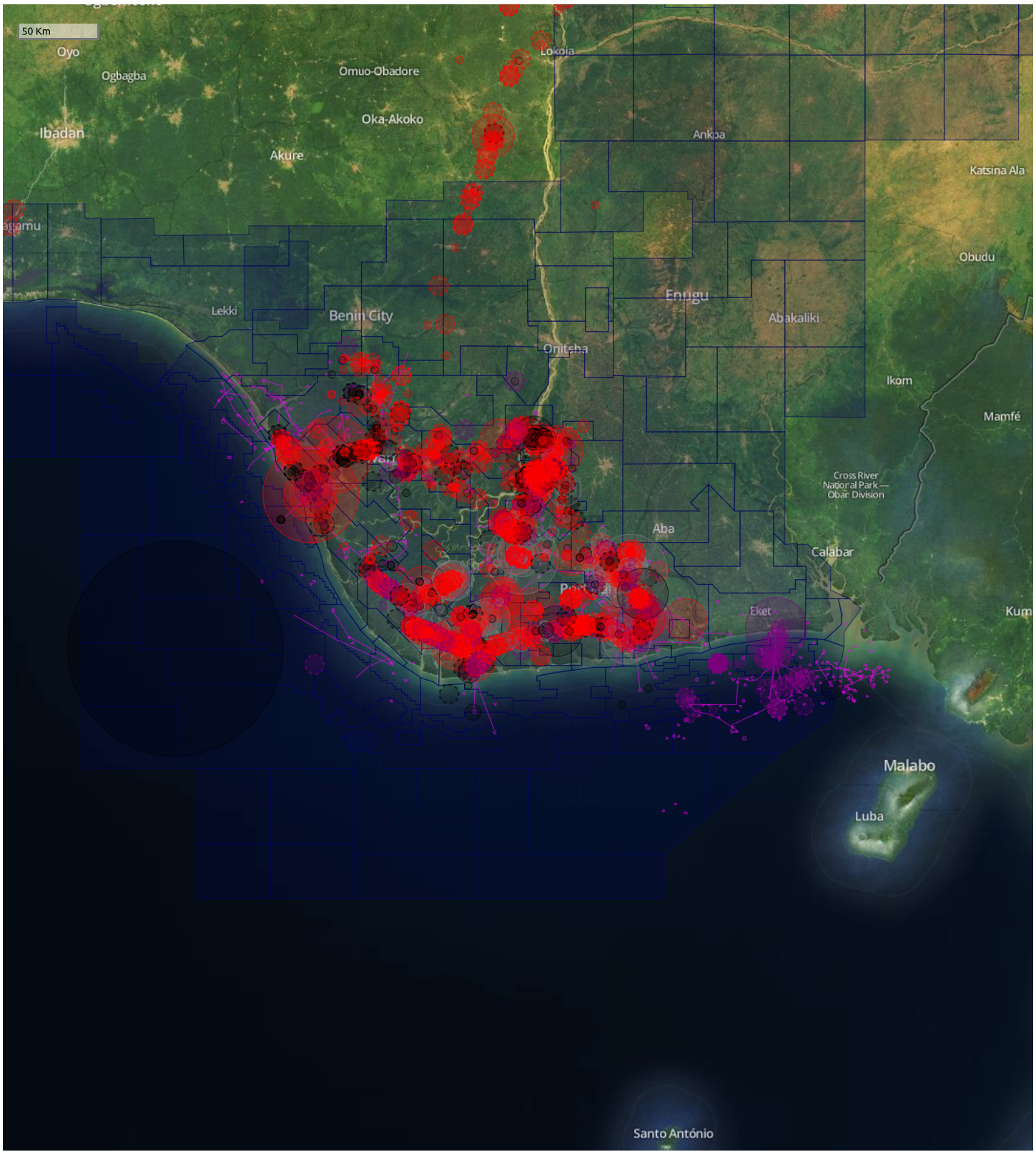
The Nigerian Oil Spill Monitor



Alberto González Palomo

2019-12-31

Sentido Labs.com

The Nigerian Oil Spill Monitor

Alberto González Palomo

alberto@sentido-labs.com

Date: 2019-12-31

Contents

Introduction

The Nigerian Oil Spill Monitor is the software system used by the Nigerian National Oil Spill Detection and Response Agency (NOSDRA) for recording, mapping and monitoring oil spills. Citizens and other stakeholders can verify the accuracy of those reports at https://oilspillmonitor.ng

It was developed as part of the oil spill mapping project managed by the British NGO Stakeholder Democracy Network (SDN), funded mainly by the United Kingdom's Department for International Development (DFID) through the Facility for Oil Sector Transparency in Nigeria (FOSTER[1]) program.

From the 5th annual report of the FOSTER program:

> Major achievements under this output through-out the life of the programme include the innovative platforms created by Stakeholder Democracy Network (SDN) and hosted within Nigeria's environmental regulator (NOSDRA), that track and monitor oil spills and gas flaring with the aim of improving the responsiveness of the Nigerian government to these issues.

The gas flaring platform mentioned there is the Nigerian Gas Flare Tracker (GFT).
In April 2019 SDN handed operation of both OSM and GFT over to NOSDRA.

1.1 Oil spill transparency

I wrote the following in an e-mail to SDN and NOSDRA on January 2014:

> At this point, Nigeria is at the forefront of oil spill transparency in the world. As far as I know, the only other country that provides such transparency is the USA:
> http://primis.phmsa.dot.gov/comm/reports/safety/SIDA.html?nocache=9374
>
> They even have pipeline maps online so that, for instance, workers can check where pipelines lie before digging.
>
> The Canadian government (actually the province of Alberta which is the main oil producer) claims to be transparent but when some journalists actually took them on their word they resisted: claimed that they could not deliver the reports in electronic form and instead printed them out on paper, which the journalists had to transcribe.
>
> You can read the full story here:
> http://globalnews.ca/news/572572/reporters-notebook-how-we-covered-this-story/
>
> They finally compiled the information on oil spills in the province from 1975 to 2013, and made it freely available from their newspaper:
> http://globalnews.ca/news/622513/open-data-alberta-oil-spills-1975-2013/
>
> As for Europe, only reports of oil spills in the Baltic Sea are publicly available: (HELCOM's map requires Flash Player)
> http://helcom.fi/baltic-sea-trends/data-maps/
>
> I've asked other entities about the data they have on other European regions, and the only answers I got were that the data exists but it's not available to citizens which I find ridiculous.

1.2 How it happened

In March 2013 SDN asked me to take over the software development part of the project they had started in 2012. They had been delayed for half a year trying to get results from their previous technology partner and decided to look for someone else.

[1] http://www.opml.co.uk/projects/facility-oil-sector-transparency-and-reform-nigeria-foster

The application was supposed to be “very nearly there” and “need only minor final code amendments and css interface changes”, but after working on it for almost two months it became clear to me that it was just a rough proof-of-concept (PoC) demonstration.

My work during that time is recorded in the GitHub project repository Sentido-Labs/monitoring-platform and included speeding up the data processing time from 6 minutes to roughly 6 seconds.

The 15th of May, at 8:52 according to my notes, I started working on a new program from scratch without billing SDN as they hadn’t requested that. I had already patched what I could in the PoC, but its lack of structural integrity made new features and improvements harder and harder.

I had been evaluating different components: database management systems with geography support (PostGIS, SpatiaLite, MongoDB), front-end components (LeafletJS, OpenLayers, D3, AngularJS, jQueryUI, and others) and by 22:40 I settled on LeafletJS + h5bp + Bootstrap (with jQueryUI) + AngularJS with the back-end still undecided: it would just load a prepared JSON file for now.

Three days later I simplified it further by removing jQuery and things were progressing well. There was a pause in the project while SDN was in Abuja meeting NOSDRA and collecting feedback and requirements; I took on a small job of a few hours for another customer in the meantime and worked on the new application in my spare time. By the 22nd it had most of what the old prototype did and some new features. At this point it became clear that this approach would work and I proposed SDN to abandon the old PoC for the new prototype, to which they agreed provided that we didn’t lose any features.

In the first week of June the main parts of the application were working including a simple data storage back-end, much more efficient (14 times faster) than the previous one.

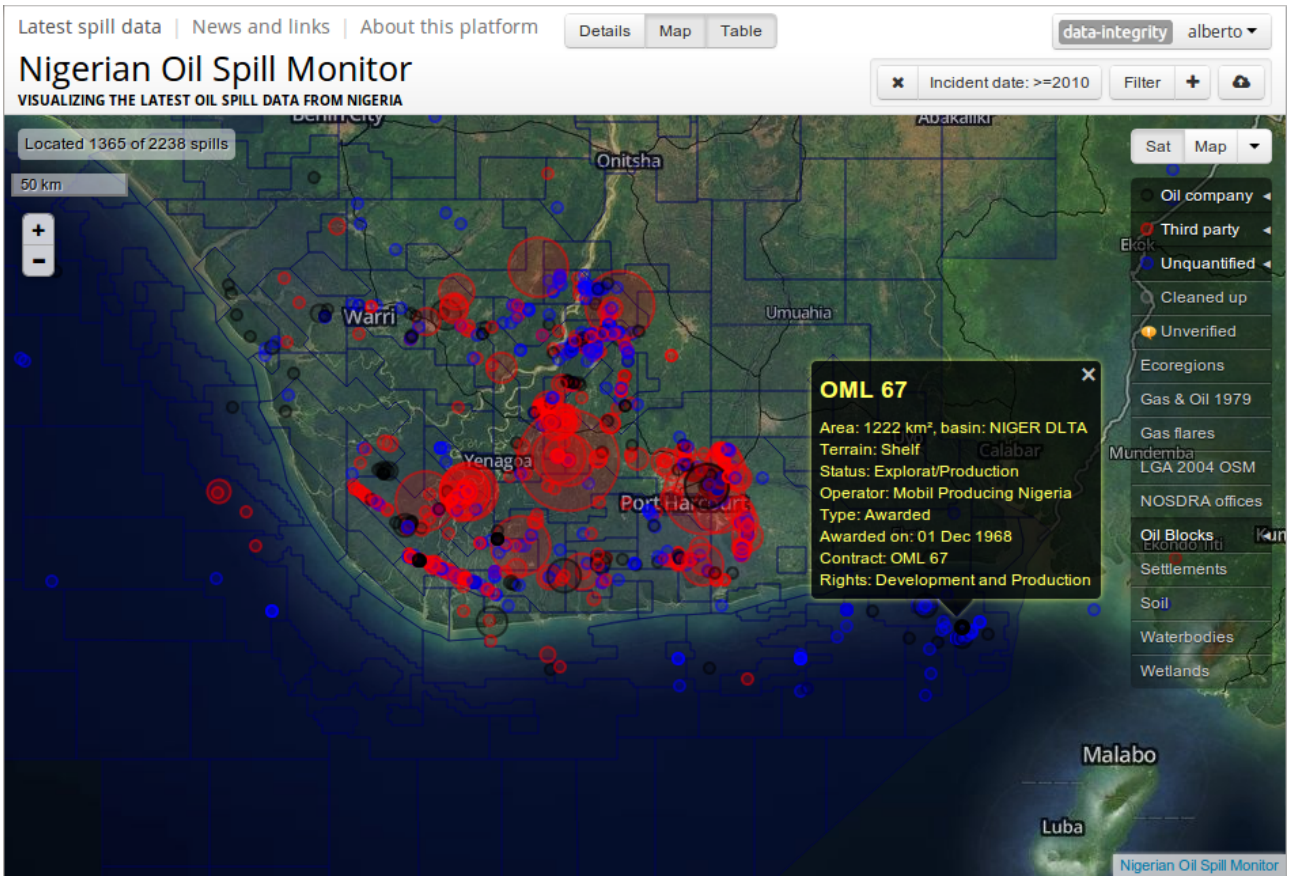


Figure 1.1: The new prototype as it was in July 2013.

Since then I’ve been working sporadically on this whenever SDN could get funding. It has been a great experience so far.

It must be noted that the initial data set that was used both for the PoC before I joined, and for the real application afterwards, came from the Microsoft Access database application developed by NOSDRA employees *Ahmed* Ismail and *Mohammed* Gumsuri.

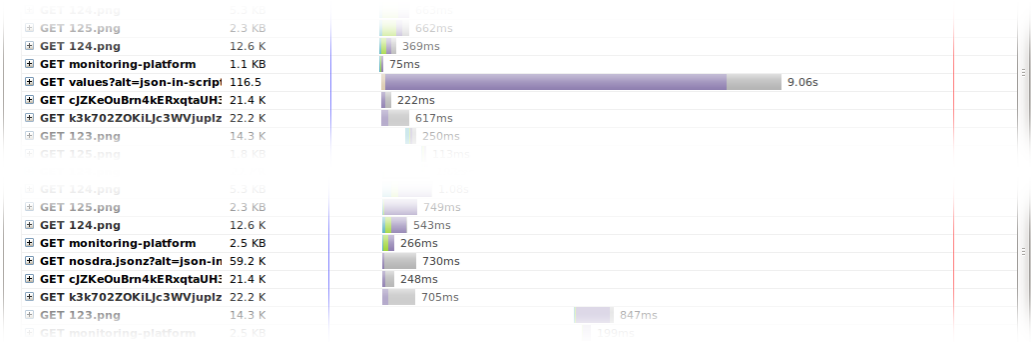


Figure 1.2: Loading the spill data went from taking 9 seconds (top graph) to 0.7 seconds in the new prototype (bottom graph).

The application's user interface and feature set are the result of requirement analysis and suggestions from SDN (especially Rory Hodgson) and NOSDRA, and my own ideas. The software engineering and implementation work was done entirely by me, Alberto González Palomo.

1.3 Technology basis

The application is built on these main foundations:

- LeafletJS, which displays the map layers and handles the interaction with them such as panning, zooming, etc. This is the most important third-party component because it would be very difficult to replace or build from scratch. Vladimir Agafonkin did an excellent job making it and I wholeheartedly recommend it.

  I modified it to allow reaching spill markers (the circles) covered by others, make it compatible with Harvard's AfricaMap, and other little things.

- AngularJS (plus UI Bootstrap), the framework on which the whole application is built. I did not notice until a few days after I started using it that Angular's object name prefix is "ng", the country code for Nigeria. That was a nice pun, but although it was the right tool at the time (2013) to get the application built quickly, I do not expect to ever use it again. When building such a performance-critical application as this, you reach a point where the framework gets in the way more than it helps, and Google's habit of abandoning products and their abysmal documentation make them unsuitable for long-term projects.

- Bootstrap for the visual style (v2.3). I kept that version even as new ones were released because they followed the "flat design" trend which makes it more difficult to notice which things are clickable.

- FontAwesome for the icons and Ubuntu as main font. I tested different fonts and found that to my eyes, the Ubuntu font provided the best readability for the data displays.

- The basic HTML page skeleton was from HTML5 Boilerplate, a common-sense starting point to avoid browser compatibility issues.

- PHP for the HTTP API server and data storage. I chose it over Java (which would deliver better performance) because it is the simplest to install and keep running by SDN without my assitance.

Fortunately SDN agreed to discard support for old browsers right from the start. It was clear to all of us that the effort of having to build different versions of each feature for the non-standards-compliant browsers was more than we could afford. In the meantime since that decission was made in 2013 those browsers have practically disappeared.

Exploring the spill data

The main page contains a filter toolbar (§ 2.1 *Filter toolbar*) and four tiled views of the same data, which is the result of applying the selected filters to the full dataset:

- The **map view** (§ 2.3 *Map view*) shows the location of all incidents for which there are coordinates available, combined with a variety of data layers.
- The **detail view** (§ 2.2 *Detail view*) is the area on the left of the map view, which shows the details of any incident selected by clicking on its marker on the map, or on its row in the table view.
- The **table view** (§ 2.4 *Table view*), which is initially hidden, shows all the incidents as a table. It can be toggled using the "Table" button at the top right corner.
- The **workflow view** (§ 2.5 *Workflow view*), also initially hidden, contains pre-made filters to streamline NOSDRA's workflow. It can be toggled using the "Workflow" button at the top right corner.

All four views are synchronized: selecting or modifying an incident causes the other views to update themselves immediately. The separate **summary page** (§ 2.6 *Summary page*) shows aggregate graphs.

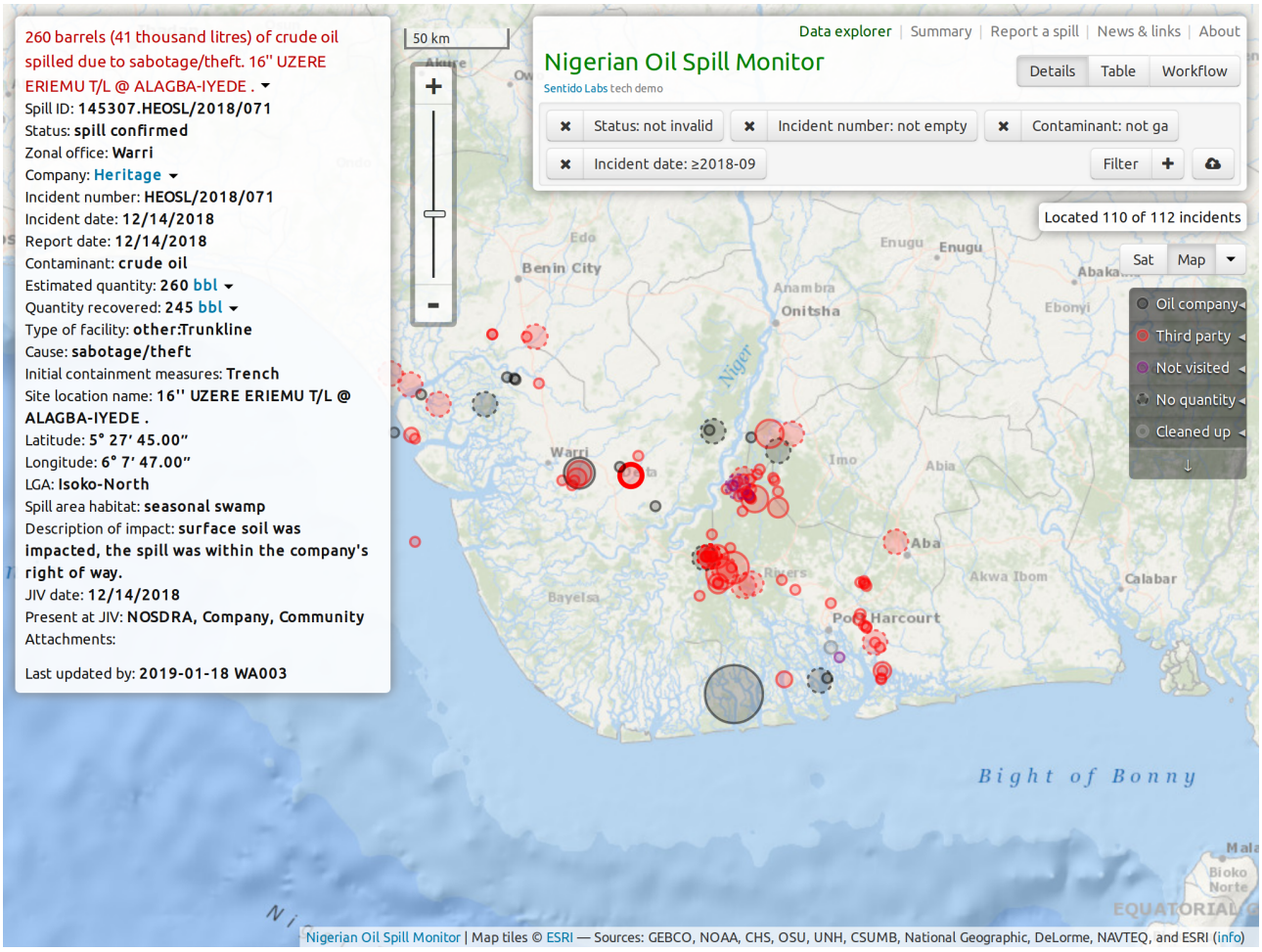


Figure 2.1: The initial view after selecting an incident, the **thick red circle** in the center of the map. The area of each circle is proportional to the spill size: one pixel per barrel, except for small spills that would be difficult to see and have a radius of 4 pixels.

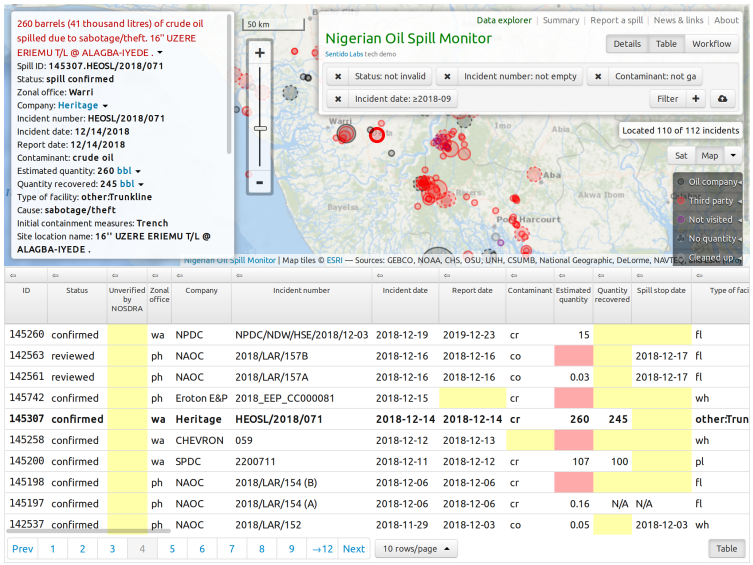


Figure 2.2: Table view (bottom half) in addition to map and detail views.

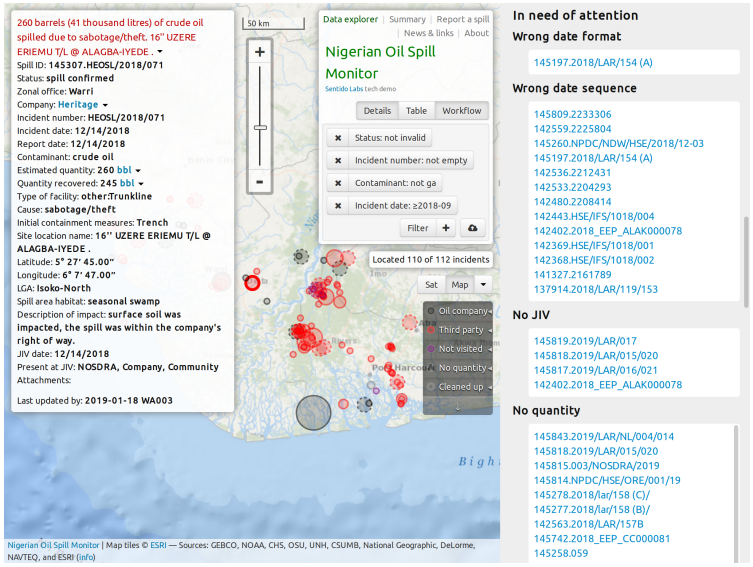


Figure 2.3: Workflow view at the right side.

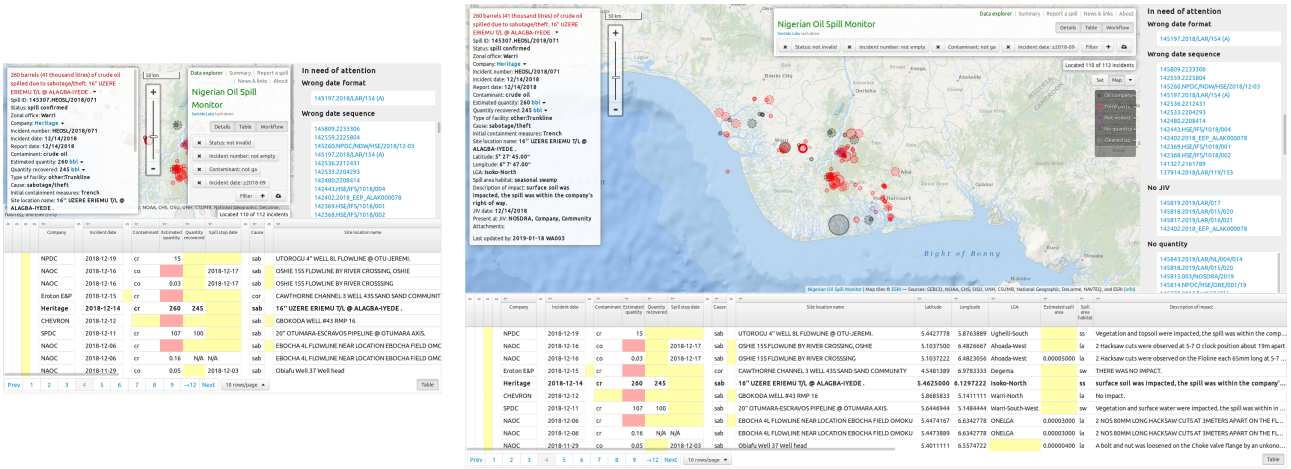


Figure 2.4: A 1024×768-pixel screen (left) gets crowded with all views active at the same time, but on a 1900×1080 HD screen (right) it becomes more comfortable.

2.1 Filter toolbar

The initial filters when loading the page select for display only those reports where the **status** is **not `invalid`**, each report **has an incident number** and **JIV date**, the **contaminant** is **other than gas**, and then a date filter ensures that only the latest incidents are shown as they are likely to be the most relevant ones.

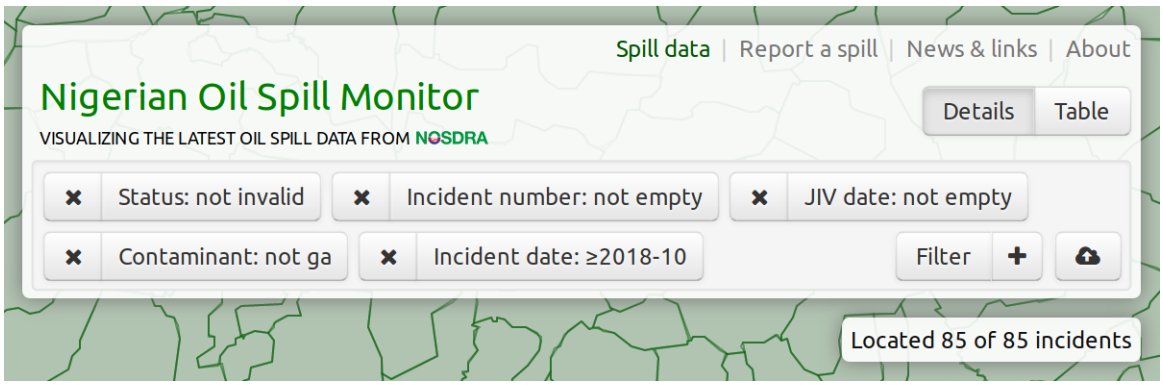


Figure 2.5: Filters can be removed by clicking the "×" button in each of them. **All dates** are in ISO 8601 **format**: *YYYY-MM-DD* or four digits for the year, two for each month and day, left-padded with zeros (*2019-4-2* → *2019-04-02*), separated by hyphens. See § 3.3 *Dates*.

The **date filter** is automatically adjusted to only show the incidents since two years ago. At the right side you will see the message "**Located 85 of 85 incidents**". That means that the filters selected 85 incidents, and all of them had valid coordinates. In Figure 2.6 it reads "**Located 998 of 1018 incidents**" which means that 20 incident reports lack proper geographical coordinates.

To apply the filters after adding or removing them you must click the **Filter button** at the right side, which will light up in yellow whenever there are changes that have not been applied yet.

