



## Raising Open and User-friendly Transparency- Enabling Technologies for Public Administrations



Project number 645860  
H2020-INSO-2014

### D4.1 Alpha version Of SPOD

(Final, version 1.0, 1/31/2016)



WISE & MUNRO



**Document produced by**

Organisation: Università di Salerno

Author / email: Vittorio Scarano, vitsca@dia.unisa.it

Main contributions by: Delfina Malandrino, Gennaro Cordasco, Ilaria Manno, Giuseppina Palmieri, Carmine Spagnuolo, Luca Vicidomini, Donato Pirozzi, Renato De Donato, Luigi Serra, Andrea Petta

Subject: Alpha version of SPOD

Due date: 31 January 2016

**Dissemination level:** Public (PU)

**Reviewed and approved by**

Date	Name	Organization
21-01-2016	Albert Meijer	Utrecht University (UU)
22-01-2016	Adegboyega Ojo	National University of Ireland, Galway (NUIG)

**Revision History**

Version	Date	Authors	Status	Description of Changes
0.1	02-12-2015	G.Palmieri	Draft	A first organization
0.2	07-01-2016	V.Scarano	Draft	Reorganization
0.3	08-01-2016	G.Palmieri	Draft	Architecture section added, Subdivision of the work
0.4	14-01-2016	V.Scarano, G.Palmieri, D.Pirozzi, A.Petta, L.Serra, C.Spagnuolo, I.Manno, D.Malandrino R.De Donato	Draft	Added all the sections independently developed: state of the art, architectural design, system design, testing, privacy (missing the non functional reqs, the spod design, the bibliography)
0.5	14-01-2016	V.Scarano, G.Palmieri, R.De Donato, L. Vicidomini, D.Pirozzi	Draft	Non-functional reqs, bibliography, SPOD design, change in order of sections of Requirements and Design Principles, SPOD-TET interactions
0.6	15-01-2016	V.Scarano, G.Palmieri	Alpha	Cleaned all the comments from the 0.5

0.7	25-01-2016	G.Palmieri	Draft	Unisa review, Addressed all reviewers comments from UU and NUIG. Included part from CNRS.
0.8	30-01-2016	V.Scarano G.Palmieri	Beta	Final check of the document (cosmetics, references, pages, etc.).
1.0	31-01-2016	V.Scarano G.Palmieri	Final	Release

## TABLE OF CONTENTS

1	Introduction .....	9
1.1	Roadmap .....	10
2	Requirements .....	11
2.1	Functional requirements.....	11
2.1.1	Functional requirements from WPs .....	11
2.1.2	User Interface Design for the Discussion Room (Public Rooms) .....	22
2.2	Non-functional requirements .....	25
2.2.1	Non-functional Requirements from the Quality in Software .....	25
2.2.2	Domain-specific non-functional requirements .....	27
2.2.3	Non-functional Requirements from State-of-the-Art Deliverable D2.1 .....	28
3	Design Principles and Technologies .....	28
3.1	Component-based Development: “Reuse rather than Reinvent” .....	28
3.2	Tackling the Heterogeneity of the context .....	29
3.3	Building Blocks and Software Technologies .....	30
3.3.1	The Social Platform .....	30
3.3.2	Libraries for Data Visualization and Charting .....	33
3.3.3	Web Programming Frameworks .....	38
3.4	Conclusions .....	40
4	Architecture design of SPOD.....	41
4.1	The overall architecture .....	41
4.2	SPOD on Oxwall.....	43
4.3	DatalEt-Ecosystem Provider (DEEP).....	43
4.3.1	Architecture .....	44
4.3.2	DEEP Work cycle .....	44
4.3.3	DEEPCIENT: an example of use.....	45
4.3.4	Web Components in DEEP .....	46
4.3.5	Datalets in SPOD .....	48
4.3.6	Interoperability list with Dataset Providers .....	49
4.4	The Authentication Server for Joint Authentication .....	50
4.4.1	Concepts .....	50
4.4.2	The Architecture .....	51
4.4.3	Implementation details.....	53
4.5	Design Language .....	53
4.5.1	Material design .....	53
4.5.2	Screen Resolution and Browser .....	54
5	System Design of SPOD .....	55
5.1	Oxwall and plugins .....	55
5.1.1	Open Data Enabler (ODE) .....	56
5.1.2	Open Wall .....	61
5.1.3	Agora.....	61
5.1.4	My Space .....	64
5.2	DEEP Controllets and Datalets .....	65
5.2.1	Controllets .....	65

5.2.2	Datalets.....	71
5.2.3	DEEP: additional services.....	80
5.3	Interactions between SPOD and TET.....	81
5.3.1	Information flow from SPOD to TET and vice versa.....	82
5.3.2	Actions forwarded from SPOD to TET and Vice versa.....	83
5.4	A recap of features and requirements.....	85
6	Development, Deployment and testing.....	90
6.1	Agile methodology.....	90
6.1.1	Agile methodology in SPOD.....	90
6.1.2	Incremental delivery (Feedback).....	93
6.2	Automated Testing.....	94
6.2.1	Test cases & Test Scripts.....	95
6.2.2	Automated Testing Architecture.....	96
6.3	Deployment of the architecture.....	97
6.4	Usability: criteria and activities.....	97
7	Cross-Cutting Design Concerns.....	98
7.1	Privacy.....	98
7.2	RRI-ICT.....	100
8	Relevant publications.....	102

## List of Figures

Figure 1: A general overview of ROUTE-TO-PA project: SPOD on the left side and TET on the right side. ....	29
Figure 2: Elgg MVC Architecture. ....	31
Figure 3 - Oxwall MVC Architecture.....	32
Figure 4: Overall architecture .....	42
Figure 5: DEEP and datalets repository architecture: a list of available datalets provided by DEEP and known name of datalet (e.g., Line-chart datalet), DEEP provides its implementation loading from the “datalets” repository. ....	44
Figure 6: DEEP workcycle (DEEP Component should be Web-Component). ....	45
Figure 7: Processing, querying and visualization dataset using a datalet. ....	47
Figure 8: A life-cycle of a Datalet in SPOD .....	49
Figure 9 - ROUTE-TO-PA Platform Authentication Architecture. ....	51
Figure 10. The datalet button for new posts. ....	57
Figure 11. The datalet button for new comments. ....	57
Figure 12. Step 1: Dataset selection. ....	58
Figure 13. Step 2: Fields selection. ....	59
Figure 14. Step 3: Visualization definition. ....	60
Figure 15. When a user creates a new Datalet in a Private room, he can access the list of suggested datasets. ....	61
Figure 17. The Agora, viewing inside a room.....	63
Figure 18. My space.....	64
Figure 19. Creating a text memo in My Space. ....	65
Figure 20. Attaching an element from My space to a new post. ....	65
Figure 21: a Datalet embedded, e.g., an interactive map of Dublin, into WordPress page of the research laboratory of the SALERNO team. ....	81

## List of tables

Table 1: Features between Elgg and Oxwall .....	33
Table 2: all criteria of charting libraries .....	34
Table 3: an overview among chart types .....	37
Table 4: A comparison among chart types .....	37
Table 5: A comparison among chart types .....	38
Table 6: Trends of computer screen resolution formats by StatCounter. ....	55
Table 7: Browser popularity. ....	55
Table 8 - Information Flow from SPOD to TET and vice versa.....	82
Table 9 - User actions forwarded from one system to the other one.....	83
Table 10: Features for each release .....	93
Table 11: Feedbacks received from each partner .....	94

## Acronyms

The following are definitions and acronyms used within the document:

Term	Definition
OD	open data
CC	Creative Common
SPOD	<u>S</u> ocial <u>P</u> latform of <u>O</u> pen <u>D</u> ata
TET	<u>T</u> ransparency- <u>E</u> nhancing <u>T</u> ools
ICT	Information Communication Technology
DEEP	<u>D</u> ata <u>E</u> t- <u>E</u> cosystem <u>P</u> rovider
HTML	HyperText Markup Language
CSS	Cascading Style Sheets,
RRI	Responsible Research and Innovation
GPL	General Public License
PHP	Hypertext Preprocesso
MVC	Model View Controller
LAMP	Linux, Apache, MySQL, PHP/Python/Perl
WAMP	Windows, Apache, MySQL, PHP
IE	Internet Explore
API	Application Program Interface
DOM	Document Object Model
WP	Work Package
RAS	ROUTE-TO-PA Authentication Server
WC	Work Cycle
WC	Web Component is a set of standards designed to provide reusable components for the web.
ODE	<u>O</u> pen <u>D</u> ata <u>E</u> abler
DoW	Description of Work(i.e. Annex 2 of the Grant Agreement)”
WF	Web Framework
WAF	Web Application Framework
ASD	Agile Software Development
DB	DataBase
PA	Public Administration
UC	Use Case

## Executive Summary

This document presents the work in order to carry out Task 4.1 of workpackage 4 (“Technological Development and Integration”): “Social Platform for Open Data (SPOD)”.

The deliverable reports the design, implementation, deployment and testing of the Alpha version (vers. 1.0) of the SPOD. The process followed during the design and implementations is described in the sections, as follows.

- Section 1 introduces the objectives of the project and the organization of the whole document.
- In Section 2, we present the “Requirements”, coming from the research and the activities of the WP, and from the DoW and from the activities conducted in WPs 2 and 3. We present, first, the functional requirements, and, then, the non-functional requirements of our SPOD. They will lead, respectively to the System Design and to the Architecture Design.
- Section 3 describes our “Design Principles and Technologies”, by motivating our technological choices for the SPOD design and implementation. After introducing and motivating two fundamental principles of our design, i.e., “Component-based Development” and “Tackling Heterogeneity of the context”, we explain, briefly, the existing systems and technologies that are needed to leverage upon for the integration. A brief analysis of the state of the art in each field (social networks, libraries, frameworks) and, then, the motivations for the choice for the design of SPOD platform are reported.
- Section 4 addresses the non-functional requirements described in Section 2.2, by describing the “Architecture design of SPOD”, with all its components and the interactions among them. Here the section describes how each building block has been integrated and modified to be connected to a distributed architecture, to provide the functionalities required. While the SPOD plugins are described in Section 5, here we describe how visualizations is provided by DataEt-Ecosystem Provider (briefly DEEP). DEEP is a repository of visualization web-components (Datalets) to use within SPOD and within any other web site or system that needs to visualise data. Then, we present a ROUTE-TO-PA Authentication Server (RAS), which acts as OpenID Authentication Provider and administration tool to manage users’ accounts and the main objective is to allow all users seamless switch between SPOD and TET features in a user-friendly way.
- Section 5 deals with the functional requirements provided in Section 2.1, showing the “System Design of SPOD”, with a brief description of all the features and how they are designed and implemented. A description follows of how Oxwall open-source social network is integrated and customized through the implementation of several plugins (i.e., Open Data Enabler, Open Wall, Agora, My Space and OpenID Connector).
- Section 6 presents the “Development, Design and Testing” of our WP4. The implementation methodology is based on the Agile approach following some guidelines (e.g., requirements received from D2.1, D2.4, D3.1 and D4.1, frequent delivery of usable software (two/three weeks), incremental delivery (feedbacks received from Pilots), testing during development with Pilot of the project receiving feedbacks). To assess the software quality, a software testing has been planned and tested with Selenium framework.
- Section 7 deals with some “Cross-Cutting Design Concerns” addresses issues like Privacy, and Responsible Research and Innovation in ICT (RRI-ICT).
- Section 8 presents a list of publications where some of the activities that led to this deliverable are described.



# 1 INTRODUCTION

---

*“The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.”*

*(Brooks “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987).*

The main aim of the ROUTE-TO-PA project is to improve the impact of ICT-based technology platforms for transparency. As stated in the "Digital Agenda for Europe", in order to provide better public services to citizens and businesses, new opportunities for Public Administrations (PAs) to become more efficient and effective, provide user-friendly services, while reducing costs and administrative burden, are grounded on open government, i.e., increasing information and knowledge exchange, enhanced connectivity, openness and transparency.

In ROUTE-TO-PA we address the issue of transparency, i.e., opening up public data and services and facilitating citizen participation and engagement, with the result of making government processes and decision open. In terms of international practices in transparency, widespread access to the Internet has greatly reduced the cost of collecting, distributing, and accessing government information. But an important effect of the diffusion of networks in the population is that ICT, by promoting good governance, strengthening reform-oriented initiatives, reducing potential for corrupt behaviours, enhancing relationships between government employees and citizens, allowing for citizen tracking of activities, and by monitoring and controlling behaviours of government employees, is able to effectively reduce corruption [1].

Our focus is in contrasting the potential barriers to using ICT for transparency, that we identify in the complexity of the information provided, the lack of tools to facilitate the comprehension and the limited acceptance of ICTs for transparency among citizens.

Concretely, the project is aiming to (a) develop a Social Platform for Open Data (SPOD) enabling social interactions among open data users and between open data users and government data; (b) build Transparency-Enhancing Toolset (TET) as extensions for existing major Open Data Platforms; and (c) develop a set of recommendations (GUIDE) as good practice guide for open data publishers for achieving higher quality transparency through open data.

ROUTE-TO-PA objectives are concrete since we will experiment in practice the innovation proposed by conducting 5 pilot studies in Prato Town Council (Italy), The Hague Town Council and Groningen Town Council (The Netherlands), Dublin Town Council (Ireland) and Issy-les-Moulineaux (France).

In this document, we report about the status of the design, implementation and testing of the first version (Alpha) of the SPOD, that is going to be used in the second year of the project.

The main objective of SPOD is to improve comprehension and increase accessibility of open data and, by favouring social collaboration through a citizen-friendly environment, stimulate and foster comprehension of open data, increase Transparency and proactively generate interest around open data.

Our project adopts a collective intelligence and scenario-based design approach and the initial scenarios developed for each Pilot (during the workshop sessions and interviews on open data and transparency we conducted in the we conducted in the first 6 months of the project) highlighted information, social and

collaborative, and understandability, usability and decision-making needs and requirements that are important for the development of the SPOD, TET and GUIDE. We, later, analysed the scenarios and mapped them into the most suitable user stories based on TET and SPOD objectives.

From the technological point of view, the collective intelligence approach is followed by the application of agile methodologies, allowing citizens to participate already into the early development of the software [2].

## 1.1 ROADMAP

The deliverable is reporting the design, implementation, deployment and testing of the Alpha version of the SPOD. The sections below describe the process that was followed during the design.

To begin with, in Section 2, we describe the “Requirements”, coming from the DoW, from the research activities and from WPs 2, 3 and 4 of our project.

Then, in Section 3, we describe our “Design Principles and Technologies”, by motivating our technological choices for the SPOD design and implementation.

In Section 4, we address the non-functional requirements described in Section 2.2, by describing the “Architecture design of SPOD”, with all its components and the interactions among them.

In Section 5, we deal with the functional requirements provided in Section 2.1, showing the “System Design of SPOD”, with a brief description of all the features and how they are designed and implemented.

In Section 6, we present the “Development, Design and Testing” as it was followed in year 1, while in Section 7 we deal with some “Cross-Cutting Design Concerns”.

The software developed is available at the following URLs:

URL	Description
spodalpha.routetopa.eu	Alpha Version of SPOD: to be used for the review and not subject to change until the review (demo accounts for reviewers are presented in the Appendix)
service.routetopa.eu:7480	Open source code repository (requires sign-in but registration is free)
ckan.routetopa.eu	Sample installation of CKAN installation with demo data
spod.routetopa.eu	Global version of SPOD using for testing and early features preview (it will follow the development in the second year)
deep.routetopa.eu	The DEEP server and repository, with documentation, samples and Datalet creator
den Haag.routetopa.eu	Localised version of SPOD for Den Haag
dublin.routetopa.eu	Localised version of SPOD for Dublin
groningen.routetopa.eu	Localised version of SPOD for Groningen
issy.routetopa.eu	Localised version of SPOD for Issy-les-Moulineaux.
platform.routetopa.eu	Landing page for the ROUTE-TO-PA websites (offers information on both SPOD and TET)

This Deliverable is supplemented by Appendix where user demo accounts for the reviewers are given to access the SPOD Alpha version, that is accessible at the address <http://spodalpha.routetopa.eu> (the server will be kept at version 1.0 while the rest of the SPODs will follow the evolution in the second year with versions 1.1 etc.).

The Manual for the SPOD Alpha version (referred to in the text as the Manual) is available at the address:

<http://spodalpha.routetopa.eu/Manual/SPODManualCurrentVersion.pdf> .

## 2 REQUIREMENTS

---

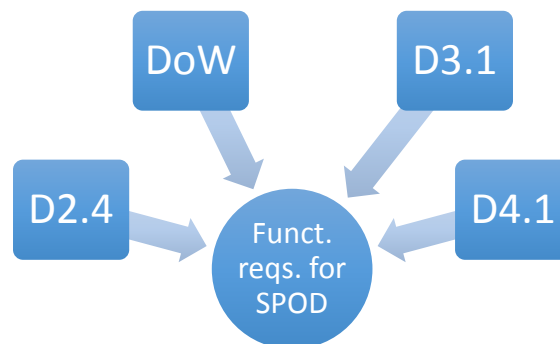
In systems engineering and requirements engineering, a “software requirement” is a software capability needed by a user to solve a problem to achieve an objective, by satisfying a contract, standard, specification, or other formally imposed document (see 3.2790 in [3]).

Requirements can be canonically distinguished in non-functional and functional. The former category encompasses all the requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviours. In a sense, they describe not what the software will do but how the software will do it. They are contrasted with the latter category of functional requirements that define specific behaviour i.e., broadly speaking, what a system is supposed to do.

*System design* details how the functional requirements are implemented, while *system architecture* details how the non-functional requirements are fulfilled. Then, we will use Non-functional and Functional requirements in Section 0 (Architecture design of SPOD) and Section 5 (System design of SPOD).

### 2.1 FUNCTIONAL REQUIREMENTS

The design and realization of the alpha version of SPOD (WP4) was achieved considering as input the analysis of “User and system Requirement”s Workpackage (WP2) through the D2.4, the “Models and Methods” Workpackage (WP3) through their deliverable D3.1, the requirements within this document and DoW (Description of Work, the Annex 2 of the Grant Agreement).



It must be said that in literature an extremely clear distinction between functional and non-functional requirements is not present. Sometimes, whether or not a requirement is expressed as a functional or a non-functional may depend by the level of detail to be included in the requirements document and by the comprehension of the context and by the experience of who is formulating the requirement.

We report here, first, the Functional requirements coming from WP2, WP3 and DoW and then, the ones that come from considerations on the state of the art tools and projects about advanced discussion tools for citizens.

#### 2.1.1 FUNCTIONAL REQUIREMENTS FROM WPS

A significative work in the project was conducted in close collaboration, by the WP2 (“User and System Requirements”) and WP3 (“Models and Methods”). In fact, they have produced two deliverables, DL2.4 and DL3.1 that provide detailed descriptions (motivated either empirically or grounded on users’ requirements) of the requirements/implications for the SPOD (and the TET, of course).

Of course, part of the requirements are contained in the original Description of Work (Annex 2 of the Grant Agreement) that describes the goals of the platform and, somehow, describe the way to get there (the functionalities).

Finally, it is not rare, in the design phase, that some properties of a system may be expressed either as a functional or non-functional property and it often happens that non-functional requirements may result in functional requirements statements. Therefore, among the possible source of functional requirements we also consider some of the non-functional requirements (listed in the column “DL 4.1”).

A list of functional requirements, were taken into account, grouped in categories and are described by the table below.

	Requirements (by category)	Notes	DL 2.4 Use cases and scenarios	DL 3.1	DL 4.1	DoW
	Social capabilities					
1	Login/Logout	Each user is logged				Section 2.1.2.2: Main concepts of ROUTE-TO-PA (Citizen-friendly environment for social information)
2	Unified login for SPOD/TET	Citizens should seamlessly move from TET to SPOD and viceversa without logging-in each time.	UC23: Login; s28.1: Login to TET/SPOD and enrich his profile			
3	Remote Authentication	Allow interoperability between different SPODS of different PAs. Authentication with Facebook is also technically feasible for Beta and will be considered	UC8: External Social Media Authentication			Section 2.1.2.2: "Cross-product integration": 1)Remote Authentication (like OPENID); principle "once only": simplification that consists in not requiring to citizen that Public Administration already has.
4	User Profile	Include editing of personal info (real name, gender, birthday, photo)	UC24: Enrich profile			
5	Private messages	Send private msgs to users	s17.14: Proposal through SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates participation. It facilitates open communication, a dialogue, a two-way flow of information in which citizens can make their voices heard		
6	Chat	Synchronous chat	s19.4: Chat on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates participation. It facilitates open communication, a dialogue, a two-way flow of information in which citizens can make their voices heard		

7	Attach a document/image	To any comment, it should be possible to attach documents and images	UC4(A): Attach a File as Support. UC5: Share a link s1.2: Insert file on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.		
8	Notifications of activities (what's new)	Notify all the activities (e.g. posts, likes, photos etc...) to everybody or only the friends		6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates participation. It facilitates open communication, a dialogue, a two-way flow of information in which citizens can make their voices heard		Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
9	<i>My Preferences: Privacy</i>	How my profile, presence, newsfeed and my friends are public to all, visible only to my friends or private			ROUTE-TO-PA privacy policy and further discussions on the user's control of personal data (sec. 7.1)	
10	Post a message (status)	Share a post or dataset visualisation	s1.3: post a message on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates participation. It facilitates open communication, a dialogue, a two-way flow of information in which citizens can make their voices heard		Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
11	Browsing users	Browse all the users, only online, search by name or gender				Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
12	Post a link or chart visualisation previously placed in personal "My space"	Share a link or chart visualisation from private space (e.g., My space page)		6.3 Activity System, Implication for TET/SPOD: "SPOD should facilitate a space where users can work on their own analyses and ideas and at the same time there should also be <b>a space where they can share information</b> and work together"		
13	Add friends and friends requests	Send a private message, add a friend or send a friends request				Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
14	Accessibility for users with visual deficiencies	In Alpha version only a particular type of visual deficiencies is considered (see sec. 4.5.1.1)	UC4(B): Share a dataset with visualisation		"Accessibility" (sec. 2.2.1.7)	

	<b>Welcome (Open Wall)</b>					
15	Provide welcome and summary (anonymous) information about the status of the SPOD, also to unauthenticated users	It enables and empowers social interactions by stimulating potential users in registering. Useful information include the number of users registered, logged users, number of discussions, recent content, etc. Useful to provide a URL to perspective potential users			"Usability" (sec. 2.2.1.3) "Familiarity " (sec. 2.2.2.2)	Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
16	Interactive navigation among datasets	Encourage users to see what is available in the associated dataset providers			"Usability" (sec. 2.2.1.3) "Familiarity " (sec. 2.2.2.2)	Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
	<b>What's new</b>					
17	Show all users' activities	Show all social activities (e.g., posts etc...) of all the users. Potentially (when the platform is in use) with many updates and therefore potentially long	UC22: "POST on SPOD" s1.1: "Browse on SPOD"	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.		
18	Show only friends' activities	Show all social activities (e.g., posts etc...) of all the friends. Since the list of activities of all the users may be very long, it is more helpful to see what the friends are currently interested in.		6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.		Section 2.1.3.1 "Social Platform for Open Data": supporting collaboration and social interaction
	<b>Public Discussion (Agora)</b>			6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.		
19	Create a Public Room	A discussion forum should be offered to the citizens on the SPOD	s4.1: Create a Forum on SPOD			
20	Recommended datasets	To stimulate the discussions, relevant datasets should be offered for specific discussions		6.3 Activity System. Facilitate a space where users can work on theirs analyses and ideas SPOD should facilitate a space where users can work on their own, and there should also be a space where they can share information and work together	"Usability" (sec. 2.2.1.3) "Familiarity " (sec. 2.2.2.2)	

21	Synthetic visualisation (graphical metaphor) of the number of comments	A participant should be able to see all the ongoing discussions and recognize easily the ones that are more active			"Usability" (sec. 2.2.1.3) "Learnability" (sec. 2.2.1.4)	
22	Synthetic visualisation (graphical metaphor) of the number of datasets used	A participant should be able to see all the ongoing discussions and recognize easily the ones that use more datasets, as discussions that are fruitfully employing open data			"Usability" (sec. 2.2.1.3) "Learnability" (sec. 2.2.1.4)	
23	Synthetic visualisation (graphical metaphor) of the number of views	A participant should be able to see all the ongoing discussions and recognize easily the ones that collect more interest among the users			"Usability" (sec. 2.2.1.3) "Learnability" (sec. 2.2.1.4)	
24	Synthetic visualisation (graphical metaphor) of the recency of activities	A participant should be able to see all the ongoing discussions and recognize easily the ones that have been more recently active (i.e., a comment was posted)			"Usability" (sec. 2.2.1.3) "Learnability" (sec. 2.2.1.4)	
	<b>Public discussion (Public Room)</b>			6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.		
25	Attach a dataset visualisation to a comment, by using a previously created datalet from "My space"	Import from "My space" a link or visualisation of a datalet into a comment/contribution to a room		6.3 Activity System, Implication for TET/SPOD: SPOD should facilitate a space where users can work on their own, and there should also be a space where they can share information and work together	"Usability" (sec. 2.2.1.3)	
26	Attach a dataset visualization to a comment, among suggested datasets or a new dataset	Attach a dataset visualization to a comment, using one of the suggested datasets for the public room, or using a new dataset			"Usability" (sec. 2.2.1.3). "Attach Design Visualisations" (sec. 2.1.2.3) "Familiarity" (sec. 2.2.2.2)	
27	Upload Image	Upload a media file (e.g., image) with a contribution to the discussion	s1.2: Insert file on SPOD			



28	Sentiment indicators for a post	Sentiment indicators (agree/disagree/neutrality) for a post	UC7: Attach Widgets and Tag Tools on SPOD forum			
29	Split view	Besides the time-dependent thread like visualisation of the comments, additional information is shown			"Split view" (sec. 2.1.2.1)	
30	Time-dependent thread-like visualisation of comments	On the left of a public room, the discussion is threaded, with 3 levels of comments, most recent at the top.		6.1 Three democratic traditions, Implication for TET/SPOD: SPOD should facilitate a space where users can work on their own, and there should also be a space where they can share information and work together	"Time-dependent thread-like visualisation of comments" (sec. 2.1.2.2)	
31	Visualisation of additional, synthetic information about the discussion	The participants can see the structure of the discussion by accessing different alternative views, centered around some of the components (like, e.g., users, data, opinions, etc...)			"Usability" (sec. 2.2.1.3) "Visualisation of Additional Information" (sec. 2.1.2.4)	
	<b>Dataset visualisation (Controllet)</b>					
32	Search box for datasets	Several types of search (e.g., search facilities while typing, search all dataset, search by typing only initial character of a word, search all dataset containing any word or letters)	UC4(B): Share a dataset with visualisation on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Usability" (sec. 2.2.1.3) "Familiarity" (sec. 2.2.2.2)	Section 2.1.1 "Objectives": Provide user- friendly services
33	Visualisation of one of the datasets among a list of easily accessible datasets	List of available dataset for each Pilot (Prato, Issy, etc.)	UC4(B): Share a dataset with visualisation on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Usability" (sec. 2.2.1.3) "Familiarity" (sec. 2.2.2.2)	

34	Visualisation of any dataset from a compatible provider	Remote address via API available from any CKAN or OpenDataSoft portals (e.g., TET-enabled CKAN platform, Issy-les-Moulineaux, Île-de-France etc...) allows dataset visualisation	UC4(B): Share a dataset with visualisation on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Interoperability" (sec. 2.2.1.2)	Section 2.1.2.2: "Cross-product integration": Integration SPOD with CKAN and UltraC@rity.
35	Choice of a chart	List of available charts: table, barchart, column chart, areachart, donut 3d and pie chart, treemap, leafletjs, bubble and bubble 3D, scatter chart and heatmap	UC4(B): Share a dataset with visualisation on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Usability" (sec. 2.2.1.3)	
36	Step by step	A wizard allows to create and share visualisation of a chosen dataset simply steps	UC4(B): Share a dataset with visualisation on SPOD	6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Usability" (sec. 2.2.1.3) "Familiarity" (sec. 2.2.2.2)	
37	Search box for chart	Visualisation of dataset with the search of a chart			"Usability" (sec. 2.2.1.3) "Familiarity" (sec. 2.2.2.2)	Section 2.1.1 "Objectives": Provide user- friendly services
38	Title and comments	Customize a chart with title or description			"Usability" (sec. 2.2.1.3)	
39	A preview of selected chart	A preview before sharing the dataset visualisation			"Usability" (sec. 2.2.1.3) "Familiarity" (sec. 2.2.2.2)	
40	Explore datasets with an interactive navigation	Select a dataset from an interactive tool for navigation, like, for example, a treemap built dynamically	UC4(B): Share a dataset with visualisation on SPOD		"Usability" (sec. 2.2.1.3) "Familiarity" (sec. 2.2.2.2)	
	<b>Personal reflections (My space)</b>					
41	Insert a dataset visualization into my space	A private space of an user	UC 28: Digest Information	6.3 Activity System. SPOD should facilitate a space where users can work on their own, and there should also be a space where they can share information and work together		
42	Take and insert notes into my space	A private space of an user	UC 28: Digest Information	6.3 Activity System. SPOD should facilitate a space where users can work on their own, and there should also be a space where they can share information and work together		

43	Take and insert a link into my space	A private space of an user	UC 28: Digest Information	6.3 Activity System. SPOD should facilitate a space where users can work on their own, and there should also be a space where they can share information and work together		
	<b>Open Data access</b>					
44	Associated dataset provider	Each SPOD is associated with a set of Dataset providers (TET, CKAN, OpenDataSoft, etc.) that is easily browsable and immediately accessible (and seachable) when inserting a new visualisation		6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Interoperability" (sec. 2.2.1.2) "Familiarity " (sec. 2.2.2.2)	
45	Easy navigation of associated dataset provider (treemap)	Graphical easy-to-use navigation of all the associated providers at once, directly from SPOD		6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates also collaboration e.g., being able to share documents, share visualizations and work together in these documents or visualizations.	"Usability" (sec. 2.2.1.3) "Familiarity " (sec. 2.2.2.2)	
	<b>Support to user</b>					
46	Manual easily accessible	Manual is easily accessible within SPOD	s5.1: Learn TET/SPOD		"Learnability" (sec. 2.2.1.4)	
47	Video instructions on YouTube easily accessible	Videos is easily accessible within SPOD	s5.1: Learn TET/SPOD		"Learnability" (sec. 2.2.1.4)	
48	Nationalisation in each language (EN, FR, IT, NL)	Selectable by the user	s5.1: Learn TET/SPOD		"Learnability" (sec. 2.2.1.4)	
49	Context-sensitive help	Tooltip or brief explanation to define page or widget of UI (e.g., button etc...), precisely how to interact with the control/page in question	s5.1: Learn TET/SPOD		"Learnability" (sec. 2.2.1.4)	
	<b>Support to management and integration</b>					
50	Integration with CKAN platform	Integration with CKAN portal			"Interoperability" (sec. 2.2.1.2)	Section 2.1.1 "Objectives": Integration of SPOD with Utrac@rity

51	Integration Ultraclarity	Integration with UltraClarity portal			"Interoperability" (sec. 2.2.1.2)	Section 2.1.1 "Objectives": Integration of SPOD with Utrac@rity
52	Roles	Actions are governed by the role and by access policies for each action (like administrators are the only users who can manage the platform)	UC24: Enrich profile	6.4 Tensions, Implication for TET/SPOD "In order to build a citizens community TET and SPOD should also facilitate the option of inviting outsiders to join by being able to send them graphs or visualizations that will lead them to SPOD or TET" (in this case the roles are needed to allow guest users into the SPOD)		
53	Customizing the "What's new" page with widgets	The page the shows the current activities can be customized providing extra information about what is happening in the SPOD		6.1 Three democratic traditions, Implication for TET/SPOD: SPOD facilitates participation. It facilitates open communication, a dialogue, a two-way flow of information in which citizens can make their voices heard		
54	New or customized page in SPOD	The administrator can help users to make easy-to-use interface in SPOD, adding or writing a new page			"Operability" (sec 2.2.1.5) "Reusability" (sec. 2.2.1.10)	"Sustainability and reusability" (sec. 2.1.2.1)
55	Adding multiple languages	The administrator can upload a new language to change SPOD platform			"Operability" (sec 2.2.1.5) "Learnability" (sec. 2.2.1.4) "Maintainability" (sec. 2.2.1.8)	"Sustainability and reusability" (sec. 2.1.2.1)

Already planned for Beta version:

Requirement	Notes	DL 2.4	DL 3.1	DL 4.1	DoW
<b>Events</b>					
Discuss about an off-line event	An event is added, with discussions and relevant datasets. Already planned for Beta	UC26: Add an Event	6.3 Activity System. SPOD offers links to off-line events, relevant to the community		
<b>Collaborative space</b>					
Small-group collaboration	Support to small, private (by invitation) group activities, including tools specifically designed for small groups. Already planned for Beta.		6.2. Organizational Level. SPOD, in the initial level (Collaborative Space) facilitates discussions around the meaning of data, and then, in the public room enables participation and collaboration		Small-group collaboration
<b>Open data access</b>					

Advanced support for charts	When presented with the choice of the charts, the Controllet will present some suggestions on the right kind of charts to be used for the dataset selected, based on the popularity (how many users have used that visualization chart for that dataset) but also on similarity of charts and on heuristics on content, besides (if present) the use of metadata	UC20: View Chart and UC21: View Infographic (see in "Special Requirements")		"Usability" (sec. 2.2.1.3)	
-----------------------------	--	---	--	----------------------------	--

Among the use cases described in D 2.4 specifically for SPOD and the ones described jointly for TET and SPOD, we have addressed in the Alpha a list of use cases (see table below). Indeed, the use cases that have not been addressed will be subject to further analysis and design considerations in the Beta version of SPOD (see table below).

