DELIVERABLE

D3.7 Policy experimentation and functional design 3

Project Acronym:	PoliVisu
Project title:	Policy Development based on Advanced Geospatial Data Analytics and Visualisation
Grant Agreement No.	769608
Website:	www.polivisu.eu
Contact:	info@polivisu.eu
Version:	1.0
Date:	August 8, 2019
Responsible Partner:	AIV
Contributing Partners:	GEOS, HSRS, INCO, P4All, ISSY
Reviewers:	Pavel Kogut (21C) Grazia Concilio (POLIMI) Stamatis Rapanaki (ATC) Jiri Bouchal (INCO) Bart Rosseau (GENT) Berdien De Roo(GEOS) Karel Jedlička (P4A)
Dissemination Level:	Public — X
	Confidential – only consortium members and European Commission Services

This project has received financial support from the European Union Horizon 2020 Programme under grant agreement no. 769608.

Revision History

Revision	Date	Author	Organization	Description
0.1	30/4/2019	Gert Vervaet	AIV	Initial draft, based on D3.6
0.2	5/6/2019	Gert Vervaet, Frank Maes, Tomas Reznik, Jeroen Saegeman, Pepijn Viaene, Berdien De Roo, Jiri Bouchal	AIV GEOS HSRS GEOS GEOS GEOS INCO	adding content
0.3	9/7/2019	Lieven Raes	AIV	Review
0.4	13/7/2019	Matteo Satta	ISSY	Contribution to 3.2.5
0.5	02/8/2019	Gert Vervaet	AIV	review
0.6	07/08/2019	Lieven Raes	AIV	review
1.0	08/08/2019	Lieven Raes	AIV	final version

Table of Contents

List of Figures

List of Tables

Executive Summary

Triggered by the iterative approach taken by Polivisu, this document describes the increased insights gathered during the second iteration of the project.

Extending the Toolbox to support more types of posts to enable pilots and policymakers to populate the Toolbox towards a knowledge base for policymakers is the first part of functional design. The Toolbox allows linking the PoliVisu policymaking and policy visualisation methodology to the practical pilot cases and visualisation tools.

Starting from the policy design cycle as defined in D3.5, the PoliVisu interactive visualisation tools were analysed, using a tailored methodology. Starting from the pilot plans and experiences, specific visualisation goals were determined. The next step was to create or obtain the right data sources, followed by preparing the data as "policy ready data". Both steps lead to the visualisation description (based on the functional and technical design). The results of these steps are part of the described visualisation status. According to the latest user testing, future improvements are listed-up for future implementation.

PoliVisu tested and experimented with many visualisation methods to visualise mobility-related data in a policymaking context. However, the used methods on itself are not entirely new; the context of using these techniques for dynamic visualisations are innovative. These dynamic visualisations allow users to select data, experiment with data and to elaborate policy problems, the existing policy impact or future policy impact when forecast and modelling techniques are used.

The described interactive visualisation tools are not a complete list, but several highly useful tools that are used in practice and will be more refined during the next testing cycles and iterations.
These iterations are essential to test the current and future visualisations in terms of user interface and their applicability for policymaking purposes.

In the next stage of PoliVisu, the described improvements will be built and tested. A number of the existing applications will be further tested. Special attention will go to further integrate the visualisation with policymaking initiatives in the pilot cities and testing with the defined target groups in each of the pilots.

1 Introduction

Based on the designs created in D3.6 where we explained the concept of the Toolbox as a knowledge base that gathers all information about case studies, dataset types, techniques, visualisation types, process steps, policy elements and the policy making process as defined in D3.5. And using the approach defined in that deliverable to start from visualisation goals strongly linked to real needs of policy makers and as such to policy elements that will benefit from the visualisations.

The described iteration focused on extending the Toolbox to support more types of posts to enable pilots and policy makers to populate the Toolbox towards a knowledge base for policymakers where they can discover tools, techniques, visualisation types, etc. in their policy making processes.
In parallel, the technical teams also worked on fine tuning the technical design as insights grow. As there are no incremental deliverables defined for D4.1, the updates to the technical specifications are provided as an annex to this document.

This Policy experimentation and functional design reports has four chapters. Chapter one is the introductory chapter. In Chapter two, the PoliVisu toolbox is analysed emphasising the policy design process flow and metadata integration. In chapter three, we're focussing on the visualisation tools itself starting from the visualisation goal, the visualisation description, status, future improvements and conclusions. In chapter four we will end up with a number of conclusions.

2 Toolbox

2.1 Design Policy Process flow

The different artefacts generated during the policy development are stored and described in the Toolbox. The proposed policy design process itself is as follows:

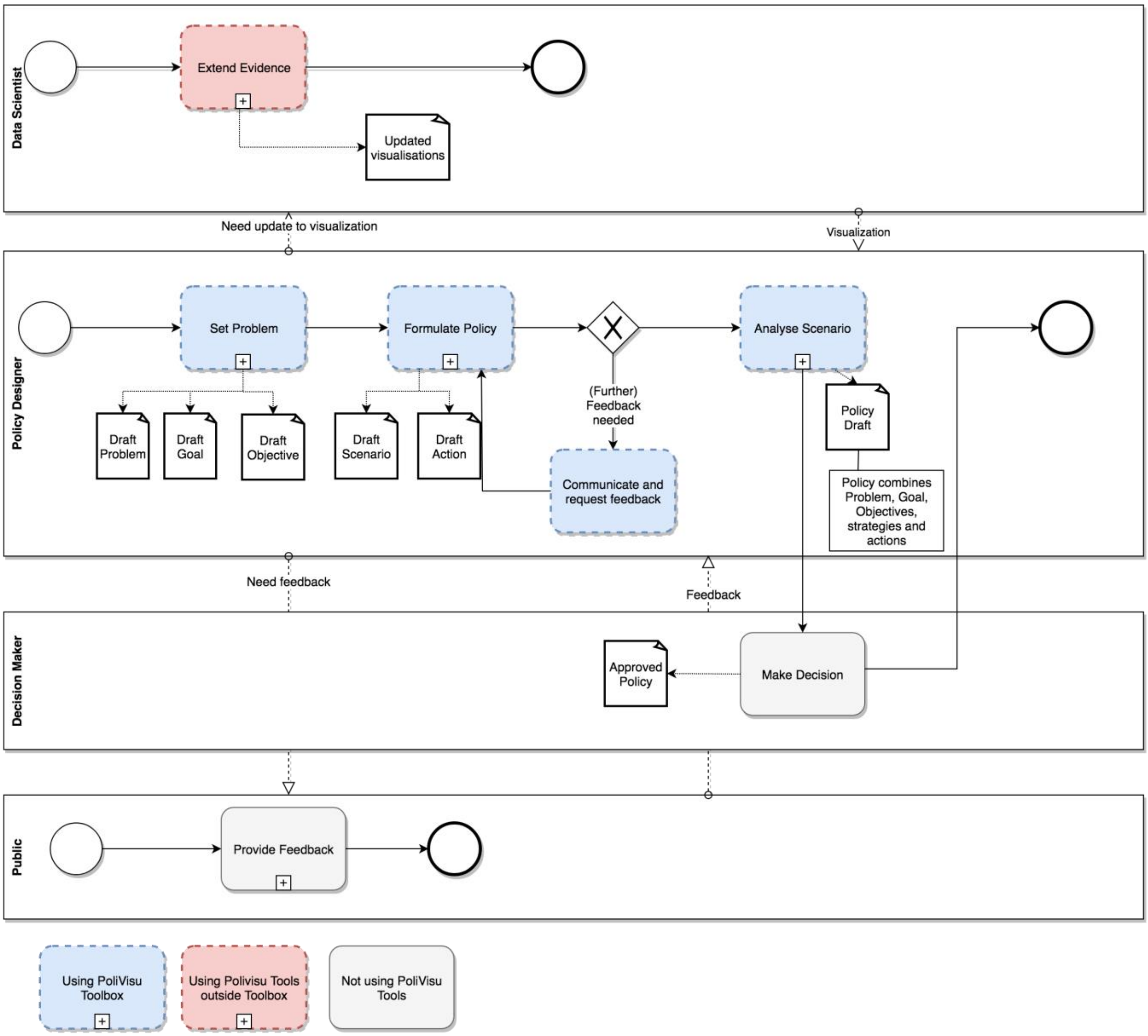

Figure 1: The Policy Design Process

The polivisu toolbox focuses on providing on content related to problem setting, policy formulation, communication and feedback techniques and scenario analysis. These steps are related to policy design. The tools suitable for data scientists are also available as a result of PoliVisu, but are available as separate tools outside the toolbox. The toolbox only describes how these tools can be used for policymaking purposes.

2.2 Toolbox

The diagram below recaps the different types of data stored in the Toolbox. The empty element in the middle connected with double arrows indicate toolbox entries of a certain type that can reference posts of any other type. For example, it's possible for a post that describes a certain dataset type to refer to a technique it could be useful for.

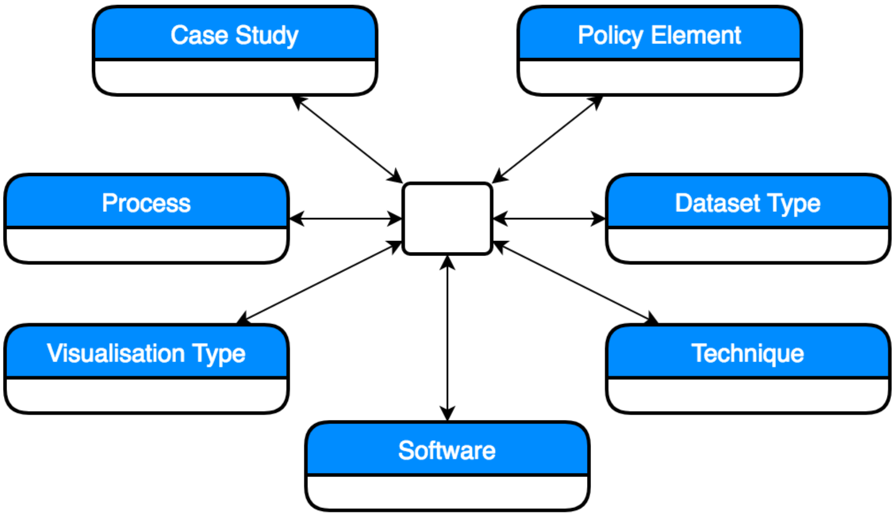

Figure 2: Types of data in the KnowledgeBase

After reviewing the initial set of mock ups with key stakeholders, a new and improved design was created. It was decided to use the case studies as a central piece of information since this type of information is well fit to tie all the other pieces of information together. From a UI perspective, it means that we have on the one hand Case Studies and on the other hand all the other types brought together under a Policy Ingredients header. On the main page, we put 4 random elements under a general header and a search bar. At the bottom, there is a map showing the geocoded elements.

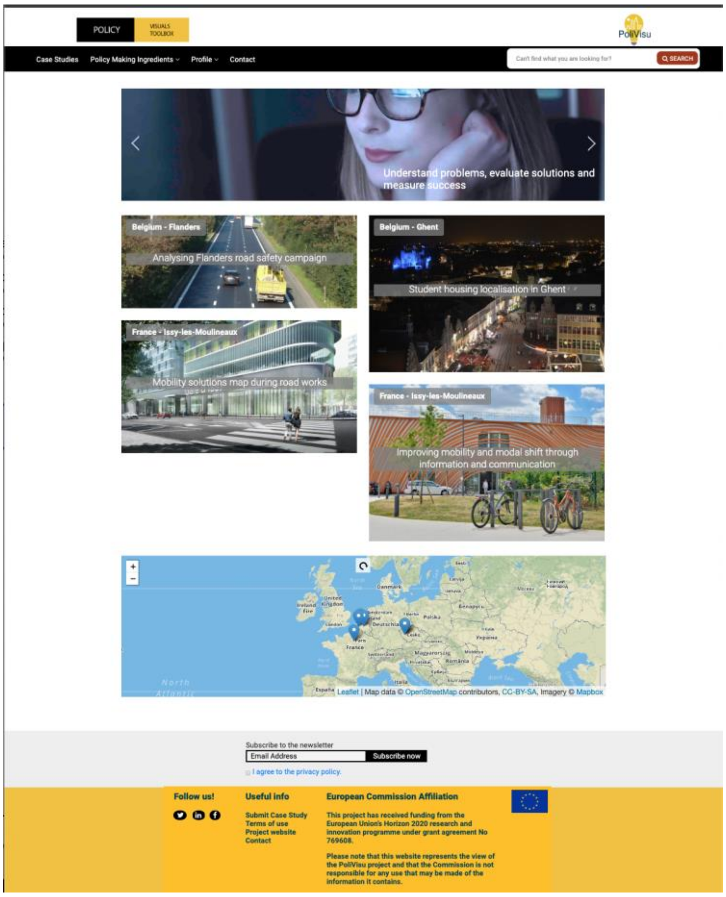

Figure 3: PoliVisu Toolbox front page

For each type of data, a template is defined to match the data model. Each type has a page listing all elements of that type.

Figure 4: PoliVisu toolbox case study overview and articles

Every article of the site can get a review, both textual and with a 5 star rating.

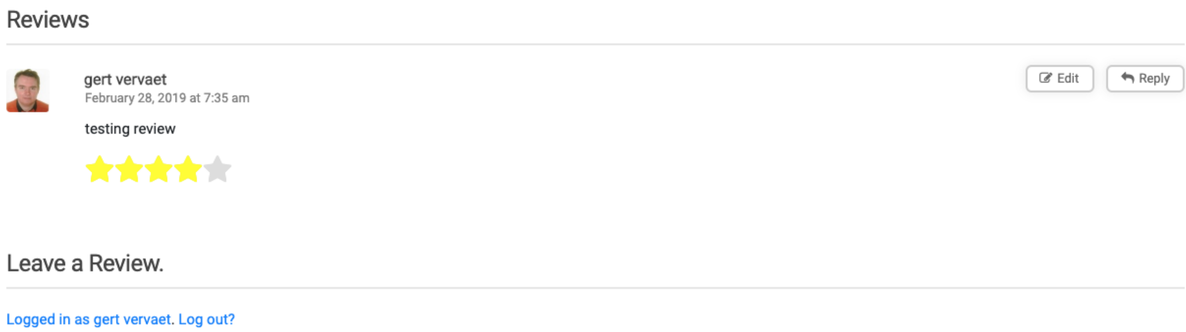

Figure 5: PoliVisu toolbox review

Indicating the location for the case study etc. happens using below User Interface (defined "UI" in this document). A user can indicate a location in a textual way and/or put a pin on the map.

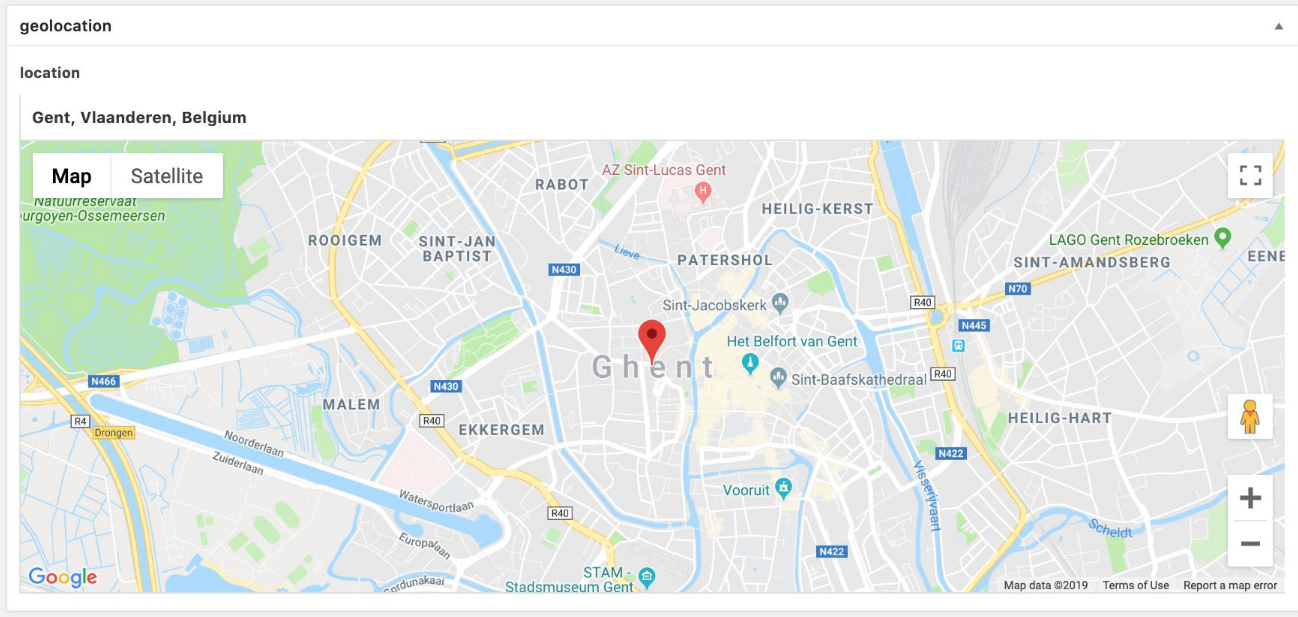

Figure 6: PoliVisu toolbox Geo-selection

Since initially the number of people providing content will be rather limited, we decided to use the already proven rather basic wordpress authoring UI.

2.2.1 Metadata integration

Dataset types are defined as a content type in the context of Toolbox, so they appear as articles. We will describe a number of custom data types such as cellular data and floating car data. Next to that, we need the capability to point to specific datasets, like the speed trajectory controls in Belgium. Trajectory speed control measures the vehicles travel time between two fixed points along a road equipped with ANPR cameras, and infers its average speed from those observations.

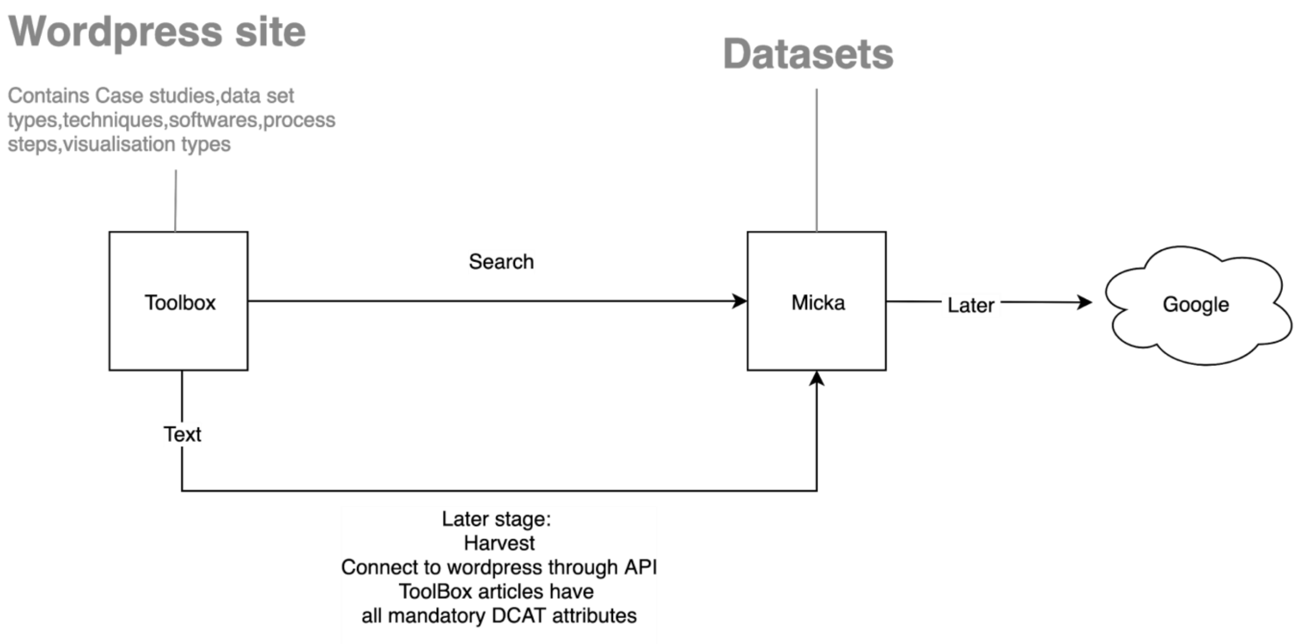

Figure 7: PoliVisu toolbox metadata search approach

The above design describes how the linking to the datasets is done through Micka[1], the set of metadata tools. The first integration with metadata will allow searching for datasets from the Toolbox in Micka and linking datasets to articles in the Toolbox. The user will be able to search for datasets through a search form. The search criteria include search by free text, resource type, contact and scale fields. It should be noted that only users with admin rights have access to this functionality.

