



# Enlighten-Web - data visualization tool and virtual research environment

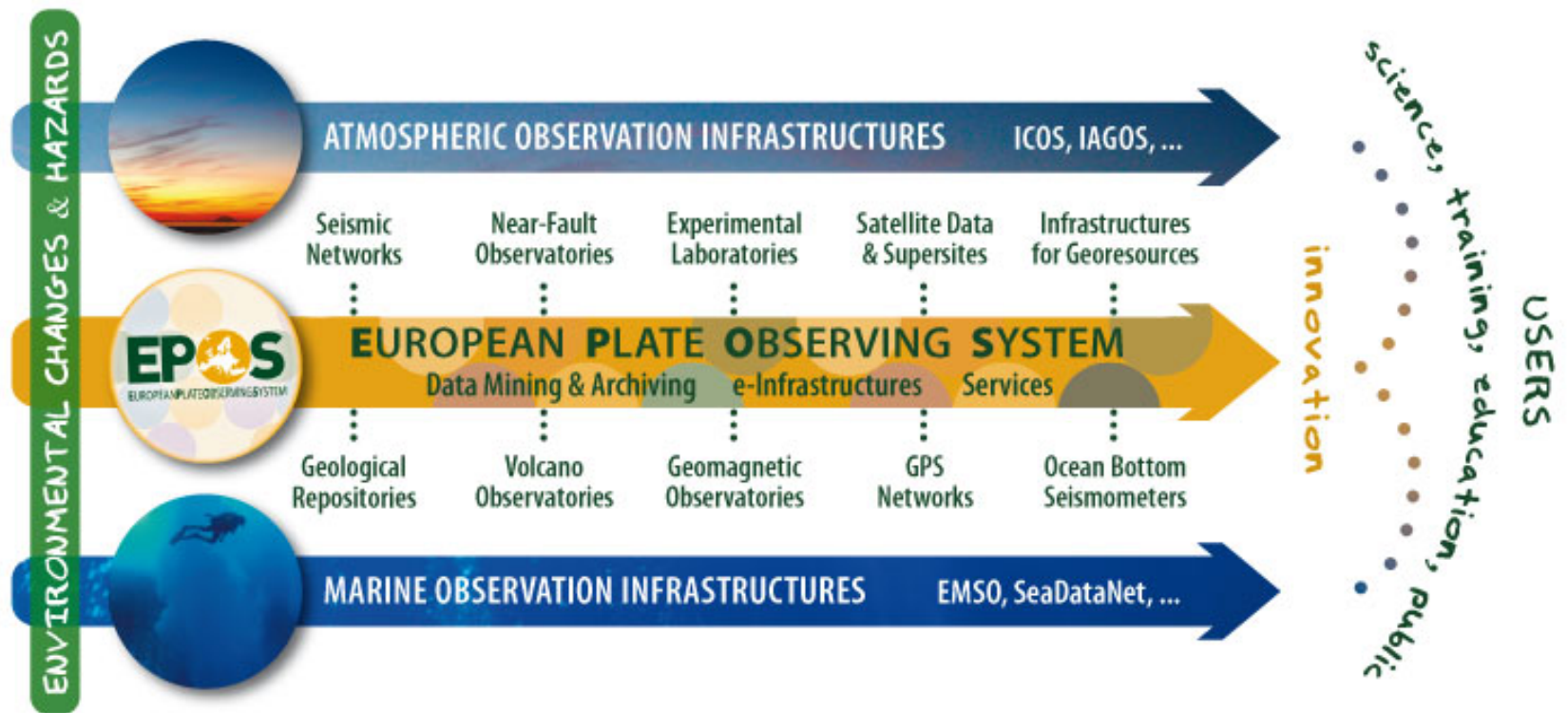
Jan Michálek<sup>1</sup>, Tor Langeland<sup>2</sup>, Ove Lampe<sup>2</sup>, Gro  
Fonnes<sup>2</sup>