Platforms	DL 2.4	Version
SPOD	UC4(A), UC4(B), UC7, UC22, UC25, UC23, UC24;	Alpha
SPOD/TET	UC5, UC8	Alpha
SPOD	UC9, UC18, UC26 (already planned), UC27, UC29, UC17, UC19	Beta

In summary, 9 out of 16 use cases were already covered by the first version of SPOD. All the implications in D 3.1 have been either addressed by Alpha version or planned for Beta (e.g., the collaborative space for small groups).

### 2.1.2 USER INTERFACE DESIGN FOR THE DISCUSSION ROOM (PUBLIC ROOMS)

The user interface of the discussion tools has been designed taking into consideration advantages and shortcomings reported in literature [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], in several projects [14, 15, 16, 17, 18, 19], and products [20, 21, 22, 23] supporting communities and grassroots initiatives.

In literature [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], several papers analyse the usage of computing systems (including email, chat, wikis, web forums, open innovation systems, group decision support systems, as well as debate and argumentation systems) to support crowd scale deliberation. These system can foster massive volunteer contributions, idea synergy as well as high-quality results (crowds can collectively make better judgments than the individuals). Time-centric systems (where the discussion follows a chronological timeline) have to face several challenges [24, 13, 12, 10]:

- Low signal-to-noise ratios: corpuses are highly redundant, and of highly variable quality. Enormous effort is required to “harvest” these corpuses for the most useful content.
- Insular ideation: ideas are typically generated quickly by single individuals without reference to other submitted ideas.
- Balkanization: users often self-assemble into groups that share the same opinions. This tends to lead people to take on more extreme, but not more broadly informed, versions of the opinions they already had.
- Non-comprehensive coverage of the problem space.
- Dysfunctional argumentation: postings are often bias- rather than evidence- or logic-based and no support is provided to help to correct and build upon each other’s facts and reasoning.
- Opaque Processes: it is very difficult to get a sense of the “health” of a deliberation, e.g. where the content is mature and ready to harvest, which topics need more attention, when dysfunctional behaviours are taking place and need to be addressed.

These issues seems mitigated by argumentation-maps tools [5, 24, 13, 10, 13], which provides a structure fostering several advantages:

- All content on a given question is co-located
- Foster more systematic and complete coverage
- Counteract balkanization by putting all competing ideas and arguments right next to each other
- Careful critical thinking is encouraged

However, argumentation-map tools present also some issue [7, 9]: they are founded on a hyper-rational view of collective deliberation and are designed to suppress conversational dynamics (confrontation, antagonism, rhetoric, and manipulative argumentation strategies). In other words, they privilege knowledge reification (the creation of a collective output) over users' participation. Mapping tools seriously disrupt people communication ways; furthermore users can have trouble to learn and use the argumentation formalism. Often, users enjoy more a platform supporting higher engagement and limited knowledge objectification capability than the opposite.

Further work in supporting grassroots initiatives is conducted by the partners of the CATALYST project [14], who are developing collective intelligence tools [20], [21], [25], [17], [23] for social and societal innovation.

These tools offer a wide spectrum of capabilities and kinds of visualization, that range from collective sensing (where a collective gathers data on its environment), through sensemaking (interpreting data to identify patterns that warrant action), ideation (developing ideas about which actions to pursue), decision-making (selecting the best actions), and finally collective action (implementing these actions in a coordinated effective way). The project aims at improving collaborative knowledge creation through mobilization of Collective Intelligence for the public good.

The Assembl tool [15] focuses on moving an unstructured debate toward a structured set of ideas. Ideas are extracted and organized (by expert users) into a table that provides an overview of the discussion. The unstructured discussion is key to facilitating the co-creation of new ideas while the structuring process allows people to quickly hone in on the area of discussion that interests them. The process works via 'rough consensus' to determine the direction of the debate. After the synthesis is sent out and validated, the discussion process enters a new cycle.

The Litemap tool [16] is a Web tool for mapping out visually the content of online debates across different forums and Websites. Users can harvest issues, ideas, pros and cons of public debates and connect them in meaningful network graphs. Litemap provides a wide set of Alternative Views and Statistic Analytics of the conversation, providing therefore further information to the community managers.

The DebateHub tool [17] provides communities functionalities to raise issues, share ideas, debate the pros and cons of each idea and vote contributions in order to collectively organize and progress good ideas forward. DebateHub is distinctive in its use of advanced analytics to show you the best argued ideas, and visualizations of your community.

The Edgesense tool [18] is a Drupal module that adds social network analytics to Drupal forum and community sites. By augmenting online conversations with network analytics, it aims to foster collective intelligence processes. Community managers can check who is talking to whom, which users are central vs peripheral in the conversation, who are the gateways etc.. Individual users might use Edgesense to find herself in the graph and get an idea of her position in the conversation.

The Collective Intelligence Dashboard tool [19] aims at monitoring, measure and understand the nature and quality of the collective intelligence processes emerging with the community debate. In other words, it is the place in which advanced analytics on social and conversational dynamics can be made visible and fed back to the community for further awareness and reflection on the state and outcomes of a public debate.

Based on the state of the art, each kind of visualization (of data and comments) present its own advantages, therefore we have designed the following set of visualization features which allows to integrate multiple visualizations of information. The overall goal, with these functional requirements of the discussion room, is to accomplish a step toward the goal of the non-functional requirement of "Usability" (see Sec. 2.2.1.3).

#### 2.1.2.1 SPLIT VIEW

Given the advantages of the several kinds of visualization both of comments and data, we aim to offer a combination of the visualization of users' comments with further information by splitting the view to provide different kinds of the conversation and data representations.

#### 2.1.2.2 TIME-DEPENDENT THREAD-LIKE VISUALIZATION OF COMMENTS

The main visualization of the conversation follows the time-dependent thread-like timeline. It offers a well-known not-disruptive communication way, which does not require users' training and leverages users' familiarity with similar wide spread tools (forum, social, ecc) to support system adoption [7, 9]. By splitting the view, this time-dependent thread-like visualization of comments can be integrated with further information. This requirement follows, in particular, from what the users are expecting from a Social Network user interface (as in the non-formal requirement of "Familiarity" (see Sec. 2.2.2.2)).

#### 2.1.2.3 ATTACH DATASET VISUALIZATIONS

In the thread-like conversation, the users have the possibility to include data visualizations in each comment. This feature is inspired by studies on data journalism blog [6], where comments can support collaborative sensemaking around data as well as preoccupation about non-present yet data. Moreover, it is useful providing (by using the split view) a graphical representation of structure of the comments including data visualizations. Again, here, users are at their ease with the concept of "attaching" something to a comment in a thread, because this is the traditional action in Social Networks (see the non-formal requirement of "Familiarity" in Sec. 2.2.2.2).

#### 2.1.2.4 VISUALIZATION OF ADDITIONAL INFORMATION

In literature, several additional synthetic views of a discussion (or of all ongoing discussions) can be shown to the participants to the discussion. As a classical example, a graph-like representation of the discussion is often used. The so-called comments graph is an argumentation map representing the structure of the conversation going on in the thread-like visualization. In the Deliberatorium project [23] a moderator creates a structured visualization of the discussion starting from the chronological discussion. In this direction, the Split View can be used to provide a structured visualization of the discussion. The thread-like and the structured visualization of the discussion should be interlinked, to allow users to explore each of the visualizations and can come back to the same point in the other one.

As shown by the Litemap tool [16] and the Collective Intelligence Dashboard [26], it is useful to show additional synthetic views, beyond the thread-like and the structured conversation. It is usually done by using the Split View that can offer a set of additional graphs representing different kind of information of the discussion. An example is a Users graph that shows how each user has responded to other users. An edge connects users that have interacted. As in the Edgesense tool [27], this feature allows users to have an overview about the relationships about them, to find themselves in the graph and get an idea of their position in the conversation, as well as to identify central and peripheral users in the conversation. The width of the edge can show also the amount of interactions (the thicker the lines, the more interactions).

As another example, and following the experiences in the Assembl tool [15] and the DebateHub tool [17], monitoring the conversation evolution and the building up of consensus is a critical point to support the organization and progress of good ideas. Thereby, providing synthetic views of the opinions in the discussions is another possible information to be conveyed.

Central, also, is to show relationships about the artefacts involved in the discussion, particularly in our setting, where dealing and interacting with datasets is required. Therefore, further useful information can be provided about the network of relationships among the comments including a dataset visualization. This allows firstly

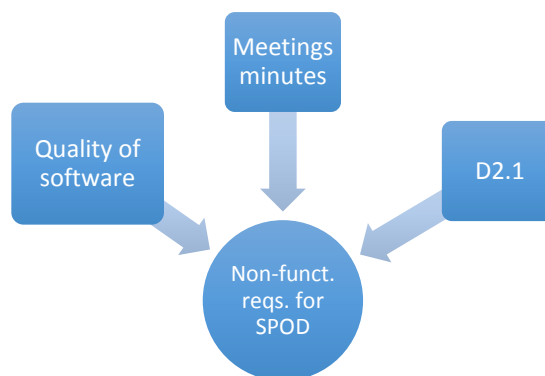


identifying quickly all the comments including data visualizations and then eliciting which conversation threads are more supported with data.

## 2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements specify criteria that can be used to judge the overall system, and not any specific behaviours. In a sense, they define *how*, and *how well*, a system should be, rather than define what a system is supposed *to do* (which is the task of functional requirements).

The non-functional requirements described below are described here on the basis of the research meetings of the project and the deliverable D2.1 “State of the art Report and Evaluation of existing open data Platforms” in particular, about the perceived barriers, and the generic characteristics that an open data Platform should have, as described by ISO/IEC standards on the quality of software.



### 2.2.1 NON-FUNCTIONAL REQUIREMENTS FROM THE QUALITY IN SOFTWARE

The ISO/IEC 25010:2011 standard [28] defines two models: a “quality in use” model and a “product quality” model. Some of the features of the first model can be helpful in determining some SPOD non-functional requirements.

The “quality in use” model consists of 5 characteristics (and subcharacteristics) that are related to the outcome of interaction when the system is used in a particular context. A “product quality” model consists of 8 characteristics (and sub-characteristics) that relate to static and dynamic properties of software and the computer system.

We use the ISO/IEC 25010:2011 characteristics and subcharacteristics in order to provide a standard and consistent terminology for specifying and evaluating the non-functional requirements of our system.

Some of the terminology definitions that are relevant and of some interest for SPOD (as from the research and plenary meeting conducted in the project first year), taken verbatim from the ISO standard [28], are described in the following sections.

#### 2.2.1.1 PERFORMANCE EFFICIENCY

It is the performance relative to the amount of resources (software, hardware, etc.) used under stated conditions. In particular, it deals with *time behaviour*, defined as the degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements; *resource utilization*, defined as the degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements; and *capacity*, defined as the degree to which the maximum limits

of a product or system parameter meet requirements, where the parameters can be, in our case, the number of concurrent users, the communication bandwidth, throughput of transactions, and size of database of the SPOD.

#### 2.2.1.2 INTEROPERABILITY

It is the degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.

#### 2.2.1.3 USABILITY

It is defined as the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

#### 2.2.1.4 LEARNABILITY

It is considered to be the degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.

#### 2.2.1.5 OPERABILITY

It is defined as the degree to which a product or system has attributes that make it easy to operate and control.

#### 2.2.1.6 USER INTERFACE AESTHETICS

It is the degree to which a user interface enables pleasing and satisfying interaction for the user.

#### 2.2.1.7 ACCESSIBILITY

It is defined as the degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

#### 2.2.1.8 MAINTAINABILITY

It is the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers. Modifications can include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

#### 2.2.1.9 MODULARITY

Defined as the degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.

#### 2.2.1.10 REUSABILITY

It is the degree to which an asset can be used in more than one system, or in building other assets.

#### 2.2.1.11 PORTABILITY

The degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another. Portability can be interpreted as either an inherent capability of the product or system to facilitate porting activities, or the quality in use experienced for the goal of porting the product or system.

#### 2.2.1.12 ADAPTABILITY

It is defined as the degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.

#### 2.2.1.13 REPLACEABILITY

It is the degree to which a product can replace another specified software product for the same purpose in the same environment (e.g., replaceability of a new version of a software product is important to the user when upgrading).

#### 2.2.1.14 FLEXIBILITY

Defined as the degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements.

### 2.2.2 DOMAIN-SPECIFIC NON-FUNCTIONAL REQUIREMENTS

Several considerations, specific to the application domain (transparency and open data) and to the peculiar context (3-years research and innovation project in EU Horizon 2020) suggest also some additional non-functional requirements.

#### 2.2.2.1 UNIVERSALITY OF ACCESS

It is important, given the nature of the project, to ensure a wide impact on citizens in Europe. In order to do so, the *widest* set of devices and software should be supported by the project. This statement is, of course, mitigated by the finite amount of resources that are devoted to the project and the definite timeframe for results. Nevertheless, we are conscious that a significant requirement is ensure the diffusion of the project in any way. Moreover, it must be considered that our project is targeting the European scenario in 2018, being the final release in January 2018, and the architectural choices must ensure to cover a significant amount of possible alternatives, in order to avoid being stuck in a technological *cul-de-sac* where the chosen platform happens to be no longer in use. Therefore, our choice (from the proposal) to use the Web, of course, is positive, although it needs some considerations on the choice of the client side, both from the software and from the hardware point of view.

From the hardware point of view, the **mobile scenario**, where the citizens happen to be spending a consistent part of their “digital life” via a mobile access, will be the most used ones, and therefore, with all the limitations of a small size screen, the platform should be targeting smartphones and tablets, at least in its perspective. To wit, Gartner Group [29] is saying that “*By 2018, more than 50 percent of users will use a tablet or smartphone first for all online activities*” and Ericsson [30] is expecting that “*By 2020, around 90 percent of the world’s population will be covered by mobile broadband networks*”.

Another (orthogonal) instance of the universality requirement is the need to cover the widest set of WWW browsers in the widest set of operating systems. In this context, again, it is importante to consider the timeframe of the results and their impact, thereby avoiding to place resources on browsers that are doomed (by their manufacturers) to be discontinued in a short-medium time.

In conclusion, we will define “Universality of access” as the “*widest range of browser/devices and software that the project is reasonably able to support with the given resources, allocated to the project*”.

#### 2.2.2.2 FAMILIARITY

We are designing a particular system a “Social” platform, in a world where users are very used to advanced services provided by world-size software giant companies, such as Facebook, Twitter, Google, etc. that can be

used with a very sophisticated and thereby simple and easy to use interface. Our task is, therefore, very hard and challenging in that we are somehow offering a service to users that are expecting an elaborated and advanced service, but, seeing our “glass half full” we can count on their familiarity with several complex instruments and “attach” our research and innovation on that.

### 2.2.3 NON-FUNCTIONAL REQUIREMENTS FROM STATE-OF-THE-ART DELIVERABLE D2.1

In the Deliverable 2.1 “Review of the State of the Art and Evaluation of Existing Open data Platforms”, among other issues, is addressing the perceptions of different categories of stakeholders of the shortcomings of the current open data Platforms. In this context, the stakeholders were stimulated in identifying some desirable platform features and characteristics of open data Platforms, that were later used to evaluate and compare existing work.

Part of the work in D2.1 is very relevant to identifying non-functional requirements, as the perceptions of stakeholders are a very valuable input. As a matter of fact, the results outlined in the D2.1 are reinforcing several of the non-functional requirements already outlined in the previous subsection. In particular, it is recognized that Usability (sec. 2.2.1.3), Accessibility (sec. 2.2.1.7), Flexibility (sec. 2.2.1.14), Learnability (sec. 2.2.1.4) and Operability (sec. 2.2.1.5) are all important and relevant characteristics, that are expected by an open data Platform.

## 3 DESIGN PRINCIPLES AND TECHNOLOGIES

---

### 3.1 COMPONENT-BASED DEVELOPMENT: “REUSE RATHER THAN REINVENT”

One of the guidelines to long-term success of ICT in Transparency [1] is “Reuse rather than reinvent” and we adopt as project “mantra” both from technological point of view (i.e., integration of existing popular open source products) and from the PA point of view (plug ROUTE-TO-PA onto existing experiences and needs by involving the Pilots).

As underlying principle, our project is, in fact, devoted to provide results that are sustainable and reusable.

In order to further improve the sustainability and the reusability of our results, we adopt “Free and Open Source” (FOSS) software development model since (as recognized by EU guidelines) open source software favours integration, standardization, interoperability, diffusion and reusability of software products, thereby enhancing their *effectiveness* on the technological, social and economical context and improving the *efficiency* in terms of resources and budget.

In order to achieve the aim of stimulating discussion and interactivity through the wider uptake and best use of ICT-based platform by citizens, governments and businesses, our project is aiming at realising a sustainable infrastructure for open data and effective citizen engagement strategies, named Social Platform of Open Data (SPOD).

SPOD is a social platform designed to enhance and make transparency the value of data between the Public Administration and citizen through interactive visualization of dataset made available from open data portal

powered by CKAN (e.g. TET platform or any CKAN portal), by OpenDataSoft (e.g. open data portal of Issy-les-Moulineaux). SPOD provides an user-friendly interface to publish and start social discussion exploring datasets of the open data portals. SPOD is available in different languages (English, Italian French, Dutch).

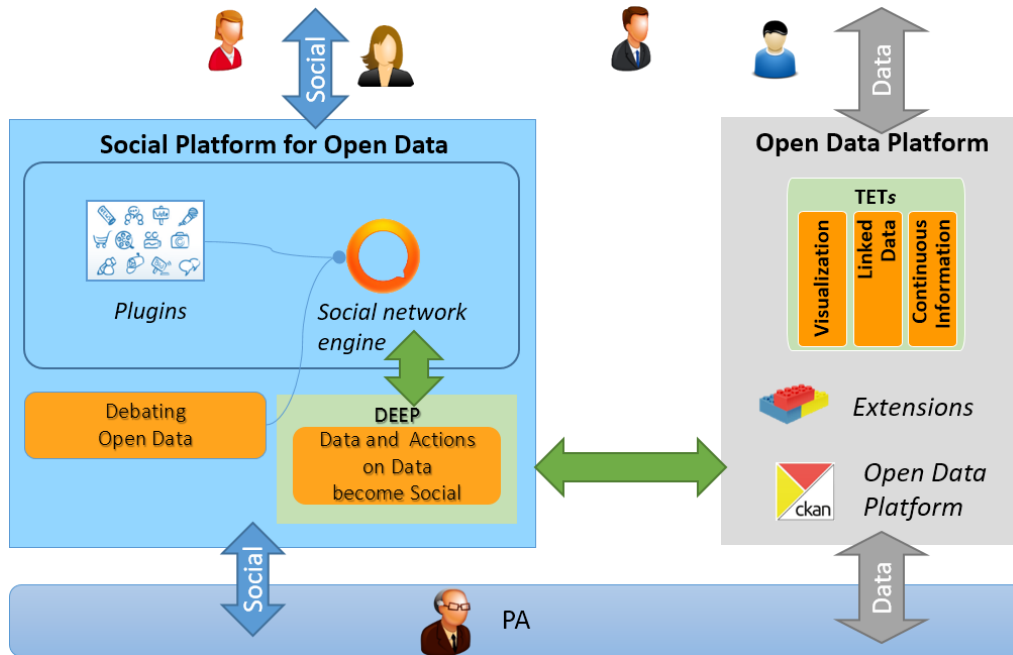


Figure 1: A general overview of ROUTE-TO-PA project: SPOD on the left side and TET on the right side.

SPOD (shown on the left side in Figure 1) is an infrastructure that includes a social platform component, and an open source, extensible, modular and pluggable architecture, **DataEcosystem Provider (DEEP)** providing datalets, which are web-component visualizations of datasets content that enable the sharing and collaboration around data visualizations [31], [32]. The entire project is strongly based on existing software platforms, whose *integration* allows to reach sustainable, reusable and effective results with a limited timeframe.

### 3.2 TACKLING THE HETEROGENEITY OF THE CONTEXT

Our design is pervasively inspired by the heterogeneity of the context we place our technologies in. In a sense, this is both a motivating factor as well as an always present challenge, and our technological and design choices do not dodge heterogeneity and, on the contrary, tackle it and, if possible, make it a starting point for the design. Heterogeneity in our context comes from different sources. First and foremost, our design is meant to address the *heterogeneity of Public Administrations*, that differ significantly in Europe in terms of level of engagement with Open Data, relationships of engagement with citizens and social context. Since we plan different pilots (see D5.1) and many PAs are partners in the project, thereby providing first-hand input in the design, the heterogeneity of different PAs is addressed since the very beginning and subsequently considered for the evaluation.

At the same time, *audience heterogeneity* is very important in the design, as our social platform will have to be placed in contexts where both citizens (with no particular technical skill) and skilled users (PAs employees, policy makers, activists, data journalists, ...) are going to collaborate together. The different levels of skills and competencies between, for example, journalists in the role of watchdog or citizens, is described by D3.1 in sec. 6.3.

Then, in our project we tackle *Open Data heterogeneity*, that comes in different dimensions. First, there is the *heterogeneity of quality* and kind of datasets available, in terms of structure, file format, machine-readability, standards to identify elements, and linking. Our approach is to try to be less demanding as possible on the requirements. Heterogeneity here is also manifested through the *variety of platforms* that are available, and our

platform is addressing different servers already and will be widening the compatibility in the beta and final versions.

Finally, it must be mentioned that our platform is meant to address the *heterogeneity in the societal context*, as described in Sec.2 of D3.1, namely Monitorial democracy, Deliberative democracy, and Participatory democracy, by providing different tools for social interactions, reflection and collaboration.

### 3.3 BUILDING BLOCKS AND SOFTWARE TECHNOLOGIES

In this section we describe, briefly, the existing systems and the technologies that are needed to leverage upon for the integration. We first provide a brief analysis of the state of the art in each field, and then provide the motivations for the choice. Later, in section 0 (Architecture design) and 5 (System Design), we describe how each building block has been integrated and modified to be connected to a distributed architecture, to provide the functionalities required.

#### 3.3.1 THE SOCIAL PLATFORM

Although in the DoW, the mentioned FOSS server for Social Network was Elgg, we decided to refine and review the choice, considering the time passed from the original design of the proposal to the effective implementation. We first, searched through the social network platforms that were available for integration. We chose some initial criteria to scan through the list, in order to set the minimum standard that were required:

- FOSS (Free and Open Source Software): the license of the software must be FOSS
- License: the kind of FOSS license should allow the kind of integration that we are aiming at
- Quality of code
- Community Support
- Number of ready-to-use tools
- Responsive UI
- Documentation (presence and quality)
- Development continuity
- Diffusion
- Plug-in based architecture
- ORM (Object-relational mapping) for data persistence
- Ability to support hundreds of simultaneous users
- Plug-in creation complexity
- Theme creation complexity

Only two projects meet all the selected requirements Elgg and Oxwall.

##### 3.3.1.1 ELGG

Elgg [33] is an open-source (GPL 2) web based social network. It runs on LAMP (Linux, Apache, MySQL, and PHP) or WAMP (Windows, Apache, MySQL, PHP) platform. It offers a networking platform, combining elements of blogging, e-portfolios, news feed aggregation, file sharing, and social networking. Elgg has a plug-in based architecture, and is extensible through custom plug-in. Elgg is written in PHP, uses MySQL for data persistence, and includes jQuery for client-side scripting.

##### 3.3.1.1.1 Elgg main features

- Blog: every registered user has a blog connected to its account. A blog post can be public or private and other platform users can comment blog post.

- Bookmarks : users can exchange and comment link
- Groups: probably the most used and important feature. This tool allows creating virtual community that share common interest. Inside a group, Elgg provides a set of tools that foster collaboration and communication. In particular are present tool for blogging, bookmarking file sharing and page creation. The group can be public or private.
- Page: this tool allows asynchronous collaboration for page creation through a rich text editor.
- Messages: Elgg has a built-in mail message service that allows intra platform private communication.
- The wire: is a microblogging tool very similar to Twitter and it allows the creation of 140 character long post. This tool can be connected to an existing Twitter account.

#### 3.3.1.1.2 Elgg Architecture

Elgg architecture is based on Model-View-Controller design path. Elgg has a small core that handle user creation, user preferences and site administration. The other functionalities are provided through plug-in, this architecture is very flexible and easily extendible.

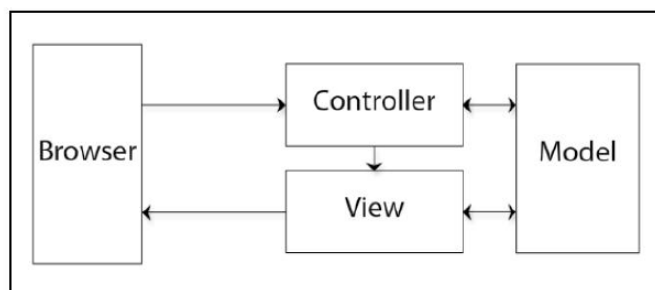


Figure 2: Elgg MVC Architecture.

#### 3.3.1.2 OXWALL

Oxwall [34] is a free and Open Source Software (FOSS) written in PHP based on MVC design pattern. It is created to power online social networks and community-enabled websites for various purposes. Oxwall project is developed and curated by Oxwall Foundation. Oxwall provides all the classical mechanism of social networks. It gives friendship mechanism, allows post or image comments, group or event creation/joining. Oxwall is written in PHP use MySql for data persistence, it exploits Smarty PHP template Engine for view layer and Javascript/JQuery for client side page manipulation. Oxwall architecture is plugin-based. It has few core-plugin that handle base platform functionalities (user management, platform management, access management) the other functionalities are provided by plugins.

##### 3.3.1.2.1 Oxwall main features

- Dashboard shows all the user friend activity (posts/likes/photos) by date
- Main shows all the platform event (posts/likes/photos/events etc.) even for not related users
- Members permits active or registered user visualization/exploration and search
- Forum is a classical topic based forum. Only Oxwall administrator can create a new Forum, every logged user can create a new topic inside a created forum (if it's role is enabled). The administrator can promote a user to Forum administrator. Forum can be configured as private for rule of users e.g. guest, free and moderator.
- Groups allows to create a small community inside the platform based on common interests/topics. On Group creation, the group owner can decide if the group is publicly visible or visible by invitation and if the group participants can invite other user or only the group creator can invite other user. The Group

creator can delete other user content and ban user from the Group. Group can have forum associated. Every group has its newsfeed. User activities in the group are published inside its Dashboard. Inside the Main them are published only the join to a Group.

- Blogs are associated to every registered user in the platform. Moreover, Oxwall presents a page where collects the most recent/top rated/most discussed blog posts. Is possible a research by tags.
- Event allows every logged user to create an event (and specifying who can view it [Anybody, By invitation] and who can invite [Participants, Event Creator]). Event plugin has its Wall, all the invited users can post a comment inside the Event Wall. User can signal if it will attend the Event and inside the Event main page there is a recap of invited users.

### 3.3.1.2.2 Oxwall architecture

Oxwall implements the Model View Controller design pattern. Oxwall, as mentioned before, is entirely written in PHP for the server side, exploits MySQL for data persistence, Smarty PHP template Engine for View layer and Javascript/JQuery for client side page manipulation. The data persistence, also, is “plug-in based”. Each plug-in must explicitly handle its persistence layer. Every plug-in developer must create the necessary table/s inside the database for data persistence and must create the ORM class to handle with the table/s.

Moreover Oxwall has an event driven mechanism for platform extension. Every action inside the Model or Controller can trigger an event. Every plugin can register to an event in order to perform additional action for that specific event. This architectural organization allows altering even the base functionalities of the platform with minimum impacts on the source code.

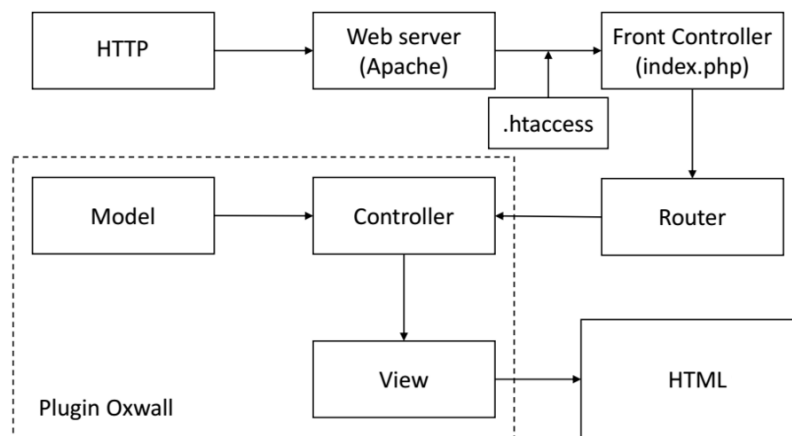


Figure 3 - Oxwall MVC Architecture

In Figure 3, we show the page lifecycle in Oxwall. It implements the Front Controller design pattern. In this design pattern all the request to the platform are addressed to a single point (the *Front Controller*). The Front Controller uses a Router class that, based on requested address, selects the plug-in that will process the user request. In addition, the plug-in internally must respect the MVC pattern. Every plug-in has private database table/s for persistence. Programmer can create custom database table/s for its needs.

In order to modify base platform behaviour through plug-in, we used Event. Event are generated by action inside the platform, a plug-in can register to Event and perform action that alter/enhance the standard platform operation.

### 3.3.1.3 THE COMPARISON AND OUR CHOICE: OXWALL



Both Elgg and Oxwall are web based MVC project based on LAMP (or WAMP) stack. From a technological point of view the two project are close. Therefore we decided to dig deep into each platform implementation.

Three are the main criteria that made us choose Oxwall instead of Elgg

- **Code structure:** Elgg codebase has a mixed approach, the Elgg core is written in pure Object Oriented paradigm while for plug-in implementation is possible to choose an Object Oriented, or a Procedural approach. This architectural design delegates the decision to the developer, in our case (since, potentially, several programmers from several institution will develop SPOD and maintain) this could lead to low quality code or mixed approach in plug-in creation. Therefore, the entire platform will be difficult to maintain. Oxwall, instead, has a really strong structured approach. Both core code and plug-in code are Object Oriented, and plug-in structure is very rigid. So plug-in looks very similar in their structure. This offers several advantages on code maintenance and quality. As “Maintainability” is one of our non-functional requirement (see Sec. 2.2.1.8), it is a very important aspect to consider.
- **Template engine:** MVC paradigm expects strong separation between Model, Control and View layer. In Elgg the View layer is based on plain PHP/HTML/CSS, this approach delegates to the programmer all the complexity of software layers separation. Poor architectural choices in plug-in implementation can easily lead to mix up MVC layers and consequently lead to poor quality code. Oxwall, instead, uses Smarty a PHP template engine. This engine allows an easier software layer separation and permits viewing code reuse, addressing the non-functional requirements of Modularity (see Sec. 2.2.1.9) and Reusability (see Sec. 2.2.1.10).
- **Mobile friendly:** Oxwall has a built-in theme for mobile devices and every plug-in can handle special Views (UI) for mobile devices. In Elgg we have to write an ex-novo theme for mobile devices, both for core plug-in and for core functionality, this is a high time consuming task. This addresses the non-functional requirement of “Portability” (see Sec. 2.2.1.11).