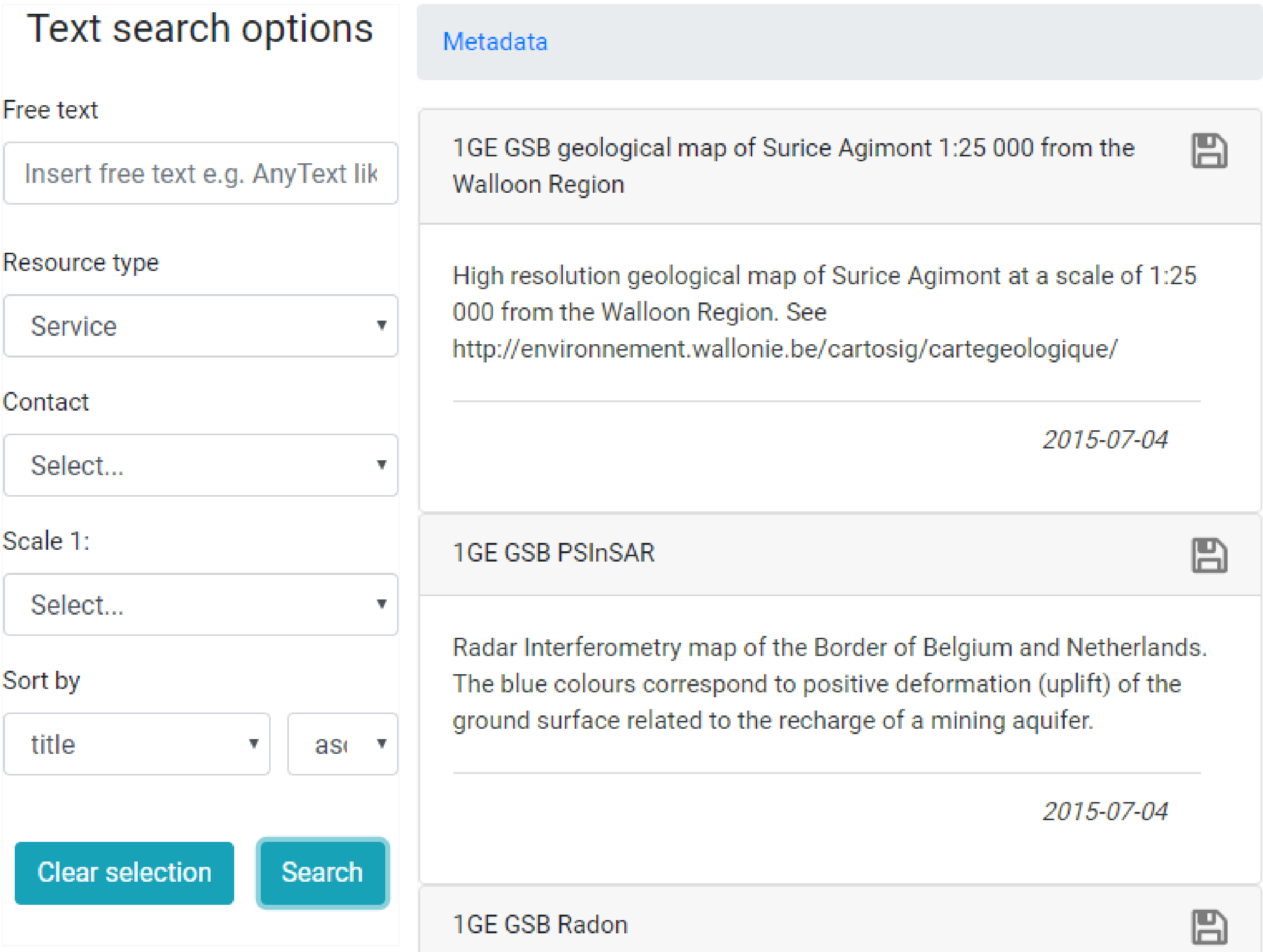

Figure 8: PoliVisu toolbox metadata text search - Dataset overview

The dataset search form is displayed at the end of the article. The list of the article datasets (already saved) is shown above the search form. A dataset can be removed from the list by clicking on the "remove" icon. Similarly, in order to link/relate a dataset to the article, the user has to click on the "save" icon in the search results list. The save operation is disabled for datasets that have already been associated with an article previously.

The first version of the Micka plugin for Toolbox uses only a text search-field based approach, e.g. the user cannot interact with a map to query for datasets. In the future versions, the user will be able to view the bounding boxes of the datasets on a map and interact with the map (select a certain area) to get related datasets. The current implementation is a WordPress plugin that uses jQuery JavaScript Library, Handlebars JS

[1] http://micka.bnhelp.cz/

plugin (to build semantic templates) and the WordPress plugin API (the article's datasets list is saved as the post's metadata). The design and the user interface are influenced by the Micka demo search page.

By clicking on a dataset title, the user can view all the details related to that dataset as shown in the image below.

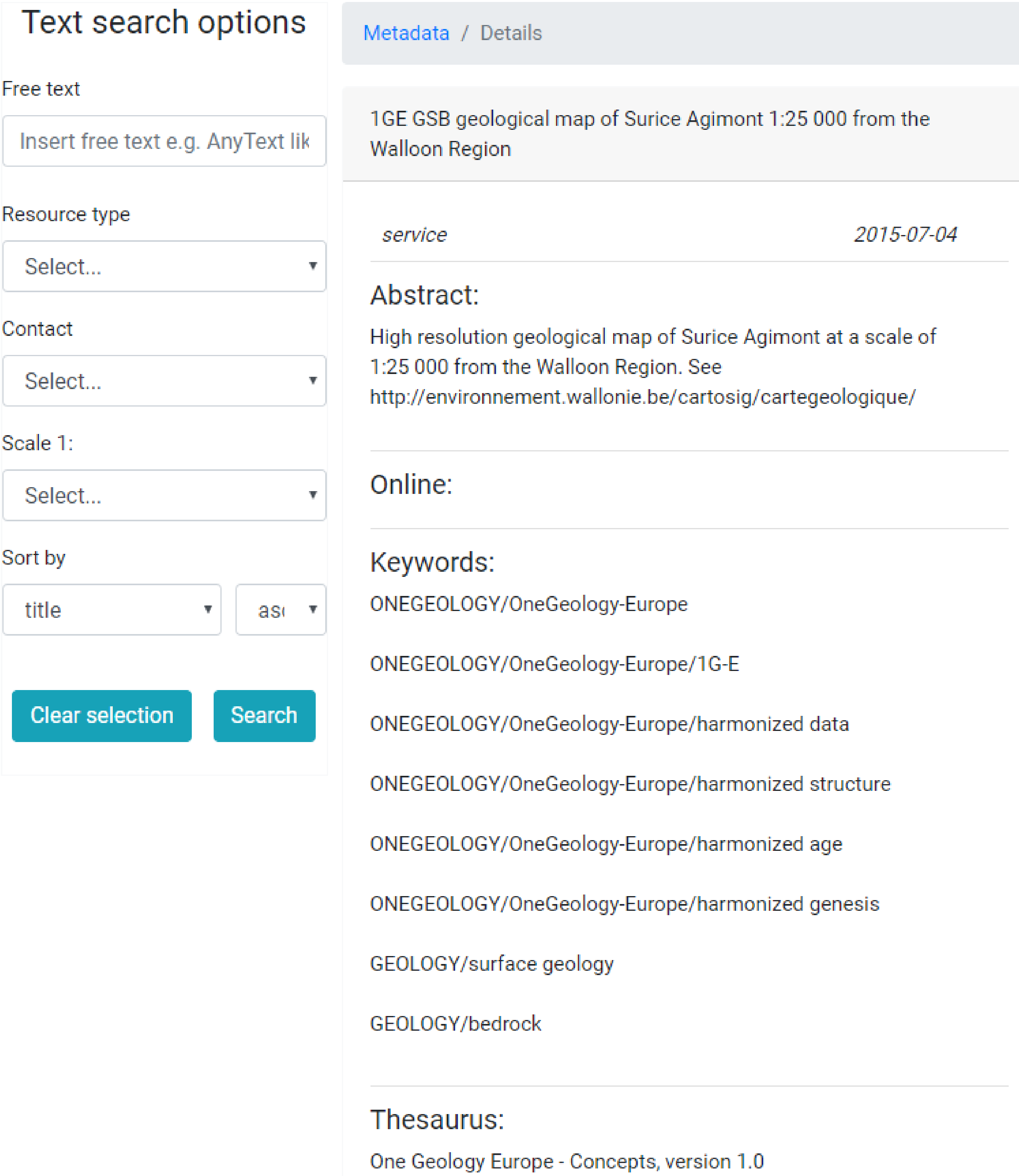

Figure 9: PoliVisu toolbox metadata text search - Dataset details

Initially, functionality to add metadata entries through the Toolbox was also considered to be part of the scope. However, during a stakeholder meeting, it was decided not to implement this in the first instance. Datasets not yet in Micka will be added using the existing Micka UI. Afterwards they can be linked in the Toolbox.

In a following phase, we will implement the option to export metadata of all the other content types (e.g. tools, techniques) into Micka, making this knowledge more readily available.

Mock-Up for spatially selecting a dataset from Toolbox

Figure 10: PoliVisu toolbox metadata Geographical extent (bounding box)

The blue rectangles depict the bounding boxes of existing datasets in Micka. The red box is the selection area that is drawn using the 3 icon from the top. Using this functionality, it’s very easy to identify datasets that cover a certain area of interest.

3 Interactive Visualizations

PoliVisu focuses on Interactive Visualisations. The advantage and USP of Interactive visualisations are that an interactive visualisation allows the user to play with the visualisation by making advanced selections, simple data manipulation and by having the results directly. Interactive Visualisations provide a whole new world of immediate insights useful for operational and strategic decision making.

This chapter lists the results of the analysis performed to update different interactive visualisations. The table below shows the table used in the visualisation analysis.

Pilot	Policy Element	Visualisation Goal	Description of main data fields	Visualisation Techniques	Visualisation	Visualisation Tool	Visualisation summary
Pilsen	Policy Goal: Reduce Congestion	Show actual, historical and predicted traffic volume for each road segment	Line data set of traffic volume for road segments - Location (road segment geometry) - Traffic volume (quantitative) - Date "past, current, future" (ordinal) - Time of the day	Choropleth linemap Text and position (calendar widget) Text and position (hour slider widget)	Live and Historic Traffic Volume Map for Pilsen	HSLayer line map	HSLayers based flow linemap of traffic, heat of the road segment is defined by the traffic volume
Pilsen	Policy Goal: Reduce Congestion	Compare the increase/decrease in traffic volume between two moments in time by allowing an expert to select those moments using either measured data from the past or predicted data (model)	Line data set of traffic volume for road segments - Location (road segment geometry) - Increase/decrease in traffic volume (quantitative) - Traffic volume (quantitative) - Relation of traffic volume to road segment capacity (quantitative) - Moment in time (ordinal)	Flow linemap Color (for increase/decrease) Line width (for traffic volume) Line pattern (for relation) Text and position (in form of calendar widget) wiil be used to select the time between which the traffic status will be compared	Traffic Volume Delta Map for Pilsen	HSLayer line map	The color of the line (road segment) will represent increase (color range of red color) or decrease (color range of green color) in traffic volume between two selected moments for each road segment.
Pilsen	Policy Objective: "To be ready for extraordinary events influencing traffic" Policy Goal: Better coordination of roadworks	Simulate the effect/impact (difference in traffic volume) when adding, removing, changing traffic attributes (closure time, free flow speed, etc.) on road segments in the Pilsen region	Line data set of road segments - Location (X,Y) - Traffic Volume (quantitative) - Status "new, update, delete" (nominal) - Description (nominal)	Flow Linemap Color (for traffic volume) Text for Status and Description	Traffic Simulation Map for Pilsen	HSLayer line map	HSLayer based flow linemap of traffic volume, the color of the road segment is defined by the calculated traffic volume. Visualization of changes on road segments before and after the change in the traffic model
Pilsen	Policy problem identification and agenda setting via traffic data analysis	Explore and investigate traffic volume on road segments in time and space to discover patterns, correlations and extremities using aggregated data	Line data set of traffic volume for road segments - Location (road segment geometry) - Traffic volume (quantitative) - Temporal dimensions	Flow Linemap Color (for traffic volume) Bar chart, text position (calendar widget) for temporal	Traffic Volume Heatmap for Pilsen	WebGlayer line map	WebGLayer based heatmap of traffic, heat of the road segment is defined by the traffic volume

			(quantitative)	dimensions			
Pilsen	Policy problem identification and agenda setting via traffic data analysis	Explore and investigate traffic incidents (traffic accidents and other police records) in time and space to discover patterns, correlations and extremities using aggregated data	Point data set of traffic incidents - Location (X, Y) - Severity (ordinal) - Classification vehicles (nominal) - Temporal dimensions (quantitative)	Point Heatmap Bar chart, text position (calendar widget) for temporal dimensions Text and barcharts for classification and severity	Traffic Incidents Heatmap for Pilsen	WebGlayer point map	WebGLayer based heatmap of traffic incidents, heat of the area is defined by the volume of historical traffic incidents.
Issy-les-Mouli neaux	Policy Goal: Reduce Congestion	Show how much time a traveller loses on road segments in Issy compared to the ideal situation (no traffic) for specific periods and timings.	Line data set of wasted time for road segments - Location (road segment geometry) - Lost time "average of extra time spent vs. time without traffic" (quantitative) - Time window	Flow linemap Color (for lost time) Bar chart	Delay Due to Traffic Map for Issy (3.1 in D3.7)	WebGlayer line map	WebGLayer based heatmap of traffic, heat of the road segment is defined by the extra time spent due to traffic versus theoretic free flow speed
Vlaan deren	Problem Setting: Identify accident black spots	Show locations where accidents happened in Flanders in the last 5 years. For the accidents show information relevant to find black spots due to the infrastructure	Point data set of traffic incidents - Location - Time and date - Weather, road and traffic conditions - Modes of transport - Consequence for people involved	Point Heatmap	Traffic Accident Heatmap for Flanders	WebGlayer point map	WebGLayer based heat map, showing the locations of accidents with relevant attributes allowing to investigate the severity and whether the location is the main contributor to the accident or other factors (like alcohol)
Gent	Problem setting: Identify student residences	Display on a map of Ghent the number of students that are spending the evening (18:00-00:00) at a certain location.	polygon dataset with as attribute the presence of possible residing students	Polygon choropleth map	Mobile telecommunic ation data visualisation in Ghent	TBD	Visualisation is being fine tuned with Proximus, provider of mobile telecom operator

Table 1: Visualisation methodology results overview

3.1 Analysis for Interactive Visualisation Tools

3.1.1 Storytelling

During the discussions inside the consortium about how to visualize the data, and while investigating different visualisation techniques, it became clear that in order to clearly convey the outcomes of analyses we could benefit from building visual stories that take citizens and policy makers step by step through a predefined story.

We pursued this idea further and decided to provide easy tools for telling geo-data based stories. A first mock was made based on the Flanders accident map as a proof of concept (link).

We set the goal of making storytelling part of the tools provided by Polivisu, and want to provide a fully operational storytelling tool. As a first step we had to decide to either build a new solution, or use an existing open source solution that can convey map-based data effectively.

- Option A: build storytelling tool;
- Option B: apply existing GIS-storytelling tool;
- Option A+B: adapt existing storytelling tools.

Initial functional analysis

In our initial functional analysis we identified what a storytelling should look like. We defined technical requirements (e.g. is it open to use? Does it technically allow to write a story?) and operational/useability requirements (can a story writer perform action A, B, ...? Is it user-friendly and can a non-technical person use this (with a short learning curve)?)

The map-based storytelling tools that were investigated were: Odyssey, Mapbox, Story Map JS, Twine, Tableau, Waypoint.js, Scrollmagic, Scrollytime / mapbox GL and Arcgis. Based on the requirements that were set out in the analysis, Odyssey and Story Map JS came out best[2].

Tool	Link	Open source?	Open? Or licenced	Map via API/embed	user can login to create/ modify/ delete his story	user can create a story through a basic WYSIWYG html editor	user can add pictures /videos (through links) to the story	user can define how the story is split in story steps/pages	user can bookmark a map composition and link it with the story step	user can share his story through unique URL	screen needs to have a map and a story part	story part needs to have a way to go to next/previous	story part can have different fonts, colours, bullets,...	a step in the story changes the way the map looks (pan, zoom, selection of layers, filter data)	story part needs to allow finetuning selections in a targeted way (we only give the user access to some filtering, what is applicable. This makes it less overwhelming)	stories can be created by non technical people and 'uploaded' without extra development	
Odyssey	http://cartodb.github.io/odyssey.js/documentation/																12
Mapbox	https://www.mapbox.com/about/																12
Story Map JS	https://storymap.knightlab.com/																13
Twine	http://twinery.org/																10
Tableau	https://www.tableau.com/data-storytelling																9
Waypoint.js	http://imakewebthings.com/waypoints/																7
ScrollMagic	http://scrollmagic.io/																9
Scrollytime / mapbox GL	http://mappable.info/2018/01/12/																11
Arcgis																	11

Legend: Green = Available, Red = Not available, Rose = No clear info

Table 2: Storytelling tool functionalities

However, both tools were deemed not flexible enough for the kind of stories we wanted to lay out.

Odyssey is a good standalone tool which is easy to operate, can embed maps and is in development, which means it will likely be expanded with new features. However, there were limited options for displaying maps, there was no in-map filter, no login-system and no video-embedding. Sharing a story can be complex, and there is no (auto-)saving. Once a user leaves the tab, all edits are lost.

Story Map JS also is a good standalone tool, which is very easy to operate. It has account-based auto-saving, video- and map-embedding enabled. However, there are no in-map-filters, the visualization options are rudimentary (colors, fonts), and there is limited flexibility in style: only slide-based storytelling is possible.

[2] https://docs.google.com/spreadsheets/d/155WpyJaPHxEzjhlfbuWoWVpSEEcJ0FNeOaxanUURbh4/edit#gid=0

Second off-the-shelf analysis

As both tools did not meet our needs, we decided to expand the list of requirements, and categorize these under ‘must’ ‘very nice to have’, and ‘nice to have’, and to open up the analysis to non-map based storytelling tools as well.

Open source?	MUST	
Map via embedding (URL)	MUST	via permalink, rather than URL as that would reload the whole map
Map via API	MUST	Need to be able to support this. One generic API may not enough.
user can login to create/modify/delete his story	MUST	User needs to manage content him/herself
user can create a story through a basic WYSIWYG html editor	VERY NICE TO HAVE	Could be wordpress, drupal, any HTML5 that allows templates ...
user can add pictures	MUST	preferably also .gif
user can add videos	NICE TO HAVE	
user can define how the story is split in story steps/pages	MUST	
user can create a map composition and link it with the story step	MUST	
user can share his story through unique URL	MUST	
screen needs to have a map and a story part	MUST	
story part needs to have a way to go to next/previous	MUST	
story part can have different fonts	NICE TO HAVE	
story part can have different colors	NICE TO HAVE	
story part can have bullets	NICE TO HAVE	
a step in the story changes the way the map looks (pan, zoom, selection of layers, filter data)	MUST	
story part needs to allow finetuning selections in a targeted way (we only give the user access to some filtering, what is applicable. This makes it less overwhelming)	NICE TO HAVE	
stories can be created by non technical people and ‘uploaded’ without extra development	MUST	
User management needs to be included	MUST	
Provide multiple stories based on same map (e.g. copy a story)	MUST	Start from storytelling option and refer to maps rather than the other way around
Making stories multilingual in editor	OBSOLETE	
Making stories multilingual and make user switch languages (language support)	VERY NICE TO HAVE	Workaround via copying stories
Non-linear stories	NICE TO HAVE	
Scrolling story	NICE TO HAVE	
Floating window	NICE TO HAVE	

Table 3: Storytelling tool - Requirements overview

In this second analysis we compared map-based storytelling tools (Odyssey, Story Map Js, Arcgis, Scrollytime/Mapbox GL), non-map based storytelling tools (Pageflow, Storybuilder, Twine, Tableau) and CMS-systems that may integrate WebGLayer (Wordpress, Joomla, Drupal, OpenCMS, ModX, Django CMS, Silverstripe, Wix, Google Sites).

