The OGC Web Map Service (WMS) specification defines an HTTP interface for requesting **georeferenced map images** from a server.

A WMS client first asks what layers are available, in what formats, styles, and projections (GetCapabilities). Then ask an image with specific layers, a specific bounding box and style (GetMap)

The WMS server builds the requested image and returns it as a rectangular png, jpeg, tiff etc. Behind the WMS server is some type of data repository, either vector data or imagery.

The WMS could also provides the ability to query the map (info tool, "what's here?") (GetFeatureInfo).

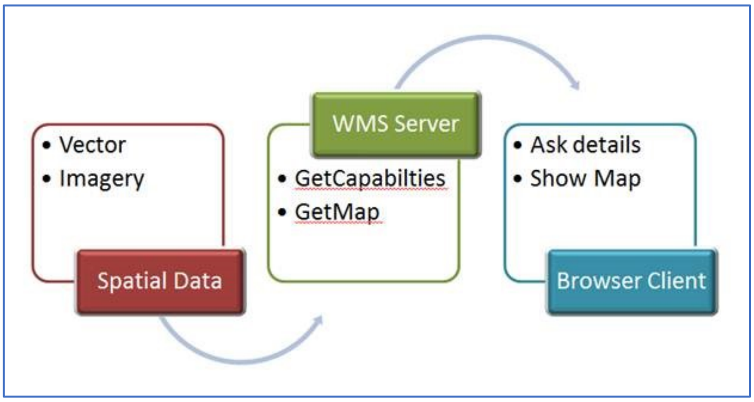

FIGURE 2 - OGC WEB MAP SERVICE (WMS)[2]

The Styled Layer Descriptor (SLD) standard is an extensions to the WMS specification to allow user-defined symbolization of feature. These are defined in OpenGIS Styled Layer Descriptor Profile of the Web Map Service Implementation Specification, Version 1.1.0

WMS provides a standard interface for requesting a geospatial map image. The benefit of this is that WMS clients can request images from multiple WMS servers, and then combine them into a single view for the user. The standard guarantees that these images can all be overlaid on one another as they actually would be in reality. Numerous servers and clients support WMS.

WMS requests can perform the following operations:

Operation	Description
GetCapabilities	Retrieves metadata about the service, including supported operations and parameters, and a list of the available layers The GetCapabilities operation requests metadata about the operations, services, and data (“capabilities”) that are offered by a WMS server.
GetMap	Retrieves a map image for a specified area and content

[2] http://www.web-maps.com/gisblog/?cat=19

GetFeatureInfo (optional)	Retrieves the underlying data, including geometry and attribute values, for a pixel location on a map
DescribeLayer (optional)	Indicates the WFS or WCS to retrieve additional information about the layer.
GetLegendGraphic (optional)	Retrieves a generated legend for a map
Exceptions	If an exception occur

2.1.4 OGC:WFS - Web Feature Service

The Standard Web Feature Service (WFS), similarly to the WMS, provides a simple HTTP interface to directly request geographic objects (and not map images) from one or more distributed servers over the Internet. The demand and response mechanisms are similar to WMS, with the difference that no images are returned, but the descriptions of the individual spatial objects contained within the area of interest to be processed (spatial coordinates and any alphanumeric attributes).

The Web Feature Service (WFS) is a standard created by the Open Geospatial Consortium (OGC) for creating, modifying and exchanging vector format geographic information on the Internet using HTTP. A WFS encodes and transfers information in Geography Markup Language (GML), a subset of XML.

The WFS standard defines the framework for providing access to, and supporting transactions on, discrete geographic features in a manner that is independent of the underlying data source. Through a combination of discovery, query, locking, and transaction operations, users have access to the source spatial and attribute data in a manner that allows them to interrogate, style, edit (create, update, and delete), and download individual features. The transactional capabilities of WFS also support the development and deployment of collaborative mapping applications.

The current version of WFS is 2.0.0 and define eleven operations.

