PUBLICAMUNDI

SCALABLE
REUSABLE
OPEN
GEOSPATIAL
DATA

REPORT FOR

DELIVERABLE D2.1

1 INTRODUCTION

This report provides an overview the technical characteristics and functionality of deliverable D2.1 “Core CKAN Extensions”. Its purpose is to provide a short introduction to the software and present the major functionalities and improvements introduced by the PublicaMundi project. In particular:

- We first provide an overview of CKAN, the data catalogue software which is extended in the project with advanced geospatial capabilities.
- In the following we highlight our contributions in CKAN’s core code base. Following the best practice examples set by the community and leading data catalogues (e.g. data.gov, data.gov.uk) we strived to minimize changes in the core CKAN and introduce our work as CKAN extensions. In this manner, we maintain interoperability and backwards compatibility with existing CKAN catalogues, minimizing the effort for integrating PublicaMundi technologies.
- In the final section, we provide an overview of our ongoing work regarding the CKAN extensions developed by PublicaMundi. Their final versions will be available in deliverable D2.1.

The reader is encouraged to visit our project repository (https://github.com/PublicaMundi) to receive:

- Up-to-date versions of the software, along with documentation targeted to developers
- Detailed information regarding all development effort (commits, activity, issues)
- Instructions regarding the installation of the software and its dependencies

2 CKAN

CKAN is an open-source data hub software package, written in Python, initially developed by the non-profit Open Knowledge Foundation and currently developed under the CKAN Foundation. It is used to power local, national and supranational open government data portals around the world, as well as community data hubs in various countries.

Examples are the UK's data.gov.uk, US's data.gov (Fig 1) and the European Union's publicdata.eu (Fig 2), as well as many others. Community instances such as the DataHub (thedatahub.org) allow anyone to publish data for free.

CKAN is a fully-featured, mature, open source data management solution. CKAN provides a streamlined way to make open data discoverable and presentable. Each dataset is given its own page with a rich collection of metadata, making it a valuable and easily searchable resource.

CKAN is built with Python on the backend and JavaScript on the frontend, and uses the Pylons web framework and SQLAlchemy as its ORM. Its database engine is PostgreSQL and its search is powered by SOLR. It has a modular architecture that allows extensions to be developed to provide additional features such as harvesting or data upload. CKAN uses its internal model to store metadata about the different records, and presents it on a web interface that allows users to browse and search this metadata. It also offers a powerful API that allows third-party applications and services to be built around it.

