D4.1 - PRESENTATION LAYER: TECHNICAL SPECIFICATIONS REPORT

Co-funded by the Horizon 2020
Framework Programme of the European Union

DELIVERABLE NUMBER	D4.1
DELIVERABLE TITLE	Presentation Layer: Technical Specifications Report
RESPONSIBLE AUTHOR	Athanasios Ballis(UoA)

GRANT AGREEMENT N.	731001
PROJECT ACRONYM	AGINFRA PLUS
PROJECT FULL NAME	Accelerating user-driven e-infrastructure innovation in Food & Agriculture
STARTING DATE (DUR.)	01/01/2017 (36 months)
ENDING DATE	31/12/2019
PROJECT WEBSITE	www.plus.aginfra.eu
COORDINATOR	Nikos Manouselis
ADDRESS	110 Pentelis Str., Marousi GR15126, Greece
REPLY TO	nikosm@agroknow.com
PHONE	+30 210 6897 905
EU PROJECT OFFICER	Mrs. Georgia Tzenou

WORKPACKAGE N. | TITLE	WP4 | Presentation Layer
WORKPACKAGE LEADER	UoA
DELIVERABLE N. | TITLE	D4.1 | Presentation Layer: Technical Specifications Report
RESPONSIBLE AUTHOR	George Kakaletris (UoA)
REPLY TO	gkakas@di.uoa.gr
DOCUMENT URL	http://www.plus.AGINFRA.eu/sites/plus_deliverables/D4.1.pdf
DATE OF DELIVERY (CONTRACTUAL)	30 June 2017 (M6), 30 June 2018 (M18, Updatedversion)
DATE OF DELIVERY (SUBMITTED)	31July 2017 (M7), 03 July 2018 (M18,Updatedversion)
VERSION | STATUS	V2.0 | Final
NATURE	R (Report)
DISSEMINATION LEVEL	PU(Public)
AUTHORS (PARTNER)	George Kakaletris (UoA) Dimitris Katris (UoA) Athanasios Ballis (UoA)
REVIEWER	Leonardo Candela (CNR)

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Deliverable Structure	29/03/2017	Athanasios Ballis (UoA)
0.2	Content added	20/06/2017	Athanasios Ballis (UoA)
0.3	Chapter Restructure	14/07/2017	George Kakaletris (UoA)
0.4	Content added	17/07/2017	Athanasios Ballis (UoA)
0.5	Content added	21/07/2017	George Kakaletris (UoA)
0.6	Pre-final draft for review	24/07/2017	Athanasios Ballis (UoA)
0.7	Comments from internal reviewer	28/07/2017	Leonardo Candela (CNR)
1.0	Finalization of D4.1 1st version	31/07/2017	George Kakaletris (UoA) Athanasios Ballis (UoA)
1.1	Pre-final draft for 2nd version	25/06/2018	Athanasios Ballis (UoA)
1.2	Comments from internal review	28/06/2018	George Kakaletris (UoA) Dimitris Katris (UoA)
1.3	Pre-final draft	29/06/2018	George Kakaletris (UoA) Athanasios Ballis (UoA)
2.0	Updated version	03/07/2018	George Kakaletris (UoA) Athanasios Ballis (UoA)

PARTICIPANTS		CONTACT
Agro-Know IKE (Agroknow, Greece)	Agroknow	Nikos Manouselis Email: nikosm@agroknow.com
Stichting Wageningen Research (ALTERRA, The Netherlands)	ALTERRA WAGENINGEN UR	Rob Lokers Email: rob.lokers@wur.nl
Institut National de la Recherché Agronomique (INRA, France)	INRA SCIENCE & IMPACT	Pascal Neveu Email: pascal.neveu@inra.fr
Bundesinstitut für Risikobewertung (BfR, Germany)	BfR Bundesinstitut für Risikobewertung	Matthias Filter Email: matthias.filter@bfr.bund.de
Consiglio Nazionale Delle Richerche (CNR, Italy)		Leonardo Candela Email: leonardo.candela@isti.cnr.it
University of Athens (UoA, Greece)	National and Kapodistrian University of Athens	George Kakaletris Email:gkakas@di.uoa.gr
Stichting EGI (EGI.eu, The Netherlands)	egi	Tiziana Ferrari Email: tiziana.ferrari@egi.eu
Pensoft Publishers Ltd (PENSOFT, Bulgaria)	PENSOFT	Lyubomir Penev Email: penev@pensoft.net

ACRONYMS LIST

API	Application Programming Interface
AuthN	Authentication
AuthZ	Authorization
CPU	Central Processing Unit
CSS	Cascade Style Sheets
CSV	Comma Separated Value
CSW	Catalogue Service
D	Deliverable
DoA	Description of Action
DOM	Document Object Model
GIS	Geographic Information System
HDFS	Hadoop Distributed File System
HTML	HyperTextMarkup Language
HTTP	HyperText Transfer Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
M	Month
NetCDF	Network Common Data Form
OGC	Open Geospatial Consortium
RDD	Resilient Distributed Dataset
REST	REpresentational State Transfer
SDI	Spatial Data Infrastructure
SQL	Structured Query Language
SVG	Scalable Vector Graphics
S/W	Software
UI	User Interface
URI	Uniform Resource Identifier
VO	Virtual Organization
VRE	Virtual Research Environment
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WP	Work Package
WPS	Web Processing Service
XML	Extensible Markup Language

EXECUTIVE SUMMARY

The three use cases (UCs) from the respective project communities pose the basis for further analysis of the overall project requirements. The identified functional requirements include visualization types like geospatial maps, charts, network charts and tables, and interactions that may emerge among several of the aforementioned types. On the other hand, the identified non-functional requirements include a number of generic suggestive requirements which may improve the quality of the offered system, such as cross browser support, performance, responsiveness and data security and correctness. Another identified type of requirement relates to interoperability of processes and standards. Indicatively, OGC WMS provides a simple HTTP interface for requesting geo-registered map images from geospatial databases, REST API calls are expected to be used for the communication of different sub-systems and support of a number of file formats may be provided to the user (NetCDF, GeoTIFF, etc).