	Elgg	Oxwall
License	Open Source under GPL 2.0	CPAL 1.0
Cost	Free	Free
Source code	Development SVN	Yes
Codebase	PHP, MySQL/PostgreSQL	PHP, MySQL
Social Grouping	Yes	Plugin

Table 1: Features between Elgg and Oxwall

The main criteria that made us choose Oxwall instead of Elgg are the “code structure”, “template engine” and “mobile friendly” discussed in details in the previous section (see section 3.3.1.3). Moreover, we studied the “online reputation” of Oxwall and Elgg through Google

Keyword Trend <sup>1</sup> and we find out that in the last two years Oxwall trend is quickly growing instead Elgg trend is decreasing. So, even if Oxwall is a younger project is quickly becoming known worldwide.



### 3.3.2 LIBRARIES FOR DATA VISUALIZATION AND CHARTING

Data visualization is a relevant aspect to display Open data and SPOD platform needs clear, effective, interactive ways to be able to explain the data correctly and that makes it all the more important to choose the right kind of charting library and application frameworks that best matches the needs of SPOD.

<sup>1</sup> <https://www.google.com/trends/>

### 3.3.2.1 CRITERIA AND FEATURES

The choice of the charting library to create a good data visualization for our SPOD platform has been done by following guideline mentioned below:

- **Cross Browser Compatibility:** a charting library must be compatible with all browsers or modern browser taking in account the target audience.
- **Cross Device Compatibility:** a charting library must be used and responsive both on desktop that on the hand-held devices.
- **Input Data Format:** in the charting libraries, JSON (Javascript Object Notation) is becoming the standard format especially for the charting libraries.
- **Customizability:** this decision factor is important to configuring legends, attaching events (e.g., hover, click) and so on.
- **A set of available charts :** consider availability of all the charts offered by the library that will be used to our SPOD platform.
- **Performance:** many factors must be consider, such as, size of library, memory usage, garbage collection and number of browser repaint cycles.
- **Exporting:** this feature is applicable when you want to export chart to PDF or images (e.g., common export formats are JPEG, PNG, SVG or PDF).

The following criteria were used for the choice of charting libraries that are also free and Open Source with options like interactive and animated:

Rendering Technologies	Features	Platform	Supported Charts Types
HTML5/Javascript	<ul style="list-style-type: none"><li>• Ability to zoom in and out of charts</li><li>• Annotations on the chart</li><li>• Combination of charts</li><li>• Data label</li><li>• Date-time axis</li><li>• Dynamic charts</li><li>• Export files</li><li>• Interactive (responds to mouse hover/click)</li><li>• Print</li><li>• Text Rotation for Labels</li></ul>	<ul style="list-style-type: none"><li>• Firefox</li><li>• Internet Explorer</li><li>• Google Chrome</li><li>• Safari</li><li>• Opera</li><li>• iPhone</li><li>• iPad</li></ul>	<ul style="list-style-type: none"><li>• Data table</li><li>• Bar chart<ul style="list-style-type: none"><li>○ Stacked</li><li>○ Horizontal</li></ul></li><li>• Column chart</li><li>• Column 3D</li><li>• Line chart</li><li>• Timeline</li><li>• Donut, 2d e 3D Pie chart</li><li>• Bubble, Bubble 3d charts</li><li>• Scatter chart</li><li>• Area Chart</li><li>• Interactive geographic Maps with a marker with text or image in a popup</li><li>• Tree Map</li><li>• (New) Bubble map</li><li>• Combination charts: multiple axes</li><li>• Force-Directed Graph</li></ul>

Table 2: all criteria of charting libraries

### 3.3.2.2 AVAILABLE LIBRARIES

A list of JavaScript libraries for chart or graph to compare before choosing the adequate and the right charting libraries for SPOD.

#### 3.3.2.2.1 Dygraphs

Dygraphs is fast, flexible open source javascript library that allows users to explore and interpret dense data set. It is customizable, it works in all major browsers with zoom and mobile and tablet devices support.

#### 3.3.2.2.2 Leaflet

Leaflet is a javascript library for maps based on OpenStreetMap. It provides an interactive geo-localized data visualization in an HTML5/CSS3 wrapper.

It is extendible with a wide range of plugins with specific functionality such as animated markers, mask and heatmaps.

#### 3.3.2.2.3 jqPlot

jqPlot is a jQuery plugin for line and point charts. It has a few additional features such as the ability to generate trend lines automatically. It is extensible by plugins. There are plenty of hooks into core code allowing for custom event handlers, creation of new plot type and more. It has been tested on IE7, IE8, Firefox, Safari and Opera.

#### 3.3.2.2.4 D3.js

It is a very powerful JavaScript library that uses HTML, SVG and CSS to render different set of diagram and chart from a variety data source. It is, more than most, capable to provide some seriously advanced visualization with complex features and includes some advanced user interaction support.

#### 3.3.2.2.5 Highcharts

Highcharts is a Javascript charting library with a very huge range of chart options available. It uses SVG for modern browsers and VML in Internet Explorer.

All charts have a really attractive looks and animation. It is well documented and used by a tens of thousands of developers (great community support) and it is also very simple to use.

#### 3.3.2.2.6 Google Charts

It is a set of powerful chart tools that provides a way to visualize data on a website.

The charts gallery provides a large number of ready-to-use chart type, from simple line chart to complex hierarchical tree map.

It is used by embedding simple Javascript in the webpage. Charts are highly interactive and expose events that the developer could use to create complex dashboard or advanced user interaction mechanism.

All charts are rendered using HTML5/SVG technology to provide cross-browser compatibility.

#### 3.3.2.2.7 Crossfilter

It is a Javascript library for exploring large multivariate datasets in the browser.

Crossfilter is specialized to support extremely fast (<30ms) interaction with coordinates view with datasets containing a million or more records.

This library displays data and make user able to restrict the range of data and see other linked charts react.

#### 3.3.2.2.8 Polymaps

Polymaps is a free Javascript library for making dynamic, interactive vector-tiled maps and images in modern web browser using SVG.

It allows to define the design of the data by CSS rules and provides the display of multi-zoom datasets over maps, supports a variety of visual presentation for tiled vector data. It is ideal for showing information from country level on down to state, cities and individual streets because of it can load data at full range of scales.

### 3.3.2.2.9 Flot

Flot is an attractive Javascript plotting library for JQuery with focus on simple usage, attractive looks and interactive features. It is extensible by plugin and provides a basic support for lines, points, filled areas, bar and other type of charts.

### 3.3.2.2.10 Raphael

It is a very simple and small Javascript library that provides support for a wide range of data visualization options, which are rendered using SVG.

### 3.3.2.2.11 jQuery Visualize

Written by the team behind jQuery's ThemeRoller and jQuery UI websites, jQuery Visualize Plugin is an open source charting plugin for jQuery that uses HTML Canvas to draw a number of different chart types. One of the key features of this plugin is its focus on achieving ARIA support, making it friendly to screen-readers.

### 3.3.2.2.12 OpenLayers

OpenLayers is probably the most robust of mapping libraries. The documentation is not great and the learning curve is steep, but for certain tasks nothing else can compete. When you need a very specific tool no other library provides, OpenLayers is always there.

### 3.3.2.2.13 Dimple

Dimple is a library to aid in the creation of standard business visualizations based on D3.js. It makes easy for anyone, analyst or not, to develop stunning, three-dimensional graphics without any real JavaScript training. The dimple API tested against Firefox, Chrome, Safari and IE9. Its browser support is largely inherited from d3 so using it on IE8 and earlier will be difficult/impossible.

### 3.3.2.2.14 Google Maps

Google Maps is a desktop web mapping service maintained by Google. It offers a set of JavaScript APIs to provides supplemental features and several type of maps and layers such as satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in Beta), or public transportation.

### 3.3.2.2.15 A comparison and our choices: HighCharts, Google Charts and D3.js

We evaluated all the libraries for 2D and 3D charts described above and we decided, first, to base Highcharts and Google Charts, motivated by the following criteria:

- They cover all type of charts available in the other libraries
- They offer a larger and more advanced set of visualizations
- They have a better quality of code, performance, cross browser/device compatibility and offer a better way to customize the style of the charts.



Moreover we decided to use also D3.js because it is a very powerful low-level library that allows us to create specific and advanced types of visualization (such graphs for instance).

The tables below show the main features that positively suggested the adoption of Highcharts, Google Charts and D3.js.

	HighCharts	Google Charts Tools	D3.js
--	------------	---------------------	-------

<b>Charts and Map Types</b>	2D and 3D Chart types, Maps	2D charts types (13 availables). Maps (GeoChart)	Not available pre-build charts. Available a library of 200+ example
<b>Download and modification of Source code available</b>	✓	✓	✓
<b>License</b>	Free for non commercial use	Free for all usage	BSD-3
<b>Documentation</b>	Getting started guide, API references and examples	Getting started guide, API references and examples	Text and video tutorials, API references and a gallery of examples
<b>Support Options</b>	Community forum	Discussion group	Discussion group

Table 3: an overview among chart types

#### Features among the charts

	HighCharts	Google Charts Tools	D3.js
<b>Animation</b>	✓	✓ Turned off by default	✓
<b>Drill-down</b>	Using JavaScript code	Using JavaScript code	✓
<b>Tool-tips</b>	✓	✓	✓
<b>Legend</b>	✓	✓	✓
<b>Zooming</b>	✓	✗	✓
<b>Scrolling/Panning</b>	✓	✗	✗
<b>X-Axis Label Management</b>	✓	Not available	✓
<b>Multiple Axis</b>	✓	✓	✓
<b>Number Formatting</b>	✓	✓	✓
<b>Gradient</b>	✓		✓
<b>Export charts as JPG, PNG or PDF</b>	✓	✗	✗

Table 4: A comparison among chart types

	HighCharts	Google Charts Tools	D3.js
<b>Column Charts</b>	2D and 3D charts	2D only	✓
<b>Bar Charts</b>	2D and 3D charts	2D only	✓
<b>Line Charts</b>	✓ (zooming and panning)	✓ (no zooming and panning)	✓
<b>Area Charts</b>	✓	✓	✓
<b>Pie and Donut Charts</b>	✓	✓	✓
<b>Scatter and Bubble charts</b>	✓	✓	✓
<b>Combination of column, line and area charts across multiple axis</b>	✓	✓	✗

<b>Additional chart types</b>	Box plot, Error bar, Polygon series, spiderweb	Treemap charts, Stepped Area Charts, Table Charts / Grids	Non standard visualization
-------------------------------	--	---	----------------------------

Table 5: A comparison among chart types

Regarding the library for maps, we selected Leaflet respect to OpenLayers, Polymaps and Google Maps, because it is based on OpenSteerMap, like OpenLayers and Polymaps, but it adds a software layer to make develop easier. Furthermore, Leaflet respect to Google Maps is a library not a service and this mean that is possible work with code if necessary.

### 3.3.3 WEB PROGRAMMING FRAMEWORKS

#### 3.3.3.1 WEB COMPONENTS

In order to allow the creation of reusable web widgets in the SPOD platform we considered using the web components standard, currently being produced by Google engineers compliant with W3C specification [35].

The intention behind them is to use component-based software engineering to develop a bundle of web widgets to use whenever needed without having to rewrite the common fragments shared by several pages of the platform.

The components model allows for encapsulation and interoperability of individual HTML element.

The web component technologies allow to create your own HTML elements and the support for these components is present in some WebKit-based browsers like Google Chrome and Opera and is in Mozilla Firefox (requires a manual configuration change). Microsoft's Internet Explorer has not implemented any Web Components specifications yet. Backwards compatibility with older browsers is implemented using JavaScript-based polyfills (a library you can import in the web page that implements the web component specification).

Web components specifications consists of four elements, which can be used separately or all together:

- Custom Elements (<http://w3c.github.io/webcomponents/spec/custom/>): it defines the method to create new types of DOM elements in a document. The element definition consists of custom element type, local name, namespace, custom element prototype and lifecycle callbacks.
- HTML Imports (<http://w3c.github.io/webcomponents/spec/imports/>): it is a way to include and reuse HTML documents, typically web component definitions, in other HTML document called master document. The imported definitions are linked as external resources to master document.
- Templates (<https://html.spec.whatwg.org/multipage/scripting.html#the-template-element>): it describes a method to declare inert DOM subtree in HTML and manipulate them to instantiate document fragment with identical content.
- Shadow DOM (<http://w3c.github.io/webcomponents/spec/shadow/>): it defines a method of establishing and maintaining functional boundaries between DOM tree and how these trees interact with each other within document enabling better functional encapsulation within the DOM.

At this time the major libraries for working and extending web components are:

- X-Tags by Mozilla (<http://x-tag.github.io/>): "X-Tag allows you to easily create elements to encapsulate common behaviour or use existing custom elements to quickly get the behaviour you're looking for. For example, check out Brick, which contains 14 custom elements optimized for Web Applications."
- Polymer by Google (<https://www.polymer-project.org/1.0/>): "Web Components usher in a new era of web development based on encapsulated and interoperable custom elements that extend HTML itself."

*Built a top these new standards, Polymer makes it easier and faster to create anything from a button to a complete application across desktop, mobile, and beyond."*

- Bosonic (<http://bosonic.github.io/index.html>): "We loved the Introduction to Web Components spec. What we wanted was a way to build components as this spec describes, shielding ourselves against potential spec changes, and supporting not-so-modern browsers like IE9. And so Bosonic was born!"

All the three libraries share and collaborate upon these polyfills building on them in unique way.

*"X-Tag and Polymer are both high-level sugar libraries that build upon the W3 Web Components specs - each introduces a different approach to making development of web components an even more amazing experience. To help make this more relatable, consider the following: jQuery: DOM :: X-Tag/Polymer : Web Components."*

The idea behind web components usage is that we will build different widgets such as:

- **Visualization Components:** these components will be the charts that users will be able to create from datasets made available from public administration as open-data. Anyway the users could use these widgets to visualize any datasets can support discussions in the SPOD platform.
- **Control Components:** these components will be a set of control widgets such as containers, customized menu, treeview or combination of customized classic web control widget, which implement complex user interactions and advanced behaviours.

### 3.3.3.2 WEB APPLICATION FRAMEWORK

Another possibility is to use some JavaScript Web Application Framework instead Web Components.

A **Web Framework (WF)** or **Web Application Framework (WAF)** is a software framework that is designed to support the development of web applications including web services, web resources and web APIs.

The concept behind this approach is very different from the previous one because the Web Framework makes available a set of high-level API for things like services, routing, server communication and similar tasks.

Conversely, Web Components approach focuses on allowing you to create rich, powerful, reusable web components, which could be used to build webapps like those built with Web framework. In the future, the lines could be blurred further as Web Frameworks may leverage Web Components.

We consider the two most important Web Application Frameworks based on JavaScript:

- AngularJs (<https://angularjs.org/>): it provides a framework for client-side model-view-controller and model-view-viewmodel architecture. The AngularJS library works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. It is maintained by Google and community.
- React (<https://facebook.github.io/react/>): it is an open source library maintained by Facebook, Instagram and community of individual developers. It was built to solve the problem of building large applications with data that changes. With react is possible to crate components (React components) that make the code reusable by providing a view for data rendered as HTML.

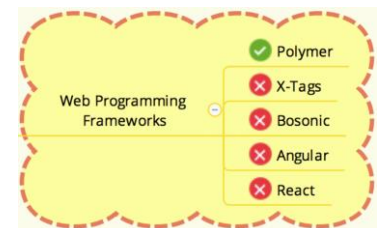
### 3.3.3.3 A COMPARISON AND OUR CHOICE: POLYMER

A comparison of all features among X-Tags, Polymer, Bosonic, AngularJS [36] and React (<https://facebook.github.io/react/>) is displayed in the following table:



Feature	X-Tags	Polymer	Bosonic	Angular	React
Template	✗	✓	✓	✓	✓
Web Components	✓	✓	✓	✗	✗
Material Components	✗	✓	✗	Limited	✓
Data Binding	✗	✓	✓	✓	✓
Filters	✗	✓	✗	✓	✗
Events Handling	✗	✓	Limited	✓	✓
Touch and Gestures	✗	✓	✗	✗	✗
Routing	✗	Limited	✗	✓	✓
Ajax/REST	AJAX	AJAX	AJAX	REST	AJAX

Besides the considerations described in the table above, also the consideration that Polymer represents also a de-facto reference implementation for the web components and that is supported by Google, clearly suggested the choice of adopting Polymer.

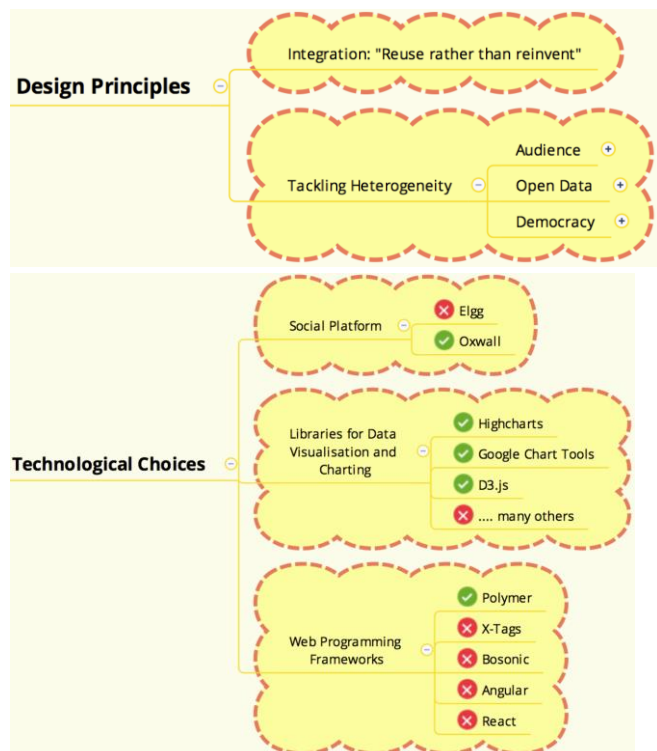


### 3.4 CONCLUSIONS

The main purpose in the design of SPOD has been to have a social platform that would allow social interactions between users, citizens, facilitating discussion and understanding around the open data; but at the same time, the architecture of SPOD had to be modular and extensible to allow easily always add new features to meet the needs of kind of user (i.e., citizen, public administration etc... ).

The principles and the choices for our platform have been described in this section. We presented our two design principles and, then, we presented a detailed description of the technological choices for the building bricks of our architecture.

As a result we selected Oxwall as open-source social platforms, HighCharts, Google Chart Tools and D3.js as data visualization libraries, and chose Polymer (and web components) as web programming frameworks





## 4 ARCHITECTURE DESIGN OF SPOD

---

### 4.1 THE OVERALL ARCHITECTURE

The SPOD platform architecture has multiple decoupled and modular components that communicate together. The architecture is based on mainstream, open source and modular technologies to guarantee interoperability with other external systems. The overall architecture is distributed, as the load of different tasks is taken by different servers/components, both server-side and client-side (e.g., the client-side visualization of data), thereby achieving the important non-functional requirement of Performance Efficiency (see Sec. 2.2.1.1).

SPOD is a Social Platform for Open data, so its primary requirement is the retrieval of data from Open data Providers. Therefore, SPOD interoperates with TET, any CKAN based platform, UltraClarity and OpenDataSoft. In addition, SPOD can retrieve open data from other existing third party data providers that use restful API. Hence, the interoperability with data provider platforms has based on Web 2.0 mainstream technologies (fulfilling the non-functional requirement of Interoperability, see Sec. 2.2.1.2); in this way, SPOD can retrieve the open data to use within the social discussions. For instance, the user can create visualizations from the data available in the open data provider and use them to support its argumentation. SPOD has a set of associated data providers and it is extensible to easily add and integrate additional data providers. The platform administrator, using the administration pages, can add another data provider and make it available to end-users. In addition, in order to maximize the flexibility, during the creation of a data visualization, any user can copy and paste the data URL from any other external open data Provider as well as post directly the link or content on SPOD.

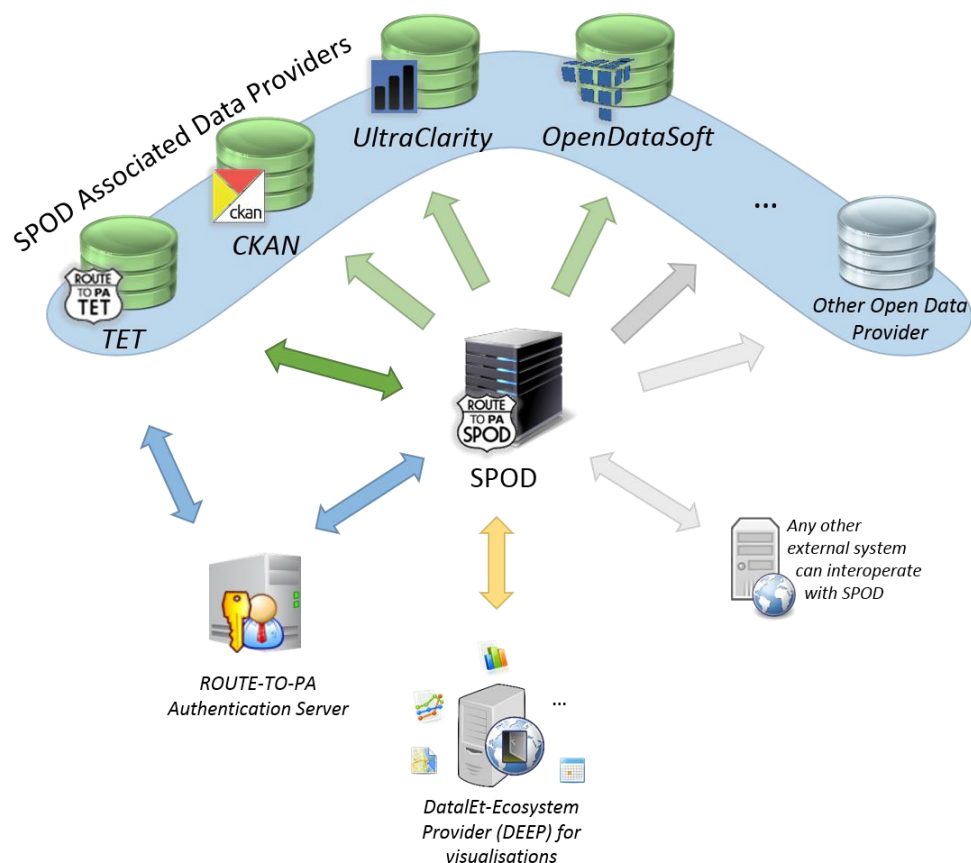


Figure 4: Overall architecture

The architecture has a ROUTE-TO-PA Authentication Server (RAS), which acts as OpenID Authentication Provider and administration tool to manage users' accounts. In accessing to the ROUTE-TO-PA platforms, users must seamlessly switch between SPOD and TET features in a user-friendly way. Therefore, a user must access to SPOD and TET, and any other federated system, with a unique username and password credential. In order to provide this, the architecture has a ROUTE-TO-PA Authentication Server (RAS) based on the OpenID protocol. Any time a user logs in SPOD or TET, his/her browser redirects to the authentication provider log in page. All the authentication server pages have a consistent Graphical User Interface (GUI) with SPOD and TET, indeed they have been specifically designed within the ROUTE-TO-PA project and it is based and compliant with the material design. In this way, TET, SPOD and authentication look and feel is the same, and the switching among their pages is seamless. The overall architecture has a specific server for the authentication and the entire platform has deployed with a dedicated authentication server. In order to support the authentication through OpenID, SPOD team developer designed a new Oxwall plug-in that supports OpenID. SPOD can be deployed without the activation of the OpenID plug-in so it works without a dedicated authentication server and uses the existing Oxwall registration, log in pages and user accounts management.

The introduction of a ROUTE-TO-PA Authentication Server allows the interoperability of other systems with the ROUTE-TO-PA platforms, following the non-functional requirement of Interoperability (see Sec. 2.2.1.2). Both SPOD and TET can provide specific services and data to external systems in form of Restful API. Therefore, not only SPOD interoperates with existing data providers, but also itself can provide services to other platforms. Any other external system can authenticate to RAS and interact with SPOD through the restful API services. For instance, based on this architecture, the platform can provide a social widget to embed within any web site to easily share open data, add the content in the own private room or participate in a discussion. In a federated

architecture, after the authentication the federated system can invoke a restful API service to perform an action on SPOD (e.g., post of content on SPOD, etc.).

SPOD enables the social collaboration around open data; in particular, it aims to support the collaboration around visualization of open data, allowing their creation, sharing, change and comment. The SPOD architecture provides the visualizations and their services through the DataEt-Ecosystem Provider (briefly DEEP). DEEP is a repository of visualization web-components to use within SPOD and within any other web site or system that needs to visualise data. DEEP component is the point in the architecture to provide statistics on the visualizations within the ROUTE-TO-PA platform.

The overall architecture is also useful to envision a “federation” of ROUTE-TO-PA systems. Several “federated” SPODs and TETs or other institutional systems can access to the ROUTE-TO-PA Authentication Server and share the same ID.

The SPOD architecture is modular and scalable (see the non-functional requirements of “Modularity” in Sec. 2.2.1.9 and “Performance Efficiency” Sec. 2.2.1.1). For instance, the DEEP component can be replicated and distribute around the world to improve performances and serve visualizations to end-users with high availability. Of course, multiple architecture instances can be deployed in different places.

Our distributed architecture, in general, follows the non-functional requirement of “Replaceability” (see Sec. 2.2.1.13), as each single component can be substituted with another specific one with the same interface (like a different dataset provider, or another Authentication server with OpenID).

## 4.2 SPOD ON OXWALL

Oxwall is a free and Open Source Software (FOSS) that is able to power customizable online social networks and community-enabled websites. It provides all the basic functionalities of a social network, such as users’ friendship, posting text or media comments, handling (private) groups group or event creation/joining.

Oxwall architecture is based on plugins, few core plugins handle user/platform/access management. Additional features can be provided by plugins.

In the architecture its role is to be the Social Platform entry point. Base features and additional SPOD functionalities (like Public rooms, Personal Space, etc.) are provided by several plugins that are described in the System Design Section 5.1. Among the SPOD plug-ins, the only one that is relevant to the architecture is the OpenID-Connector plug-in, which supports the authentication through the OpenID protocol. The SPOD OpenID-Connector plug-in connects to the ROUTE-TO-PA Authentication Server (RAS) that implements an OpenID provider.

## 4.3 DATALET-ECOSYSTEM PROVIDER (DEEP)

The ROUTE-TO-PA software architecture is based on system integration process. The system exploits a modular programming design in order to develop independent software. To achieve this design goal, an architecture completely decoupled from the main project SPOD was designed. The key idea was to realize a repository of components (software services) to be used for different purposes, so that it is possible to enclose some functionalities in a kind of widget and make it available on the Web. This architecture is realized using the Web-Component (WC) standard. We designed a Web service that allows distributing the code of each software component dynamically.

Therefore, each software component is a WC that is, an auto-consistent and independent component that provides some functionalities. WC functionality are grouped into two main categories: WC controllet that is a view controller generating an output view and an embedded controller, and WC datalet that is an output

presentation to the user based on the changes in the model (where the model is a data source). The service that allows downloading and using the WCs is the **DataEt-Ecosystem Provider (DEEP)**. The DEEP service allows the programmer to write simple code in order to insert at loading time a WC in a standard Web page. This architecture is also useful for generating metrics and statistics on the usage of the WC.

Within the SPOD software, DEEP architecture is an open, extensible, modular and pluggable service that provides WCs for visualization of open data datasets. DEEP allows the sharing, collaboration and creating around customized data visualizations. Further users can create, reuse and share visualizations both in SPOD that in any Web page or other systems. Its modularity and extensibility fulfils the non-functional requirements of “Adaptability” (see Sec. 2.2.1.12) and “Replaceability” (see Sec. 2.2.1.13).

It must be noticed that DEEP can provide any kind of visualization in a web page, without any limitation to open data Datasets. Therefore it can be flexibly used in other context, thereby addressing the non-functional requirement of “Flexibility” (see Sec. 2.2.1.14). In this section, we describe the architectural aspects of DEEP, while the detailed description of DEEP components will be given in the next section.

#### 4.3.1 ARCHITECTURE

DEEP is developed as a simple Restful service, providing the list of available datalets (i.e., listing service) and the mapping among the visualization names and their relevant URL within the WC repositories. The system is online and is the base of the architecture of SPOD (<http://deep.routetopa.eu/>).

Both the DEEP and the WC repository have been designed to be extensible: they can collect all the visualization requests so, as planned future work, they could also provide aggregated statistics on both users preferences and on data and their visualizations. For instance, the most popular datalet visualizations, most used datasets, most popular visualizations for a particular dataset, most visualised fields for a particular dataset, and so on.

The DEEP main task are the listing services which provides a list of available dataset and the mapping between the visualization names and their relevant URL within the “datalets repositories” (see Figure 5).

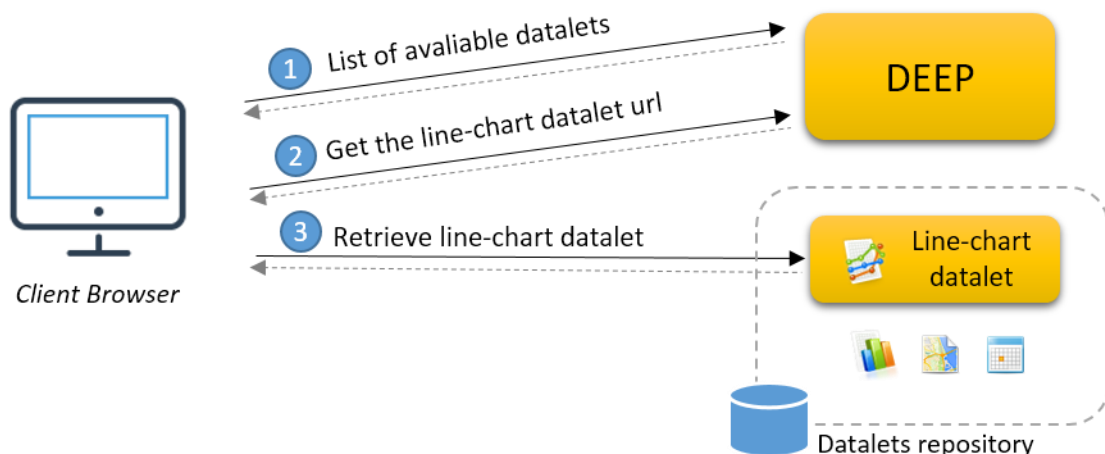


Figure 5: DEEP and datalets repository architecture: a list of available datalets provided by DEEP and known name of datalet (e.g., Line-chart datalet), DEEP provides its implementation loading from the “datalets” repository.

#### 4.3.2 DEEP WORK CYCLE

The DEEP work cycle (see Figure 6) to include a WC in a Web page, comprises three main actors:

- The Client page that exploits DEEP-Client functionality;
- The DEEP (**DataEt-Ecosystem Provider**);
- The Web-Component.

First, (1) the Client page send a request to DEEP for a specific WC. Then, (2) the DEEP responds with the information needed to inject the datalet into the page. Finally, (3) the Client retrieves the WC from the DEEP repository and include it into the page.

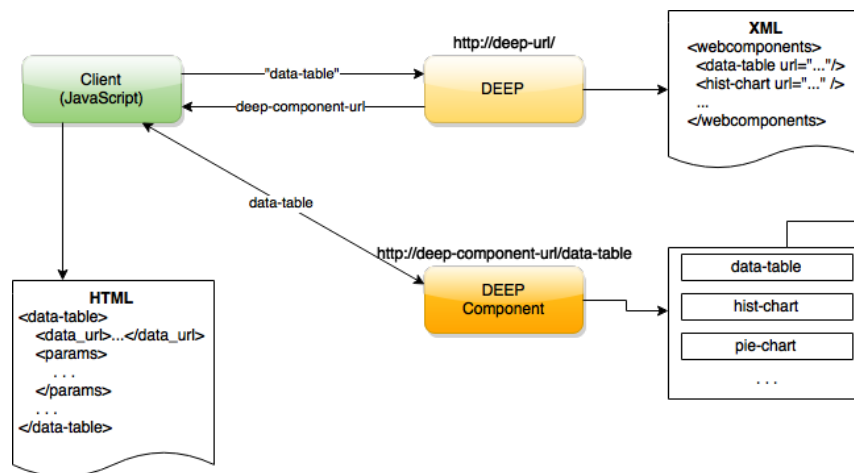


Figure 6: DEEP workcycle (DEEP Component should be Web-Component).