Operation	Description
GetCapabilities	Generates a metadata document describing a WFS service provided by server as well as valid WFS operations and parameters
DescribeFeatureType	Returns a description of feature types supported by a WFS service
GetFeature	Returns a selection of features from a data source including geometry and attribute values
LockFeature	Prevents a feature from being edited through a persistent feature lock
Transaction	Edits existing feature types by creating, updating, and deleting
GetPropertyValue	Retrieves the value of a feature property or part of the value of a complex feature property from the data store for a set of features identified using a query expression

GetFeatureWithLock	Returns a selection of features and also applies a lock on those features
CreateStoredQuery	Create a stored query on the WFS server
DropStoredQuery	Deletes a stored query from the WFS server
ListStoredQueries	Returns a list of the stored queries on a WFS server
DescribeStoredQueries	Returns a metadata document describing the stored queries on a WFS server

2.1.5 OGC:WCS - Web Coverage Service

The Web Coverage Service (WCS) is a standard created by the OGC that refers to the receiving of geospatial information as “coverages”: digital geospatial information representing space-varying phenomena. One can think of it as Web Feature Service (WFS) for ***raster*** data. It gets the ‘source code’ of the map, but in this case its not raw vectors but raw imagery.

An important distinction must be made between WCS and Web Map Service (WMS). They are similar, and can return similar formats, but a WCS is able to return more information, including valuable metadata and more formats. It additionally allows more precise queries, potentially against multi-dimensional backend formats.

WCS provides a standard interface for how to request the raster source of a geospatial image. While a WMS can return an image it is generally only useful as an image. The results of a WCS can be used for complex modeling and analysis, as it often contains more information. It also allows more complex querying - clients can extract just the portion of the coverage that they need.

WCS can perform the following operations:

Operation	**Description**
GetCapabilities	Retrieves a list of the server’s data, as well as valid WCS operations and parameters. The GetCapabilities operation is a request to a WCS server for a list of what operations and services (“capabilities”) are being offered by that server.
DescribeCoverage	Retrieves an XML document that fully describes the request coverages.
GetCoverage	Returns a coverage in a well known format. Like a WMS GetMap request, but with several extensions to support the retrieval of coverages.

2.1.6 OGC:STA – SensorThings API

Sensor data have much to do with geographic data since they most of them comes from stations that have a specific location on the surface of Earth and/or refers to a specific area, point or, generally speaking, to a spatial feature. In situ observation data falls into this category as well as remote sensing data are very similar to satellite or aerial surveys images, even if the time dimension plays a very important role and so does the multidimensionality of the observed data.

There are also some kind of observations that are more difficult to assimilate to common geographic data, like data registered by moving sensors (like sounding balloon or ships or other vehicles on which sensors can be mounted) or remote sensing data that refers to volumes instead of surfaces.

For this reason, OGC has tackled the topic of sensor data in the larger context of geographic data, assembling a suite of standards called Sensor Web Enablement (SWE) and has tried to harmonize common concepts and to use the same protocols and encodings used for the better-known spatial services.

This suite has first hosted a conceptual model for observations and forecasts called O&M (Observations and Measurements) that become a ISO abstract specification and then a XML implementation that has been used to code the entities transferred by the Sensor Observation Service (SOS). Together with O&M, another coding specification has been released to describe sensors and, generally, computational processes used in a previous or in a following phase of the measurement.

The formal definition of SOS was defined likewise WMS, WFS or the other OWS (OGC Web Services), thus having a SOAP binding, an XML payload, a GetCapabilities operation for self-description and several other different operations.

SensorThings API (STA) is the evolution of the OGC:SOS Sensor Observation Service that addresses data access for the **Internet of Things** (IoT).

The vision of IoT is that of devices all over the world directly connected to the Internet to allow data retrieval and control.

This OGC standard is actually divided into two parts: the former dealing with **access** to data (Sensing Profile) and the latter is about **control** of devices (Tasking Profile) and in the context of Mistral we can ignore it.

SensorThings API provides a RESTful, JSON encoded API to retrieve data and metadata about 'things' that generate streams of data. The underlying information/resource model for the API uses Observation and Measures (O&M) and has been influenced by the other Sensor Web Enablement (SWE) standards. The API follows patterns defined by the OData protocol. Table 1 shows the main differences between the previous standard SOS.