Based on those and other identified requirements, a follow-up analysis summarizes the expected specifications to be used for the proposed system in a tabular format for efficient handling and future reference.

Consequently, a high-level system architecture is described with what is acknowledged until this point in the project's lifetime. In brief, the proposed architecture is modular and consists of three distinct areas: the **enabling services**, which may include external services like authorization ones, the **backend presentation services**, which are responsible for the business logic AGINFRA+ is planned to provide, and finally the **frontend presentation services**, which in fact provide the user with visualization components through web technologies and widgets.

This is the second release which follows the evolvement of the internal project interrelations, concretely accompanying the requirements of the three identified AGINFRA+ communities. The current main envisioned technologies being used, are based on open standards (HTML5, JavaScript visualization libraries, Geoserver OGC standards), Apache Spark for cluster computing techniques to support demanding distributed visualization services and storage (SQL/NoSQL/Object) alternatives.

The current specification report, being consistent with the agile methodology, is a dynamic document which provides an updated view of presentation technologies at the half of the project and forms the main reference for any architectural system design pattern to be applied on the implementation of project’s related technology.

TABLE OF CONTENTS

LIST O FTABLES

LIST OF FIGURES

1 INTRODUCTION

The current document presents the architectural design of the AGINFRA+ Visualization and Publishing components. It summarizes the visualization requirements of researchers in the agri-food sector and defines specifications, standards and supported processes of the technology to be delivered. It follows the evolvement of the internal project interrelations, concretely accompanying the requirements of the three identified AGINFRA+ communities. Being consistent with the agile methodology, it is a dynamic document which provides an updated view of presentation technologies through existing e-infrastructures, third party applications and means interfacing with these infrastructures, as emerged from the analysis of the use cases concerning visualization and publishing.

1.1 STRUCTURE OF THE DOCUMENT

This document is structured as follows:

- The first chapter presents the context of the document and briefly describes its content.
- In the second chapter the overall requirements emerged from the AGINFRA+ communities are translated to technical visualization requirements, properly categorized.
- In the third chapter the main specifications are displayed.
- In the fourth chapter a high level architectural proposal is provided.
- Finally, before the conclusion, the fifth chapter indicates the main technologies identified.

1.2 INTERCONNECTION TO OTHER WPS/TASKS

This document is based on the outcomes of three deliverables (namely D5.1, D6.1 and D7.1) emerging from the analysis of use cases of the three AGINFRA+ communities related to Agro-climatic and Economic Modelling, Food Safety Risk Assessment and Food Security .In those, three different use case categories are analysed in bottom-up method. In technical terms, this document poses the physical sequence of the neighbouring technical specification deliverables D2.1 and D3.1, which deal with (a)Data and semantics layer and (b) Data analytics and processing layer respectively. It is expected that the current deliverable will pose the reference basis for visualization aspects when updating each aforementioned deliverable to its next version. This final version of the technical specifications fine tunes the architectural schema being applied to all three communities using agile methodology.

2 REQUIREMENTS ANALYSIS

The three AGINFRA+ communities have different needs and requirements, which are expressed through documented use cases. Here, those needs and requirements are translated in visualization technical terms and grouped under different categories. This version concludes the requirement analysis, as it has emerged from the several identified use cases.

2.1 FUNCTIONAL REQUIREMENTS

In general, the functional requirements define a function of a system or its components, where a function is described as a specification of behaviour between outputs and inputs. Within the project, the functional requirements explore the necessary operational needs for the overall visualization schema to be effectively applied. Each documented functional requirement is accompanied with a reference to the UC it emerged from, so as to better track and explore the triggering cause for its rationale.

2.1.1 Visualization Types

In general, the main documented UC visualization types involve the usage of geospatial maps, charts, network charts and tables.

2.1.1.1 Geospatial Maps

A user/researcher is in place to explore a geospatial map zooming in and out, selecting a layer and receiving information when hovering on it. Several layers may apply to the map in an overlay style, comparing over time or providing zonal statistics visualization. There are two types of maps based on the image files used, namely vector and raster. Those two generic types of image files are used from many scientific sectors, including agriculture, while the above formats depend on the data used.