#### 4.3.3 DEEPCIENT: AN EXAMPLE OF USE

In order to better describe the DEEP workflow in the following is described the software interactions for the inclusion of a WC in a standard Web page:

1. An HTML Web page that must includes WCs: need a Javascript library named DEEPCIENT (configured with the URL of the DEEP repository) that exploits the client side functionalities.
2. The previous figure (see Figure 6) depicts all the interactions between the client and the DEEP. The Web page contains a data-table WC (a table visualization of some data). The WC takes as arguments: the data\_url (the URL of a data-source) and some parameters (depending on the particular WC).
3. The DEEP CLIENT sends a request to the DEEP asking for the location of the code for the data-table WC (represented by an URL).
4. The DEEP answers with the URL where the client can download the code for the data-table WC.
5. At this point the DEEPCIENT requests the data-table WC code from the repository (basically a web server compatible with HTTP/Ajax that serves WC HTML code).
6. Once downloaded, the data-table WC code is placed in the HTML client page and rendered within the browser.

The code to include a WC in a web page follows:

```

<html>
<head>
<script type="text/javascript" src="js/jquery-1.11.2.min.js"></script>
<script type="text/javascript" src="js/webcomponents.js"></script>
<script type="text/javascript" src="js/deepClient.js"></script>
<script type="text/javascript">
  jQuery(document).ready(function($) {
    var datalet_params =
    {
      component : "DATALET_NAME",
      params : {
        data-url : "DATA_URL",

```

```

        layout-param-1 : "LAYOUT-VALUE",
    }
    fields : Array("FIELD1", "FIELD2"),
    placeholder : "HTML_PLACEHOLDER"
};

    ComponentService.deep_url = 'DEEP_URL';
    ComponentService.getComponent(datalet_params);

});

</script>
</head>
<body>
    <div id="HTML_PLACEHOLDER"></div>
</body>
</html>

```

In details, in order to include a WC in Web page the programmer has to:

- create a placeholder DIV where the WC code will be injected.
- Configure the WC request
- call the `getComponent` method on the DEEPCIENT to download and inject the code. The parameters needed to inject the WC in this example are:
  - component: datalet name (Eg. datatable-datalet)
  - params.data-url: `datasource` `URL` (Eg. [http://demo.ckan.org/api/action/datastore\\_search?resource\\_id=8324a4c9-66d6-47bf-a898-94237cc39b9f&limit=50](http://demo.ckan.org/api/action/datastore_search?resource_id=8324a4c9-66d6-47bf-a898-94237cc39b9f&limit=50));
  - params.layout-params: parameters of the WC (Eg., chart title, axis labels);
  - fields: an array with user selected fields from datasource (Eg., `Array("Supplier","Amount")`);
  - placeholder: the ID of the DOM element where the Datalet has to be placed.

`ComponentService.getComponent` function takes a `datalet params` object as parameter. This function is responsible for datalet info retrieved from DEEP or datalet/controllet retrieved from the DEEP repository. Once the datalet/controllet code is available, the DEEPCIENT adds the datalet to the web page and initializes the datalet with the desired parameters. The datalet/controllet will use only the data available in the `fields` parameter.

#### 4.3.4 WEB COMPONENTS IN DEEP

We have designed several WCs (datalets and controllets) according to the Object-Oriented design paradigm in order to achieve the maximum reuse of the code and modularity. Our WCs have been developed using the Google library Polymer. Unfortunately, Polymer 1.0 does not support components inheritance. We devised a simple mechanism, which allows to realize inheritance like scenario: A bundle of objects, named `base-datalet`, that implements basic behaviors has been implemented and is available to be reused in other WCs.

##### 4.3.4.1 CONTROLLET

A controllet is a view controller WC that implements control behaviour like form, slider, etc. It generally uses a set of Polymer core elements realizing a more complex component using composition paradigm. This kind of WC is used to design and develop SPOD functionalities.

##### 4.3.4.2 DATALET

A datalet is a view WC, which is used to create rich, reusable visualization of open data. All the processing is performed client-side by the user browser, so that ensures scalability in terms of number of visualization

(compared to others libraries as Google Chart). This is an architectural choice following the Performance Efficiency non-functional requirement (see Sec. 2.2.1.1). The datalet can query data providers provided that they return data in JSON or CSV format. All the datalets inherit from a base web-component called base-datalet. The datalet runs on the client-side within the user browser, thereby ensuring scalability on server-side: no additional heavy load is placed on the server, besides unloading few additional lines of JavaScript. In order to visualise a dataset, the following steps need to be performed. The user types the URL of a Web page which contains a datalet. The browser during the page content loading, takes the datalet URL and connects to the “datalets repository”. Then it loads the datalet source code to be executed (different visualization are rendered using different datalets). The datalet takes in input any external dataset URL and a query to recover the data. An optional filter can be applied to further refine data. Finally, the datalet renders an interactive visualization by using the filtered data.

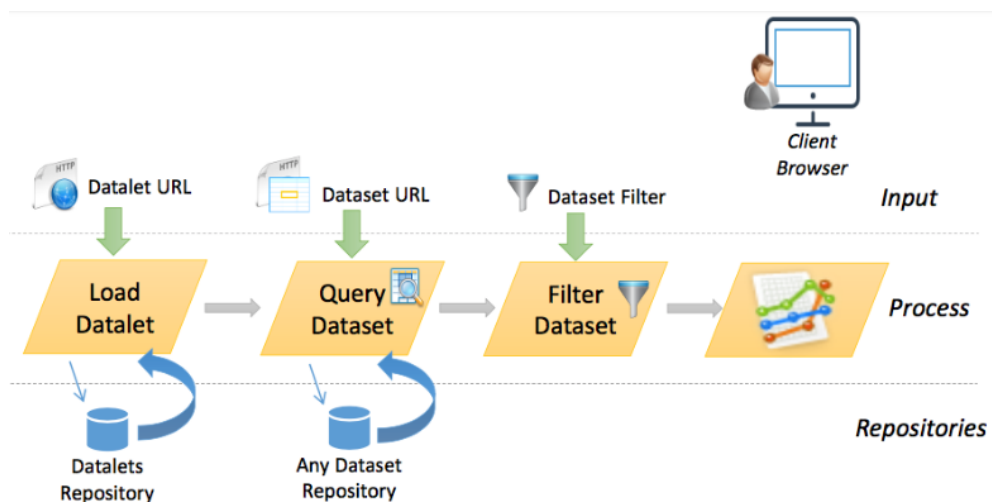


Figure 7: Processing, querying and visualization dataset using a datalet.

Datalets guarantee the provenance of data by (1) loading the dataset from the external source without changing the data, (2) providing the URL of the data source so that the user can check the visualised data on the original source, (3) introducing a certified repository of datalets.

Datalets have been designed to process any dataset as input. *In the current implementation, datalets can visualise data from CKAN installations, and from commercial providers, such as OpenDataSoft<sup>2</sup>, and Enigma<sup>3</sup>.*

The base datalet are divided in two categories: BaseDataletsBehaviors and CustomDataletBehaviors. The BaseDataletBehaviors define the lifecycle for all DATATLETs. Every lifecycle step in BaseDataletBehaviors is implemented in the CustomDataletBehaviors. This mechanism provides a components' hierarchy that standardize datalet production and make reusable common behaviors.

The BaseDataletBehaviors are:

- BaseDataletBehavior;
- WorkcycleBehavior.

The BaseDataletBehavior defines the mandatory attributes that all datalet must have:

- data\_url (String): the URL used to get the data from the open data Provider (e.g. the CKAN API).

<sup>2</sup> <https://www.opendatasoft.com/>

<sup>3</sup> <http://enigma.io/>

- fields (String): it represents a JSON array of user selected dataset fields.
- data (array): data structure that store the data retrieved from open data Provider.

The WorkcycleBehavior implements the work-cycle of each datalets, composed by the following steps:

- Get the data from a source: It is responsible for data retrieving from an open data Platform.
- Select a subset of fields from the data: this operation is data format dependent. It is responsible for extract a query related subset of information from the entire dataset. A multidimensional array will be made available for the transformation step.
- Filter the data by applying one of the following operators (<, >, =, <=, >=, etc.).
- Transform data in order to obtain a coherent data representation.
- Present the data for a visualization.

#### 4.3.5 DATALETS IN SPOD

In SPOD platform, everyone can, without no technical expertise, create and share visualization of a chosen dataset with a controllet, with the following simply steps:

- Upload open data dataset from recommended datasets by SPOD, or a search suggested by a treemap, or external (remote) dataset via API;
- Select the field of the chosen dataset;
- Select from a variety of visualizations with customized chart (title of the chart, etc.);
- Share your visualization over the SPOD.

The datalet life-cycle in SPOD can be described in the following way:

- 1) SPOD provides a page with a datalet embedded (with all the parameters in the URL);
- 2) The browser gets the datalet from the Dealer DEEP;
- 3) The datalet gets the dataset from the provider and selects the data;
- 4) Visualization is shown in the page.



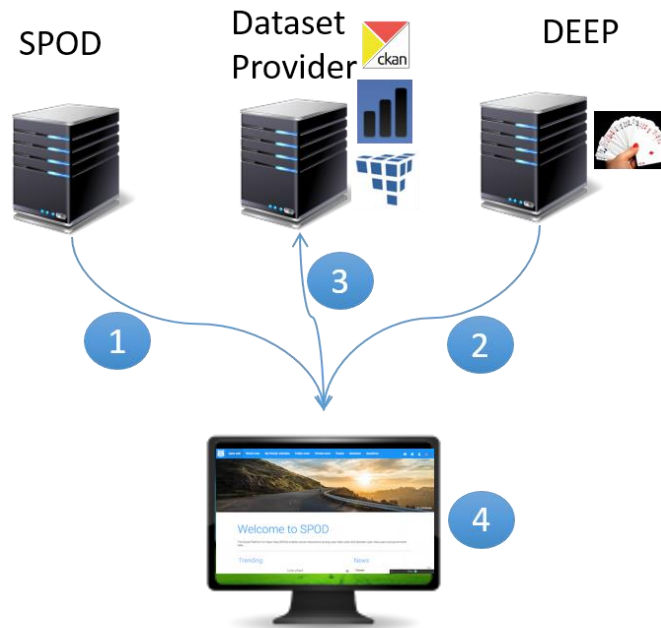







Figure 8: A life-cycle of a Datalet in SPOD





#### 4.3.6 INTEROPERABILITY LIST WITH DATASET PROVIDERS

During the first year, we have checked the interoperability with the following well-known, external dataset providers for the countries of the project. This is an initial proof that our architecture is working toward non-functional requirement of “Interoperability” (see Sec. 2.2.1.2).

	data.jsy.com	data.iledefrance.fr	data.gov.uk	data.gov.ie	dati.lazio.it	Vmdatagov01.der1.le:8080	data.overheid.nl	datiopen.istat.it	www.europeandataportal.eu	Enigma.io	open-data.europa.eu	Ultraclearity.yperdiangeia.gr
Platform	OpenDataSoft	OpenDataSoft	Harvest	Harvest	CKAN	CKAN	CKAN	SPARQL	SPARQL	User need to login	Harvest/CKAN	CUSTOM
API to Get dataset by JSON URL			-	-							-	
API to Search by keyword on the Portal			?	?	?	?	-				?	
API to get the list of available open datasets (to be used with treemap)												?

Get Dataset page from its id (to be used for datalet link)		?	-	-			-				-	?
--	---	---	---	---	---	---	---	---	---	--	---	---

**Legend:**

	Enabled	Disabled
API		
SPOD uses the portal provided API		

## 4.4 THE AUTHENTICATION SERVER FOR JOINT AUTHENTICATION

The ROUTE-TO-PA platform aims to deliver SPOD and TET systems, each one with its own goals and features. From Graphical User Interface (GUI) point of view, the requirement is to have a seamlessly switching between the two systems in an unnoticed way. The term “seamless” means that the users perceive the two systems as a whole. Of course, the first step is to have the same coherent look and feel based on the material design. In particular, SPOD and TET provide each one their features, and the seamless switching between them occurs when a user action on one system that must be forwarded to the other one. So in order to have a seamless switching is fundamental to provide one shared authentication mechanism. These requirements have emerged in the user scenarios (Deliverable 2.2, s28.1), requirements and use cases (Deliverable 2.4, UC23-UC8) but also from the original Description of Work.

The joint authentication mechanism is also useful to envision a “federation” of ROUTE-TO-PA systems, where access to several “federated” SPODs and TETs of collaborating PAs or other institutional or private actors can share the same ID. An immediate example could be a hierarchy of PAs, from regional, to province, to local level, that all offer to their citizen an integrated platform, with a single sign on.

This section introduces the main concepts of authentication and the designed architecture of the “ROUTE-TO-PA Authentication Server” (RAS).

### 4.4.1 CONCEPTS

Authentication is the process of establishing authenticity of something or someone as a valid entity. The common authentication method is to use username and a password that has used in SPOD/TET.

Usually independent systems have their own authentication mechanism, forcing the users to have different username/password for each system. SPOD and TET systems have designed to use a third authentication server based on the OpenID. OpenID<sup>4</sup> is an open and standard protocol, which allows users to be authenticated by certain co-operating sites using a third party service. In this way, a user registers one time and has one username/password to log in into multiple systems. Therefore, the user logs in one system with a single ID and password to have access to all independent systems using the same login/password. From user point of view, he/she has only one pair of username/password and can log in multiple systems. In particular, once the user has logged, he/she can use any of the systems without log in again. The opposite operation is the single sign-off whereby a single sign-out terminates the access to multiple software systems.

<sup>4</sup> OpenID protocol web-site <http://openid.net/>

#### 4.4.2 THE ARCHITECTURE

The architecture has the following components:

- User Agent or Web Browser where the log-in process starts;
- The OpenID Server Identity Provider;
- The Consumers SPOD and TET use the OpenID provider services to authenticate the end-user.

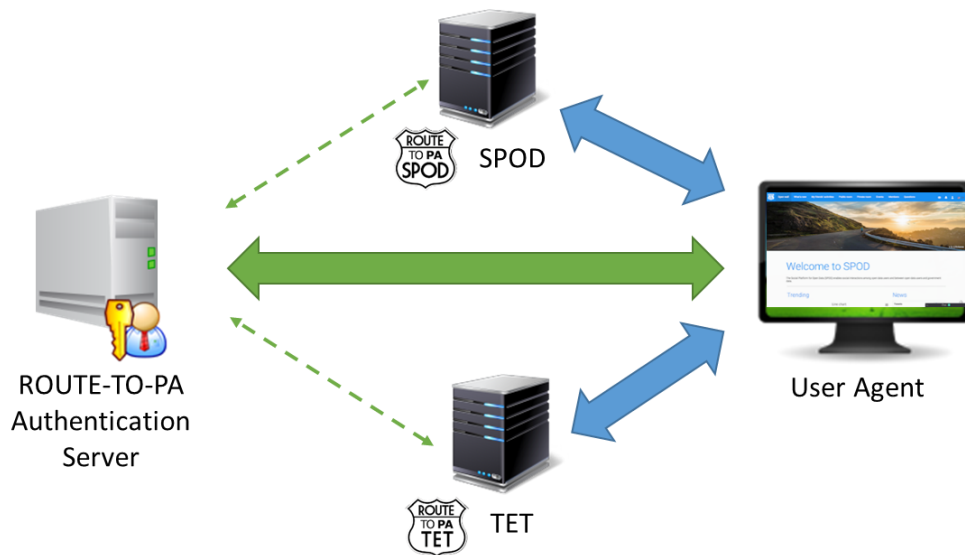
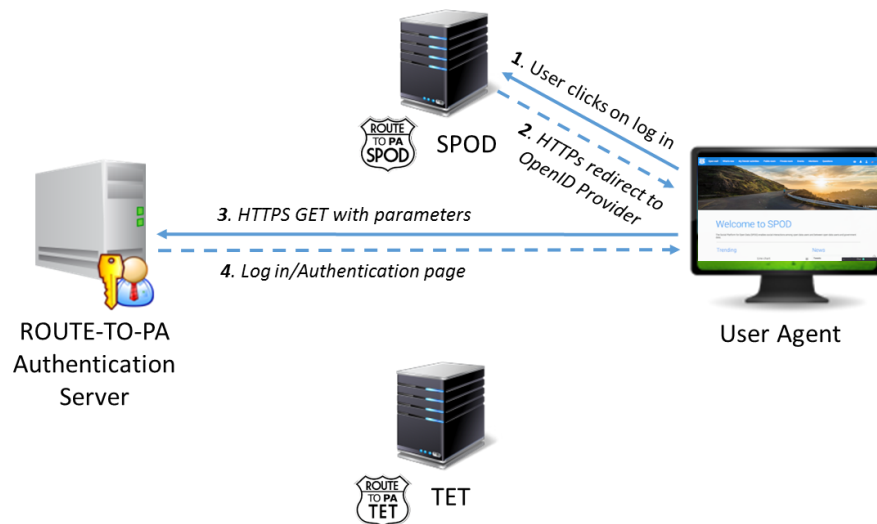


Figure 9 - ROUTE-TO-PA Platform Authentication Architecture.

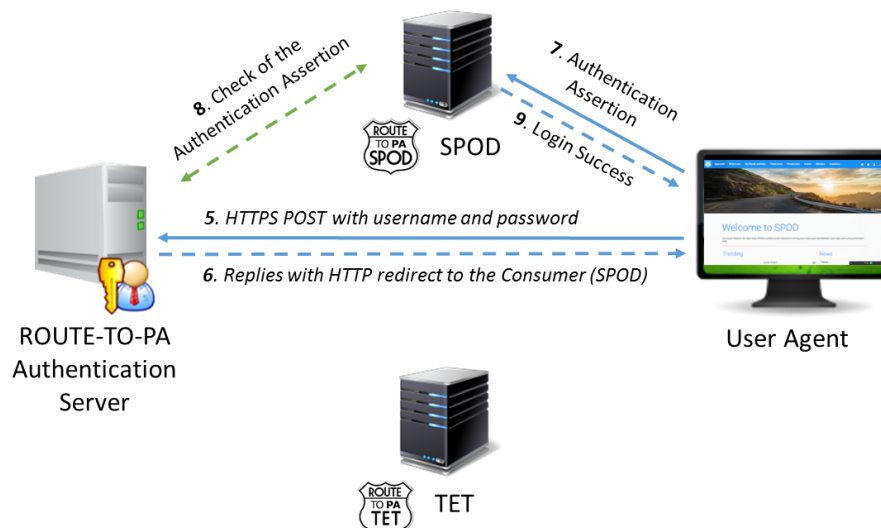
The architecture uses an Indirect Communication between the OpenID provider and the Consumer (SPOD/TET). All communications between SPOD and the OpenID provider are indirect because they pass through the User Agent using HTTP redirects.

The process has the following steps:

1. End-user visits the SPOD or TET home page. The user always starts the authentication process, trying to log in into the TET or SPOD (Consumer Web site) by clicking on the “log in” link.
2. SPOD or TET server receives the log in request; it knows the OpenID Provider URL, so SPOD/TET replies to the client with an HTTP redirect (302) indicating the OpenID provider URL and other parameters to useful for the authentication. As additional parameters, this redirect contains the URL “return\_to” which indicates the SPOD web page to contact when the log in will success (the redirect path to the Consumer).
3. The User Agent performs the redirect sending an HTTP GET Request to the OpenID provider
4. Assuming that the user has not already logged in, the OpenID provider replies with the log in page where the user can type his/her username and password. This part may be skipped if the user has already logged in.



5. The user types his/her username and password in the authentication page and clicks on submit sending these information to the OpenID provider
6. The Identity provider will return the assertion information with its signature to the Consumer via browser redirect.
7. The SPOD Server (Consumer) receives the authentication assertion
8. The SPOD Server (Consumer) establishes a direct connection with the OpenID provider. SPOD will request the authentication information directly from the OpenID Provider and compare it with the assertion information it received via User Agent (Browser). This is to double check the validity of the assertion in case a User Agent (or malicious attacker) is trying to cheat.
9. The end-user has successfully logged in SPOD



At this step, the end-user has successfully logged into the SPOD system. Once logged in one system, the architecture has designed in a way that when the user goes to the other one, he/she has already logged. In this way, once logged, for the user the switching between SPOD and TET is seamless. When the user has logged in one system through the OpenID provider and accesses to the other one, it follow the same steps in of the previous

process to authenticate, but at step 4, the user results to be already logged, so the provider does not show the authentication page but replies directly with the authentication assertion.

#### 4.4.3 IMPLEMENTATION DETAILS

SPOD is based on Oxwall. Actually, in its last version, Oxwall does not have a working plug-in to log in with the OpenID protocol. Therefore, as system development we designed an “OpenID Connector” plug-in to manage the authentication through OpenID, (see section 5.1). The plug-in implementation uses a PHP library specifically designed to provide OpenID in web-applications called *php-openid*. The *php-openid*<sup>5</sup> is open source and released with Apache license. The *php-openid* library has used also by other systems, for example the Wordpress OpenID implementation uses this library. When the user clicks on the log in link in SPOD, the authentication plug-in builds the request to the OpenID server provide and in according to the protocol, it redirects the browser to the OpenID server authentication page. When the user successfully logs into the system, the plug-in checks whether it is the first time that the user enters in the system, if it is the first time, it creates a SPOD profile for the user.

RAS is based on the same *php-openid* library, which provides the server implementation compliant with OpenID protocol. RAS also provides facilities for user registration, authentication, and administration dashboard for user accounts management, and finally invitation by e-mail. The switching between TET, SPOD and the Authentication Server pages must be seamless unnoticed by end-users. Therefore, the look and feel must be the same based on the same design language (i.e. Material design, described in Section 4.5). The OpenID provider offers the following features to the users: Authentication, password recovery, and log out. The platform administrators have also access to these features: user creation and deletion, password change (documented in the Manual).

### 4.5 DESIGN LANGUAGE

Google Material Design inspires our project design language. The main idea is develop a single underlying system that allows a unified experience across platforms devices and sizes. It is a very well known Design Language, whose choice is meant to accommodate the non-functional requirements of “Usability” (see Sec. 2.2.1.3) (because of its very simple and effective style) and “Familiarity” (see Sec. 2.2.2.2) since it is a common graphical interface to the widely diffused Google applications, both on desktop and mobile.

#### 4.5.1 MATERIAL DESIGN

Material design is inspired by typography, grids, space, scale, color, and imagery in order to create hierarchy, meaning and focus, the goal is create a simple but powerful user experience. It is a coherent visual language whose choice from us is aiming at addressing the non-formal requirement of “User Interface Aesthetics” (see Sec. 2.2.1.6). Material design encourages consistency across environments by repeating visual elements, structural grids and spacing across platforms and screen sizes. These layouts scale to fit any screen size, which simplifies the process of creating scalable apps. The main goal of material design is create a visual language that synthesizes classic principles of good design, with the innovation and possibility of technology and science.

Material Design metaphor is a 3D space where all object has three dimension and represent a material object, this is a unifying theory of a rationalized space and a system of motion.

Every object that form the UI has physical properties in material space. Multiple material element cannot occupy the same space simultaneously, neither an object can permeate other objects. Every object can vary in width and height but all objects has uniform thickness. Every object casts shadow that reflect relative elevation between material elements. In Material Design, the physical properties of paper are translated to the screen. The

---

<sup>5</sup> The *php-openid* official web-site and source code repository <https://github.com/openid/php-openid>

background of an application resembles the flat, opaque texture of a sheet of paper, and an application's behaviour mimics the paper's ability to be re-sized, shuffled, and bound together in multiple sheets. Elements outside of applications, such as system bars, are separate from the app content beneath them, and are not treated as material.

#### 4.5.1.1 MATERIAL DESIGN ACCESSIBILITY

A software product is accessible when all people can navigate it, understand it, and use it correctly and successfully. This guideline has inspired SPOD design, and we trying to make it accessible to the widest possible audience, including users with some kind of impairments (see the non-formal requirement of "Accessibility" in Sec. 2.2.1.7). All the technological choices are accessibility aware. The material Design guidelines application allows creating accessible portal and application through a simple and friendly UI. Moreover Google applies Material Design to all its products, so a SPOD user will find a familiar UI (see the non-functional requirement of "Familiarity" in Sec. 2.2.2.2).

We introduced explanatory tooltips on all the developed tools. This tooltips will allow the use of the platform without the manual and will help to understand all the functionality developed.

Color blindness is a common condition, this condition affects the 8% of male population and 0,4% of female population worldwide. For increasing the accessibility we have chosen a colour palette that is visible for protanopia affected people. Moreover Highchart library associates to all data series inside a chart as well as a colour also a shape in order to increase legibility and accessibility of the charts for users affected by color blindness. [37]

#### 4.5.1.2 SPOD FOR MOBILE

All the technological choices we made are fully compatible with mobile devices. Is possible to visit SPOD platform with modern tablet devices. The platform is fully interoperable with mobile devices, only the datalet creation on old tablets is not supported at the moment due to UI screen size limitation. In general, this is a positive step in the direction of ensuring the "Universality of access" non-functional requirement (see Sec. 2.2.2.1).

#### 4.5.2 SCREEN RESOLUTION AND BROWSER

SPOD is developed and tested on several platforms, we have selected the top three popular browsers according to StatCounter Global Stat (Google Chrome, Firefox and Safari) plus Microsoft Internet Edge. We have developed a UI that fits over the 80% of screen resolution, paying particular attention to the top ten screen resolutions (StatCounter Global Stat).

The reason why we do not include Microsoft Internet Explorer is that Microsoft is betting on Internet Edge as publicly disclosed, as witnessed that they have already discontinued, on January 12<sup>th</sup> 2016 the support for IE versions 8, 9 and 10, and only version 11 is currently supported. Since the project will be public available only in the next years, we decided to not include Internet Explorer among SPOD supported browsers (which will be costly and not rewarding in the end of the project), but to focus on Microsoft Edge instead, following our non-functional requirement of "Universality of access" (as in Sec. 2.2.2.1.).

A table displays screen resolution display statistics<sup>6</sup>

---

<sup>6</sup> [http://stats.areppim.com/stats/stats\\_screenresxtime\\_eu.htm](http://stats.areppim.com/stats/stats_screenresxtime_eu.htm)

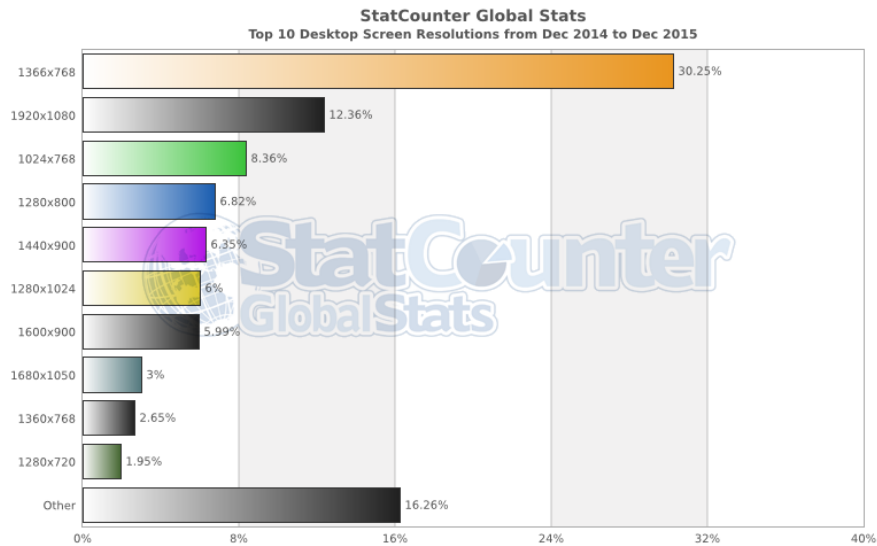


Table 6: Trends of computer screen resolution formats by StatCounter.

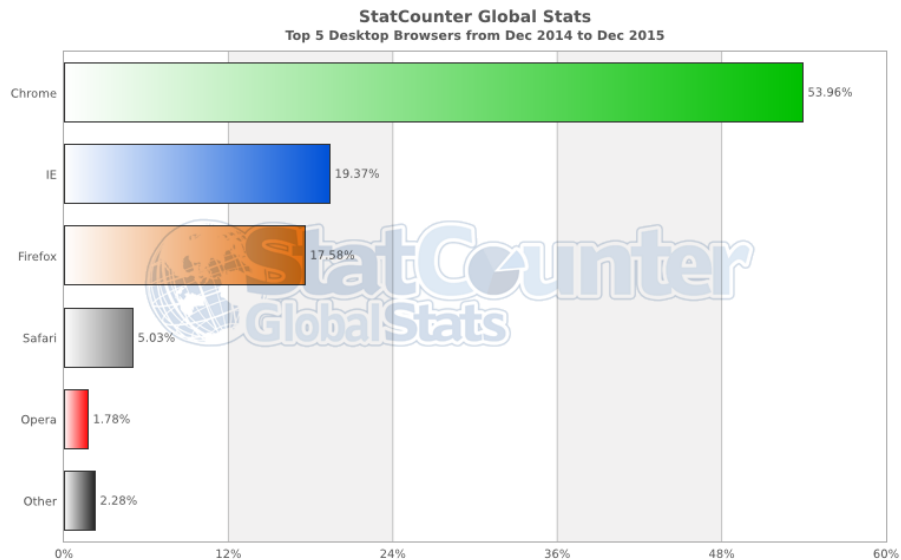


Table 7: Browser popularity.

## 5 SYSTEM DESIGN OF SPOD

The functional requirements expressed in Section 2.1 informed and motivated our actual design of the platform. In this section we report on the design of the components of the architecture shown in the previous section.

### 5.1 OXWALL AND PLUGINS

SPOD is based on the Oxwall architecture, and is customized for our purposes through the implementation of plugins and is personalized through the themes. The source code is available in three different GIT repositories:

- <http://service.routetopa.eu:7480/isislab/spod-core.git> hosts the core of SPOD. We are hosting our own version of Oxwall because SPOD is, in fact, a software fork of Oxwall, as we made minor cosmetic changes and may need in the future more significant modification for new features.
- <http://service.routetopa.eu:7480/isislab/spod-themes.git> hosts the ROUTE-TO-PA's Oxwall theme. Oxwall UI guidelines require that theme development does not involve changes in the HTML code of the pages, so an Oxwall theme is mostly a collection of images and style rules. The ROUTE-TO-PA Oxwall theme is based on Google Material Design guidelines and is called "MATTER". The theme default scheme is based on the Light Blue colour, but theme's integration with Oxwall allows the platform manager to easily change the colours via the administration dashboard without modifying the theme's source code.
- <http://service.routetopa.eu:7480/isislab/spod-plugins.git> hosts the plug-ins developed into Oxwall for the ROUTE-TO-PA project. The development of SPOD is mainly focused on the design and implementation of these plug-ins as they play the most considerable role in SPOD architecture. At present, five plugins have been developed:
  - open data Enabler (spod-plugins.git/ode) is the central part of the SPOD architecture, connects SPOD with DEEP and allows the use of datalets into SPOD (see section 5.1.1).
  - Open Wall (spod-plugins.git/openwall) welcomes the user with a landing page showing summarized information about the SPOD platform such as total and online users, datasets from associated providers, number of discussion topics, etc. (see section 5.1.2).
  - Agora (spod-plugins.git/public\_room) allows the creation of "rooms" where users can discuss a topic in a threaded chat. Moreover, they can support their argumentation attaching datalets, images, linking other websites, etc. Users can "navigate" and analyse the discussion thanks the interactive graphs: comments, users, datalets, opinions (see section 5.1.3).
  - My Space (spod-plugins.git/private\_room) is a private space (i.e. not visible to others) where a user can collect datalets, links and notes that he can use later in discussions in an Agora room, on its own newsfeed, and so on (see section 5.1.4).
  - OpenID Connector (spod-plugins.git/openidconnector) enables the user authentication through the OpenID protocol, so it provides the log in and log out functionalities. In particular, the plug-in manages the authentication connecting to the ROUTE-TO-PA Authentication Server (RAS), which is an OpenID Provider (see section 4.4.3).