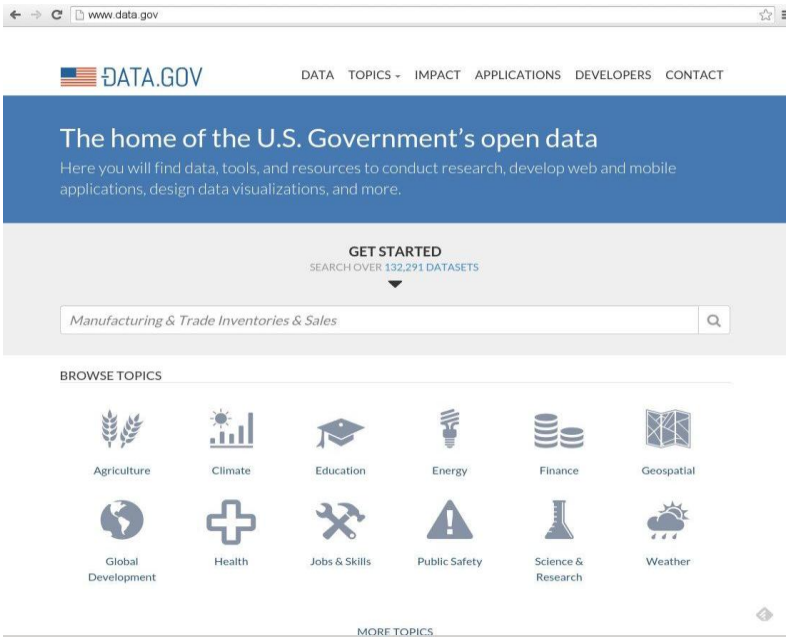

Figure 1: US's data.gov CKAN implementation and theme

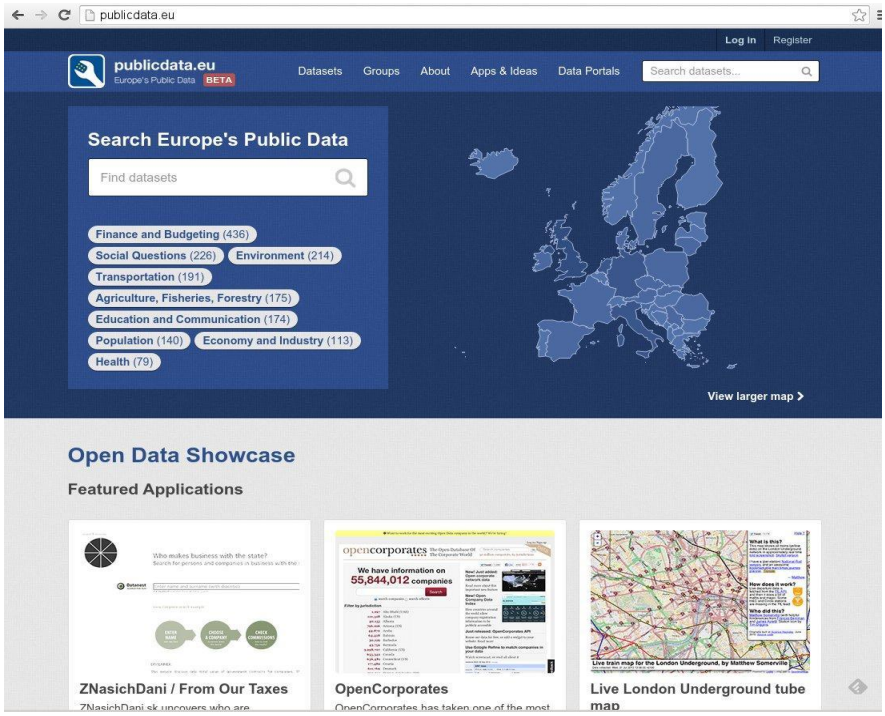

Figure 2: publicdata.eu CKAN implementation and theme

CKAN provides an intuitive web interface enabling dataset publishers and curators to register, update and refine datasets in a distributed authorisation model called 'Organizations'. 'Organizations' allow each publisher to have their own dataset entry and approval process with numerous members. This means responsibility can be distributed and authorization access managed by each department or agencies' admins instead of centrally.

3 PUBLICAMUNDI CONTRIBUTIONS

PublicaMundi aims to make open geospatial data easier to discover, reuse, and share by fully supporting their complete publishing lifecycle in open data catalogues. To achieve this, we are extending and integrating CKAN, the leading data catalogue, into treating geospatial data as first-class citizens and providing OCG and INSPIRE compliant access to geospatial services.

A first integrated prototype is already available on labs.geodata.gov.gr, providing beta access to data publishers and developers for Greek open geospatial data.

As described in the previous chapter, CKAN is a Pylons/Python application, providing a mechanism to add functionality through extensions. This way the core application is improved but at the same time the source code is easier to be maintained through mainstream code revisions (releases).

Our main objective in PublicaMundi is our software to be reusable and extensible, while maintaining our goal to spatially enable CKAN. Throughout the first year of development, our architecture was focused on maintaining compatibility with upstream CKAN core, so that our software would be easily applicable not only to geodata.gov.gr, but also on other CKAN deployments, without breaking existing functionality. Also, a main goal was to able to easily port PublicaMundi's source code to future versions of CKAN, thus we needed to be able to backport changes and bug fixes from upstream (core CKAN development git repository).

In order to achieve these goals, we followed the example of successful projects that were based on CKAN in the recent past and have been successfully maintained ever since.

The good practice in extending CKAN functionality was derived from USA's data.gov, maintained by GSA (USA General Systems Administration) on GitHub (https://github.com/GSA/ckan). In that project, the development team has created a separate copy of CKAN source code (*fork* in open source terms and on GitHub), in order to have control over the source code that gets merged from development source tree into the production code. Towards this, the data.gov development team has initially selected the most recent version of CKAN (version 2.1) and has been maintaining it ever since, applying bug fixes and backporting stable code into a "release" branch (https://github.com/GSA/ckan/tree/release-datagov).