Feature	SensorThing	SOS

Encoding	JSON	XML
Architectural Style	Resource Oriented Architecture	Service Oriented Architecture
Binding	REST	SOAP
Pagination	$top/$skip/$next Link	Not supported
Pub/Sub Support	MQTT and SensorThings MQTT Extension	Not supported
OGC model link	Location entity	Confusion between feature and feature of interest
Insert new Sensors and Observations	HTTP POST	SOS specific interface: RegisterSensor() and InsertObservation()
Deleting Existing Sensors	HTTP DELETE	SOS specific interface: DeleteSensor()
Updating Properties of Existing Sensors or Observations	HTTP PATCH and JSON PATCH	Not supported
Deleting Existing Observations	HTTP DELETE	Not supported
Linked Data Support	JSON-LD	Not supported

TABLE 1 - MAIN DIFFERENCES BETWEEN STA AND SOS

SensorThings splits the O&M model across two classes: Datastream and Observation. Datastreams are the top-level class with a subset of the O&M Observation properties: observedProperty, resultTime, phenomenonTime. The observations property is analogous to result in the OM_Observation type, but in this case contains a set of Observation objects. Each of these is typed according to its observation type from O&M (Measurement, Geometry etc.).

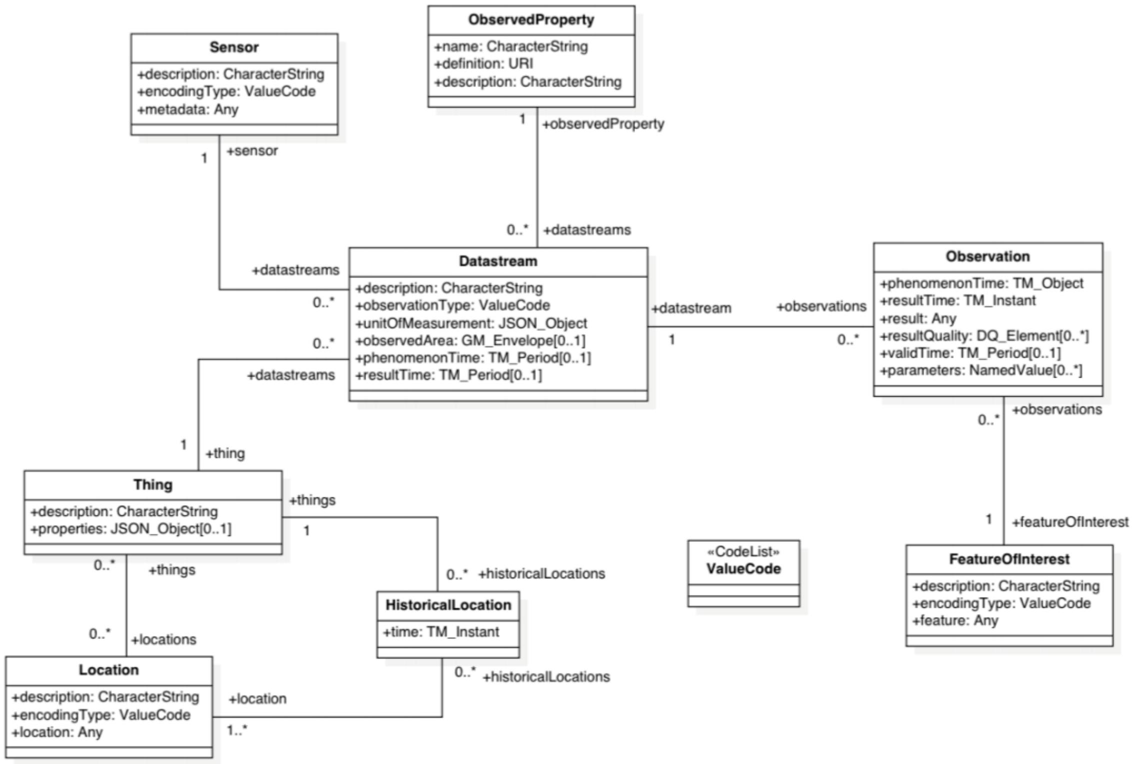

FIGURE 3 - SENSORTHINGS RESOURCE MODEL

The structure of the JSON encoding follows patterns defined by OData. It does not strictly follow the OData specification, but reuses the common patterns, identifier, and link structures.

Key characteristics:

1. Uses OData as a pattern for response encoding and link handling.
2. Generic IoT problem space – applicable to most sensor applications.
3. Light metadata.Linked resources.
4. Implicit CRS

Standards and Services in Mistral platform

In this section we will examine how the various standard services will be used inside Mistral platform to achieve its goals of data dissemination.