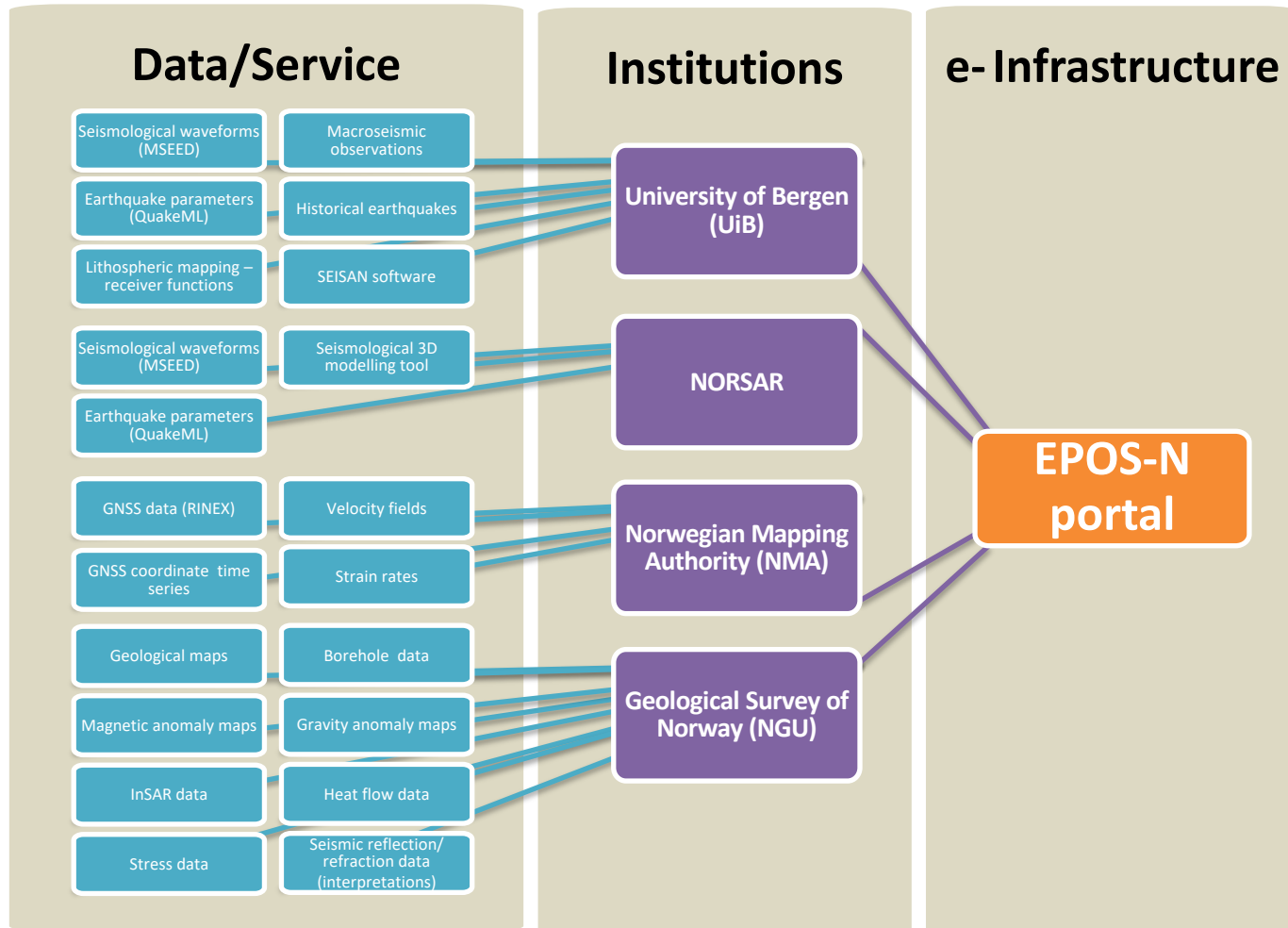
University of Bergen, Norway<sup>1</sup>  
NORCE, Bergen, Norway<sup>2</sup>



