Modern Ways of Spatial Data Publication

Developer's Perspective

Rob Knapen (rob.knapen@wur.nl)

Wageningen University & Research

August 31, 2016

Geo4Web Testbed

Introduction

Geonovum has been working for years towards the realisation and success of the public sector spatial data infrastructure (SDI) in the Netherlands. Geographical data is now available using Open Geospatial Consortium (OGC) standards like [WCS], [WMS], and [WFS]. At the moment most geographical data is published as WMS what can be seen as a spatial picture, and not as data.

From the outside of the geospatial domain, the OGC standard is complex, because the data is published behind specialised web services and not readily available for the majority of web developers [Taylor and Parsons 2015].

Developers from outside the geospatial domain are increasingly making use of, and creating, geospatial data, and are therefore seen by Geonovum as an important new target group to disseminate geospatial data to the SDI. The question is how the current SDI can be leveraged and how the majority of web developers can be reached. Geonovum sees the web as an important dissemination channel and wants geospatial data to be accessible to web developers with no specific geospatial expertise. This testbed should therefore explore and explain possibilities of making spatial data a useful, integral and common part of the web. It should help provide some of the answers the Spatial Data on the Web standards working group is searching for.

The testbed has four research topics:

- Research topic #1: Modern ways of spatial data publication
- Research topic #2: A usable spatial data publication platform
- Research topic #3: Crawlable spatial data using the ecosystem of the Web and Linked Data
- Research topic #4: Spatial data on the web using the current SDI

In a previous tender research topics #2 and #3 were carried out which led to a series of lessons learned. In this research we focus on research topic #1: Modern ways of spatial data publication and in particular on task 2.

Assignment

The question for research topic #1 (Modern ways of spatial data publication) is: How do these lessons learned meet the constraints (e.g. budgets) and capabilities (e.g. in-house know-how) of governmental organisations on the one hand, and of data users on the other?

In task 2 of topic 1 this question will be answered from the perspective of the data user. Data users are in fact everyone who wants to use digital information on the World Wide Web for websites, apps and so on. This includes the geo-information professionals, but not what is typically called the "end users", the people who visit the web sites and use the apps that make use of the spatial data.

Task 2 specifically focusses on spatial application builders. Within the task a use case is to be implemented in a web application, using the data described in paragraph 3.3 and 3.4 of the tender. In addition other data that is published on the web may be used. The use case must also enact a part of the environmental law use case. In this was it simulates the new environmental act in a small setting.

The task is to:

1. Create a working application, using the data as described in paragraph 3.3 and 3.4 of the tender. This application has been described in our proposal.
2. Report on our findings: are the data findable, usable, and programmable?
3. Evaluate wether the information in the Lessons Learned help us to work with the data. For this the same questions should be answered as will be answered by the data publishers:
    - Evaluate the usability of the lessons learned. How useful is the lessons learned as documentation? Are the contents understandable, the examples workable and the form of documentation useful?
    - Report any errors found in the lessons learned.
    - Identify any gaps in the lessons learned, i.e. aspects of spatial data publishing on the web about which guidance is needed, but missing in the lessons learned.
    - Identify any bad advice in the lessons learned. Are any parts of the lessons learned arguably not good practice at all?
    - Give suggestions, preferably as text proposals, for the improvement of the lessons learned in areas mentioned in the items above.

The task has the following deliverables:

1. A working application published on the WWW.
2. A report describing what was done and containing the findings and answers to the questions above regarding the lessons learned.

Approach

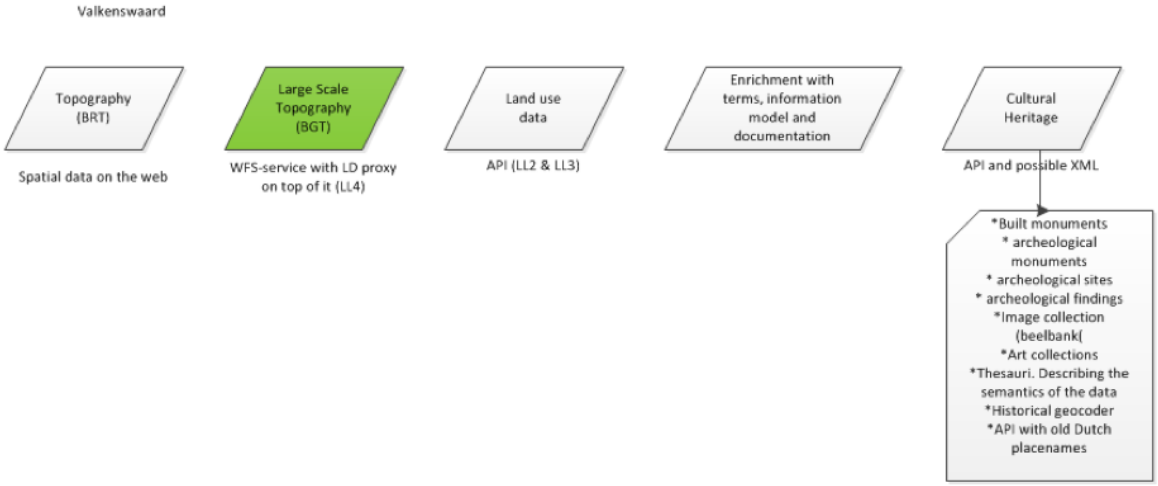
The first part of the task is to create a working application that enacts at least partly the new Environmental and Planning Act, using data on cultural heritage, topography and land use data. In the 'rapport informatiehuis cultureel erfgoed' use cases are described. For this project we selected the use case of Victor (p.12), a volunteer who wants to protect the 'veen-weide landschap' and historical farmhouses in his town from housing development.