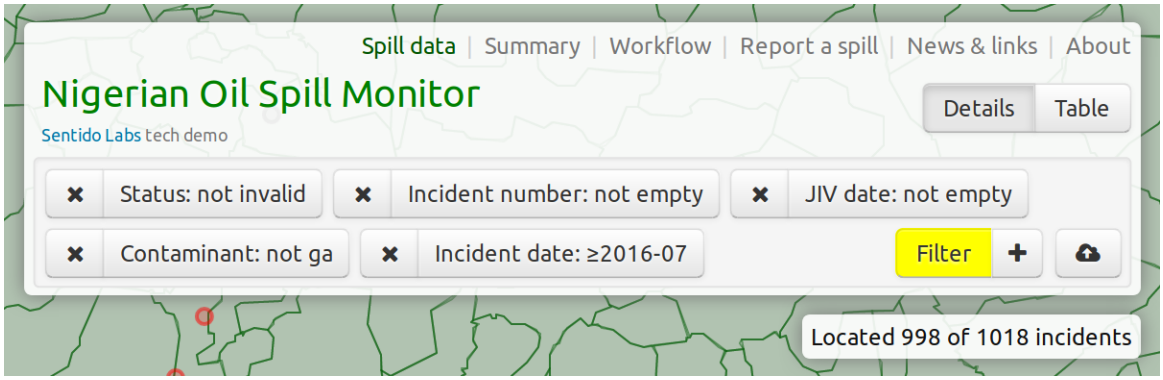
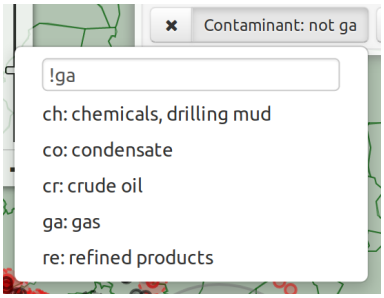
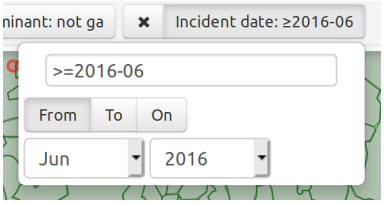


Figure 2.6: The **Filter button** lights up to remind you to apply the filters by cliking on it when you are ready. You can also apply the filters by pressing the ↩ **or** Enter **keys** while modifying a filter.

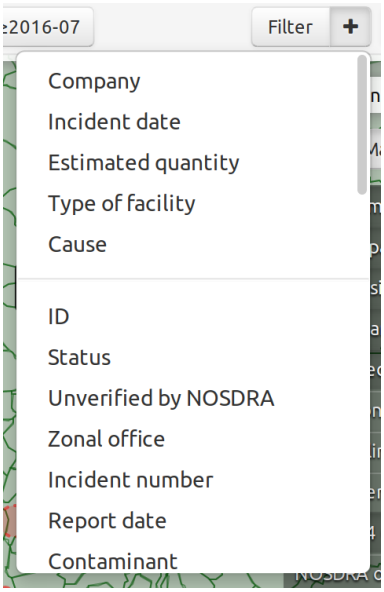
The reason for not applying filters immediately is that it if you are adding several filters and each was applied immediately, the computer might become unresponsive while trying to apply the last one as you start adding the next. Using the Filter button allows you to set up filters without interference, then apply them in one go.



Many fields use abbreviated codes, in this case for different contaminants. You need to use those codes for filtering and when entering reports. The **!** operator means "**not**".



Date filters select reports **from** a date on with the **>=** operator, up **to** a date with **<=**, or only those **on** the given date.



When adding a filter with the "Filter **+**" button you need to pick the database field. At the top of the menu are the most usual fields to filter by, and below the grey line are all the others.

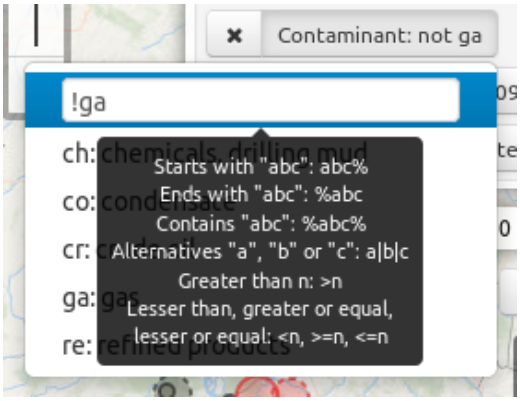


Figure 2.8: SQL-like filter expression language, shown by a tooltip when editing a filter.

The expression language for filters is similar to SQL[1]: "**Port%**" selects items whose value in that field starts with "Port", and similarly "**%court**" those ending in "court", and "**% Har%**" for containing anywhere a space followed by "Har".

To get all reports that match any of a number of patterns, just put those patters in one filter separated by vertical bars, like "**Port%|% Har%|%court**" which would find all reports matching any of those patterns.

For numbers and dates, you can also use the arithmetic comparison operators:

- "**<**" to find values less than the given value, for instance "**<2014-08**" on the "Incident date" to select spills that happened before, but not including, August 2014. To include August, use "**<=2014-08**" instead.
- "**>**" for values greater than the given value, for instance "**>180**" on the "Estimated quantity" field to find incidents with more than 180 barrels of contaminant spilled.
- "**<=**" same for "lesser-or-equal" $\leq$, but that symbol is not found in most keyboards so we use **<=** instead.
- "**>=**" same for "greater-or-equal" $\geq$.

To find reports where a field is **empty**, use a single "equals" sign as the pattern: "**=**"

If you want to **invert a filter**, to select the reports that do not match the pattern, you can put the symbol **!** or ¬ in front of it. For instance, "**!=**" (or "¬**=**") will select reports where that field is not empty.

[1]This application's database management system does not use SQL (see § 6.3 *Data storage*), but I figured that if users had to learn a pattern language, it should be as close to a standard as possible.

Filter examples

Find reports with missing attachment files

If you enter only a caption for an attachment, but forget to enter the URL or upload the file, it will appear as a link but it will not work.

To find those reports, add a new filter for the “Attachments” field, then enter the filtering expression “**`%"url":""%`**”.

The Attachments field value is a small JSON-encoded *key-value database* and that pattern finds those where there is some entry with the key `url` and an empty string `""` as value.

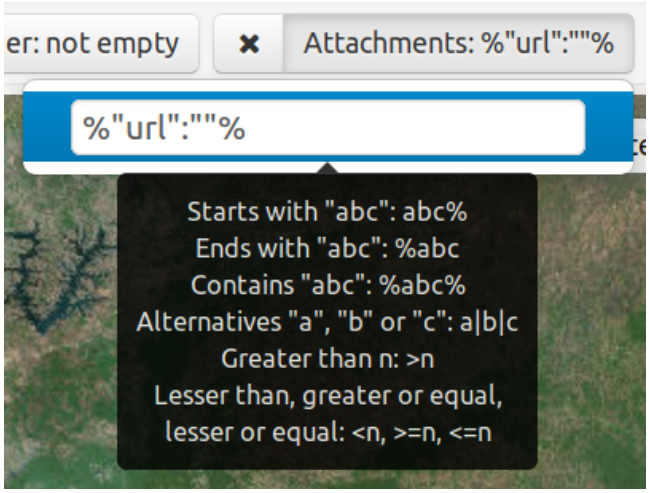


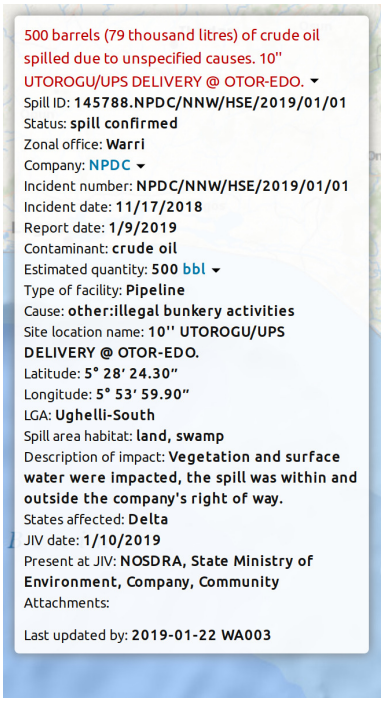
Figure 2.9: finding missing URLs in attachments.

2.2 Detail view

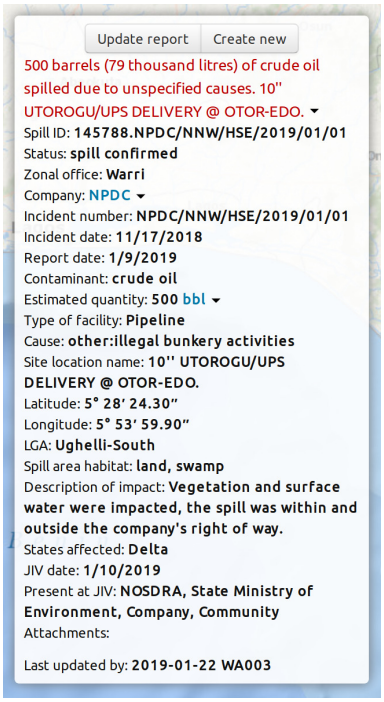
Here all the incident's records are displayed and, if you sign-in as a user with a role that allows it, you can **create and modify incident reports**.

The first part, in red letters, is a summary of the event generated by the application from the data entries below it using a simple template-based natural language generator.

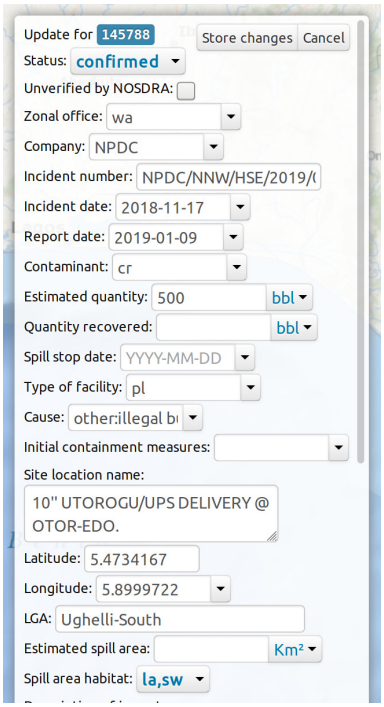
Below that are all the fields which have values in this report. Empty fields are omitted.



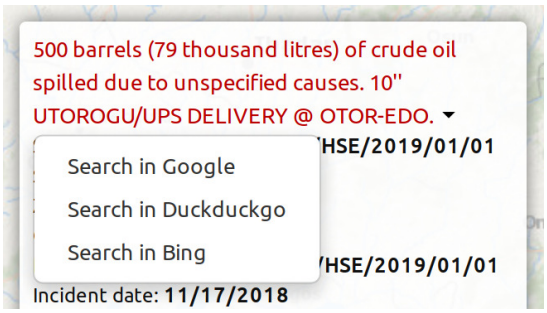
Before signing in.



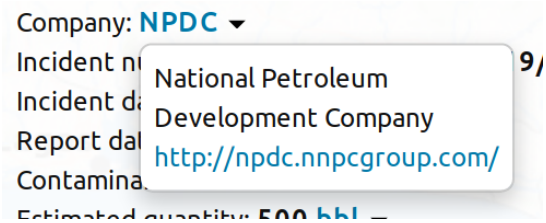
After sign-in, there are two buttons in the detail view: Update report and Create new.



Updating a report, changes will be stored only if you click on Store changes. Otherwise, Cancel will discard them. See § 3.1 *Storing or cancelling changes*



For some of the biggest incidents, it is possible to find additional information on the web.



The company name drop-down (little triagle to the right of the name) shows the company's web address if known.

The field Last updated by shows the date of all the changes done to the report. If the report was modified several times in the same day, it is displayed as one change although the server stores all of them.

2.3 Map view

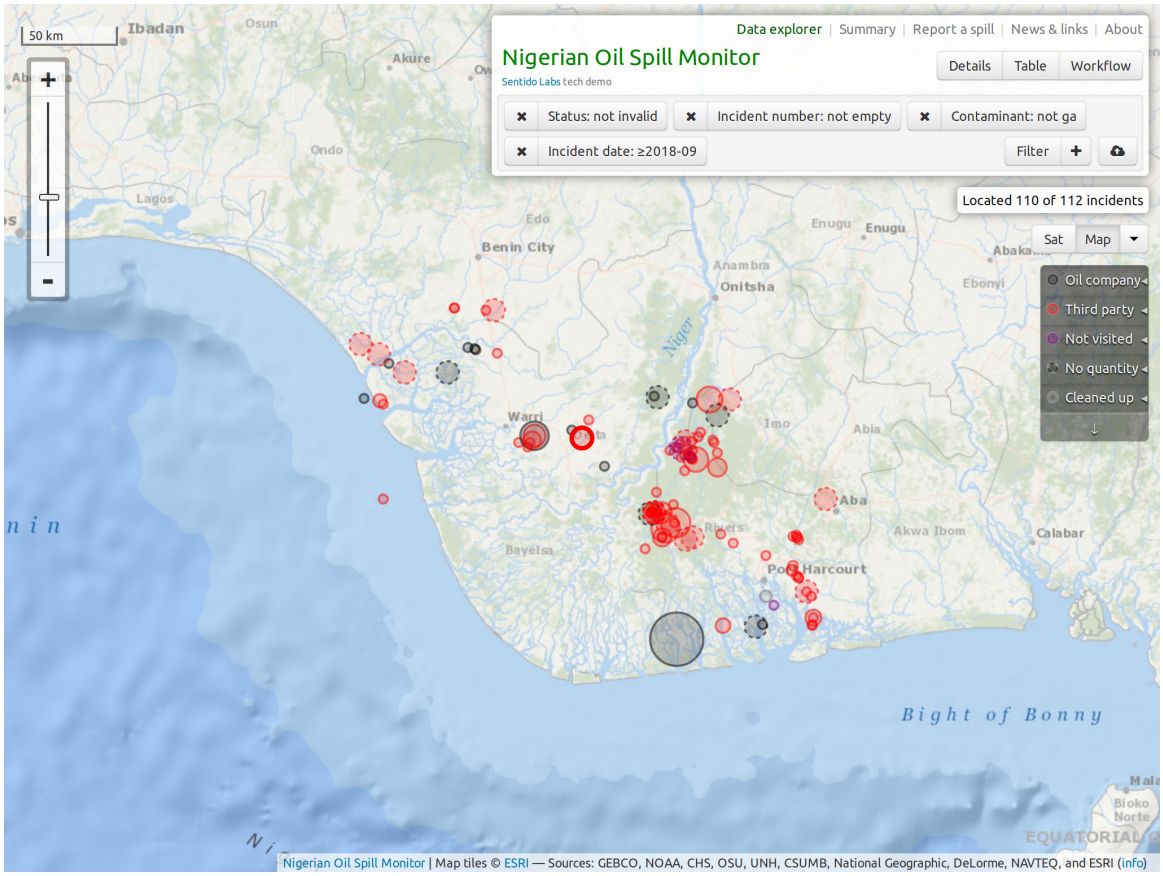
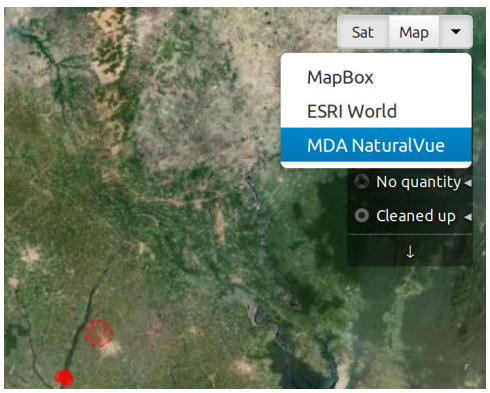


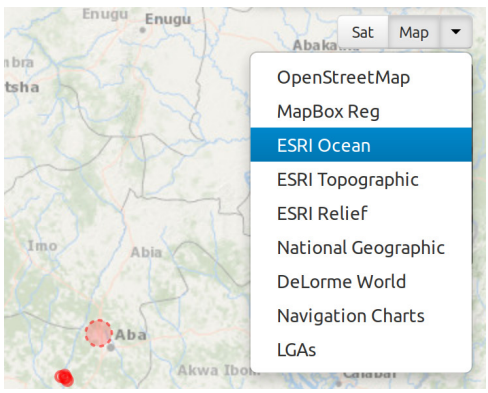
Figure 2.12: The map view contains a scala (top left), a zoom slider, and at the right side the layer selectors. When spill markers (circles) overlap, you can reach them by clicking repeatedly: the first click brings a marker on top, a second click will push it to the bottom and reveal the next one.

Base maps

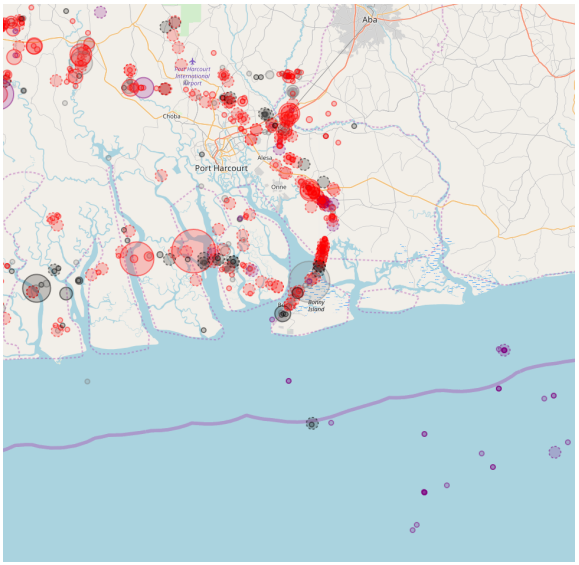
The bottom layer is called the “base layer” and there are two kinds:

- **Sat** layers are photographs taken from planes or satellites, further processed to remove clouds (a common issue in the Niger delta most of the year) and to highlight certain features.
- **Map** layers are schematic representations that make it easier to see other layers on top, such as the additional data layers described in § 2.3 *Map overlays*.

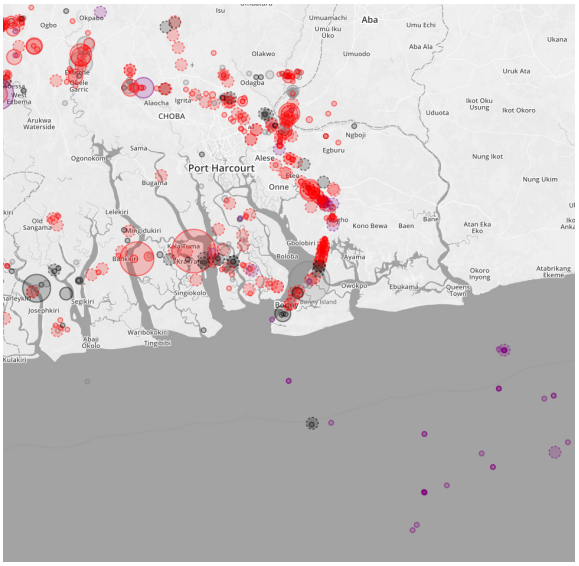




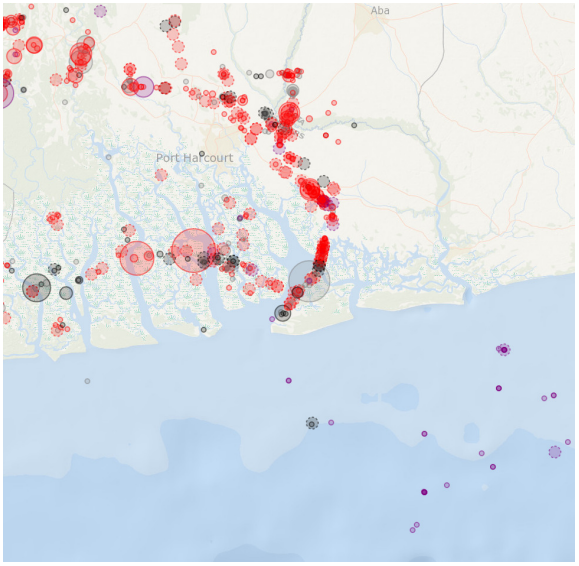
The different base map layers have different styles and serve different purposes. Some offer more information by themselves, while others are more bare to serve as base for complex data overlays.



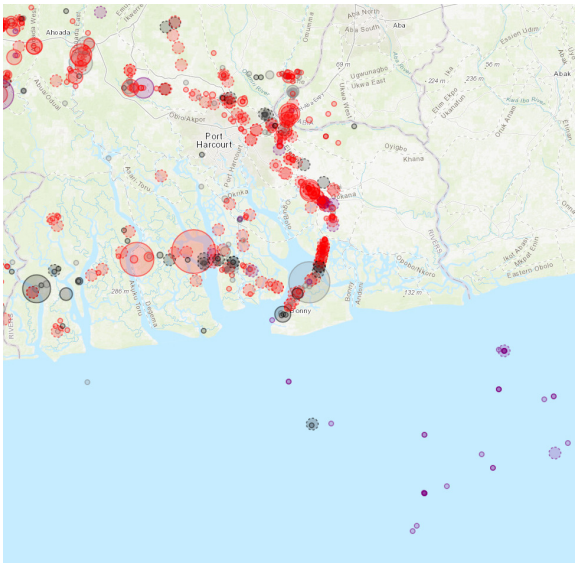
OpenStreetMap



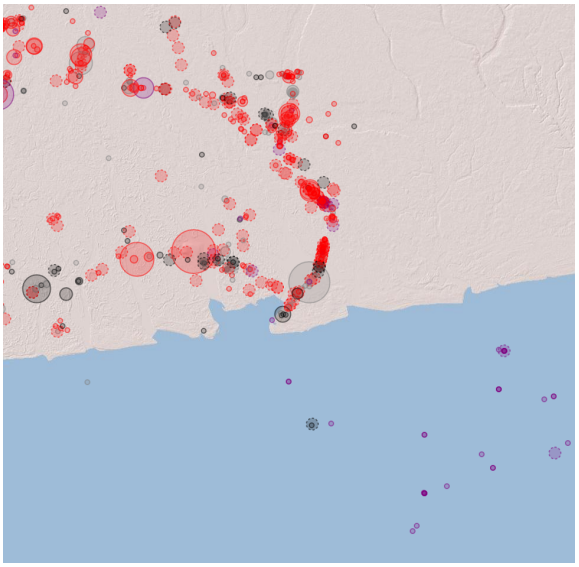
MapBox Reg



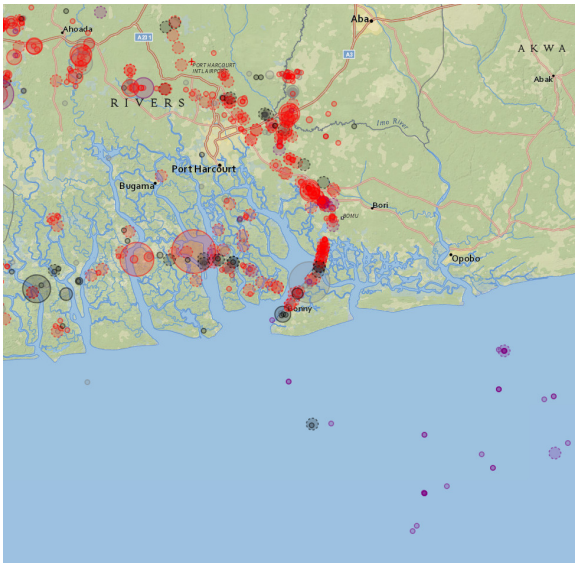
ESRI Ocean



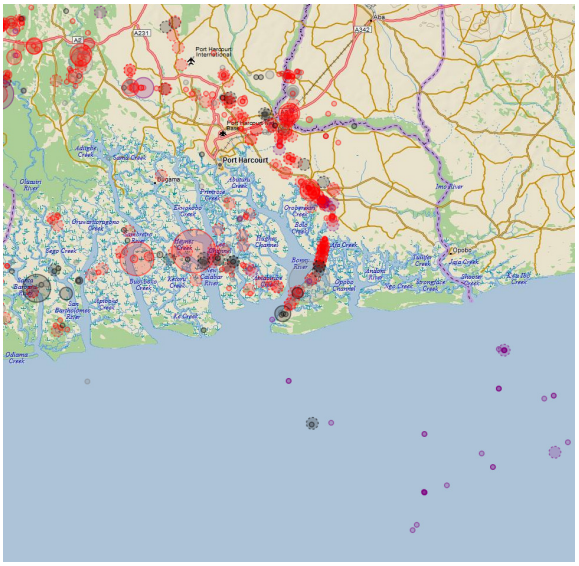
ESRI Topographic



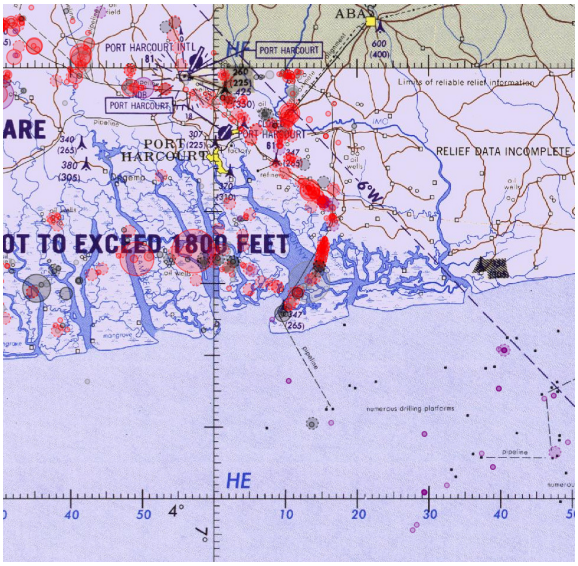
ESRI Relief



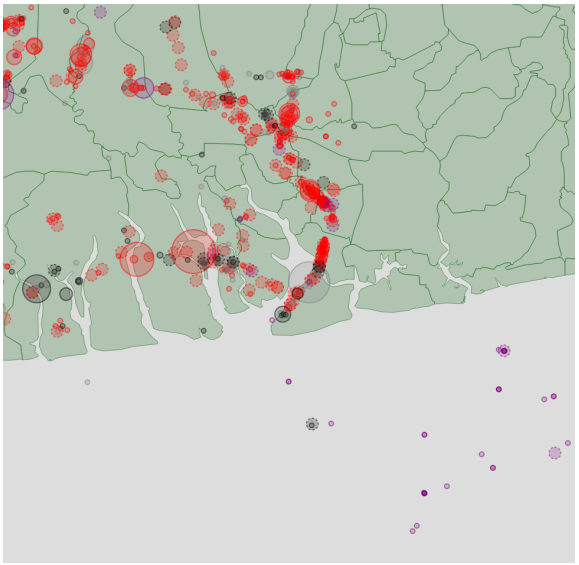
National Geographic



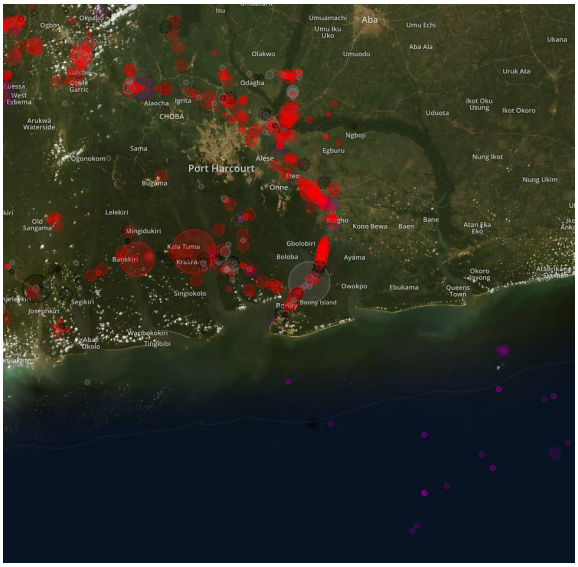
DeLorme World



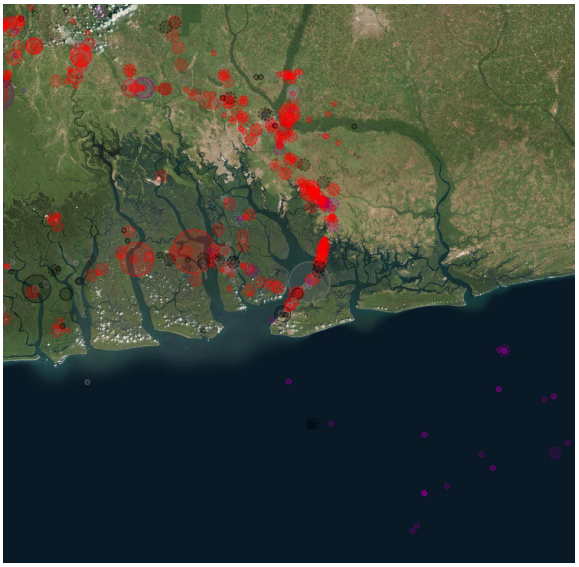
Navigation Charts



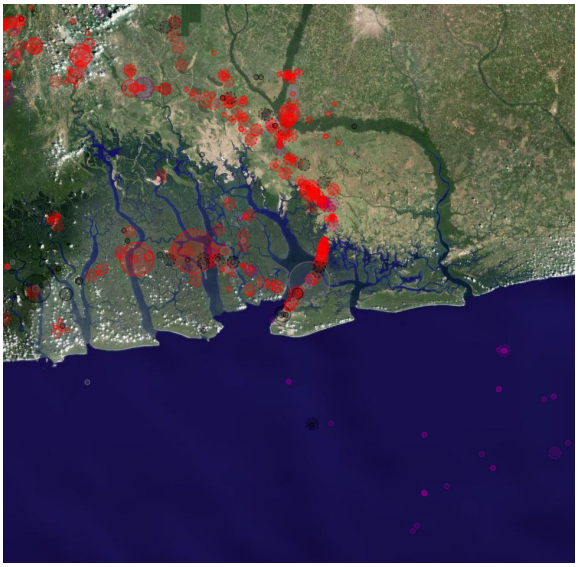
LGA built-in data, available off-line



MapBox



ESRI World



MDA NaturalVue

Map overlays

Overlays are layers of data on top of the map base layer.