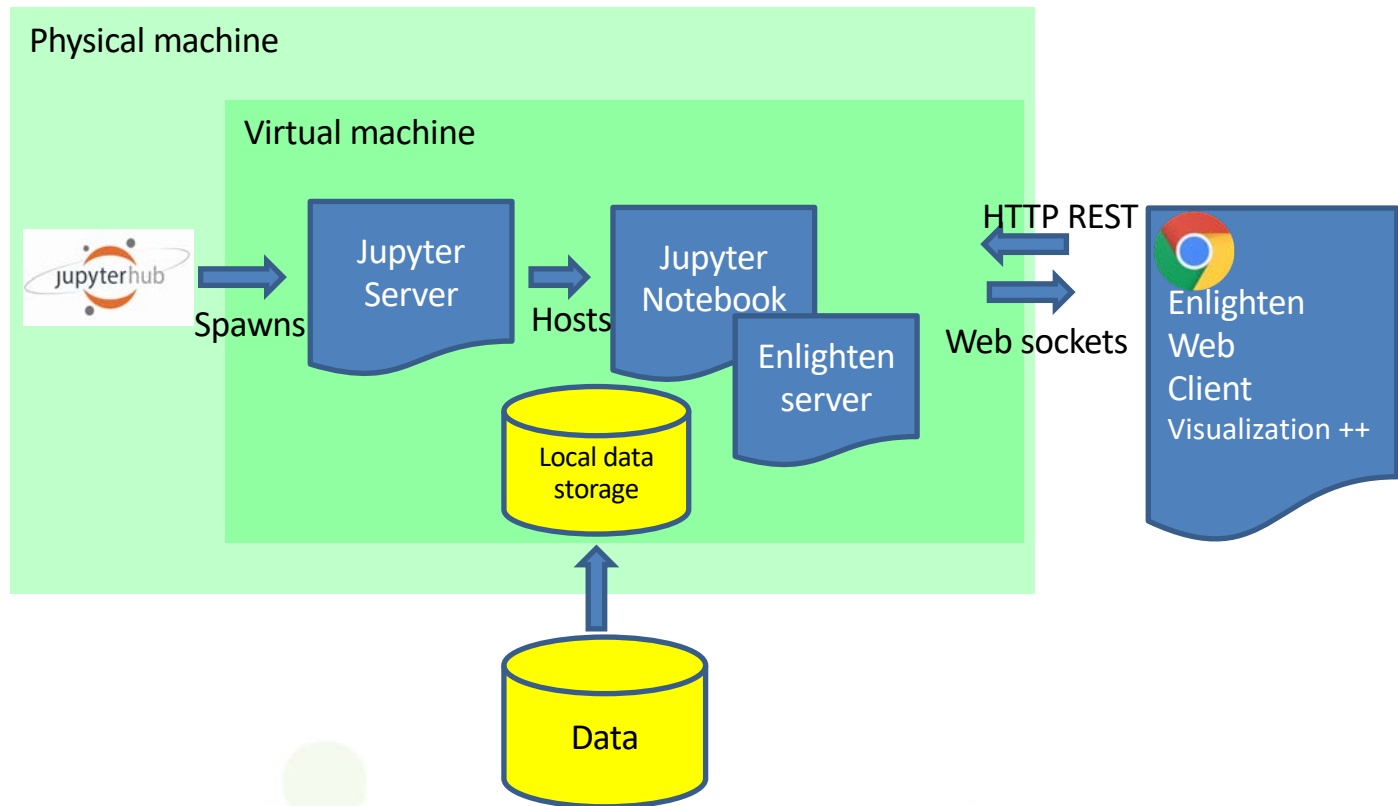
# EPOS and other e-infrastructures



# EPOS-N project

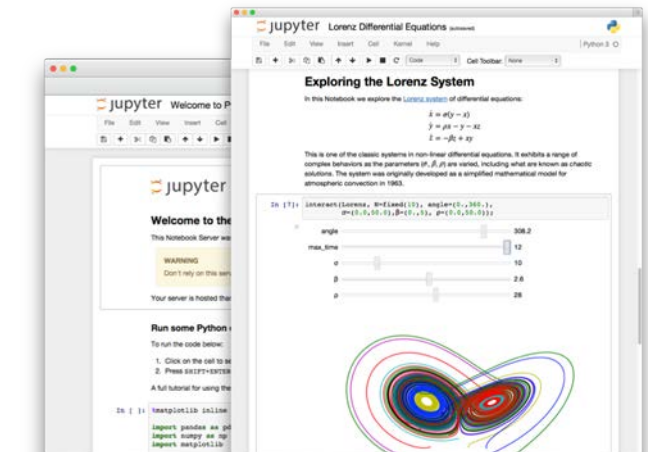


# Enlighten-Web VRE

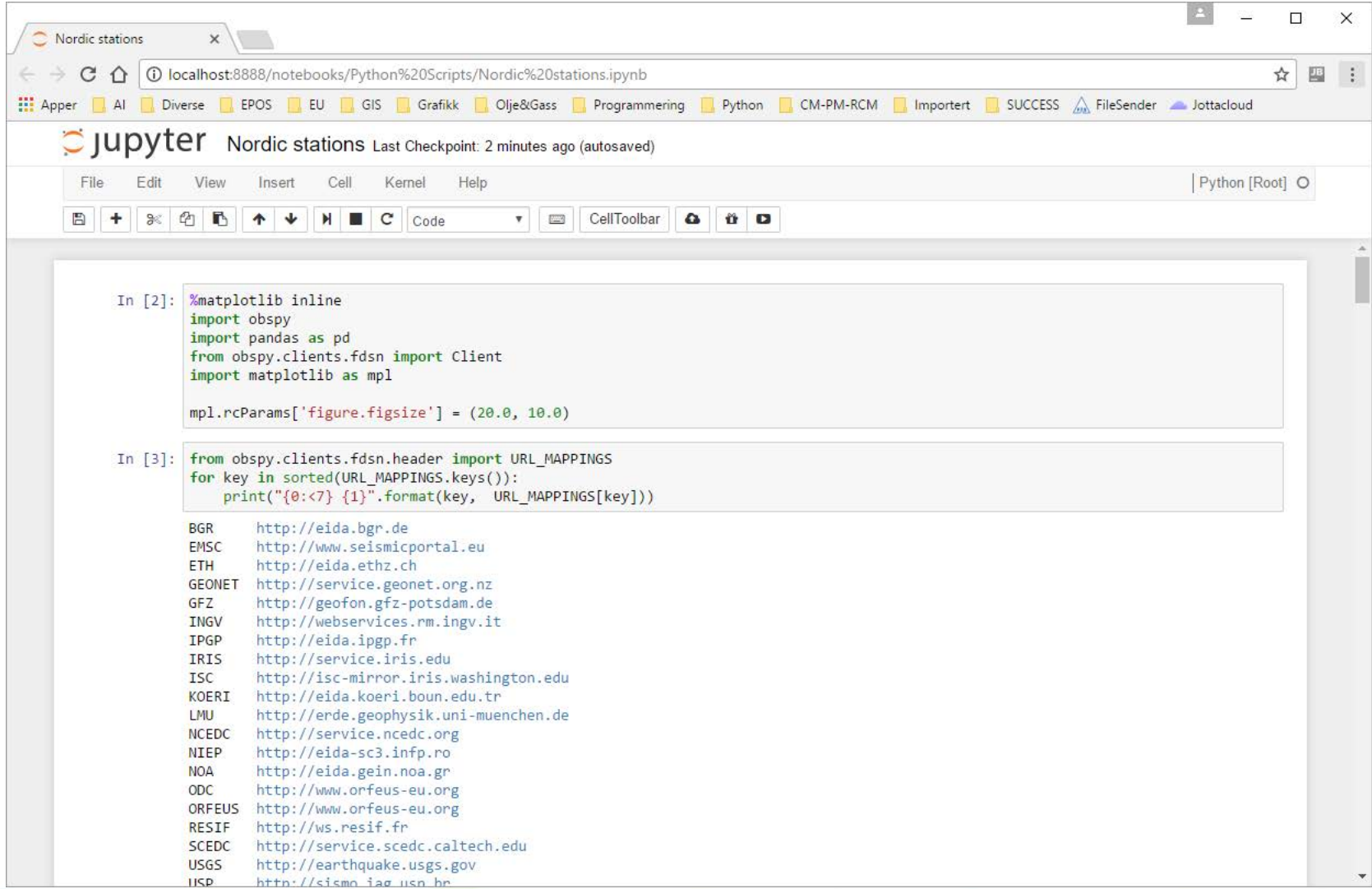


# Jupyter notebook

- A web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text or presentations
- Enables programming from a web-browser in programming language by choice
  - Python, R, C++, Fortran
- Code runs on server



# Notebook example



The screenshot shows a Jupyter Notebook window titled "Nordic stations". The browser address bar shows the URL `localhost:8888/notebooks/Python%20Scripts/Nordic%20stations.ipynb`. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with various icons. The code is displayed in two input cells:

```
In [2]: %matplotlib inline
import obspy
import pandas as pd
from obspy.clients.fdsn import Client
import matplotlib as mpl

mpl.rcParams['figure.figsize'] = (20.0, 10.0)

In [3]: from obspy.clients.fdsn.header import URL_MAPPINGS
for key in sorted(URL_MAPPINGS.keys()):
    print("{0:<7} {1}".format(key, URL_MAPPINGS[key]))
```