In the use case Victor collects a lot of information about the area he lives in, so based on a certain location. He wants to use this information to be sure that the decision making is following the ideas of the community he represents (the 'Boerderijenstichting'). The application we want to create will support Victor in finding the right information.

In our working application the use case will be slightly different because the application will be built for Valkenswaard (since testbed data is available specific for this region).

Victor, however, is the end user of the application, and not the main focus. Therefore we will look at the task from the perspective of Bob, one of Victor's friends who happens to be a web developer. Furthermore we will assume that he only has a general idea of (spatial) maps on the web and no hands-on experience with developing them. Same for the equally complicated world of Linked Data on the web.

Within the testbed several data sources for the use case are delivered according to the formats shown in this figure:



We will have Bob try to find and use this data in a web application, both as a thought experiment and by really writing the source code. Not all these data services might be targeting a geo-layman like Bob but looking at them from his perspective should be helpful when later evaluating the Lessons Learned from the testbed as the second part of the task.

Results

Part 1 - The Application

Use Case

End User Context

Meet Victor

Victor van Beers is born and raised in Valkenswaard, a province town in Noord-Brabant. When he was 10 he gave a presentation at school about Castles with the castle "Loon" in Waalre as an example. The castle Since then he is interested in the cultural history of his town. Now he is an enthusiastic volunteer for a foundation that ins committed to preserve historical farms and the corresponding landscapes.

In the neighbouring town Waalre stands a castle, which land covered the present towns of Waalre and Valkenswaard until 1795. Because of the low location in the landscape it was not possible to develop as a castle-village. Near the castle there were tree historical farms that influenced the landscape. Also the Dommel valley is of importance of the historical landscape.

Victor likes to walk and o ride his bike in the surrounding areas and two times a year he organises excursions to teach about the historical landscape and monuments in town.

Motive

Victor read in the local newspaper that there is an initiative to build a new housing estate. This planned estate is threatening to fragment the historical landscape he loves. He understands the urgency of the housing estate in order to preserve other local amenities as the primary school and supermarket.

Victor wants to protest against the new housing estate in favour of the landscape. However when he does so, he needs to come with a reasonable alternative. He makes a phone call to the civil servant to get more information. He wants to hear how far the planning process is and what he can do to protest. The civil servant points him to the municipality website where he can see which steps are already taken.

Victor visits the website but finds out that it looks nice but is hardly helpful to his quest. He searches for the information he wants on 'bouwplannen' but as it turns out he has to make an appointment with the municipality office to have a look at them and there might be costs involved.

Some Friendly Help

As things happen Victor bumps into his friend Bob at a party and shares his story. Bob decides that he wants to build something in his spare time that helps Victor, and perhaps other people with similar goals.

Developer Context

Meet Bob

Bob, who is a web developer, has seen lots of websites that manage to show interesting information on maps, making things more insightful. And he has read many interesting articles about Open Data. Besides, Bob is also a little bit of a revolutionist, a fan of WikiLeaks, and thinks all government data should be freely accessible. He likes to be a little bit disruptive in that regard.