		Open source?	user can login to create/modify/delete his story	user can create a story through a basic WYSIWYG html editor	user can create a map composition and link it with the story step	story part needs to have a way to go to next/previous	User management needs to be included	Provide multiple stories based on same map (e.g. copy a story)		
Map based storytelling tools										
Odyssey	http://cartodb.github.io/odysse								5	
Story Map JS	https://storymap.knightlab.cor								7	
Arcgis	Storymap								4	
Scrollytime / mapbox GL	http://mappable.info/2018/01/								1	
Non map-based storytelling tools										
Pageflow	pageflow.io								6	
StoryBuilder	https://storybuilder.jumpstart.c								6	
Twine	http://twinery.org/								4	
Tableau	https://www.tableau.com/data								2	
CMS systems for integrating WebGlayer										
Wordpress	Aesop Story Engine: https://w								3	
Drupal	flexible taxonomies that can h								5	
Wix									3	
Google sites									3	
Elements for storytelling (not tools)										
Waypoint.js	http://imakewebthings.com/wa								JS scroll function	
Mapbox	https://www.mapbox.com/abo								Not applicable as Storytelling tool as standalone. Has features provided by Webglayer	
ScrollMagic	http://scrollmagic.io/								JS scroll function	

Legend: Green = Available, Red = Not available, Rose = No clear info

Table 4: Storytelling tool - Requirements overview in relation to the candidate tools

Out of this list of tools we selected the best candidates for the comparison. The best potential fits were Odyssey, Story Map JS, Pageflow, Storybuilder and the Wordpress Aesop Story Engine.

Odyssey	http://cartodb.gith	
Story Map JS	https://storymap.k	
Pageflow	pageflow.io	
StoryBuilder	https://storybuilde	Is this still supported?
Wordpress	Aesop Story Engi	
Odyssey	43	
Story Map JS	52	
Pageflow	52	
StoryBuilder	40	
Wordpress	45	
MUST =	4	
VERY NICE TO HAVE =	2	
NICE TO HAVE =	1	

Legend: Green = Available, Red = Not available, Rose = No clear info

Table 5: Storytelling tool - Storytelling tool - Requirements detailed overview in relation to the final candidate tools

We scored every tool per requirement. If a requirement was a 'must' 4 points were given. If it was a 'very nice to have' 2 points, and a 'nice to have' 1 point. If the requirement was not entirely available, but could be added, it received half of the points. The 5 selected tools received the following scores:

Tool	Score
Pageflow	59
Story Map JS	56
Wordpress	52.5
StoryBuilder	50
Odyssey	45

Table 6: Storytelling tool - Final selection

Example stories using Pageflow

Pageflow is deemed to be the best candidate for the job. As a next step we reached out to Codevise, the company that built and supports Pageflow to see if a tool such as WebGLayer can be integrated. In parallel, we built an example story which used images of the filtered maps provided by WebGLayer.

The conclusion of these conversations however is that the benefits of building interactive maps don't outweigh the costs of building this solution, and that these images cover the needs that most stories pose. See example of the Flanders accident map:

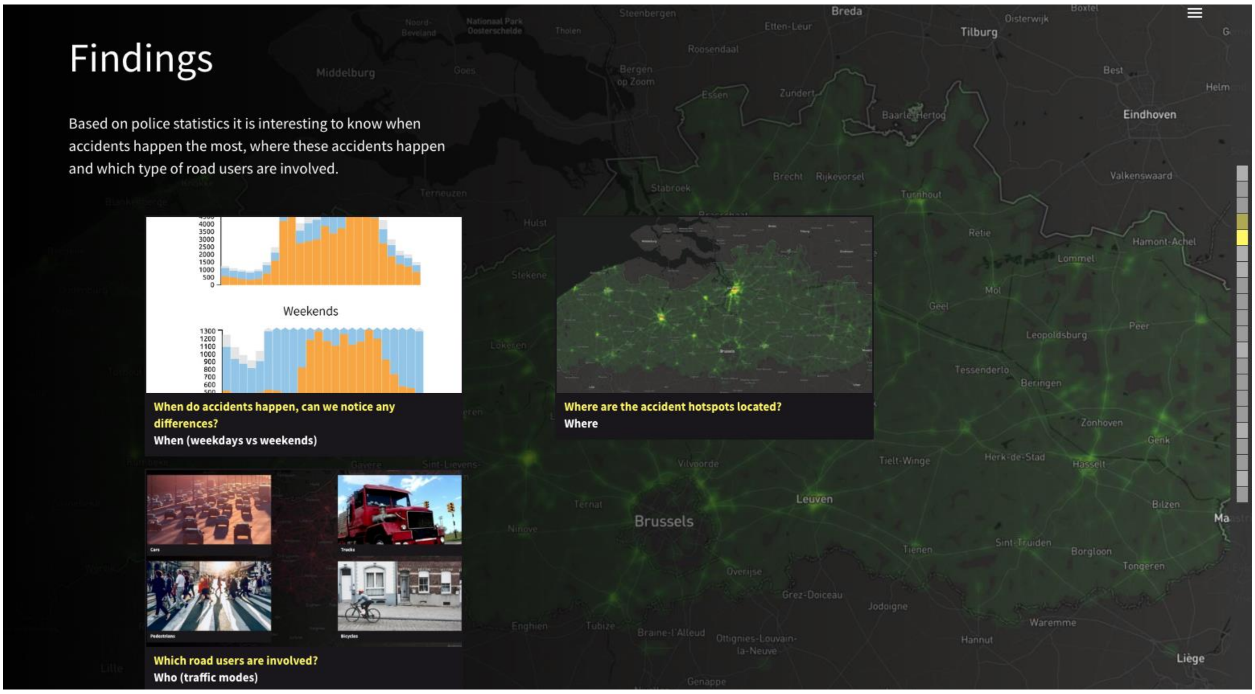

Figure 11: Storytelling - Pageflow traffic accidents story (Example 1)

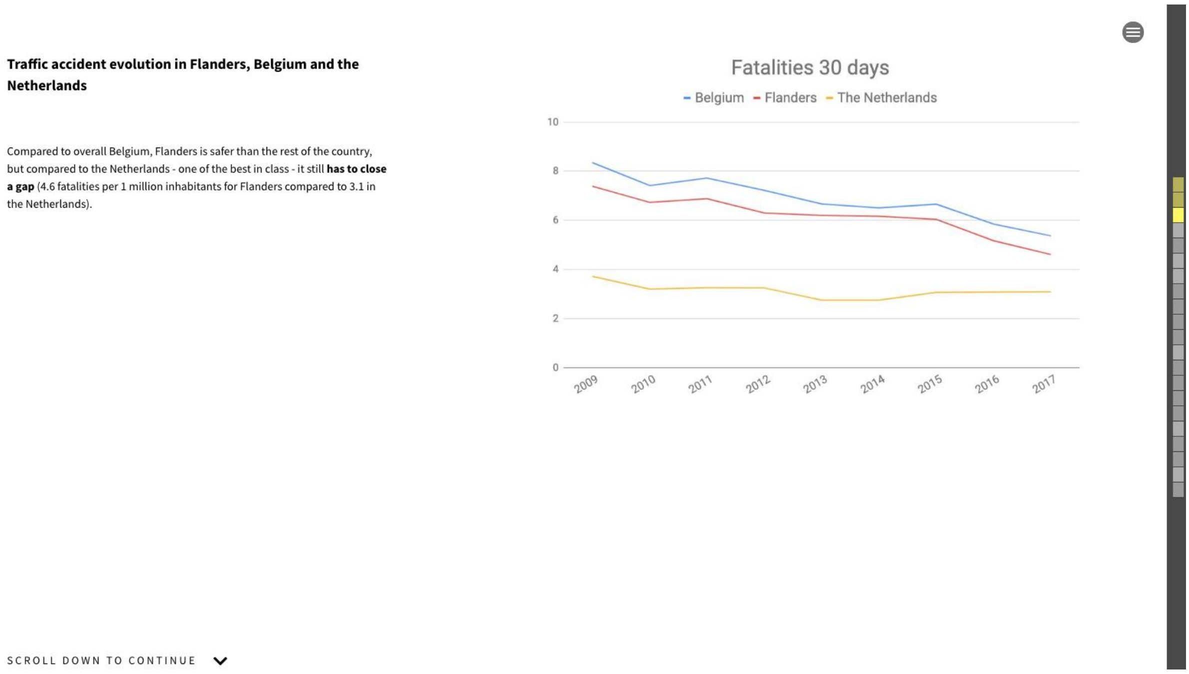

Figure 12: Storytelling - Pageflow traffic accidents story (Example 2)

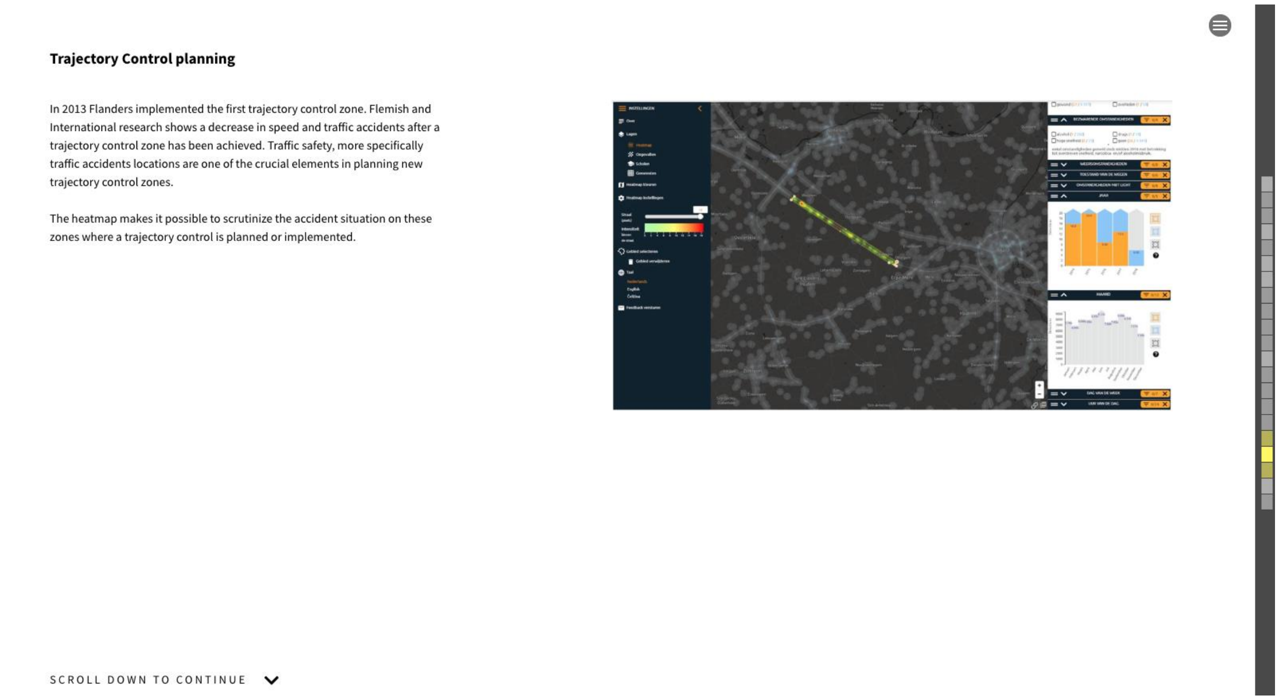

Figure 13: Storytelling - Pageflow traffic accidents story (Example 3)

Further stories will be provided using Pageflow as a tool. Once licences are arranged, these can also be published.

3.1.2 Traffic intensity Line and Point Map

The WebGLayer tool allows to visualise aggregated linear and point data and analyse it for trends and patterns using interactive filtering through charts of different types and through map selection (exploratory explanation). It can be used for a spatial and temporal analysis of any linear (or other vector-based) and point data such as traffic intensity and traffic accidents in case of PoliVisu.

Figure 14: Example of a WebGlayer-based linear analytical map

The user testing performed in the second test cycle of the project suggested the following improvements for next iterations:

- Redesign the UI in two collapsable panels left and right that drive the map;

- UI on smartphones and other touch devices could be improved;
- requests for better language support;
- update the accident data and add additional data layers to the Flanders traffic safety map (schools, communities and city transportation regions);
- allow users to export the selected data in .csv file;
- store map and language settings in cookies;
- improve documentation and add legend to the map;
- use averages per time window instead of total detections in the Pilsen Traffic map to improve the usefulness of the map;
- allow to activate expert map settings (extra traffic intensity layer) in the Pilsen Traffic map;
- improve the accuracy and completeness of the data in the Pilsen Traffic map;
- implement the latest traffic intensity data (through an API) in the Pilsen Traffic app;
- allow to choose date(s) from a single calendar-like chart to analyse the historic traffic in Pilsen instead of combining data selection and calendar grid as was the case in the previous version;
- allow to consult segment details in a tooltip to get insights on the statistics for the segment in the Pilsen Traffic map;
- allow to share the current situation of the map, barcharts and traffic statistics through permalink.

3.1.3 Traffic Volume line map

Line map is a cartographic concept of using line color and width to represent a real phenomenon. HSLayers line map chapter (3.3.3) in D3.6 described the basic concept and also the thematic domain which the line map concept is going to be used for - visualization of traffic volume. Please note that requirements identified in D3.6. Are now being implemented in chapters 3.2.3 Live and historic Traffic Volume map, 3.2.4 Traffic Simulation Map for Pilsen and 3.2.5 Traffic Volume Delta Map for Pilsen.

3.2 Analysis for Visualisations

3.2.1 Live and Historic Traffic Volume map for Pilsen

Please note that these maps for Pilsen (Live and historic Traffic Volume Map, Traffic Simulation Map and Traffic Volume Delta Map) will all be integrated into one web app. For the current version of this app designed for the Pilsen pilot see: http://app.hslayers.org/stm/apps/pilsen.

3.2.1.1 Visualisation Goal

The map visualizations ensures these Goals:

- Showing the actual level of traffic volume for each road segment (with traffic sensor);
- Showing the historical state of the traffic situation.

3.2.1.2 Data Preparation

The cornerstone of this map are data from traffic sensors (data measured from sensors located underneath the road segments). More info about the traffic model can be found in D5.1.

3.2.1.3 Visualisation description

The user can choose the date and hour in the calendar box and the map will display via the visualisation the traffic situation in the city according to this date:

- The defaut map window displays the actual traffic situation in the city according to the traffic sensors;
- when user will choose a date in the past, the map will display traffic volume map from traffic sensors;

The map is currently being developed. For preliminary visualization see figure below.

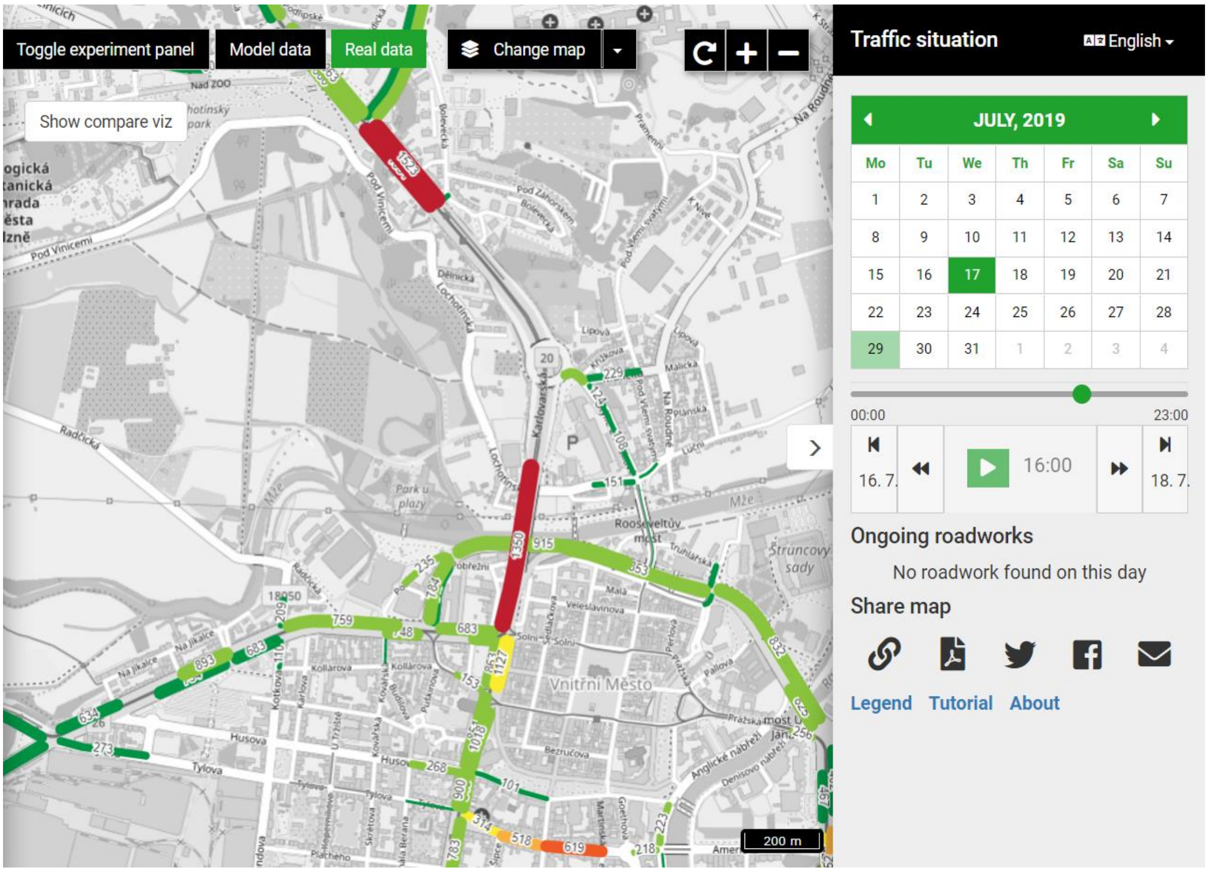

Figure 15: Life and historic traffic volume map - Visualisation of an afternoon traffic peak hour based on historical data

3.2.1.4 Visualisation Status

The status of the visualization can be checked at its url: http://app.hslayers.org/stm/apps/pilsen. Currently implemented functionality follows:

- User can select a specific view on historical data.
- User can compare traffic volumes of two different historical time slots

3.2.1.5 Future improvements

The next development is traffic prediction visualisation.he map and visualisation techniques will be used and modified if necessary to display the predicted traffic volumes model calculated from the street network, traffic generators and the planned roadworks. (*see 3.2.4 Traffic Simulation Map for Pilsen*).

3.2.2 Traffic Volume Delta Map for Pilsen

Please note that these maps for Pilsen (Live and historic Traffic Volume Map, Traffic Simulation Map and Traffic Volume Delta Map) will all be integrated into one web app. For current version of this app for the Pilsen pilot see: http://app.hslayers.org/stm/apps/pilsen/.

3.2.2.1 Visualisation Goal

The goal is to compare changes (increase or decrease) in traffic volume between two moments in time by allowing experts selecting two different times using measured data from the past or predicted data calculated by the traffic model.

The user will be able to:

- Compare two moments in the past based on historical sensor data;
- compare two simulated traffic models based on calculated forecasts;
- compare historical sensor based data with the simulated forecast models.

3.2.2.2 Data Preparation

The used data will be based on the traffic volume map and the recalculation of the difference between the two states by the traffic modeller application.

3.2.2.3 Visualisation Description

This visualisation is based on the HSLayers Line Map. The color of the line (road segment) will represent an increase (orange color range) or a decrease (violet color range) in traffic volume between two selected moments for each road segment. The specific ways of visualisation will be discussed during the next iterations and will be part of the user testing.