The output of the second cell is a list of station codes and their corresponding URLs:

```
BGR      http://eida.bgr.de
EMSC     http://www.seismicportal.eu
ETH      http://eida.ethz.ch
GEONET   http://service.geonet.org.nz
GFZ      http://geofon.gfz-potsdam.de
INGV     http://webservices.rm.ingv.it
IPGP     http://eida.ipgp.fr
IRIS     http://service.iris.edu
ISC      http://isc-mirror.iris.washington.edu
KOERI    http://eida.koeri.boun.edu.tr
LMU      http://erde.geophysik.uni-muenchen.de
NCEDC    http://service.ncedc.org
NIEP     http://eida-sc3.infp.ro
NOA      http://eida.gein.noa.gr
ODC      http://www.orfeus-eu.org
ORFEUS   http://www.orfeus-eu.org
RESIF    http://ws.resif.fr
SCEDC    http://service.scedc.caltech.edu
USGS     http://earthquake.usgs.gov
USP      http://sismo.iag.usp.br
```

# Notebook example cont.

```
In [20]: orfeus = Client("ORFEUS")
stations = orfeus.get_stations(minlongitude=3, minlatitude=54, maxlongitude=32, maxlatitude=81)

gfz = Client("GFZ")
stations += gfz.get_stations(network="UP,HE,DK")

print (stations)
```

Inventory created at 2017-05-18T09:14:26.000000Z

Sending institution: SeisComP3 (ODC)