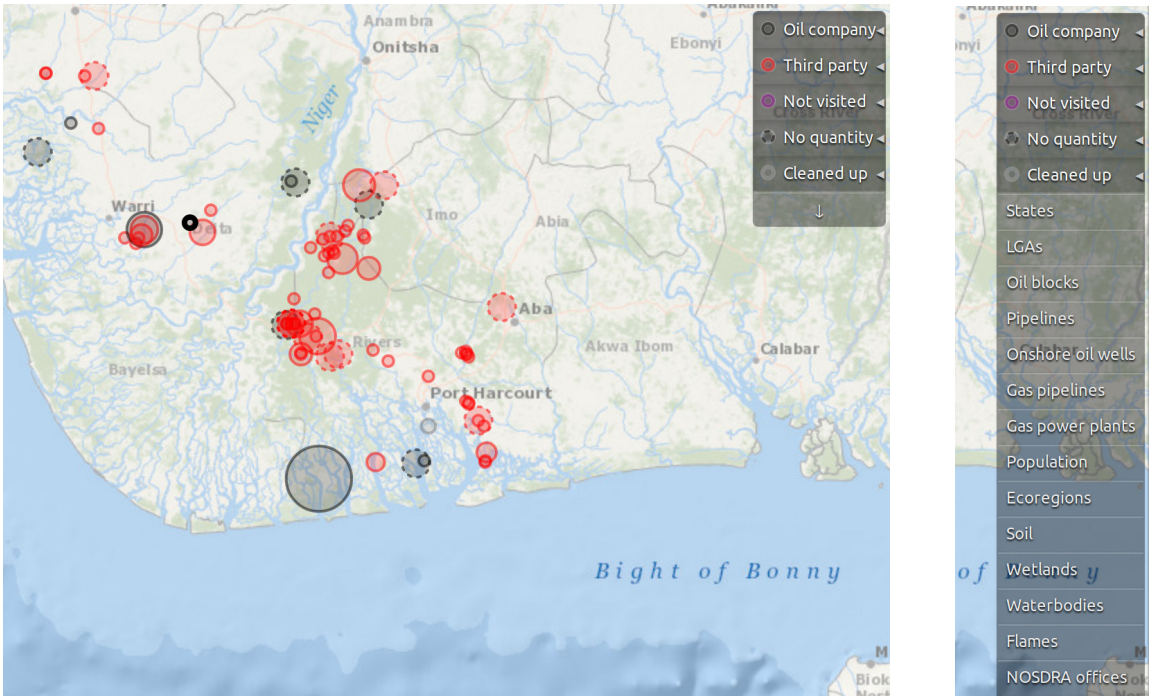


Figure 2.20: The overlay selector is initially folded up: click on the down arrow "↓" button to unfold it.

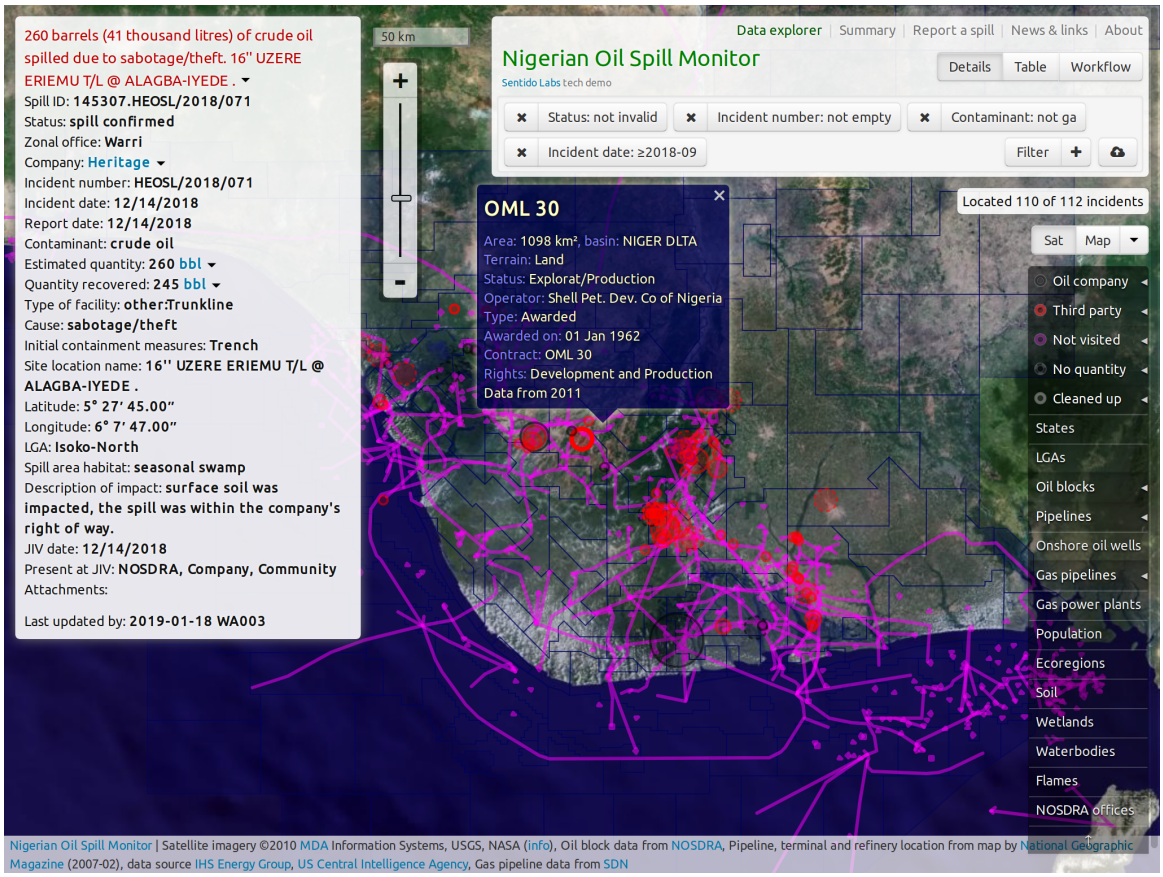


Figure 2.21: You can activate the data layers by clicking on them at the right side: in this case we have gas and oil pipelines and oil blocks. Clicking on an oil block (OML 30 here) pops-up its details.

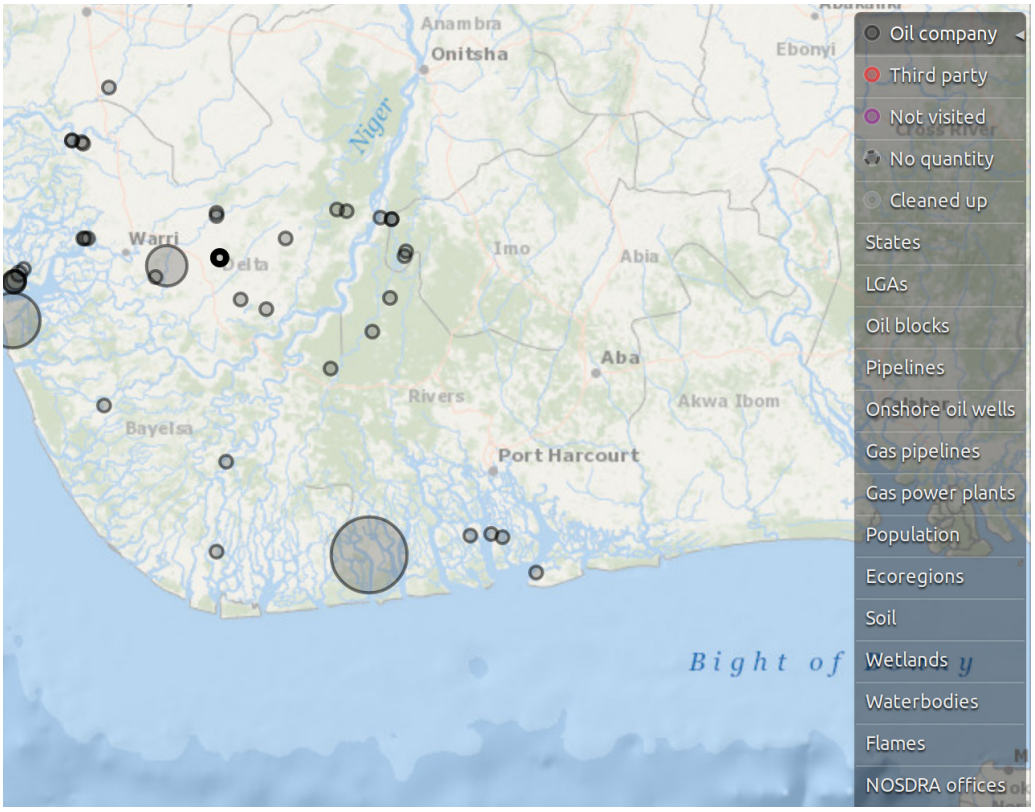


Figure 2.22: Oil company spills, those for which the operating company is considered responsible.

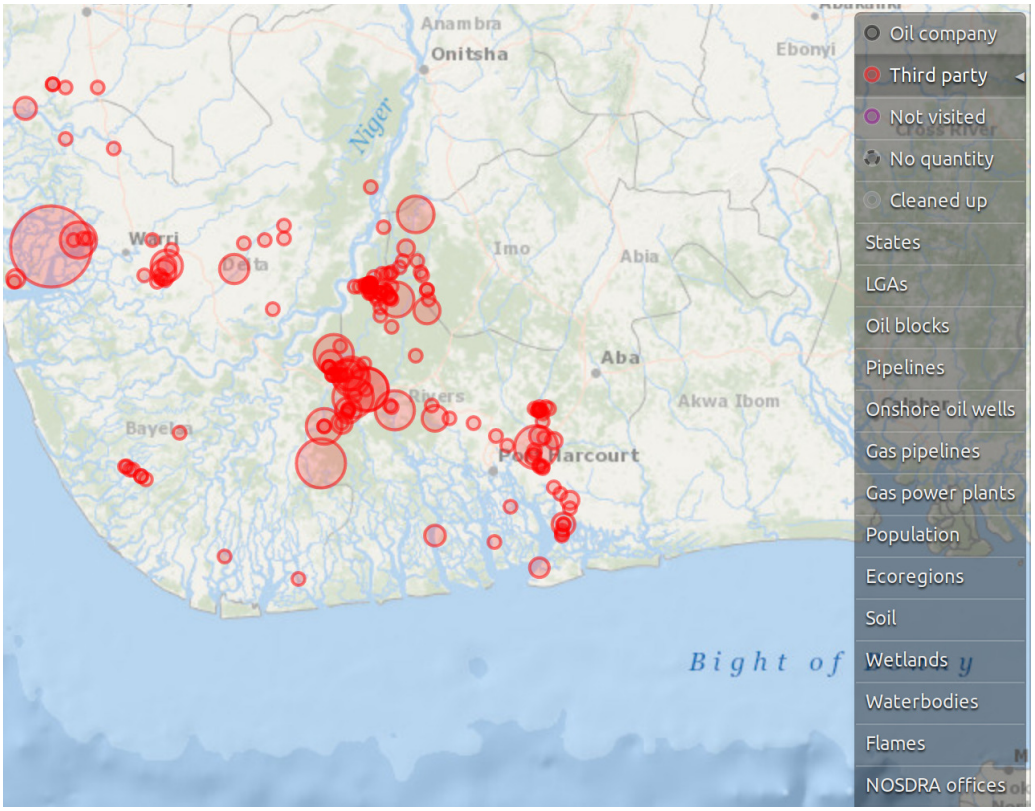


Figure 2.23: Third party spills, for instance due to oil theft, sabotage, or accidents. See SDN's report "'More Money, More Problems' – Economic Dynamics of the Artisanal Oil Industry in the Niger Delta Over Five Years"

.

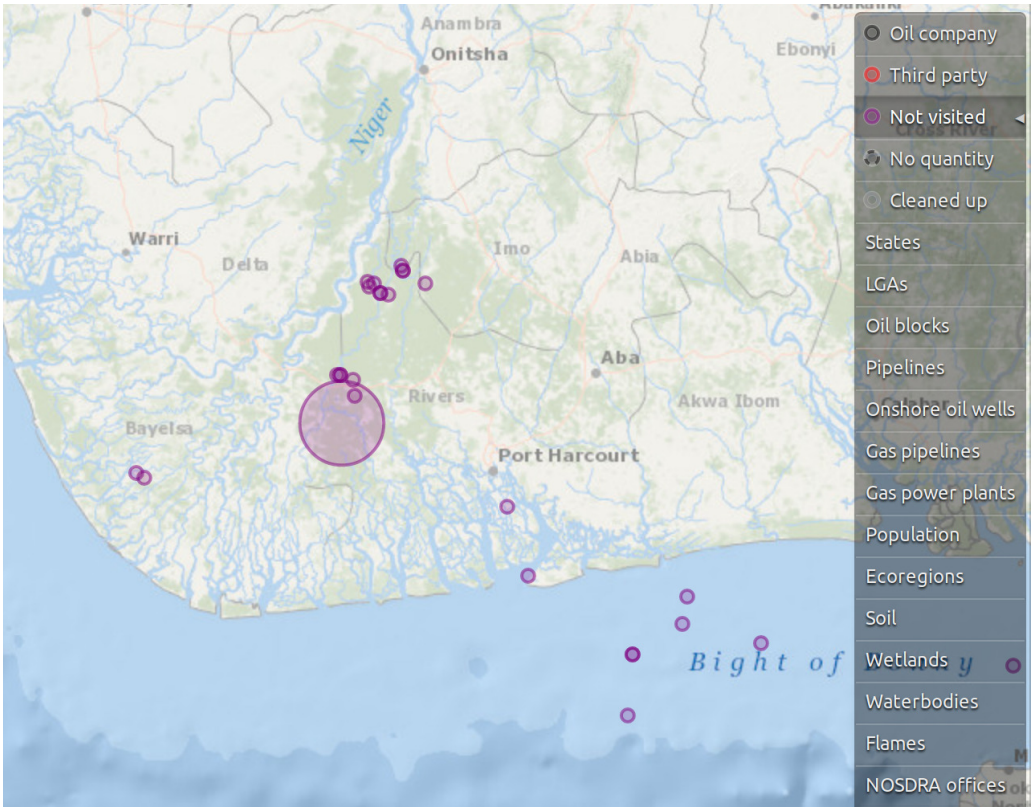


Figure 2.24: The Joint Investigation Visit (JIV) has not been conducted yet at these locations.

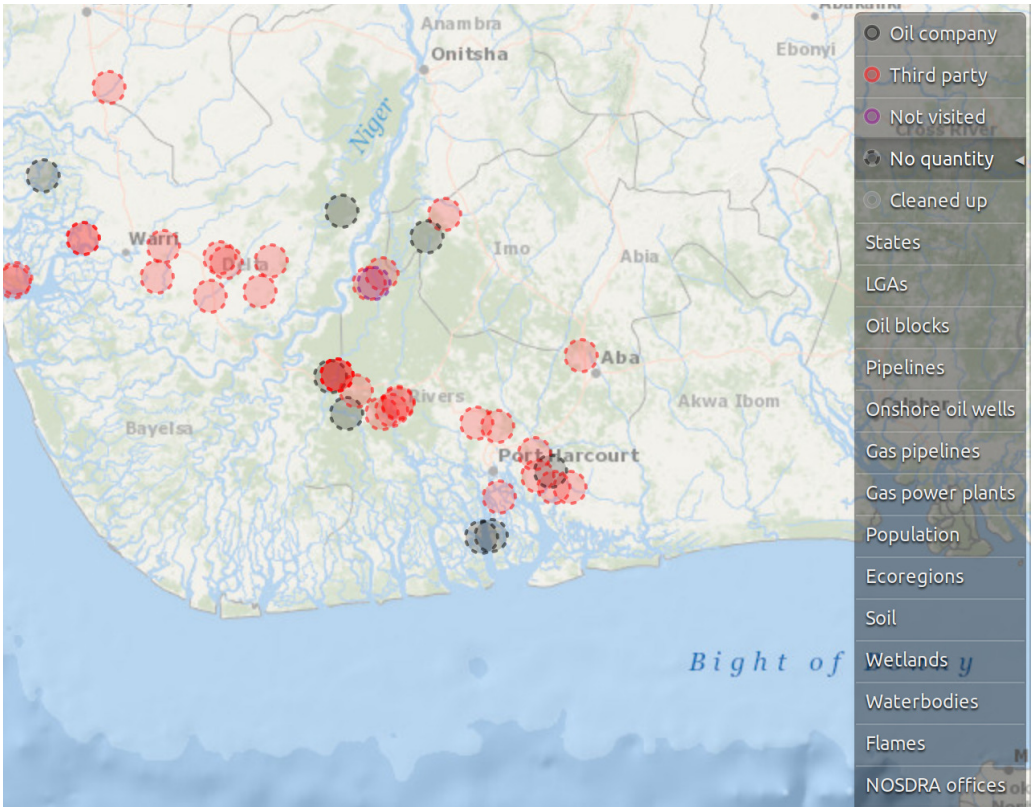


Figure 2.25: The amount of oil spilled could not be estimated in some incidents.

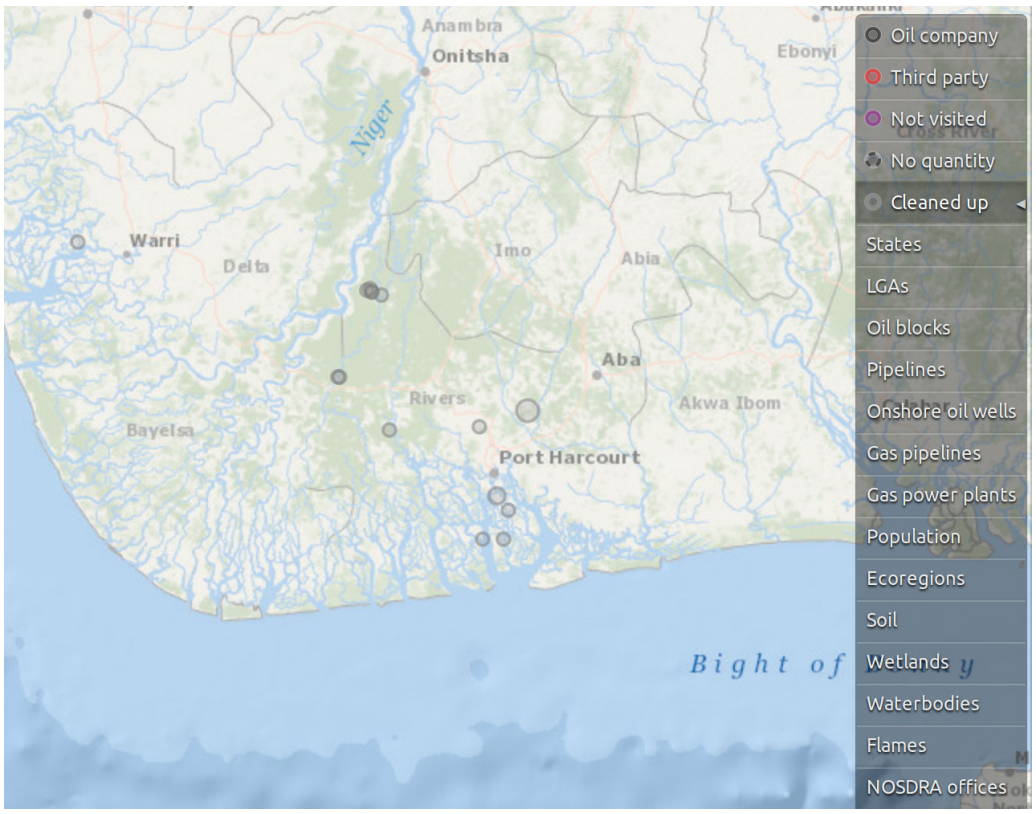


Figure 2.26: Spills that have been cleaned up appear in grey.

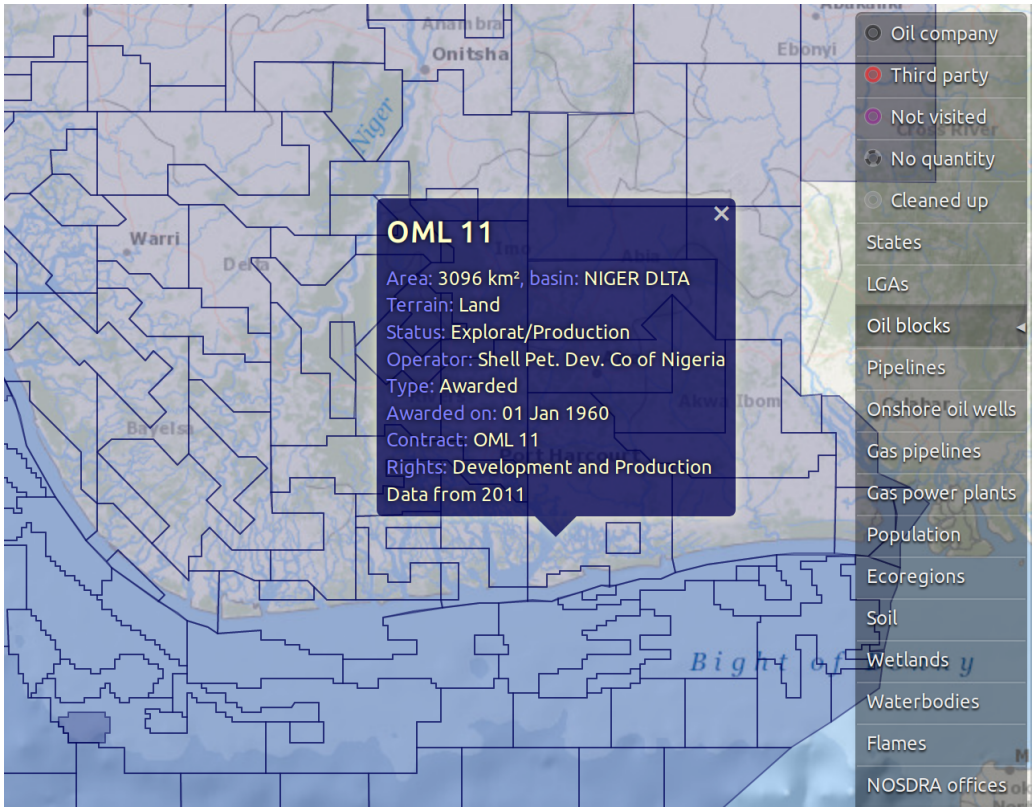


Figure 2.27: Oil blocks are exploitation concessions by the Nigerian Government to oil companies.

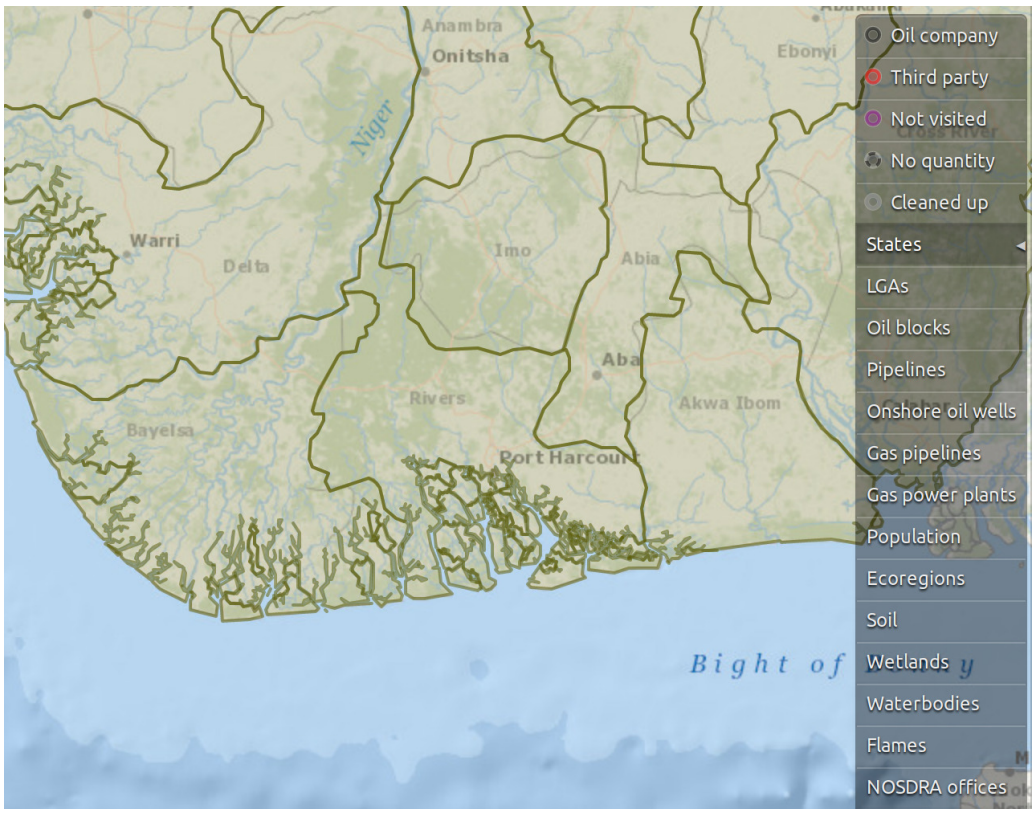


Figure 2.28: Nigerian states.

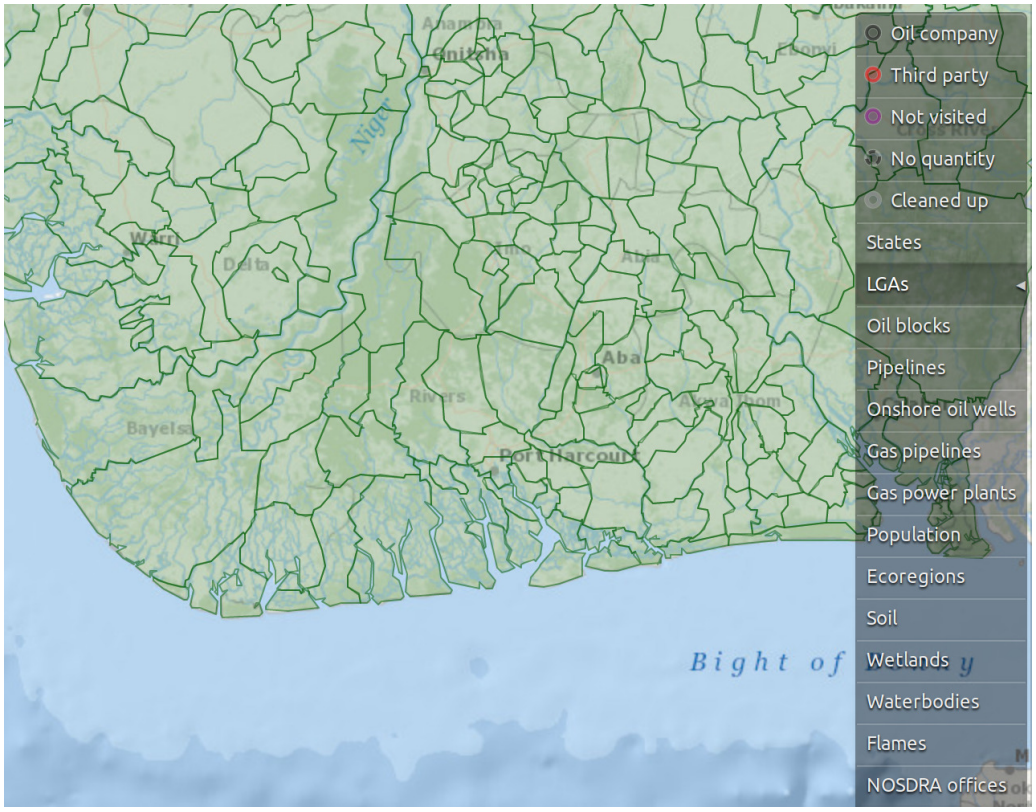


Figure 2.29: Local Government Areas (LGAs).