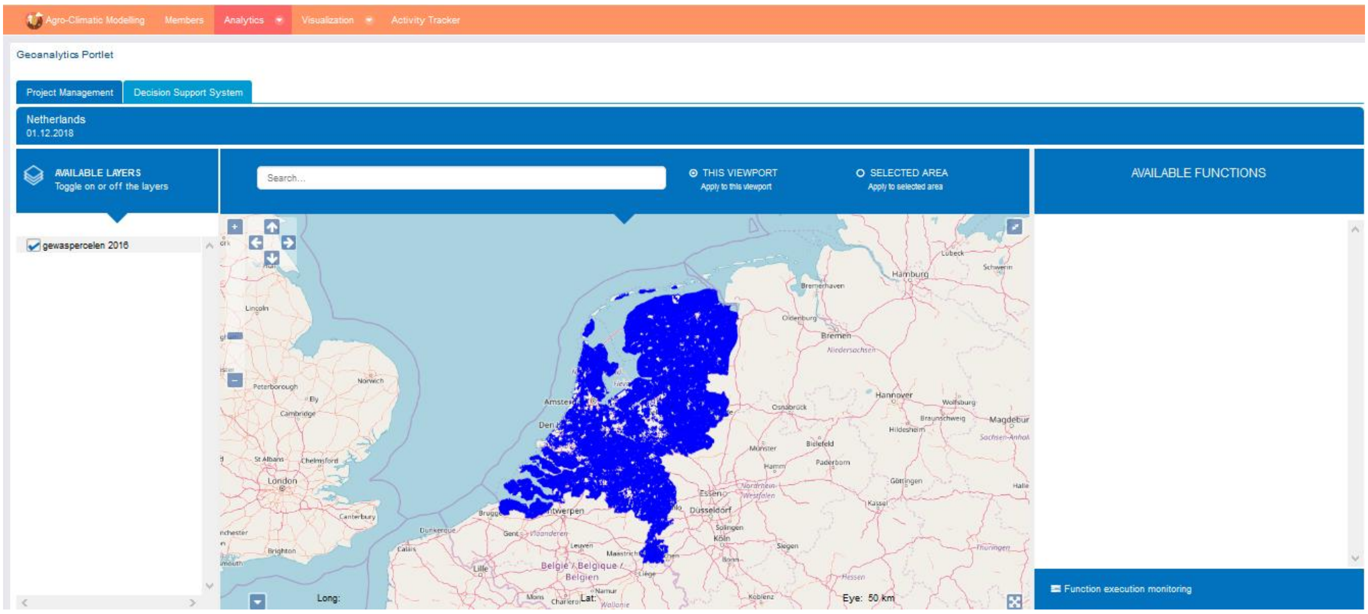

Figure 1: Exemplar geospatial map with raster data representation

The vector image type draws the outline of shapes based on mathematical vectors dealing with shapefiles, format which is popular for geographic information system (GIS) software. The shapefile format can spatially describe vector features like points, lines, and polygons.
The raster image type or bitmap image is array data structure, representing a generally rectangular grid of pixels or points of colour that can be rendered on a display medium. A raster is technically characterized by the width and height of the image in pixels and by the number of bits per pixel (or

colour depth, which determines the number of colours it can represent[1]). This visualization type is expected to be used by all the identified UCs of WP5 and the first two of WP7.

2.1.1.2 Charts

The conventional charts may include line, bar, pie or other types of plots. They are the most popular forms to graphically display data of a wide range of forms and types, and shall be included in the project's offered output formats. "x-y" plots are based on the Cartesian Coordinates system, where data is displayed relatively to two axes (horizontal, vertical), each representing different quantitative categories sourced from several datasets. Bar charts[2] present grouped data with rectangular bars with lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally to show comparisons among categories. One axis of the chart shows the specific categories being compared, and the other axis represents a discrete value. Some bar graphs present bars clustered in groups of more than one. A pie chart[3] is a circular statistical graphic which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice (and consequently its central angle and area), is proportional to the quantity it represents. Stack charts can be set to stack on top of each other either in order (normal display) or via percentage (filling the plot area and correspondingly drawing each point of the same category).

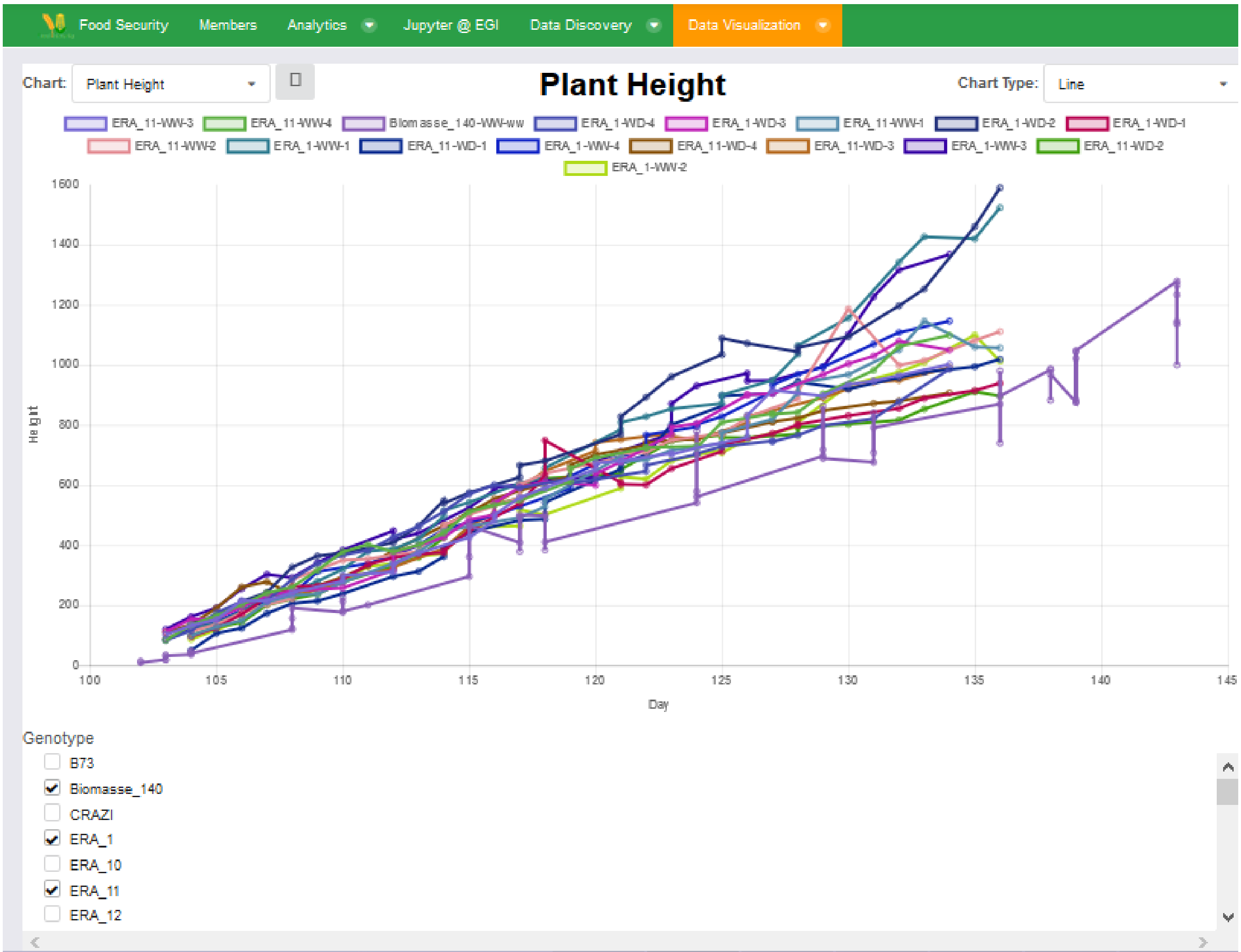

Figure 2: Exemplar charts visualization

[1]https://en.wikipedia.org/wiki/Raster_graphics
[2]https://en.wikipedia.org/wiki/Bar_chart
[3]https://en.wikipedia.org/wiki/Pie_chart

The third UC of WP5 and the UCs of WP6 and WP7 are assumed to use charts for visualizing data.