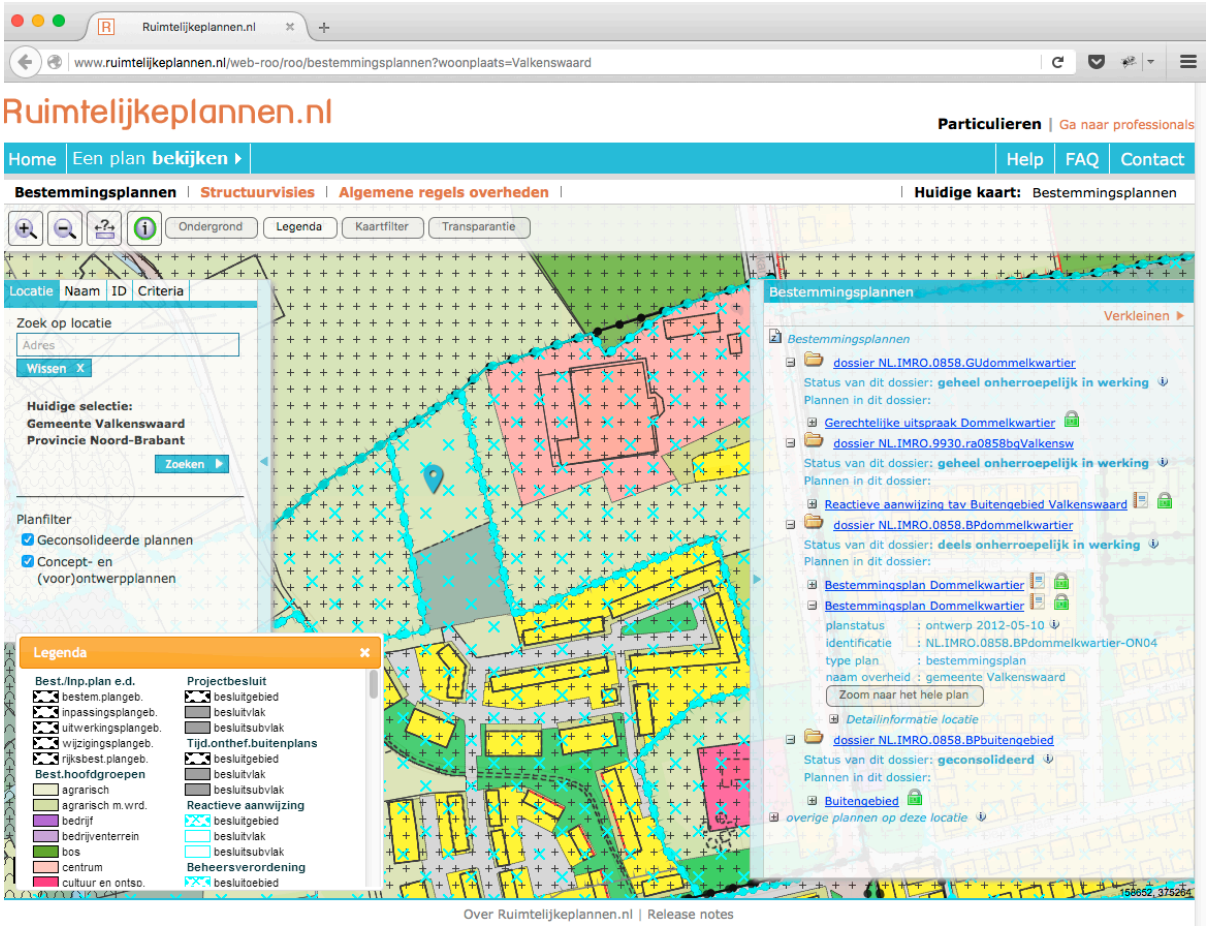
As a web developer Bob spends most of his work time developing applications for the World Wide Web. Usually he works in small teams or on his own so his is very interdisciplinary and able to cover all tiers (client, server, and database) of modern web applications. On the other hand he has none or only very little experience with working with spatial data, maps, and Linked Data. So he has some research to do.

Exploring

Looking for inspiration Bob fires up his favourite browser, Firefox, and searches for 'kaart Valkenswaard'. Top of the list he sees a Google Maps map of Valkenswaard, which is promising. Next he searches for 'bouwplannen Valkenswaard'. Now he has to scroll a little

further in the results where he finds the website www.ruimtelijkeplannenvalkenswaard.nl, which contains a link to the website: http://www.ruimtelijkeplannen.nl/web-roo/roo/bestemmingsplannen?woonplaats=Valkenswaard.

Bob is impressed by this website and how much information is available through. However he also finds it overwhelming and thinks it targets more expert users than he and Victor are.



By now Bob is fascinated by the idea of displaying government data as simple as possible on a map. His next step is to visit data.overheid.nl to see what is available there. He soon finds out that for spatial (geo) data they refer to another website, www.nationaalgeoregister.nl/. 'This sure looks promising' he thinks to himself. Plenty of interesting geo-data and they even have a tab on the webpage 'for developers'. Here he reads about WMS, WFS, Leaflet, OpenLayers, and so on, which all seems rather complicated. Going back and reading the 'About NGR' web page it becomes clear to Bob that they target Geo-ICT specialists, among others, that want to build a website or application that uses geo-information. He thinks he can

get it to work with some study of all these geo standards but still has hopes there is a much easier way with quicker results. Although he has been putting quite some spare time into it already.

Somewhere on the NGR website Bob stumbled across a link to a WIKI from Geonovum. Interested he looks them up on the Internet, at http://www.geonovum.nl/ and finds out that their goal is to make governmental geo-information more accessible. 'Exactly what I need', he says to himself. He also wishes they would just offer a 'for developers' link, or have a developer.geonovum.nl website like so many other organisations, to make things easier. He finds something about a geo4web testbed that he might want to give a try, but first he likes to get started building with what he has learned so far.

Building

Getting Started

Bob decides to use Netbeans as IDE for building the application, and to set it up as a Node.JS project. He also plans to use the Leaflet and D3 Javascript libraries for working with geo-information.

Versions he uses are:

- Netbeans: 8.1
- Node.js: 4.4.7
- Google Chrome: 52.0.x
    - Netbeans Connector extension

