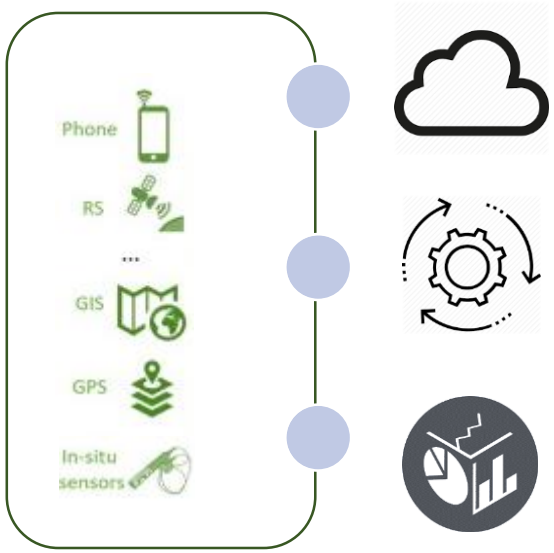
Big Data and Tools

(GBDX, aWhere, etc)

GEOAGRO ICARDA CGIAR

S. Ghosh
C. Biradar

Geoinformatics and Big Data



Big Data Framework

Insights

Volume	**{Amount of space}**	**{Petabytes of archive data}**
Variety	**{Diversity}**	**{heterogeneity in data and sensors}**
Velocity	**{Momentum}**	**{Cloud storage and processing, API Services}**
Veracity	**{Accuracy}**	**{Machine learning algorithms}**