Time-series

Agricultural data is accustomed to be displayed in time-series format. Thus, the project shall support any point-in-time representation of various data within the envisaged schema, visualizing datasets under specific perspectives.

Uncertainty / Probability

Several agricultural models promote the display of graphs that present values with error margins or estimate intermediate sampling values, thus they shall be also included in the offered visualization options.

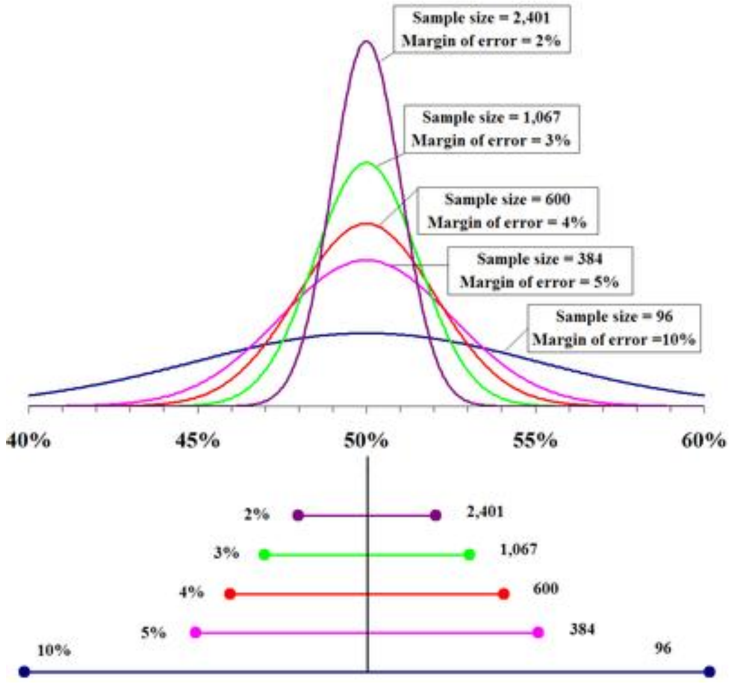

Figure 3: Exemplar error margin chart (source: wikipedia)

2.1.1.3 Network Charts

The network charts refer to the visualization of workflows and processes nodes. Nearly all the documented UCs are based on existing S/W models which export workflows with algorithmic interrelations.

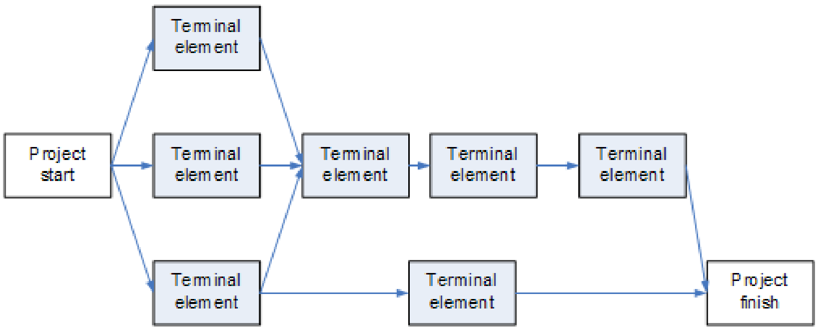

Figure 4: Exemplar network graph (published under GNU free documentation license v1.2)

2.1.1.4 Tables

The tabular representation of data poses another basic visualization format. A user may be in place to sort and/or filter data displayed.

2.1.1.5 Mind maps

Diagrams used to visually display information in hierarchical format clearly denoting the existing relationships among terms. It is expected to serve the needs of WP6 UCs.

2.1.2 Interaction Requirements

Given the discussions with all three communities, this section states the interaction which may occur among the usage of the aforementioned visualization types based on each domain experience. The following parts are indicative and pose a first approach based on what has been declared by each UC on their respective deliverables.

Geospatial maps

When a user processes a geospatial map, they may be in place to proceed to further visualization schemas in forthcoming sequential levels. For example, when selecting an area on a map, a new chart may be triggered displaying specific information on the selected region, a new table may display explicit data on the referenced region or a combination of visualization occurs.

Charts

The representation of a graph may accordingly lead the user to either a transformation of an existing chart to another chart format based on the same reference data, or provide the user with the ability to amend the reference data and extract new charts. The aforementioned amendment may also include the display of other visualization types, like the inclusion of a table or a network chart.

Network Charts

The envisioned display of a network chart may provide the user with the alternative to choose a specific node, leading to a new visualization type (chart or table).

Tables

The expected interaction is focused on the fact that a user may amend the tabular data displayed, consequently leading to the extraction of different charts.

2.1.3 Publishing Requirements

On the scope of AGINFRA+ project a number of publishing technologies have been identified and developed under the prism of the creation of a relative scientific journal. Their scope is to promote open science, serving among others a wide range of “open-related terminology” (access, data, open-source s/w, peer-review, science policies, funding, science evaluation, science tools, education) and aiming at providing reusable, downloadable and easy to analyze data. A few key elements are hereby listed:

- Machine-readable content, which practically supports pdf file analysis and post-publication markup and data extraction.
- Authoring tool ability, which provides a fully web- and XML- based life cycle of a manuscript.