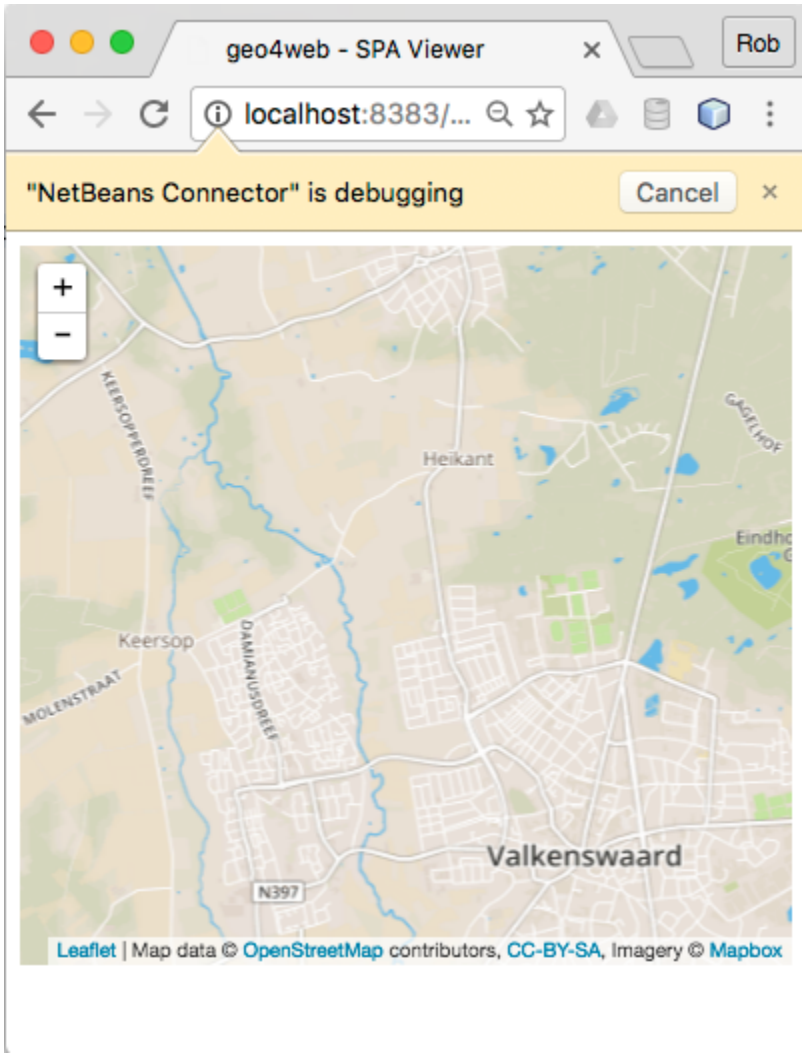
Quickly Bob has an empty project created. Now for some maps!

He adds:

- Leaflet.js: 0.7.7

And start following the Quick Getting Started webpage. For this tutorial he needs a Mapbox

API key, which is free, so he creates one (making a mental note to look into Mapbox later). It takes some tinkering but pretty soon Bob has the first result, a (Open Street) map of Valkenswaard!



Using PDOK

Bob decides to now have a look at this PDOK data he came across while exploring the options. He tries developer.pdok.nl but no such website exists. Searching for 'developer' on the www.pdok.nl website also finds nothing. He starts browsing the website and notices something about data services in the product section. It says that data services can be used by copying and pasting an URL, which sounds easy enough! However unfortunately none of

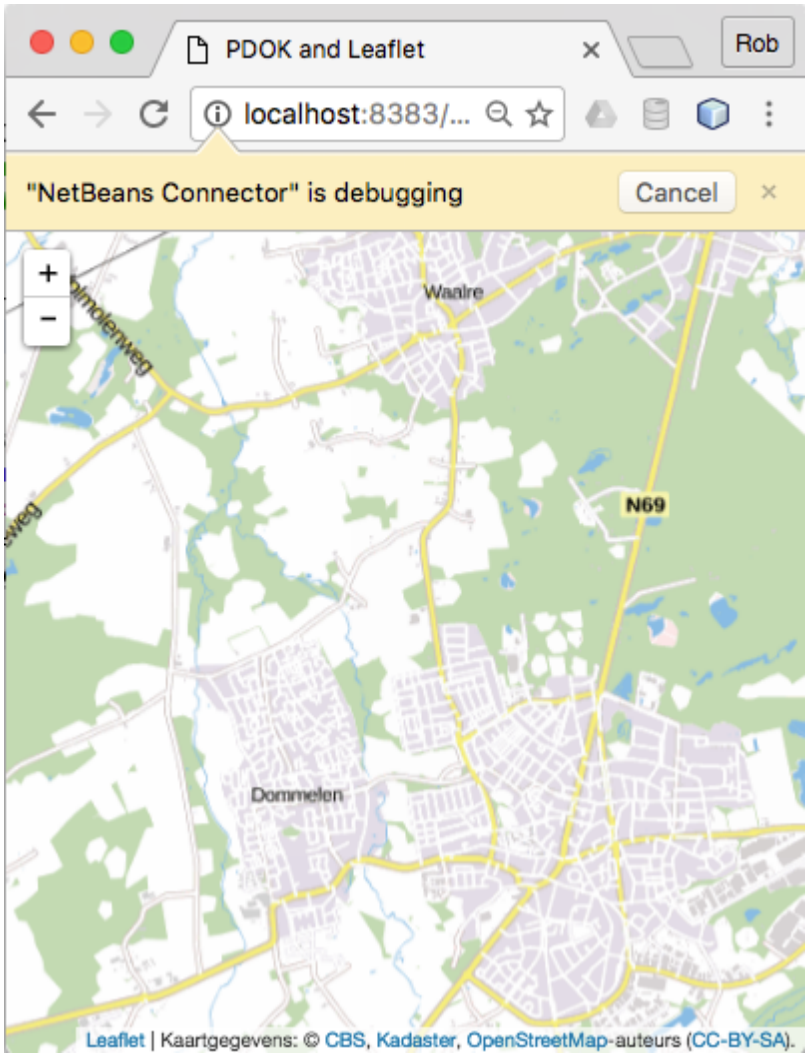
these URLs match what Leaflet expects. They are all WMS, WFS, WMTS, WCS, etc. So looks like he needs a way to use those with Leaflet. Fortunately on the Leaflet website he finds a whole explanation about how to use WMS in Leaflet.