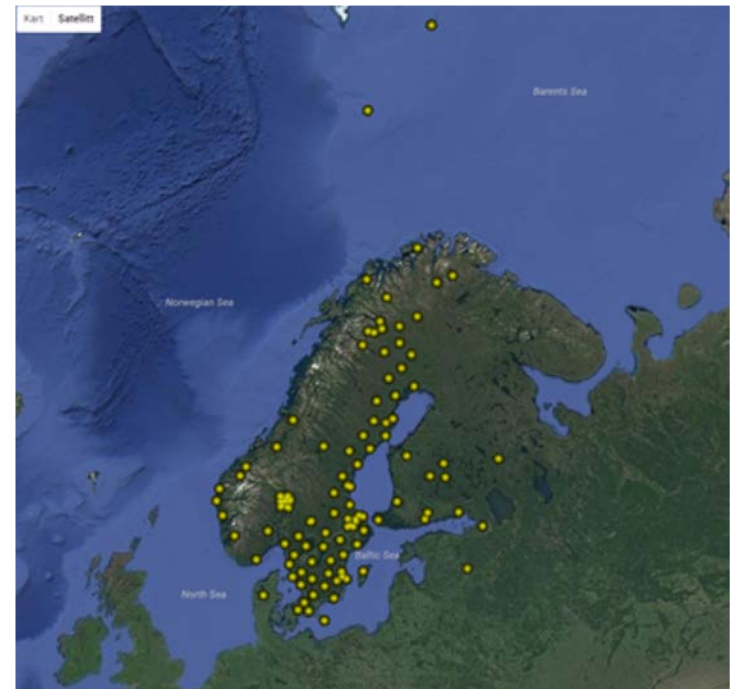
Contains:

Networks (6):

HF  
IU  
NO  
NR  
NS  
UP

Stations (55):

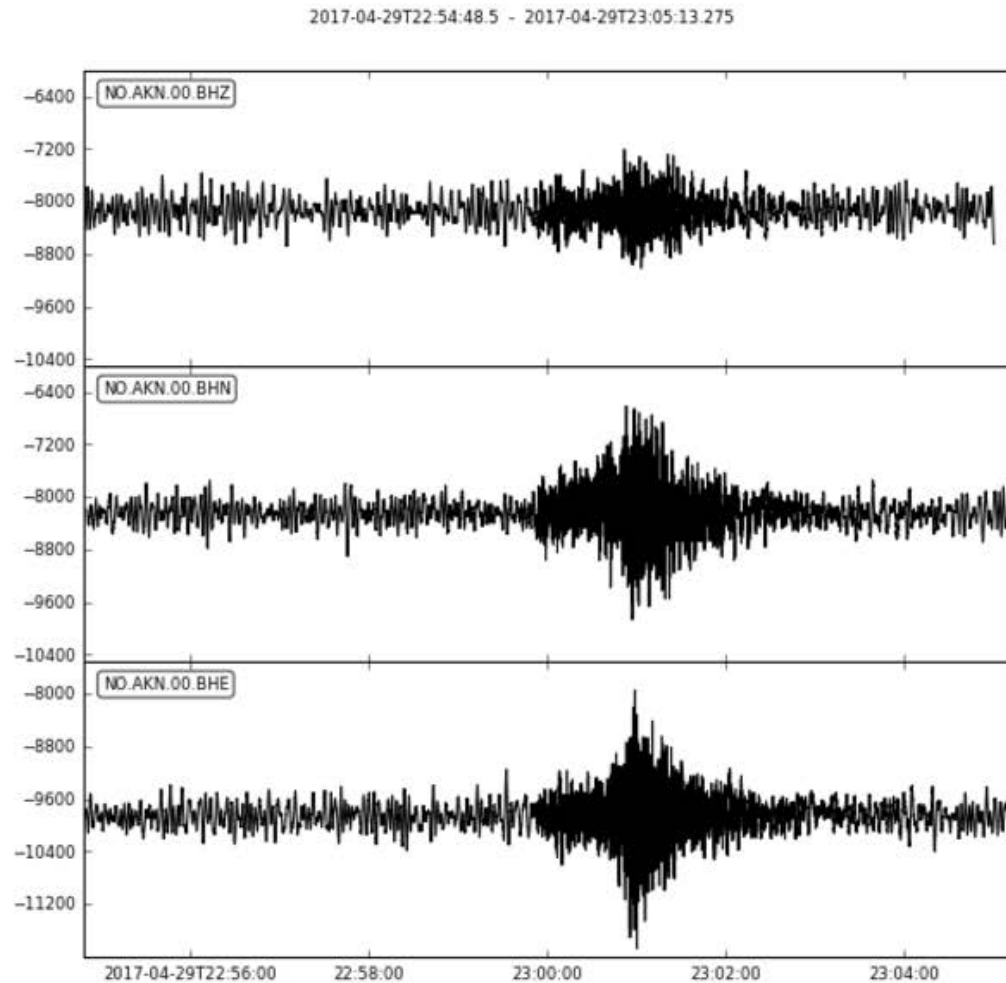
HF.HFC2 (HFC2)  
HF.HFSC2 (old Hagfors Array Site)  
IU.KBS (Ny-Alesund, Spitzbergen, Norway)  
IU.KEV (Kevo, Finland)  
IU.KONO (Kongsberg, Norway)  
NO.AKN (Aaknes, Norway)  
NO.ARE0 (ARE0)  
NO.NAO01 (NORSAR ARRAY SITE 01A01)  
NO.NAO03 (NORSAR ARRAY SITE 01A03)  
NO.NB201 (NORSAR ARRAY SITE 02B01)  
NO.NB204 (NORSAR ARRAY SITE 02B04)  
NO.NBO00 (NORSAR ARRAY SITE 01B00)  
NO.NBO03 (NORSAR ARRAY SITE 01B03)  
NO.NC203 (NORSAR ARRAY SITE 02C03)  
NO.NC204 (NORSAR ARRAY SITE 02C04)  
NO.NC301 (NORSAR ARRAY SITE 03C01)  
NO.NC303 (NORSAR ARRAY SITE 03C03)