- Publication of all outputs of the research cycle, starting from the research idea and proposal up to the open and public peer review process.

2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe qualitative and quantitative characteristics of the system, rather than specific behaviour or attribute that directly link to the functionality delivered to its intended users. The following list contains the most representative non-functional requirements that are assumed to play important role in the overall visualization schema, improving the overall quality of the service provided.

- **Cross browser support**. The envisaged visualization schema shall be independent and uniformly run in whatever browser is used. In specific, it is expected to function properly in the last updated version of five of the top used browsers, namely Google Chrome, Mozilla Firefox, Safari, Microsoft Internet Explorer, Microsoft Edge.
- **Responsiveness**. The ultimate visualizations provided shall be responsive in the sense that they maintain all critical information independent of the size of the screen or web browser one is viewing. Thus, in principle this should work in desktop, smartphone, ipad and any other device type used.
- **Security**. Data should be secured in the sense of privacy and confidentiality, where and if such policy exists.
- **Correctness**. Information provided should be based on correct data, while error-free computations should be applied.
- **Performance**. Computations used should not substantially add more effort and resources either in the background processing or during data accessing.
- **Visual feedback provision**. It is preferable for users to be provided with visual feedback, so that they are aware of the operation being or to be performed.
- **Open licensing support**. Open licensing should be supported to become compliant with the project's scenarios.
- **Embedding**. Support of embedding visualizations in other (web) applications is preferable.

2.3 INTEROPERABILITY REQUIREMENTS

A number of existing standards and processes has already been identified and may be utilized, adding value and functionality to the proposed architecture.

- **WMS** – Web Map Service[4] provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. It poses one of the quite many protocols belonging to OGC[5].
- File format specifications:
    - **Shape file** spec – The shapefile format[6] is a digital vector storage format for storing geometric location and associated attribute information. This format lacks the capacity to store topological information. It is possible to read and write geographical datasets using the shapefile format with a wide variety of software. The shapefile format is simple because it can store the primitive geometric data types of points, lines, and polygons. Shapes (points/lines/polygons) together with data attributes can create infinitely many representations about geographic data. Representation provides the ability for powerful and accurate computations.

[4] https://en.wikipedia.org/wiki/Web_Map_Service
[5] http://www.opengeospatial.org/
[6] https://en.wikipedia.org/wiki/Shapefile

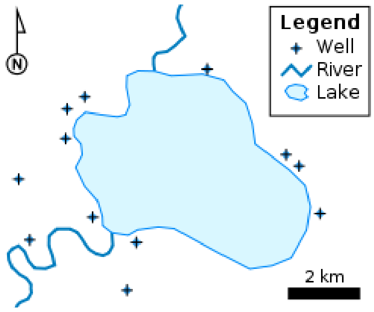

Figure5: Exemplar shapefile representation[7]

- **NetCDF** spec – NetCDF[8] is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. NetCDF was developed and is maintained at Unidata. Unidata provides data and software tools for use in geoscience education and research.
- **GeoTIFF** spec – GeoTIFF[9] is a public domain metadata standard which allows georeferencing information to be embedded within a TIFF file. The potential additional information includes map projection, coordinate systems, ellipsoids, datums, and everything else necessary to establish the exact spatial reference for the file.

- **REST API** – Representational state transfer (REST) or RESTful web services[10] is a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. In a RESTful Web service, requests made to a resource's URI will elicit a response that may be in XML, HTML, JSON or some other defined format. The response may confirm that some alteration has been made to the stored resource, and it may provide hypertext links to other related resources or collections of resources. Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP verbs GET, POST, PUT, DELETE and so on. By making use of a stateless protocol and standard operations, REST systems aim for fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running.

[7] source wikipedia [https://en.wikipedia.org/wiki/File:Simple_vector_map.svg]
[8] http://www.unidata.ucar.edu/software/netcdf/docs/
[9] https://en.wikipedia.org/wiki/GeoTIFF
[10] https://en.wikipedia.org/wiki/Representational_state_transfer

3 SYSTEM SPECIFICATIONS

The system specifications are based on the requirements analysis and are briefly stated in tabular format per category, receiving a unique identifier (title) so as to be encoded and efficiently referenced in forthcoming deliverables or future activities, keeping track on their status. The specification origin may source explicitly from the DoA or the UCs, or may be assumed to emerge from other needs. Regarding priority, 'high'-marked is considered a must for the implementation of the envisioned system, 'normal'-marked is a concrete suggestion while 'nice-to-have'-marked is an optional suggestion.

3.1 FUNCTIONAL SPECIFICATIONS

The following table summarizes the main functional system specifications that are expected to apply within the visualization schema.

nr	title	description	origin	priority
1	F_001	User zooms in and out a specific region in a georeferenced map	DoA	High
2	F_002	Provide georeferenced overlay information in a map	DoA	Normal
3	F_003	Extract statistical or other type of information when hovering on a specific region in a map	Other	Normal
4	F_004	Use specific visualization file format (raster and vector)	UCs	High
5	F_005	Simulate GIS functionality	UCs	Nice to have
6	F_006	User may draw specific chart type	UCs	High
7	F_007	User may change dataset values	UCs	Normal
8	F_008	User may change chart type displayed	UCs	Normal
9	F_009	User may draw probability graphs	Other	Nice to have
10	F_010	User may draw network chart	Other	Normal
11	F_011	User may sort and/or filter tabular data	Other	Normal
12	F_012	User may draw a new chart (e.g. bar) when interactively dealing with geospatial map data	Other	Normal
13	F_013	User may choose specific network node from network chart and draw new chart	Other	Nice to have
14	F_014	User may amend tabular data and extract new charts	Other	Normal