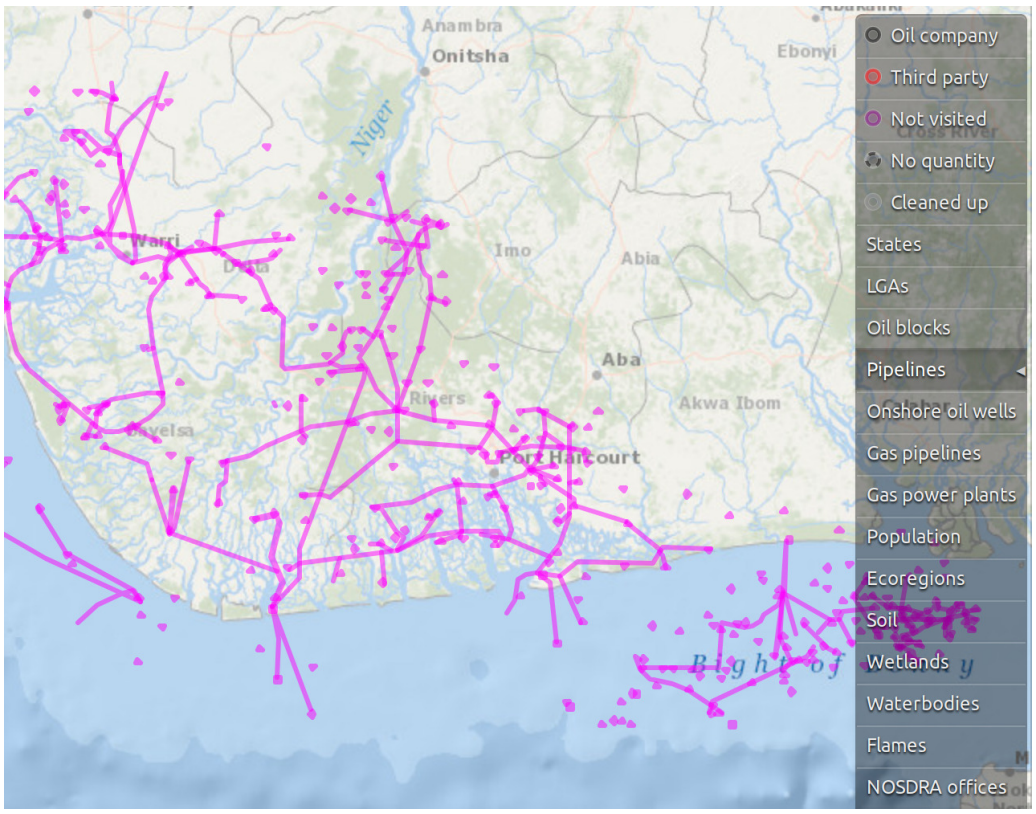


Figure 2.30: Pipelines.

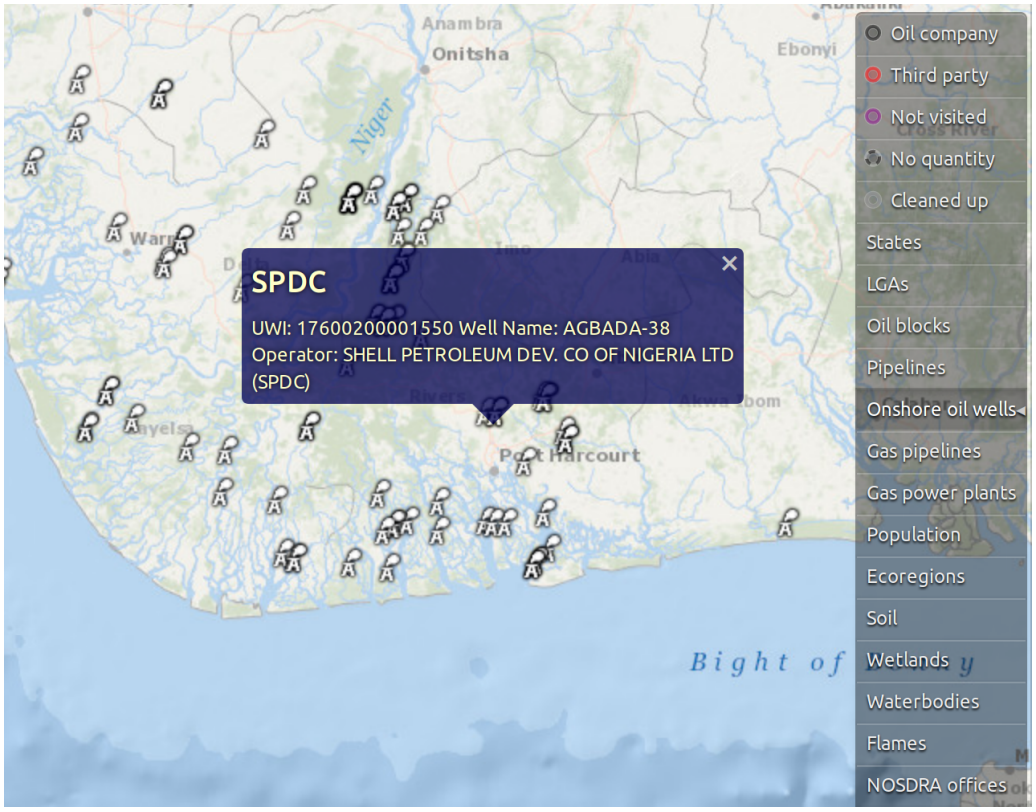


Figure 2.31: Onshore oil wells.

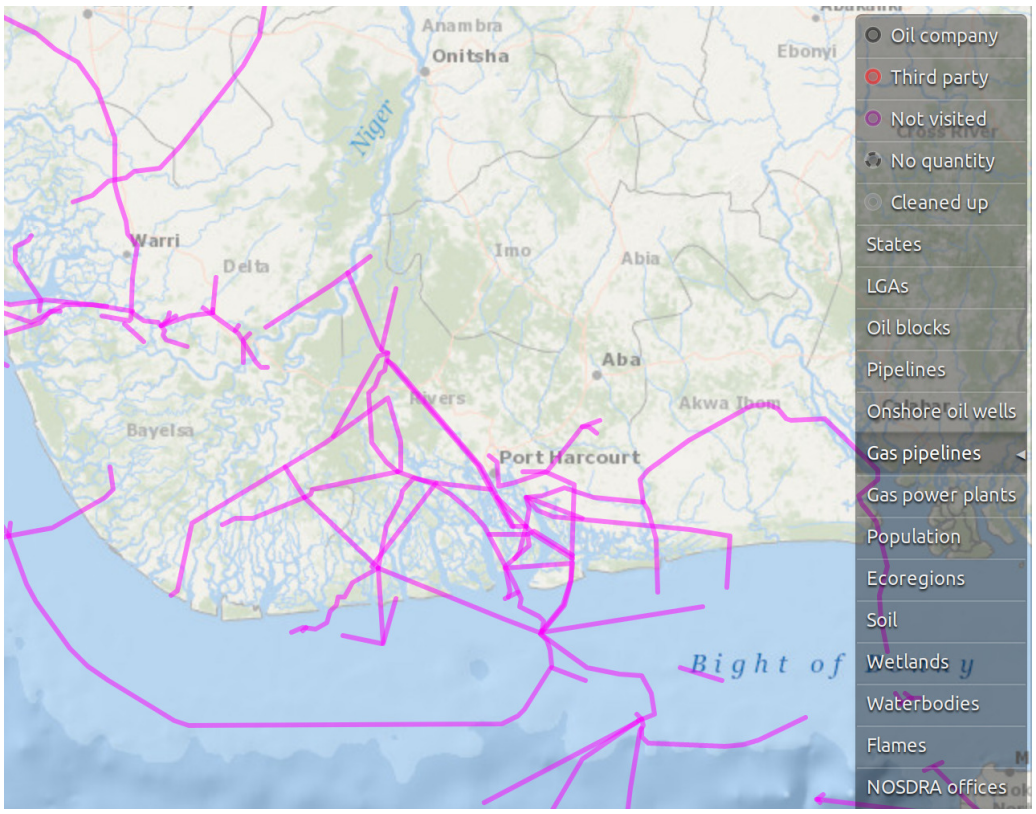


Figure 2.32: Gas pipelines.

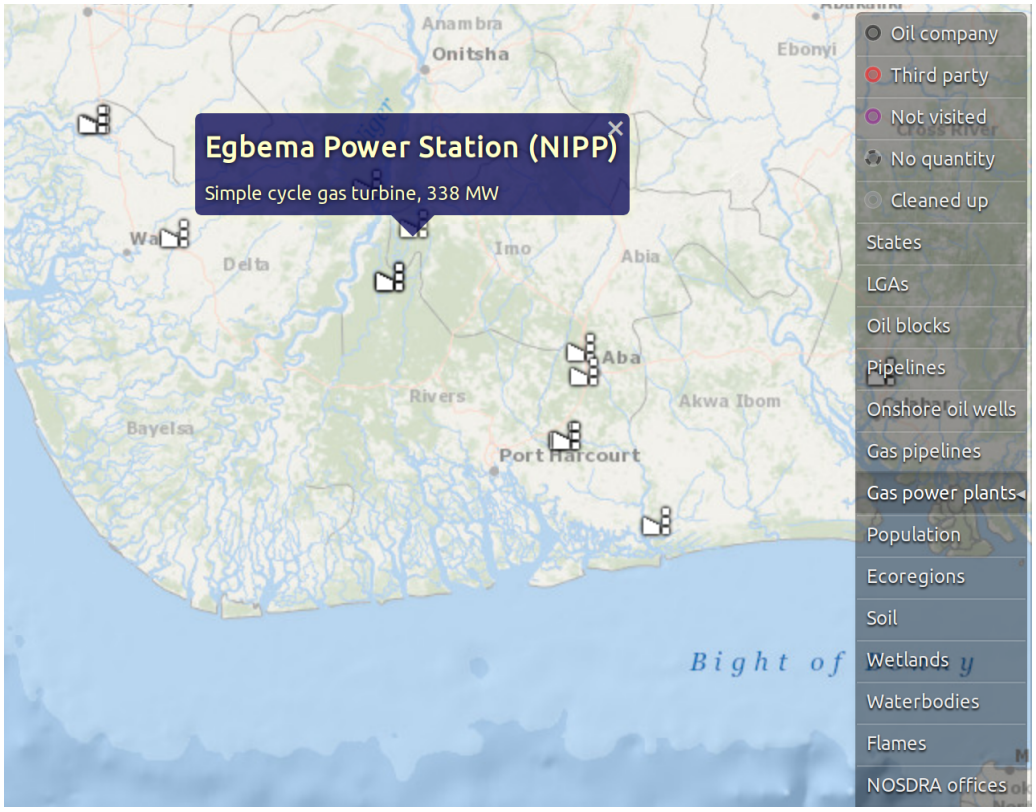


Figure 2.33: Gas power plants.

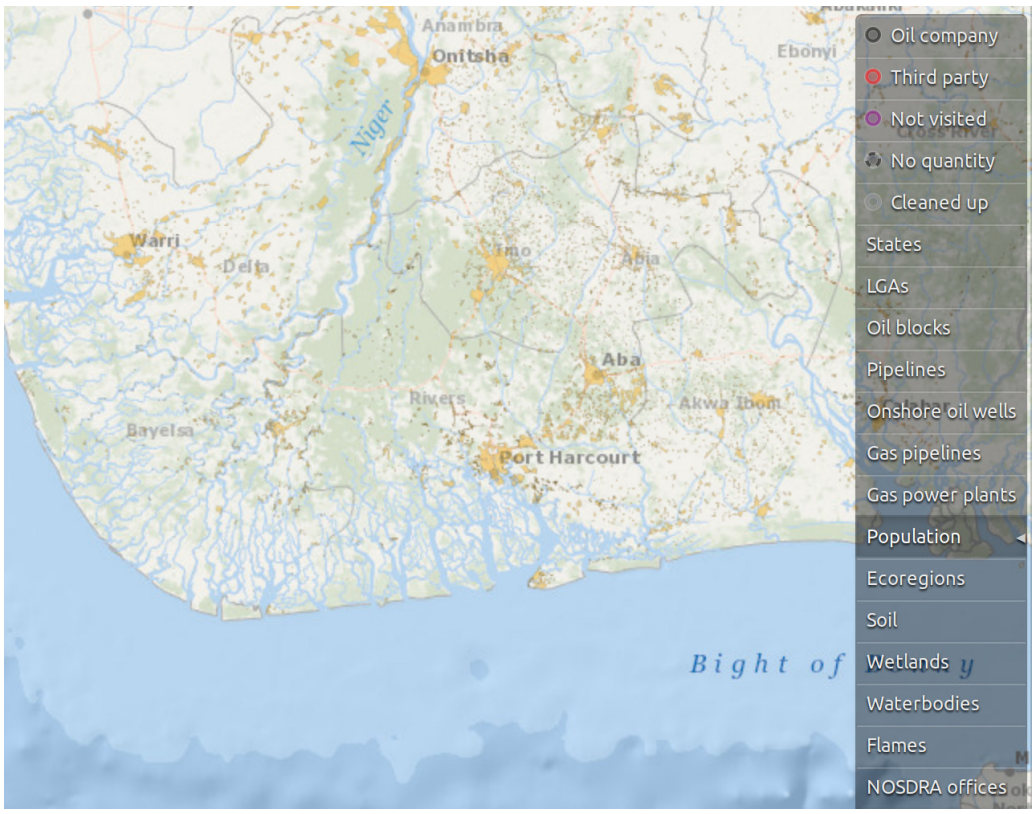


Figure 2.34: Population settlements, in yellow.

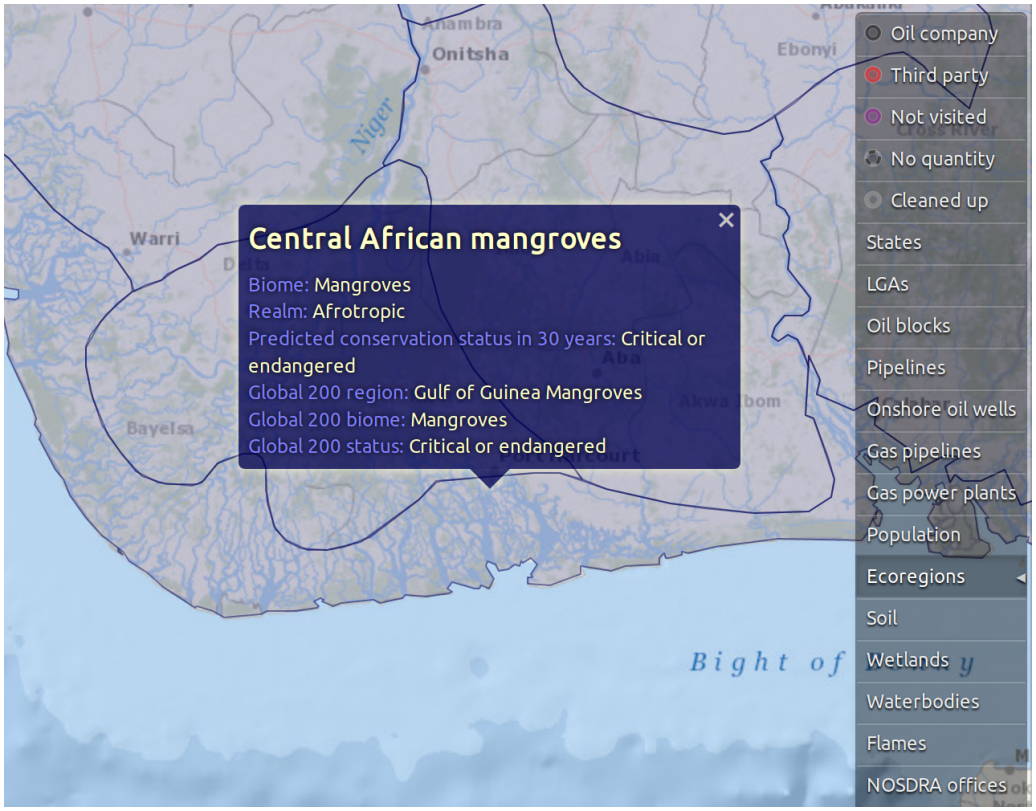


Figure 2.35: Ecological regions. Mangroves in particular are hit very hard by oil spills.

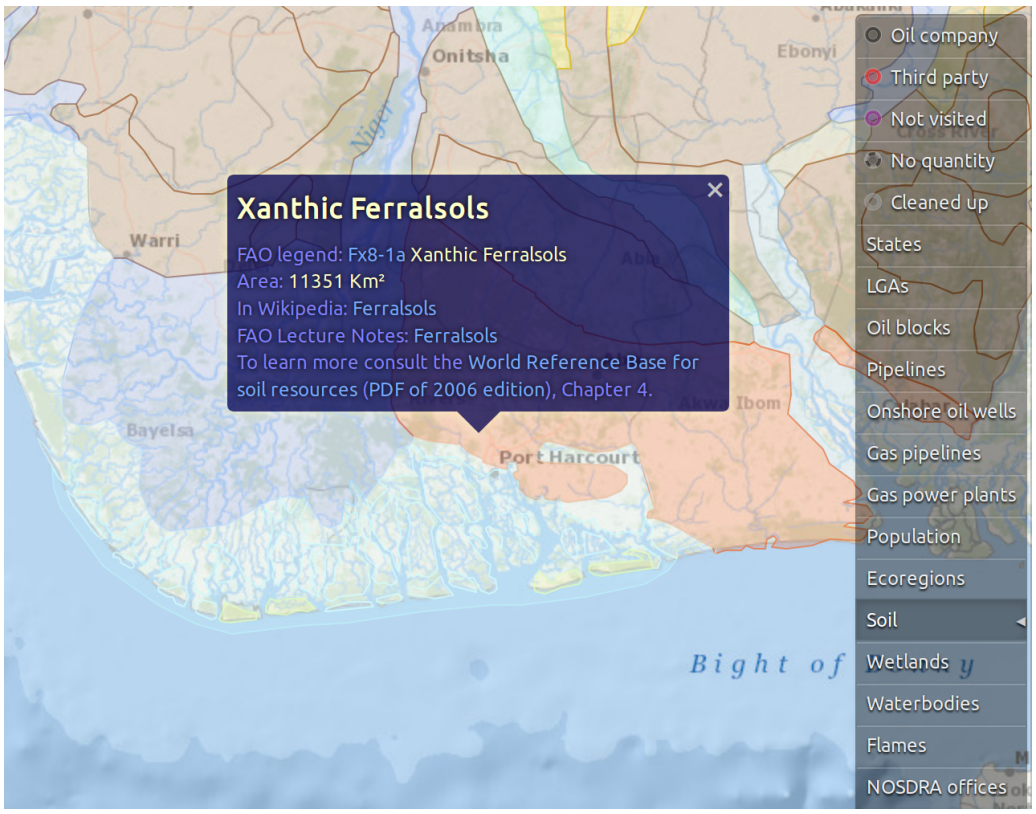


Figure 2.36: Soil types, with links to additional information in Wikipedia and the United Nations' FAO.

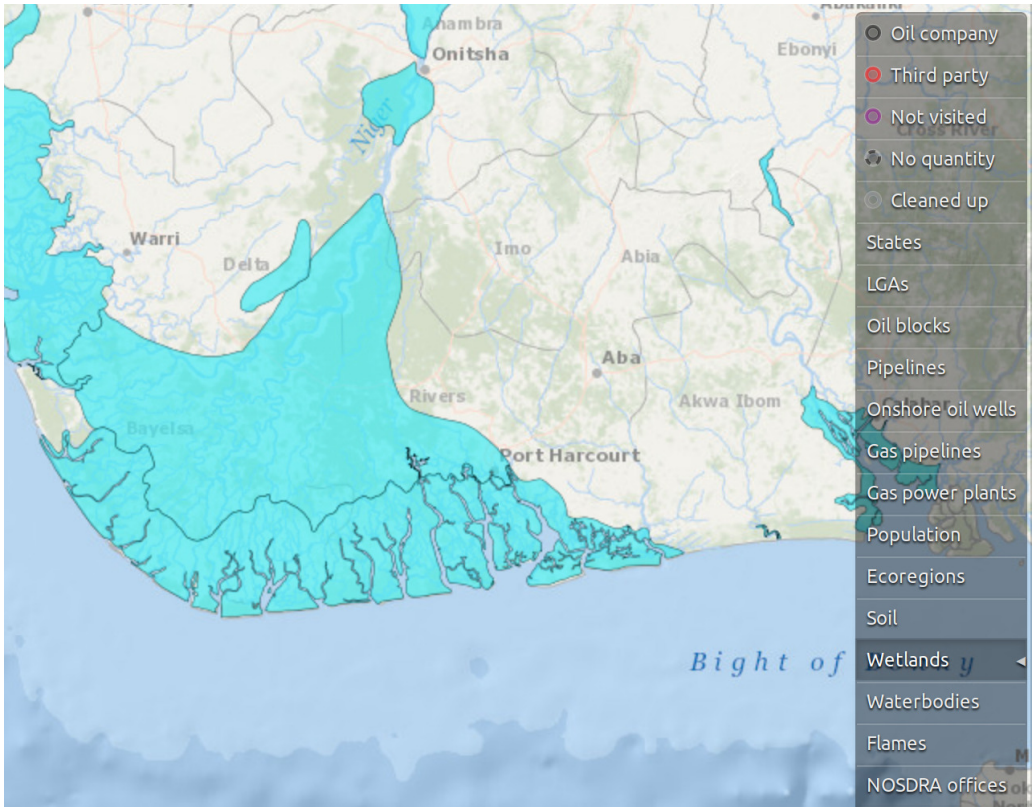


Figure 2.37: Wetlands.

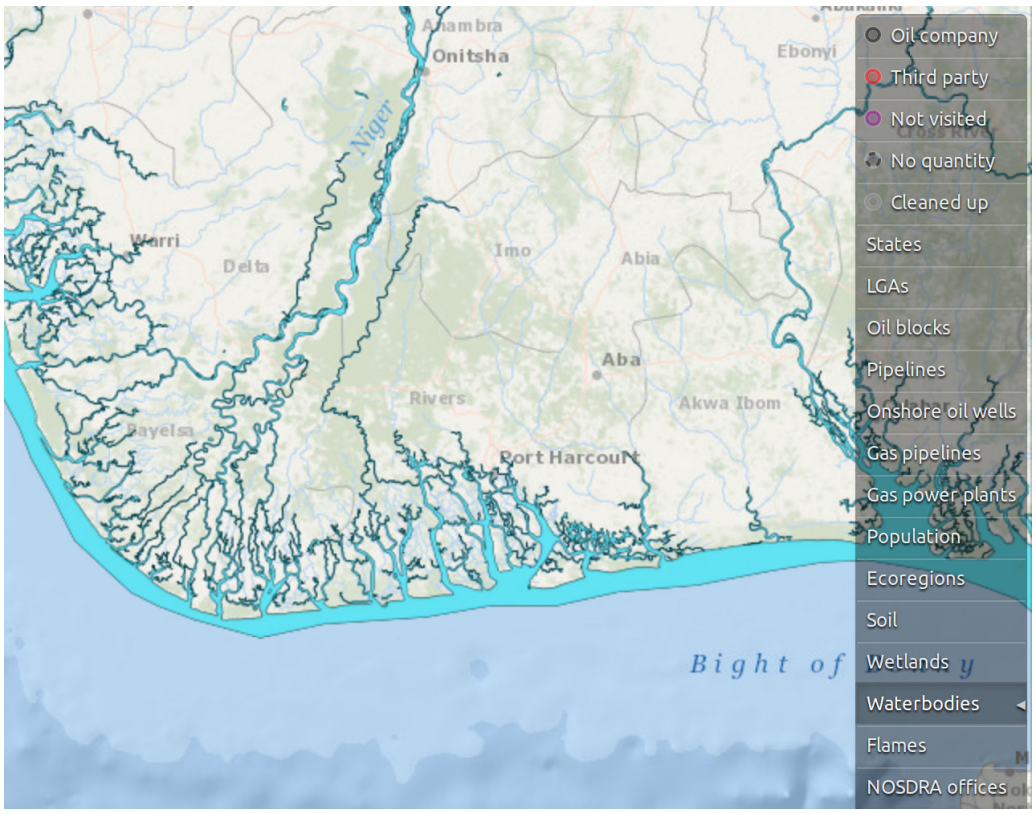


Figure 2.38: Water bodies.

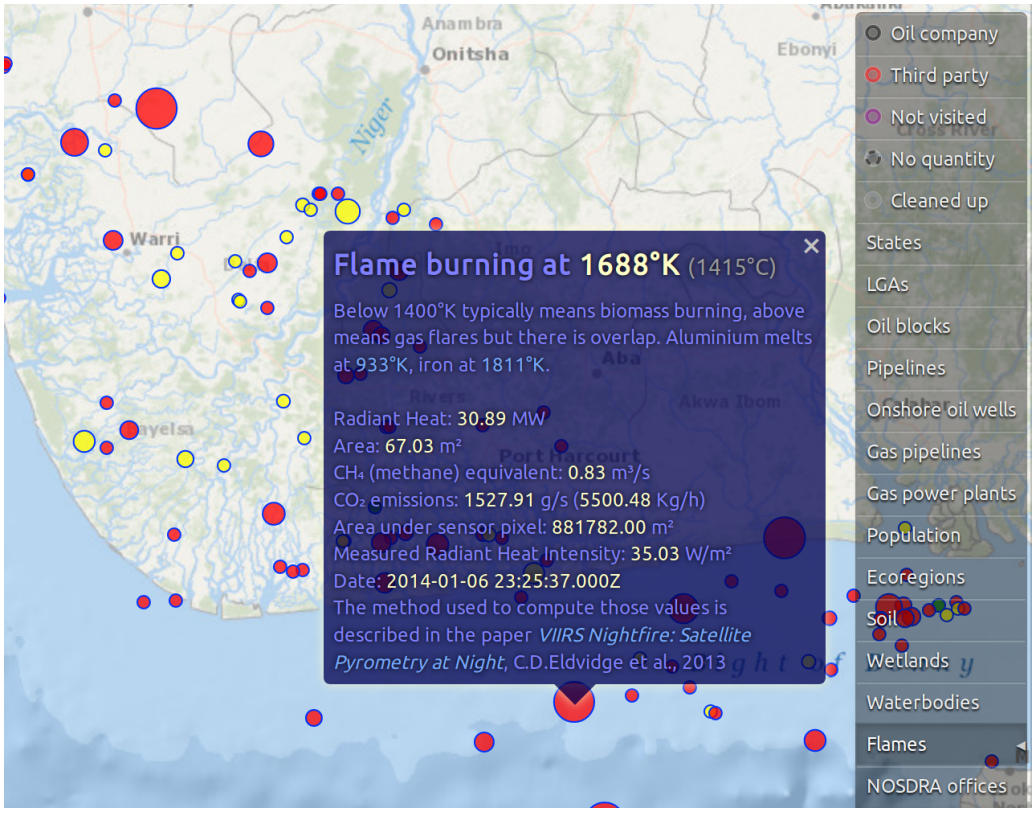


Figure 2.39: Flames detected by NOAA's satellites. Blue means lower temperature which points towards biomass (wood, grass) burning, red ones are very hot and likely gas flares, and yellow ones are in between. This was expanded into the Nigerian Gas Flare Tracker (GFT).