In the next sections we will analyse the plug-ins in more detail.

### 5.1.1 OPEN DATA ENABLER (ODE)

The ODE plug-in connects SPOD with the Datalet-Ecosystem Provider (DEEP). In particular, it:

- Enables the integration of Datalets into Oxwall
- Allows the user to create and modify Datalets (open data Visualization) through a wizard implemented by a "Controllet"
- Takes care of persisting user-created Datalets into the SPOD database.

ODE is a fairly large plugin that cover every aspect of the use of datalets into SPOD. We will examine ODE features in more detail in the next subsections.

#### 5.1.1.1 INTEGRATION WITH OXWALL



ODE adds a button (see Figure 10. The datalet button for new posts. Figure 10) near the post creation control, so that a user can create new posts attaching a datalet.

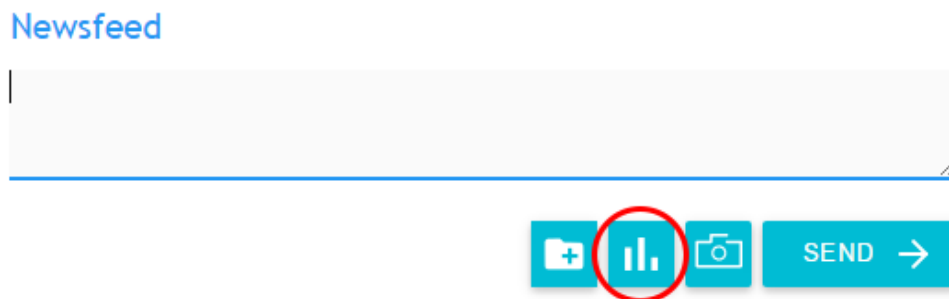


Figure 10. The datalet button for new posts.

The same feature is available when the user replies with a comment (see Figure 11)



Figure 11. The datalet button for new comments.

#### 5.1.1.2 THE DATALET WIZARD CONTROLLET

Everyone can, without technical expertise, create and share visualization of a chosen dataset with few simple steps using the wizard whose technical name is data-sevc-controllet (from now on: “Controllet”). The wizard assists the user step-by-step through the creation (or modification) of a datalet.

The first step is the dataset selection (to not to be confused with Datalet) from a compatible provider. There are three methods the user can choose a dataset:

1. It can be selected from a searchable drop-down list of datasets available from one of the associated data providers (such as in Figure 12).
2. It can be selected while exploring interactively the datasets available from associated data providers using a treemap.
3. It can be manually entered as a URL.

1

SELECT DATASET

Search or copy and paste the url of dataset.

→

LIST VIEW

TREE MAP VIEW

Available datasets

Q Dublin City Council Commencement Notices

▼ ⓘ

Selected url

[http://ckan.routetopa.eu/api/action/datastore\\_search?resource\\_id=ee00de68-f2e4-482f-a003-3c0561351075](http://ckan.routetopa.eu/api/action/datastore_search?resource_id=ee00de68-f2e4-482f-a003-3c0561351075)

Figure 12. Step 1: Dataset selection.

Once the user chose a dataset, he proceed to the second step: fields selection. The Controllet downloads the dataset in a compatible format (for instance: JSON), parses it and builds a treeview (a hierarchical list) of fields found in the dataset. The user can select the fields he wants to include into the Datalet (Figure 13) and then proceeds to the last step.

←

2

## SELECT DATA

Select the fields from tree-view. The multi-table will show the values related to the selected fields.

→

**records**

BC\_Ref

Decision

**LONG**

App\_Date

**LAT**

Y\_ITM

**Location**

Application\_Type

X\_ITM

Proposal

\_id

**result / records**

Show 10 entries
Search:

LONG	LAT	Location
53.3041801453	-6.2923431396	47, Rathdown Drive, Terenure, Dublin 6w
53.3083152771	-6.2847619057	28, Fergus Road, Terenure, Dublin 6w
53.3089771271	-6.2969498634	55, Parkmore Drive, Terenure, Dublin 6w
53.3104095459	-6.2803769112	4,5,6,7,8 & 9, The Townhouses, Terenure Road East, Terenure, Dublin 6
53.3106536865	-6.2650709152	44, Saint Kevin's Park, Dartry, Dublin 6
53.3106899261	-6.2989220619	3, Lavama Road, Terenure, Dublin 6w
53.3139419556	-6.2732594013	85, Rathgar Road, Rathgar, Dublin 6
53.3143386841	-6.2802495956	71, Brighton Square, Terenure, Dublin 6
53.3151397705	-6.2810525894	44, Brighton Square, Terenure, Dublin 6
53.3160171509	-6.2941257954	282, Kimmage Road Lower, Kimmage, Dublin 6w

Showing 1 to 10 of 100 entries

 Previous   1   2   3   4   5   ...   10   Next

Figure 13. Step 2: Fields selection.

In the final step, the user has to setup the chart. The Controllet obtains from DEEP the list of charts, also including name, type and description of inputs and settings supported by each chart. The user must select, for each input required by the chart, a corresponding field from the set s/he selected in step two. Optionally, the user can customize the chart by changing the settings (e.g., s/he can set a title and a comment for the graph, put labels on the axes, and so on). While the user makes changes to the inputs or the settings, the Controllet updates the chart preview (Figure 14). When the user is satisfied with the look of the chart, s/he clicks the “Add” button.

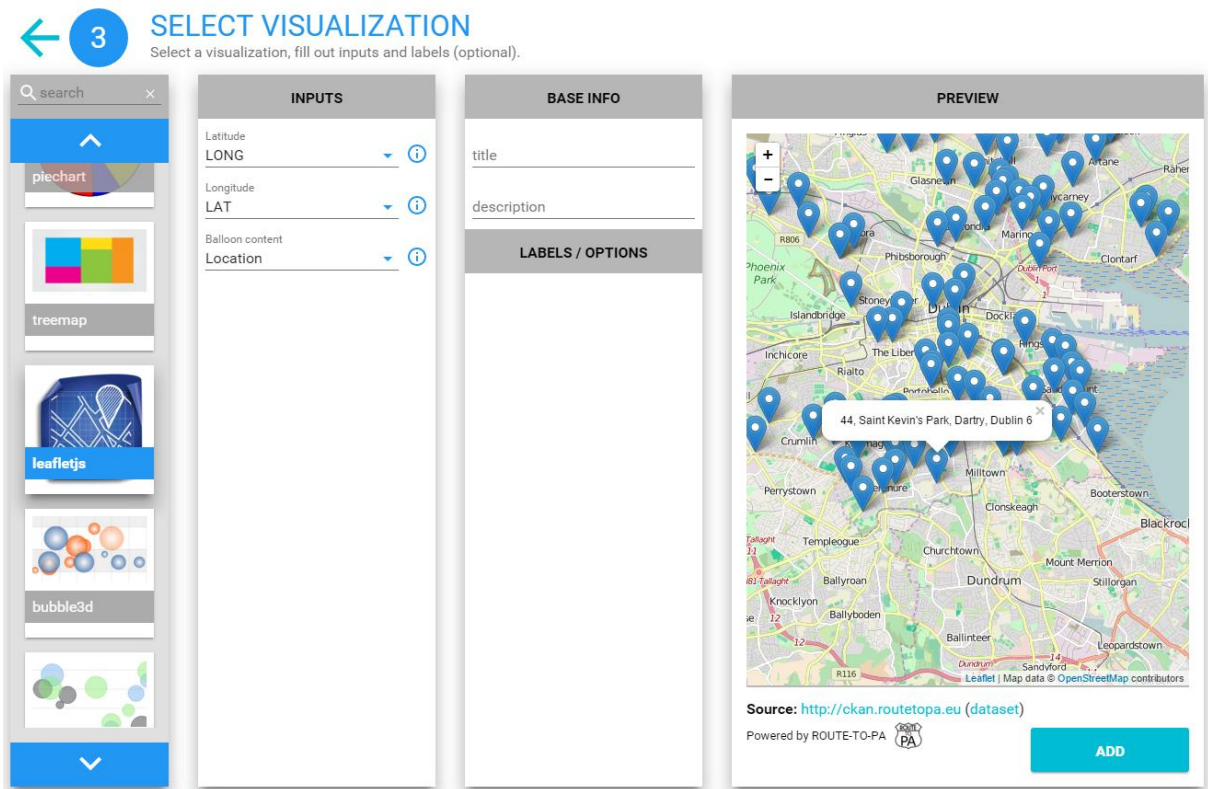


Figure 14. Step 3: Visualization definition.

At this point the ODE saves the Datalet in its database.

It is worth noticing that the Controllet is a complex Polymer web-component composed by four web-components:

- **page-slider-controllet:** a page container that allows two-way navigation emphasized by a slide-animation; each page shows an index, a title, a subtitle and a content. In our case we have 3 pages, one for each step of Datalet creation.
- **dataset-selection-controllet:** through this component the user can choose an open data dataset from recommended datasets by SPOD using a classic drop-down-list, a treemap, or simply copying and pasting an external (remote) dataset url.
- **fields-selection-controllet:** this component is composed by and connect two sub-component, the tree-view-controllet and the multi-table-controllet. Enable users to choose the fields of dataset from a tree-view and visualize the related data in the multi-table (a table for each tree branch).
- **datalet-selection-controllet:** this component allows the user to select a visualization using the subcomponent items-vslider-controllet, select inputs, customize the results via labels and options and see a preview of the datalet.

#### 5.1.1.3 PERSISTING DATALETS IN THE DATABASE

When a user creates a Datalet, the following information is saved in the database:

- The chart name (bar chart, area chart, map etc...).
- The chart configuration (inputs-fields mapping, visualization settings, etc...).
- The input dataset URL.
- Cached data used by the chart.

When SPOD needs to render a Datalet, it performs the following steps:

1. Loads all the above information from the database.

2. Asks DEEP for the embeddable datalet (identified by chart name).
3. Injects the Datalet configuration.
4. Places the Datalet in the page.

The Datalet will fetch the data from the dataset URL and visualise it. In fact, visualised data is always downloaded in real-time when the user tries to visualise the Datalet in order to guarantee the freshness and the truthfulness of the data.

Cached data stored in the DB is only used when the Datalet is shown into the My Space plugin, for performance reasons.

### 5.1.2 OPEN WALL

The goal of the Open wall plug-in is to welcome users, showing anonymous information at a glance about the status of the SPOD installation.

This information is provided by “widgets”, a core concept of Oxwall. A widget is a small component of a plug-in that can be embedded in other pages. By integrating different widgets, the Open wall can fetch data from different plugins and show it to the user.

Some examples of information that the Open wall can show:

- Number of total registered users and active users on the SPOD.
- Number of rooms in the Agora.
- Last discussed dataset.
- Interactive navigation among datasets provided by associated data providers.

The Open wall is the only “dynamic” section of SPOD visible to unauthenticated users, and its primary purpose is to let guest visitors know that the platform is alive, that other people are discussing on interesting topics, and therefore try to encourage them in signing in.

Since the Open wall is publicly visible, it doesn’t expose any kind of sensitive information; for instance, widgets are designed so that users’ names are not visible until the visitors log into the platform.

### 5.1.3 AGORA

The Agora plug-in adds to SPOD a new area, where users can create a “public room” to start a new discussion/debate. The discussion is public, therefore all users will be able to participate.

When a user creates a public room, he can create a set of recommended dataset. Recommended datasets are available to other users in the room that can use them if they want (Figure 15).

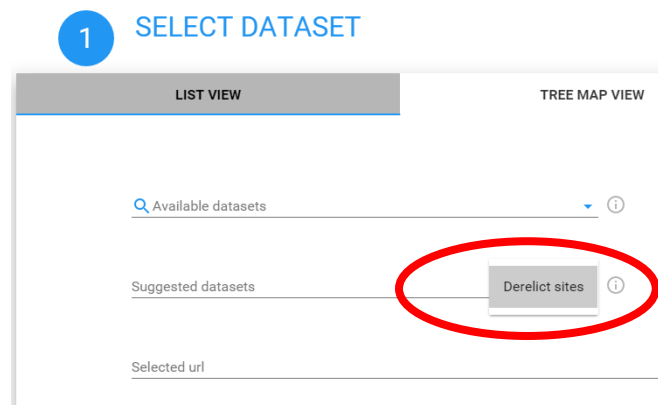


Figure 15. When a user creates a new Datalet in a Private room, he can access the list of suggested datasets.

Once created a room, it can not be removed later (the deletion of a room will be provided in the Beta version, according to the role of the user). Each room is represented using a synthetic visualization by three kind of graphical metaphors: colour, height and width:

- Colour is a shade that goes from blue to red. If the room is little visited by users, it is considered as a *cold* topic and it will be shown in blue. The more visit has the room, the more reddish will be its colour. So the colour of the room is picked from a blue-to-red gradient proportionally to the number of visits: the least visited room will be represented in blue; the most visited room will be represented in red.
- Height is picked between two values, so a room can be short or tall (unlike the colour, there are no intermediate values). Short rooms will have fewer comments; while tall rooms will have a lot of comments.
- Width, just like height, is picked between two values, so a room can be narrow or wide. Narrow room will have fewer datalets used in it; while wide rooms will have a lot of datalets attached.

Height and Width are picked based on simple calculation. The plug-in calculates the median of the number of comments per room (for the height) and the median of the number of datalets per room (for the weight). For instance, if a specific room has fewer comments than the median, that room will be short; otherwise it will be tall. The same reasoning applies for the weight and datasets per room.

A Room is a Polymer web-component named `fullsize-page-card-controllet`. The disposition of rooms complies with the temporal order and is managed by a Polymer component `animated-grid-controllet` which uses Masonry [38] (a popular JavaScript grid layout library). Masonry works by placing elements in optimal position based on available vertical space.

When the user clicks on a room, it is open and the screen is split vertically: on the left side there is a threaded chat. On the right there is an area (hidden by default), dedicated to graphs.

Using the threaded chat, users can reply messages. They can also attach a dataset visualization to their comment and specify if their comment is in agreement, disagreement or neutral to the comment they are replying to. Dataset visualization can be chosen from “My space”, created starting from a recommended dataset or created a new. The number of nested levels for the threaded chat is limited to avoid straying too far from the main topic and to keep conversation easily readable.



Figure 16. The Agora main view, with tiled rooms.

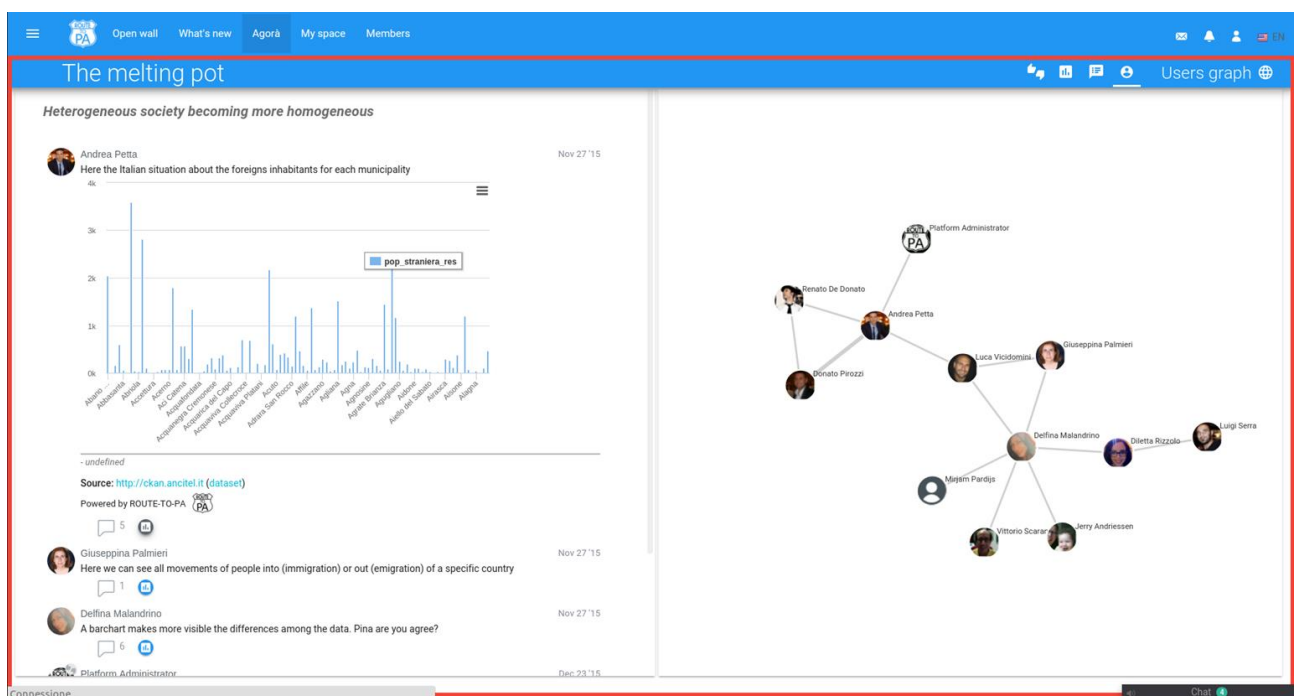


Figure 17. The Agora, viewing inside a room.

There are four graph visualising additional, synthetic information about the discussion: Comments Graph, Datalets Graph, Users Graph, Opinions Graph.

1. **Comments Graph**: the nodes represent the comments and the edges the parent-child relations. The size of a node is directly proportional to the magnitude of sub-tree (total numbers of nested comments). The node colour can be green, red or blue, depending on the comments, respectively in agreement, disagreement or neutral with the parent comment. If the user places the mouse over a node, it is shown the comment that refers to that node and on click the threaded chat will move to and will highlight the selected comment.

2. *Datalets Graph*: the nodes represent the datalets, the edges can represent a parent-child relations or the use of the same dataset. The dimension of a node is directly proportional to the magnitude of sub-tree (numbers of nested comments). On mouse over a node it is shown the comment that refers to that node and on click the threaded chat will move to and will highlight the selected datalet.
3. *Users Graph*: the nodes represent the users participating in the discussion, the edges represent the presence of interaction between two users, and the thickness is directly proportional to the number of interactions.
4. *Opinions Graph*: comments graph with the green comments grouped. That's because the sentiments of the agreement are rowing in the same directions, while one cannot say the same of the sentiments of disagreement, which may diverge from each other.

The graphs leverage D3.js, a JavaScript library for manipulating documents based on data.

#### 5.1.4 MY SPACE

The “My space” plug-in adds to SPOD a new area where users can manage datalets, text and link in a personal area, as reminder or for later use as post or comment (Figure 20).

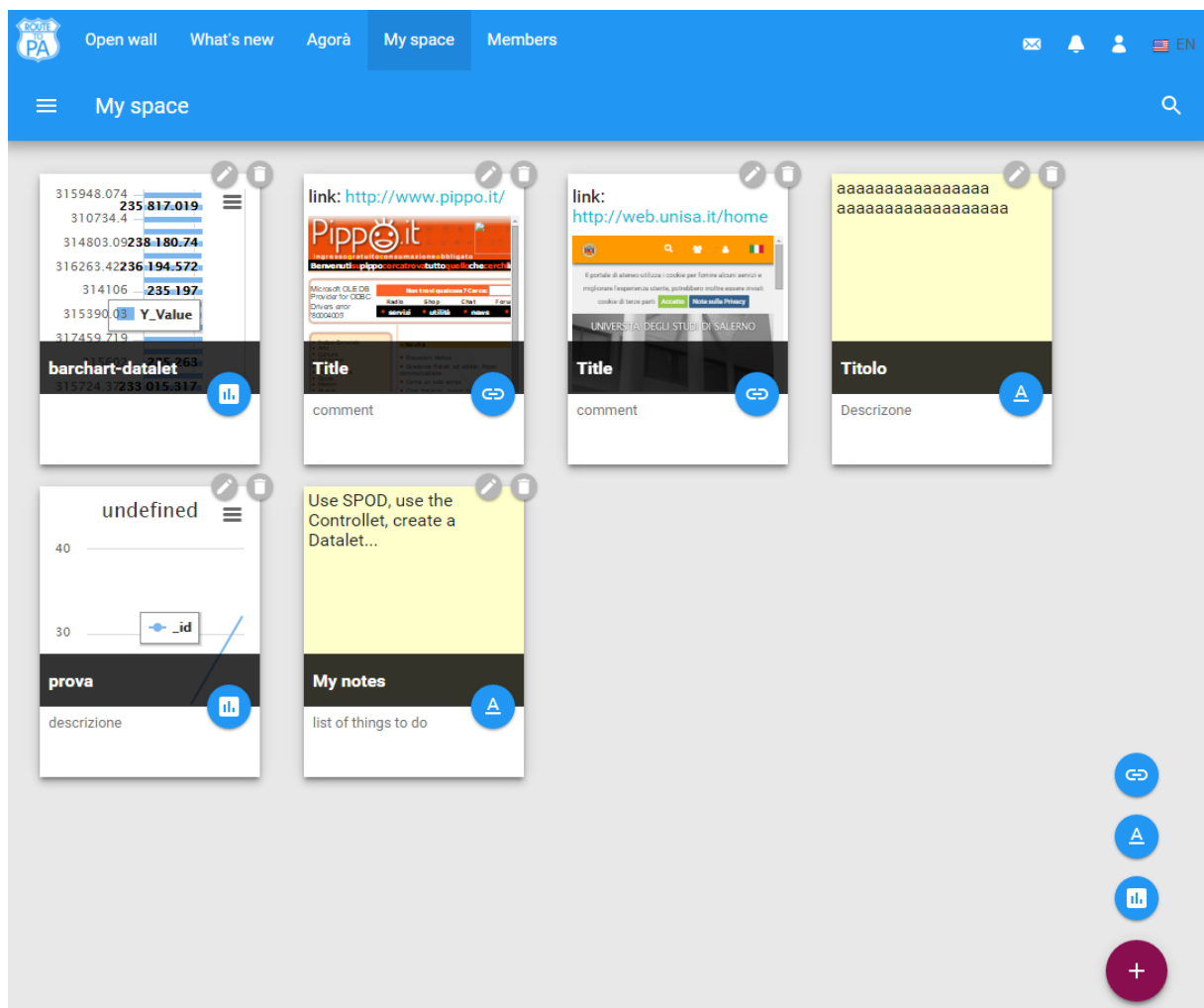


Figure 18. My space.

The created content (Datalet, text or link) is put in a Polymer web-component named paper-card-controllet. Makes it possible at any time modify or delete a card using the buttons on the top-right.



To display content, just click on the blue button on the right side of the card. When a user tries to view its own datalets in their private space, SPOD does not contact the actual data provider but retrieves the data from its own database. This allows a great speed improvement and a better user experience.

The dimension of cards is fixed. The layout of cards complies with the temporal order and is managed by a Polymer component animated-grid-controllet which uses Masonry (as seen for the Agora plug-in).

It is possible to create a datalet using the Controllet interface meanwhile for the creation of text and links there is a dedicated interface.

Figure 19. Creating a text memo in My Space.

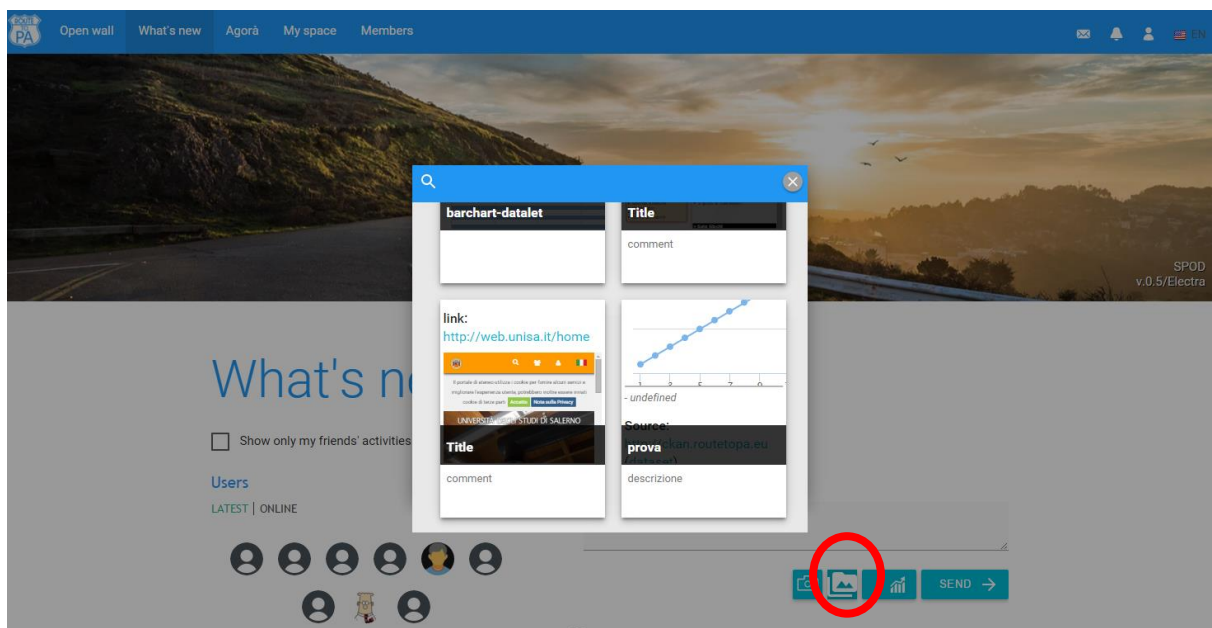


Figure 20. Attaching an element from My space to a new post.

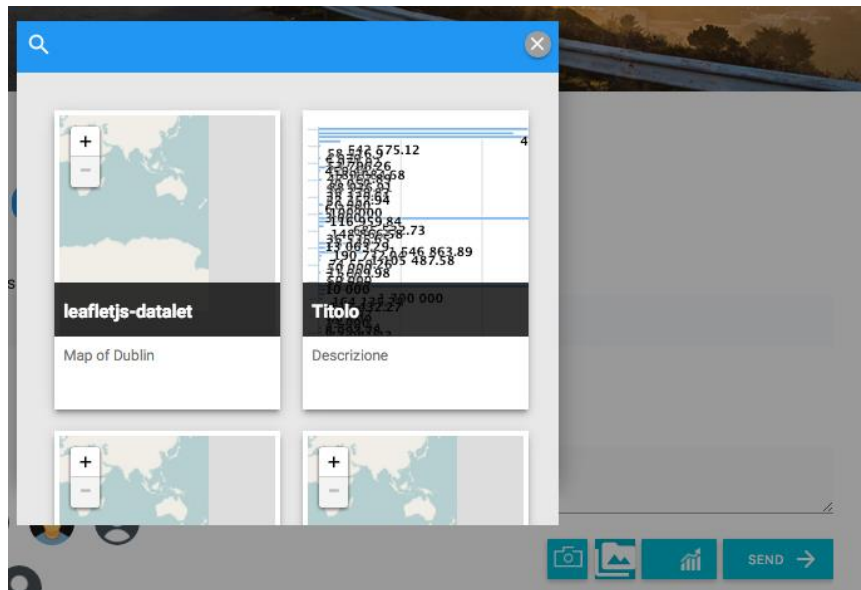
## 5.2 DEEP CONTROLLETS AND DATALETS

In the following section, we show a list of of the Controllets and Datalets that are available within the DEEP component of the architecture, described in section 4.3.

### 5.2.1 CONTROLLETS

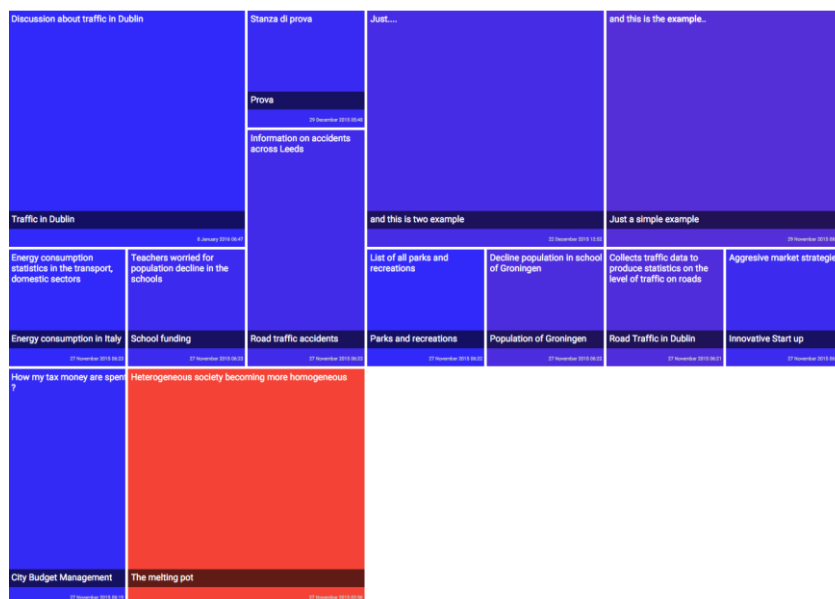
#### 5.2.1.1 ANIMATED-BUTTON-CONTAINER-CONTROLLET

In SPOD, an user might want to share his/her activities in the own private space (i.e., My space) and clicking on the Datalet button all activities will be shown, such as data visualisation or an interesting link previously stored.



### 5.2.1.2 ANIMATED-GRID-CONTROLLET

All public discussions will be shown and gathered in the Agora of SPOD, that it is represented as a set of discussion room where all users will be able to participate (see 5.1.3).



### 5.2.1.3 CREATE-CARD-CONTROLLET

It provides the user interface for creating a new card in the My space of SPOD, where an user can work on their ideas and in the same time s/he can take notes in the own private space (see 5.1.4).

BASE INFO

title

description

TEXT

text

SAVE

PREVIEW

#### 5.2.1.4 DATA-SEVC-CONTROLLET

It provides the basic user interface container to create a new datalet in SPOD.