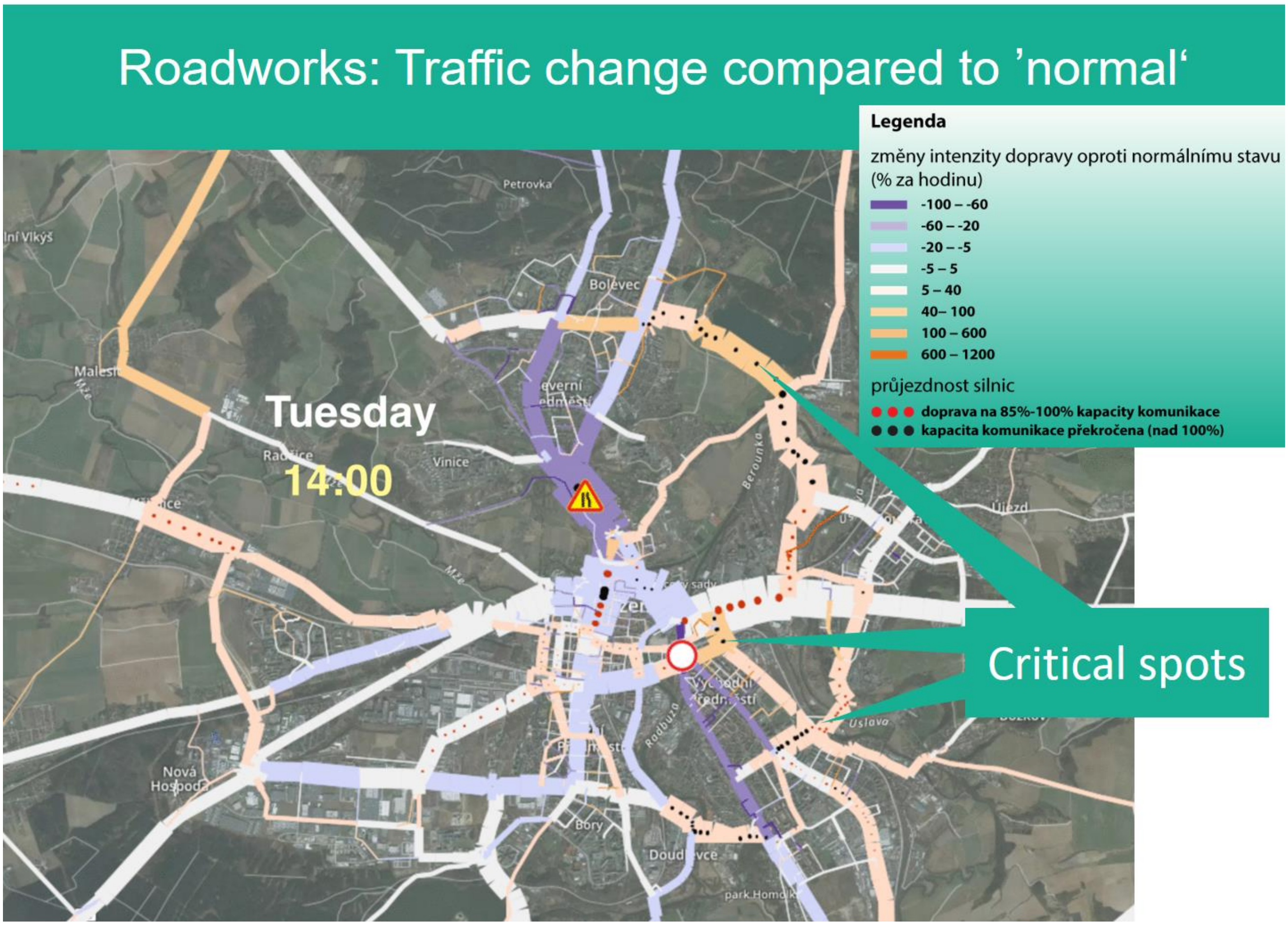

Figure 16: Traffic volume delta map - A Mockup based on an application developed in the Open Transport Net H2020 project

3.2.2.4 Future improvements

This map below is currently under development and give an overview of the future. The future developments are all in line with the scenario comparison goal that fits well into the polivisu policy design approach, especially concerning policy co-creation.

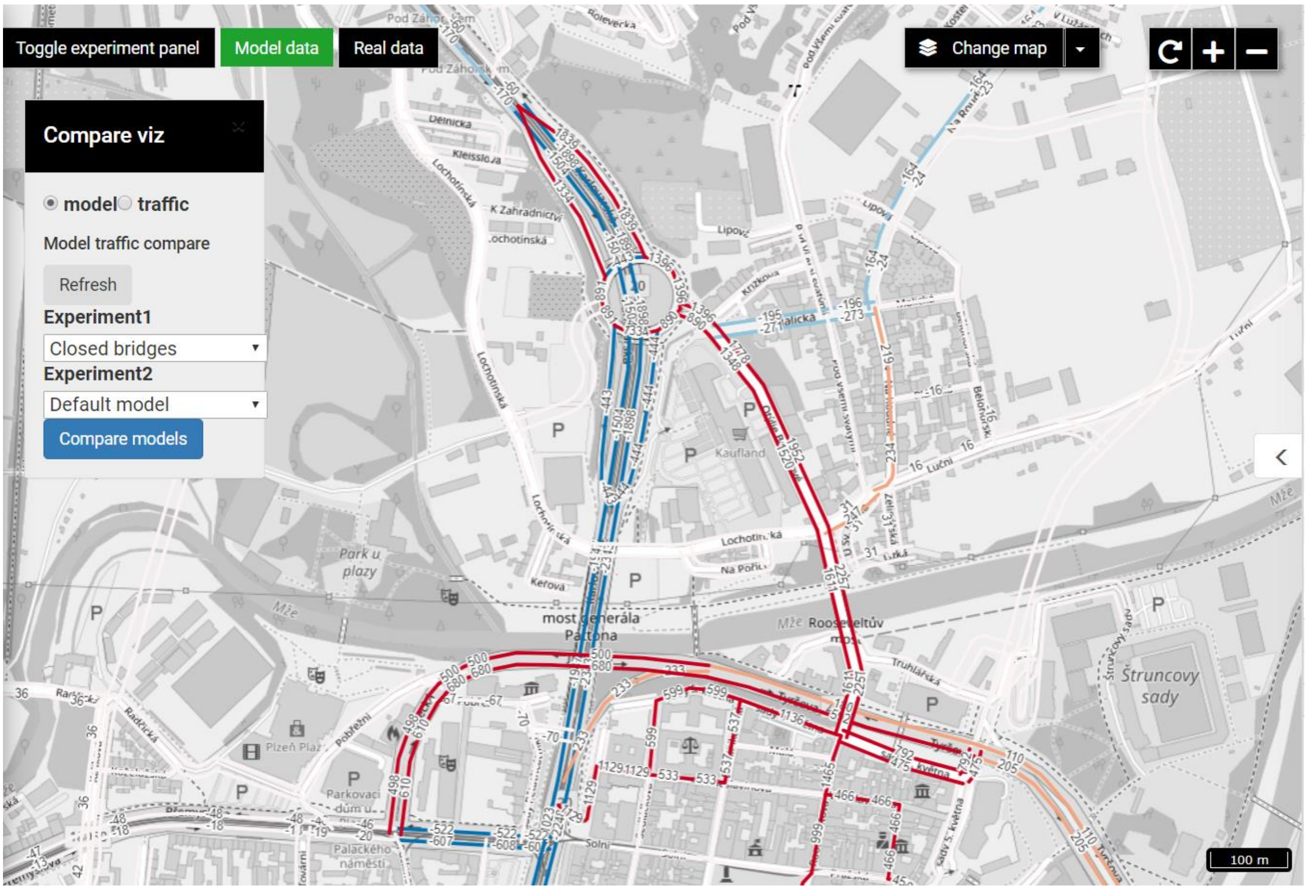

Figure 17: Traffic volume delta map - Visualization of the modelled traffic flow difference between a default model and a simulated road closure

The example below shows a comparison based on historical data.

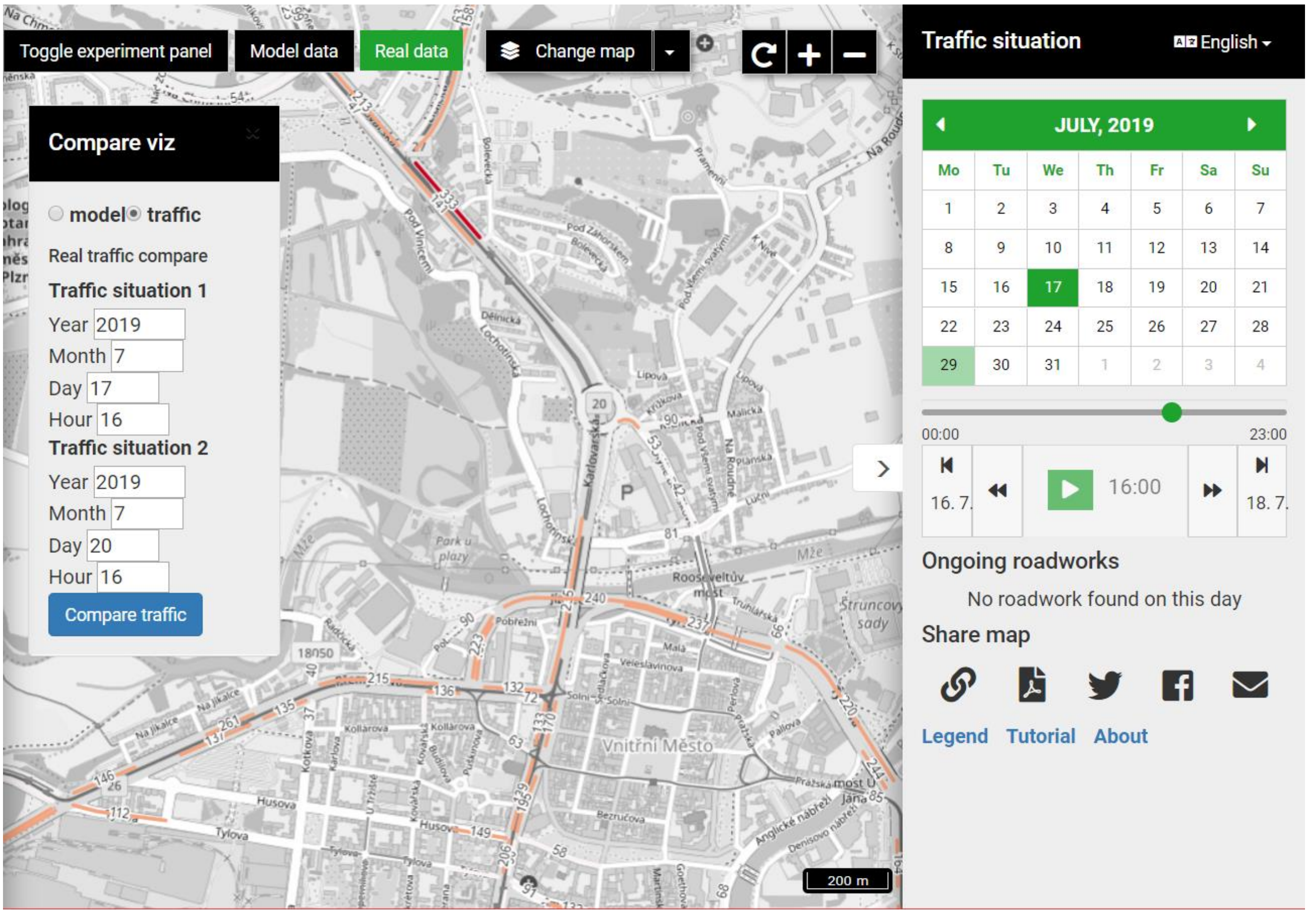

Figure 18: Traffic volume delta map - Visualization of historical traffic flow difference between two peak hours (on Wednesday and on Saturday)

3.2.3 Traffic Simulation Map for Pilsen

Please note that these maps for Pilsen (Live and historic Traffic Volume Map, Traffic Simulation Map and Traffic Volume Delta Map) will all be integrated into one web app. For the current version of this app for the Pilsen pilot see: http://app.hslayers.org/stm/apps/pilsen.

3.2.3.1 Visualisation Goal

The map visualizations will ensure:

- Informing the citizens about the time schedules and the extent of actual and planned roadworks in the city within a defined time range with simulated traffic and planned roadworks. (*Please note that this refers to D3.6 3.4.6 Planned Roadworks Map for Pilsen*);
- Simulating the effect/impact (difference in traffic volume) when adding or removing road segment, changing traffic attributes (closure time, free-flow speed) on road segments in the Pilsen region. Showing the impact of roadworks on traffic volume for each road segment.

3.2.3.2 Data Preparation

The data for the visualization will be calculated by Traffic Modeller based on the developed traffic model. According to the changed configuration of the traffic network or the attributes of the selected segments, the traffic flow will be recalculated and the network will be visualized.

3.2.3.3 Visualisation Description

The modelled status of the traffic network will be visualized similarly as the Live and historic Traffic Volume map for Pilsen. The road segments will be coloured according to the traffic flow, newly added segments or closed segments. These will be displayed with additional map symbols. The user can define segments to be closed, draw new segments by hand mode, update attributes of the selected segments and select the time range of the change.

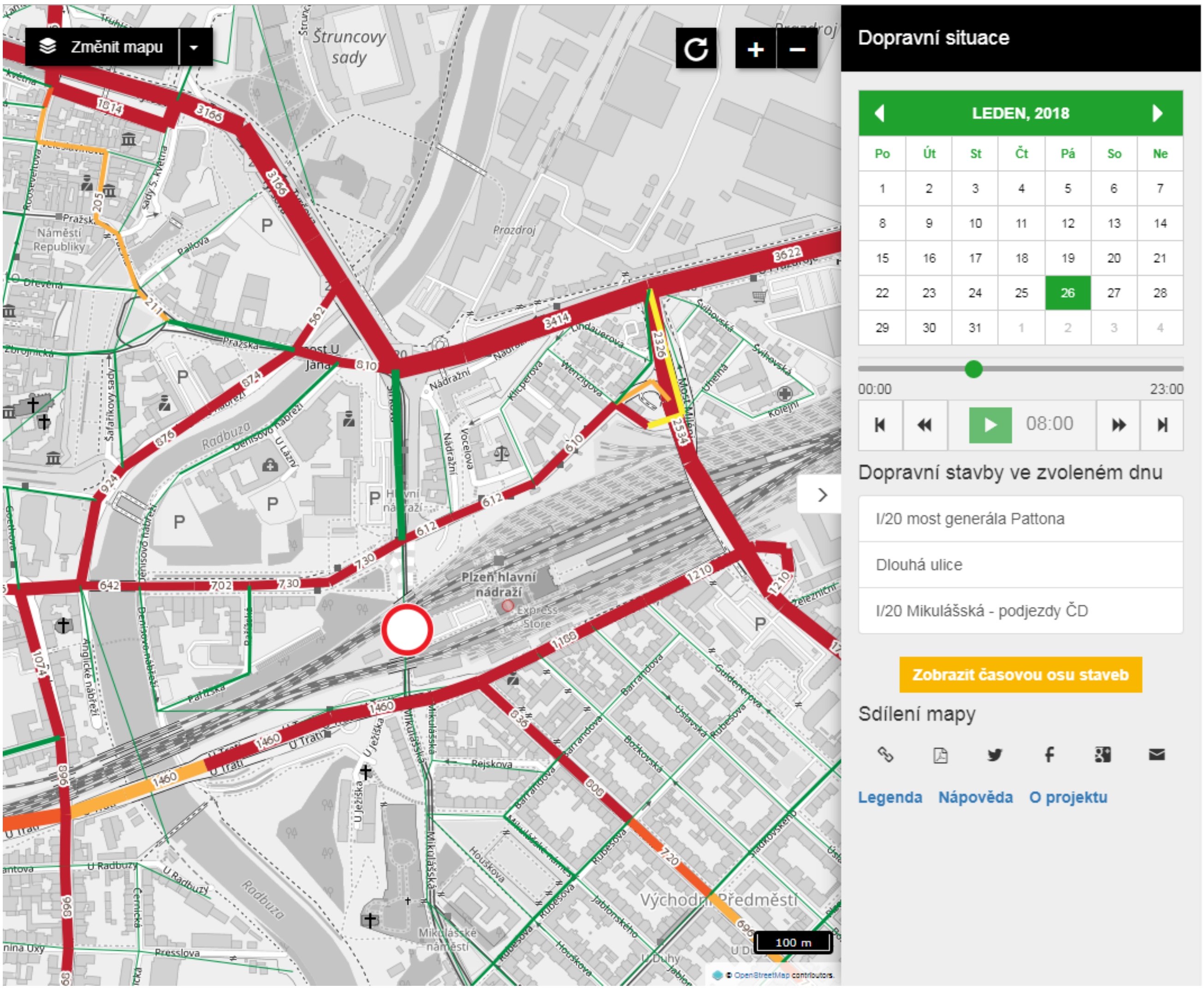

Figure 19: Traffic simulation map of Pilsen - Visualisation of a traffic closure

3.2.3.4 Visualisation Status

The status of the visualization can be checked via: http://app.hslayers.org/stm/apps/pilsen. The implemented functionality allows:

- The user can view the traffic model visualising the new planned and already existing traffic obstructions and can elaborate the possible traffic influence;
- a user with appropriate user rights, typically a city manager can store the modelled situation and can publish these modelled situation on a web page.
- the user can compare the traffic volumes of two different traffic models.

- (again) a user with appropriate user rights, typically a city manager to store the modelled situation and even publish it to a web page.

3.2.4 Traffic Volume Heatmap for Pilsen

The Map of Traffic in Pilsen is a WebGLayer-based heatmap visualising the traffic intensity. The heat of the road segment is defined by the traffic level (calculated as traffic intensity / road capacity) and the traffic intensity. The map allows the user to analyse the traffic level in the city in an interactive way. The traffic data is measured at 307 street segments where detectors (induction loops) are installed.

3.2.4.1 Visualisation Goal

To explore and investigate the traffic volume on road segments in time and space in order to discover patterns, correlations and extremities.

3.2.4.2 Visualisation status

The first release of the app was tested by a group of users early 2019[3]. The analysis of the user feedback resulted in the functional requirements below. The results will be used in the next major release (2.0) of the map:

Data-related requirements

- Ensure the accuracy and completeness of the data;
- daily update of the data in the app through an API fetching the data from the city server, create script for daily download of the data;
- update the geometry with capacities from the new traffic model to better reflect reality.

Filters and map functionalities

- Implement a new interactive date selection chart (calendar-like) combined with data loading (Pilsen traffic) that will replace the original date selection and calendar grid of the version 1.0
- simplify the charts for the Pilsen traffic in a way it displays the 'selected' data (i.e. remove the 'unselected' and 'out of the map' data from the columns);
- change the provided traffic intensity figure to an average per segment (instead of the total number of detections on all segments visible on the map);
- rename the basic map layer to 'Traffic Level' (calculated as a ratio between intensity and capacity);
- add a new Traffic Intensity layer as an 'advanced map setting';
- add light background map;
- create a new colour scale for Traffic Intensity and update the original one for Traffic Level;
- use grey colour street segments and for calendar dates with no data;
- improve segment selection: widen the clickable margin and graphically highlighthe segment;
- Nice to have: add arrows and street numbers to segments (needs to be added in data).

UI, UX, documentation

- Implement the UI with two collapsable panels (left menu and right panel with charts):
- improve the documentation: add an “About text” and “Legend”;
- add Czech and English translations;
- store the user selection of map settings and language in cookies, add cookies message;
- implement a loading screen with an 'Intro' dialogue;

[3] See D7.3 Recommendations for future deployments 2

- add a warning when user selects a combination of filters with no data selected;
- add Help texts and tooltips for charts;
- add data labels to day and hour charts when hover over the columns;
- add tooltips for open/hide/reorder charts and menu;
- add 'contact us' functionality.

Map sharing and re-use
- Implement permalink to share the current state of the map and filters;
- add CSV export of the selected data;
- nice to have: PDF export of the current state of the map and filters.

The above-listed requirements have been processed into new mockups that can be found below. These contains improvements to the user interface and the interactivity. These improvements will be tested during the next phase of the user testings.