Seems that to figure out which 'layers' are available in a WMS data service he needs to use another application called QGIS, so he sets out to install it ...

With QGIS 2.14.3 installed Bob is slowly starting to feel lost in all the GIS terminology. He manages to set up a connection to the TOP10NL WMS, which looks interesting, but QGIS shows that there are things in it called 'functioneelgebied_vlak_label', 'inrichtingselement_lijn', 'terrein_vlak', and so on. While Bob is looking for a simple map. He tries opening 'terrein_vlak', but QGIS runs into network problems and crashes, leaving Bob feeling disappointed.

The next day Bob decides to have another look at the PDOK website thinking that maybe there are other options. But the only thing that comes close that what he wants is the 'PDOK Kaart'. The possible ways to use it do not really appeal to Bob (as a web developer he likes his freedom of choice). However there is also a link to github (https://github.com/search?q=pdok) where he finds some source code from another developer that uses PDOK with Leaflet. Exactly what he was looking for! Some more coding to do to update his app ...

It works! With not too much effort the PDOK map is displayed. From the source code by the other developer Bob now knows that it is something called a 'brtachtergrondkaart' and there is something done with map coordinates to get from lat, long to RD? He writes it done for further study (the list is getting longer), for now he is just happy with the result!



Adding Features

While panning and zooming the PDOK map is fun, Bob thinks about the useful features he can now add to help Victor. He can program the map to initially show the area around Valkenswaard, but a search box for a place would be nice and make the app more flexible. And then when the map is zoomed to a location of interest to the user it should display additional information, like the 'bestemmingsplannen' and e.g. things about cultural heritage so Victor can build his case.

Bob decides to skip the place name search for now and focus on adding the useful data, so he has something to impress Victor with. And perhaps he can use it to find a sponsor or other

developers to help with adding features to the app! He looks at his notes and remembers the Geonovum geo4web initiative, so he heads back to github to look it up.

It takes some searching but he finds a few interesting links:

- http://www.ldproxy.net
- https://swaggerhub.com/api/apiwise/landcover/1.0
- http://146.185.182.204/map?lng=51.34&lat=5.47

Using LDProxy

On the website www.ldproxy.net Bob sees that they currently offer three data services:

- INSPIRE Adresses WFS
- Basisregistratie Grootschalige Topografie (BGT), Valkenswaard
- Agrarisch Areaal Nederland WFS

After his experience with trying to work with WMS and Leaflet he is not really ready to try another of those W*S services. On the Leaflet website there is something about support for WFS-T (called an OGC WFS-T client layer for Leaflet, at: http://flexberry.github.io/Leaflet-WFST/). Perhaps he can try that later although he is not so sure the data provided will be useful for Victor. But maybe the BGT is.

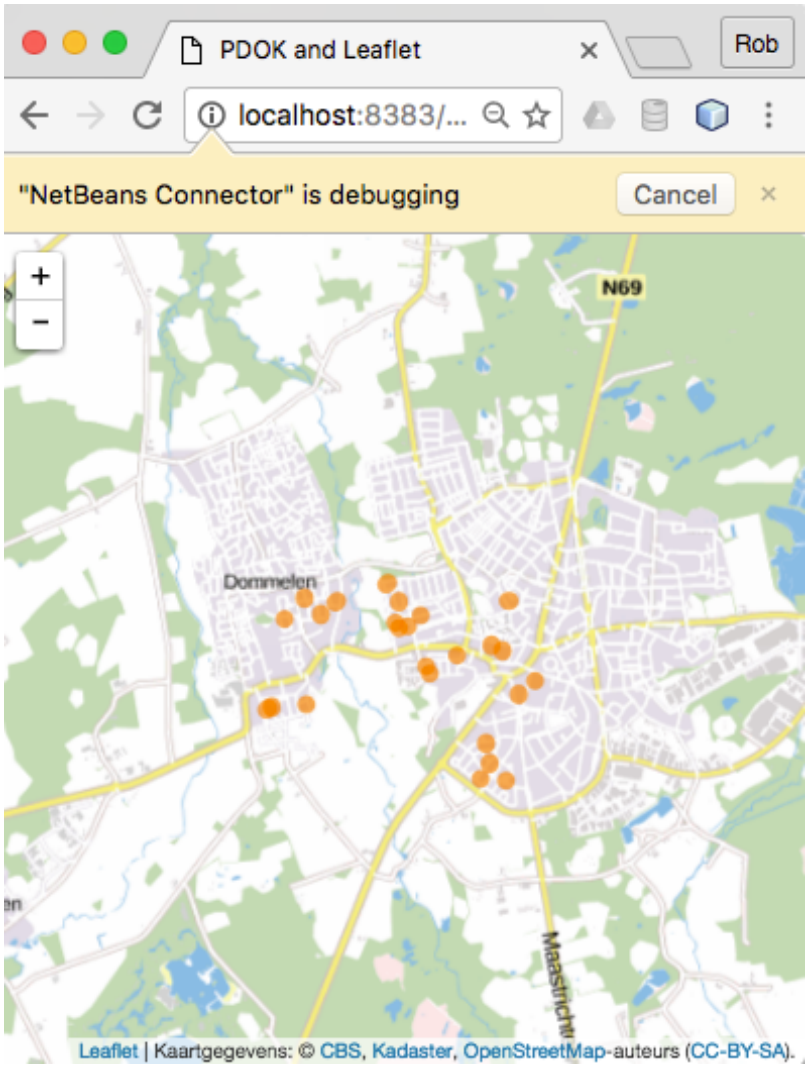
Following the link for the LDProxy BGT service Bob gets a webpage with some information about the service and the data. It is not very clear what it all means and how he can use it in his Leaflet app. 'Wegdeel' and 'Waterdeel' probably are road and river segments. What 'Kunstwerkdeel' means is more mysterious to him. After some exploring Bob thinks that it looks like a kind of REST API, where he can use http://www.ldproxy.net/bgt/Kunstwerkdeel/ to get a list of items, and for example http://www.ldproxy.net/bgt/Kunstwerkdeel/