2.1.7 CKAN data portal platform

The publication of data takes place through the open source web portal CKAN (Comprehensive Knowledge Archive Network). CKAN is one of the reference software for the publication of both institutional and governmental Open Data, such as municipalities and Regions, both governmental and private, being widely used also by companies that make available to their users information related to the functioning of the services provided and the work done (transport, citizen services, utilities in general).

The CKAN portal is structured in the following integrated components:

1. The backend (Python), which manages the graphic part, the integration between the various components, the authentication and the publication of the API;
2. the PostgreSQL database, which contains the published metadata, references and datasets (where present);
3. the search engine (Apache Solr), which queries the database based on the datasets requested by the user, both through predefined identifiers (tags, groups, organizations) and through full-text searches. In particular,

The CKAN portal of the Mistral project is used to index the data collected and produced by the MISTRAL project, but also other metereological data available in Italy has been described.

Most of these do not reside on CKAN itself, being accessible through Meteo-Hub portal specialized in providing large amounts of data, applying filtering and post processing. CKAN therefore contains links to these data.

2.1.8 WMS/WFS OGC Implementation GeoServer

GeoServer is an open source software server written in Java that allows users to share and edit geospatial data through OGC Standard Services like WMS (Web Map Service) or WFS (Web Feature Service).

In Mistral project will be implemented a prototype of WMS and WFS with Geoserver dealing with observed temperature.

2.1.9 SkinnyWMS

SkinnyWMS will be used to improve the present “slideshow” manner used to visually evaluate the results of the forecast elaboration chain of LAMI models. An hourly geographic representation of the forecasted value of each single main weather parameter is produced for each of the 72 hours that follow the model run time. These images, though representing a geographic area, are not georeferenced and contain the country borders merged in each of them. They can be shown sequentially in a sort of animation, and even a kind of zoom/pan interaction is possible on the image, but this is the only “web map-like” functionality available, because you can’t overlap them with any other representation of geographic data, since they are not transparent.