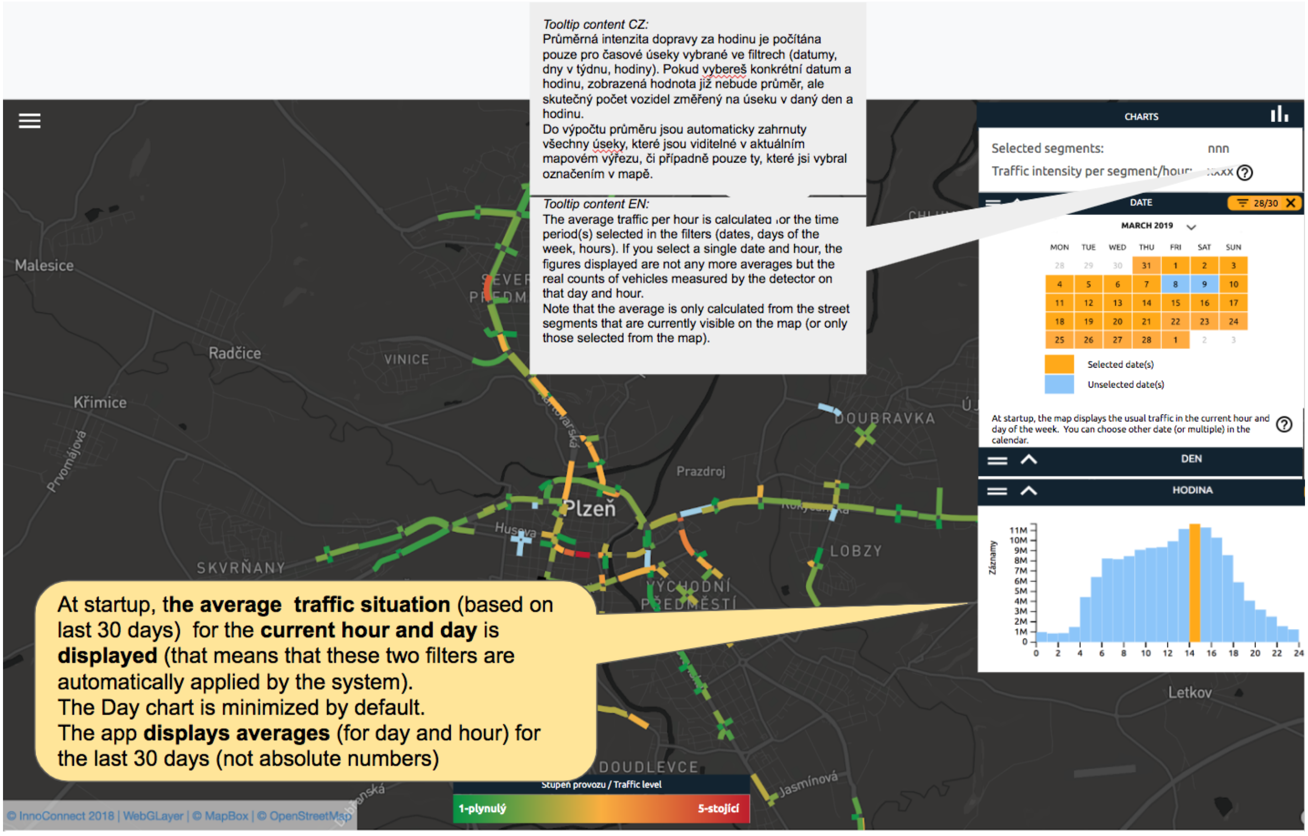

Figure 20: Traffic volume heatmap of Pilsen - Design mock-up 1

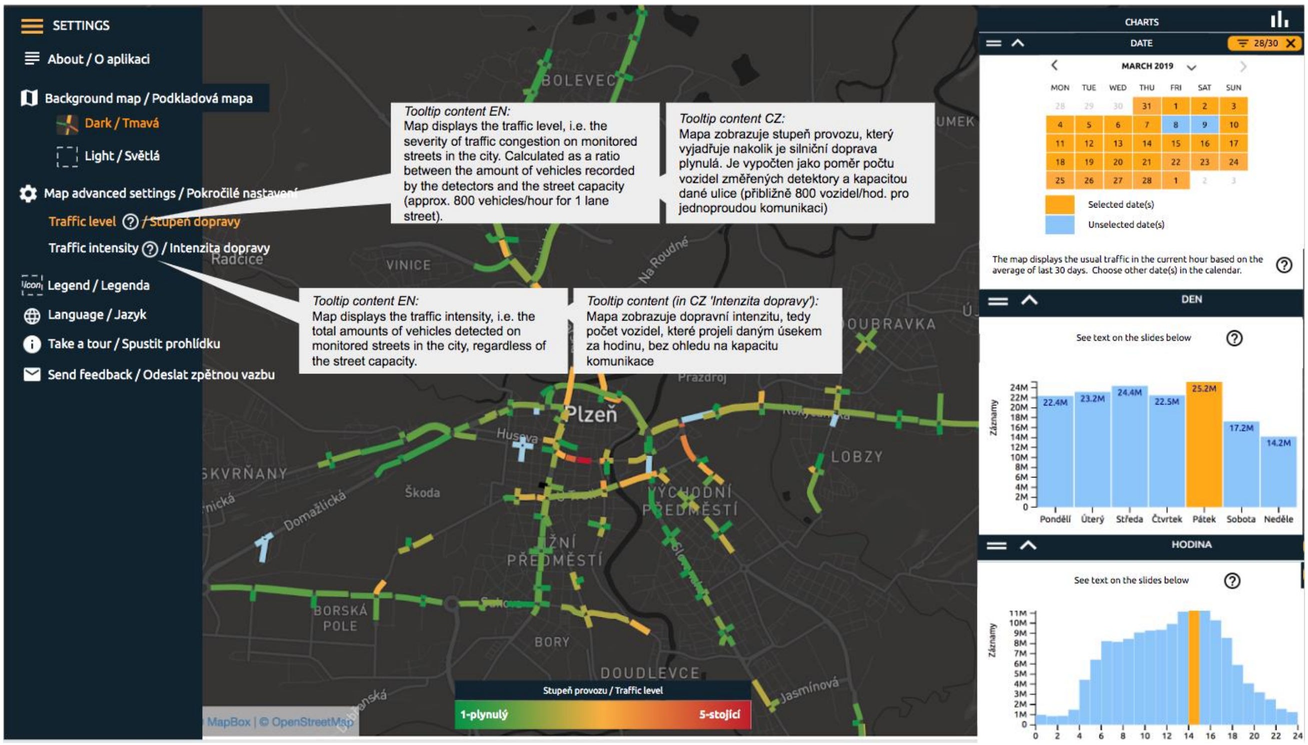

Figure 21: Traffic volume heatmap of Pilsen - Design mock-up 2

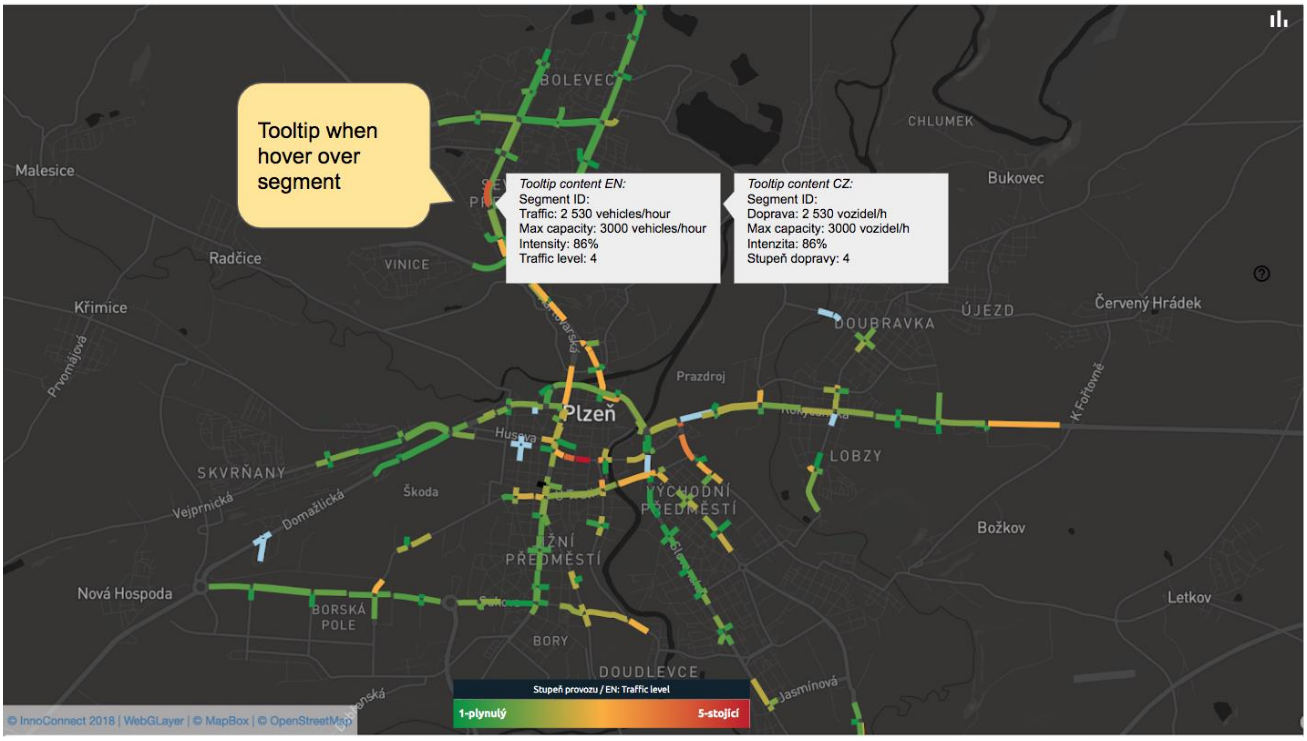

Figure 22: Traffic volume heatmap of Pilsen - Design mock-up 3

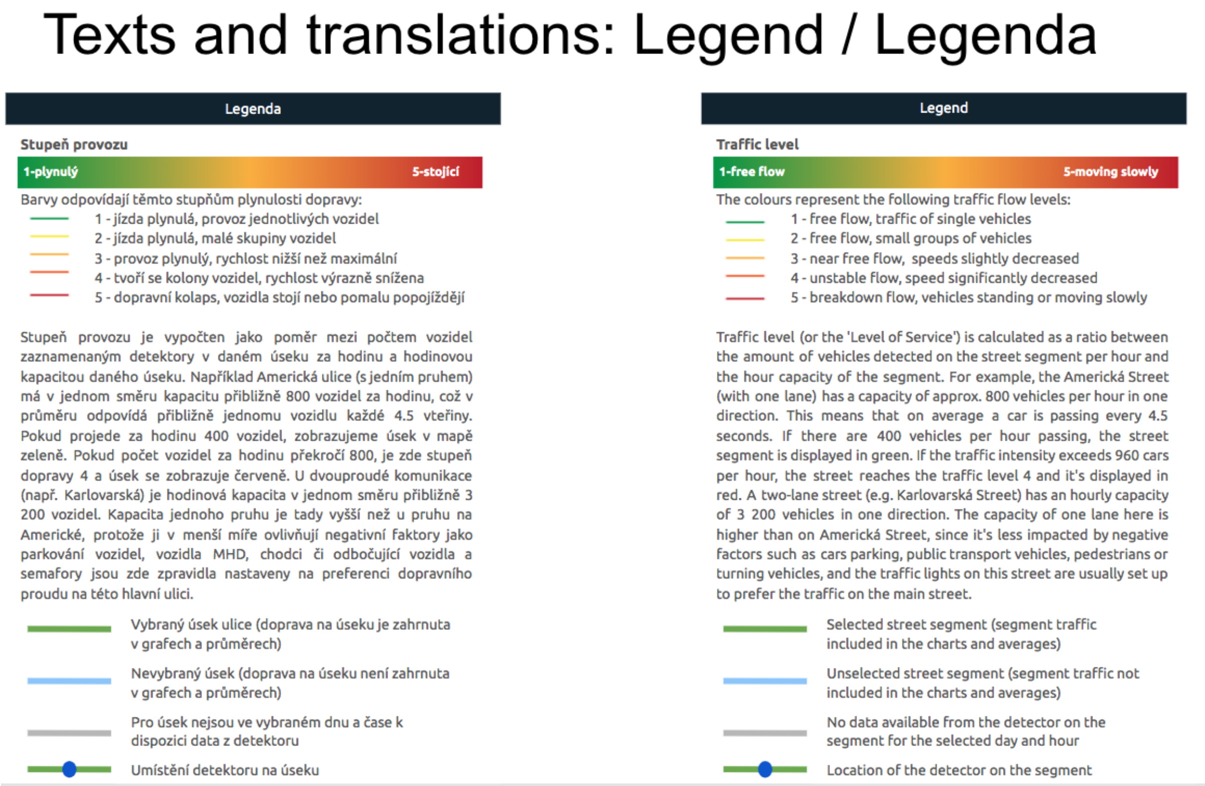

Figure 23: Traffic volume heatmap of Pilsen - Design mock-up 4

3.2.5 Traffic Incidents Heatmap for Pilsen (registered by Municipal Police)

The WebGLayer based heatmap of traffic incidents and offences registered by Municipal Police in Plzeň. The heat of the area is defined by the volume of historical traffic incidents.

3.2.5.1 Visualisation Goal

To explore and investigate traffic incidents (traffic accidents, offences and other police records) in time and space to discover patterns, correlations and extremities using aggregated data.

3.2.5.2 Visualisation Status

No updates on this app in since the D3.6 submission (November 2018). The map currently contains the Pilsen Municipal Police data for 2015 only (45 thousands incidents). The Municipal Police expressed its interest in the analytical map in case the data from the state Police can be included.

3.2.5.3 Future improvements

The City of Plzeň tries to negotiate with the state Police to provide its traffic accidents and criminality data that could be combined in the app with the Municipal policy data. The Police promised to cooperate and provide the data, next steps will follow.

3.2.6 Delay due to Traffic Map for Issy

3.2.6.1 Visualisation Goal

Showing travel losses on each road segment in the Issy and south Paris region compared to the ideal situation (fluid traffic) for specific periods and timings.

3.2.6.2 Data Preparation

Issy managed to get a deal with MediaMobile, resulting in the delivery of floating car data for the Paris region for a whole year. The dataset contains:[4]

- Size source data approx. 120 GB;
- 232.518 road segments, estimated traffic speed is provided every 3 minutes.

This results in 4E10 data points. On top of that we also received information about the individual probes driving around;

To visualise speed and/or time delays via a WMS-T or other web services, the road segment dataset and the estimated traffic speed dataset needed to be processed in order to comply with several data requirements imposed by commonly used spatial information processing tools: geographical information system (e.g. QGIS), database management systems (DBMS) (e.g. PostGIS, GeoMesa-Accumulo) and web service providers (e.g. GeoServer).

As such, the road segment geometries were converted to a readable WKT format. Furthermore, the dataset containing traffic speeds was restructured. This way, each record in the resulting dataset provides an estimated traffic speed for a variable time period (expressed in a ISO-standard date time format) for a given road segment. In contrast, the old format listed 480 speed estimations (e.g. one every three minutes) for each road segment.

Column1	Column2	Column3	Column4	Column5	Column6	Column7
linkId	01/10/2017 00:00:00	01/10/2017 00:03:00	01/10/2017 00:06:00	01/10/2017 00:09:00	01/10/2017 00:12:00	01/10/2017 00:15:00
12500001120926	0	110	0	0	0	0
12500001315080	0	74	75	66	66	66
12500001326752	0	0	0	0	0	0
12500000961407	0	90	90	90	90	90
12500001046265	0	0	30	30	30	30

Table 7: Traffic heatmap for Issy-Les-Moulineaux - Floating car data preparation original data format

	arc_id bigint	rec_speed integer	starttime timestamp without time zone	endtime timestamp without time zone
1	12500001204761	0	2017-10-01 00:00:00	2017-10-01 00:03:00
2	12500001204761	38	2017-10-01 00:03:00	2017-10-01 00:12:00
3	12500001204761	0	2017-10-01 00:12:00	2017-10-01 00:18:00
4	12500001204761	30	2017-10-01 00:18:00	2017-10-01 00:33:00
5	12500001204761	0	2017-10-01 00:33:00	2017-10-01 00:42:00
6	12500001204761	45	2017-10-01 00:42:00	2017-10-01 00:45:00
7	12500001204761	44	2017-10-01 00:45:00	2017-10-01 00:54:00
8	12500001204761	18	2017-10-01 00:54:00	2017-10-01 01:03:00
9	12500001204761	45	2017-10-01 01:03:00	2017-10-01 01:18:00
10	12500001204761	20	2017-10-01 01:18:00	2017-10-01 01:21:00

Table 8: Traffic heatmap for Issy-Les-Moulineaux - Floating car data preparation new data format

[4] https://docs.google.com/presentation/d/1xX8bFKdgOZ3YTespPQCwoaW_2SgKnSl_vB6xmOhtsl8/edit#slide=id.g49a93c0634_0_9 & https://drive.google.com/open?id=1EhJmhu6UKpD4ZTSFCbNBbL5HxzJw3WiL

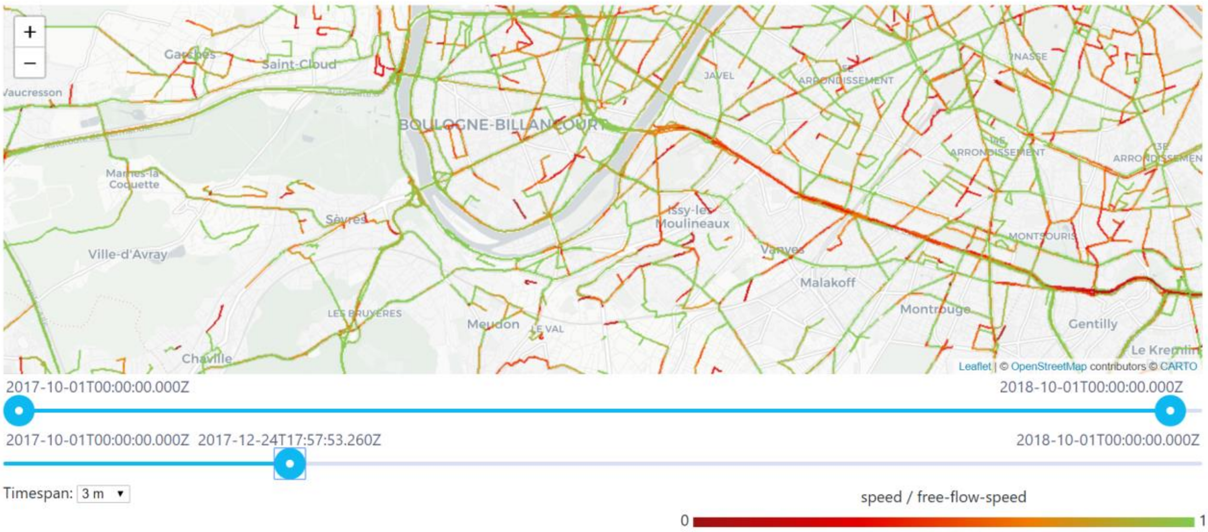

Figure 24: Traffic heatmap Issy-les-Moulineaux - map interface overview after iteration 2

3.2.6.3 Visualisation description and future improvements

After discussions with the pilot following the results of the latest iteration, it was decided to change the approach to move to a dashboard with a map as visualisation. A two-level approach will be followed to improve the existing interface towards a valuable traffic policy tool.

- Highest level: show traffic situation for the complete area with map and dashboard
  - Define the traffic situation category (worst case, average, modus, worst minute/hour - To Be Defined)
    - % deviation of time;
    - defining traffic jam levels (everything below that is not considered as a traffic obstruction);
  - Showing the data on traffic link-level (road segment(s) between important crossings), not on the level of a cluster of links. Displaying the data depending on the zoom level, we filter on road classification. Very congested streets are always shown on the dashboard.

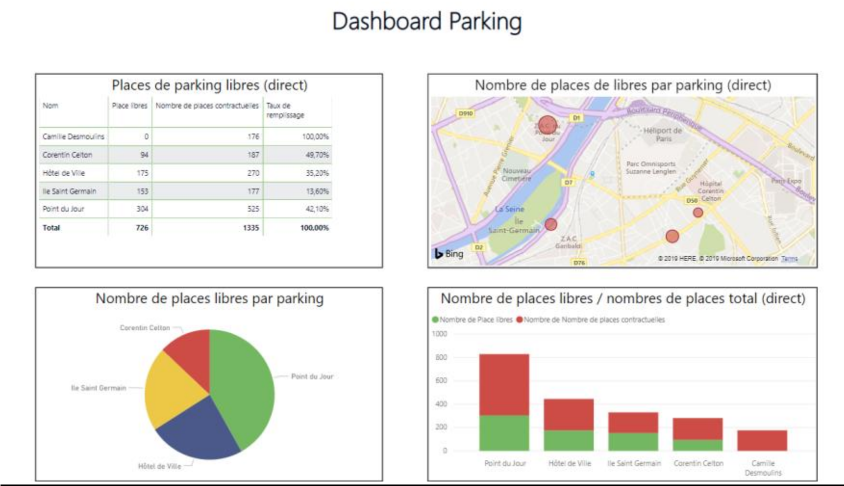

Figure 25: Issy-les-Moulineaux - Parking dashboard example

- On a detailed level:
    - ‘Classic WMS-T’ detailed visualisation on a street segment level;
    - detailed time selection interface;
    - possibility to add data on other mobility means later;
    - possibility to select date(s) on a calendar;
    - Detailed segment info, user can gather insights (time lost, av. speed, delta of speed) when selecting a single road segment.