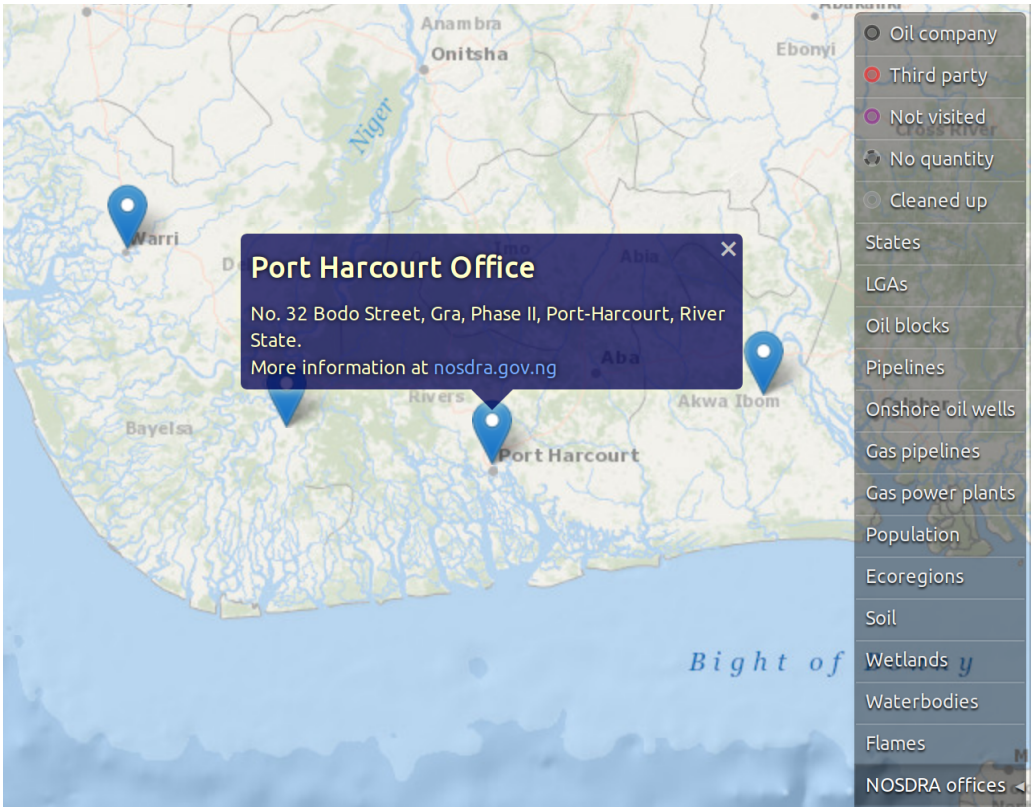
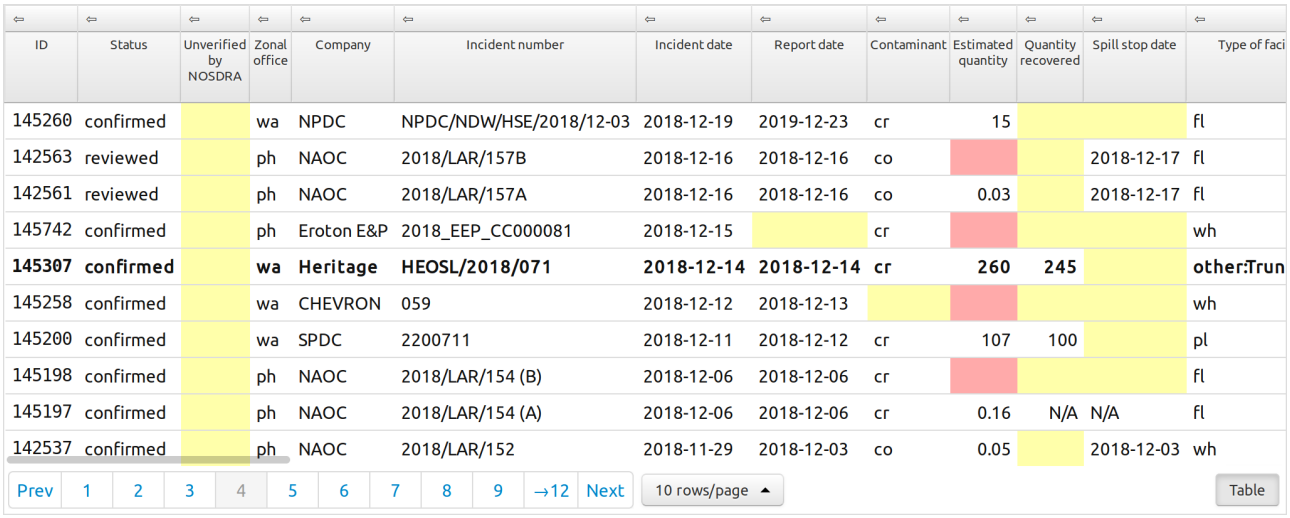


Figure 2.40: NOSDRA operates several Zonal Offices.

2.4 Table view

This view contains the table headers, the data rows, and at the bottom the pagination controls.

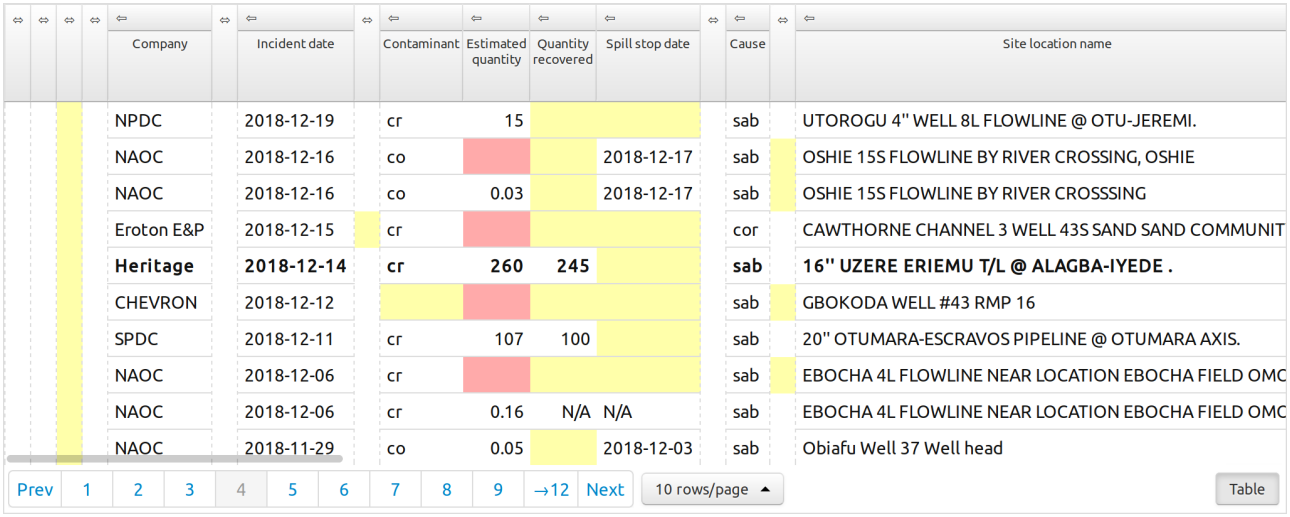


ID	Status	Unverified by NOSDRA	Zonal office	Company	Incident number	Incident date	Report date	Contaminant	Estimated quantity	Quantity recovered	Spill stop date	Type of faci
145260	confirmed		wa	NPDC	NPDC/NDW/HSE/2018/12-03	2018-12-19	2019-12-23	cr	15			fl
142563	reviewed		ph	NAOC	2018/LAR/157B	2018-12-16	2018-12-16	co			2018-12-17	fl
142561	reviewed		ph	NAOC	2018/LAR/157A	2018-12-16	2018-12-16	co	0.03		2018-12-17	fl
145742	confirmed		ph	Eroton E&P	2018_EEP_CC000081	2018-12-15		cr				wh
145307	**confirmed**		**wa**	**Heritage**	**HEOSL/2018/071**	**2018-12-14**	**2018-12-14**	**cr**	**260**	**245**		**other:Trun**
145258	confirmed		wa	CHEVRON	059	2018-12-12	2018-12-13					wh
145200	confirmed		wa	SPDC	2200711	2018-12-11	2018-12-12	cr	107	100		pl
145198	confirmed		ph	NAOC	2018/LAR/154 (B)	2018-12-06	2018-12-06	cr				fl
145197	confirmed		ph	NAOC	2018/LAR/154 (A)	2018-12-06	2018-12-06	cr	0.16	N/A	N/A	fl
142537	confirmed		ph	NAOC	2018/LAR/152	2018-11-29	2018-12-03	co	0.05		2018-12-03	wh

Prev 1 2 3 4 5 6 7 8 9 →12 Next 10 rows/page Table

Figure 2.41: Table view with the incident `HPDC/HSE/IYE/001/17` selected. Empty cells are coloured **yellow**, and **red** cells are those that should have a value but do not.

Each header corresponds to one field in the database and contains two buttons: the top one is the **column folding** control to hide momentarily columns we do not need to see right now, and the bottom one with the field name triggers **sorting** of the table rows according to that field. Each time you click it, the sorting direction changes from ascending to descending and viceversa.



Company	Incident date	Contaminant	Estimated quantity	Quantity recovered	Spill stop date	Cause	Site location name
NPDC	2018-12-19	cr	15			sab	UTOROGU 4" WELL 8L FLOWLINE @ OTU-JEREMI.
NAOC	2018-12-16	co			2018-12-17	sab	OSHIE 15S FLOWLINE BY RIVER CROSSING, OSHIE
NAOC	2018-12-16	co	0.03		2018-12-17	sab	OSHIE 15S FLOWLINE BY RIVER CROSSSING
Eroton E&P	2018-12-15	cr				cor	CAWTHORNE CHANNEL 3 WELL 43S SAND SAND COMMUNIT
Heritage	**2018-12-14**	**cr**	**260**	**245**		**sab**	**16" UZERE ERIEMU T/L @ ALAGBA-IYEDE .**
CHEVRON	2018-12-12					sab	GBOKODA WELL #43 RMP 16
SPDC	2018-12-11	cr	107	100		sab	20" OTUMARA-ESCRAVOS PIPELINE @ OTUMARA AXIS.
NAOC	2018-12-06	cr				sab	EBOCHA 4L FLOWLINE NEAR LOCATION EBOCHA FIELD OMC
NAOC	2018-12-06	cr	0.16	N/A	N/A	sab	EBOCHA 4L FLOWLINE NEAR LOCATION EBOCHA FIELD OMC
NAOC	2018-11-29	co	0.05		2018-12-03	sab	Obiafu Well 37 Well head

Prev 1 2 3 4 5 6 7 8 9 →12 Next 10 rows/page Table

Figure 2.42: Some columns are folded using the arrow buttons on top of the headers.

The pagination control has buttons for the **Prev**ious / **Next** page, for jumping to the first page (**1**←) or the last (→**12** indicates that there are 12 pages in total), and between them are direct links to the pages surrounding the current one which is colored in grey.

Next to the pagination control there is a selector for the number of rows to show in each page: the initial value is 10 to avoid covering up the map but it can be increased up to 100 rows per page.

The Table button at the bottom-right corner is the same as the one at the top-right. It is duplicated here for cases where the table view gets tall enough to cover the other buttons, so that the user always has a way of closing the table view even on small screens.

2.5 Workflow view

The Workflow view contains pre-made filters that are combined with the currently active filters from the query/filter bar at the top. Clicking on an incident number (blue) selects that incident in all views.

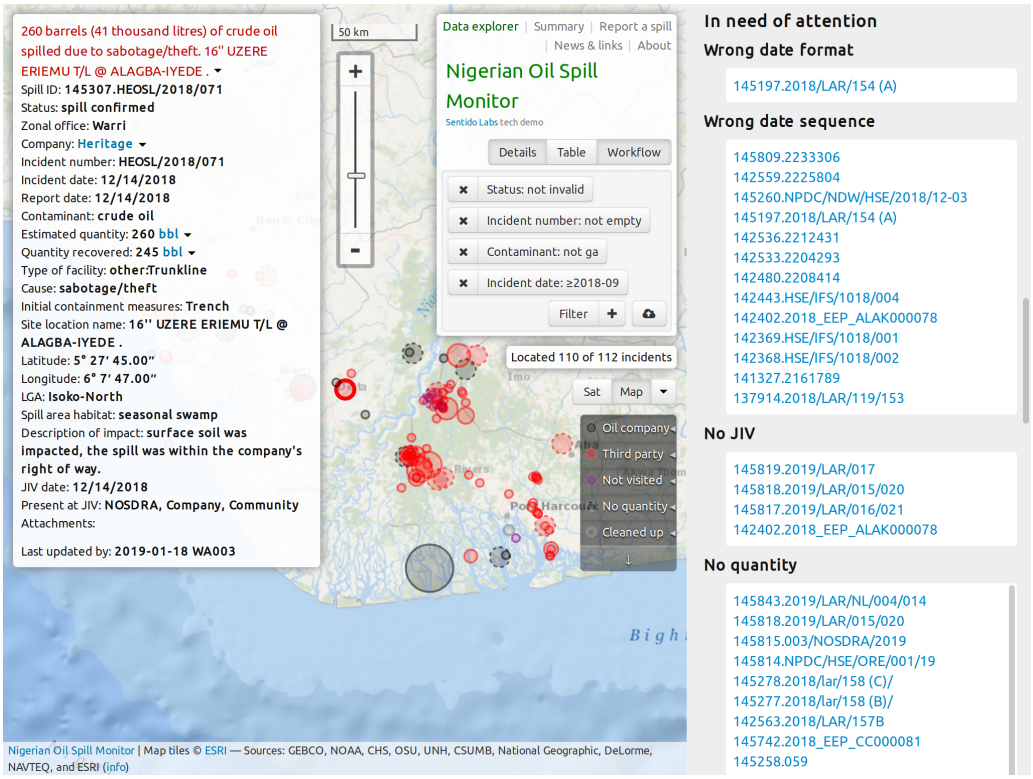


Figure 2.43: When you activate the Workflow view by clicking on the button near the top-right corner, the Map view shrinks to make space for it on the right side.

- In need of attention
  - Wrong date format: one of the date fields contains a value in the wrong format, such as "2/5/2014".
  - Wrong date sequence: date values are out of sequence. For instance, the JIV date is earlier than the incident date, or any of them is in the future.
  - No JIV: reports without a JIV date.
  - No quantity: there is no estimated quantity reported.
  - Marked "new" for over one month.
  - Contaminant spill unspecified.
  - Awaiting confirmation.
- Recent spills
  - Latest spills: incidents in the last two months.
  - Recent spills without JIV: reports in the last two months still without a JIV date.
- By size: these are classified according to the amount of contaminant spilled, with different thresholds depending on the sensitivity of the ecosystem: swamps and inland waters are considered more delicate.
  - Very minor spills: less than 2.5 bbl for delicate ecosystems, or 25 bbl for others.
  - Minor spills: less than 25 or 250 bbl respectively.
  - Medium spills: less than 250 or 2500 bbl.
  - Major spills: above 250 or 2500 bbl.
- Gas leaks: spills where the contaminant is gas instead of oil. These typically do not need clean-up.
- Outstanding cleanup

  - Minor spills, classified as above in By size.
  - Medium spills.
  - Major spills.
- Where compensation is due: spills under the operating Oil Company's responsibility.

2.6 Summary page

You can reach this from the navigation menu at the top of the page, by clicking on “Summary”.
It displays the same data you see in the main page (map, detail and table views) but in aggregated form by month, year, or as a single total of all incidents.

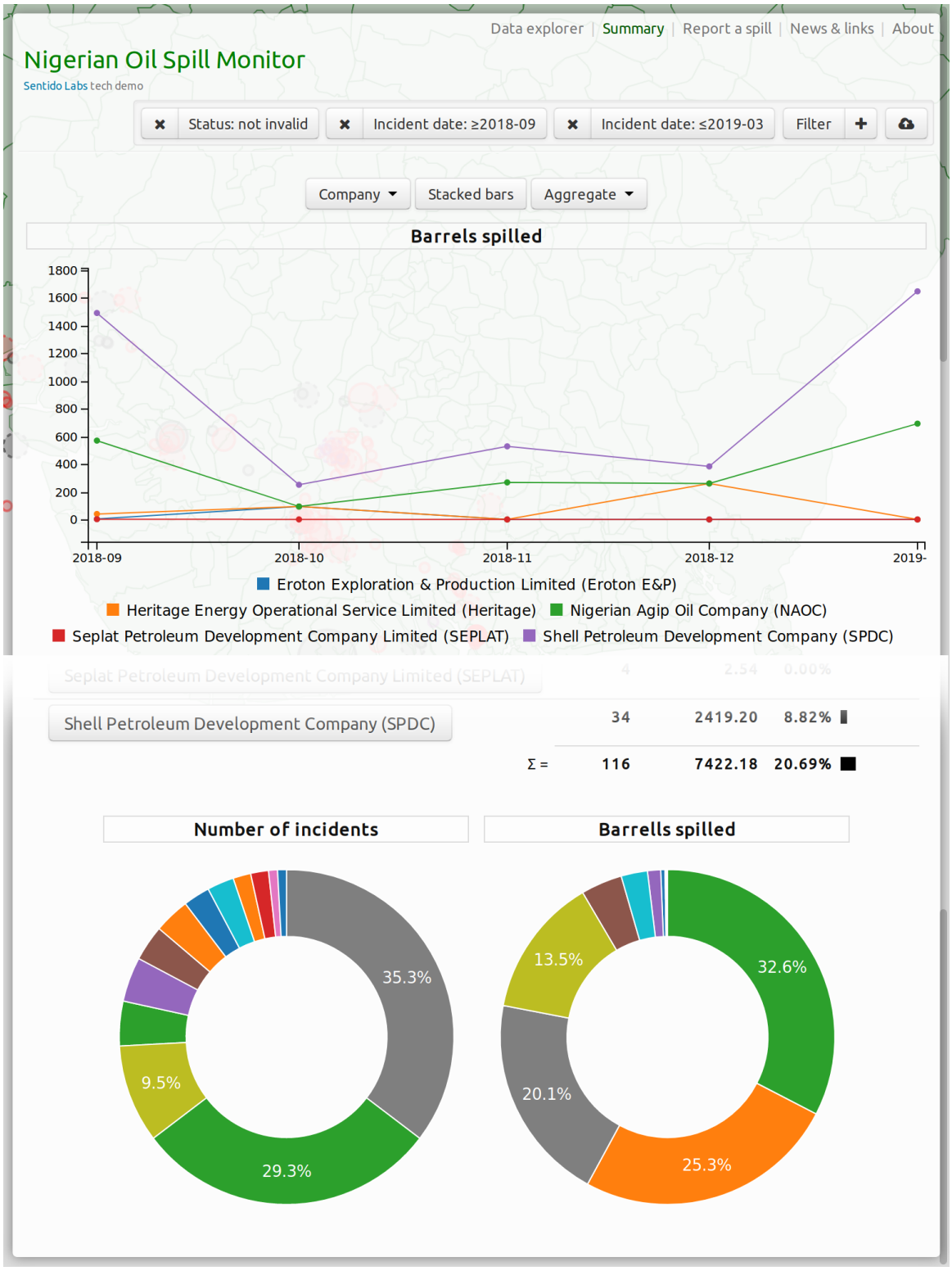
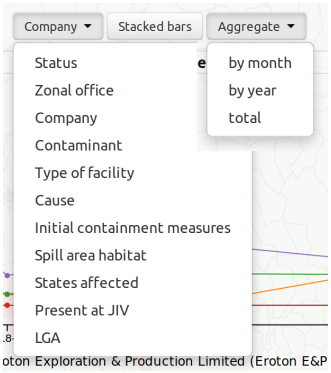
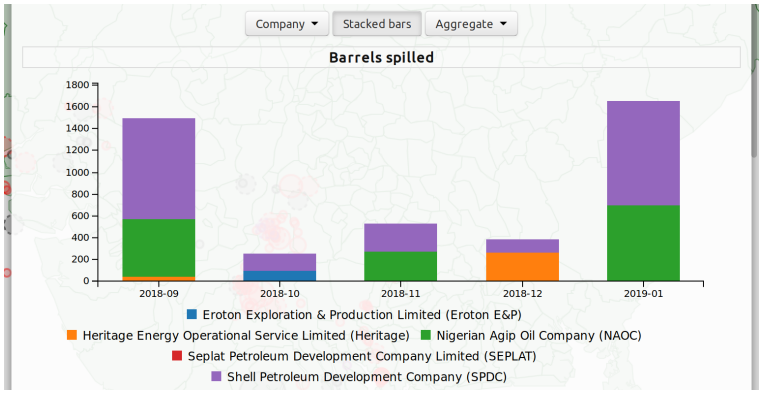


Figure 2.44: The top part contains a time-series graph of the spill amount in barrels, initially by company and month. Below (cut out for brevity) are tables with figures per month, and at the bottom the total amounts per company as doughnut charts.



These menues specify the aggregation dimensions.



The time series chart can be displayed as lines or as stacked bars.

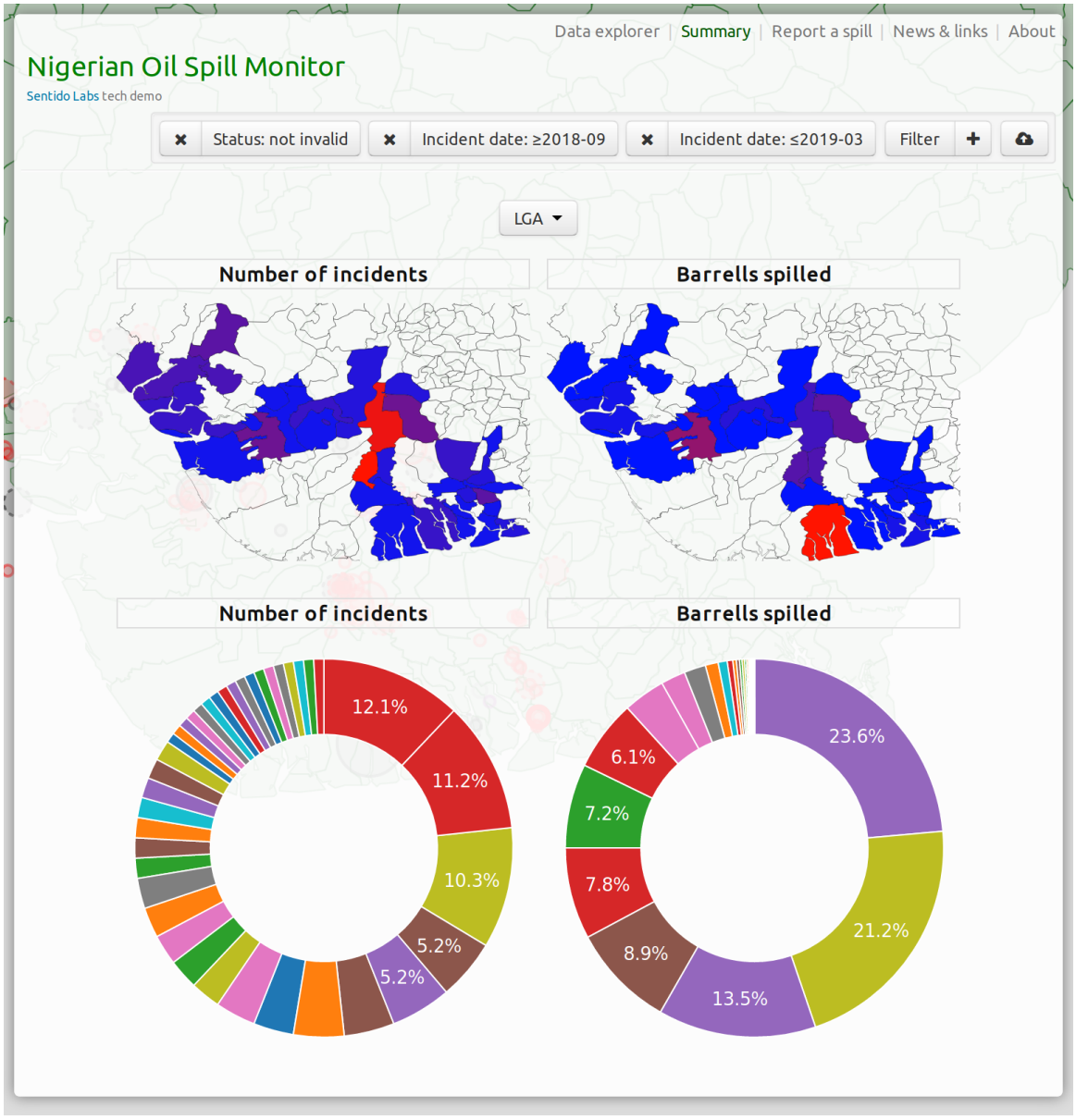


Figure 2.46: If you select a field referring to a geographical area, the graph switches automatically to a choropleth map.

Drill down

Clicking on a value in the tables (the headers on the left), on the doughnut charts (segments) or on the choropleth map (areas) will add a filter for that specific value, and also add one pair of doughnut charts at the bottom for each other dimension.

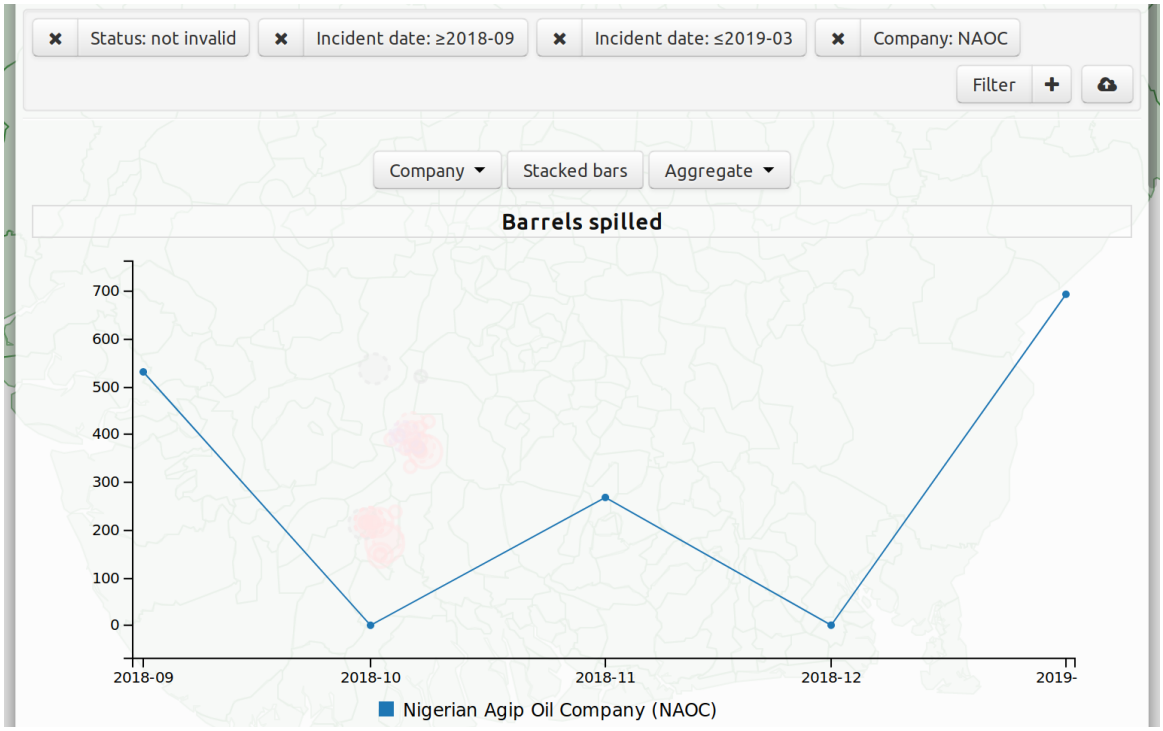


Figure 2.47: Clicking on a specific company adds a filter for it (top filter toolbar, right side). To remove it, either click again on the same value, or use the  button on the filter. These filters work exactly the same as if you added them by hand.

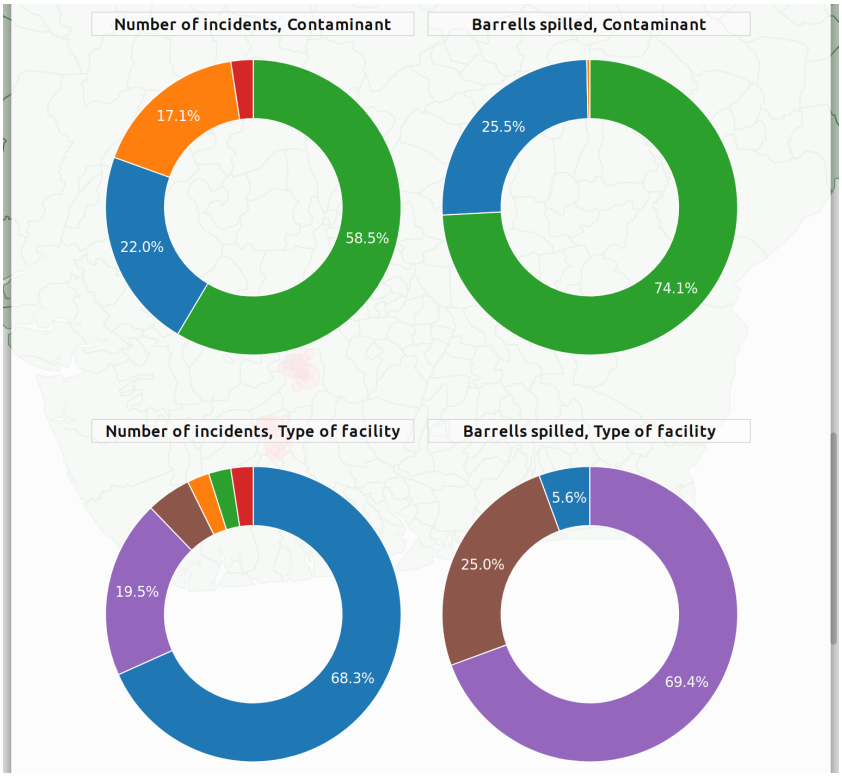
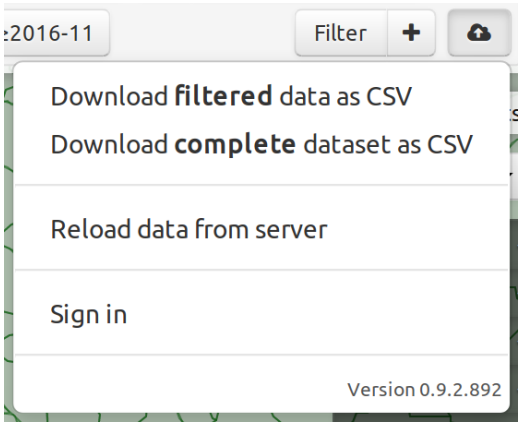


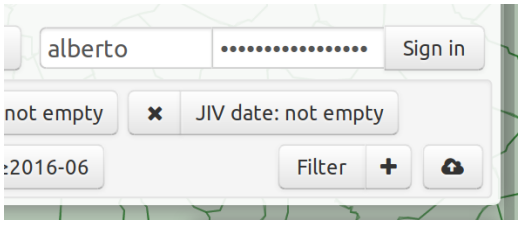
Figure 2.48: Some additional charts that appear when drilling-down.