The same strategy was applied in PublicaMundi project. We copied the source code of CKAN into our public repository and selected the most recent stable version as our core system, in order to build our extensions (CKAN version 2.2). Since the day that we "forked" the CKAN project on GitHub (https://github.com/PublicaMundi/ckan) a new stable minor release has happened (version 2.2.1), which we applied on our repository and tested extensively for possible new bugs. All the bugs we located were reported back to the OKFN upstream CKAN project and we were able to fix some of them and contribute back our fixes through GitHub "Pull Requests".

Close to the end of the first year, we decided to freeze our source code and we created a production branch, similar to data.gov, which we named "labs.geodata.gov.gr":

- https://github.com/PublicaMundi/ckan/tree/labs.geodata.gov.gr

while we are still maintaining a development branch named "dev.publicamundi.eu":

- https://github.com/PublicaMundi/ckan/tree/dev.publicamundi.eu

When the source code is stable enough and well tested, those two branches get merged, a new stable version gets released and the development cycle continues like that.

For the Deliverable 2.1 (D2.1), we have prepared a source code archive of the most recent stable release, which is now named "D2.1" on our GitHub repository (https://github.com/PublicaMundi/ckan/tree/ckan-D2.1) and is based on CKAN 2.2.1 release. The source code is available from the following link:

- https://github.com/PublicaMundi/ckan/archive/ckan-D2.1.zip

3.1 PUBLICAMUNDI CKAN EXTENSION

Following the best practice example from data.gov, we have developed a CKAN extension in order to provide the core functionality needed for the purposes of our project: https://github.com/PublicaMundi/ckanext-publicamundi.

Our main contributions to the CKAN functionality affect the following areas:

- Publication and management of data
- Search and discovery of data
- Publication and management of metadata
- Geospatial support

- Visualization of data
- Customization of UI
- Data storage
- Extensions
- Federation and harvesting

A detailed description of our contributions to CKAN through our ckanext-publicamundi extension is provided in the next sections.

3.1.1. Publication and management of data

We have extended CKAN with an easy to use and flexible data publishing workflow for geospatial data, supporting vector and raster data, in addition to the majority of standardized metadata schemata. Data publishers are guided step-by-step into creating metadata in their schema of choice (Figures 3,4,5), or importing existing metadata. The metadata they provide can then be transformed on-the-fly in any supported schema. If an existing data catalogue or Spatial Data Infrastructure is available (e.g. INSPIRE), the data publisher only needs to provide a simple entry point, and all available metadata are automatically harvested.

Our data publishing workflow significantly lowers the entry barrier for data publishers, while also accommodating the data publishing needs of organizations with existing data catalogues and SDIs, maintaining full compatibility with INSPIRE and OGC web services.

For the management of data resources we have implemented a new administrator dashboard, which adds capabilities of creating new derivative resources (e.g. create a WMS resource from a GML file) (Figure 6)