_A2F312CA9F00E4B61E0532B0B5B0ADC22 to get an individual item. And by adding a parameter to the URL the data will be returned in JSON format: http://www.ldproxy.net/bgt/Kunstwerkdeel/_A2F312CA9F00E4B61E0532B0B5B0ADC22/?f=json.


```
{
 "type" : "Feature",
 "id" : "_A2F312CA9F00E4B61E0532B0B5B0ADC22",
 "geometry" : {
  "type" : "LineString",
    "coordinates" : [ [ 5.430309110117319, 51.34613899275969 ], [ 5.4303095099695184, 51.346136844306116 ], [ 5.430300107904979, 51.346136020835495 ], [ 5.430299708094898, 51.34613821423266 ], [ 5.430309110117319, 51.34613899275969 ] ]
 },
 "properties" : {
  "objectBeginTijd" : "2014-12-08",
  "identificatie.namespace" : "NL.IMGeo",
  "identificatie.lokaalID" : "G0858.33501c2f2dd9442081ca56be0541a40d",
  "tijdstipRegistratie" : "2014-12-08T10:59:27",
  "LV-publicatiedatum" : "2015-04-21T10:24:44",
  "bronhouder" : "G0858",
  "relatieveHoogteligging" : 0,
  "bgt-status" : "bestaand",
  "plus-status" : "geenWaarde",
  "bgt-type" : "niet-bgt",
  "plus-type" : "duiker"
 }
}
```


He notes that this 'Kunstwerkdeel' is a 'duiker', and considers this must be jargon. Perhaps 'Pand' (http://www.ldproxy.net/bgt/Pand/) has more useful information for Victor. And since Leaflet supports the Geo-JSON format Bob thinks he should give it a try.

Bob retrieves the data with the URL: http://www.ldproxy.net/bgt/Pand/?f=json, and adds it to the application (as a file called bgt_pand.json). On GITHUB he finds a library for Leaflet so that the file can be loaded (https://github.com/calvinmetcalf/leaflet-ajax) and the features added to the map.



Bob notices that there are only a small number of markers, and then figures out that the LDProxy service uses a paging mechanism (also when asking the full list of items). Unfortunately he can not (spatially) query the data or find out the total number of items or pages available. So it would take some additional coding to make it work. Bob decides to talk to Victor first to decide if spending more time on this would make sense.

Using Land Use Data

The next data service Bob takes a look at is https://swaggerhub.com/api/apiwise/landcover/1.0. Although it has "land cover" in the URL the data provided by it has to do with spatial planning ("bestemmingsplannen"). There is a very straight forward REST API with a parameter to request data in pages and another for search queries.

There are some practical issues though that make it less easy to use the API in the app with Leaflet. The paging is nice, but there seems to be no way to retrieve the total number of pages or features available from the API. And there is no way to do a spatial search, for example to get only the features within a certain area, or near to a location. Bob thinks he either has to retrieve all the data and build the processing logic in the app, or cache some information and then retrieve the features needed based on the IDs depending on the visible area of the map. This is more memory efficient but would require more calls to the data service (hence more network data usage).

Another obstacle is that the data is returned as JSON, but not as GeoJSON. Bob downloaded a small piece of the data and tried to add it to Leaflet but nothing is displayed. The data seems similar enough to the GeoJSON that Bob retrieved from the LDProxy so a conversion might be possible. Perhaps somebody already has developed a converter and put it on GITHUB, for instance this one looks like a good start: https://github.com/respec/leaflet.wfs-t (Bob starts to like GITHUB more and more).

Another thing to talk to Victor about first before investing more time in it.