Entering data

The first thing you need before entering reports is to sign in to the system with your user account, using the Sign in option in the menu that pops up when you click on the cloud button, to the right of the Filter toolbar.



Sign in from the **cloud button** menu.



Enter your user name and password, and click “Sign in”.

Figure 3.1: Sign-in process.

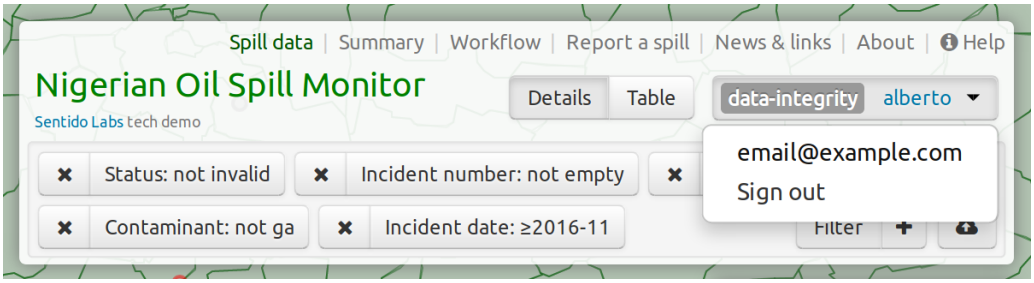


Figure 3.2: Once you sign in, you will see your user name with a badge indicating the account type (in this case “data-integrity”) and, if you click on it, you get a menu reminding you of the e-mail address associated with your account, and the **Sign out** option.

3.1 Storing or cancelling changes

When you create or update a report in the Detail View, the changes you make are instantly reflected in the other views, but those changes are only temporary until you click the Store changes button.

The Store changes button stores the modified report locally, in your computer, but does not send it yet to the server. The cloud button lights up in yellow to warn you that there are changes made locally that are still not comitted to the central database in the server.

The application will try to keep your changes until they are stored safely in the server, but the browser might discard them if you reload the page or close the window/tab.

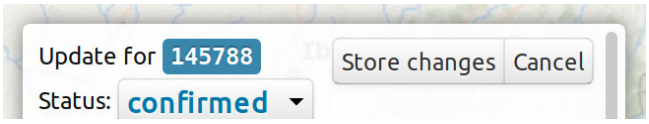


Figure 3.4: The Cancel button will restore the report to its state before you started modifying it.

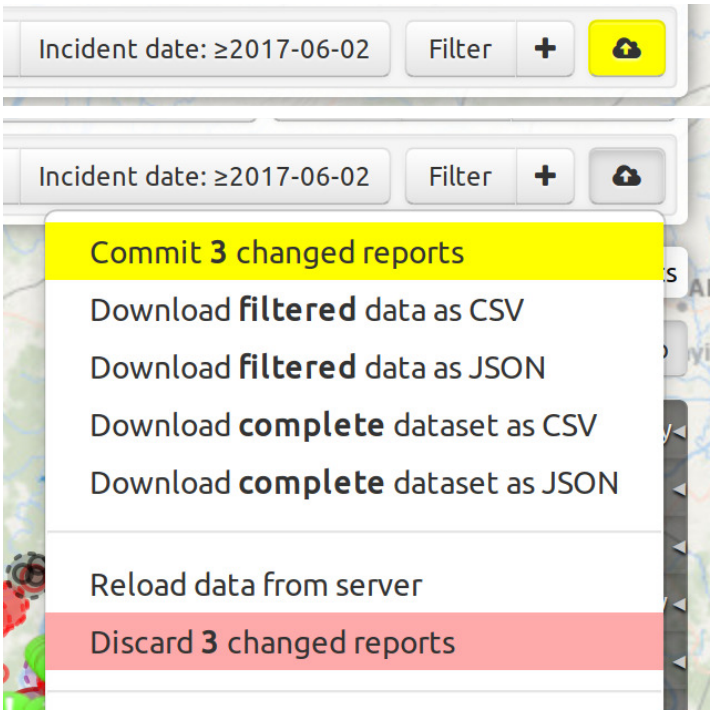


Figure 3.5: Commit will send your changes to the server to be stored in the central database. Discard will erase them as if you had never made them.

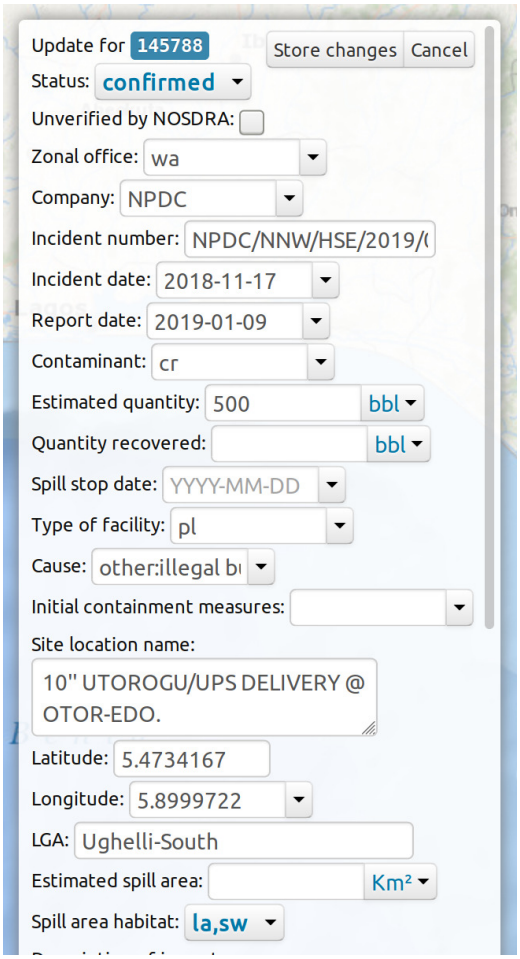


Figure 3.3: Updating a report in the Detail view (§ 2.2 *Detail view*).

To store your changes permanently in the central database, use the Commit option in the cloud menu, right of the filter toolbar.

You can decide whether you want to send the reports to the server after each change or in bigger batches: the best option depends on the network connection, but in general it is more efficient to send a handful of them at a time.

They are **permanently** stored **only after** the **yellow light** in the cloud button **turns off**.

3.2 Data format validation

Some fields require specific formats for the data. Two examples are dates and amounts.

This application is built on the assumption that the user knows best, so it only warns them about wrong/unexpected input instead of blocking them.

For fields that have a list of acceptable values, such as the company name, I chose to allow the user to override the application's list because since there might not be funding for maintenance, allowing only the expected values would prevent them from doing their work in the future.

The downside is that users will occasionally enter alternate forms of values such as "Shell" for "SPDC", but that can be fixed by users while there is nothing they could do about missing values (for instance when a new company starts operating) if the system forced them to pick from a list.

Several common mistakes can be detected using the Workflow view described in § 2.5 *Workflow view*.

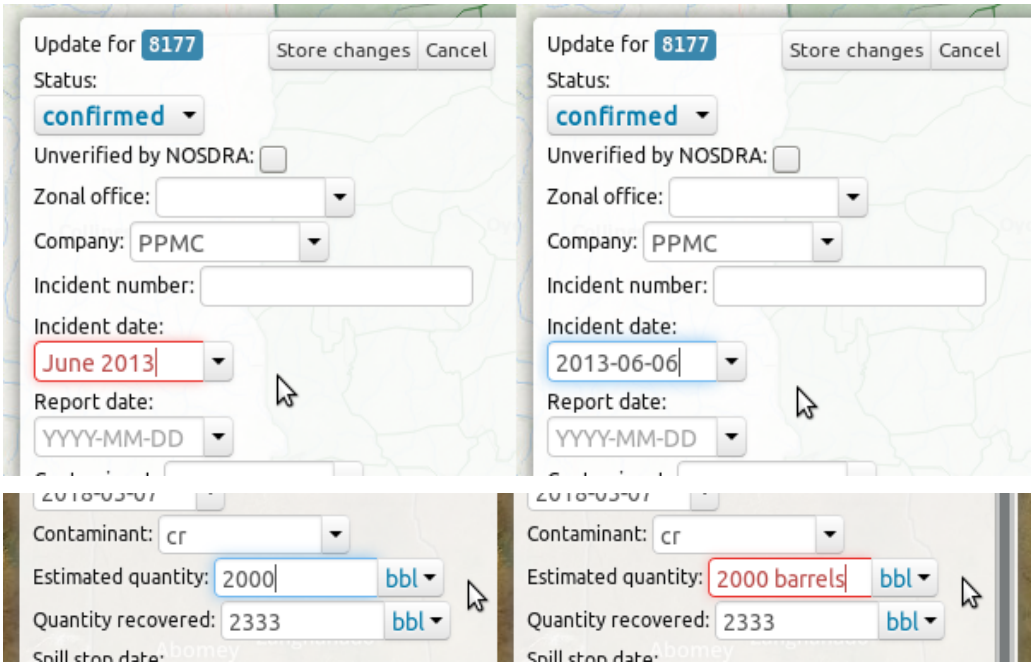


Figure 3.6: Value format mismatches are indicated by the red colour for the field border and value. The required date format is displayed to the user by the greyed-out pattern "YYYY-MM-DD".

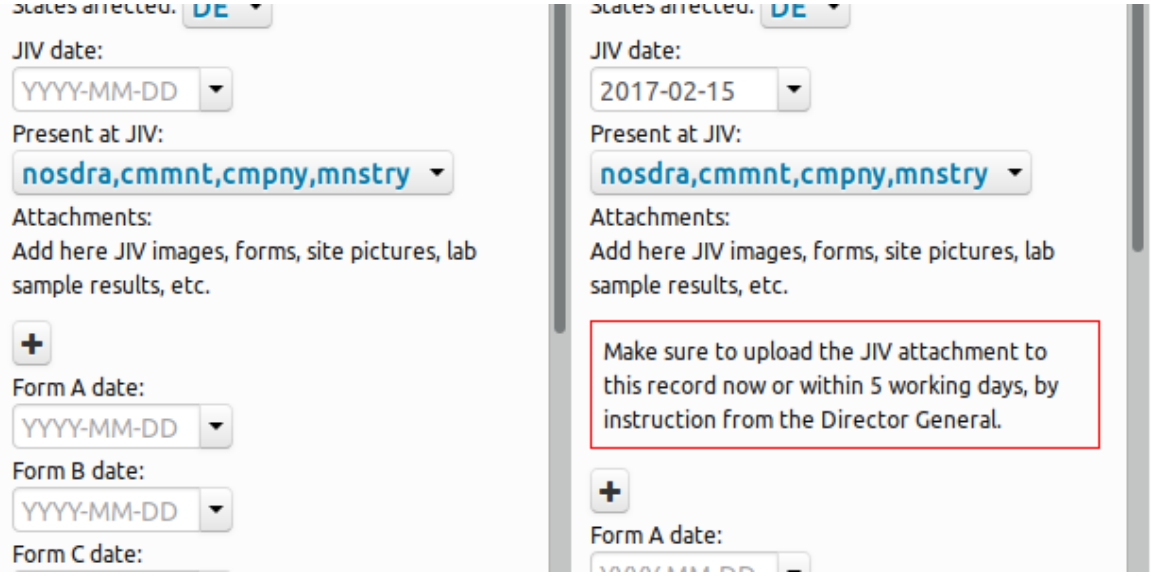


Figure 3.7: Entering a date for the JIV brings up the note until a file is attached.

3.3 Dates

All dates in the system are stored in ISO 8601 format (e.g. `2019-03-06` for 6th of March of 2019), although they might be formatted in other ways for display. Entering dates in an incident report must be done in that format to avoid ambiguity.

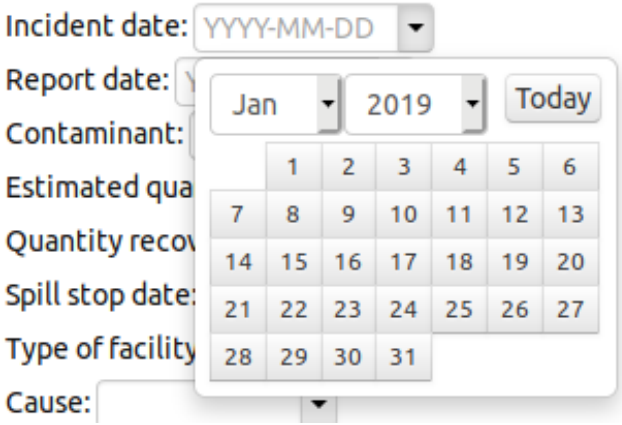


Figure 3.8: Dates can be typed directly for speed, or using the date picker. The grey YYYY-MM-DD hint shows the expected format (four year digits, a hyphen, two month digits, another hypen, and two day digits) and disappears as soon as something is written there. Clicking Today will immediately enter today's date.

3.4 Amounts

These fields are stored in fixed units, but there is a converter accessible by clicking on the button labelled with the required units.

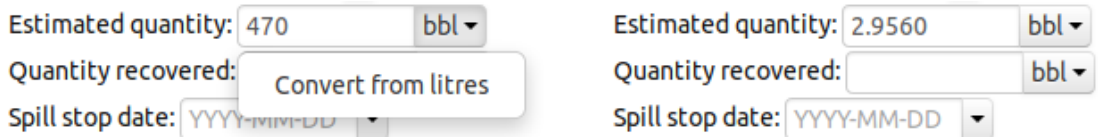


Figure 3.9: Oil amounts need to be entered in barrels (bbl), but clicking on the bbl button brings up the option to convert an amount entered in litres to barrels. Here, 470 litres are 2.956 barrels.

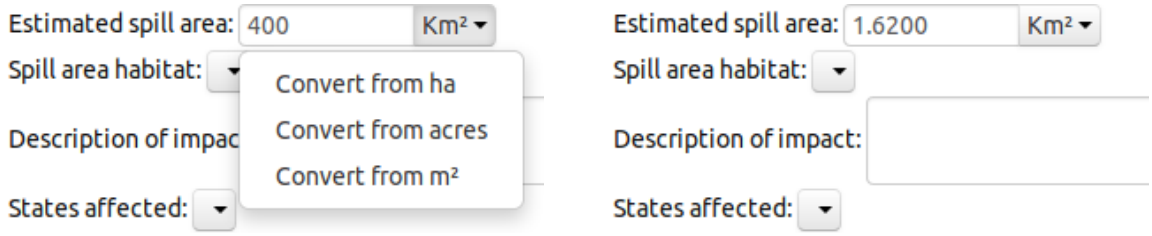
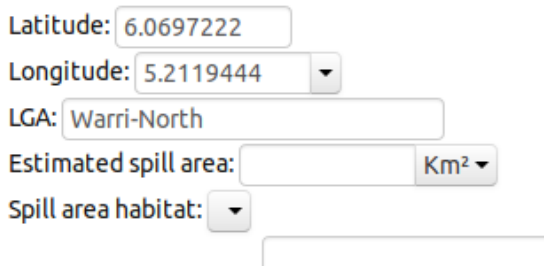


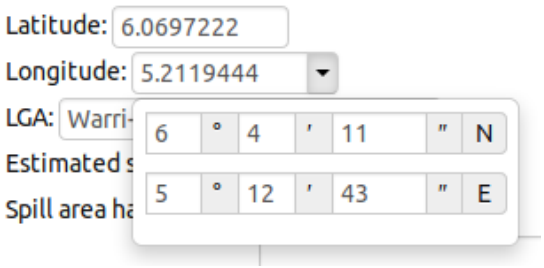
Figure 3.10: Areas must be entered in Km^2, but you can enter an amount in either hectares, acres, or square meters, and have it converted. For instance, 400 acres correspond to 1.62 Km^2.

3.5 Location

The location coordinates are decimal degrees in EPSG:4326 projection, the same used by the GPS satellite navigation system.



The LGA field is filled-in automatically from the coordinates entered, but you can override it if necessary.



Coordinates must be decimal degrees, but can be converted from DMS format with the drop-down button at the right side of the "Longitude" field.

Initially the application allowed entering coordinates also in eastings/northings form and converted them automatically, but there were small inaccuracies due to the different coordinate systems employed when recording the values so NOSDRA decided to do the conversion using their own GIS tools. See § 6.6 *Geographical coordinates*.

3.6 Attachments

These are scanned JIV and other forms, pictures of the incident site, and any other files that are related to the incident.

Attachments can also be links to external files (another website for instance) but it is better to upload the files if possible to make sure they stay available.

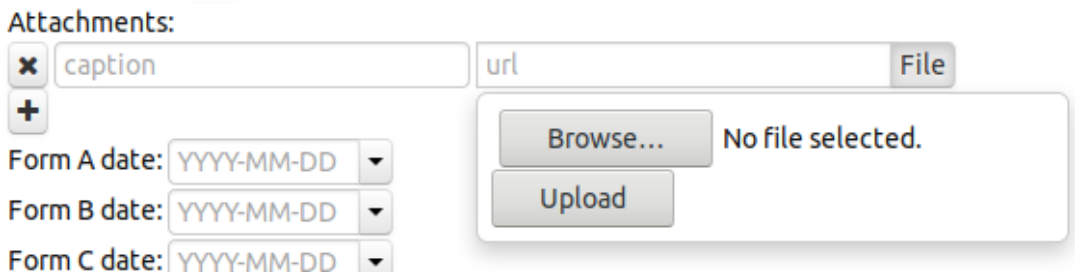
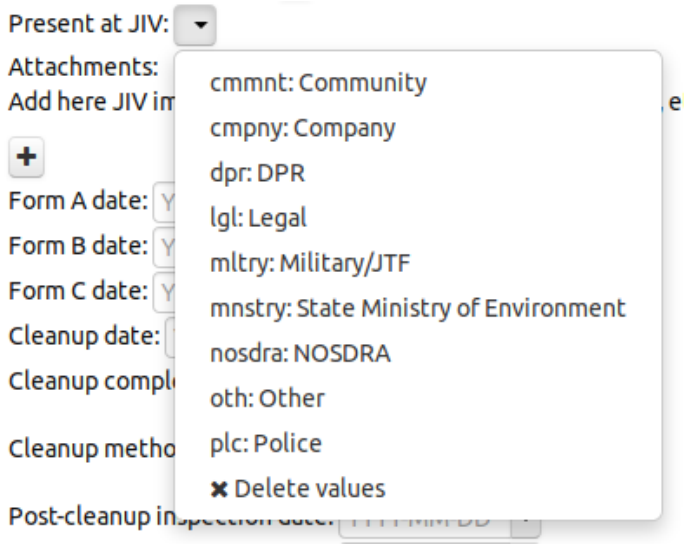


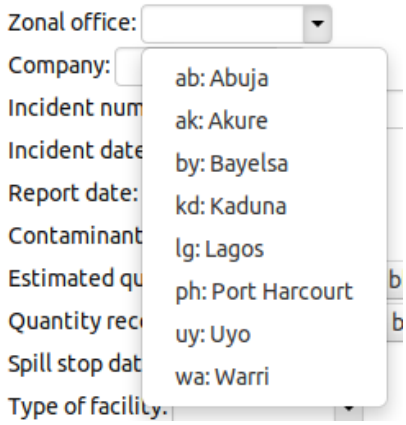
Figure 3.12: Attachments can be added with the + button, and removed with the × button. Each can have a caption and a URL pointing to an external location or, by using the File button, a file can be stored permanently in the system.

3.7 Lists

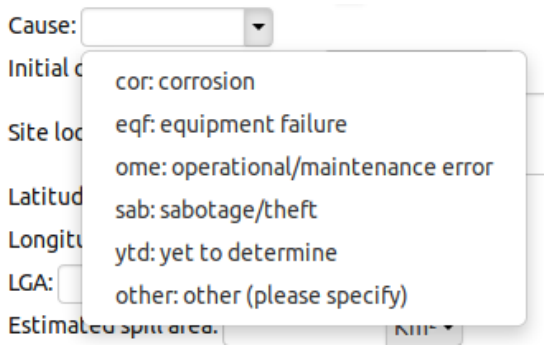
Some fields take a single value, and others can hold a list of values, in which case each time an item from the list is selected, it gets added to the list. The option × Delete values clears it.



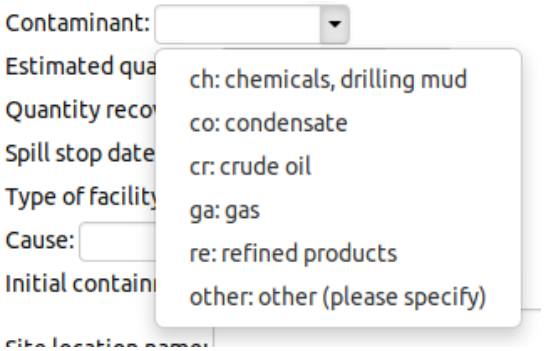
Representatives present at the Joint Investigation Visit for this incident.



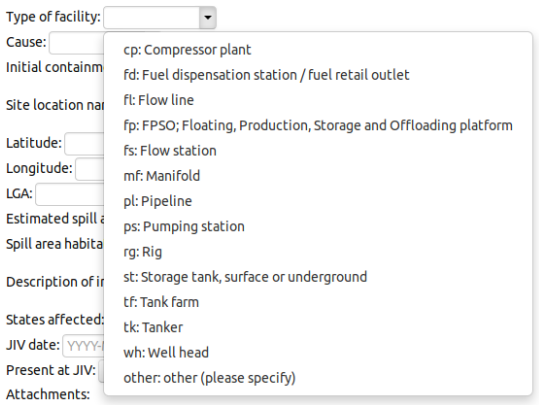
NOSDRA's zonal office responsible for recording the incident.



Cause of the spill.

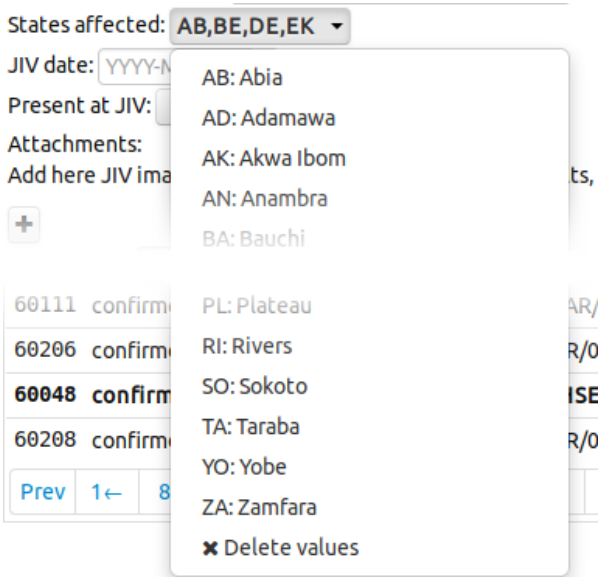


Contaminant substance spilled.

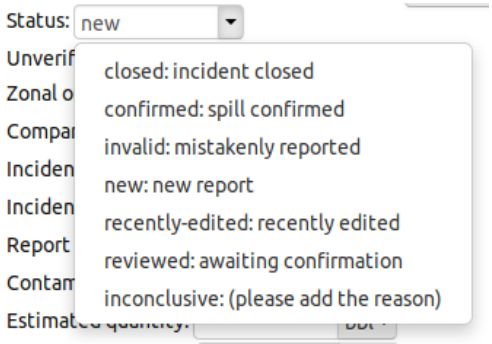


Type of facility where the incident happened.

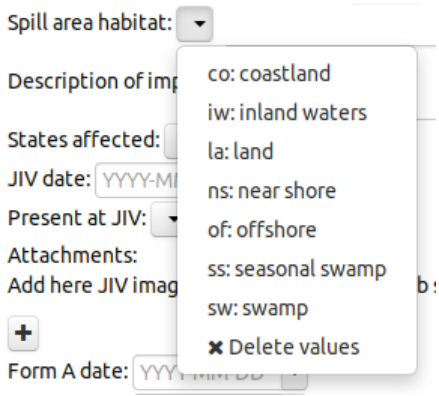
Company that owns the infrastructure where the spill occurred. At the time of writing there are 50 companies known to the application.



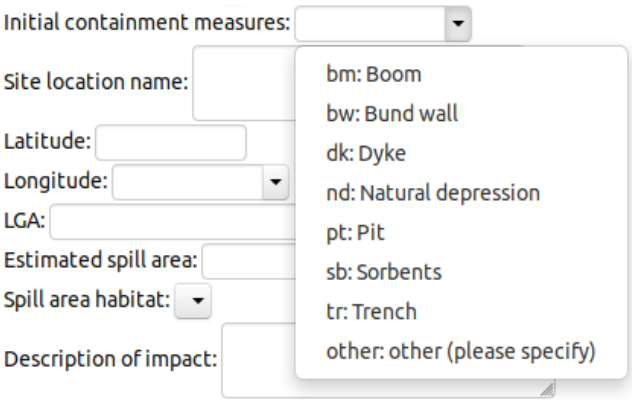
States affected.



Current status of the report, updated as the investigation progresses.



Habitat affected by the spill.



Containment measures deployed when discovered.

3.8 Mobile-friendly data entry

This interface is available through the following URL:
`https://oilspillmonitor.ng/m`

That is the usual URL with "`/m`" appended, which naturally stands for "mobile" and is meant to be easy to type even in a constrained on-screen keyboard.

This variant avoids loading the map, but it still needs to load the incident and LGA data, so when planning to use it at a remote location it is best to pre-load it while broadband is available.

NOSDRA personnel have told me that they prefer using this interface for data entry even on desktop computers.

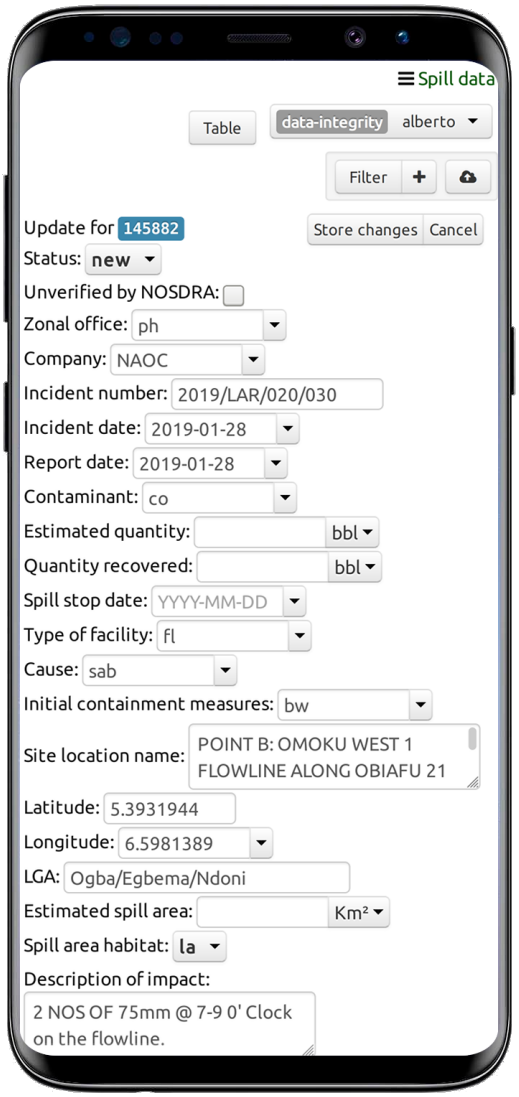


Figure 3.18: The OSM mobile data entry user interface.

User administration

This page is only available for accounts of type `user-admin` and, as observer only (can not modify user accounts), `management`.

The top part, above the user list, has the button to add a new user, the input field for filtering the list and showing only some users, and the button that simplifies writing to all users at once.