#### 5.2.1.5 DATALET-SELECTION-CONTROLLET

It provides the third user interface of the controllet (data-sevc-controllet) to create new datalet. In SPOD, an user clicks on Datalet button and a wizard contains some steps for facilitating the data visualization of a data source (see 5.1.1.2).

search

↑

datatable

barchart

columnchart

linechart

column3dchart

areachart

donutpie3dchart

↓

INPUTS

Categories

FENOMENI

Series

PUNTORUGIADA °C

BASE INFO

title

test

description

test description

LABELS / OPTIONS

x-axis-label

text x

y-axis-label

test y

suffix

degree

PREVIEW

test

test test description

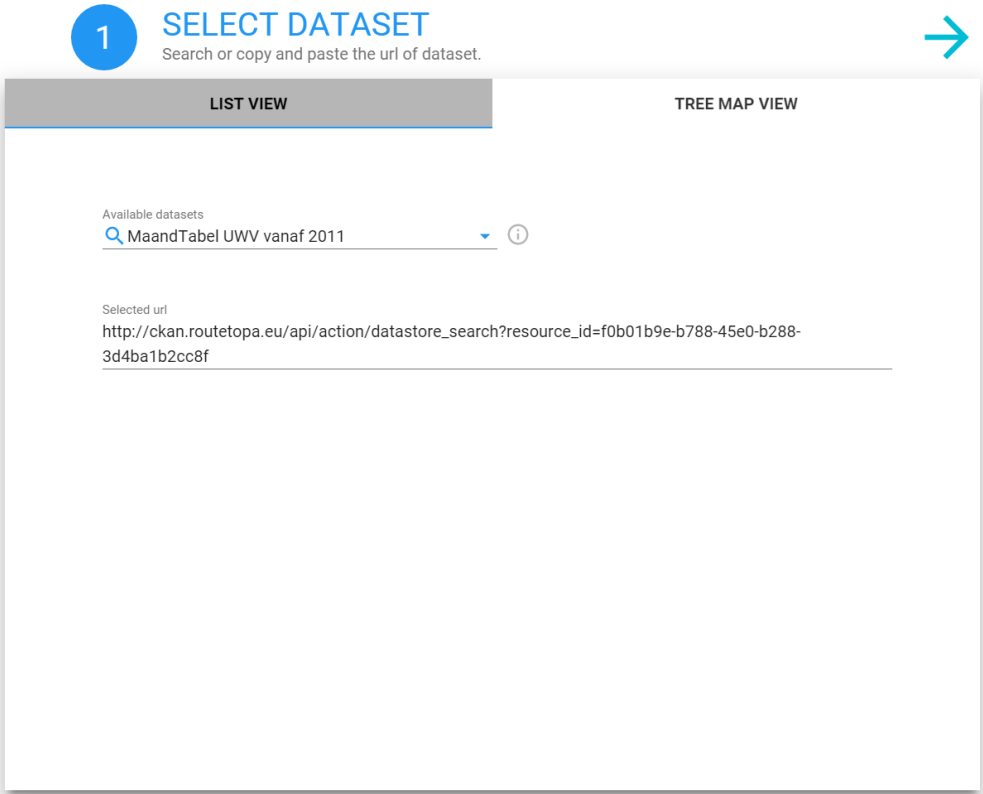
Source: <http://ckan.routetopa.eu> (dataset)

Powered by ROUTE-TO-PA <sup>2020</sup> (PA)

ADD

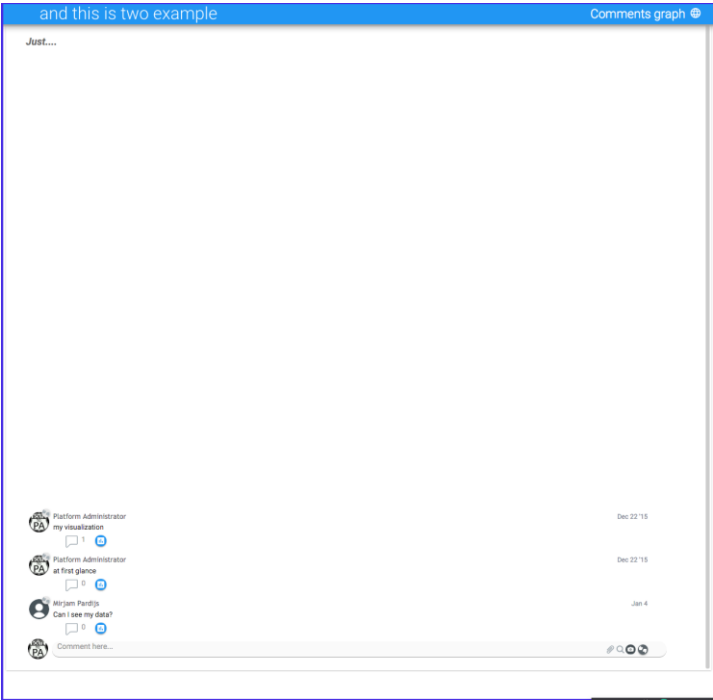
#### 5.2.1.6 DATASET-SELECTION-CONTROLLET

It provides the first user interface of the controllet (data-sevc-controllet) to create new datalet, allows the user to select the data source in different ways: list view, tree map view and copy and paste (see 5.1.1.2).



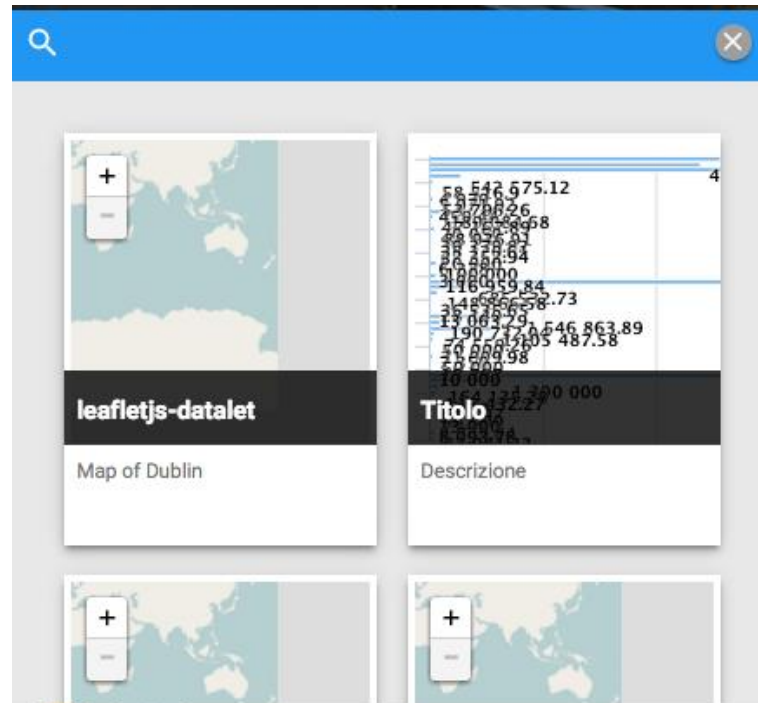
5.2.1.7 FULLSIZE-PAGE-WITH-CARD-CONTROLLET

It provides the basic page for a discussion in the Public Room of the Agora where an user clicks on a discussion room and a threaded discussion can start, for example, replying messages or attaching a datalet to the discussion (see 5.1.3).



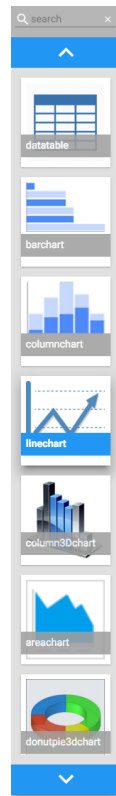
#### 5.2.1.8 GENERIC-CARDS-CONTAINER-CONTROLLET

It provides the user interface for selecting a datalet or a link to post previously organized by the user in their own private space, i.e., My space of SPOD (see 5.1.4).



#### 5.2.1.9 TEMS-VSLIDER-CONTROLLET

It provides a slider to select the datalet for visualizing a data source. When a user decides to visualise a data source, s/he clicks on the Datalet button and the step 3) of the wizard provides a data visualization catalogue with different types of charts that an user can use to visualize a dataset (see 5.1.1.2).



#### 5.2.1.10 MULTI-TABLE-CONTROLLET

It provides a user interface for visualizing a data source fields in a table. It is a multi table because can visualize different branches of the data available when a user chose a dataset with the JSON format (e.g., OpenDataSoft portal) in the step1) of the Controllet (see 4.3.5).

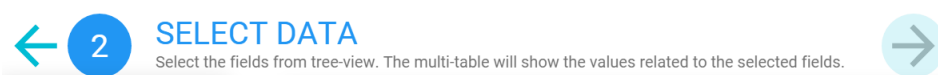
result / records					
Show <input type="text" value="10"/> entries			Search: <input type="text"/>		
_id	aant_won_zelfst	naam_subbuurt	nr_subbuurt	aant_won_totaal	aant_won_onzelfst
1	189	Academiekwartier	14000000	533	344
2	441	Ebbingekluft	14000001	902	461
3	159	Bisschopskwartier	14000002	393	234
4	453	Westerkluft	14000100	723	270
5	511	Winkelkwartier	14000101	1291	780
6	255	Sint Geertruidskwartier	14000102	582	327
7	246	Drenkelaarskwartier	14000103	467	221
8	495	Buurmande	14000104	770	275
9	405	Het Olde Rondeel	14000105	564	159
10	226	Oosterhaven	14000200	450	224

Showing 1 to 10 of 100 entries

Previous 1 2 3 4 5 ... 10 Next

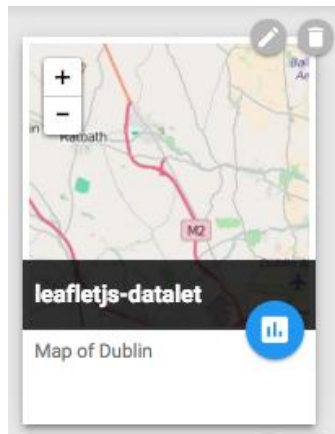
#### 5.2.1.11 PAGE-SLIDER-CONTROLLET

It provides a slider component to move between different pages. In the Controllet, a wizard can help the user among the steps of a data visualization (see 4.3.5).



#### 5.2.1.12 PAPER-CARD-CONTROLLET

It provides a card component used in the My space of SPOD, when a user works in his/her private room, adding datalets, notes or links that s/he can use later in public discussion, i.e., the Public Room (see 5.1.4).



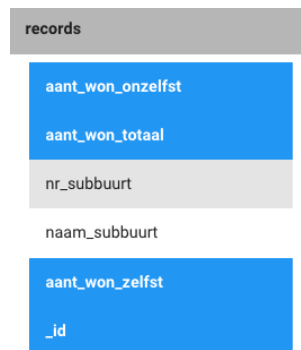
#### 5.2.1.13 SEARCH-PANEL-CONTROLLET

It provides the user interface to search something. An user can quickly make fast search own private space, i.e., My space, (see 5.1.4).



#### 5.2.1.14 TREE-VIEW-CONTROLLET

It provides the user interface to select the data source fields in a tree view. When in SPOD, a user decides to visualize a data source, s/he clicks on the Datalet button and the step 2) of the wizard provides a view of all data of the data source (see 5.1.1.2).



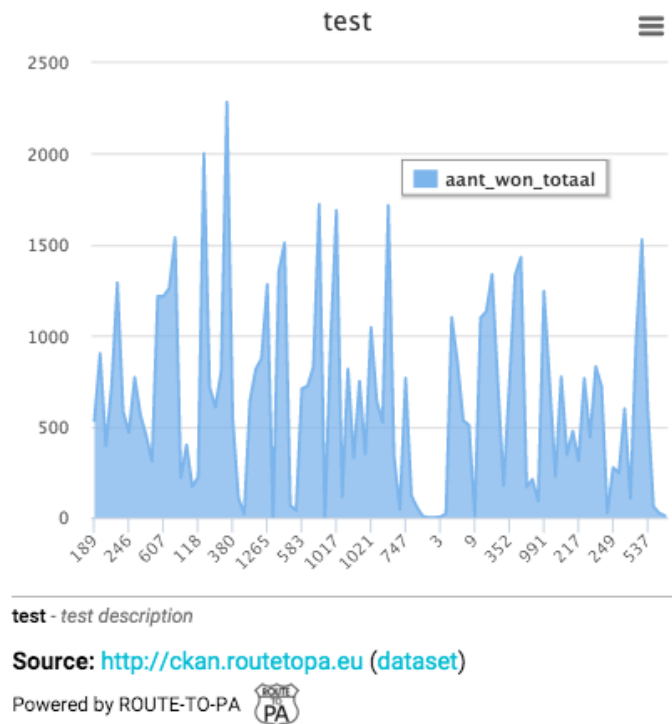
#### 5.2.1.15 TREE-VIEW-MULTI-TABLE-CONTROLLET

It provides the second user interface of the controllet (data-sevc-controllet) to create new datalet, allows the user to select the fields of a data source from a tree view and visualize it in a multi table.

### 5.2.2 DATALETS

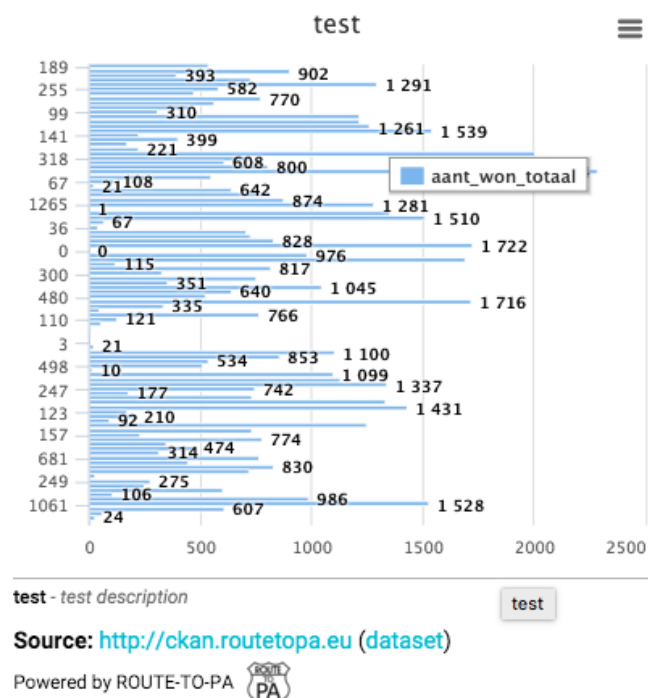
#### 5.2.2.1 AREACHART-DATALET

Area chart datalet based on Highcharts project.



#### 5.2.2.2 BAR-CHART-DATALET

Bar chart datalet based on Highcharts project.



#### 5.2.2.3 BASE-AJAX-JSON-JSONPATH-DATALET

Defines the base datalet and defines the mechanism to access and select data. This datalet handles AJAX request with JSON response and uses JSONPATH for data selection and filtering.

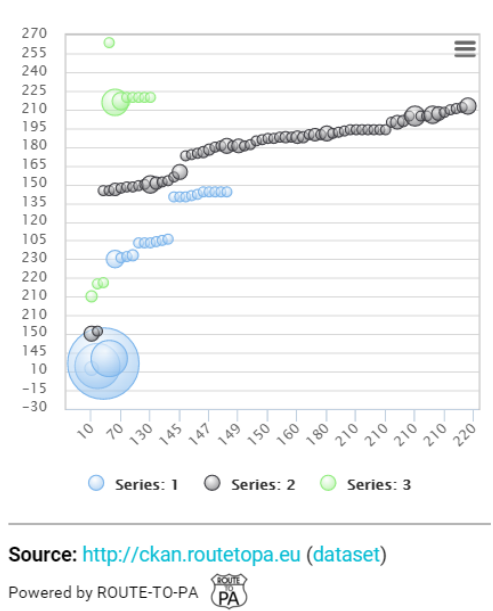
#### 5.2.2.4 BASE-DATALET



Defines the base component that includes datalet footer (with information about dataset domain and ROUTE-TO-PA project). Base datalet includes polymer.html and the BaseDataletBehaviors javascript file that define the datalet workcycle and the datalet base properties: dataUrl, fields and data.

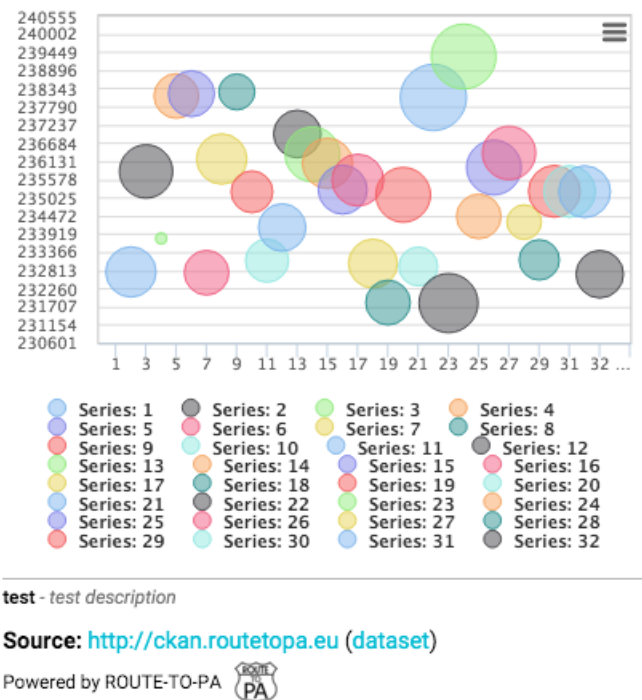
5.2.2.5 BUBBLE3D-DATALET

3D bubbles chart datalet based on Highcharts project.



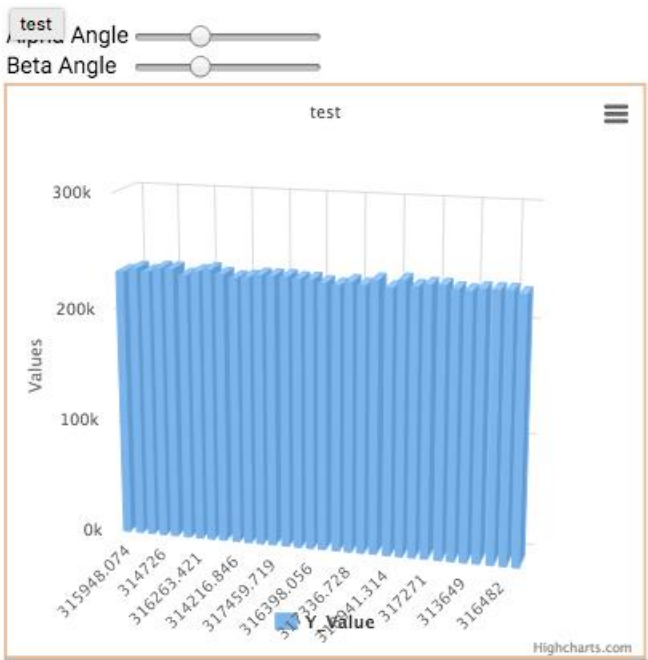
5.2.2.6 BUBBLECHART-DATALET

Bubbles chart datalet based on Highcharts project.




5.2.2.7 COLUMN3DCHART-DATALET

Column 3D chart datalet based on Highcharts project.



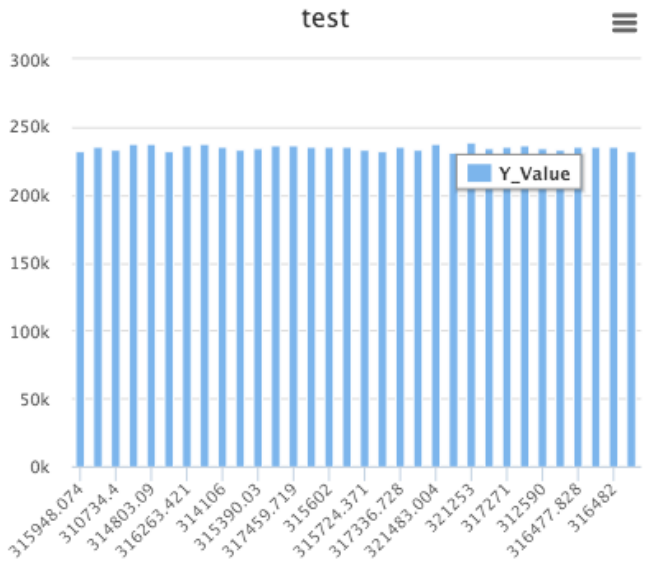
test - test description

Source: <http://ckan.routetopa.eu> (dataset)

Powered by ROUTE-TO-PA 

5.2.2.8 COLUMNCHART-DATALET

Column chart datalet based on Highcharts project.



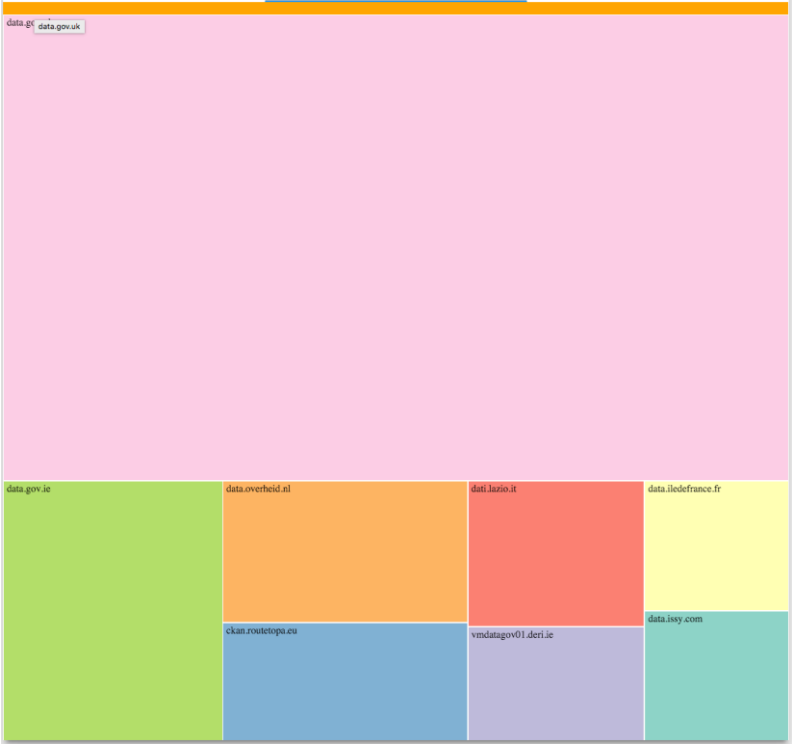
test - test description

Source: <http://ckan.routetopa.eu> (dataset)

Powered by ROUTE-TO-PA 

5.2.2.9 DATASETEXPLORER-DATALET

Treemap datalet based on D3.js treemap project <http://bl.ocks.org/mbostock/4063582> . A treemap recursively subdivides area into rectangles; the area of any node in the tree corresponds to its value. This is an enhanced version designed to help users to navigate associated dataset providers.



5.2.2.10 DATATABLE-DATALET

Porting of Datatables JQuery library in a web component that has built up by using Polymer. Pass to this component a data URL (CKAN API URI) and a string with one or multiple query in JSONPath format (separated by spaces) and it'll show a table with data and query attributes with all Datatables library features.

Show  entries

Search:

_id	Y_Value	X_Value
1	test	0.188 315948.074
2	235817.019	316887.551
3	233777.46	310734.4
4	238112	314726
5	238180.74	314803.09
6	232736	314694
7	236194.572	316263.421
8	238235	312776
9	235197	314106
10	233103.522	314216.846

Showing 1 to 10 of 32 entries

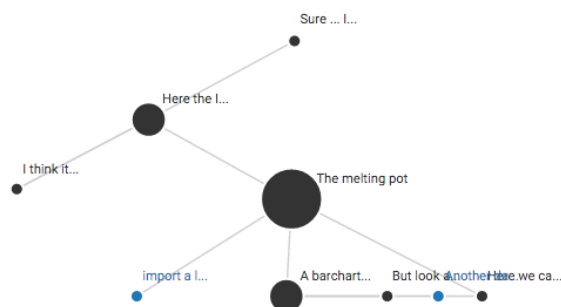
test - test description [Previous](#) [1](#) [2](#) [3](#) [4](#) [Next](#)

Source: <http://ckan.routetopa.eu> (dataset)

Powered by ROUTE-TO-PA 

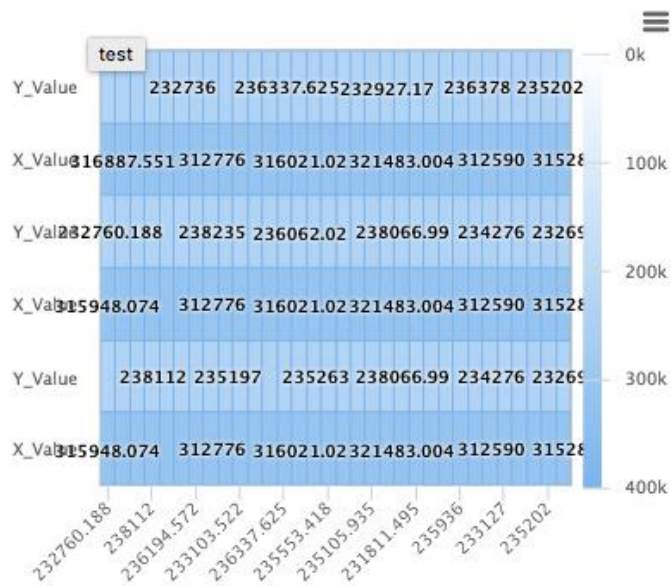
#### 5.2.2.11 GRAPH-DATALET:

Graph comments datalet used in the Agora to visualize the comments on a topics.



#### 5.2.2.12 HEATMAP-DATALET

Heat map datalet based on Highcharts project.

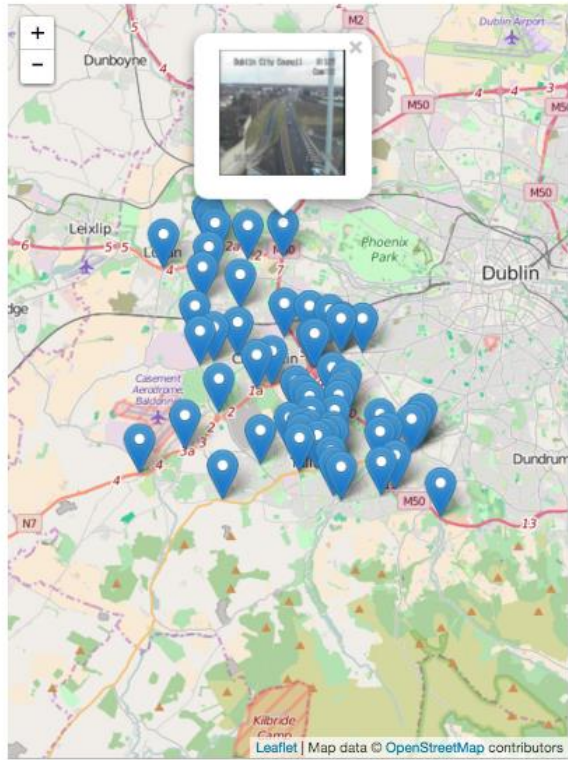


#### 5.2.2.13 HIGHCHARTS-DATALET

Extends the base-ajax-json-jsonpath-datalet as data retrieving and selection mechanism. It defines the common properties for all the Highcharts datalets: categories, series and series\_type. Overmore override the transformData method to create and populate categories and series array. Every specific Highcharts chart implementation (e.g., linechart, barchart, columnchart ...) will use Highcharts-datalet properties to create the appropriate chart.

#### 5.2.2.14 LEAFLETJS-DATALET

Map datalet based on open source project Leafletjs.



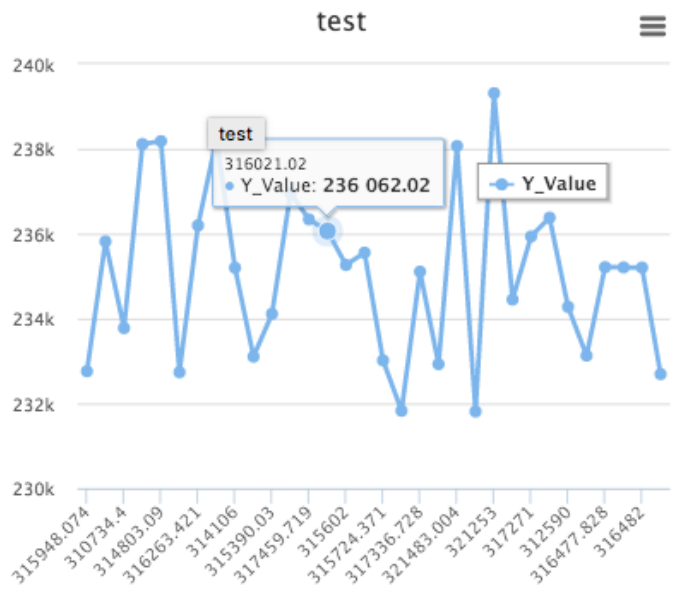
test - test description

Source: <http://ckan.routetopa.eu> (dataset)

Powered by ROUTE-TO-PA 


#### 5.2.2.15 LINECHART-DATALET

Line chart datalet based on Highcharts project.



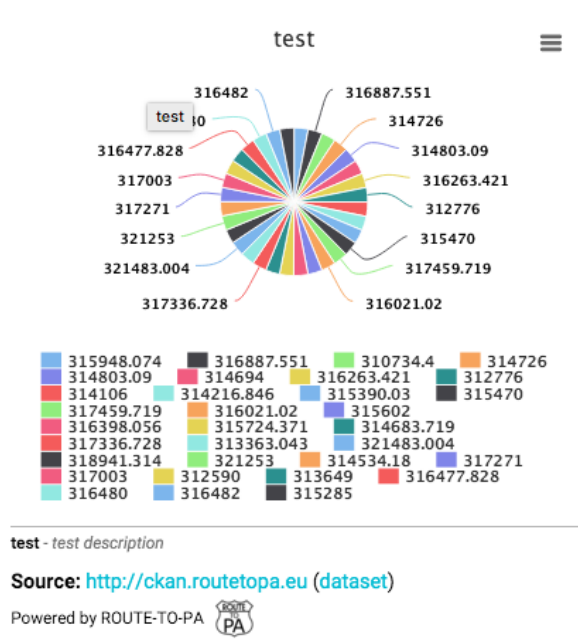
test - test description

Source: <http://ckan.routetopa.eu> (dataset)

Powered by ROUTE-TO-PA 

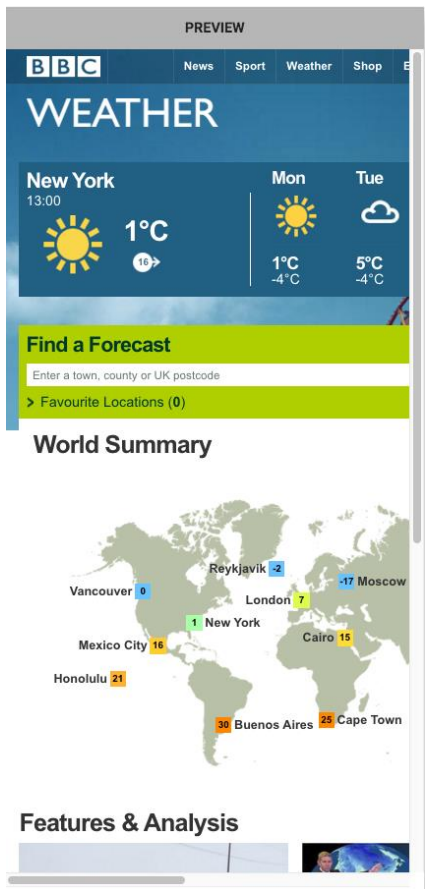
#### 5.2.2.16 PIECHART-DATALET

Pie chart datalet based on Highcharts project.



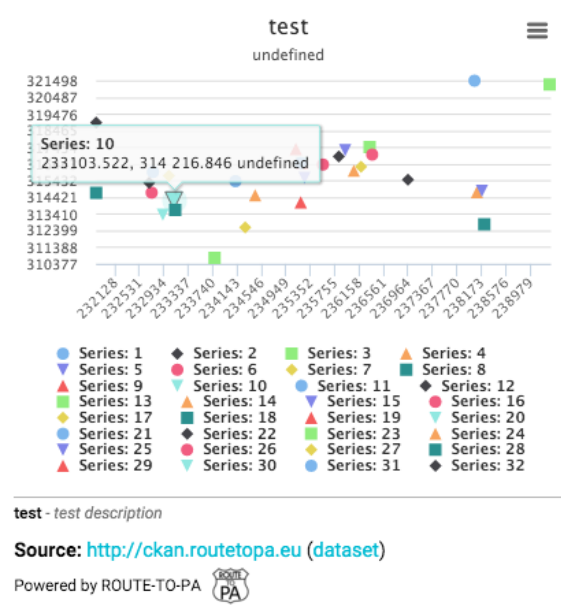
5.2.2.17 PREVIEW-DATALET

Datalet that allow user to preview the content of a web page. It creates a thumbnail of the site using the data-url attribute passed as input.



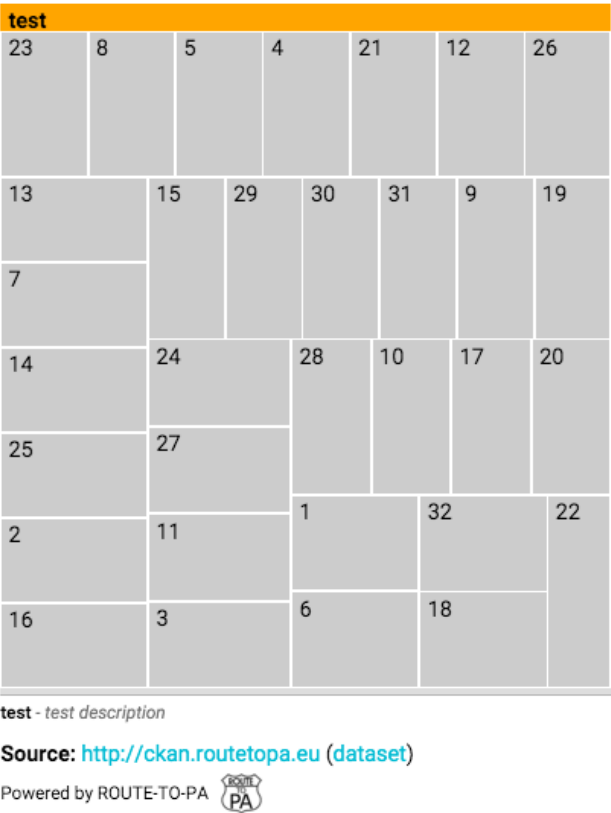
5.2.2.18 SCATTERCHART-DATALET

Scatter plot chart datalet based on Highcharts project.