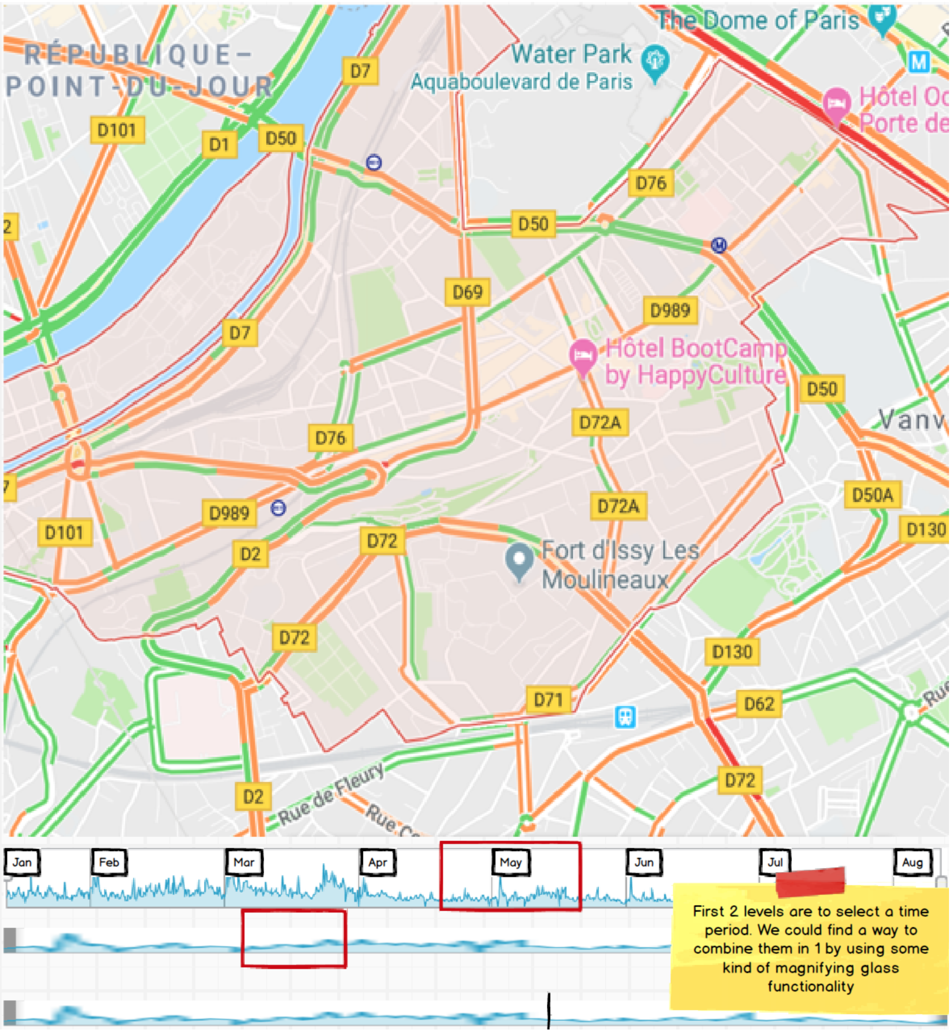

Figure 26: Traffic heatmap Issy-les-Moulineaux -Mock-up 1

At the bottom of the visualisation, it could be also possible to use a javascript component that allows zooming in on a graph, like Zingcharts[5] or AmCharts[6], as it can be seen in the examples below.

[5] https://www.zingchart.com/gallery/chart/#!infinite-zoom
[6] https://www.amcharts.com/demos/line-chart-with-scroll-and-zoom/

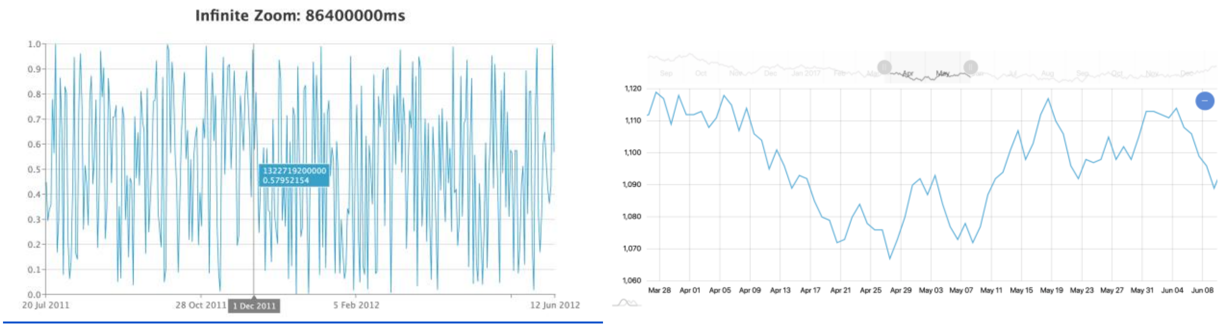

Figure 27: Traffic heatmap Issy-les-Moulineaux - Mock-up 2

3.2.7 Traffic Accident Heatmap for Flanders

3.2.7.1 Visualisation Goal

Show the locations where accidents happened in Flanders in the last 5 years. Showing relevant road accident information to find black spots due to road infrastructure or unadapted speed.

3.2.7.2 Visualisation Status and future improvements

Since the previous version of this doc, adding newer versions of the data has been planned, also including part of the latest 2019 data. The supplier has issues with the geocoding.

Based on feedback from users, the following updates of the map have been planned:

- Adding a layer with municipality boundaries. These can be used for municipalities to analyse the relevant accidents and facilitates integration in existing dashboards;
- adding a new layer with transportation regions;
- UI functionality improvements (tooltips to help users understand the functionality);
- documentation of the map improvement:
    - add information about data sources in the 'About' dialogue;
    - add Legend.

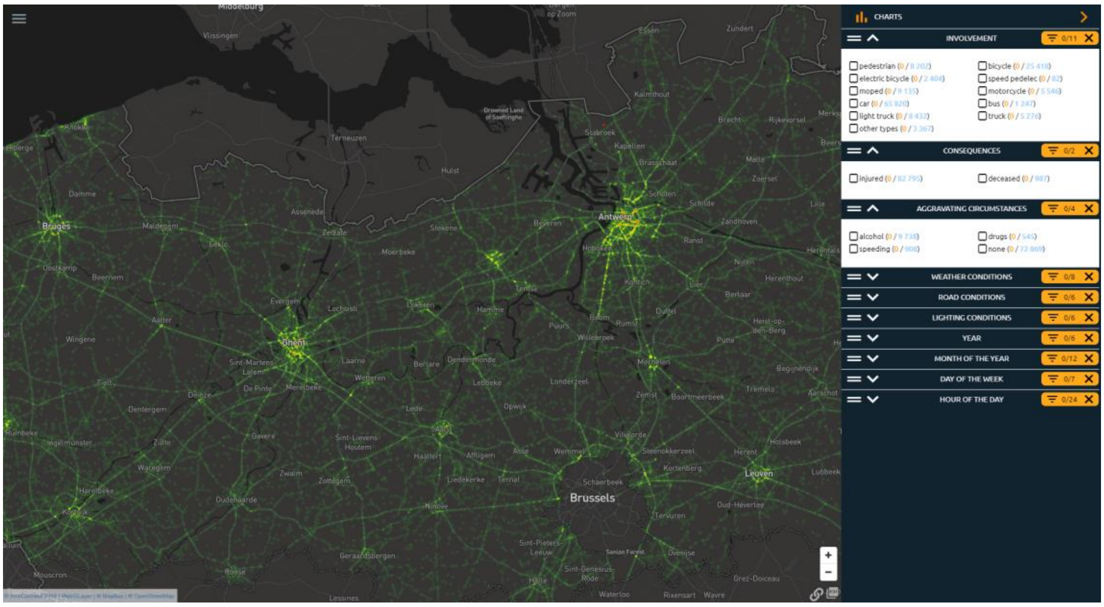

Figure 28: Flanders accident heat map - Map overview after iteration two

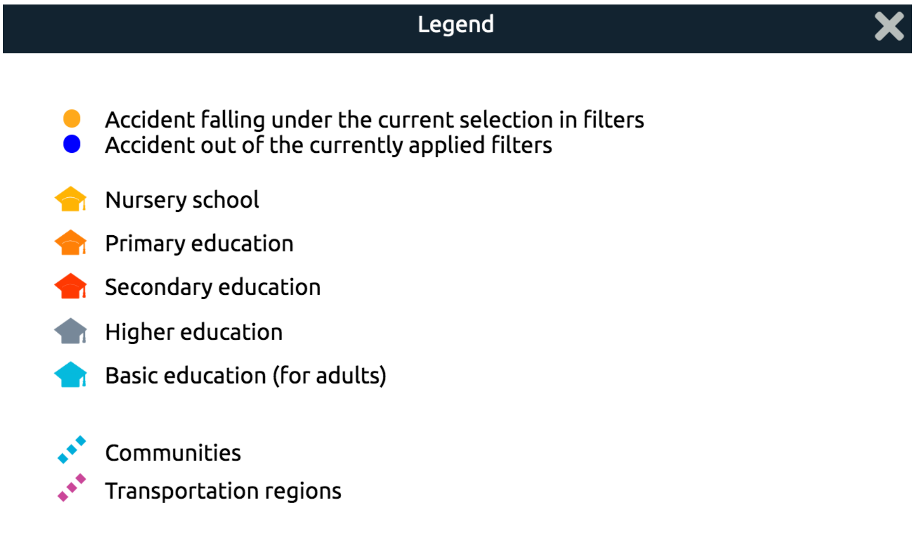

Figure 29: Flanders accident heat map - Legend overview after iteration two

3.2.8 Mobile telecommunication data visualisation in Ghent

Since the beginning of the PoliVisu project, Ghent wanted to visualise student displacement patterns. The mobile telecommunication data from one of the biggest telecom providers, Proximus, turned out to provide the most useful information. The Ghent team therefore planned several meetings and whiteboard sessions together with Proximus, which lead to a contract with the analytical team of Proximus to dedicate time to the student case, as described in D6.1.

3.2.8.1 Visualisation Goal

Display on a map of Ghent the number of students that are spending the evening (18:00-00:00) at a certain location.

3.2.8.2 Visualisation Status

In June 2019, the analytical team of Proximus presented a first version of the results and the visualisation. Proximus identified 3951 users as 'dorm students', over a period of 4 weeks. The 3951 users were identified by using a predefined pattern of a 'dorm student', which was:

- A user that was not in Ghent during the first week of the measurements (because there was a lesson-free week, after the exams, only for the university & university college students);
- a user that is not in the city during the weekend.

By using this pattern, Proximus defined the 'Most Likely Lodging Place'.

Currently the Ghent team and Proximus are fine tuning the results by creating a more detailed pattern for a 'dorm student'. We will also start the negotiations with Proximus to re-do this exercise at a later stage, so we can compare the results.

3.2.8.3 Visualisation description

The map below gives an overview of the student displacement measuring zones based on the location of the Proximus mobile phone receptors. The darker the cell, the more presence of possible dorm students.

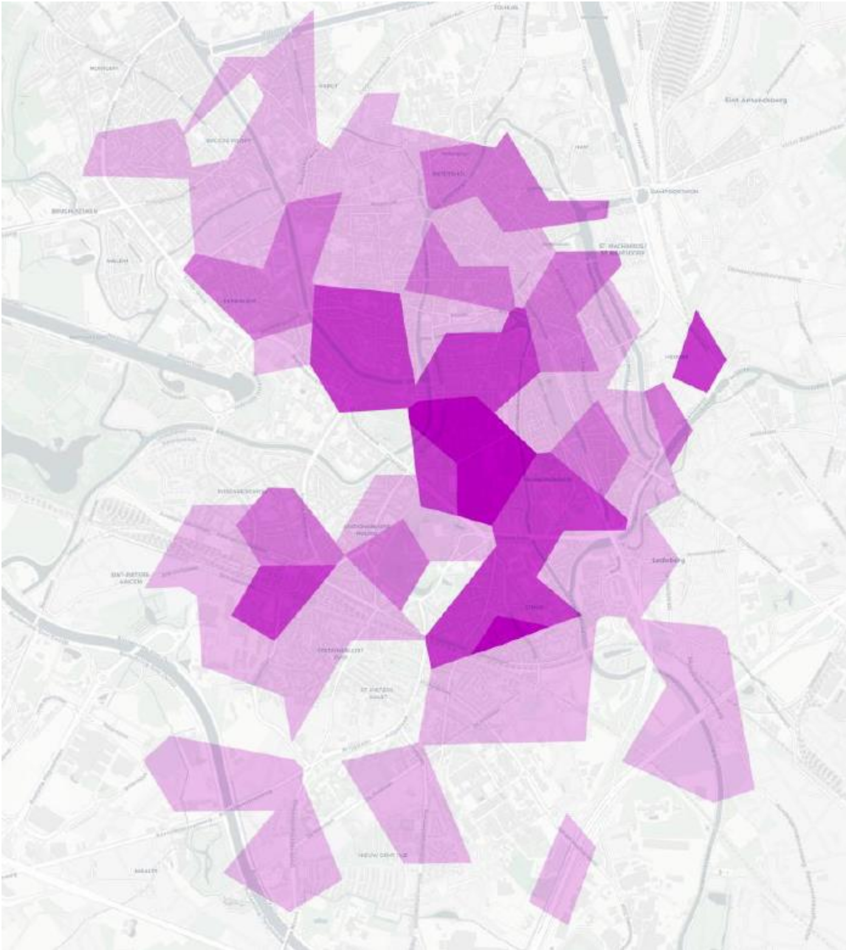

Figure 30: Ghent student locations based on telecommunication cellular data - presence of dorm students during the evening (6PM - midnight)

3.2.8.4 Future improvements

Ghent will elaborate the results of this visualisation with the student community and will visualize the results, when it turns out that they are reliable and trustful, in a permanent way.

3.2.9 Visualisation of traffic, roadworks and ANPR cameras in and around Mechelen

In a workshop with the pilot there was a need specified to have a user friendly view on the available traffic model for Mechelen from the Flemish government. As a result, a prototype of the Traffic Modeler is built for the whole province of Antwerp. Initial deployment of the province of Antwerp done was done as a proof of concept (http://app.hslayers.org/stm/apps/mechelen).

3.2.9.1 Visualisation goal

In a subsequent workshop, 3 use cases were identified:

1. Integrate Traffic Modeler with GIPOD, an API to capture events and roadworks;
2. Simulate impact of the planned extension of the area with reduced traffic in Mechelen;
3. Support the introduction of so-called Schoolstreets: areas around schools where traffic is banned during the hours children arrive in/leave school.

3.2.9.2 Data preparation and processing

An important step to make the Traffic Modeler useful for policy making is to connect the tool to GIPOD. GIPOD is a Flemish system that tracks occupation of public space taking into account roadworks and other factors. We are investigating what the best API calls are for this use case, our current view is these 2 calls are best:

- Webservice displaying events: http://api.gipod.vlaanderen.be/demo/manifestation.html
- Webservice displaying work assignments (planned public works): http://api.gipod.vlaanderen.be/demo/workassignment.html

The availability of an integrated data service allows us to integrate these services into a business process. High level: we see 2 moments a city official may want help from STM: before registering the hindrance in GIPOD, so as part of the decision making process to accept the request and to make sure the impact is acceptable (this will be done for cases where big hindrance is expected) and after registering in GIPOD so the expected impact is planned for future decisions. Below BPMN describes this.

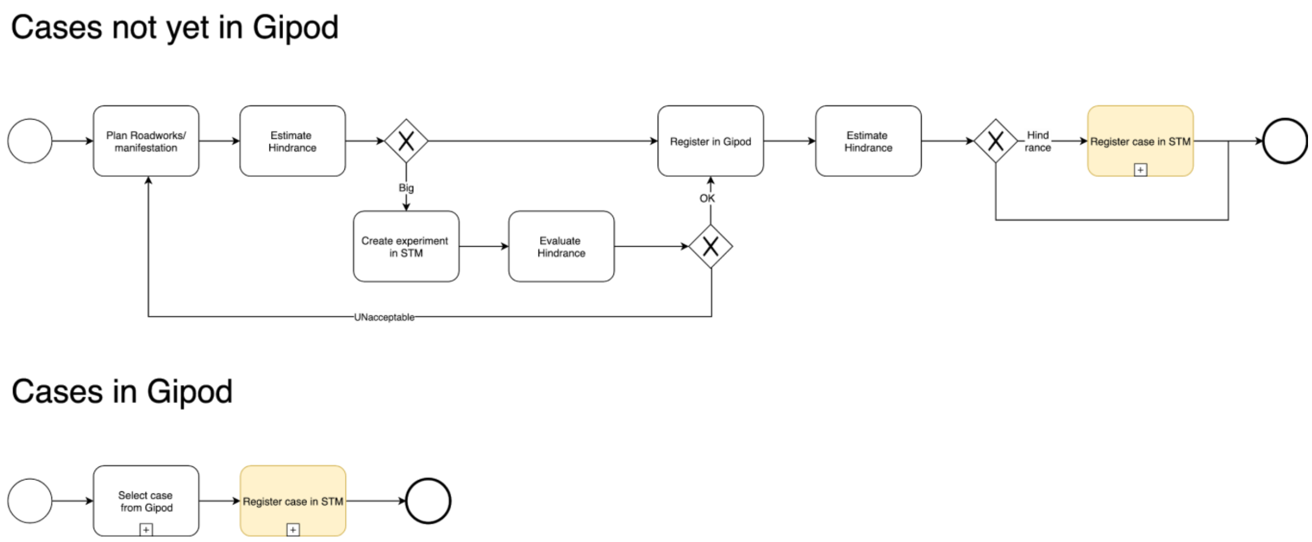

Figure 31: Mechelen traffic model visualisation - Road works database integration process

3.2.9.3 Visualisation description and future improvements

Initial mock-ups have been created to give an indication of what would be required. There still needs to be a discussion with the development team as well as Mechelen and other end-users.

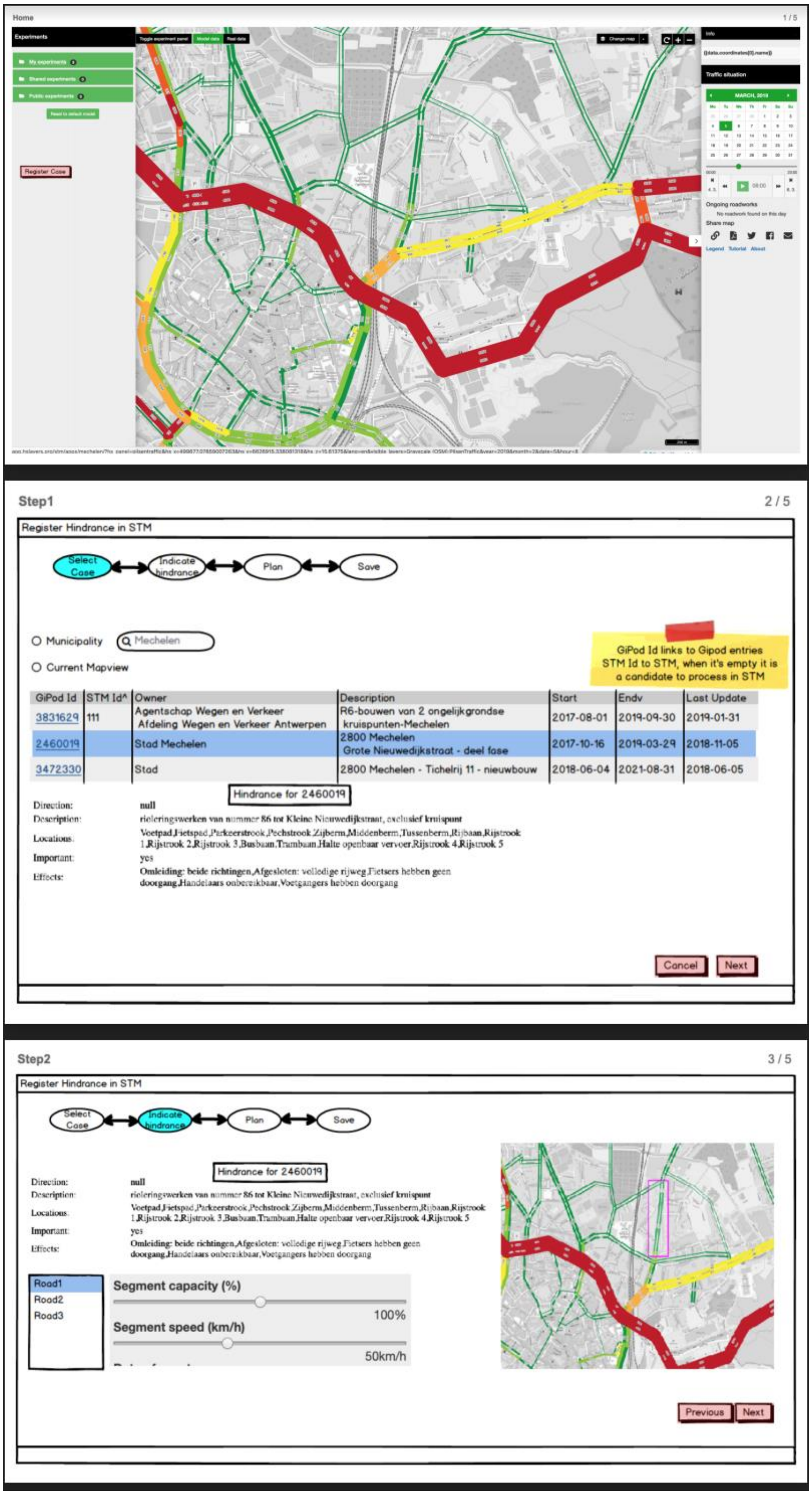

Figure 32: Mechelen traffic model visualisation - Road works database integration - Mock ups 1