# Notebook example cont.

```
In [43]: from obspy import UTCDateTime
         t1 = UTCDateTime("2017-04-29T22:55:00")
         st = orfeus.get_waveforms("NO", "AKN", "*", "*", t1, t1+10*60)
```

```
In [44]: st.plot()
```





# Notebook example cont.

```
In [12]: import pandas as pd
import matplotlib as mpl

import obspy
from obspy.clients.fdsn import Client
emsc = Client('EMSC')

from obspy import Catalog
events = Catalog()

from obspy import UTCDateTime
t0 = UTCDateTime("2016-05-01T00:00:00.000")
t1 = UTCDateTime("2017-05-18T00:00:00.000")
dt = 3*24*60*60
```

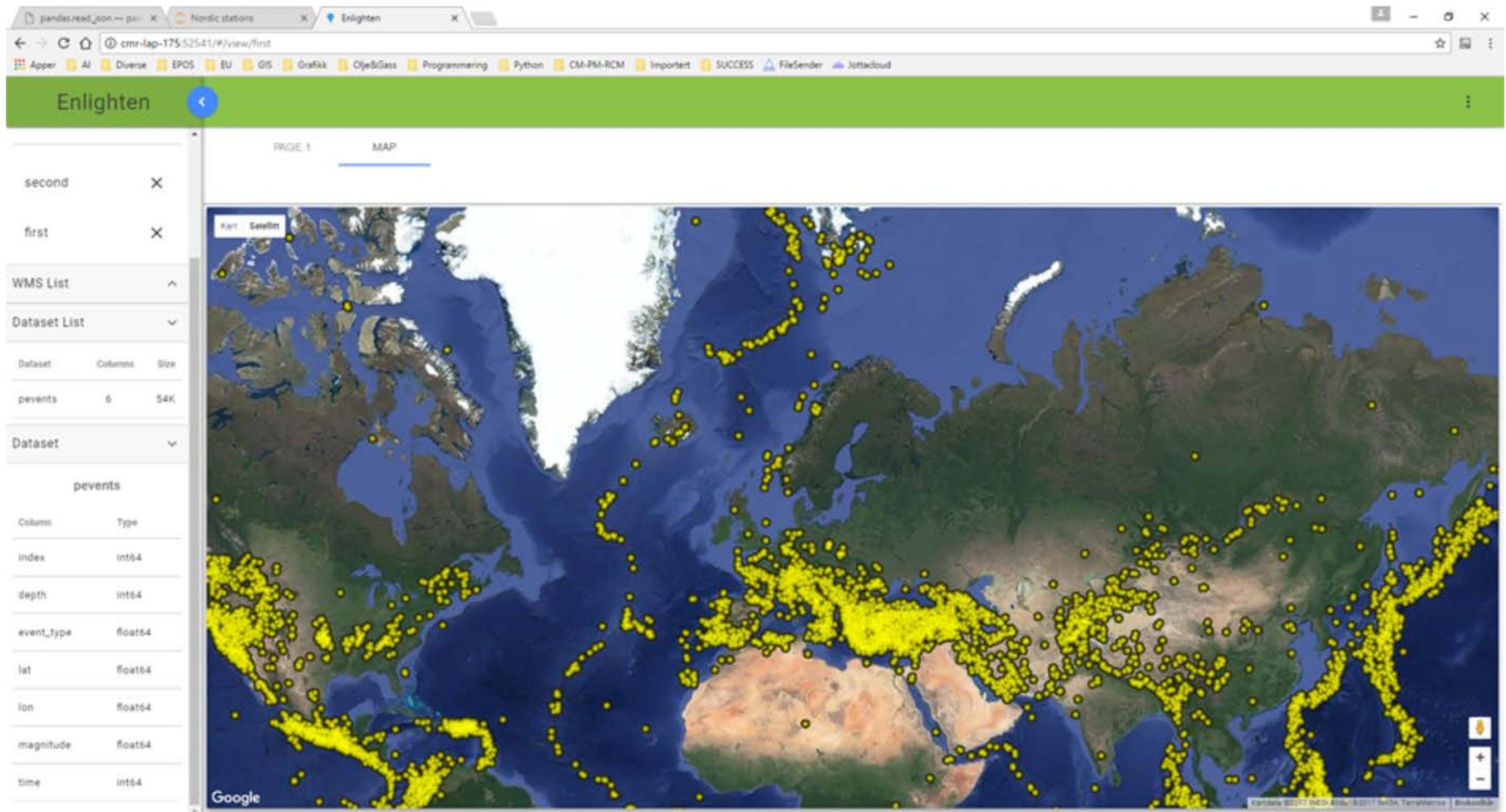