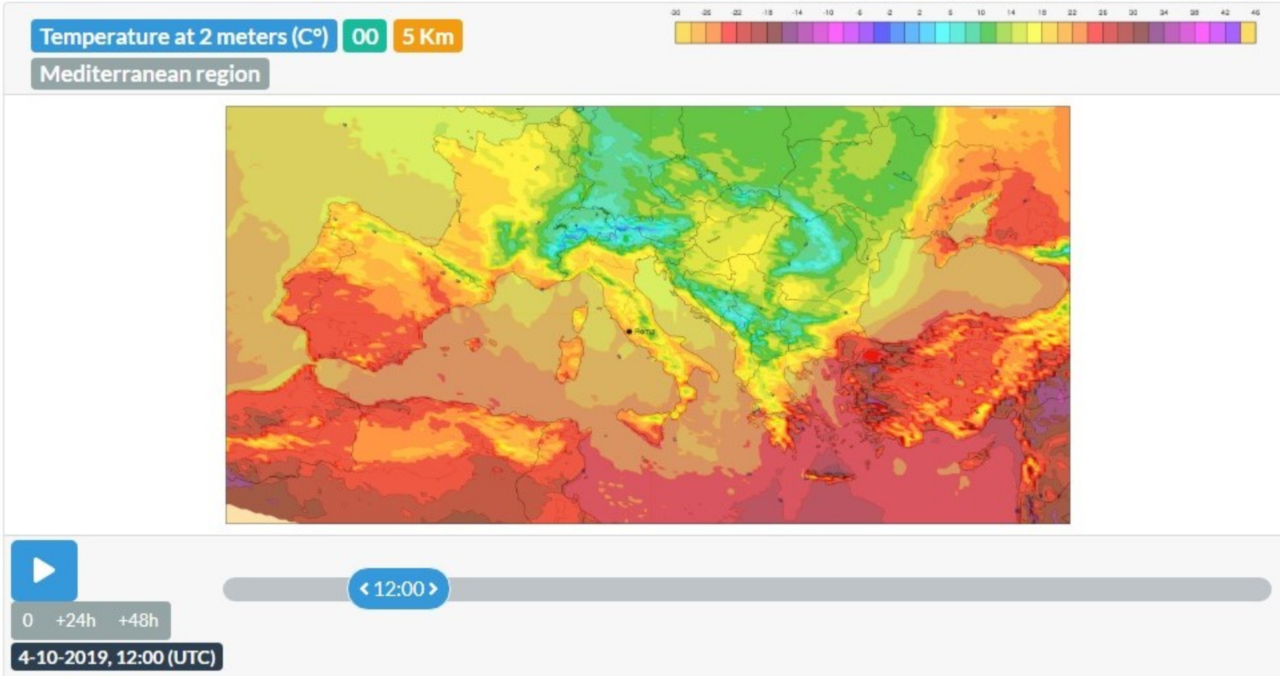

FIGURE 4 - THE CURRENT "SLIDESHOW" APPLICATION TO VIEW FORECAST MAP

Let's take the temperature at 2 meters forecast image of Figure 4; the original value of each pixel that represented the forecasted temperature for that specific time in that specific location has been mapped to an (R, G, B) triple that represents the colour that is be used in the representation of the image.

The real-world coordinates of the pixel cannot be inferred from the image. Furthermore, being a single image, it is necessary to reach a compromise between file size and image resolution.

To overcome this limitation SkinnyWMS is used to generate the georeferenced tiles.

The skinny WMS[3] is a small WMS server that will help you to visualise your NetCDF and Grib Data. The principle is simple: skinny will browse the directory, or the single file passed as argument, and try to interpret each NetCDF or GRIB files. From the metadata, it will be built the getCapabilities document, and find a relevant style to plot the data.

SkinnyWMS implements 3 of the WMS endpoints:

4. **getCapabilities**: Discover the data, build an XML Document presenting each identified parameter in the file(s) as a layer with the list of their predefined styles. (There is always a default style)
5. **getMap** : Return the selected layer suing the selected style.

[3] https://confluence.ecmwf.int/display/MAGP/Skinny+WMS

6. **getLegendGraphic**: Return the legend.

SkinnyWMS depends on the ECMWF *Magics* library.

SkinnyWMS will be used in Mistral to generate tiles, instead of single images of the main forecast parameters.

2.1.10 FROST implementation of the OGC SensorThings

FROST-Server (**FRaunhofer Opensource SensorThings Server**) is an **open-source implementation of the standard**, developed by the German research institute Fraunhofer IOSB, to cover their need for a standards-based, easy to use sensor management and sensor-data storage platform, for use in various research programs.

It is written in the **Java** programming language and can be deployed on **servlet** containers such as Apache Tomcat.

For data persistence, it currently has backends for the **PostgreSQL database** management system, with either numeric, string or UUID entity identifiers.

Since it was developed to cover the wide range of use-cases that appear in research projects, the focus of development was on feature completeness and extendibility.

The server implements the complete specification of the OGC SensorThings API Part 1: Sensing, including all extensions.

Experimentation of OGC SensorThings API FROST Server using CSV file sample of observed temperature. The csv file sample has been provided by Arpa Piemonte and represent one of the inputs of Multi Model Forecast.

Some Pentaho scripts has been implemented as described below:

1. Take as input the CSV data file (station data + observations)
2. A Pentaho script processes an XML file of the Stations or of the Things having as input the file in point 1)
3. A Pentaho script processes an XML file of the Observations which must be loaded together with the Things having as input the file in point 1)
4. An XSLT processing transforms the XML files into json files for FROST (Things and Observations) - made with XML Spy
5. A Pentaho script uploads the json files of Things to FROST
6. A Pentaho script uploads the Observation json files to FROST
7. The dashboard shows the stations loaded on the FROST server and the data