Table 1: Functional specifications summary

3.2 NON-FUNCTIONALSPECIFICATIONS

The following table summarizes the main non-functional system specifications that are expected to apply within the visualization schema.

nr	title	description	origin	priority
1	NF_001	Visualizations function properly and nearly in the same way within the 5 top-used browsers	Other	Nice to have
2	NF_002	User experience is unaffected regardless the size of the screen and the output means used (desktop, smartphone, etc), providing actual responsive behaviour	Other	Nice to have
3	NF_003	Privacy and confidentiality rules are applied concerning	Other	Nice to have

		security of data		
4	NF_004	Data used should be correct and error-free throughout their overall process	Other	Nice to have
5	NF_005	Performance in process computations shall not add more effort and resources usage	Other	Nice to have
6	NF_006	Users shall be visually guided when an activity is being performed, receiving clear corresponding messages and/or explanatory icons	Other	Nice to have
7	NF_007	Open licensing should be supported in several of the protocols used	Other	Nice to have
8	NF_008	Users may embed the extracted visualizations in other compliant web applications	Other	Nice to have
9	NF_009	Users may save a visualization schema in different file formats	Other	Nice to have
10	NF_010	Users may provide preferred minimum and maximum zoom levels when dealing with geospatial visualizations	Other	Nice to have

Table 2: Non-functional specifications summary

3.3 INTEROPERABILITY SPECIFICATIONS

The following table summarizes the main interoperability system specifications that are expected to apply within the visualization schema.

nr	title	description	origin	priority
1	IN_001	WMS compatibility spec, interconnecting georeferenced images to geospatial databases	Other	Normal
2	IN_002	Users shall conduct REST API calls in order for several components to be able to interoperate via internet	Other	Normal
3	IN_003	Users shall be able to use files (shapefiles, NetCDF, GeoTIFF) to exploit and interoperate with several georeferenced components	Other	Normal
4	IN_004	Users shall be able to exploit well-known web (w3c) or georeferenced (OGC) standards	Other	Normal

Table 3: Interoperability specifications summary

4 HIGH-LEVEL SYSTEM DESIGN

The following diagram depicts the high-level architecture of the proposed Presentation system.

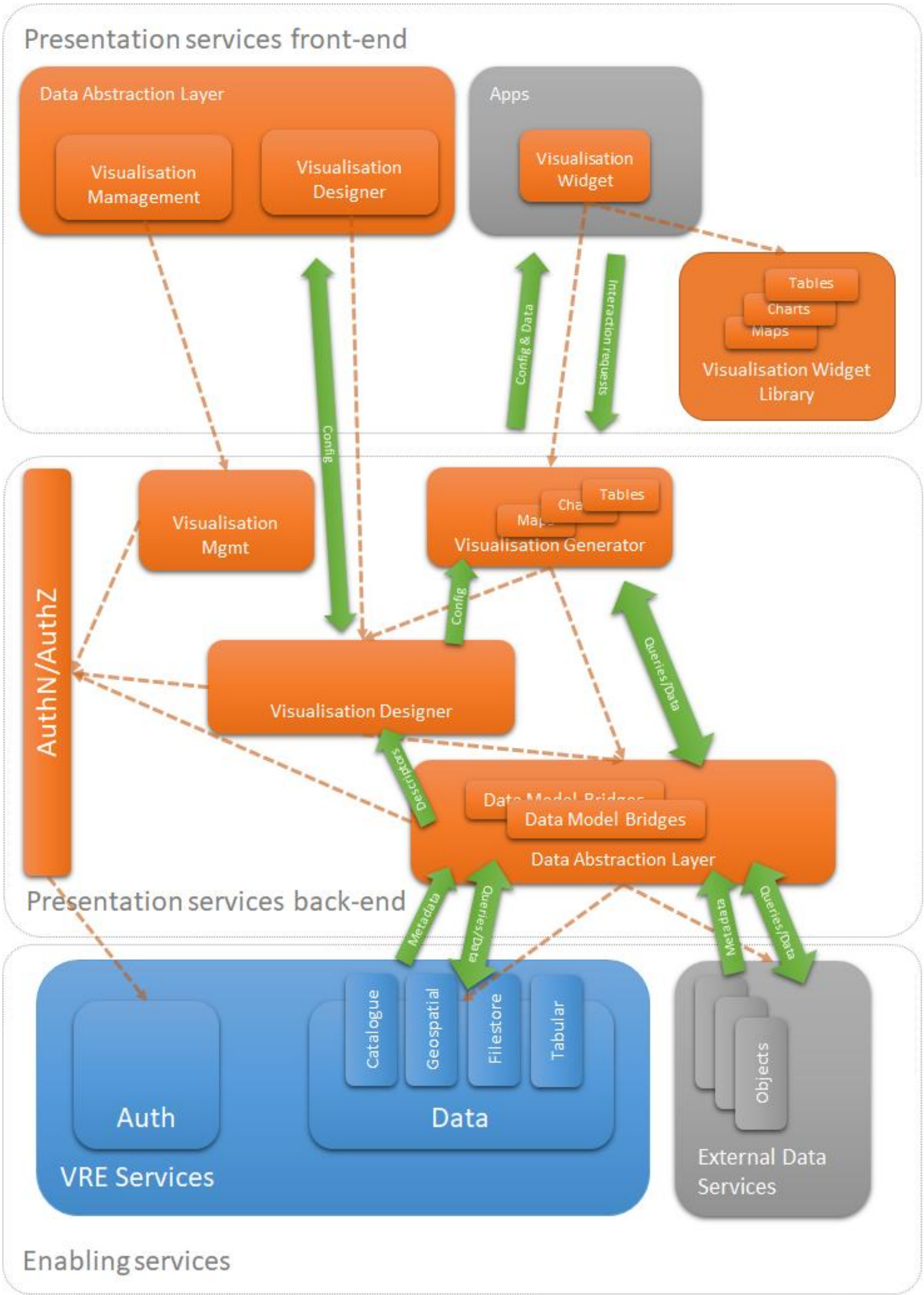

Figure 6: Presentation system architecture diagram

The architecture distinguishes three areas:

- The area of the Enabling Services, i.e. services that exist outside the Presentation system and allow it to build on top of them in order to offer its services to the user. Services that fall in this area are, not limited to, the following:
  - Authorization services
  - Data management services
  - Resource registries
  - Data catalogues

  In the context of AGINFRA+ it is expected that all those services will be covered mainly by the Infrastructure VRE services.
- The area of Presentation backend services that handle all the business logic of the presentation layer and are the main focus of the work performed in the context of the project. Those services will be briefly presented in the following paragraphs.
- The area of Presentation front-end services that deliver tools and visualisation components to their audience via web technologies. Those fall into two categories:
  - Management and design tools that handle the respective backed services,
  - Presentation widgets, that allows embedding of visualisation features in 3rd party applications.

More details on those services will be presented in the following paragraphs. In the architecture diagram presented previously, the following colouring conventions are followed:

- Grey blocks depict services and applications that are not part of the AGINFRA+ core infrastructure.
- Blue blocks depict components of the VRE that are not part of the Presentation Services.
- Orange blocks depict Presentation Services components.
- Orange dotted arrows depict (API) dependencies.
- Green arrows depict major flows of information/data in the system. *Note: Only the most important flows for comprehending the diagram are presented.*

In the following paragraphs we present in brief the roles and key specs of each component/service of the Presentation layer.

The **AuthN/Auth Z**component is the subsystem that allows on one hand all services of the platform to communicate with standards' compliant authentication systems in order to identify the user that invokes their services, while on the other hand offers a standard set of tooling for handling authorization on service specific terms. Almost all services of the system depend on the AuthN/AuthZ mechanism. However the direct dependency of the visualization service itself is optional, as other layers already ensure proper access to system resources is needed for generating graphs etc. However, if specific acceleration techniques are to be embedded in the system, the **Visualization Designer** component is a subsystem that allows its authorized user to configure a visualization scenario, by:

- Selecting the data to use,
- Selecting the queries to be performed on the data,
- Configuring the presentation details and the behaviour of the user interaction Visualization.

The set of all this information is packaged as a configuration resource to be utilized by the Visualization Generator. The Visualization Designer is accompanied by a web UI that may offer tooling for the user to design visualizations in various forms. However, it is assumed that graphical designers will be out of scope for AGINFRA+ implementation and only expert tools (i.e. developer oriented) will be offered.

The **Visualization Management** component, accompanied by a UI, offers the persistency and retrieval tools required to handle visualization configurations (resources) designed via the designer. It is the starting point of management for the entire platform.

The **Visualization Generator** is the key element in presenting data to end users via various means. It is the intermediary between the end user application and the entire system. Although it does not render the data at the end user application, it provides all the information to renderers (Visualization Widgets) to not only present data, but also, where required, to appropriately handle the interaction with the user. The visualization generator will be providing enough data/information to:

- Render UI options for configuring the visualization at client level, depending on the settings placed by the visualization designer.
- Render data at client side, either by directly providing the data or by providing the endpoint where data (raw or rendered) is to be retrieved from.
- Allow the UI to navigate the user in the data or out of those (drill-in or link to other views or information panels etc).

The rendering of data for the end user is performed by components collectively called **Visualization Widgets**. A sufficient set of those is offered in a library assembled by the project, based on reusable elements, enhanced in order to address the needs and requirements of AGINFRA+ Presentation Service and also exploit its advanced offerings. The baseline widgets to be offered will be pure Javascript components. However the option to develop widgets for other platforms is open for 3rd party groups.

The **Data Abstraction Layer** is where the Visualization Generator is based to retrieve data for its operations and offers adapters for a number of standard datasources found in the infrastructure. The data in focus will be Geospatial and tabular (or relational ones), the latter being placed in no-sql or sql stores. By definition, all services of the presentation layer offer REST APIs.

5 TECHNOLOGIES

The visualization technologies proposed are based on Open Standards. The following list indicatively displays the most important tentative ones, which are followed by brief description:

- **JavaScript** – is a high-level, dynamic, untyped, object-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. A number of visualization JavaScript libraries are hereby enlisted.
    - **D3.js** – D3[11] is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3’s emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.
    - **Chart.js** – Chart.js[12] is a JavaScript library for creating animated, interactive graphs for inclusion on web pages. It uses the HTML5 canvas element and is responsive, it supports Line, Bar, Radar, Polar Area, Pie and Doughnut charts and includes options to extend these chart types and write new ones.
    - **Plotly.js** – Plotly[13] is an online data analytics and visualization tool, which may handle error bars. It provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino and REST.
- **GeoServer**– It is an OGC compliant implementation[14] consisting of a number of open standards.
- **GeoNetwork** – It is a catalogue application to manage spatially referenced resources. It provides powerful metadata editing and search functions as well as an interactive web map viewer. It is currently used in numerous Spatial Data Infrastructure initiatives across the world. GeoNetwork[15] provides an easy to use web interface to search geospatial data across multiple catalogues. The search provides full-text search as well as faceted search on keywords, resource types, organizations, scale. Users can easily refine the search and quickly gets to the records of interests.
- **OGC** Specifications – The Open Geospatial Consortium is an international not for profit organization committed to making quality open standards for the global geospatial community. These standards are made through a consensus process and are freely available for anyone to use and improve sharing of the world's geospatial data. A number of protocols is expected to be used by the project:
    - **CSW** – Catalogue Services[16] support the ability to publish and search collections of descriptive information (metadata) for data, services, and related information objects. Metadata in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software. Catalogue services are required to support the discovery and binding to registered information resources within an information community. OGC Catalogue interface standards specify the interfaces, bindings, and a framework for defining application profiles required to publish and access digital catalogues of metadata for geospatial data, services, and related resource information. Metadata act as generalised properties that can be queried