Query events from EMSC. Need to split in shorter time spans for each query because of limitations on response size

```
In [13]: while True:
    ev_period = emsc.get_events( t0,t0+dt )
    events += ev_period
    print(t0, ", len = ", len(ev_period))

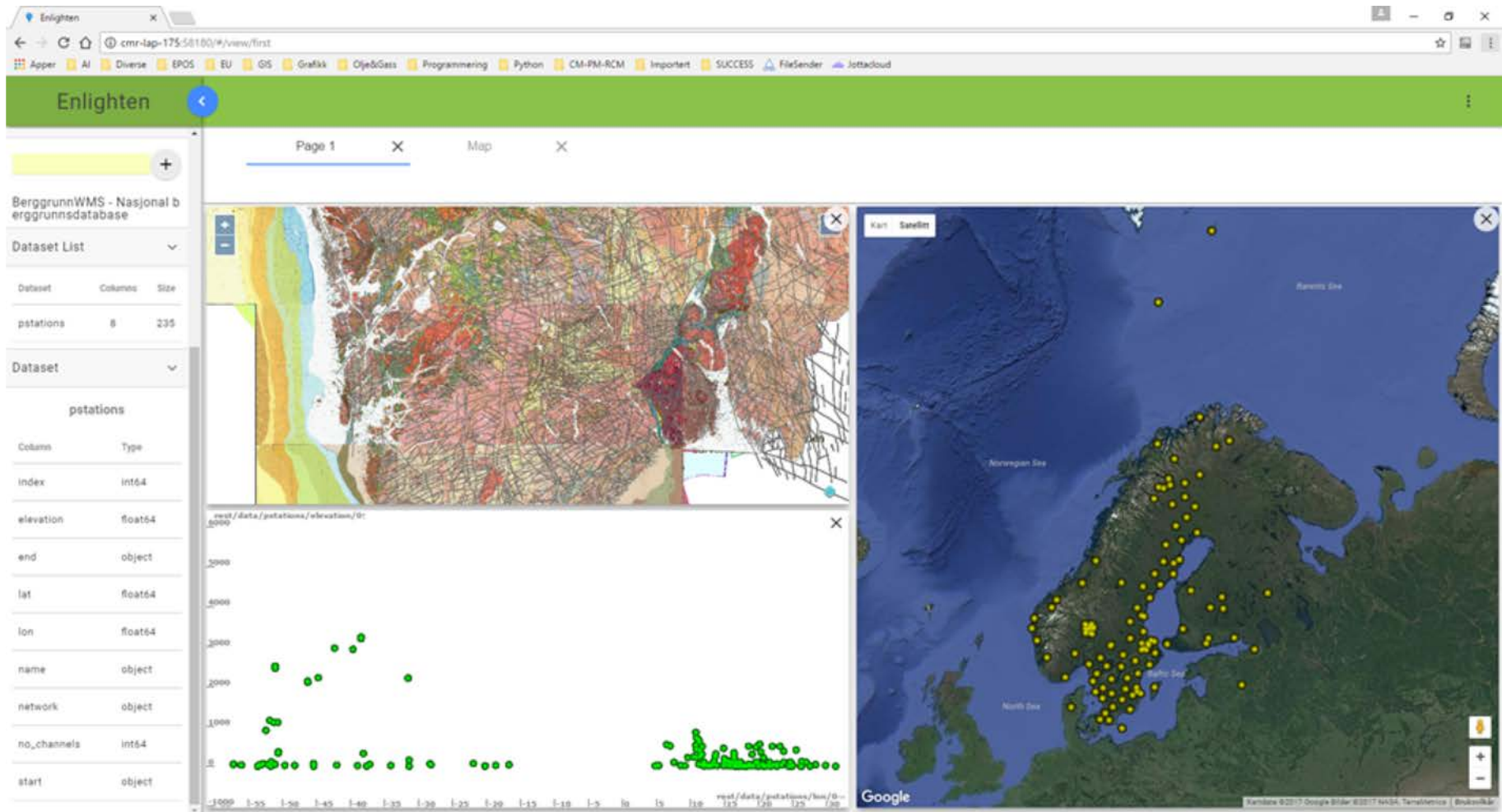
    t0+=dt
    if t0 >= t1:
        break
```

```
2016-05-18T00:00:00.000000Z , len = 388
2016-05-21T00:00:00.000000Z , len = 406
2016-05-24T00:00:00.000000Z , len = 380
2016-05-27T00:00:00.000000Z , len = 416
2016-05-30T00:00:00.000000Z , len = 378
2016-06-02T00:00:00.000000Z , len = 426
```