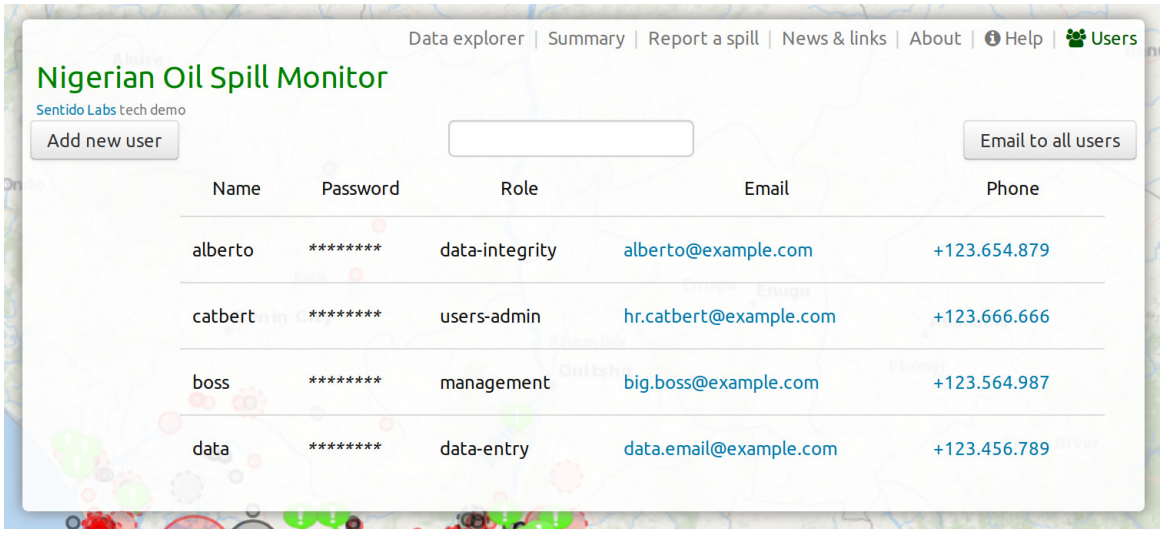


Figure 4.1: List of users in the system. The Email entries are `mailto:` links that will launch your e-mail program, and the phone numbers are `tel:` links to make direct phone calls without having to copy-&-paste the number. Recently modified user accounts are listed first.

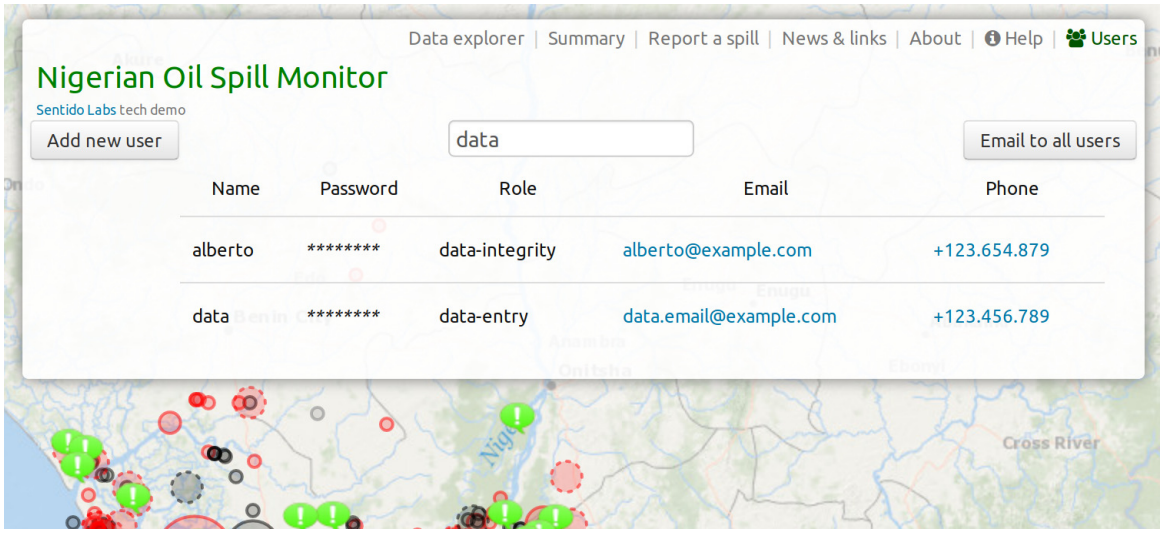


Figure 4.2: Filter the list of users by typing on the input box at the middle top. In this case we are interested only in users whose description (**name**, **role**, **e-mail** or **phone**) contains the text "**data**".

4.1 Modify or delete user accounts

To **modify** or **delete** a user account, click on the name, then either apply the changes with Ok, or delete the account with the red Delete button at the left side.

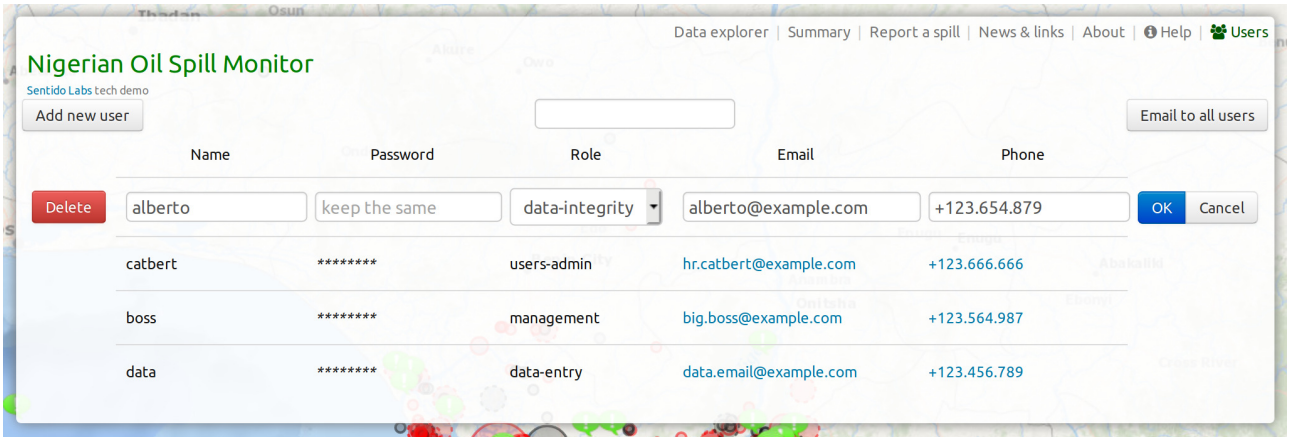


Figure 4.3: Adding, modifying or deleting a user account.

4.2 User account roles

`management` and `civil-society` roles can monitor the latest data but are not responsible for entering or correcting it. Thus, a user logged in as `management` can see what is being done without having to worry about accidentaly destroying records or user accounts.

Role	Edit reports	Change report status to	View user accounts	Modify user accounts
`management`			Yes	
`civil-society`				
`users-admin`			Yes	Yes
`data-integrity`	Yes	new, reviewed, confirmed, invalid, inconclusive		
`data-entry`	Yes	new, reviewed		

In addition, `management` and `data-integrity` accounts have access to additional report tools such as the § 2.5 *Workflow view* and § 2.6 *Summary page*.

There are other roles defined in the system that are not currently used: `civil-society` and `company`.

- `civil-society` is an *observer* role that allows inspecting the latest reports including unconfirmed reports that are normally hidden from the public until confirmed.
- `company` was a role meant for companies to enter their own reports in a separate database, from which they could be transferred to the main database. It worked but was finally not put to use.

One account for each role

There is no overarching user role with access to all areas, and that is a conscious decision.

Some time ago I worked on a system where I added such a user role for easier testing: I could use it to quickly test all the application's functions without having to sign out and in again as a different user.

I called the role `sysadmin`, in the tradition of computer system administrator accounts that have full access to the operative system. In my mind it was like a janitor who has all the keys in the building because he needs to clean and do minor repairs anywhere.

However, it was perceived as a privilege.

I made such an account for the contact person at the customer so that they too could comfortably test all parts of the application as it was being developed. Soon after, they changed the role of one of the final users to `sysadmin` as a *reward*. That particular person was absolutely trustworthy but I saw where this was going: word would get out of this *privileged* `sysadmin` role, and everyone would want it, making it into a political currency.

This is how corruption can start innocently, with the best of intentions.

I immediately removed that account role from the system, including my own user account, and set everyone in the exact role they needed, with several accounts for those that needed them.

Since then, I limit each user role in the applications I build to specific tasks. People that play several roles are better served by having several accounts, which allows the application to customize the user interface for each task. At the end it is better for everyone if you refuse the temptation of system-administrator-type accounts from day one when developing your multi-user application.

Chromebook version

There is an offline-capable installable version for Chromebooks and other computers running ChromeOS.

Figure 5.1: The ChromeOS version prototype running on my Samsung Chromebook 3, in October 2013.

In late September 2013, SDN asked me for advice on how to lock down Windows in the laptops they were planning to acquire for NOSDRA, to ensure they would be used only as Oil Spill Monitor workstations.

My proposal was to use Chromebooks instead, which are laptops built to run Google's ChromeOS. The specific model was the Samsung Series 3 Chromebook XE303C:

- excellent build quality for the price.
- very light and thin, long battery life and small power brick: excellent for on-site work.
- passive cooling: no fan sucking in dust, thus maintenance-free.
- bright and matt screen: narrow field of view is bad for watching movies but fine for one person working directly in front of it.
- good keyboard and trackpad.
- its ARM CPU makes it impossible to re-purpose it for Windows, deterring theft.
- ChromeOS has a robust security model, which makes the software-side also maintenance-free.

In conclusion, it is perfect for the requirements. I went to test it at a local shop, and the performance of the application on it was excellent so I bought their display item for development.

Figure 5.2: In May 2014, SDN delivered a handful of Chromebooks with the ChromeOS version of the application pre-installed. Here they are shown being used during the training sessions in Port Harcourt.

It had also the capability of using map tiles stored on it. I prepared a set of tiles from OpenStreetMap for the whole country of Nigeria, but it turned out not to be worth the trouble.

Over time, as NOSDRA personnel got more modern laptops those Chromebooks have become less necessary, and as mobile Internet access has improved, the advantage of having the application pre-loaded has also lost importance. As of 2019 everyone uses the normal web application, and the installable ChromeOS version has been abandoned. The program that builds the package from the web application is still there though; it could be updated and made to work again if desired.

Server side

6.1 HTTP API

The JSON data can be retrieved with this URL:

`https://oilspillmonitor.ng/api/spill-data.php?dataset=nosdra&format=json`


```
[
    {"id":"2", "status":"confirmed", "company":"ADDAX",
     "incidentnumber":"HSE\/OBO\/0611\/101",
     "incidentdate":"2006-11-23",
     "contaminant":"cr", "estimatedquantity":"225",
     "sitelocationname": "Subsea Pipeline @ Mimbo Platform (OML123)",
     "spillareahabitat": "of",
     "lastupdatedby": "NOSDRA"
    },
    {"id":"3", "status":"confirmed", "company":"ADDAX",
     "incidentnumber":"HSE\/OBO\/0612\/108",
     "incidentdate":"2006-12-18",
     "contaminant":"cr", "estimatedquantity":"0.5",
     "cause":"eqf",
     "sitelocationname":"EBNE04HT @ Ebughu Platform (OML 123)",
     "spillareahabitat":"of", "lastupdatedby":"NOSDRA"
    },
    {"id":"4", "status":"confirmed", "company":"ADDAX",
     "incidentnumber":"HSE\/OBO\/0612\/110",
     "incidentdate":"2006-12-27",
     "contaminant":"cr", "estimatedquantity":"25",
     "sitelocationname":"South East of BOGI Platform",
     "spillareahabitat":"of",
     "lastupdatedby":"NOSDRA"
    },
    ...
]
```


Figure 6.1: A few reports formatted as JSON data.

The server response includes these CORS (Cross-Origin Request Sharing) headers specifically to allow any other front-end to fetch the data.

`Access-Control-Allow-Headers:`	`X-Requested-With, Content-type`
`Access-Control-Allow-Methods:`	`OPTIONS, GET, POST`
`Access-Control-Allow-Origin:`	`*`
`Access-Control-Max-Age:`	`6000`

6.2 Database schema

You can obtain the current database schema (in a bespoke JSON format) from this HTTP endpoint:
`https://oilspillmonitor.ng/api/schema.php?dataset=nosdra`

Dates are in `YYYY-MM-DD` format. For instance, the 11th of April of 2013 is written as `2013-04-11`. If the time is included, it is in `HH:MM:SS` format, after the date separated by a space as in `2013-04-11 11:42:27`. The time zone is WAT (West Africa Time[1], UTC+01).

Latitude and longitude values are in decimal degrees[2], on the WGS[3] 84 datum used by the Global Positioning System GPS[4].

In the following list, fields from NOSDRA's MS Access application (§ 1.2 *How it happened*) are referenced as `ACCDB:FIELD` where `FIELD` is the field name. For instance, the "`DATE OF INCIDENT`" field is written `ACCDB:DATE OF INCIDENT`.

- **`id`**: is the incident identifier (`ACCDB:ID`) assigned automatically by the system, which is a form of serial number known as an auto-increment key. The published spill identifier is composed from that number and, if an incident number is available, a dot "." and the `ACCDB:INCIDENT NO`.
  For example, the incident with serial number `3351` and incident number `2012/lar/015/021` has the spill identifier `3351.2012/lar/015/021`.
- **`updatefor`**: if this report is an update for a previous one, this field contains the `id` of that report.
- **`status`**:
  - **`unverified`**: new or modified report, awaiting integrity check
  - **`investigation`**: under investigation
  - **`verified`**: verified and merged with main database
  - **`unreachable`**: it was not possible to reach the site to verify
- **`reportdate`**: date when the incident happened or, if unknown, when it was first noticed. From `ACCDB:DATE OF INCIDENT`.
- **`company`**: the abbreviated company name, same as `ACCDB:COMPANY`
  - **`ADDAX`**: Addax Petroleum Development Company Nigeria Limited
  - **`AENR`**: Agip Energy Natural Resources Limited
  - **`CHEVRON`**: Chevron Nigeria Limited
  - **`ESSO`**: Esso Exploration and Production Nigeria Limited
  - **`MPN`**: Mobil Producing Nigeria Unlimited
  - **`NAOC`**: Nigerian Agip Oil Company
  - **`NDPR`**: Niger Delta Petroleum Resources Limited
  - **`NPDC`**: National Petroleum Development Company
  - **`POOCN`**: Pan Ocean Oil Corporation Nigeria Limited
  - **`PPMC`**: Pipelines and Products Marketing Company
  - **`SEEPCO`**: Sterling Exploration and Energy Production Company Limited
  - **`SEPLAT`**: Seplat Petroleum Development Company Limited
  - **`SNEPCO`**: Shell Nigeria Exploration and Production Company Limited
  - **`SPDC`**: Shell Petroleum Development Company
  - **`TOTAL`**: Total Exploration and Production
  - **`WRPC`**: Warri Refining and Petrochemical Company
  - **`PHRC`**: Port Harcourt Refining and Petrochemical Company
  - **`KDRC`**: Kaduna Refining Company
- **`incidentnumber`**: an incident reference assigned by the reporting company. It can be any sequence of printable characters.

[1] West Africa Time: `https://en.wikipedia.org/wiki/West_Africa_Time`
[2] Decimal degrees: `https://en.wikipedia.org/wiki/Decimal_degrees`
[3] World Geodetic System WGS: `https://en.wikipedia.org/wiki/World_Geodetic_System`
[4] GPS: `https://en.wikipedia.org/wiki/Global_Positioning_System`

- **incidentdate**: date when the incident happened or, if unknown, when it was first noticed. From ACCDB:DATE OF INCIDENT.
- **contaminant**: the format is text with the classification from the list below, a colon ":", and free text specifying more details. For instance, a spill of refined automotive gas oil is stored as "re:automotive gas oil".
  The initial data comes from ACCDB:TYPE OF CONTAMINANT.
  - **cr**: crude oil
  - **gs**: gas
  - **co**: condensate
  - **ch**: chemicals, drilling mud
  - **re**: refined products
  - **other**: other substance
- **estimatedquantity**: estimated number of barrels (bbl[5]) spilled so far. From ACCDB:QUANTITY OF SPILL (bbl
- **quantityrecovered**: estimated number of barrels (bbl) recovered so far.
- **datespillstopped**: date when the spill stopped. As long as it continues this field is left blank.
- **typeoffacility**: type of facility that leaked the oil:
  - **fp**: FPSO; Floating, Production, Storage and Offloading platform
  - **wh**: well head
  - **tf**: tank farm
  - **mf**: manifold
  - **st**: storage tank, surface or underground
  - **fd**: fuel dispensation station / fuel retail outlet
  - **ps**: pumping station
  - **rg**: rig
  - **tk**: tanker
  - **pl**: pipeline
  - **fl**: flow line
  - **fs**: flow station
  - **cp**: compressor plant
  - **other**: other
- **cause**: the cause of the spill, same as ACCDB:CAUSE OF SPILL
  - **cor**: corrosion
  - **eqf**: equipment failure
  - **sab**: sabotage / third party interference
  - **ome**: operation/maintenance error (human error)
  - **ytd**: yet to determine
  - **other**: other cause
- **initialcontainmentmeasures**: initial measures taken to contain the spill:
  - **bm**: boom
  - **tr**: trenches
  - **bw**: bund wall
  - **sb**: sorbents
  - **other**: other
- **sitelocationname**: location name such as "10" Kwale/Akri pipeline at Agwe-Etiti", from ACCDB:LOCATION.
- **latitude**: EPSG:4326[6] latitude of the spill origin if known, otherwise the rough centre of the affected area. Either copied from ACCDB:LAT or computed from ACCDB:NORTHINGS/EASTINGS (see § 6.6 *Geographical coordinates*) which is an integer in metres.

[5] Oil barrel, bbl: https://en.wikipedia.org/wiki/Barrel_%28unit%29#Oil_barrel
[6] EPSG:4326 http://www.spatialreference.org/ref/epsg/4326/

- **longitude**: EPSG:4326 longitude of the spill origin if known, otherwise the rough centre of the affected area. Either copied from ACCDB:LON or computed from ACCDB:NORTHINGS/EASTINGS.
- **lga**: name of the LGA where the spill occured, computed from the latitude/longitude fields when importing legacy data
- **estimatedspillarea**: a rough estimation of the affected area, in Km^2
- **spillareahabitat**: the kind of habitat affected, from ACCDB:SPILL AREA
  - **la**: land
  - **ss**: seasonal swamp
  - **sw**: swamp
  - **co**: coastland
  - **iw**: inland waters
  - **ns**: near shore
  - **of**: offshore
- **descriptionofimpact**: textual description of the spill impact
- **statesaffected**: list of states that are affected, not just the one where the spill occurred
- **attachments**: links with captions to photographs, videos, form and other document scans, stored as a JSON string
- **datejiv**: date of the Joint Investigation Report
- **datecleanup**: date when the spill cleanup started
- **datecleanupcompleted**: date when the cleanup ended
- **methodsofcleanup**: textual description of the methods used for cleaning up the spill
- **dateofpostcleanupinspection**: date when the spill site was inspected after cleanup
- **dateofpostimpactassessment**:
- **furtherremediation**: further measures taken to remediate the damage
- **datecertificate**: from ACCDB:CERT_DATE

6.3 Data storage

The incident reports are stored in a bespoke append-only database. This allows tracking all changes, which also record the user account that made the modification. It is used for the "Last updated by" field shown in Figure 6.2. If necessary, the application could be extended to show the data before and after each change because adding a new version of a record does not delete the old one.

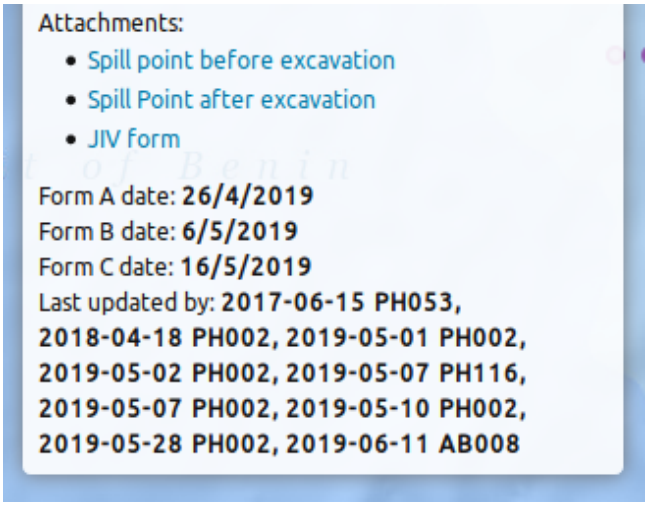


Figure 6.2: The history of changes of a record is visible in the Detail View, at the bottom, under the "Last updated by" label. Each entry, for instance "2017-06-15 PH053", contains the date of the modification and the name of the user account that did it. NOSDRA assigns coded account names to preserve the privacy of its personnel. An account beginning with "PH" belongs to their Port Harcourt office, and "AB" means it was entered by someone at their Abuja office.

In 2013 it was not clear whether the project would get the funds to get finished, continue after the first version, or be maintained afterwards. Therefore it was crucial to keep the data in a form that could be immediately used by NOSDRA on other applications without assistance, and the best data format for that was CSV: it is obvious how to extract it from the server (just copy the single file `"nosdra.csv"`), and if nothing else it can simply be loaded in a spreadsheet program like Microsoft Excel, or back into the MS Access database they already had.

Security

Since the database component does not need to delete entries, only add new ones that might override previous ones for updates, the HTTP API simply does not have the capability of deleting data.

Thus, an attacker would be unable to destroy the records no matter which user account they compromise.

There is still a way of destroying the database: access to the machine through the hosting provider. This can be mitigated by regular off-site back-ups.

Consistency checks

The append-only nature of the database storage means that comparing two copies of the data taken at different points is very simple: the older file must be smaller, and it must match exactly the beginning of the bigger file. Otherwise, at least one of those files has been modified outside of the application's control, either by tampering or by unintentional data degradation.

6.4 HTTP API sequence diagram

The server tries to keep latency low by storing the reports as quickly as possible: the new records are given their permanent identifier, formatted as CSV rows and appended in one go to the database file, and sent back to the web browser. After the data is safely stored, the browser compiles some performance metrics about how long it took from its side, and reports that telemetry to the server. The telemetry payload is secondary, and as such the server handles it differently: it first closes the connection without wasting bandwith sending a response, and then stores it without further processing to minimize its impact on the server's performance.

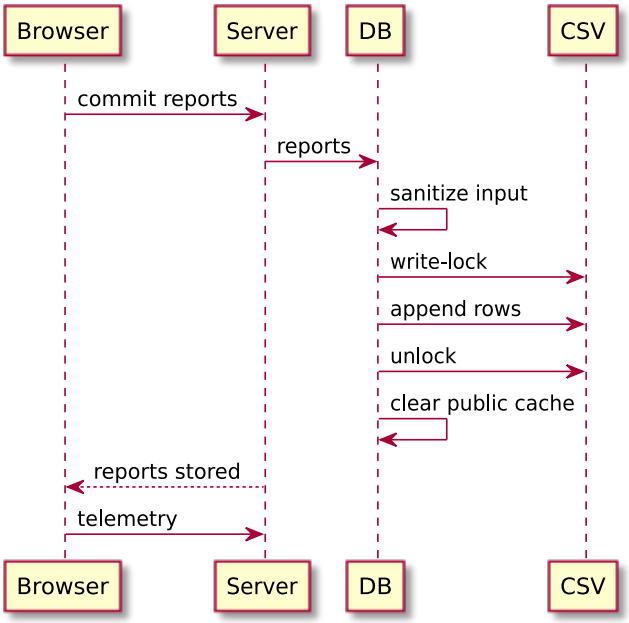


Figure 6.3: Commit sequence. *Telemetry* contains measurements of how long the request took.

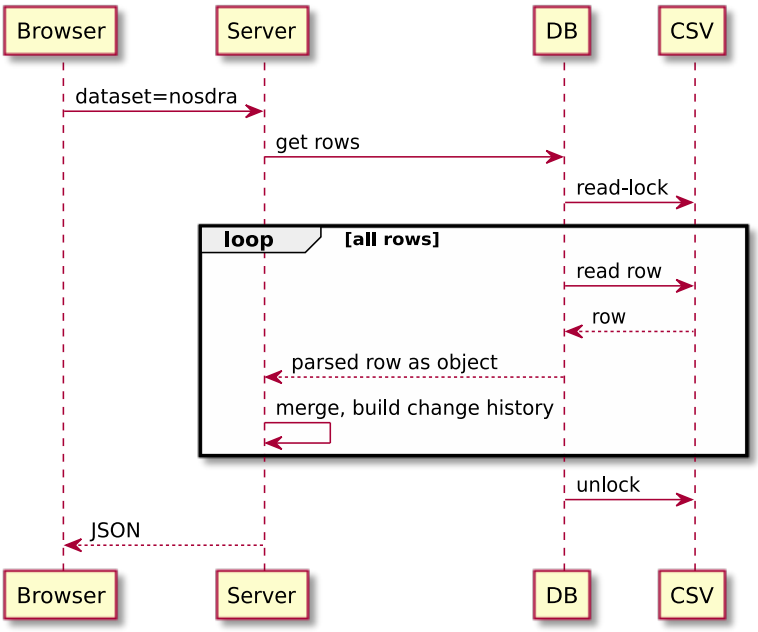


Figure 6.4: Load sequence for authenticated users. Others get the cached JSON content directly.

Reading the data is handled differently depending on whether the user is authenticated (signed-in) or not. Authenticated users get a current view of the database directly from the master file "nosdra.csv", merged and formatted as JSON, while others get sent the contents of the JSON-encoded cache file to

keep the server load to a minimum: in this case all data sent consists of static files. If the cache file does not exist, it is rebuilt following the same procedure as in Figure 6.4.

The public cache file is deleted whenever data is written to the database, so that a fresh updated version will be made when the next unauthenticated request arrives. This avoids wasting resources when there are more authenticated requests than public requests, as the cache would be invalidated and rebuilt before being used.

6.5 Attachments

Attached files such as scanned forms, damage pictures, etc. are not stored in the main database file, but as unmodified individual files under `"data/attachments/"`.

When uploading a new attachment it could conflict with another already there that belongs to another report, for instance `"jiv-form.pdf"`. The simplest way to deal with it would be to assign an auto-generated unique file name such as "8892754921919987 3.pdf", but it many cases the original file name helps managing the files once downloaded so the application preserves it by first making a subdirectory with the report identifier as name, then putting the file there.

Therefore a file called `"jiv-form.pdf"` for report `283` will be stored as `"283/jiv-form.pdf"`, while the same file for report `731` will be at `"731/jiv-form.pdf"`.

If the attachments are sent while creating a new report, that report does not have a permanent identifier yet, so the path will be something like `"t1821/jiv-form.pdf"`, and files attached later will use the permanent identifier once assigned by the server. The consequence is that the attachments for a report might be distributed among different directories, but those will never conflict with any other report.

6.6 Importing data from the original MS Access database

The initial import of data into the OSM required some adaptations that are briefly described here.

Automated data repair

The schema transformation function has several heuristic rules for recovering some information that was misplaced in the `LOCATION` field. The function `extract_fields_from_location(row)` applies regular expression pattern matching to identify and extract those misplaced data. It can recover latitude/longitude coordinates and spill area information.

The results when importing NOSDRA's database from 2006–2012 are summarized here:

```
Pre-processing LGA data
Finished pre-processing data for 774 LGAs
Processing input data from /tmp/uploadNZfQ3g
615: found misplaced SPILL AREA in LOCATION: [Of]
...
805: found misplaced coordinates in LOCATION: 04°23.070'N, 008°04.193'E
805: found misplaced SPILL AREA in LOCATION: [Of]
806: found misplaced coordinates in LOCATION: 04°22.6N, 007°56.7E
...
1291: found misplaced SPILL AREA in LOCATION: [OF]
1292: found misplaced SPILL AREA in LOCATION: [Oth]
1293: found misplaced coordinates in LOCATION: E007° 40.7' N04° 20.74'
1293: found misplaced SPILL AREA in LOCATION: [Off]
1294: found misplaced coordinates in LOCATION: E007°31.3' N04°01.8'
1294: found misplaced SPILL AREA in LOCATION: [Of]
1295: found misplaced coordinates in LOCATION: E007°29' 29.6025 N04°00'55.387
...
1381: found misplaced coordinates in LOCATION: N4°20.74'E7°40.7'
1381: found misplaced SPILL AREA in LOCATION: [off]
1382: found misplaced coordinates in LOCATION: 007°40.7'E,04°20.74'N
...
2748: found misplaced coordinates in LOCATION: N05°29'35.6" E06°43'09.9"
...
2807: found misplaced coordinates in LOCATION: N04°21'45.11" E06°18'40.02"
2808: found misplaced coordinates in LOCATION: N05°01'13.2" E06°32'08.2"
Coordinates already entered, so not modified: N 5° 1' 13.02″, E 6° 32' 8.02″
2808: found misplaced SPILL AREA in LOCATION: [La]
SPILL AREA already entered, so not modified: LA
2809: found misplaced SPILL AREA in LOCATION: [Sw]
...
4740: found misplaced SPILL AREA in LOCATION: [La]
Processed 4742 rows
Seconds for pointInPoly(): 6.68041658401
Seconds for transformTo(): 0.364021539688
Total time in seconds: 9.26057887077
Company: PPMC
  processed 42 reports
  of which  32 reports lack location data
Company: AENR
  processed 1 reports
  of which  1 reports lack location data
Company: MPN
  processed 947 reports
```

```
  of which  781 reports lack location data
...
Company: ESSO
  processed 26 reports
  of which  24 reports lack location data
Written full data into /var/www/data/nosdra.csv
extracted 196 coordinates from LOCATION field
extracted 2856 habitat/area values from LOCATION field
OK
```

As reported there, in 196 cases the coordinates were successfully recovered which means that those reports can now be seen in the map, and in 2856 cases the habitat/area information (land, swamp, offshore, etc.) too was recovered.