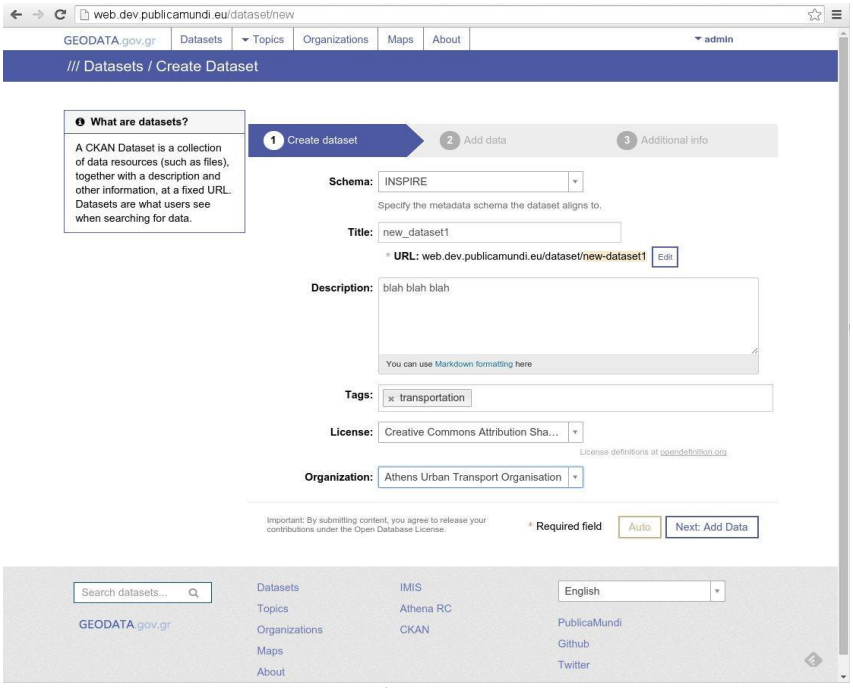

Figure 3: The PublicaMundi publishing workflow to create a new dataset with INSPIRE metadata.