5.2.2.19 TREEMAP-DATALET

Treemap datalet based on D3.js treemap project <http://bl.ocks.org/mbostock/4063582>. A treemap recursively subdivides area into rectangles; the area of any node in the tree corresponds to its value.



5.2.3 DEEP: ADDITIONAL SERVICES

The DEEP provides some additional services:



First of all, it is possible to generate the code for a new datalet to be added in any standard HTML page (a demo is available at the link [demo](http://deep.routetopa.eu/COMPONENTS/demo.html)<sup>7</sup>). This service allows the user to insert the datalet of SPOD in a personal blog, as well as other social networks and/or private webpages. DEEP will be also used to develop a SPOD-analytics services that is a report of statistics and metrics about the utilization of data sources, the number of discussions involved, the kind of visualization used and so on. This kind of information can be used to suggest popular data set about some topic and popular visualization for some particular dataset. We acknowledge that it is extremely hard to envision the correct visualization of a dataset, since the visualization is extremely correlated to the semantics of the data and the kind of information the user is looking for. Nevertheless, we believe the statistic information collected using the DEEP can be useful to provide some hint to the users.

### 5.2.3.1 DATALET WITHIN “DATALET CREATOR”

Datalets can be statically embedded in any web page (e.g., blog, forum, institutional web-site, and so on) through a copy-and-paste of HTML source code made available by the “Datalet Creator”, thereby including the Datalet statically into any HTML editor. Thereby, this visualization tool is an open source model for non- programmers, that is any users can quickly generate multiple view on the same dataset to integrate into any website with the same wizard (e.g., “Controllet”), described in the previous section. Among the most popular CMSs, we verified the full compatibility with WordPress<sup>8</sup> and Tumblr<sup>9</sup>. The Datalet Creator was released with the SPOD version 0.5. Due to some changing in the WC architecture, a new version of the Datalet Creator must be developed and will be released for the Beta version of SPOD.

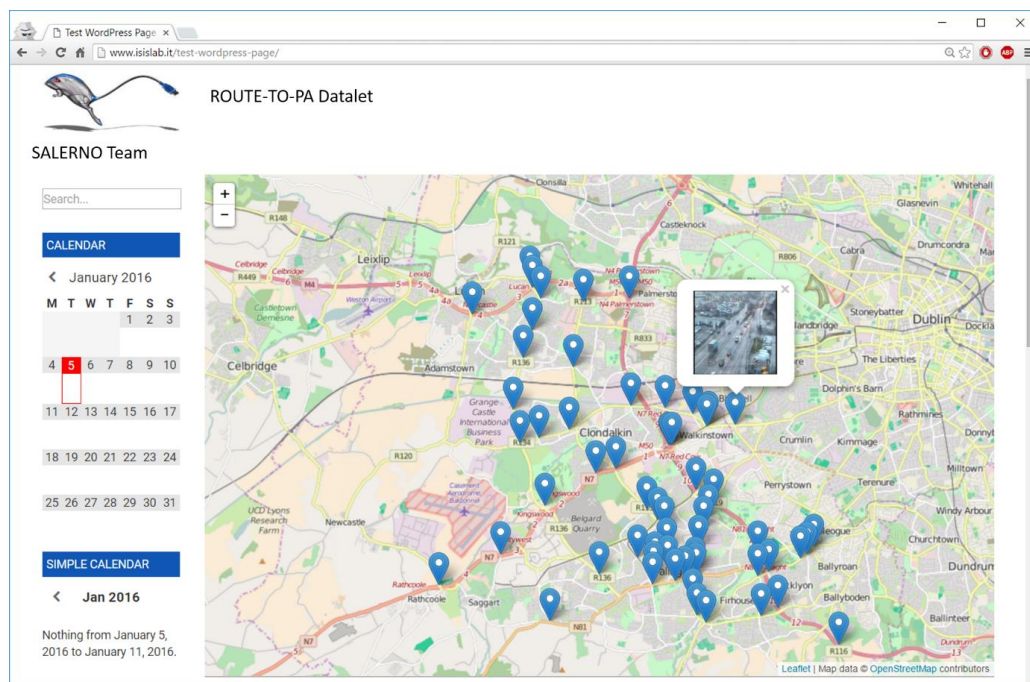


Figure 21: a Datalet embedded, e.g., an interactive map of Dublin, into WordPress page of the research laboratory of the SALERNO team.

## 5.3 INTERACTIONS BETWEEN SPOD AND TET

<sup>7</sup> <http://deep.routetopa.eu/COMPONENTS/demo.html>

<sup>8</sup> <https://wordpress.com/>

<sup>9</sup> <https://www.tumblr.com/>

A ROUTE-TO-PA user (i.e., a citizen, PA, etc.) should be able to log in once, and use both SPOD and TET functionalities. Each user will only need a single authentication credential to use both systems, and can have relevant profile details (i.e., personal info, rewards, roles, etc.) on both the platforms.

Interactions between SPOD and TET systems can be organised considering the exchange of data and the actions. The **information flow** concerns the flow of data from one system to another one, the system A reads the data from system B and use them, without changing the status of system B. The **actions** are interactions performed by the user on the system A and forwarded to the System B; actions may change the status of both systems. This model of communication between SPOD and TET will allow the (TET or SPOD) user to move easily from one to another through the “contextual” links provided by each of the two systems.

In this section we provide the analysis of the interactions between SPOD and TET, that will be useful in the Beta version, in order to address some of the Use Cases from D2.4 that are not currently covered by version Alpha.

### 5.3.1 INFORMATION FLOW FROM SPOD TO TET AND VICE VERSA

The following table summarises the information flow between SPOD and TET.

ID	Name	Short Description
<b>Information flow from SPOD to TET</b>		
<b>DataFlowSPOD1</b>	User profile	TET can access to the SPOD users’ profile
<b>DataFlowSPOD2</b>	Social Activities	For each dataset, TET shows the social interaction (e.g., number of comments)
<b>DataFlowSPOD3</b>	Activity Level of a dataset	For each dataset, TET shows the level of activity on SPOD (e.g., number of views, linking, etc.)
<b>Information flow from TET to SPOD</b>		
<b>DataFlowTET1</b>	All available datasets	TET provides the available datasets in the open data platform
<b>DataFlowTET2</b>	Available datasets by keyword	TET provides the available datasets filtered by keyword
<b>DataFlowTET3</b>	Suggested datasets	Based on the metadata and other similarity measures, TET recommends published datasets
<b>DataFlowTET4</b>	New dataset notification	SPOD receives notifications of new datasets uploaded on TET
<b>DataFlowTET5</b>	TET statistics	Provides overall statistics about usage and datasets available on TET

Table 8 - Information Flow from SPOD to TET and vice versa

Data that flow from SPOD to TET concern the SPOD user profile and the SPOD social activities.

**[DataFlowSPOD1] Information flow from SPOD to TET, User Profile** - Each user on SPOD has a profile with personal information (i.e., gender, age, etc.), and roles (i.e., occupation or position held etc.), rewards, etc. This profile must be accessible from TET in order to enhance personalization. Furthermore, all activities of a user will allow to build a user profile for personalised searches. This requirements have been pointed out in the UC17 (“Personalize Search”, TET must be able to retrieve user profile, including user profile on SPOD) and UC19 (“Request Recommendations for Data Sets”, TET creates a list of potential interested users with a profile on SPOD) of deliverable D2.4.

**[DataFlowSPOD2] Information flow from SPOD to TET, Social Activities** - For the social activities, TET should be able to show for each dataset, the “social” behaviour of its author on SPOD. Therefore, TET will be able to show for each user, author of a dataset, information about:

- Number of friends and network graph
- Level of activity (number of comments, number of positive votes, etc.)
- Most recent activities (links to the recent discussion where the author intervened, etc.)

- Content of posts and comments made by the user

**[DataFlowSPOD3] Information flow from SPOD to TET, Activity level of a Dataset** - TET asks SPOD to provide the required information for a given dataset:

- how much dataset is used and how, e.g., linked, seen, acted upon, etc...
- in how many different discussions (with links to them)
- from how many different SPOD users
- with what degree of success (if there is a “like button” 😊)
- what kind of visualizations are mostly used
- related datasets used in the same discussion

**[DataFlowTET1] Information flow from TET to SPOD, All available datasets** - SPOD asks TET the list of all the available datasets. CKAN already has such an API (but maybe has to be expanded).

**[DataFlowTET2] Information flow from TET to SPOD, Available datasets by keyword** - SPOD asks TET the list of all the available datasets, searching with keyword. CKAN already has such an API (but maybe has to be expanded).

**[DataFlowTET3] Information flow from TET to SPOD, Suggested datasets** - SPOD asks TET some “relevant” published dataset with respect to a given one, based on metadata and other similarity measures.

**[DataFlowTET4] Information flow from TET to SPOD, New dataset notification** – A SPOD user can receive notifications of a new uploaded datasets on TET.

**[DataFlowTET5] Information flow from TET to SPOD, TET Statistics** - SPOD requests TET to get statistics about datasets and TET usage.

### 5.3.2 ACTIONS FORWARDED FROM SPOD TO TET AND VICE VERSA

The following table summarises actions that each system should provide in order to have a seamless user experience.

ID	Name	Short Description
<b>Actions performed on SPOD and forwarded to TET</b>		
<b>ActionSPOD1</b>	Open TET dataset page	User clicks on the dataset link on SPOD, and TET shows the corresponding dataset page
<b>ActionSPOD2</b>	Dataset rating and promoting	When rewarding is activated, the user can rate a TET dataset
<b>ActionSPOD3</b>	Ask for a dataset to PA	A user can ask a dataset to PA. The request is performed through TET.
<b>ActionSPOD4</b>	Contact user that authored the dataset on TET	User can involve in the discussion the author of the dataset (on TET) through private messages or bringing him into the discussions by signalling an interesting discussion
<b>Actions performed on TET and forwarded to SPOD</b>		
<b>ActionTET1</b>	Copy to my private room	Within TET a logged user can copy a link to a dataset in her/his SPOD private room
<b>ActionTET2</b>	Post on SPOD	Within TET a logged user can share a dataset on SPOD

Table 9 - User actions forwarded from one system to the other one.

**[ActionSPOD1] Actions performed on SPOD and forwarded to TET, Open TET dataset page** – SPOD shows an open dataset visualization with a link to the open data source. The user clicks on the link loading the relevant data provider page.

**[ActionSPOD2] Actions performed on SPOD and forwarded to TET, Dataset rating and promoting** - After the activation of the rewarding option, a SPOD user clicks on a dataset “appreciation”/“rating” icon and the corresponding action is realized on TET.

**[ActionSPOD3] Actions performed on SPOD and forwarded to TET, Ask for a dataset to PA** - A SPOD user can ask to the PA (through TET, maybe a user with a specific role) an improved or a new dataset, by adding a quick explanation of what is required (like “I would like to have the same dataset for the last 3 years”, or “Please, add more information about the position of the services in this dataset”). The request is then forwarded to TET to the author/publisher of the dataset or to a specific user.

**[ActionSPOD4] Actions performed on SPOD and forwarded to TET, Contact user that authored the dataset on TET** - A SPOD user can ask to involve in the discussion the author (on TET) of the dataset with a private message, bringing expertise into a discussion by signalling an interesting discussion, etc.

**[ActionTET1] Actions performed on TET and forwarded to SPOD, Copy to my private room** - In a dataset page of TET, there is a mechanism (a button, a draggable area, etc.) so that the user can “take” some content from TET into his/her private room on SPOD. The user can take different kind of data, such as the URL of dataset, a text snippet, an image, etc. This is based on the UC28 of deliverable D2.4 (“Digest information in a private space”).

**[ActionTET2] Actions performed on TET and forwarded to SPOD, Post on SPOD** - On dataset page in TET, a user should be share dataset/file on SPOD. This is based on the UC4 and UC5 of Deliverable D2.4 (“Share link to dataset/File via SPOD post”).

## 5.4 A RECAP OF FEATURES AND REQUIREMENTS

Here we provide a summary of where in SPOD has been implemented the feature requested by each requirement listed in section 2.1. The SPOD Manual is available at the address <http://spodalpa.routetopa.eu/Manual/SPODManualCurrentVersion.pdf>

	Requirements (by category)	Notes	Where on SPOD and feature description	SPOD Manual
	<b>Social capabilities</b>			
1	Login/Logout	Each user is logged	Top menu bar (right)/Sign in	See Sec. 2.1.1
2	Unified login for SPOD/TET	Citizens should seamlessly move from TET to SPOD and viceversa without logging-in each time.	Top menu bar (right)/ Sign in. A single joint authentication mechanism is provided for SPOD and TET by using ROUTE-TO-PA Authentication Server (RAS)	See Sec. 2.1.2
3	Remote Authentication	Allow interoperability between different SPODs of different PAs. Authentication with Facebook is also technically feasible for Beta and will be considered	Top menu bar (right) /Sign in	See Sec. 2.1.3
4	User Profile	Include editing of personal info (real name, gender, birthday, photo)	Top menu bar (right) / <i>My Profile</i> .	See Sec 2.1.4
5	Private messages	Send private msgs to users	Top menu bar (right)	See Sec 2.1.5
6	Chat	Synchronous chat.	Bottom right/Chat	See Sec 2.1.6
7	Attach a document/image	To any comment, it should be possible to attach documents and images	Everywhere it is possible to post a comment (i.e. What's new, Public Room). Partially resolved in Alpha (e.g., upload of an image and sharing a link) and to be considered for the Beta (e.g.: attach pdf and word documents)	See Sec. 2.1.10
8	Notifications of activities (what's new)	All the activities (e.g. posts, likes, photos etc...) to everybody or only the friends	Top menu bar /Tab: What's new	See Sec 3.2
9	<i>My Preferences</i> : Privacy	How my profile, presence, newsfeed and my friends are public to all, visible only to my friends or private	Top menu bar (right) / My Preferences	See Sec 2.1.8
10	Post a message (status)	Share a post or dataset visualization	Top menu bar / Tabs What's new, Agora	See Sec. 2.1.11
11	Browsing users	All the users, only online, search by name or gender	Top menu bar /Tab: Members	See Sec 2.1.13
12	Post a link or chart visualization previously placed in personal "My space"	Share a link or chart visualization from private space (e.g., My space page)	Top menu bar /Tab: My space	See Sec 2.1.12

13	Add friends and friends requests	Send a private message, add a friend or send a friends request	Top menu bar /Tab: Members	See Sec. 2.1.9
14	Accessibility for users with visual deficiencies	In Alpha version only a particular type of visual deficiencies is considered (see sec. 4.5.1.1)		
	<b>Welcome (Open Wall)</b>			
15	Provide welcome and summary (anonymous) information about the status of the SPOD, also to unauthenticated users	It enables and empowers social interactions by stimulating potential users in registering. Useful information include the number of users registered, logged users, number of discussions, recent content, etc. Useful to provide a URL to perspective potential users.	Top menu bar /Tab: Open wall. In Beta it will be provided with more info about the content of the discussions	See Sec 3.1
16	Interactive navigation among datasets	To encourage users to see what is available in the associated dataset providers.	Top menu bar /Tab: Open wall	See Sec 3.1.2
	<b>What's new</b>			
17	Show all users' activities	Show all social activities (e.g., posts etc...) of all the users included the postings in the Public Room. Potentially (when the platform is in use) with many updates and therefore potentially long.	Top menu bar /Tab: What's new. In the same page the user can switch to show only friends' activities	See Sec 3.2.1
18	Show only friends' activities	Show all social activities (e.g., posts etc...) of all the friends. Since the list of activities of all the users may be very long, it is more helpful to see what the friends are currently interested in.	Top menu bar /Tab: What's new. In the same page the user can switch to show all users' activities	See Sec 3.2.2
	<b>Public Discussion (Agora)</b>			
19	Create a Public Room	A discussion forum should be offered to the citizens on the SPOD	In the Agora, on the bottom right corner. Each room is a tile representation. Deletion is planned for Beta, and the action will depend on the role of the user (creator of the room, administrator, or others)	See Sec 3.3.1
20	Recommended datasets	To stimulate the discussions, relevant datasets should be offered for specific discussions	Click on "My Public Room" (3 horizontal lines icon on the left) within the "Agora". Each room has a set of starting datasets, easily accessible, set by the creator of the room. In the Beta it will be possible to update the set	See Sec 3.3.4
21	Synthetic visualization (graphical metaphor) of the number of comments	A participant should be able to see all the ongoing discussions and recognize easily the ones that are more active	In a Public Room within the Agora. The vertical size of the tile shows the "depth" of discussion, i.e. the number of comments	See Sec 3.3.5

22	Synthetic visualization (graphical metaphor) of the number of datasets used	A participant should be able to see all the ongoing discussions and recognize easily the ones that use more datasets, as discussions that are fruitfully employing open data	In a Public Room within the Agora. The horizontal size of the tile shows the "width" of the discussion, i.e., the number of used datasets	See Sec 3.3.6
23	Synthetic visualization (graphical metaphor) of the number of views	A participant should be able to see all the ongoing discussions and recognize easily the ones that collect more interest among the users	In a Public Room within the Agora. The color of the tile shows if the room is "hot" (red) (meaning that there are many views among the users) or "cold" (blue) (only few visualizations)	See Sec 3.3.7
24	Synthetic visualization (graphical metaphor) of the recency of activities	A participant should be able to see all the ongoing discussions and recognize easily the ones that have been more recently active (i.e. a comment was posted)	In a Public Room within the Agora. Rooms are ordered left-to-right, top-to-bottom with the latest activities first	See Sec 3.3.8
<b>Public discussion (Public Room)</b>				
25	Attach a dataset visualization to a comment, by using a previously created datalet from "My space"	Import from "My space" a link or visualization of a datalet into a comment/contribution to a room	Choose data visualization or link among the ones that are proposed from user's personal "My Space"	See Sec 3.4.1
26	Attach a dataset visualization to a comment, among suggested datasets or a new	Attach a dataset visualization to a comment, using one of the suggested datasets for the public room, or using a new dataset	Add a new dataset or choose a suggested dataset within the controllet, to be attached to the post in the "Public Room"	See Sec 3.4.2
27	Upload Image	Upload a media file (e.g., image) with a contribution to the discussion	Choose images before sending a post in "Public Room"	See Sec 3.4.3
28	Sentiment indicators for a post	Sentiment indicators (agree/disagree/neutrality) for a post	Choose sentiment indicator before sending a post in "Public Room"	See Sec. 3.4.4
29	Split view	Besides the time-dependent thread like visualization of the comments, additional information is shown	On the top right corner of the Public Room. The user can show additional information, in the form of graphs, about the discussion on the left. It can be used also for synchronous navigation: clicking on an item on the right graph shows the corresponding items on the left and viceversa	See Sec. 3.4.3
30	Time-dependent thread-like visualization of comments	On the left of a public room, the discussion is threaded, with 3 levels of comments, most recent at the top	Select a tile (e.g., a Public Room) in the Agora	See Sec. 3.4.2
31	Visualization of additional, synthetic information about the discussion	The participants can see the structure of the discussion by accessing different alternative views, centered around some of the components (like, e.g., users, data, opinions, etc...)	On the top right corner. The Comment graph shows all the comments, nested in three levels, the larger is the node the more descendant comments it has in the graph. The Dataset graph shows the datalets that are placed in the discussion, with an additional edge between datalets that show (possibly different views of) the same dataset. The Users graph shows how each user has responded to other users. An edge connects users that have interacted, with the size of the edge proportional to the number of interactions (comments). The Opinions graph shows the opinions in the comments graph, showing a cluster of the "agree" comments.	

	<b>Dataset visualization (Controllet)</b>			
32	Search box for datasets	Several types of search (e.g., search facilities while typing, search all dataset, search by typing only initial character of a word, search all dataset containing any word or letters)	First step in the data visualization (Controllet)	See Sec. 11.1
33	Visualisation of one of the datasets among a list of easily accessible datasets	List of available dataset for each Pilot (Prato, Issy, etc)	First step in the data visualization (Controllet)	See Sec. 11.2
34	Visualization of any dataset from a compatible provider	Remote address via API available from any CKAN or OpenDataSoft portals (e.g., TET- enabled CKAN platform, Issy-les-Moulineaux, Île-de-France etc...) allows dataset visualization	First step in the data visualization (Controllet)	See Sec. 11.3
35	Choice of a chart	List of available charts: table, barchart, column chart, areachart, donut 3d and pie chart, treemap, leafletjs, bubble and bubble 3D, scatter chart and heatmap	Second step in the data visualization (Controllet)	See Sec. 11.5
36	Step by step	A wizard allows to create and share visualization of a chosen dataset simply steps	Data visualization (Controllet)	See Sec. 11.6
37	Search box for chart	Visualization of dataset with the search of a chart	Second step in the data visualization (Controllet): on the left top corner	See Sec. 11.7
38	Title and comments	Customize a chart with title or description	Second step in the data visualization (Controllet): "Base info" on the center top	See Sec. 11.8
39	A preview of selected chart	A preview before sharing the dataset visualization	Second step in the data visualization (Controllet): on the right top corner	See Sec. 11.9
40	Explore datasets with an interactive navigation	Select a dataset from an interactive tool for navigation, like, for example, a treemap built dynamically	First step in the data visualization (Controllet)	See Sec. 11.10
	<b>Personal reflections (My space)</b>			
41	Insert a dataset visualization into my space	A enlarged view of the dataset	On the right bottom corner of a tile	See Sec. 3.5.1
42	Take and insert notes into my space	A private space of an user	On the right bottom corner in "My space"	See Sec. 3.5.2
43	Take and insert a link into my space	A private space of an user	On the right bottom corner in "My space"	See Sec. 3.5.3
	<b>Open data access</b>			
44	Associated dataset provider	Each SPOD is associated with a set of Dataset providers (TET, CKAN, OpenDataSoft, etc.) that is easily browsable and immediately accessible (and seachable) when inserting a new visualization	Top menu bar /Tab: Open wall and Controllet (dataset visualization, first step)	See Sec. 7, 11
45	Easy navigation of associated dataset provider (treemap)	Graphical easy-to-use navigation of all the associated providers at once, directly from SPOD	Top menu bar /Tab: Open wall and Controllet (dataset visualization, first step)	See Sec. 7, 11



	<b>Support to user</b>			
46	Manual easily accessible	Manual is easily accessible within SPOD	Bottom menu bar	See Sec. 13.1
47	Video instructions on YouTube easily accessible	Videos is easily accessible within SPOD	Bottom menu bar	See Sec. 13.2
48	Nationalisation in each language (EN, FR, IT, NL)	Selectable by the user	Top menu bar (right)	See Sec. 16.2.1
49	Context-sensitive help	Tooltip or brief explanation to define page or widget of UI (e.g., button etc...), precisely how to interact with the control/page in question	In all the platform	See Sec. 13.3
	<b>Support to management and integration</b>			
50	Integration with CKAN platform	Integration with CKAN portal	In each post with dataset visualization	See Sec. 8
51	Integration UltraClarity	Integration with UltraClarity portal	In each post with dataset visualization	See Sec. 9
52	Roles	Actions are governed by the role and by access policies for each action (like administrators are the only users who can manage the platform)	Administrator panel. Implemented but not used by features: currently only administrator role is recognized, it is possible to create new roles but they are not used in the SPOD alpha and will be in Beta	See Sec. 17.1.2
53	Customizing the "What's new" page with widgets	The page the shows the current activities can be customized providing extra information about what is happening in the SPOD	Administrator panel. Currently the widgets available are few (text, RSS feed, slideshow, etc.) and can be useful to improve the communication between the citizens and the PA. Further widgets will be added in Beta, regarding the Agora, public rooms, etc.	See Sec 17.1.1. To be done on the manual in the administrator section, stating explicitly that "This section is just a preliminary short version of a separate "Administrator manual" that will be available for Beta.
54	New or customized page in SPOD	The administrator can help users to make easy-to-use interface in SPOD, adding or writing a new page	Administrator panel	See Sec. 17.1.3
55	Adding multiple languages	The administrator can upload a new language to change in SPOD	Administrator panel	See Sec. 17.1.4

## 6 DEVELOPMENT, DEPLOYMENT AND TESTING

---

### 6.1 AGILE METHODOLOGY

Agile is a methodology that suggests continuous iteration of developing and testing throughout the development software life cycle of a project [39].

Agile Software Development (ASD) is a theory put to paper (“Agile Manifesto” published in 2001) [39] in that it was published by seventeen software developers to discuss lightweight development methods based on 12 principle:

- client satisfaction, through rapid delivery of usable software
- meeting the specifications, even if it happens late in the development
- frequent delivery of usable software (weekly)
- the usable software is the main measure of progress
- sustained development, able to keep a steady rhythm
- cooperation between developers and clients
- face-to-face cooperation is the best way to communicate
- projects are built by motivated and credible persons
- simplicity
- individually organized teams
- adaptation to changing circumstances
- permanent attention to excellent technique and good design.

These values help to develop good software and encourage collaboration. Some agile software methodologies are: Adaptive Software Development (ASD), Feature Driven Development (FDD), Crystal Clear, Dynamic Software Development Method (DSDM), Rapid Application Development (RAD), SCRUM, Extreme Programming (XP) and Rational Unify Process (RUP).

In the ASD, a short iterative cycle provides an opportunity for rapid and motivating software process improvement. In the Agile development, the aim of a small team is to use a continuous improvement of design and testing based on feedback to reach the high quality of software.

#### 6.1.1 AGILE METHODOLOGY IN SPOD

SPOD project has followed some guideline of the agile methodology:

- Requirements received from D2.1, D2.4, D3.1, D4.1 and DoW
- Frequent delivery of usable software (two/three weeks)
- Incremental delivery (i.e., feedbacks received from Pilots)
- Testing during development with Pilot of the project receiving feedbacks

All comments, suggestions and ideas for improvements of SPOD (i.e., feedbacks from Pilots) have been planned in a bug-tracking system (e.g., Taiga, a project management platform for agile developers) and for each version of the SPOD a manual and video on YouTube have been released.

#### **Frequent delivery of usable software (two/three weeks)**

Instead of delivering the entire software at the end of a release (31th January 2016), the work is divided into smaller amounts of delivered requirements, which are taken from all requirements in order of importance.

For each release the version numbering with:

- Odd-numbered versions are “development”
- Even-numbered versions are “stable” (i.e., bugs, feedbacks etc)

Every release is characterized with a list of “features” (see the table below) and a list of the versions planned from October 2015 until January 2016:

- Version 0.1, “Antares” on October, 2<sup>nd</sup> 2015
- Version 0.2, “Betelgeuse” on October, 16<sup>th</sup> 2015
- Version 0.3, “Cheleb” on November, 6<sup>th</sup> 2015
- Version 0.4, “Diadem” on November, 20<sup>th</sup> 2015
- Version 0.5, “Electra” on December, 21<sup>st</sup> 2015
- Version 0.6, “Release Candidate” on January, 22<sup>th</sup> 2016
- Version 1.0 “Alpha Centauri” 31<sup>st</sup> January 2016

The features of each release are explained in the Table 10.

Release	Features
SPOD 0.1 “Antares”	<ul style="list-style-type: none"> <li>• Social Network <ul style="list-style-type: none"> <li>○ OpenWall: actually static information is shown, with the Twitter posts of ROUTE-TO-PA shown. It will show a summary of activities (topics of most active discussions, most used datasets, etc.), hiding personal info (usernames, etc.): it will be shown to everybody (not registered users, also).</li> <li>○ What's new: Everything (e.g., posts, likes etc...) is happening in the SPOD, only for logged users.</li> <li>○ My friends' activities: only for logged users, shows the friends activities.</li> <li>○ Public Room: In the draft version, it is like a forum. Most probably will be changed to something very different in the next versions</li> <li>○ Private Room (empty).</li> </ul> </li> </ul>
SPOD 0.2 “Betelgeuse”	<ul style="list-style-type: none"> <li>• Open Source SPOD project: <ul style="list-style-type: none"> <li>○ The source code of the SPOD project makes available DEEP, DEEPCIENT and COMPONENTS project in the GitLab, an open source software.</li> <li>○ <u>D</u>ata<u>l</u>et-<u>E</u>cology <u>S</u>ystem <u>P</u>rovider (DEEP) component provides a list of available datalets is Open Source Software with its source code made available with an MIT license.</li> <li>○ Full documentation (each component datalets and controllets), demo, examples are available on <a href="http://deep.routetopa.eu">deep.routetopa.eu</a>.</li> <li>○ Datalet generator: a static page allows to copy and paste the HTML code to include the datalet statically into any HTML editor. Full</li> </ul> </li> </ul>

	<p>compatibility with the most popular CMSs ad WordPress and Tumblr.</p> <ul style="list-style-type: none"> <li>• open data interoperability (e.g., Issy and Enigma, open data commercial portal of the USA ).</li> <li>• Customizing Charts ( title of the chart, X and Y axes names etc...)</li> <li>• Import all datasets with categories:</li> </ul> <p>An “Admin user” of SPOD platform can import all the datasets with categories from any address of a CKAN or “OpenDataSoft” portals.</p> <p>Reporting (e.g., Taiga.io is an Open Source project management)</p>
SPOD 0.3 “Cheleb”	<ul style="list-style-type: none"> <li>• Private Room: now citizens can build a private library of items to be later shared on the platform, including Datasets visualizations, URL and text notes, with a preview and the possibility to search and comment each item.</li> <li>• Associated dataset providers: now it is possible to list a set of associated open data portals to be queried, including compatibility with <ul style="list-style-type: none"> <li>○ Transparency-Enhancing Toolset (TET) <ul style="list-style-type: none"> <li>▪ CKAN</li> </ul> </li> <li>○ OpenDataSoft</li> </ul> </li> <li>• New navigation tool in the “Open Wall”: anybody (i.e. also not logged users ) can interactively navigate through a treemap into the associated datasets providers, grouped by “Category” or by “Association”</li> <li>• Improved User Interface for the dataset visualization</li> <li>• New datalets (Pie chart, Donut 3d Pie chart, Area Chart, 3D Bubble chart)</li> </ul>
SPOD 0.4 “Diadem”	<ul style="list-style-type: none"> <li>• First communication between TET and SPOD : now from every datalet shown from TET (and any CKAN), it is possible to connect back to “data source” and to the information page of the datasets.</li> <li>• Compatibility with Ortelio’s Ultraclarity engine: now it is possible to show data coming from a query to the Ortelio Ultraclarity engine.</li> <li>• Streamlined User Interface: TreeMap search for selecting the datasets to visualise; filter with CKAN dataset to display data into understandable information and multiple tables display the data for other open data providers (e.g., OpenDataSoft)</li> </ul>
SPOD 0.5 “Electra”	<ul style="list-style-type: none"> <li>• A new open space of discussion : Agora is an open space for discussion, that is, it is set of (public) rooms ; Public Room features: joint visualization and synchronization between enhanced thread chat for input and graphics for more information <ul style="list-style-type: none"> <li>○ Threaded chat with contributions with “Opinions” (👍, 👎, 🗣️)</li> <li>○ Graph visualization: a comment-based graph, an user-based graph, datalet-based graph and opinion-based graph, synchronized with the threaded chat.</li> <li>○ Support for dashboard switch between “What's new” and “My friends' activities”</li> </ul> </li> </ul>

	<p>For all graphs the level is represented with a color: Red indicates the first level, Blue indicates for the second Level and Gray for the third level.</p> <ul style="list-style-type: none"> <li>• Communication between OpendataSoft and SPOD: now from every datalet shown from Open data portal of Issy (<a href="https://data.issy.com/">https://data.issy.com/</a>) (and any OpenDataSoft portal, e.g., Open data portal of Île-de-France), it is possible to connect back to “data source” and to the information page of the datasets.</li> <li>• Streamlined User Interface: an enhanced visualization of the open data (e.g., search etc...)</li> <li>• My space: the enhanced User Interface for cards of notes and links.</li> <li>• New datalets of Ancitel: bubble, scatter charts and heatmap.</li> </ul>
SPOD 0.6 “Release Candidate”	Fixed bugs and known issues.
SPOD 1.0 “Alpha Centauri”	

Table 10: Features for each release

### 6.1.2 INCREMENTAL DELIVERY (FEEDBACK)

The software is developed in increments with the Pilots specifying the bugs and further requirement to be included in the next release.

