We want to create a worldwide map of POI like the one below:

Map

Description automatically generated

We will use open street map as the earth map generator / API.

We want the same look & feel as the project “tilia.earth” for the pin point where point of interest are aggregated when zooming out and they “split” when we zoom in.

Example in Switzerland:

Map

Description automatically generated

Instead of the following search bar:

A picture containing chart

Description automatically generated

We want to have a dropdown list where we can select a “network”.

The list of available networks comes from a call to an API which is available here: <https://tools.highstakes.ch/geoloc-api/peers>

If you call the API you’ll get results like this:

Graphical user interface

Description automatically generated with medium confidence

The data you get from the json on “level 1” are the networks we want to display (in the above example that would be columbus-5 and osmosis-1).

We need to add another entry in the dropdown which will be “All networks”.

The API respond with a Json that contains a list of networks and under each network there is a list of “nodes” which will be used to display the “point of interests” / pins. Each entry is a point of interest and an example is show below:



We want to have an overlay item on the top left corner of the screen, a bit like google.maps to allow the user to show/hide the sidebar:

Text

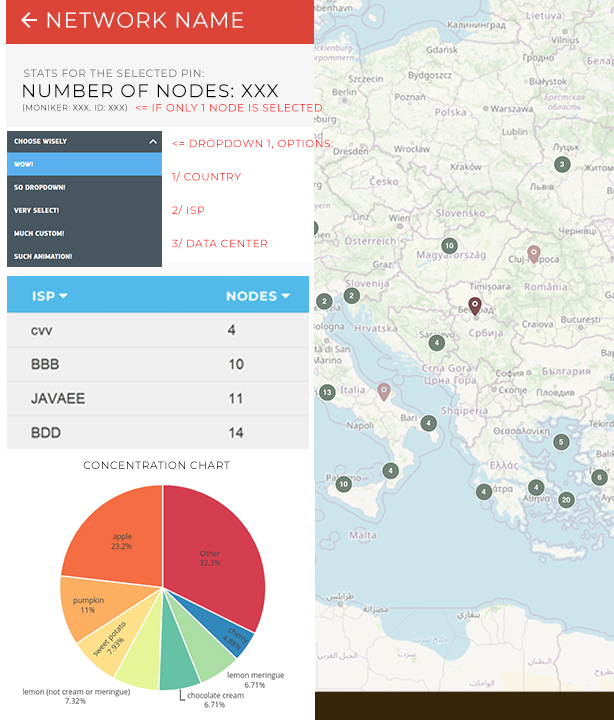
Description automatically generated

A clic on a pin will select it and will change its color (we keep the number in it if this is a “group of pins”):

A picture containing text

Description automatically generated

The click should also open up side bar on the left of the screen, a bit like google.maps, but something like the following:



* Network name in the red section is, of course, the name of the selected network. If no network are selected then this should read “All networks”
* The subtext “stats for the selected pin” should be different if no pin is selected. Instead this should read “stats for the whole world”.
* Number of nodes is the number of “pins” (point of interest) that are grouped under the selected pin. If only 1 pin is selected (not a group of pins or a pin with a number on it) then it’s okay to display only “number of node: 1” (but the “s” to nodes should then be removed – don’t forget to add it back when there is more than 1 node / pin selected)
* The details below are only to be displayed if there is just 1 point of interest on the pin and when that’s the case we will use the data in the json named “moniker” for the name of the moniker and “nodeID” for the ID of the node to be used.
* Network name should be using a red background same as the screenshot above
* Then we should see a dropdown designed like the one on the above screenshot and containing 3 entries:
  + Group by countries (this matches the “country” field in the json)
  + Group by ISP (that one is selected by default) (this matches the ISP field in the json)
  + Group by data center (that one matches the “as” entry in the json)
* Depending on which item is selected in the dropdown we should display a table and a pie chart below (same as designed above, using the framework vue.js).
* The table should display the list of countries (or ISP or data center) and the corresponding number of nodes in the 2nd column. That table should be sorted by default using the 2nd column, highest number of nodes first to the lowest. Each table header can be sorted using arrows, the 1st one triggers an alphabetical sort, the 2nd one a numerical sort.
* Below we should see a pie chart with a title: “decentralization chart”.
* The pie chart contains the same data as the table chart but uses the data from the second column to calculate percentages. Each line represents a percentage of the total number of nodes displayed on the chart. The pie chart should display both the label and percentage directly on the chart as shown on the above screenshot.
* IMPORTANT: Any data that would result in a percentage < 2% should be grouped in a category we will call “others” that should contain all “nodes” that are statistically not relevant on their own and need to be grouped to be displayed without causing pie chart fragmenting.

Any click on the world map that is not a point of interest should deselect any selected pin or point of interest and most importantly it should restart the “filters” to the “world data”.

**Important note on performance:**

The API we provide uses real data so the volumetry is roughly the same we will have in production mode. The interface should run smoothly and any interaction with the user interface that lasts or can lasts more than 1 second should be using a loader so that the user interface remains graphically nice to look at.

When zooming in or out, it should run as smoothly as the example on tilia.earth for grouping / ungrouping point of interests.

**Delivery / Server:**

Delivery will be made on our dedicated server (VPS). FTP access or other access will be available on demand. Usage of a database is permitted if necessary. We will use a UNIX based server and nginx.