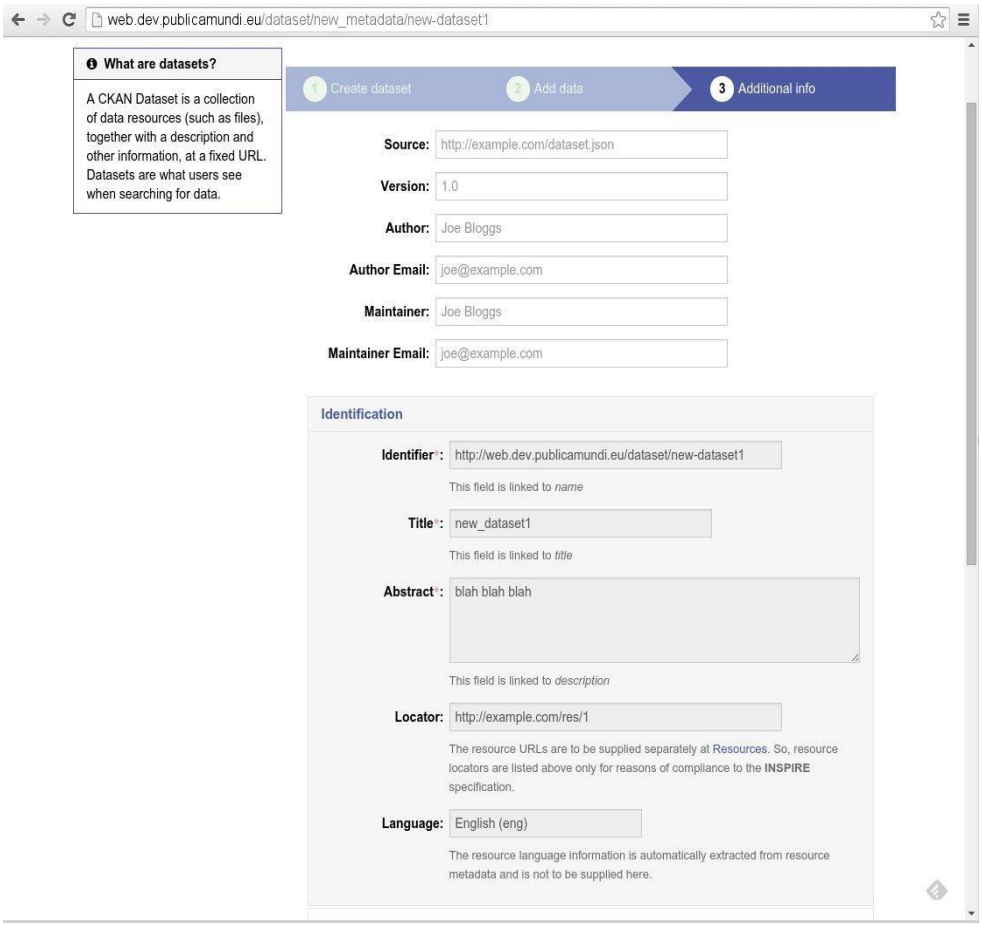

Figure 4: The PublicaMundi metadata editor

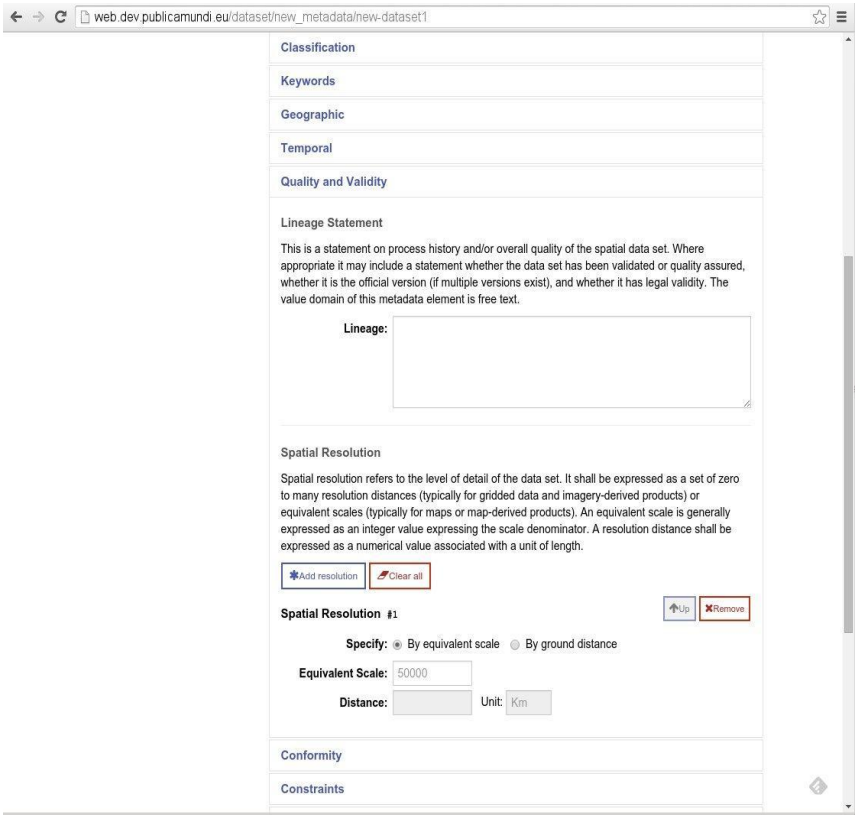

Figure 5: The PublicaMundi metadata editor (INSPIRE schema).

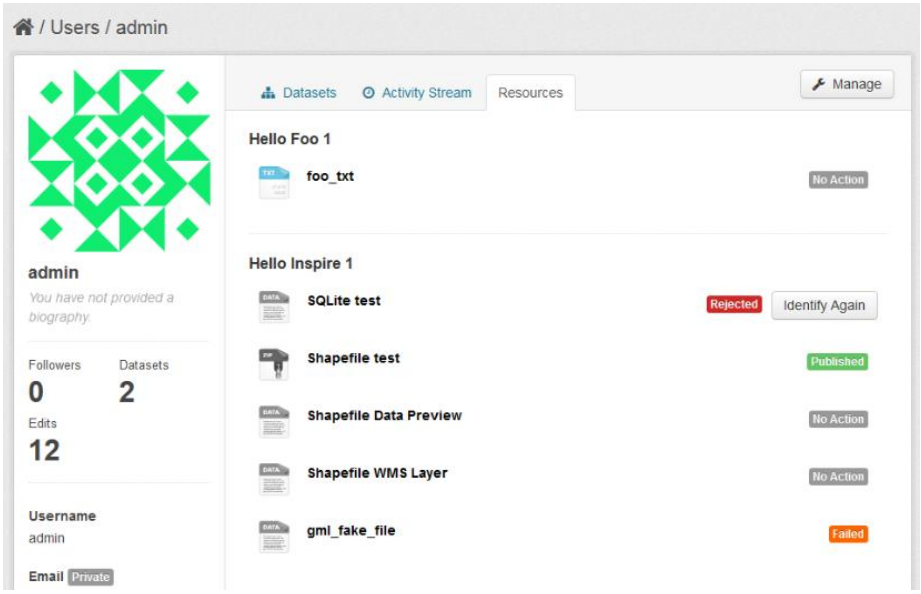

Figure 6: New CKAN administrator dashboard, for geospatial data ingestion

3.1.2. Search and discovery of data

We have integrated pycsw in CKAN for providing OGC compliant catalogue services across all published geospatial data. pycsw has been extended and improved in terms of scalability, harvesting, and metadata schema support. Our contributions have been integrated in its codebase and power the geospatial catalogue services of data.gov. Also, pycsw has been extended

to support the new OGC OpenSearch Geo/Time specification, towards implementation of CSW 3.0. pycsw was forked under our repository in order to maintain the current stable version in the production system (https://github.com/PublicaMundi/pycsw).