# Enlighten-Web visualization of seismic events



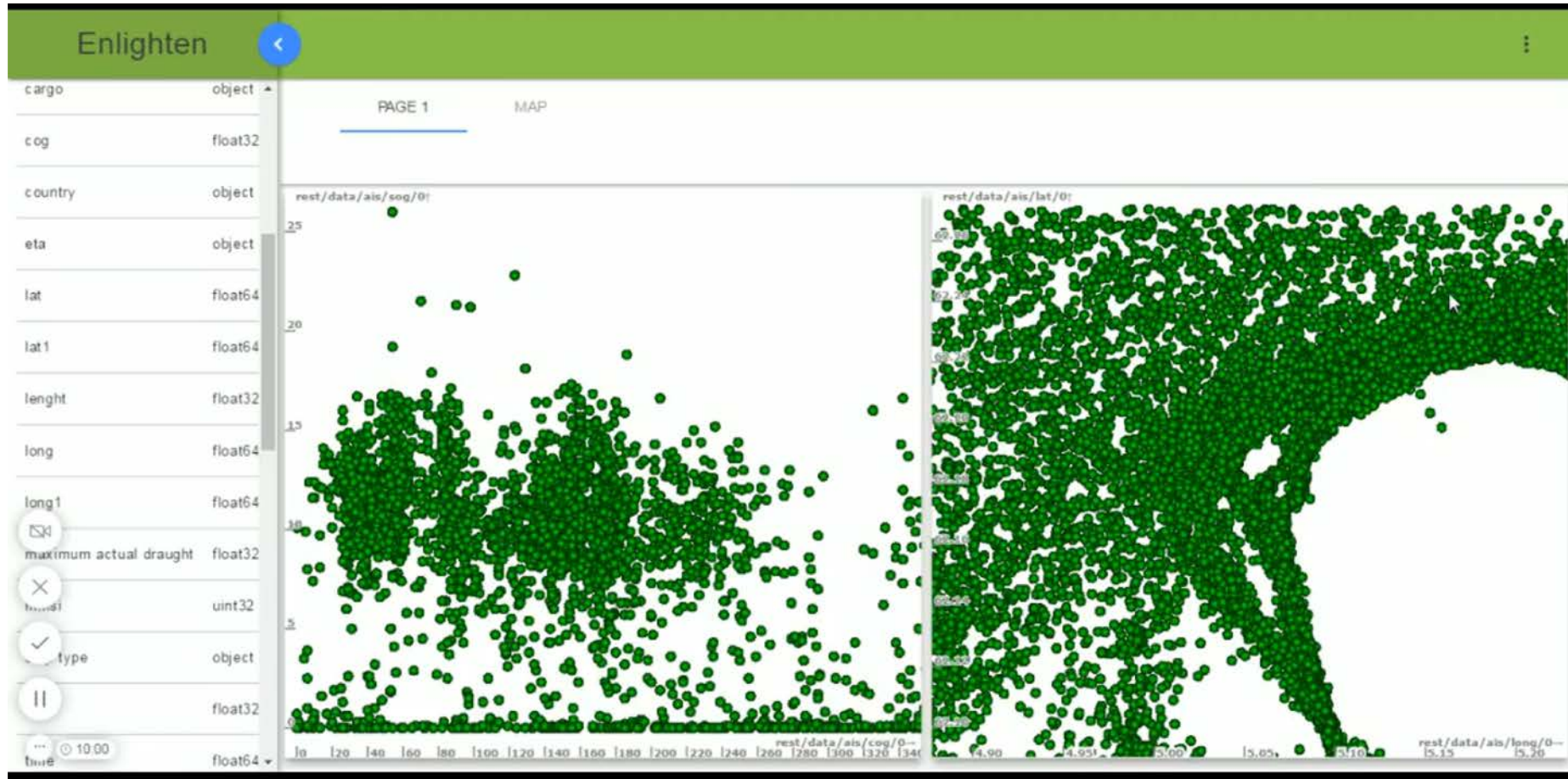
# The Enlighten-web client for visualization



# Enlighten-web features

- Interactive real-time visualization linked to a dynamic programming interface
- Effective visualization of large multidimensional data sets
- Interactive mapping of millions of points
- Explorative visualization, e.g. for extracting trends and outliers in the data
- Confirmative visualization for analysis of hypotheses
- Brushing and linking

# Brushing and linking



# Jupyter Notebooks

- Seismology
  - [http://krischer.github.io/seismo\\_live/](http://krischer.github.io/seismo_live/)
- A gallery of interesting Jupyter Notebooks
  - <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>

# Feedback

- <https://goo.gl/forms/W0nDOGfm2T2nD6pW2>
- Reward – lottery
  - Wed morning

*And now let's do some work!*