[11]https://d3js.org/
[12]http://www.chartjs.org/
[13]https://plot.ly/
[14]http://geoserver.org/
[15]http://geonetwork-opensource.org/
[16]http://www.opengeospatial.org/standards/cat

and returned through catalogue services for resource evaluation and, in many cases, invocation or retrieval of the referenced resource.
    - **WPS** – Web Processing Service[17] provides rules for standardizing how inputs and outputs (requests and responses) for geospatial processing services function, such as polygon overlay.
    - **KML** – [optional] Keyhole Markup Language[18] is an XML language focused on geographic visualization, including annotation of maps and images. Geographic visualization includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look.
    - **WCS** – [optional] A Web Coverage Service[19] offers multi-dimensional coverage data for access over the Internet.
    - **WFS** – [optional] Web Feature Service provides an interface allowing requests for geographical features across the web using platform-independent calls.
- **ISO Standards** – It is an independent, non-governmental international organization[20] with a membership of 163 national standards bodies. Through its members, it brings together experts to share knowledge and develop voluntary, consensus-based, market relevant International Standards that support innovation and provide solutions to global challenges.
    - **ISO 19115-1:2014** defines the schema required for describing geographic information and services by means of metadata[21]. It provides information about the identification, the extent, the quality, the spatial and temporal aspects, the content, the spatial reference, the portrayal, distribution, and other properties of digital geographic data and services.
- **W3C**related technologies – World Wide Web Consortium. It is an international community that develops open standards to ensure the long-term growth of the Web. W3C standards define an Open Web Platform for application development that has the unprecedented potential to enable developers to build rich interactive experiences, powered by vast data stores, which are available on any device.
    - **HTML5** is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard, offering visualization advances.
    - **CSS** is a language that describes the style of an HTML document, and how HTML elements should be displayed providing visualization variety.
    - **SVG** – Scalable Vector Graphics define vector-based graphics in XML format.
    - **XML** – eXtensible Markup Language is designed to store and transport data, being both human- and machine-readable.

Other technologies to be considered:

- **Google Charts** – Google Charts[22] provide a perfect way to visualize data on your website. From simple line charts to complex hierarchical tree maps, the chart gallery provides a large number of ready-to-use chart types.
- **SDI** – Spatial Data Infrastructure[23] is a data infrastructure implementing a framework of geographic data, metadata, users and tools that are interactively connected in order to use spatial data in an efficient and flexible way.

[17]http://www.opengeospatial.org/standards/wps
[18]https://developers.google.com/kml/documentation/
[19]http://www.opengeospatial.org/standards/wcs
[20]https://www.iso.org/home.html
[21]https://www.iso.org/standard/53798.html
[22]https://developers.google.com/chart/
[23]https://en.wikipedia.org/wiki/Spatial_data_infrastructure

- **D4Science VREs** – The D4science infrastructure[24] hosts the set of resources (computational, storage, data) which can be shared among users of a Virtual Organization (VO). A VRE aggregates and deploys on-demand content resources and application services by exploiting computational and storage resources of grid and cloud infrastructures belonging to a VO.
- **NASA WWW** – NASA World Wind[25] is a free, open source API for a virtual globe. World Wind allows developers to quickly and easily create interactive visualizations of 3D globe, map and geographical information. Organizations across the world use World Wind to monitor weather patterns, visualize cities and terrain, track the movement of planes, vehicles and ships, analyze geospatial data, and educate people about the Earth.
- **Apache Spark** – It provides programmers with an application programming interface centered on a data structure called the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines that is maintained in a fault-tolerant way. Apache Spark[26] requires a cluster manager and a distributed storage system. For cluster management, Spark supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos. For distributed storage, Spark can interface with a wide variety, including Hadoop Distributed File System (HDFS), MapR File System (MapR-FS), Cassandra, OpenStack Swift, Amazon S3, Kudu, or a custom solution can be implemented. Spark also supports a pseudo-distributed local mode, usually used only for development or testing purposes, where distributed storage is not required and the local file system can be used instead; in such a scenario, Spark is run on a single machine with one executor per CPU core to support demanding distributed visualization services.
- **SQL/NoSQL Datastores** – A data store is a repository for persistently storing and managing collections of data which include not just repositories like databases, but also simpler store types such as simple files. A database is a series of bytes that is managed by a database management system, while a file is a series of bytes that is managed by a file system. Thus, any database or file is a series of bytes that, once stored, is called a data store.
- **Object storage** – Object storage[27] is a computer data storage architecture that manages data as objects, as opposed to other storage architectures like file systems which manage data as a file hierarchy and block storage which manages data as blocks within sectors and tracks. Each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier. Object storage can be implemented at multiple levels, including the device level (object storage device), the system level, and the interface level. In each case, object storage seeks to enable capabilities not addressed by other storage architectures, like interfaces that can be directly programmable by the application, a namespace that can span multiple instances of physical hardware, and data management functions like data replication and data distribution at object-level granularity (e.g getting CSV/xlsx files).

[24] https://www.d4science.org/
[25] https://worldwind.arc.nasa.gov/
[26] https://en.wikipedia.org/wiki/Apache_Spark
[27] https://en.wikipedia.org/wiki/Object_storage

6 Conclusions

This report is rationally based on the advances of the UCs explicitly analyzed by the three project communities. The extraction of the requirements and specifications is conducted under the scope of the visualization prism, where open standards, processes and technologies have been chosen to eventually propose the architectural working schema. In practice, the system poses the basis for covering all functional specifications, aims to maximize reuse in an attempt to minimize resource usage and maximize outcome and adopts well known standards promoting interoperability.

The current version is the final release of D4.1 which has been constantly updated taking into consideration the maturity evolvement of other concurrent correlated work by other project partners (mostly the three communities) under the agile methodology.