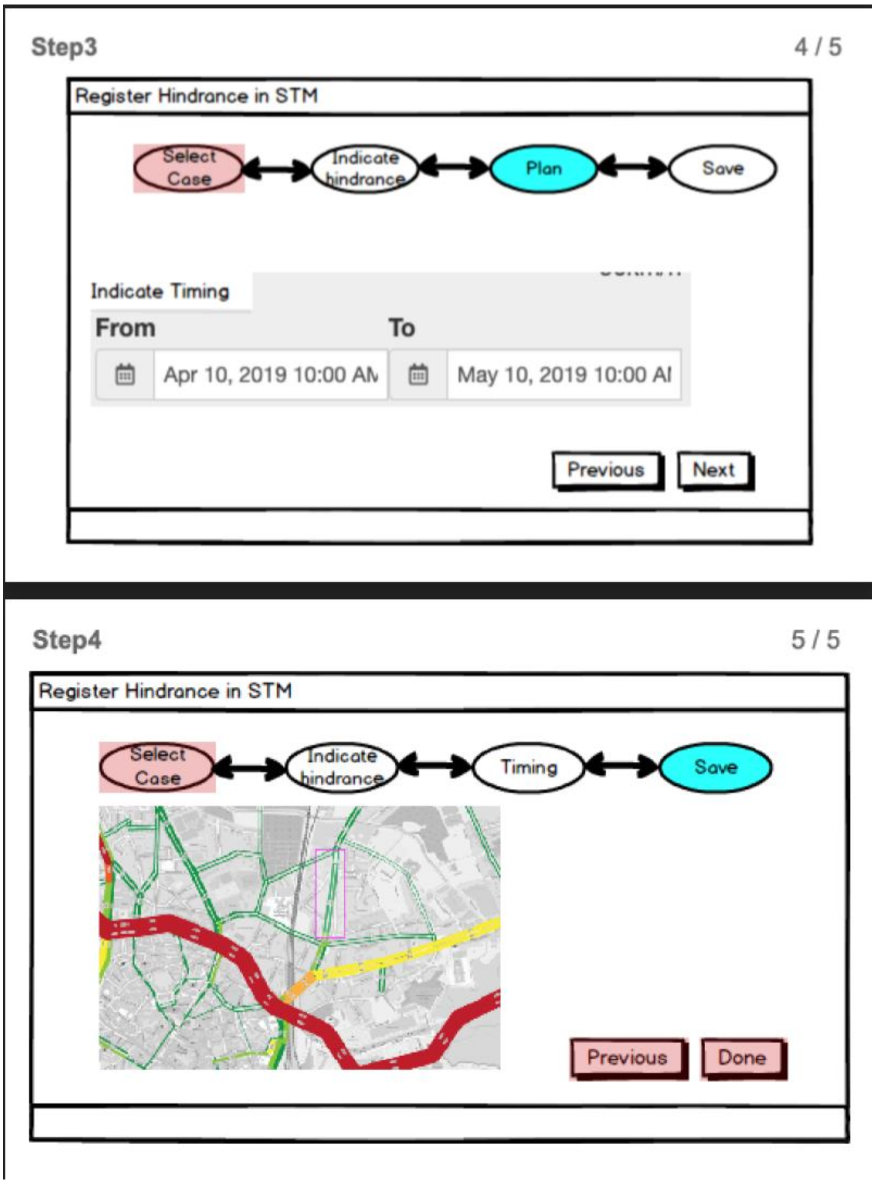

Figure 33: Mechelen traffic model visualisation - Road works database integration - Mock ups 2

Simulate impact of the planned extension of the area with reduced traffic in Mechelen

It’s planned to extend this area in Mechelen. In the past, this was done based on an educated guess what the impact will be on the traffic. It is an ideal case to introduce evidence and visualisation based policy making in Mechelen. We plan to experiment with the Traffic Modeler and assess the impact of adding streets to the reduced traffic area.

Dependency: very accurate traffic model, so we need to enrich and finetune the model using ANPR data.

Support introduction of so-called Schoolstreets

Schoolstreest are areas around schools where traffic is banned during the hours children arrive in/leave school. Also here, no detailed traffic impact analysis has been done yet. The idea is now to use ANPR cameras and loops counting traffic to assess the impact of introducing extra schoolstreets.

To assess the effectiveness of the schoolstreets, the impact on the number of pedestrians and cyclists in the immediate neighbourhood of the schools and the impact on all traffic in the wider vicinity will be visualised before and after the introduction. This way, we can identify whether there indeed is a modal shift from traffic by car to pedestrian or bicycle traffic. Identified data sources are mobile loops counting traffic placed in strategic locations, ANPR cameras, weather data (to factor in weather impact).

3.2.10 Parking data visualisation in Kortrijk

A couple of workshops took place with the Kortrijk team to come to visualisation goals.

- Assumptions:
    - A: 20% of users are parking for longer than 2 hours → this is derived from the ticket sales through smartphones. Users that use the vending machines are not taken into account;
    - B: 25% are parking without using a valid ticket. → sample check in the field.
- Main request: **verify these assumptions in a cost efficient way as right now the process is very labour intensive with a lot of manual work involved.**

Candidate Visualisation goals:

For A:

- Show for defined parking spots and parking zones the sum of the amount of time a car stays longer than 2 hours for a defined period (day, week, month?);
- show a graph showing the total amount of time a car stays longer than 2 hours per day;
- show a graph indicating the total number of times a car stays longer than 2 hours per day.

For B:

- Show the amount of money not paid for a certain period
    - How to measure?
    - during the brainstorm we decided to skip this, as the means to measure this in a cost efficient way seem to be missing.

During these workshops the available datasets related to parking have been documented:

- Parking data
    - On street parking:
        - Shop and go and Residents
            - Occupied and free spaces
            - Overstay time, frequency,...
        - Enforcement
            - Verification
                - Xy (location of parking guard is stored, not of the parking spot)
                - Fine xy (+ type)
                - Time
        - Parking meters
            - Count of tickets
            - Time
            - Duration
            - Zone (parking meter is in a user defined zone)

- Off street parking
    - Occupation
    - Duration
    - Nr of entrance, exit
    - Payment
    - Time
    - Xy of parking
- Parking guidance (BeMobile)
    - Scenarios to influence behaviour
    - When did some scenario get triggered
    - How did it get triggered (manual, automated)
- ANPR
    - To restrict access to shopping area or resident area (only residents allowed, no transit)
    - License plate
    - Picture
    - Time
- Traffic density data
    - Check with mobility department

Due to changing priorities, Kortrijk has indicated it wants to put the effort related to Polivisu on hold, so no progress has been made since.

4 Conclusions

PoliVisu tested and experimented with many visualisation methods to visualise mobility-related data in a policymaking context. However, the used methods on itself are not entirely new; the context of using these techniques for dynamic visualisations are innovative. These dynamic visualisations allow users to select data, experiment with data and to elaborate policy problems, the existing policy impact or future policy impact when forecast and modelling techniques are used.

The functional design itself was part of policy analysis with a clear definition of the visualisation goals, collecting and analysing the needed data and choosing and refining the needed visualisation techniques. The overview table can be found in chapter 3. The summary table below provides an overview of the used techniques and tools related to the current pilots and future pilot developments.

		Pilots (current)	Pilots (future)	Techniques	Software	Modelling visualisation	Datasets	Interactive	Extra info
Story telling with data									
		Flanders	Flanders	Story telling tool	Pageflow	-	-	-	
Data visualisation									
Traffic flows									
Road traffic									
	Live data	Pilsen	Pilsen	Line map (color and width)	HSLayers & WebGLayer	-	Live traffic sensors	Heatmap selection	
	Historic data	Gent, Issy, Pilsen, Mechelen, Flanders	Gent, Issy, Pilsen, Mechelen, Flanders	Line map (color and width)	HSLayers & WebGLayer Warp10 based prototype	Traffic Modeller (OSS)	Floating car data, traffic model data, cellular data, accident data	Heatmap selection & scenario modelling	
	Predictive data	-	Pilsen, Mechelen	Delta map (color and width)	HSLayers	Traffic Modeller (OSS)	Traffic model data, live traffic sensor data, GIPOD	Heatmap selection & scenario modelling	
Localisation									
Students									
	Historic data	Gent	Gent	Zoning map	GIS tool (OSS)	-	Cellular data	-	Can be used for other target groups
Accidents									
	Historic data	Flanders, Pilsen	Flanders, Pilsen	Point based heat map	WebGLayer	-	Accident data	Heatmap selection	
Road works									
	Predictive data	-	Pilsen, Mechelen	Delta map (color and width)	HSLayers	Traffic Modeller (OSS)	Traffic model data, live traffic sensor data, GIPOD	Heatmap selection & scenario modelling	See Road traffic - predictive data

Table 9: PoliVisu tools, techniques, dataset types and software components

The described interactive visualisation tools are not a complete list, but several highly useful tools that are used in practice and will be further refined during the next testing cycles and iterations.
These iterations are essential to test the current and future visualisations in terms of user interface and their applicability for policymaking purposes. Regarding the policymaking testing, the input from citizens and policymakers are both needed because the visualisations must play a role in further establishing co-creation and co-design between citizens and policymakers.
Using prediction models to calculate and visualise the delta between two moments in time (in the past or the future) offers a new range of visualisation techniques and means also specific challenges regarding data literacy. Even professional users and also non-domain expert policymakers and citizens are not used to this kind of visualisations, and special attention to testing the new developments is needed.

In the next stage of PoliVisu, the described improvements will be built and tested. A number of the existing applications will be further tested as described above. Special attention will go to further integrate the visualisation with policymaking initiatives in the pilot cities and testing with the defined target groups in each of the pilots.

5 Annex - Updated Technical Specifications

5.1 Introduction

The PoliVisu project has chosen a pilot-driven and iterative approach in which the first pilot cycle focuses on exploration, analysis and design of solutions supporting policy making leveraging big data and advanced visualisations (cf. 4 pilot cycles as described in D7.1 Evaluation Plan). User requirements and functional specifications will evolve throughout the project. Some technical specifications and parts of the architecture will also evolve, reflecting new requirements, priorities, insights and lessons-learnt.

The DoW only foresees one formal deliverable D4.1 Technical Specifications. Any subsequent and substantial updates to the technical specifications would be provided as an annex to the Policy Experimentation and Functional Design deliverables.

As part of D3.7, the technical team presents an updated technical specification for (1) a Big Data Management component (Warp10 and Geomesa) and (2) Toolbox and metadata component.

5.2 Big Data Management

Loading all floating car data in WebGLayer is not going to be possible. A couple of scenarios were discussed to make WebGLayer work with the data (filtering, dynamic loading), but it became clear that there is no good fit between the data and the application

Figure 34: Annex - Floating car big data integration

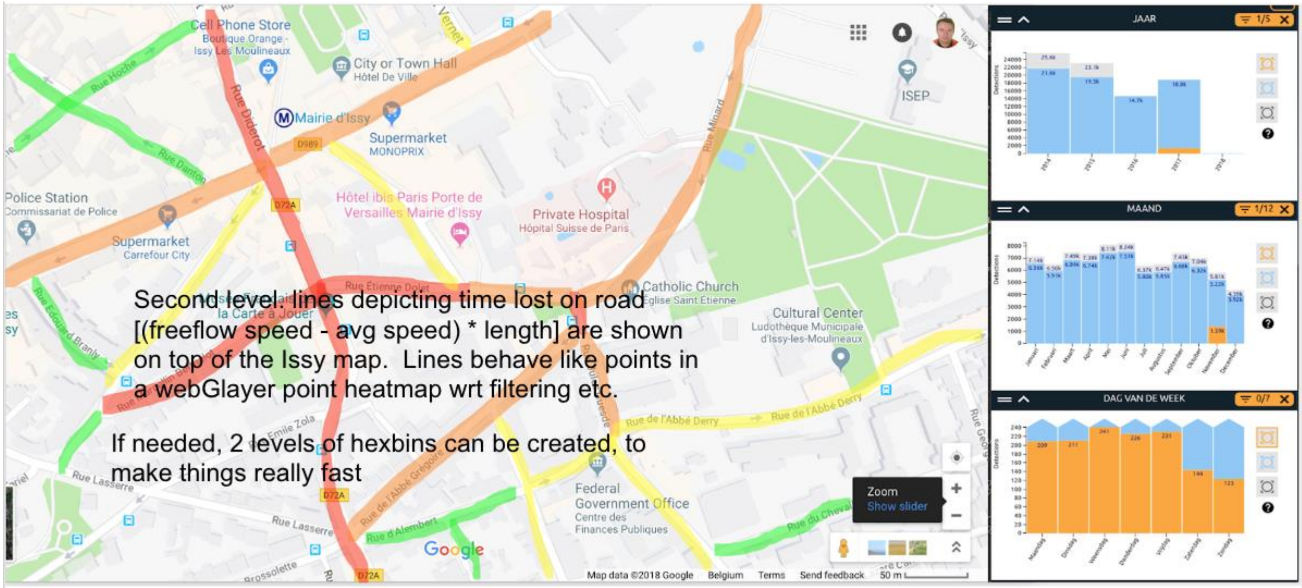

Figure 35: Annex - Floating car data - Road categorisation test

Loading all the data in a PostGIS database and building through GeoServer a WMS-T on the data showed that this solution cannot work with all the data. Since we did not want to go from Big Data to small data, we decided to do some prototypes with alternative backends, load the data and serve it out as WMS-T: using Warp10 and GeoMesa.

The most important tables of the MediaMobile data are depicted in the tables below:

Table 2 : informations du fichier Links.csv

Champ	Type	Description
id	long	Identifiant unique
length	int	Longueur en mètres
ffs	int	Vitesse en conditions fluides (free flow speed) en km/h, estimée par Mediamobile à partir de la distribution des vitesses sur l'arc
speedlimit	int	Vitesse maximale autorisée, en km/h. Vaut 0 pour NA.
frc	int	Functional Road Class, détaillé ci-dessous
netclass	int	Caractérise l'importance de l'arc dans le réseau, détaillé ci-dessous
fow	int	Form Of Way, détaillé ci-dessous
routenumber	string	Numéro de la route
areaname	string	Nom de la zone
name	string	Nom de la route
geom	string	LineString. Suite de (latitude,longitude) séparés par des barres verticales |. Décrivent la géométrie de l'arc.

Table 12 : informations des fichier Observations

Champ	Type	Description
vehicleId	string	Identifiant unique (jour) de l'observation
linkId	long	Identifiant unique de l'arc
coverage	double	Couverture de l'arc par l'observation. 1 si le véhicule a parcouru le segment de route en entier. <1 sinon.
timestamp	dd/mm/yyyy hh:mm:ss	Horodate d'entrée sur le segment de route
speedInKph	int	Vitesse estimée du véhicule sur l'arc

Table 14 : informations des fichier Vitesses

Champ	Type	Description
linkId	long	Identifiant unique de l'arc
dd/mm/yyyy hh:mm:ss 480 colonnes (3 mn)	int	vitesse estimée sur l'arc à l'horodate donné. Vaut 0 pour NA.

Figure 36: Annex - Media mobile floating car data structure

5.2.1 Geomesa based big data management

GeoMesa is an open source suite of tools that enables large-scale geospatial querying and analytics on distributed computing systems. GeoMesa provides spatio-temporal indexing on top of the Accumulo, HBase, Google Bigtable and Cassandra databases for massive storage of point, line, and polygon data. GeoMesa also provides near real time stream processing of spatio-temporal data by layering spatial semantics on top of Apache Kafka. Through GeoServer, GeoMesa facilitates integration with a wide range of existing mapping clients over standard OGC (Open Geospatial Consortium) APIs and protocols such as WFS and WMS. GeoMesa supports Apache Spark for custom distributed geospatial analytics (https://www.geomesa.org/).

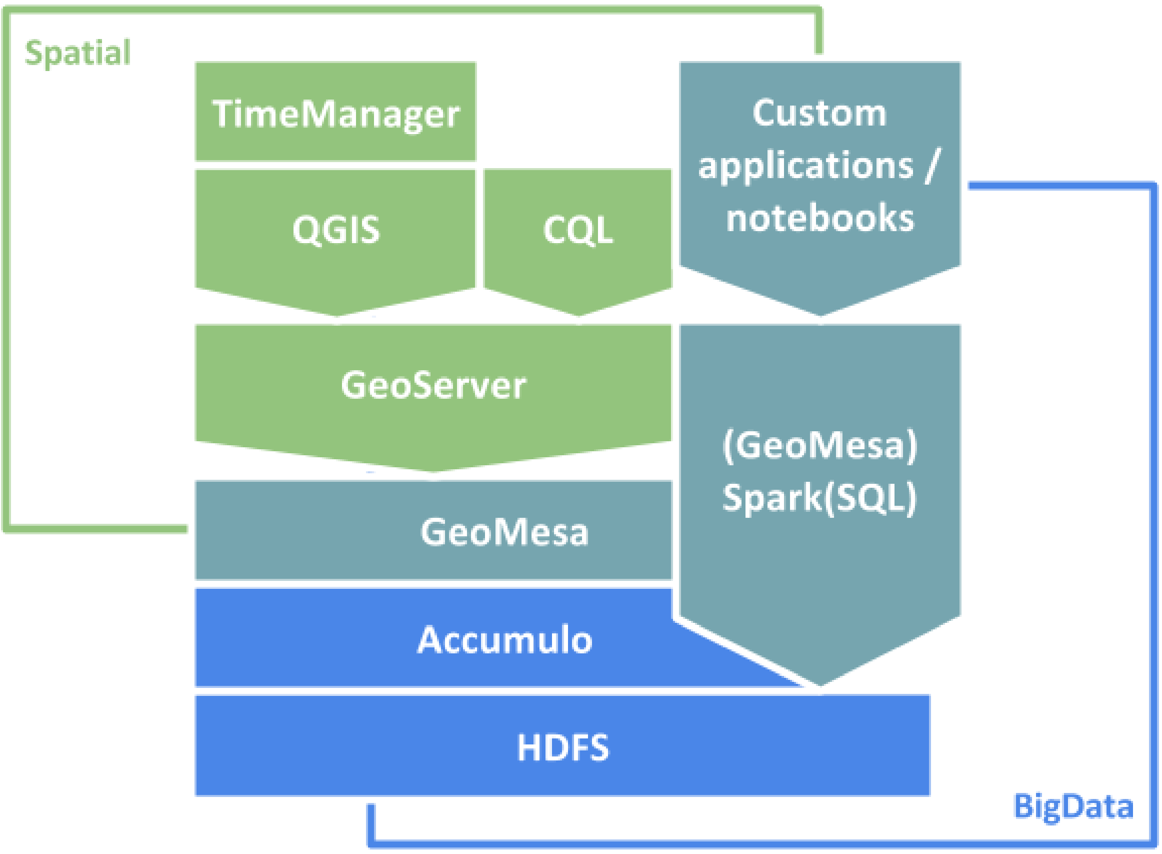

Figure 37: Annex – Geomesa big data spatial visualisation technical approach

Accumulo, the underlying big data database, is a NoSQL database. As such, it is not possible to create views or (spatially) joined tables as you would do in a standard DBMS. Therefore, the different datasets (i.e. geometries, speed recordings) must be combined into a (new) single dataset, whereby relevant attributes such as speed and time delays are calculated, before ingesting the data into the Accumulo database. This was done with custom made Python scripts.

GeoMesa is available for several platforms, mainly Linux distributions. We have chosen the GeoDocker - GeoMesa option that creates a GeoMesa cluster running on Accumulo database. The Docker container platform allows us to setup and configure GeoMesa and ship it all out as one package. By using Docker, we can be assured that the application will run on any other machine regardless of any customized settings that it might have. Docker provides repeatable development, build, test, and production environments. This is important in our case, where we want to keep track of the different settings and share different instances of GeoMesa Accumulo (e.g. setting up a test and an integration environment).

The GeoDocker - GeoMesa can be installed on a cluster locally (on a single machine) and on an Amazon Web Services (AWS) cluster. We have investigated both options. Setting up an Amazon AWS cluster required an Amazon Elastic Compute Cloud (Amazon EC2) instance of high compute capacity (at least Linux m5.large) and additional storage space. Attempts to install the GeoDocker on inferior EC2 instance's have failed (connection with the server could not be maintained). We have decided to proceed with the single machine installation. The main reason was that we had a virtual machine of high specifications available for testing (32 GB RAM) and that we wanted to focus on the setup of GeoMesa itself instead of spending additional time to get familiar with the Amazon ecosystem.