That does not count the cases where the coordinates or area information was both in the proper field and as part of the `LOCATION` field: in those cases the data is not modified as reported above with "... already entered, so not modified".

Another common issue was that the coordinates were swapped: the northing was put first (where the easting should be), and the easting second, probably because of habit from the more common latitude/longitude notation.

Data normalization

Imported data was automatically normalized according to the rules below. The values found in the database are listed separated by a slash "/", and the normalized value is at the right side of the arrow.

- **`SPILL AREA`**:
  [La]/(La)/[la]/La]/[La]/[LA]/[La}/[L]/[LN]/[ln]/[LND] → la
  (not found in existing database) → ss
  [Sw]/(Sw)/[SW]/[SWP]/(swamp)/{sw}/(SW) → sw
  COASTLINE → co
  IW/fresh water → iw
  [Near shore]/[Near Shore]/[Ns] → ns
  [Of]/[Off]/[OF]/[OFF]/(OFF)/(Off)/[Of]/[of] → of
  [Ot]/[Oth]/[oth] → other
  LA → la
  SW → sw
  OF/OFF → of
  land swamp/LAND SWAMP → la sw
  OTH → other
- **`CAUSE OF SPILL`**:
  cor/Cor/COR → cor
  Eqf/EQF → eqf
  Ome/OME → ome
  oth, Oth, OTH → other
  OTH(SAND EROSION) → other
  sab/Sab/SAB → sab
  ytd/Ytd/YTD → ytd
- **`TYPE OF CONTAMINANT`**:
  Cr/CR/cr → cr
  Ch/CH/ch → ch
  gas/GAS/Gas → gss
  Re/RE → re
  Con/CON/con → ?
  oth/Oth/OTH → other
  others(automotive gas oil) → re:automotive gas oil

- **STATES**:
  A B I A → Abia
  A K W A - I B O M → Akwa Ibom
  B A Y E L S A → Bayelsa
  D E L T A → Delta
  E D O → Edo
  I M O → Imo
  K O G I → Kogi
  L A G O S → Lagos
  R I V E R S → Rivers
  RI V E R S → Rivers

Geographical coordinates

All latitude and longitude values are translated to decimal degrees and the eastings and northings coordinates are converted to decimal latitude and longitude positions.

Easting and northing are eastward/northward distances in metres from a reference point.

- https://en.wikipedia.org/wiki/Easting_and_northing
- "Determining Correct Locations for Fuels Projects and Fires" from the US National Interagency Fire Center
  http://gacc.nifc.gov/rmcc/dispatch_centers/r2crc/dispatch/dispatch/How To Locate a Fire - Get Coordinates.pdf
- "More details about the UTM coordinate system" from MapTools
  http://www.maptools.com/UsingUTM/UTMdetails.html

They are assumed to be in EPSG:36392[7] which is a Transverse Mercator projection that uses the Minna datum on the Clarke 1880 (RGS) spheroid.

For latitude and longitude we use the `EPSG:4326`[8] projection to match commonly available web map layers. This works specially well in the case of Nigeria compared to Europe becuse of its closeness to the Equator, which makes for a small projection distortion.

[7] EPSG:26392 http://www.spatialreference.org/ref/epsg/26392/
[8] EPSG:4326 http://www.spatialreference.org/ref/epsg/4326/

Maintenance

7.1 Backup copies

There are two things you need to copy regularly:

- "`data/nosdra.csv`" is the master data file.
- "`data/attachments/`" contains all the attachments such as photographs and scanned JIV forms.

7.2 Modifying the application

The application is built from its components by the "`build.php`" program. Whenever you **change** something, you have to **run it again** to ensure that those changes are applied. It can be run from the command line in the server, or better from the following URL by signed-in users:
`https://oilspillmonitor.ng/build.php`

```
Started 2019-03-12 12:57:44+00:00

js/index.0.9.2.892.gz.js built from
  base.js
  Nigeria.js
  SpillData.js
  DetailView.js
  MapSources.js
  MapView.js
  TableView.js
  DatePicker.js
  OilSpillMonitor.js
  GraphView.js
  WorkflowView.js
  UsersView.js

css/index.0.9.2.892.gz.css built from
  index.css

js/vendor/combined.0.9.2.892.gz.js built from
  vendor/angular.min.js
  vendor/ui-bootstrap-tpls-0.8.0.min.js
  vendor/leaflet.js
  vendor/leaflet.wikimapia.js
  vendor/L.Control.Zoomslider.LICENSE
  vendor/L.Control.Zoomslider.js
  vendor/shramov/layer/tile/Google.js
  vendor/topojson.js
  vendor/proj4js-compressed.js
  vendor/Array.toNode.js
  vendor/d3.v3.LICENSE
  vendor/d3.v3.min.js
  vendor/leaflet-pip.LICENSE
  vendor/leaflet-pip.min.js

css/vendor/combined.0.9.2.892.gz.css built from
  vendor/bootstrap.min.css
  vendor/font-awesome.min.css
  vendor/leaflet.css
  vendor/L.Control.Zoomslider.css

topojson/ng-states.gz.json built from
  topojson/ng-states.json

topojson/oil-blocks-nosdra.gz.json built from
  topojson/oil-blocks-nosdra.json

geojson/viirs-ir-sources-ng.atm_cor.gz.geojson built
from
  geojson/viirs-ir-sources-20140107.atm_cor.geojson

topojson/wwf_terr_ecos.gz.json built from
  topojson/wwf_terr_ecos.json

topojson/nigeria-soil.gz.json built from
  topojson/nigeria-soil.json

geojson/pipelines.gz.geojson built from
  geojson/pipelines.geojson

geojson/gas-pipelines.gz.geojson built from
  geojson/gas-pipelines.geojson

geojson/gas-power-plants-v1.gz.json built from
  geojson/gas-power-plants-v1.json

geojson/onshore-oil-wells-v1.gz.json built from
  geojson/onshore-oil-wells-v1.json

index.gz.html built from
  index.html

m.gz.html built from
  m.html

news.gz.html built from
  news.html

about.gz.html built from
  about.html

summary.gz.html built from
  summary.html

workflow.gz.html built from
  workflow.html

report.gz.html built from
  report.html

help.gz.html built from
  help.html

admin.gz.html built from
  admin.html

manifest.json built from
  ../chrome/manifest.json

chrome/updates.xml built from
  ../chrome/updates.xml

Finished 2019-03-12 12:57:46+00:00
```

Figure 7.1: Output of the "`build.php`" program, listing which files were built from what components.

This build mechanism includes a simple templating method based on PHP, shown in Figure 7.3.

```
<?php
...

$files = array(
'js/index.$version.gz.js'
 => array('base.js',
          'Nigeria.js',
          'SpillData.js',
          'DetailView.js',
          'MapSources.js',
          'MapView.js',
          'TableView.js',
          'DatePicker.js',
          'OilSpillMonitor.js',
          'GraphView.js',
          'WorkflowView.js',
          'UsersView.js'
),
...
);

...
?>
```

Figure 7.2: In "`build.php`", the construction table describes how to build each file from its parts, for instance "`js/index.`*`version`*`.gz.js`" which is built from "`base.js`", "`Nigeria.js`" etc. Each of those files can contain PHP statements as shown in Figure 7.3.

```
    <div id="content-pages-overlay"></div>

<?php require 'app.detail-view.html' ?>
<?php require 'app.workflow-view.html' ?>
<?php require 'app.table-view.html' ?>

  </div>

  <script src="js/vendor/combined.<?= $version ?>.gz.js"></script>
  <script src="js/index.<?= $version ?>.gz.js"></script>
```

Figure 7.3: A fragment from "`components/html/index.html`". `<?php require '...' ?>` processes a template file and inserts the result at that point, and `<?= $version ?>` is replaced by the value of the PHP variable `$version`. In this case we insert it in the file names to make sure the web browser loads the current version of those files, not old cached copies.

7.3 Reordering fields in the detail view

Set the constant `preferredOrder` in "`components/DetailView.js`". It is a list with the field names.

```
const preferredOrder = [
    "id",
    "updatefor",
    "status",
    "notverifiedbynosdra",
    "zonaloffice",
...
    "descriptionofimpact",
    "statesaffected",
    "jivdate",
    "jivpresent",
    "attachments",
    "formadate",
    "formbdate",
    "formcdate",
    "cleanupdate",
...
    "certificatedate",
    "certificatenumber",
    "lastupdatedby"
];
```

Figure 7.4: Abbreviated content of the `preferredOrder` list, with `jivdate` and `jivpresent` moved above `attachments`.

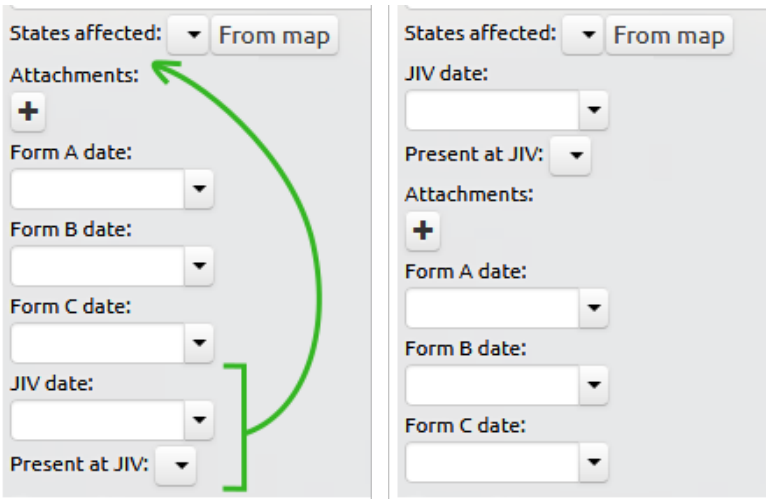


Figure 7.5: The fields *JIV* date and *Present at JIV* were relocated because NOSDRA found that many users forget to fill these the moment they encounter Form A, B, and C.

7.4 Adding field values to the database schema

The database schema is stored in the file "`api/schemas/nosdra-v1.1.json`". It contains the list of fields, and each field has several properties such as `name` (field name in the database), `label` (for display to users), `group` (for the detail view), and `type`.

There are some pre-defined types, and a customizable set type:

```
{"name": "incidentdate",
 "label":"Incident date",
 "group":"incident",
 "type":"date"}
```

Figure 7.6: The schema definition for the `incidentdate` field.

- `id`: valid database identifier.
- `string`: short text.
- `text`: plain text.
- `text.Nigeria.lgas`: this field allows free text entry, but offers the list of Nigeria's LGAs as options.
- `date`: (§ 3.3 *Dates*) date in ISO 8601 format.
- `number.litre`: (§ 3.4 *Amounts*) amount in litres.
- `number.lat`: (§ 3.5 *Location*) latitude as a decimal number.
- `number.lon`: (§ 3.5 *Location*) longitude as a decimal number.
- `number.km2`: (§ 3.4 *Amounts*) area in Km^2.
- `json`: the value is a JSON string, for instance the attachments list (§ 3.6 *Attachments*) which is a list of objects with "`url`" and "`caption`" properties.
- `{"value":"label",...}`: (§ 3.7 *Lists*) this customizable type is a set of values, from which the user picks one, or more if the field property "`multivalue`" is present.

7.5 Oil companies

This is a special case because it is not defined completely in the schema file, but in the file "`components/js/Nigeria.js`", that contains the application's knowledge about Nigeria.

Add and entry with the company name and, if known, its website URL.

```
oilCompanies: {
    'nyd': {
        'name': 'Not yet determined'
    },
    'ADDAX': {
        'name': 'Addax Petroleum Development Company Nigeria Limited',
        'url': 'http://www.addaxpetroleum.com/'
    },
    'AENR': {
        'name': 'Agip Energy Natural Resources Limited',
        'url': 'http://en.wikipedia.org/wiki/Agip'
    },
 ...
    'WALTSMITH': {
        'name': 'Walter Smith Petroman Oil Ltd',
        'url': ''
    },
    'WRPC': {
        'name': 'Warri Refining and Petrochemical Company',
        'url': 'http://www.wrpcnnpcng.com/'
    }
}
```

7.6 Zonal Offices

In the `zonaloffice` field, under `type`, add the code and name.

```
    {"name": "zonaloffice",
     "label":"Zonal office",
     "group":"db",
     "type":{"wa": "Warri",
             "kd": "Kaduna",
             "lg": "Lagos",
             "ph": "Port Harcourt",
             "uy": "Uyo",
             "by": "Bayelsa",
             "ak": "Akure",
             "ab": "Abuja"
             },
     "other":{"type":"text", "prefix":""}
    }
```

To make them appear on the map, add the additional information in the GeoJSON file "`components/js/geojson/nosdra-offices.json`".

7.7 Application sections

Updating NOSDRA's phone numbers

File: "`components/html/footer.html`"

Other pages ("Report a spill" and "News & links") include this file with `<?php include 'footer.html'?>` so that you do not need to copy and paste this content.

"Report a spill"

Comes from the file "`components/html/report.html`"

At the moment, the content of "`report.html`" just duplicates the page footer:

```
<div class="content-page single-column">
<?php include 'footer.html' ?>
</div>
```

"News & links"

File "`components/html/news.html`"

At the bottom it includes the footer content too. This means that you only need to update the phone numbers in the footer file (§ 7.7 *Updating NOSDRA's phone numbers*) to have them appear correctly in those other places.

"About"

The content is in "`components/html/about.html`"

"Help"

File: "`components/html/help.html`"

You will only see he link to this page (after "About") after you sign in.

7.8 Increasing the memory limit

As the database grows, the server needs more RAM to process the requests from signed-in users. The default value is 128 Megabytes.

Adjust it with the `memory_limit` constant in "`php.ini`". This file may be located in different places depending on the machine's configuration.

```
memory_limit=256M
```

At of December 2019, the hosting provider has it configured to a value that is several times bigger than needed by the application, so this will not have to be changed anytime soon.

JIV Form

These are the Joint Investigation Visit forms used at the time the application was designed in 2013. The work of modelling the database schema from this was made by Mohammed Gumsuri (NOSDRA) and Rory Hodgson (SDN).

NATONAL OIL SPILL DETECTION AND RESPONSE AGENCY

JOINT INVESTIGATION VISIT (JIV) FORM

Note: This JIV Form is to be completed and signed by all participating parties in the field

1. **Company**: ……
2. **Type of Complaint/Incident**:
   ☐Oil Pollution ☐ Fire/Explosion ☐Drilling Mud/Chemical Pollution
   ☐Others (Specify)……
3. **Incident Details**
   i. Date of Incident: …… ii. Date first reported:…….
   iii. Date of first investigation:……..
   iv. Date of follow-up investigation:…….
   v. Time investigation started:……..
   vi. Estimated quantity spilled:……...

4. Site Details
   i. Site/Location:…….
   ii. Position of Spill/Leak:……
   iii. **Spill area**
   ☐Land ☐Swamp ☐Freshwater ☐Mangrove ☐Coastline
   ☐Near shore ☐Offshore ☐Others (specify)……
   iv. **Structural Controls in Place**
   ☐Boom ☐Trenches ☐Bund wall ☐Sorbents ☐Others (specify)……..

5. **Circumstances Around Spill Point**
   i. **Visual observation of Hole Position**
   ☐ 12 O' Clock ☐ 10 O' Clock ☐ 2 O' Clock ☐ 3 O' Clock
   ☐ 4 O' Clock ☐ 5 O' Clock ☐ 6 O' Clock
   ii. **Type of Oil contaminant**
   ☐Crude Oil ☐Condensate ☐Chemicals ☐Refined Products
   ☐Others (specify) …….
   iii. **Facility**
   ☐Pipeline ☐Flow line ☐Wellhead ☐Manifold ☐Flow Station ☐Rig
   ☐Storage Tank ☐Compressor Plant ☐Others (specify)…….
   iv. **Cause Of Spill**
   ☐Corrosion ☐Equipment Failure ☐Third Party Interference ☐Accident
   ☐Operational Error ☐Others (specify)……..
   v. **Visible Sign of Third Party Interference**
   ☐Hacksaw Marks ☐Drilled Holes ☐Blasting ☐Theft ☐Acid
   ☐Others (specify)……..

6. Impact of Incident

i **Properties impacted by the incident**

☐Farmland ☐Fish Pond ☐Vegetation ☐Fishing Net ☐Surface water

☐Venerable Objects ☐Others (specify)……..

ii. **Nature of impact**

☐Oil stained vegetation ☐Oil stained fishing nets ☐Dead floating fishes

☐Dead floating crabs ☐Withering of vegetation ☐Others……..……

iii. Delineation of impacted area

☐Within Company's facility/ROW ☐Outside Company's facility/ROW

7. Extent of Impact

a) **Community (***indicate name***)** **b) Land (***indicate area in* m^2 **)**

i. …… i. ……

ii. …… ii. ……..

iii. …… iii…….……..

iv. ……. iv. ……..……

b) **Creeks/Creek lets** ☐Tidal ☐Non-tidal

NAME	DIRECTION OF SLICK	LENGTH OF SLICK	WIDTH OF SLICK

c) **Swamp** ☐Tidal ☐Non-tidal

NAME	DIRECTION OF SLICK	LENGTH OF SLICK	WIDTH OF SLICK

d) **River** ☐Tidal ☐Non-tidal

NAME	DIRECTION OF SLICK	LENGTH OF SLICK	WIDTH OF SLICK

e) **Shoreline/water**

NAME	DIRECTION OF SLICK	LENGTH OF SLICK	WIDTH OF SLICK

8. **Photograph/Map/Chart Ref.**
   ☐Still photographs ☐Video coverage ☐Mapping

9. **Samples taken**…….. …….

10. **Investigation carried out by**
    ☐Foot ☐Boat ☐Aircraft

11. **Remarks/Recommendation**……

    ……

    ……

12. **Time investigation ended**……..

13. **Name and Signature of Participants**

- **NOSDRA**
  1. **2.** **3.**

- **DPR**
  1. **2.** **3.**

- **STATE MINISTRY OF ENVIRONMENT**
  1. **2.** **3.**

- **LOCAL GOVERNMENT COUNCIL**
  1. **2.** **3.**

- **COMPANY**
  1. **2.** **3.**

- **COMMUNITY**
  1. **2.** **3.**

Notification report forms A,B,C

These are the additional forms attached to the JIV Forms used at the time the application was designed in 2013.

National Oil Spill Detection & Response Agency

FORM A

OIL SPILL/LEAK NOTIFICATION REPORT

This report must be submitted within 24 hours of Spill Incidence

1.GENERAL INFORMATION:				
i. Company Name:				
ii. Incident Details:- **iii. Spill Reference No:**	Date of Incidence (dd/mm/yy)	Time of Incidence (24h standard/daylight) hrs to hrs	Date of Observation (dd/mm/yy)	Time of Observation (24h standard/daylight) hrs to hrs
Survey By: Foot/Boat / Helicopter / Overlook /	Sun / Clouds / Fog / Rain / Snow / Windy			
Level of Impact: ☐ No Impact	☐ Slight Impact	☐ Heavy Impact		
Estimated quantity spilled:				
2. Site Details				
i. Site Name: **OML:**				
ii. GPS FIELD POINTS Total Length _____m Length Surveyed_____ m Differential GPS Yes/No Spill Start Point GPS: EASTINGS _____meters NOTHINGS _____meters Spill End Point GPS: EASTINGS _____meters NOTHINGS _____meters				
iii. Site area ☐ Land Swamp ☐ Freshwater ☐ Mangrove ☐ Coastline ☐ Near Shore ☐ Offshore ☐ Others (Specify)……				
iv. Containment Measures in Place ☐ Boom ☐ Trenches ☐ Bund wall ☐ Sorbents ☐ Others (Specify)……				
v. Type of Contaminant ☐ Crude Oil ☐ Condensate ☐ Chemicals ☐ Refined Products ☐ Others (Specify)……				
vi. Facilty ☐ Pipeline ☐ Flow line ☐ Wellhead ☐ Manifold ☐ Flow Station ☐ Rig ☐ Storage Tank ☐ Compressor Plant ☐ Others(Specify)……				
vii. Properties at Risk ☐ Farmland ☐ Fish Pond ☐ Vegetation ☐ Fishing Net ☐ Surface water ☐ Venerable Objects ☐ Others (Specify)……				

3. SURVEY TEAM NO ____ Name	Organization	Phone Numbers

REPORTING OFFICER: ……

DESIGNATION: ……

SIGNATURE: …… **DATE:** ……

**RBA Report must be submitted within 24 Hours of the Spill Incidence.*

National Oil Spill Detection & Response Agency

FORM B

RISK BASED ASSESSMENT OF OIL SPILL INCIDENCE (RBA)

Note: This report must be submitted within 2 weeks of Spill Incidence

1.GENERAL INFORMATION:

i. Company Name:

ii. Date of Assessment:

iii. Incident Details:

Date of Incidence (dd/mm/yy)	Date spill was stopped	Method Used
		☐ Clamping ☐ Well Shut-in ☐ Valve Shut-in ☐ F/Station Shut down ☐ Others (specify).....

iv. Estimated quantity spilled:

v. Estimated quantity recovered:

vi. Cause of Spill

☐ Corrosion ☐ Equipment Failure ☐ Third Party Interference ☐ Accident

☐ Operational Error ☐ Others (specify).....

2. Site Details

i. Site Name: **OML:**

ii. GPS FIELD POINTS Total Length _____m Length Surveyed _____m Differential GPS Yes/No

Spill Start Point GPS: EASTINGS _____meters NOTHINGS _____meters

Spill End Point GPS: EASTINGS _____meters NOTHINGS _____meters

iii. Site area

☐ Land Swamp ☐ Freshwater ☐ Mangrove ☐ Coastline ☐ Near Shore

☐ Offshore ☐ Others (Specify).....

iv. Facilty

☐ Pipeline ☐ Flow line ☐ Wellhead ☐ Manifold ☐ Flow Station

☐ Rig ☐ Storage Tank ☐ Compressor Plant

☐ Others(Specify).....

v. Site Characterization

a. Sea Conditions

☐ Calm ☐ Rough ☐ Not Applicable ☐ Low Tide ☐ High Tide

Current direction: .....

Swell Height: .....

Current Strength: .....

b. Weather Conditions

☐ Bright Sunny ☐ Party Cloudy ☐ Slight rain ☐ Others (Specify).....

Temperature: ..... Wind Direction: .....

Wind Speed: ..... Relative Humidity: .....

vi. Visual Observation of Impacted area

(i)	Any oil sheen on water	Yes	☐	No	☐	N/A ☐
(ii)	Any oil sheen on water	Yes	☐	No	☐	N/A ☐
(iii)	Any oil sheen on water	Yes	☐	No	☐	N/A ☐
(iv)	Any oil sheen on water	Yes	☐	No	☐	N/A ☐
(v)	Any oil sheen on water	Yes	☐	No	☐	N/A ☐

Receptor Assessment

Receptor	Pathway to Impacted Area(m^2)	Distance to Impacted Area(m^2)	Estimated Area of Impact Area(m^2)	Receptor Impacted (Yes/No)	Remarks
Farmland					
Fish Pond					
Vegetation					
Surface Water					
Ground Water					
Venerable Object					
Human Habitation					
Livestock					
Plantation					
Swamp					

vii. Any Casualties ☐ Yes ☐ No

If yes, give details……

……

viii. Clean-up Program details

a. Method of clean-up……

b. Time frame for clean-up……

ix. General Remarks

……

……

……

REPORTING OFFICER: ……

DESIGNATION: ……

SIGNATURE: …… **DATE:** ……

**Clean-up program report must be submitted within 2 weeks of spill incidence.*

National Oil Spill Detection & Response Agency

FORM C

SITE CLEAN-UP/REMEDIATION ASSESSMENT REPORT

1. GENERAL INFORMATION:	
i. **Company Name:**	
ii. **Date of Assessment:**	

2. Site Details

i. Site Name: **OML:**

ii. Date/Time of Incident:

iii. Area and Depth of Impact:

iv. **GPS FIELD POINTS** Total Length _____m Length Surveyed _____m Differential GPS Yes/No
Spill Start Point GPS: EASTINGS _____meters NOTHINGS _____meters
Spill End Point GPS: EASTINGS _____meters NOTHINGS _____meters

v. **Contaminated Media**
☐ Vegetation ☐ Soil ☐ Sediment ☐ Inland Surface Water ☐ Brackish Swamp Surface Water
☐ Off shore Surface Water ☐ Underground Water ☐ Others (Specify)……

3. (i) **Date Clean-up Programme Comenced:**
(ii) **Method of Clean-Up**
☐ Low Pressure Wash ☐ Mannual ☐ Mechanical ☐ Surface Wash ☐ Sorbents ☐ Chemical Disparsant
☐ Vacuum Skimming ☐ Others(Specify)……

(iii) Estimated quantity of oil / containment recovered……

(iv) **Method of Debris Disposal**
☐ Controled Inciniration ☐ Buried in lined pit ☐ Chemical Treatment ☐ Sanitary Landfill
☐ Land farming ☐ Others (Specify)……

4. Site Visual Observation
(i) **Nature of Soil**
☐ Show Heavy Impact ☐ Medium Impact ☐ Minimal Impact ☐ Others……

(ii) **Nature of Surface Water**
☐ Oil Sheen Present ☐ No Oil Sheen Present ☐ Others (Specify)……

(iii) **Nature of Vegetation**
☐ Withered ☐ Withering ☐ Luxuriant
(iv) Site Photos ☐ Yes ☐ No

(v) Date Site clean-up ended……

(vi) Sample collected after the clean-up program
☐ Soil ☐ Sediment ☐ Water ☐ Others (Specify)……

5. Result of Laboratory Analysis of Samples collected Pre/Post Remediation

Parameter	**Sample**	**Test Method**	**Result**	
			Pre-Remd.	Post Remd.
TPH				
BTEX				
Trace Metals			Pre-Remd.	Post Remd.
Arsenic				
Barium				
Cadmium				
Chromium				
Copper				
Mercury				
Lead				
Nickel				
Zinc				
Total Dissolved				
Total suspended Solids				

6. Does Site require remediation ☐ Yes ☐ No

If yes,

(i) Date Site remediation commenced……

(ii) **Method of Remediation**

☐ Land farming ☐ Biopile ☐ Bio venting ☐ Air Sparging ☐ Chemical Oxidation

☐ Washing/Leading ☐ Phyto remediation ☐ Enhanced Natural Attenuation

☐ Monitored Enhanced Natural Attenuation ☐ Thermal Desorption ☐ Others (specify)……

(iii) Is remediation method ☐ in situ or ☐ ex situ?

(iv) Details of remedial method (attached as an annex)

7. Details of rehabilitation plan for impacted population (attached as an annex)

8. Cost of Spill

a. Clean-up cost:-……

b. Clean-up remediation:-……

c. Cost of Repair works: ……

d. Naira loss due to oil Spilled: ……

e. Lost Man Hours: ……

Total……

9. Compensation paid, if any: ……

10. Method of Settlement of Claim

☐ Arbitration/Mediation ☐ Direct negotiation between Landlord and operator

☐ Court Settlement ☐ Not Applicable ☐ Others (specify)……

11. Date/Time of Visit by Regulations: .....

12. Remarks by any Third Party

……

……

……

13. General Remarks

……

……

……

14. NOTE: Officials of NOSDRA must be present when samples are collected, and when analyses begin.

REPORTING OFFICER: .....

DESIGNATION: .....

SIGNATURE: ..... **DATE:** .....