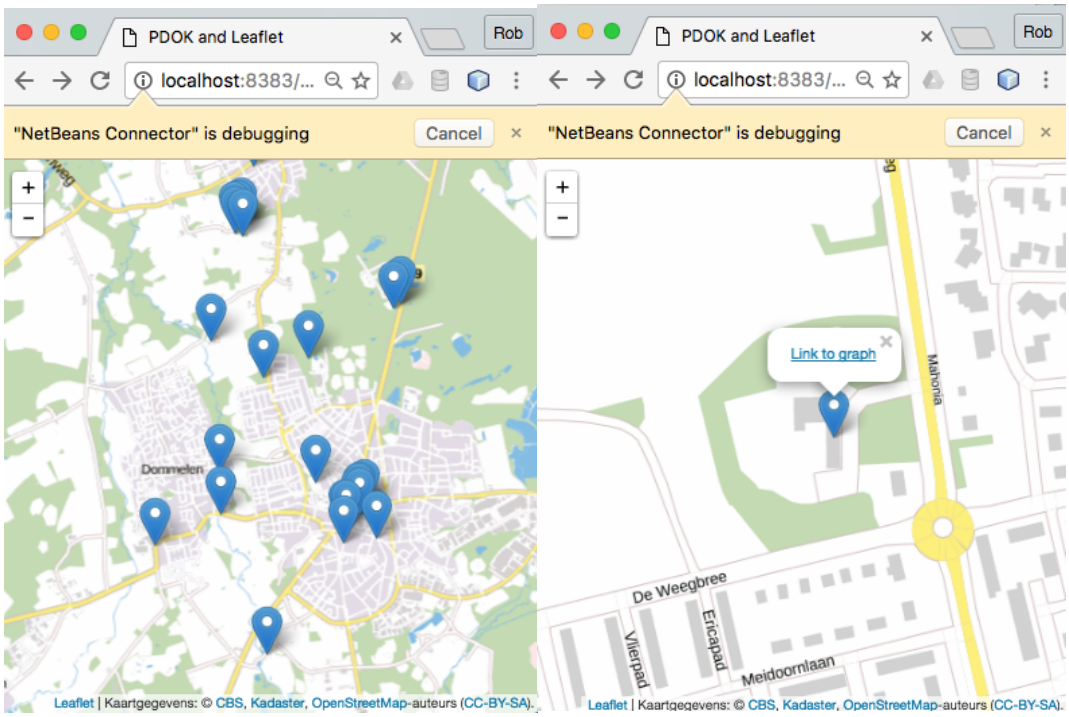
Using Cultural Heritage Data

From his list Bob has one final data service he wants to give a try before going to talk to Victor about how best to proceed. He hopes to get some more useful data on his little map from the URL: http://geonovum.triply.cc. It seems to be an experimental service but looks promising.

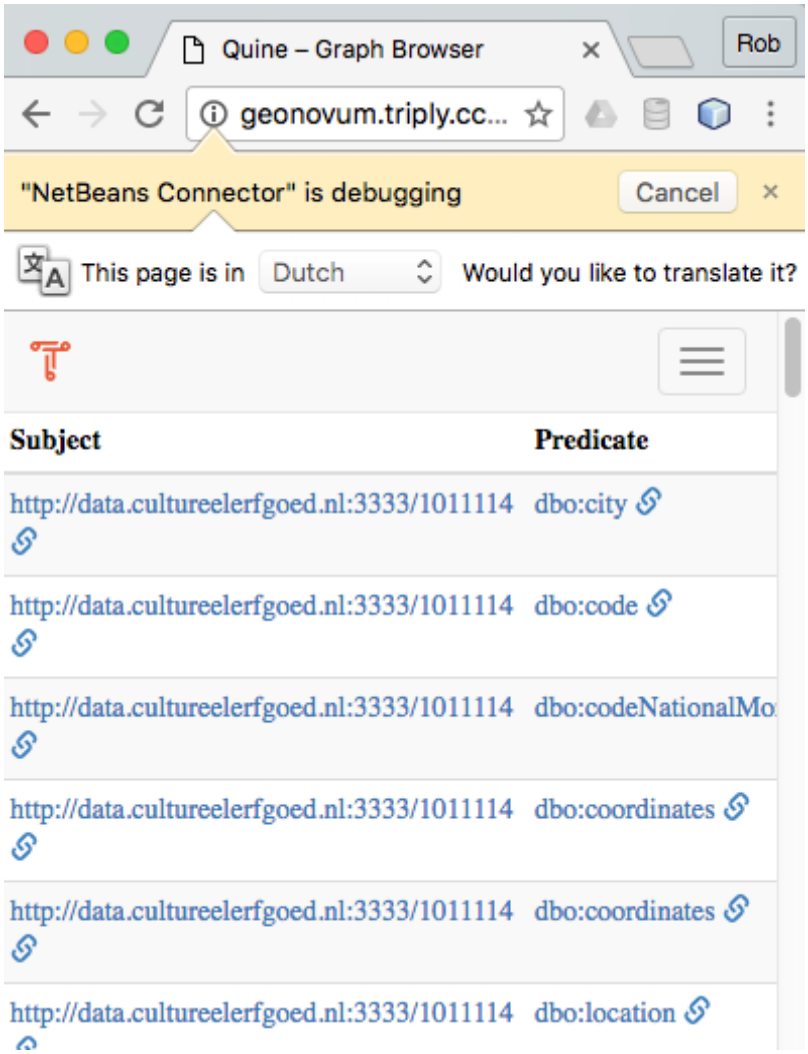
He heads over to the website and is welcomed by a list of available datasets, of which he recognises BGT. But there also appears to be data about monuments that might be useful. The website mentions a triple store and SPARQL, there is a webpage with short

documentation on URLs and parameters that can be used to search and retrieve data. And a lot of new acronyms (such as RDF, N-Quads, and N-Triples) that sound vaguely familiar to Bob. He puts them on a list to study later. For now he is most interested on the Geo webpage of the site that displays a background map and can be overlayed with markers, for example for monuments.

There are no examples (yet) in the documentation so Bob looks at the source code for the geo webpage and from it extracts the JavaScript code he needs and the IDs of the graphs (datasets) that are available. To be able to use it he needs the JQuery software library (http://jquery.com, v.3.1.0) so he adds it to his project. After some tinkering and solving of CORS (HTTP access control for cross-origin resource sharing) problems he pretty soon has markers for monuments on his map.