We have installed the latest GeoDocker - GeoMesa official release (2.2.0). We have encountered a number of technical issues with older, non official Docker releases. Following the instructions on the GeoDocker site, we overcame a number of known issues. Remote access has been allowed to a number of users (console access) and respective user accounts have been created. To enable file upload, ftp access has also been enabled. A GeoDocker instance, hosted on ATC's premises, has been used on the initial round of experiments. After a period of time, it was evident that GeoDocker needed significant hardware resources and only GeoDocker related processes should be run on the server.

The GeoDocker's documentation provides instructions and generic examples. The available data had to be converted to an appropriate format and certain features types had to be defined. Configuring the appropriate setting has been challenging and resulted in the use of the GeoJSON format as input. GeoJSON supports encoding of a variety of geographic data structures and while it is not the optimal input format for data ingestion, it is easier to manage and use. A preprocessing of the available data is required to generate a GeoJSON file. Due to the amount of the Issy visualizations data, this process requires time and results in a file around 4GB of size (that corresponds to input for a certain day or around 14 million features).

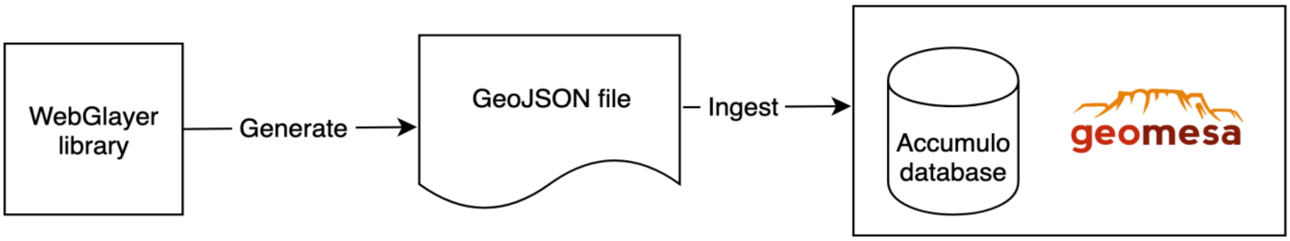

Figure 38: Annex - Media mobile floating car data management approach

On the GeoServer, a collection has been defined and several web services (WMS, WPS, WFS) have been enabled. The GeoJSON input file is uploaded on the server and copied inside the docker container. The accumulo ingestion command is executed with a convert and a feature definition file as parameters. The ingestion progress is displayed on the console. The process finishes with a message displaying the numbers of features ingested into the collection and the overall time. Data visualizations can be viewed on the GeoServer Layer Preview. The user can set filters and apply queries to get an updated visualization.

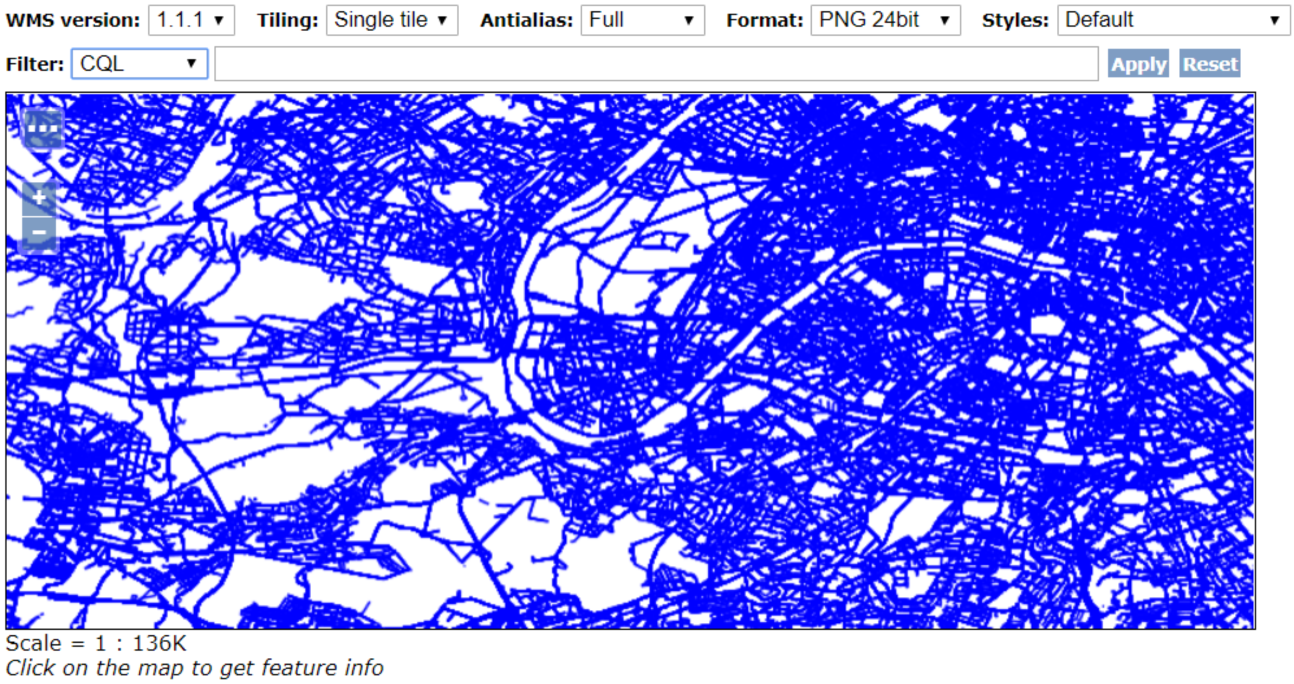

Figure 39: Annex - Media mobile floating car data visualisation

In order to process with testing and at the same time have a stable environment available, we had to install a second instance of GeoDocker. This has been done on a private Virtual Private Server (VPS) environment. The VPS server uses additional external storage which is expandable to up to 512 GBs (1 TB on ATC's instance).

After a number of ingestions, certain limitations have arisen. The ingestion of 4 GB of GeoGSON data can require more than 2 hours and that time increases in the subsequent ingestions (exceeds 3 hours). Furthermore, according to the current setting, the disc space required for that amount of data (around 14 millions features) is around 20 GBs. While this can be normal for a platform that handles big data and creates relations (indexes etc) to allow certain operations, it poses restrictions (related to the available disc space) on the amount of data we can ingest with the available resources. Optimizing GeoMesa configuration could leverage these restrictions and we are investigating a number of available options (e.g. ingesting another input file type, reducing the number of indexes on the input data).

5.2.2 Warp10 based big data management

Warp 10™ (https://warp10.io/) is an open source time series platform developed by SenX™ (https://senx.io/). This platform is built to manage and simplify time series processing. It includes a Geo Time Series™ database and a companion analytics engine. Geo Time Series are a sequence of measurements, optionally geo-localized, indexed in time. Warp 10 shines where it comes to store and analyze big time series with location datasets such as the MediaMobile floating car data. It can also greatly simplify the analysis of "not-so-big" datasets if it consists of measurements depending on time and space.

The two main goals of Warp 10 is to be optimized, to be able to handle billions of data points, and to be flexible, to adapt to many different needs. In the case of the Issy-les-Moulineaux pilot, there are two main needs: exploring the data in an interactive way and extracting concise information from the whole dataset. These two goals can be tackled with Warp 10 but in a different technical way.

For the analysis of the whole dataset, the most efficient way is to run a distributed analysis on the raw data files, which can be done with a combination of Spark™ and WarpScript™. WarpScript is a language dedicated to the analysis of Geo Time Series and is made available with Warp 10. As the analysis of the whole floating car dataset is still very much a work in progress, we won't give more technical details on this.

Interactively exploring the data require the data to be readily available with no or little preprocessing and last minute selection and aggregation depending on the requested perimeter. This can be done with the Warp 10 storage engine and a simple layer of analytics, using WarpScript, to select and aggregate. This process, which was tested and validated, is described in the following.

The first step is to store the data contained in the CSV files into the Warp 10 storage engine. We chose a flexible data modeling to address a wide range of not-yet-known problematics. This will have the drawback not to be optimized for specific applications:

- Road segments definitions are stored in a single Geo Time Series
- Raw GPS data is stored in multiple Geo Time Series, one per unique car identifier
- Map-matched GPS data is stored in multiple Geo Time Series, one per unique trip identifier
- Speed data is stored in multiple Geo Time Series, one per unique road segment identifier

Simple python script take care of reading the data from the CSV files and sending the result to Warp 10 storage engine.

In order to optimize data fetching for a visualization of road segment speeds, another Geo Time Series has been created giving, for a specific timestamp and a road segment identifier, the ratio between the speed at that time and the free flow speed, both for that road segment.

Finally, for applications to access the stored data, we created two new APIs for Warp 10. This is possible with the HTTP plugin where we can define WarpScripts handling requests on specific URLs and the Processing functions allowing Warp 10 to generate images. The first API implements the WMS-T standard and the second API implements a tile server of the form http://endpoint/$start/$end/$z/$x/$y, start and end being ISO8601 dates.

To access those API, one can use QGIS for the WMS-T API or a library in the like of Leaflet to render the tiles generated on the tile API. For the latter, a simple implementation is available online at https://warp.senx.io/standalone/http/polivisu/issy/traffic.

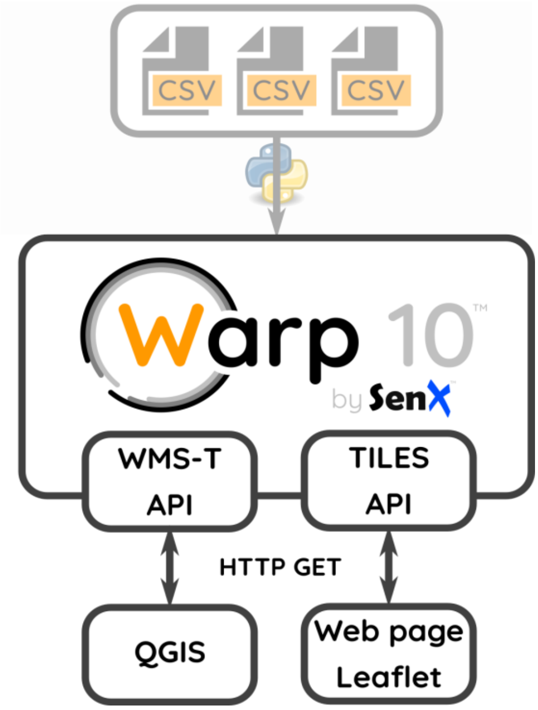

Figure 40: Annex - Warp 10 technical design overview

5.3. Polivisu metadata catalogue (Micka)

The Micka integration to the PoliVisu toolbox is realised through the OGC Catalogue Service for Web (CSW) in version 2.0.2 API. More specifically, the following operations are supported (see the OGC implementation specification for full definitions):

- GetCapabilities,
- GetRecords,
- DescribeRecords,
- Harvest,
- Transaction.

Searching in the PoliVisu catalogue is realised through the lightweight JSON API that represents a more performance efficient solution in comparison to the default XML-based API. The JSON API is an extension of the standardized CSW 2.0.2, however it remains backwards compatible to the CSW 2.0.2 implementation specification.
For instance, see the Figure X as an example of PoliVisu catalogue functionality when using the lightweight GetRecords query (https://dev.bnhelp.cz/projects/micka6/csw/?query=AnyText%20like%20%27*radon*%27%20and%20type%3Dservice&elementsetname=brief).

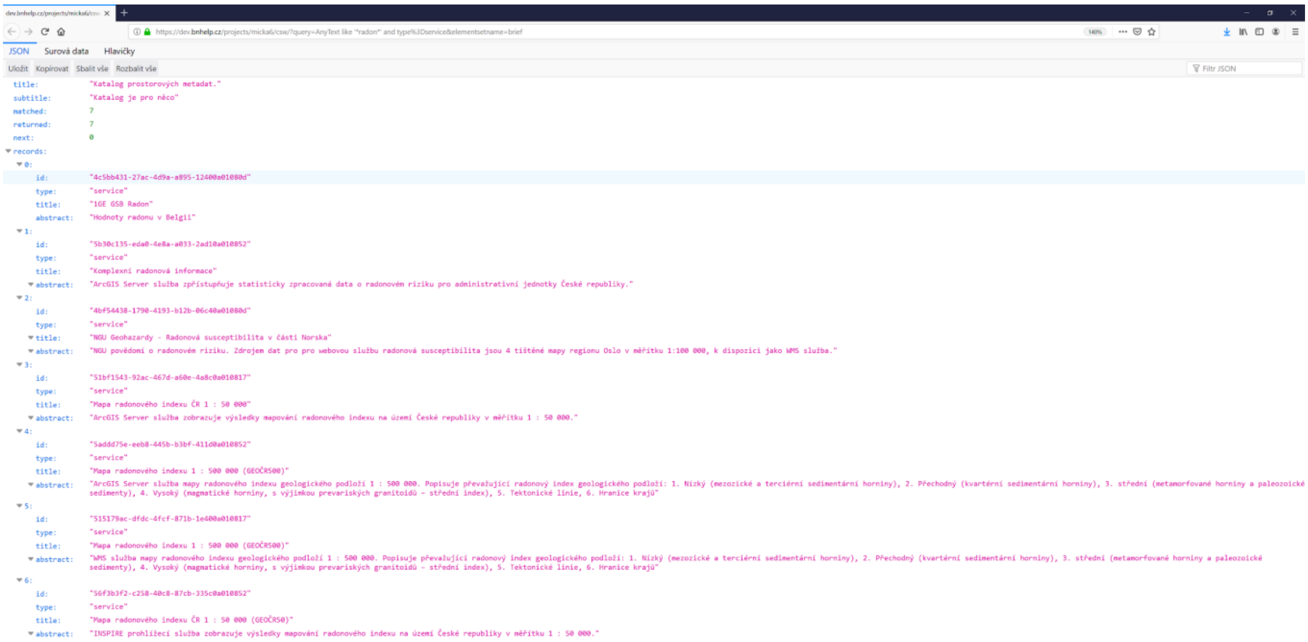

Figure 41: Annex - Example of a JSON GetRecords response in the lightweight API of the PoliVisu catalogue (Micka).

Standardized CSW 2.0.2 transaction operations (insert, update, delete) are supported to enable the management of all the items in the PoliVisu catalogue.
Moreover, the extended features, as defined by the INSPIRE directive (2007/2/EC) are supported as well.
The Polivisu catalogue, based on (Open)Micka, has also a Schema.org compliant API for publications to mainstream full-text search engines, such as Google, Yahoo, Bing etc. Such publication is based on the GeoDCAT-AP, as depicted in Figure Y.

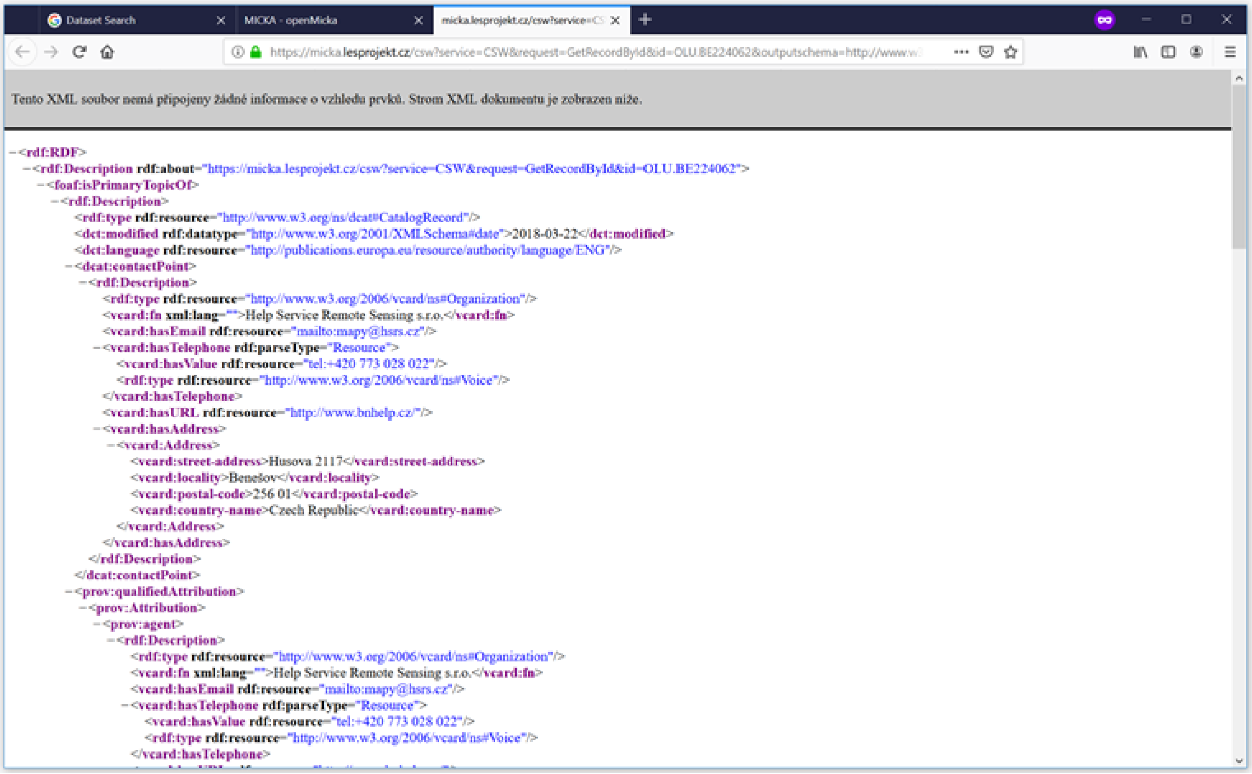

Figure 42: Annex - RDF response to a GetRecords operation request that is consumed by the Schema.org- based mainstream search engines, including Google as depicted in the next figure

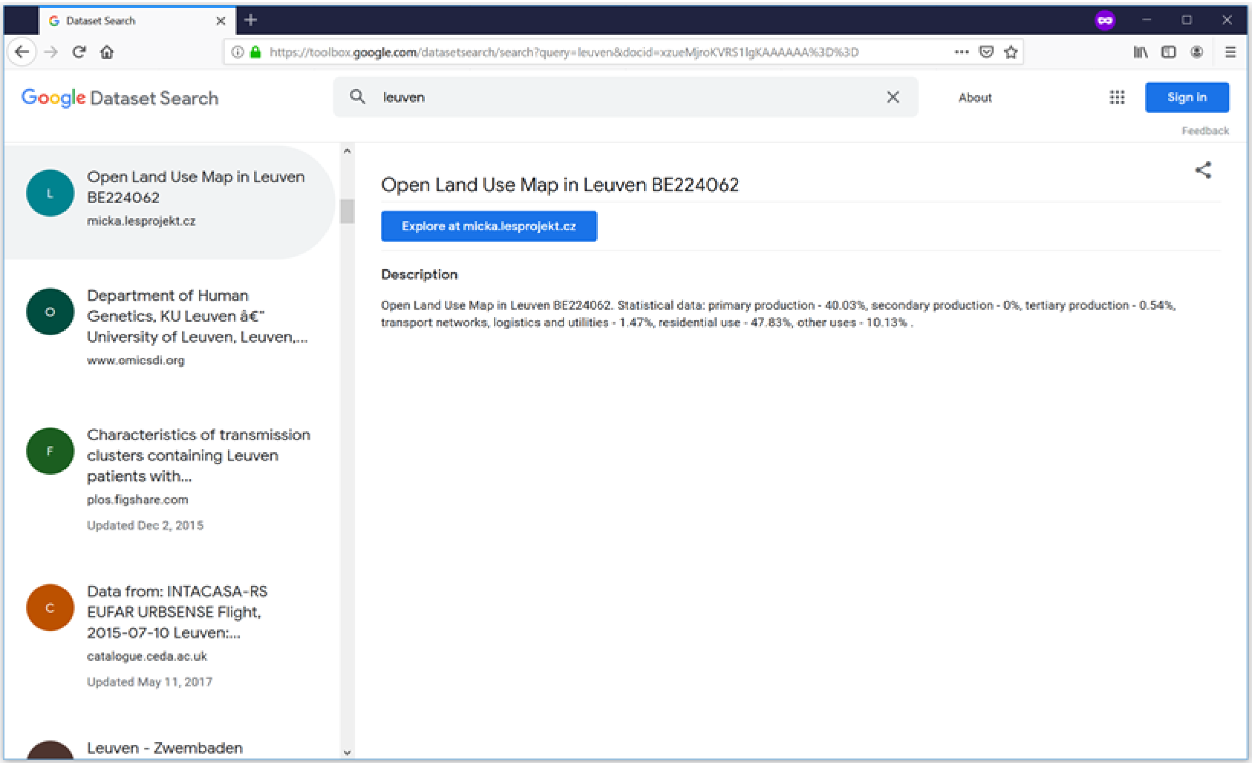

Figure 43: Annex - Google Dataset Search GUI visualizing also the results from the PoliVisu catalogue (Micka). See the first entry (Open Land Use Map in Leuven)