The base version of pycsw used in labs.geodata.gov.gr is 1.10.0 and has been also released here as D2.1 release:

- https://github.com/PublicaMundi/pycsw/tree/pycsw-D2.1

The source code is available from the following link:

- https://github.com/PublicaMundi/pycsw/archive/pycsw-D2.1.zip

The pycsw 1.10.0 version (2014-09-13) brings significant features, enhancements and fixes to the codebase, including:

- support OGC OpenSearch Geo and Time Extensions standard
- support for Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)
- support spatial dateline queries
- better support for geospatial links via the Python geolinks library
- fix CSW harvesting to harvest full records
- various fixes to OGC Filter handling
- ISO harvesting: capture all keywords / keywords sets
- ISO harvesting: support gmd:distributorTransferOptions
- GetRecords: fix maxrecords casting
- fix limit / offset queries for Django-based backends
- freetext searching: make non PostgreSQL FTS-based queries more like FTS
- fix date handling for ISO output
- add libxml2 version output in admin tools
- return distributed CSW queries in order
- support ISO-based CSW metadata harvesting

3.1.3. Publication and management of metadata

We have extended the CKAN metadata model for flexible metadata creation/editing. Publishers have several options for creating metadata aiming to minimize effort and maximize reusability. They can create metadata for several supported metadata schemata (e.g. CKAN, INSPIRE,

ISO), or import existing metadata files. Following validation, metadata are stored and can be transformed on demand in other schemata through CSW interface. This mechanism is programmatically extensible to support any other metadata schema (geospatial or not). This is a core extension that is implemented within ckanext-publicamundi, using zope interfaces, with capabilities of automatic UI creation, validation and export of metadata files. Figure 5 shows auto-generated UI for editing INSPIRE metadata, after definition of the INSPIRE zope schema.

3.1.4. Geospatial support

We have extended CKAN to natively support geospatial vector data management, by integrating PostGIS, the leading open source geospatial database. Data publishers can upload geospatial data in any format and coordinate reference system. The system automatically stores the dataset and can provide it in another data format (on-demand) or through OGC compatible services. As such, data publishers can provide any data they have at hand, without additional effort into transforming their data in specific-purpose formats. Further, as soon as the data is published, they are automatically available for querying and visualization with no extra effort.

We have also extended CKAN to natively support raster data management, by integrating rasdaman, the leading big raster data analytics server. Data publishers can publish any raster data they have available, ranging from a single satellite image, to thousands of orthomaps. The raster data are efficiently managed and are available for querying and visualization with no extra effort.

Finally, we have extended CKAN to support complex and OGC-compliant geospatial processing capabilities, by extending ZOO-WPS, the leading open source OGC WPS server. As soon as vector and raster data have been uploaded by the data publishers in the catalogue, these can be automatically reused through standards-compliant WPS services.

3.1.5. Visualization of data

As part of the CKAN spatial extension, we have integrated our Mapping API in order to provide data previewers for geospatial data. Those have been integrated in our ckanext-spatial repository and will be part of the Deliverable 2.2 (to be delivered in M18). We are including a beta version of the extension in our D2.1deliverable (included as ckanext-spatial-D2.2beta.zip).

Also, as part of Deliverable 2.1 we have integrated a map client application in CKAN (under ckanext-publicamundi extension). This is not only a

previewer but a full JavaScript client that can access external sources and services.