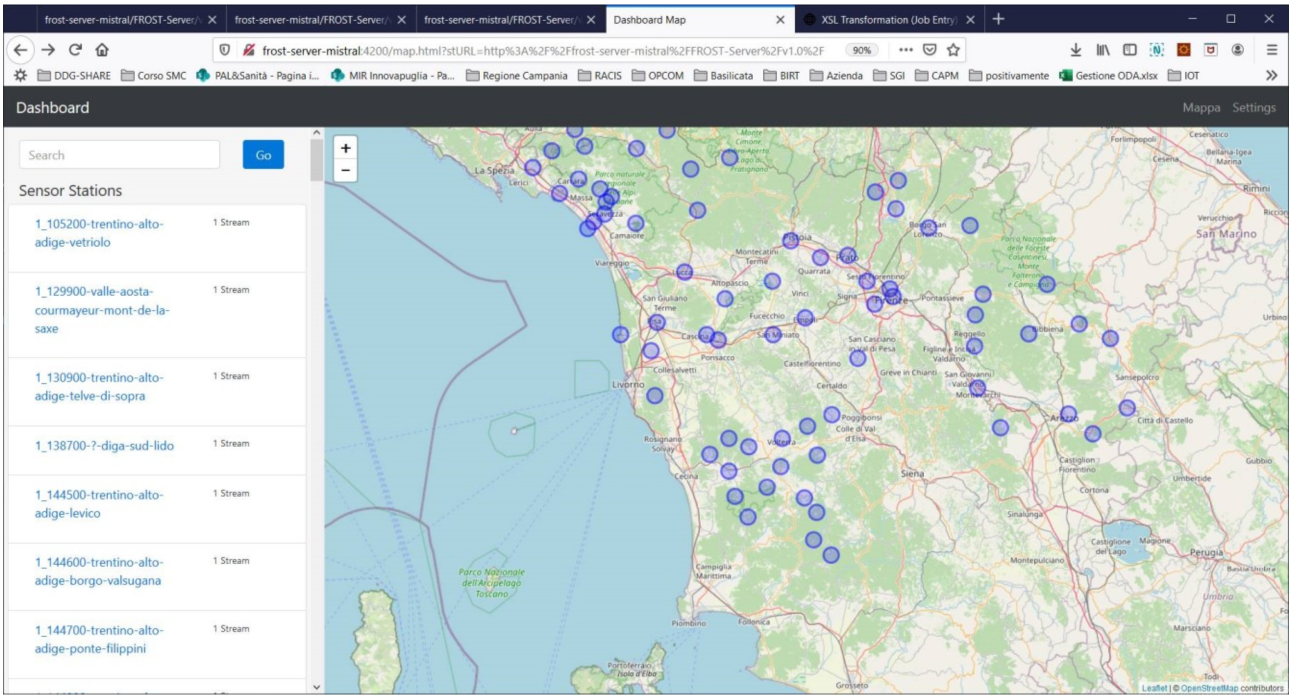

FIGURE 5 – FROST DASHBOARD – SENSOR STATIONS

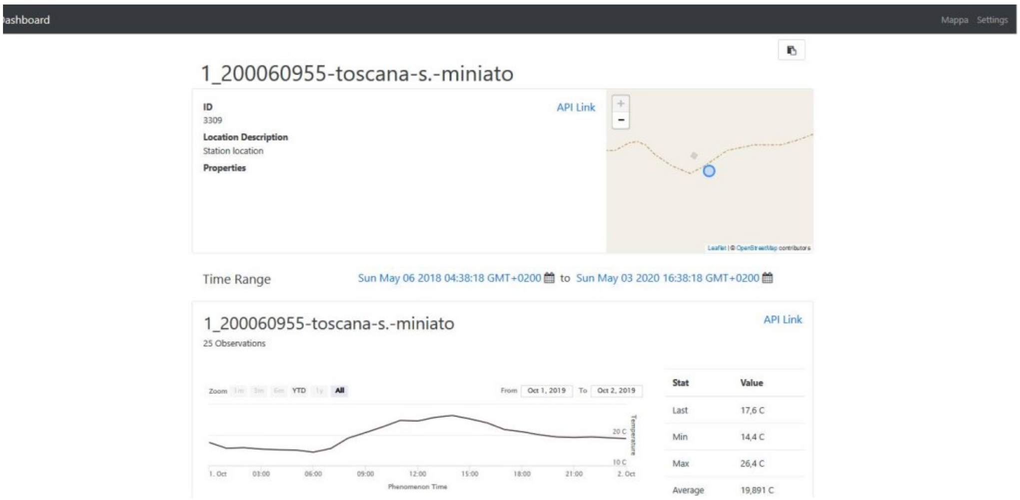

FIGURE 6 - FROST DASHBOARD – TEMPERATURE TIME SERIES

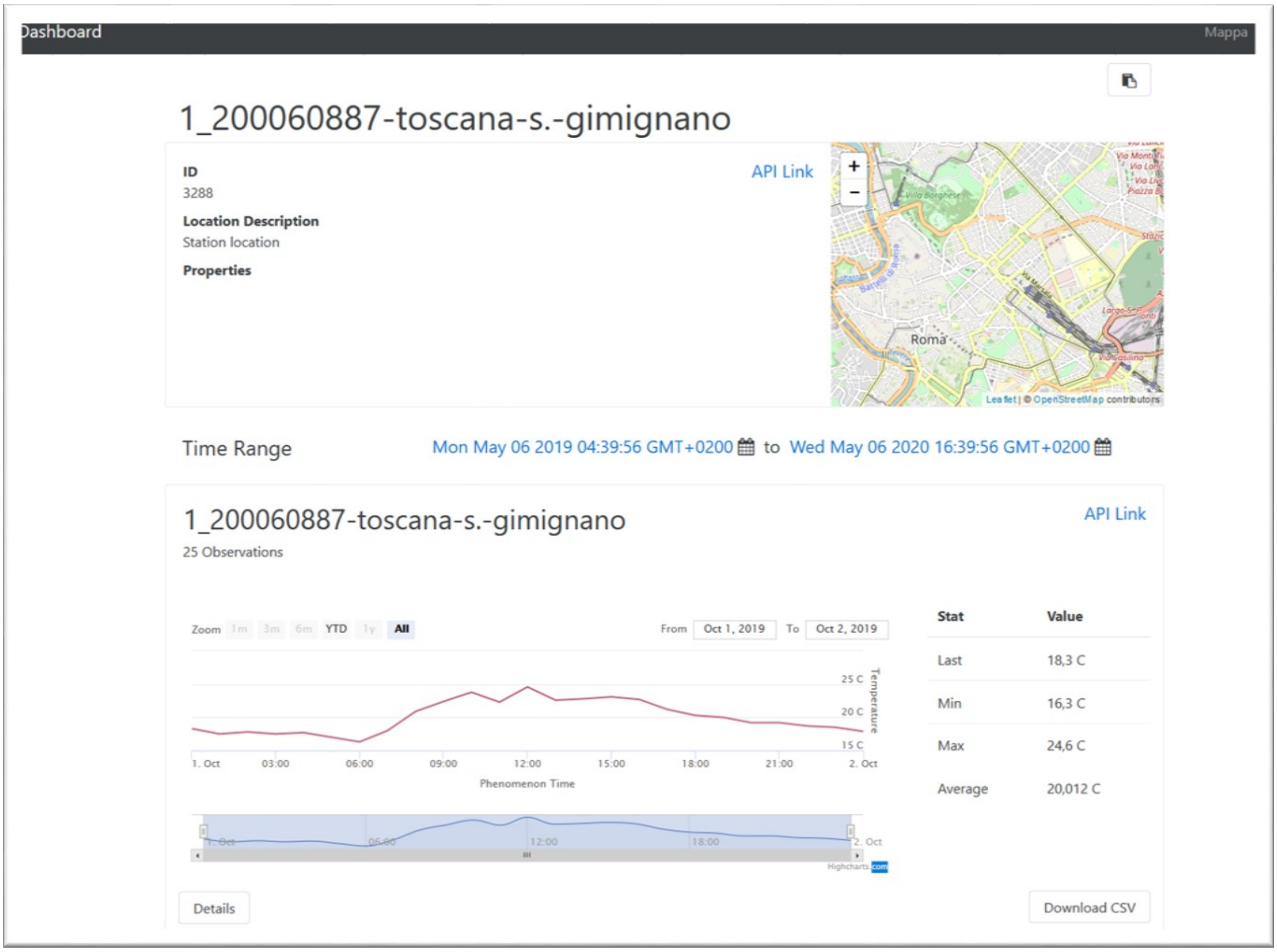

FIGURE 7 - FROST DASHBOARD - TEMPERATURE TIME SEIRES

ANNEXES

1 Annex-1-Mistral_D4.3-analisi-preliminare.pdf
2 Annex-2-Mistral_D4.3-licenze-inquadramento-generale.pdf
3 Annex-3-Mistral_D4.3-licenze-scenari-e-soluzioni.pdf
4 Annex-4-Mistral_D4.3-privacy.pdf