A lot more information seems to be available for each marker, stored as a graph, which can be retrieved separately. Clicking on the info pop-up for a marker already shows some of it.



Bob is sure he can tap into this to filter out the information that would be useful for Victor, and present it in a nicer way or use it as a bases for some fancy functionality. To do so he needs to study the data further and probably learn something more about this Linked Data that it seems to be based upon.

Part 2 - The Lessons Learned

Lessons 1

Lesson 1: Everyone in a platform or community has their own needs and capacities.

A spatial data publication platform is an online platform which enables different users to

publish, view or edit spatial data. To make the platform successful and widely supported, the platform must meet user's needs and be adapted to user's expertise and tasks.

Features that increase the ease of use and intended use:

- **Lesson 1A: Make sure the needle can be found in the haystack**

Datasets can be very large, so good search functions are required to let users find the needle in the haystack.

- **Lesson 1B: Keep it simple**

Users generally want things that are easy to learn and use. So keep your spatial data publication platform as simple as possible...

- **Lesson 1C: Think carefully about who is allowed to do what**

In a platform or community, each has its own role, knowledge and responsibility. This requires that access rights to data and features to view or edit data are assigned to different users.

- **Lesson 1D: Each speaks it own language and lives in his own world**

"What's in a name? That which we call a rose.
By any other name would smell as sweet."

Romeo and Juliet (II, ii, 1-2)

Evaluation

From the perspective of a web developer with no practical experience in usage geo-information all these lessons are very valuable and would certainly help. As remarked before,

the spatial data publication platform needs to have an entrance tailored to the targeted audience. (Web) developers are not Geo-Information specialists, and also no Data Scientists or Linked Data experts. They will come to the platform with a different purpose in mind and looking for something that helps them solve a problem and quickly and easily as possible. Time is money, most likely. Possibly there are other alternatives, e.g. if it is too difficult just use Google Maps instead. "Free" data, or a free tier of a data service for development also helps.

Lessons 2

Lesson 2: Make search engines feel comfortable to discover you.

If you want your dataset to be crawled by search engines then make it comfortable for them to discover your dataset.

The following search engine-friendly tips may improve the discoverability and crawlability of your published dataset:

- **Lesson 2A: Show a search engine the direction with an XML sitemap**

Finding something is much easier if you know where to go. Use XML-sitemaps to direct search engines to pages or data objects which are available for crawling.

- **Lesson 2B: Foster to link everything with everything**

Be aware that you don't just publish data on the web, by doing so you create data on the web, others will start to link to your data, thus creating added value for your data.

- **Lesson 2C: Think of the future, use persistent URIs**

Publish data in registries at unique persistent URIs to ensure links between data sets are available, attainable and sustainable in the future.

- **Lesson 2D: Make use of the structure, include Schema.org markup**

Structure your data in a way search engines understand. Use Schema.org markup.

Evaluation

Not surprisingly the Internet is a (maybe the) major source of information for software developers. Books, if not for the most basic theories and concepts, are outdated in no time. Technology, toolkits, methodologies, and programming languages are changing very rapidly and the best place to find the most recent information is the Internet. So to be successful any publication platform and data or data service must be on the Web and be indexed by search engines. Proper metadata has to be provided to make that possible. Furthermore it needs to be expressed in the terminology familiar to the target audience since indexing and searching is mostly text based (this is mentioned in lesson 1D).

Creating interlinkages between data opens up new possibilities for data use. For this it is important to notice that the more traditional Semantic Web technologies like RDF and SPARQL are complicated, and following and reasoning across RDF links is not easy. But just like with SOAP and XML most developers are happy with simple standards like REST and JSON, and e.g. Schema.org.

Lessons 3

Lesson 3: Deal with the unknown set of developers and devices.

The unknown is out there: you have to deal with the large set of unknown developers and devices.

To make your data API better workable for a large group of developers and devices, you

should follow these approaches:

- **Lesson 3A: Serve your data in many different flavours.**

So many developers and devices, so many flavours. Serve your data in as many flavours as possible to reach the widest possible audience.

- **Lesson 3B: Improve performance, reduce payload.**

High Definition (HD) data is not alway necessary. Performance counts for developers. So: downgrade your data, reduce the payload and improve performance.

Evaluation

These are perhaps the most important of the lessons learned with regards to software developers and in fact cover most of what is described in the Use Case and Bob's experiences. Current data services in the testbed do not apply these lessons learned and are difficult to use by a developer. The new data service by Triply allows the data to be accessed in multiple ways and returned in multiple formats, which is a nice step forward.

The use of persona to define and test the Developer Experience is a powerful method that can be used when defining data APIs. Multiple personas can be used to represent different groups of developers, with different skills, and using different devices.

Lessons 4

Lesson 4: Don't copy data, use proxy.

Last but not least: DO NOT COPY DATA!

Instead use a Linked Data Proxy approach as intermediate layer between traditional data services and webapps.

Evaluation

The use of a proxy instead of creating a copy of the data is good advice. The API and implementation of the proxy should however be performant and memory efficient so that a developer will not be forced to download datasets from the proxy and thus still create copies of the data for use in applications. If needed the developer should be able to cache data and have a way to know when data is changed and contents of a cache must be updated. So the proxy should reflect more than only the data content.