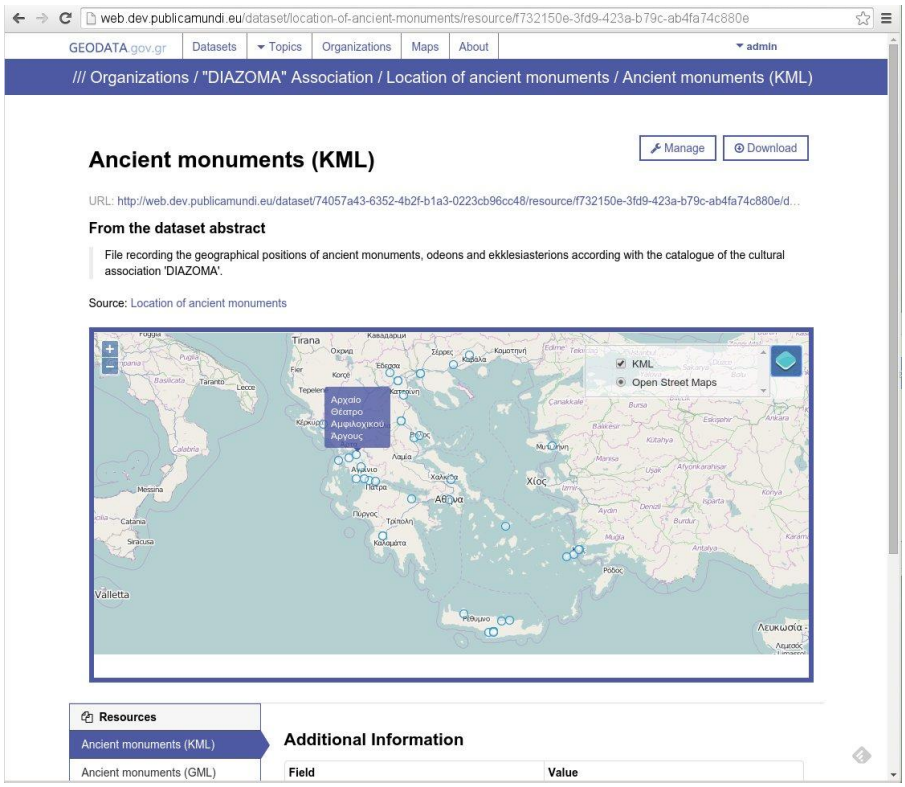

Figure 7: Geospatial data previewer based on PublicaMundi Mapping API

3.1.6. Customization of UI

Under Deliverable 2.1 we have implemented a new CKAN theme, to be released as an official geodata.gov.gr theme (Figure 7). This new theme is integrated in ckanext-publicamundi as a theme plugin. It is available in the ckanext-publicamundi-D1.1.zip

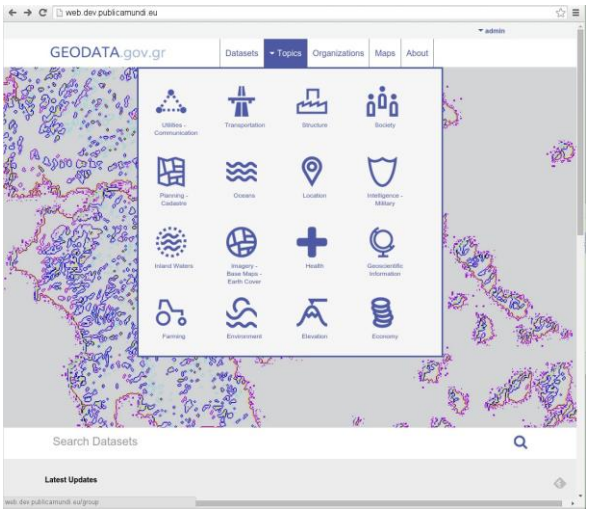

Figure 8: New theme for geodata.gov.gr, available in ckanext-publicamundi extension

3.1.7. Data storage

We have implemented ckanext-vectorstorer and ckanext-rasterstorer that provide capabilities to store vector and raster data directly into the CKAN

database, through ckanext-datastorer extension. The majority of GDAL/OGR data formats are supported, requiring no transformation. Further, data are then automatically available through the various supported APIs.

3.1.8. Extensions

We have implemented various CKAN extensions. First of all ckanext-publicamundi provides the core publishing functionality. Aditionally we have implemented ckanext-vectorstorer and ckanext-rasterstorer that provide capabilities to store vector and raster data, as well as publish them through OGC Web Services. Other extensions include the DataAPI, MappingAPI, Map Client Application and ckanext-WPS, that are part of the next deliverable D2.2.

3.1.9. Federation and harvesting

We have improved CKAN harvesting capabilities through contributions made to OWSLib, pycsw and CKAN spatial extension. Where existing data catalogues or SDIs are available, the publisher only needs to provide the relevant catalogue end-points. Harvesting is performed automatically, fetching all required metadata. Also, the CKAN catalogue now supports harvesting of INSPIRE XML, OpenDataJSON and ArcGIS JSON metadata.