Each release has received feedback from partners (e.g., Prato, CNRS, Groningen, Den Haag, Dublin) and the list of highlighted bugs by the Pilot is been managed on Taiga.

Pilots for feedback	Feedbacks	Release	Date
Prato	Suggestions to improve UI, list of strings to translate	Antares 0.1	October, 7th 2015
Groningen	User Interface, translation clarification of pages, list of strings to translate	Antares 0.1	October, 13th 2015
Prato	List of strings to translate, User Interface, UI suggestions	Betelgeuse 0.2	October, 29th 2015
Groningen	Functionalities, data visualization, list of strings to translate, Forum, suggestions UI	Betelgeuse 0.2	October, 28th 2015
Prato	Data visualization, bugs on Charts, Private Space, Public Space and Open wall with tree map	Cheleb 0.3	November 13 <sup>th</sup> 2015
Den Haag	UI suggestions, Public Room, Data visualization and Layout issues, list of strings to translate	Cheleb 0.3	November 19 <sup>th</sup> 2015
Dublin	UI in the Data visualization,	Cheleb 0.3	November 4 <sup>th</sup> 2015
Groningen	Functionality in home page, suggestions for UI of data visualization, list of strings to translate	Cheleb 0.3	November 19 <sup>th</sup> 2015

CNRS	List of strings to translate, Members, suggestions on the What's new, Public Room, Public Space and general remarks	Diadem 0.4	November 26 <sup>th</sup> 2015
Groningen	Treemap in Data visualization, suggestions on UI and layout of SPOD	Diadem 0.4	November 26 <sup>th</sup> 2015
Den Haag	Public Room, data visualization , UI suggestions	Diadem 0.4	December 16 <sup>th</sup> 2015
Prato	Translations UI suggestions	Diadem 0.4	December 14 <sup>th</sup> 2015
Groningen	Suggestions on UI for Agora, Public Room and open data controllet	Electra 0.5	December 15, 2015
Den Haag	Suggestions on UI for Agora and Public Room	Electra 0.5	January 5 <sup>th</sup> 2016
Prato	Strings to translate, suggestions on UI for Agora and Public Room	Electra 0.5	January 5 <sup>th</sup> 2016

Table 11: Feedbacks received from each partner

## 6.2 AUTOMATED TESTING

Software Testing is a fundamental activity to assess the software quality. Software testing executes a system or part of it with the intent to find a bug. With the advent of the Web 2.0 and the heterogeneities of configurations (i.e., browsers, resolutions, and so on), it is crucial to perform functional testing through the graphical user interface functionalities. When executed manually these tests are repetitive, time-consuming and error-prone activities, because the tester must use the system and follow the test step directly on the user interface. Automated tests aim to reduce as much as possible the testing effort by creating software scripts, which can repeatedly run many times. Therefore, they can reduce the effort and the costs to have someone sitting to his/her computer and manually execute the tests. In addition, the running of automated tests in parallel can stress the system and simulate its performances under high load obtaining stress tests data. Another advantage in introducing the automated tests is the support of the Continuous Integration (CI). Instead to merge the changes at the end of the feature implementation, enhancement or bug fixing; Continuous Integration tries to frequently integrate the changes avoiding the “*integration hell*”. Automated tests can faster the frequently integrations.

In particular, regression tests seek to uncover software bugs in existing functionalities after changes, such as enhancements or bug fixing. The aim is to be sure that changes to the software did not broke existing functionalities. Regression tests are sets of scripts, which contain the test cases instructions, and assertions to determine the test outcome, its failure or success. In addition, the tests can measure quantitative data during the execution, for instance the total time to execute the test script. The idea is that every time a developer changes part of the code, he/she must run all the tests again and be sure that everything still work after the source code improvements. In an Agile development methodology, the writing of tests and their executions is a very important step. In this way, automated tests can test multiple configurations, with different browsers (i.e., Chrome, Firefox, etc.) and common screen resolutions. Browsers and screen resolutions together generate a high number of configurations to test at each SPOD release, so automated testing in an Agile development process become a must.

The testing architecture, described in this section, has been designed and used to test SPOD v0.3 and SPOD v0.4. In particular, the automatically tested functionalities are: *login*, *controllet wizard to create a visualization* (map and a chart visualizations), *MySpace (or private room)* and *the posting of data visualizations*. In the MySpace page, the test created and deleted cards, in particular the card for the links, the text and the visualization of datasets. Tests have been executed on Windows 7 and Windows 8.1 using the Chrome and Firefox browsers and with multiple resolutions (i.e., 1366x768, 1600x900).

Stress tests have been performed using these automated test scripts. For example, the posting of data visualization has been tested simulating the simultaneous posting of one hundred visualizations for 1, 2, and 4 users. Test scripts written in Java automate the test cases. Test scripts execute the test steps.

Functional testing directly and strongly depends on the system GUI, and test scripts obviously are specifically wrote for a particular version of the SPOD GUI. Adopting an Agile methodology, since the beginning of a project, allows the rapid prototyping with constant and continuous feedback from the users. The user involving since the beginning of the project has the advantage and the opportunity to change the system immediately, responding to the user requirements and needs. Of course, at the beginning of a project with short and continuous iterations of planning, design and development, the GUI can heavily change in few weeks and this impacts on the test scripts that must be rewritten. Considering this, the automated test has not been applied in all SPOD versions but the architecture will be used to test all future SPOD versions.

#### 6.2.1 TEST CASES & TEST SCRIPTS

In the Agile development there is a specific phase called test planning dedicated to design test cases. Each test case specifies the steps that a tester must execute step-by-step to test a particular software functionality. One of the test case design guideline is to keep them as simple as possible. Each test case has a specific section to indicate the test outcomes after its execution.

A test case usually has the following information:

- A test case ID to univocally identify and refer it;
- A brief test case description;
- The test case steps to execute in the specified order;

In order to automatize a test case, its steps must be implemented within a test script that reproduces the actions on the system GUI. Functional test cases heavily depends on the graphical user interface (GUI). This means that every time the system GUI changes (e.g., adding of a button, link and so on), consequently the test case steps change. This has a direct impact on the test script. Therefore, every time the GUI changes, the test case specification must be re-designed and the test script must be updated.

The test case information are summarised in a table, which reports all the essential data to run the test and the information on its execution and outcomes. The following is an example of test case, and this format is the base template for all the test cases specification:

<b>Test Case ID – Title:</b> TC04 – Insert a text card in the private room				
<b>Project Version:</b> SPOD “Cheleb” (0.3)				
<b>Description:</b> <i>The user access to her/his private room and creates a new card, inserting text as notes.</i>				
<b>Test type:</b> Functional Testing			<b>Test Designed by:</b> Donato Pirozzi	
<b>Test priority:</b> High			<b>Test Designed date:</b> 2015/11/16	
<b>Pre-Conditions:</b> - User has logged in SPOD				
<b>Post-Conditions</b> - The created card is in the private room				
Step	Test Step	Test Data	Expected result	Actual Result
1	spod.routetopa.eu/spodpr		Private Room page loading	-
2	Click on the “+” Button		It shows the button to add a card	-

3	Click on the “Pencil Button”		It shows the dialog with Card details.	-	
4	Provide Title	CardTitle with a random number code	The preview updates	-	
5	Provide Comment	CardComment	The preview updates	-	
6	Click on “+” Post Button		The card is in the private room	-	
Executions					
Execution Date	Number of Executions	Number Concurrent Users	Configuration	Test Time	Status
2015/11/13	1	1	Windows 8.1, Chrome 46.0.2490.80 m, 1366x768	45s	Ok
2015/11/13	100 cards	1	Windows 8.1, Chrome 46.0.2490.80 m, 1366x768	2014s	Ok
2015/11/19	100 cards	2	Windows 8.1, Chrome 46.0.2490.80 m, 1366x768	2050s	Ok
2015/11/13	1	1	Windows 8.1, Chrome 46.0.2490.80 m, 1600x900	45s	Ok
2015/11/13	1	1	Windows 8.1, Firefox 41.0.2, 1366x768	45s	Ok
2015/11/13	1	1	Windows 8.1, Firefox 41.0.2, 1600x900	45s	Ok

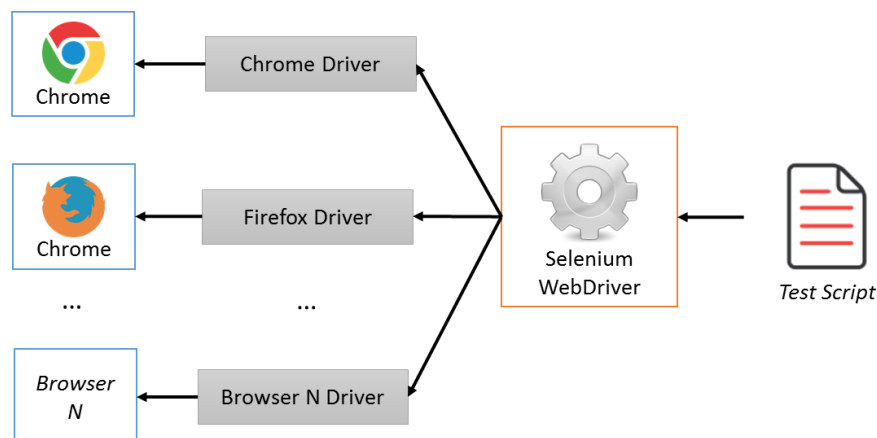
Each test case is translated in a test script, which will automatically execute all the steps and will report the results. Each step in the test script simulates the user actions and interactions (i.e., clicks, types, drag-and-drops and so on). For instance, it simulates the click on a link or fill of a form.

#### 6.2.2 AUTOMATED TESTING ARCHITECTURE

The testing architecture is mainly based on Selenium framework. In particular, Selenium WebDriver enables the direct communication with the browser supporting also Ajax calls.

The automated testing architecture has the following components:

- A test script written in Java contains the test case instructions to execute;
- The Selenium WebDriver receives the test case instructions and send them to the Browser Driver
- The browser driver specific for each browser translates the WebDriver commands and executes them in the specific browser.



The test script uses the Java language. The script specifies all the test case steps and the assertions to check the tests failures or successes. Test script configuration specifies also the browser and resolution to use; in this way, the WebDriver can load the browser and set up the resolution. The test script through the web driver executes each step simulating the user actions. In addition, it measures the total time to complete the test and the number of times that it has executed.

Therefore, the execution of a test script follows these steps:

1. Run of the test script and setup of the testing environment



2. Run of the WebDriver that will issue the commands to the browser
3. Load of the browser driver and selected browser
4. Screen resolution setting
5. The script sends a command to the WebDriver which forwards it to the browser through the driver
6. The browser executes the command
7. The WebDriver returns a feedback to the test script on the step failure or success
8. The script execution continuous to send command to the browser and perform assertions

The test execution stops when all its steps have executed until the end or a step fails.

### 6.3 DEPLOYMENT OF THE ARCHITECTURE

The details of hardware/software underlying architecture where SPODs are executed, are explained in the deliverable D1.1: “Setup of the Integrated Communication Platform”, section 1.6 “Hardware architecture”, available in the final format at the [www.routetopa.eu/public-deliverables/](http://www.routetopa.eu/public-deliverables/).

Usability testing

### 6.4 USABILITY: CRITERIA AND ACTIVITIES

The ultimate aim of the usability tests is to analyse the processes by which participants (who share values and goals) will establish relationship and create groups in computer-mediated interaction. In the alpha version only limited activities about usability have been conducted, as the testing was mostly on the functional and non-functional requirements. We report here on this limited activities that were already conducted in the first year and planned for the second year at the Issy-Les-Moulineaux pilot (more usability tests are planned within the other pilots in the second and third year).

Four experiments are planned, with the aim of analysing criteria that have impact on the intelligibility and the use of the platform.

In a first instance, participants from CNRS and IssyMedia have tested the general structure of the SPOD/TET in October & November 2015.

Researchers focused on the characteristics of the activity (Leplat, 1997, 2006). In this way, the cognitive walkthrough method (Wharton, 1992; Wharton, Rieman, Lewis, Polson, 1994) was used to distinguish the different steps necessary to perform a series of specific tasks (e.g. connect to profile page, upload data from external website).

We have also carried out an analysis of errors (Leplat, 1999). Thus, the deviation between the goal of the platform (e.g.: interact on discussion spaces, visualise data in graphical format) and the tasks performed by users was studied.

The same experiment will be planned, with beta-testers, in June 2016 and January 2017.

In a second instance, a usage scenario will be proposed to two groups of undergraduate students in February 2016. In this way they will ask to manipulate open data provided by the CNRS team with the aim of elaborating a collective solution to a practical issue.

The first group will work in a face-to-face collaborative situation. The second group will work in a virtual collaborative situation by using computer-mediated communication (SPOD). Exchanges will be recorded online (on the platform) and offline (with a video camera). (Inter)actions will be analysed as described in the sections above (quality and qualities of collaboration).

In addition, the *FMECA (Failure Modes, Effects and Criticality Analysis) approach will be explored, in a top-down perspective, to eliminate potential failures from the system (Omdahl, 1988) before they reach the final users (entrepreneurs and PAs).*

In a third instance, a participatory design approach (Greenbaum, & Kyng, 1991) will be adopted. A group of young entrepreneurs (March 2016) and of a group of PAs from Paris Region (April, 2016) will be invited to evaluate the system. They will manipulate public data provided by pilot (Issy-les-Moulineaux) on the SPOD/TET. Their interactions with the system will be logged onto a server (e.g. connection time, numbers of clicks to carry out an action, fixation time on TET/SPOD). They will ask to verbalise their needs in terms of information, exchanges and functionalities.

In this way, the system will be improved with the aim of facilitating its re-use (without specific prior knowledge) by a large proportion of target users and producers.

Finally, two groups of stakeholders (young entrepreneurs and PAs) will ask to use regularly the platform during 18 months. The CNRS team will study the community building process, the simultaneous progression of activity of each group, the knowledge co-elaboration and collective self-efficacy.

More precisely, the researchers will focus on the different forms of collaboration (information sharing, toward shared understanding and co-creation) and on their evolution. They will analyse the commitment of the participants in the collaboration process and intrinsically evolution of their role.

Qualitative (coding scheme) and quantitative (code frequencies, distributions, patterns) analysis of interactions will be performed, based on dialogic functions and epistemic contents of the exchanges.

Finally, a social Network analysis (Viegas and Smith, 2004; Welser et al., 2007, 2011) will be carried out in order to study the structure of interactions (who communicates with whom?).

## 7 CROSS-CUTTING DESIGN CONCERNS

---

### 7.1 PRIVACY

EU established that: *“All the research and innovation activities carried out under Horizon 2020 shall comply with ethical principles and relevant national, Union and international legislation, including the Charter of Fundamental Rights of the European Union [40] (in terms of Dignity, Freedoms, Equality, Solidarity, Citizens' Rights, and Justice) and the European Convention on Human Rights and its Supplementary Protocols” [41].*

Specifically, ROUTE-TO-PA related ethics deals with the usage of data retrieved from social media services (i.e., OXwall), as well as, discussion forums, blogs, and so on.

Data can be classified in personal data and sensitive data. In general, examples of personal data include: name, address, date of birth, bank account details, credit card number, social security number, phone number, employment history, geographical data (country, city, zip code), education level, gender, and so on. Examples of sensitive data include: race, ethnicity, religious beliefs, philosophical or other beliefs, political opinions,

membership of parties, trade unions, associations or organizations in the religious, philosophical, political or trade-union sectors, health and sex life.

Ethical issues may arise as a result of collecting and processing personal and sensitive information in the project. Processing broadly means *“any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction”* [42].

To protect the privacy of individuals and, specifically, to avoid any inappropriate use of their personal data, mechanisms of privacy protection should be designed (according to the privacy-by-design principle) in order to avoid identification, that mainly occurs when data are linked or matched with data available elsewhere, in other databases, etc. Specifically, a “Personally identifiable information” (PII) can be defined as an information which can be used to distinguish or trace an individual’s identity either alone or when combined with other available information. Given the popularity of Online Social Networks (OSN) a vast amount of personal information on the Internet is available for free. Research in this field has shown that it is possible for third-parties to link PII, leaked via OSNs or while browsing Web pages [43], raising therefore critical issues about privacy [44].

Finally, under the Data Protection Directive [42], collecting and processing the personal data of individuals is legitimate only in some well regulated circumstances laid down by Article 7 of the Directive. One of circumstances is about the informed consent, i.e., when the individual has unambiguously given his/her consent, after being adequately informed.

ROUTE-TO-PA relates to the following issues:

1. Usage of data retrieved from Oxwall as well as from discussion on forums, blogs, wikis, and so on
  - Personal data retrieved through accounts (i.e., full name, DOB, gender)
  - Personal and sensitive information from discussions (religious beliefs, philosophical or other beliefs, political opinions)
2. Collecting and processing of personal data
  - Both static identity-related information and dynamic information such as online presence and physical locations

In addition, a third issue is about the sharing of data with third entities for future use research.

With regard to activities involving human subjects, ROUTE-TO-PA does not plan any physical intervention on the study participants nor using any kind of invasive techniques. For all user workshops participants have been provided with material (e.g., consent forms, questionnaires etc.).

To adhere to the EU Data Protection Directive, the following document were requested to each project’s participant:

- Request letters for Ethical Approvals for the collection of personal data
  - Submitted for review to the competent entities (University Data Protection Officer, local Ethics Committee, etc.)
- Informed Consent Forms or other consent documents

- Provided by each research partner (i.e., Centre National de la Recherche Scientifique - CNRS, Wise & Munro, National University of Ireland Galway, Universiteit Utrecht - University of Utrecht)

All participants agreed to the following procedure for managing the use of personal data:

1. All data to be used in ROUTE-TO-PA is publically available, given their nature of OPEN DATA.
2. Personal data are stored in specific databases. In the whole project, the aim to is make as simple as possible for users the task of checking what is visible to the public, seen by search engines, kept private, and permanently deleted.
3. We do not sell, trade, share or otherwise transfer to outside parties (third parties) personal and sensitive information.
4. Data could be shared only for future research (but properly anonymized so that it is impossible to make any linking) but, anyway, nobody will make money on it. In that case the partner requesting data will need to forward an official personal data request to the ROUTE-TO-PA Ethics Board for approval. In the case of approval, data can be used only under the condition that approved consent forms have been received from the original owners.
5. It is possible that some personal data may be required for specific reasons related to project. Such personal data could be geographical data, education level, gender, and specific orientations/preferences (i.e., political, economical, intellectual, and so on). We would like to emphasize that such personal information will not be linked with any other data type, in order to avoid identification and ensure full privacy of the involved citizens.
6. Accounts built for citizens will not connected with other social networks (i.e., Facebook), in order to keep the discussion inside ROUTE-TO-PA platform. Conversely, posting could be shared on other social platform, and specifically, on Facebook and Twitter social networks, to foster the engagement of new users.

The defined and approved procedure has been translated in a Privacy Policy, (<http://spod.routetopa.eu/privacy-policy>) available for the alpha version of the released software.

## 7.2 RRI-ICT

Given the impact on end-users and on society, it is important to address Responsible Research and Innovation (RRI) criteria about the work described in this document.

RRI has acquired prominence by its status as a cross cutting issue of the EU framework program for R&I, Horizon 2020 [45]. In their document, RRI is defined as the action of all the societal actors working together during research and innovation process. The objective should be trying to align process (and therefore outcomes) to the values, needs and expectation of our society.

In 2014, a group of experts was appointed by the European Commission “to identify and propose indicators and other effective means to monitor and assess the impacts of Responsible Research and innovation (RRI) initiatives, and evaluate their performance in relation to general and specific RRI objectives [45]. During their work, influenced by the Rome Declaration on RRI in Europe of Nov 21<sup>st</sup> 2014, the experts suggested a list of 8 criteria to be monitored. These criteria will be addressed below for this deliverable.

First of all, in general, being our deliverable the terminal of the requirements and modeling activities developed in D2.4 and D3.1 we inherit the considerations for the criteria, that have been discussed in deliverable “D2.4 Requirement Specification and Use Case Models for TET and SPOD Subsystem”, Section 7 and in deliverable “D3.1 Societal Activity Model of Open Data Use”, Section 8.

#### **1. Public engagement**

Besides the public engagement during the empirical work of requirements elicitation, where public administrators, active citizens, researchers and data journalists, social organizations and entrepreneurs were involved, the activities of D4.1 were strongly based on continuous testing of the versions as they were developed, although in a preliminary stage without direct involvement of users but in a controlled environment where, according to the agile methodologies, the researchers and some users (in Prato, for example) were playing the role of the citizens and public administrators.

#### **2. Gender equality**

Nothing to describe.

#### **3. Science education**

The partner Università di Salerno has participated to dissemination activities with teachers of secondary schools in order to show technologies of the (coming) Social Platform for Open data. Also, activities about applications of SPOD to university courses in Salerno, about environmental pollution, are.

#### **4. Open Access**

In general, as with the other “Public” deliverables, this deliverable will be available in its final format at the [www.routetopa.eu/public-deliverables/](http://www.routetopa.eu/public-deliverables/). Publications related to these results will also be made available through the various open access archives of the respective research and academic institutions.

The whole SPOD platform with all developed plugins and the DEEP platform are freely accessible online at <http://service.routetopa.eu:7840> . The DEEP is also accessible and usable (with all the documentation) at <http://deep.routetopa.eu> .

#### **5. Ethics**

Nothing to describe.

#### **6. Governance**

The project Ethical advisor will review all processes involved in the engagement and collection of information from participants in workshops and other activities related to user stories elicitation and requirements specification.

#### **7. Sustainability**

Part of the ROUTE-TO-PA project is also to conduct a market analysis, thereby identifying sustainable business models, including the ones that will create value through technology transfer to other sectors.

#### **8. Social Justice/Inclusion**

SPOD supports accessibility features so that disabled users can personalize their interface to take advantage of the features. The choice of the Design Language as Material Design, which offers a rich alternative for usability and accessibility, in particular for vision impaired and elderly, was particularly helpful in this

context. Concrete examples, already addressed in the alpha version, include general accessibility (in terms of color vision impairment) and inter-generational differences (in terms of reachability as well as participation of older citizens).

## 8 RELEVANT PUBLICATIONS

---

The activities that led to this deliverable have been partially reported in the following publications:

- G. Cordasco, D. Malandrino, P. Palmieri, A. Petta, D. Pirozzi, V. Scarano, L. Serra, C. Spagnuolo, L. Vicidomini, "An extensible architecture for an ecosystem of visualization web-components for Open Data" in Share-PSI 2.0 Workshop, "Maximizing interoperability – core vocabularies, location-aware data and more", Berlin, Germany November 24-25, 2015.
- V. Scarano, D. Malandrino, M. Baker, F. Detienne, J. Andriessen, M. Pardijs, A. Ojo, L. Porwol, P. Szufel, B. Kaminski, A. Meijer, E. Ruijter, J. Forrester, G. Clementino, I. Trochidis, V. Banos, M. Andriessen, J. P. van de Klashorst, E. Ruijter, J. Forrester, G. Clementino, I. Trochidis, V. Banos, M. Andriessen, J. P. van de Klashorst, P. Riordan, R. Farrell, P. Boscolo, E. Palmisano, S. van der Waal, J. Gray, M. Satta, and E. Legale: "ROUTE-TO-PA H2020 Project: "Raising Open and User-friendly Transparency-Enabling Technologies for Public Administrations", In the 6th International Conference on e-Democracy, Athens, Greece, December 10-11, 2015.
- D. Malandrino, I. Manno, P. Palmieri, A. Petta, D. Pirozzi, V. Scarano, L. Serra, C. Spagnuolo, L. Vicidomini. "An Architecture for Social Sharing and Collaboration around Open Data Visualizations" In Poster Proc. of the 19th ACM conference on "Computer-Supported Cooperative Work and Social Computing (CSCW 2016)", San Francisco, USA. February 27–March 2, 2016 (accepted for publication)
- Vittorio Scarano, Gennaro Cordasco, Nicola Lettieri, Delfina Malandrino, Ilaria Manno, Giuseppina Palmieri, Andrea Petta, Donato Pirozzi, Diletta Rizzolo, Luigi Serra, Carmine Spagnuolo, Luca Vicidomini, "Fostering transparency and participation in the data-based society: a Sustainable Architecture for a Social Platform for Open Data". In Conference "Challenges e-2015" Conference Vilnius, Lithuania, November 25th- 26th, 2015.

## BIBLIOGRAPHY

- [1] Bertot J.C., Jaeger P.T. & Grimes Justin M. (2010). Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies, in *Government Information Quarterly*, Volume 27, Issue 3, July, pp. 264-2.
- [2] V. Scarano, D. Malandrino, M. Baker, F. Detienne, J. Andriessen, M. Pardijs, A. Ojo, L. Porwol, P. Szufel, B. Kaminski, A. Meijer, E. Ruijter, J. Forrester, G. Clementino, I. Trochidis, V. Banos, M. Andriessen, J. P. van de Klashorst, E. Ruijter, J. Forrester, G. Clementino, I. Trochidis, V. Banos, M. Andriessen, J. P. van de Klashorst, P. Riordan, R. Farrell, P. Boscolo, E. Palmisano, S. van der Waal, J. Gray, M. Satta, and E Legale: "ROUTE-TO-PA H2020 Project: Raising Open and User-friendly Transparency-Enabling Technologies for Public Administrations", In the 6th International Conference on e-Democracy, Athens, Greece December 10.
- [3] ISO/IEC/IEEE 24765:2010(E), "Systems and software engineering -- Vocabulary," in ISO/IEC/IEEE 24765:2010(E), pp.1-418, Dec. 15 2010, doi: 10.1109/IEEESTD.2010.5733835.
- [4] S. (. Buckingham Shum, "Cohere: Towards Web 2.0 argumentation". In P. Besnard, S. Doutre, & A. Hunter (Eds.), *Proceedings COMMA'08: 2nd International Conference on Computational Models of Argument* (pp. 97–108). Toulouse, France: IOS Press.
- [5] S. Faridani, E. Bitton, K. Ryokai, and K. Goldberg. 2010. Opinion space: a scalable tool for browsing online comments. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 1175– 1184..
- [6] J. Hullman, N. Diakopoulos, E. Momeni, and E. Adar. 2015. Content, Context, and Critique: Commenting on a Data Visualization Blog. In *Proc. of the 18th Conf. on Computer Supported Cooperative Work & Social Computing*. ACM, 1170–1175..
- [7] Iandoli, Luca, Ivana Quinto, Anna De Liddo, and Simon Buckingham Shum. 2014. "Socially Augmented Argumentation Tools: Rationale, Design and Evaluation of a Debate Dashboard." *International Journal of Human-Computer Studies* 72, no. 3 (March 2014): 298–319.
- [8] Iandoli, L., Quinto, I., De Liddo, A., & Shum, S.B. (2014b). "A debate dashboard to enhance online knowledge Sharing". *VINE*, 42(1), 67–93..
- [9] L. Iandoli, I. Quinto, A. De Liddo, and S. Buckingham Shum. 2015. On Online Collaboration and Construction of Shared Knowledge: Assessing Mediation Capability in Computer Supported Argument Visualization Tools. *Journal of the Association for Information*.
- [10] Klein, Mark, and Gregorio Convertino. "A Roadmap for Open Innovation Systems." *Journal of Social Media for Organizations* 2, no. 1 (2015): 1..
- [11] Klein, M. Klein. 2010. Using metrics to enable large-scale deliberation. In *Collective intelligence in organizations: A workshop of the ACM Group 2010 Conf*. 103–233..
- [12] Klein, Mark. 2015. "A Critical Review of Crowd-Scale Online Deliberation Technologies." MIT Working Paper, August 2015.
- [13] M. Klein, and Gregorio Convertino. 2014. "An Embarrassment of Riches." *Communications of the ACM* 57, no. 11 (October 27, 2014): 40–42. doi:10.1145/2629560..
- [14] C. Project, <http://catalyst-fp7.eu/us/project/>.
- [15] Assembl, <http://assembl.org/>.
- [16] LiteMap, <https://litemap.net/>.
- [17] DebateHub, <https://debatehub.net/>.

- [18] Edgesense, <http://matera2019.edgesense.spazidigitali.com/>.
- [19] C. I. Dashboard., <https://cidashboard.net/>.
- [20] IdeaScale, <http://ideascale.com/product-tour/>.
- [21] MindJet, <https://www.mindjet.com/>.
- [22] 1. Whysaurus., <https://www.whysaurus.com/>.
- [23] Deliberatorium, <http://franc2.mit.edu:8000/ci/show-top>.
- [24] M. K. 2010., Using metrics to enable large-scale deliberation. In Collective intelligence in organizations: A workshop of the ACM Group 2010 Conf. 103–233..
- [25] Whysaurus, <https://www.whysaurus.com/>.
- [26] C. I. Dashboard, <https://cidashboard.net/>.
- [27] Edgesense, <http://matera2019.edgesense.spazidigitali.com/>.
- [28] Software Engineering --Software product Quality Requirements and Evaluation (SQuaRE) --System and software quality models" (<https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>).
- [29] K. Dulaney, Van L. Baker, Richard Marshall, Roberta Cozza, Tim Zimmerman "Predicts 2015: Mobile and Wireless." Gartner Group, 05 November 2014 <http://www.gartner.com/document/2900317>.
- [30] "Ericsson Mobility Report: on the pulse of the networked society", June 2015, <http://www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf>.
- [31] G. Cordasco, D. Malandrino, P. Palmieri, A. Petta, D. Pirozzi, V. Scarano, L. Serra, C. Spagnuolo, L. Viciomini, "An extensible architecture for an ecosystem of visualization web-components for Open Data", Share-PSI 2.0 Workshop, "Maximizing interoperability – core vocabularies, location-aware data and more", Berlin, Germany November 24-25.
- [32] D. Malandrino, I. Manno, P. Palmieri, A. Petta, D. Pirozzi, V. Scarano, L. Serra, C. Spagnuolo, L. Viciomini, "An Architecture for Social Sharing and Collaboration around Open Data Visualisations", In Poster Proc. of the 19th ACM conference on "Computer-Supported Cooperative Work and Social Computing (CSCW 2016)", San Francisco, USA. February 27–March 2, 2016.
- [33] <https://elgg.org/>.
- [34] <http://www.oxwall.org/>.
- [35] [http://www.w3.org/standards/techs/components#w3c\\_all](http://www.w3.org/standards/techs/components#w3c_all).
- [36] <https://angularjs.org/>.
- [37] X. B. V. Chan, , Shi Min S Goh and Ngiap Chuan Tan. Subjects with colour vision deficiency in the community: what do primary care physicians need to know?, Asia Pacific Family Medicine 2014.
- [38] <http://masonry.desandro.com/>.
- [39] Beck, Kent; et al. (2001). "Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June 2010..
- [40] CHARTER OF FUNDAMENTAL RIGHTS OF THE EUROPEAN UNION, <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:12012P/TXT>.
- [41] Charter of fundamental rights of the EU, [http://www.echr.coe.int/Documents/Convention\\_ENG.pdf](http://www.echr.coe.int/Documents/Convention_ENG.pdf).
- [42] Data protection directive, <http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:31995L0046>.
- [43] D. Malandrino e V. Scarano, «Privacy leakage on the Web: Diffusion and countermeasures,» *Computer Networks*, vol. 57, n. 14, pp. 1389-1286, 2013.



- [44] B. Krishnamurthy e C. Wills, «On the Leakage of Personally Identifiable Information Via Online Social Networks,» *SIGCOMM Comput. Commun. Rev.*, vol. 40, n. 1, pp. 112--117, 2010.
- [45] M. W. Bauer, Indicators for promoting and monitoring Responsible Research and Innovation, Report from the Expert Group on Policy Indicators for "Responsible Research and Innovation", 2015.
- [46] S. Few, "Data Visualization for Human Perception", 2002, Interaction Design Foundation.
- [47] <http://webcomponents.org/>, <http://webcomponents.org/>.
- [48] <https://www.polymer-project.org/>.
- [49] Horizon2020, «The EU Framework Programme for Research and Innovation,» 2015. [Online]. Available: <https://ec.europa.eu/programmes/horizon2020/en/h2020-section/responsible-research-innovation>.
- [50] Debatewise, <http://debatewise.org/>.
- [51] I. L., L., Quinto, I., De Liddo, A., & Shum, S.B. (2014b). "A debate dashboard to enhance online knowledge Sharing". VINE, 42(1), 67–93..