Big Data Tools

GBDX
aWhere
Google Earth Engine

aWhere

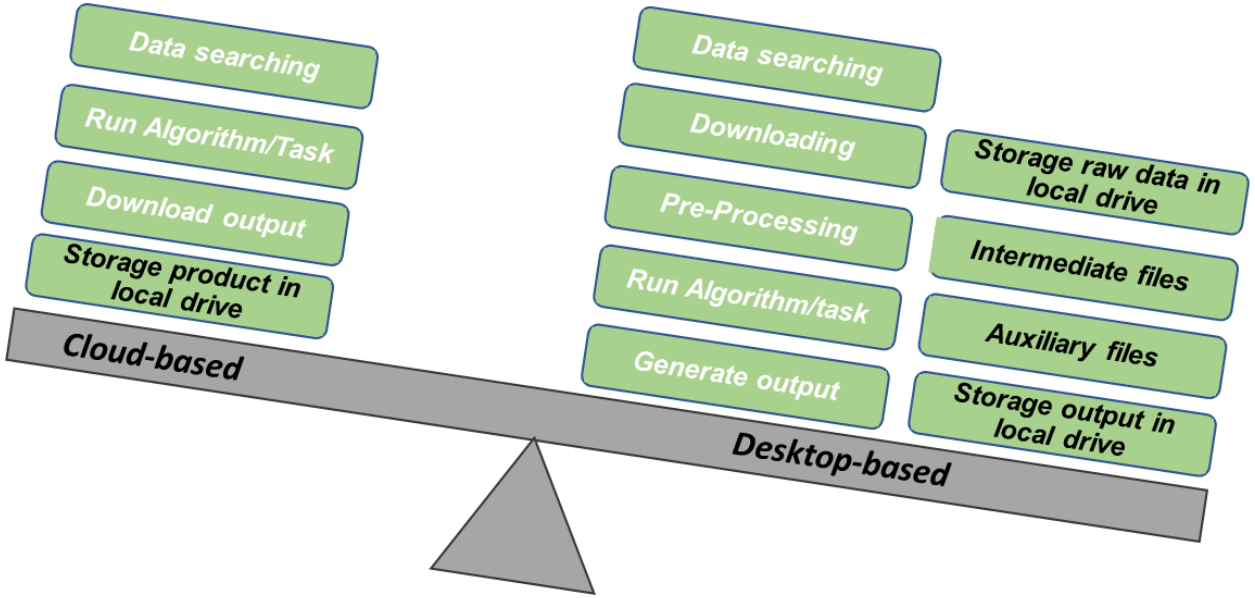
Python: Jupyter Notebook
R
Java script

jupyter

HDFS
Spark
MongoDB

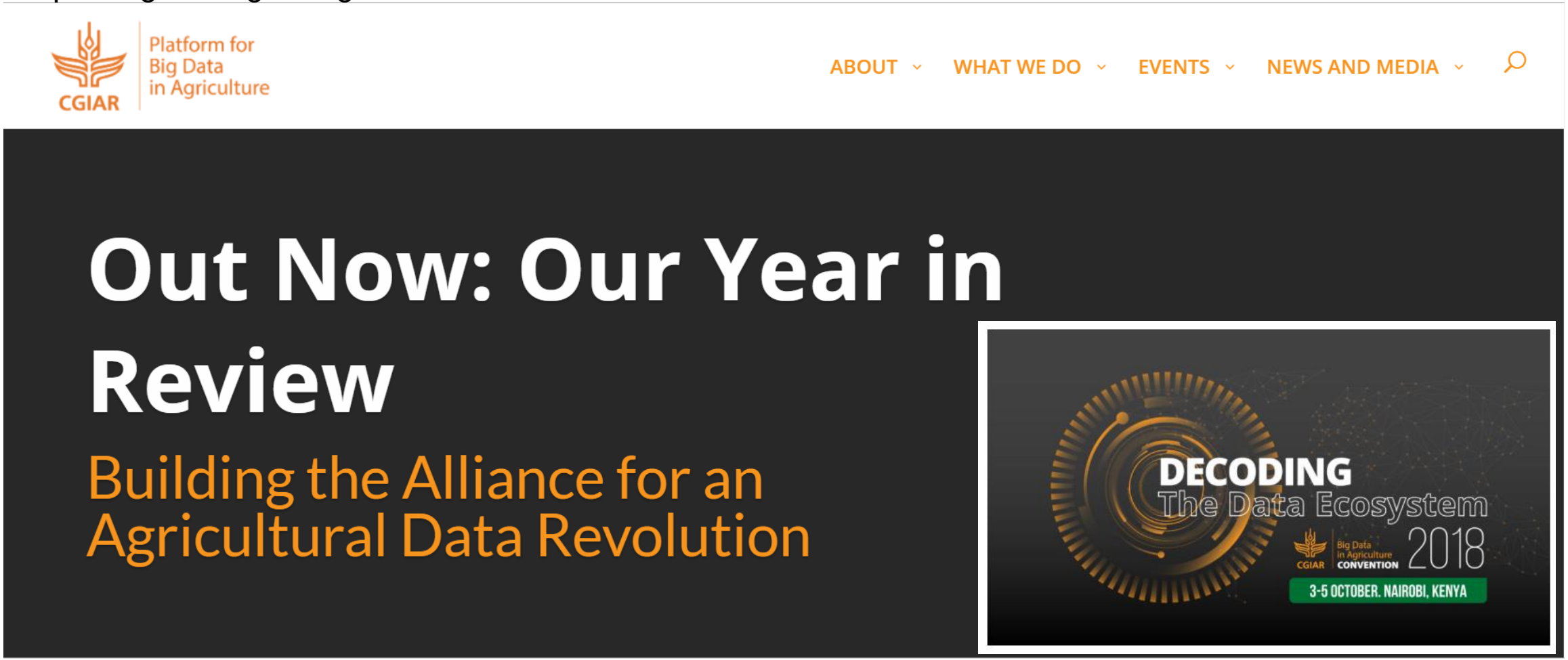
Why?

- Downloading dataset
- Processing huge dataset
- Sharing dataset



CGIAR Initiative

https://bigdata.cgiar.org/



API

An Application Programming Interface (**API**) is a set of subroutine definitions, communication protocols, and tools for building software/application.

https://www.youtube.com/watch?v=s7wmiS2mSXY

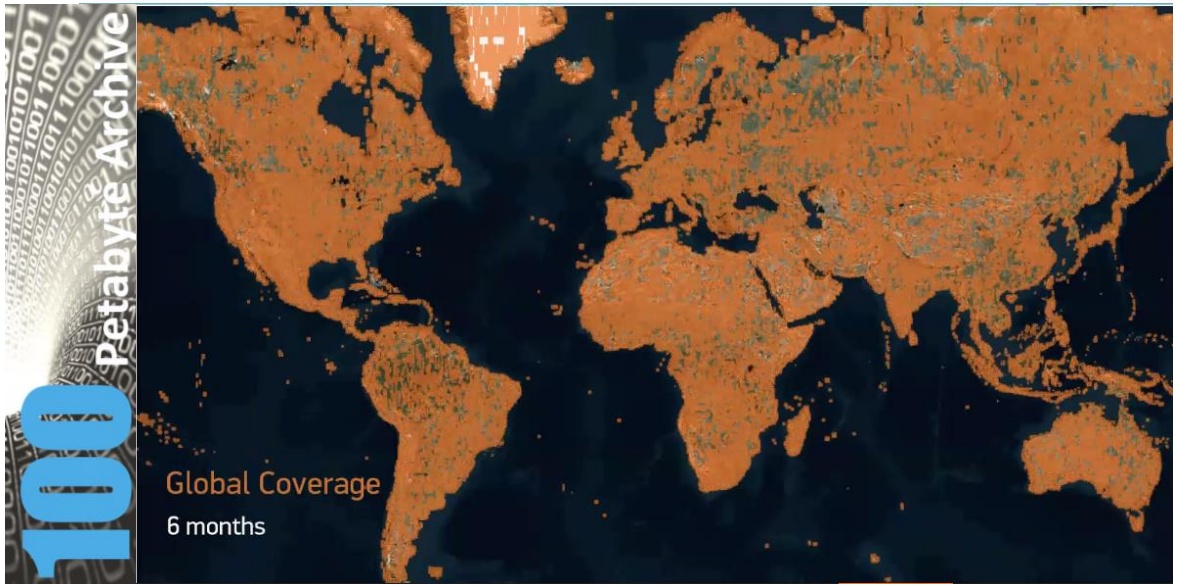
GBDX

Geo Big Data Ecosystem

- Platform and Workflow
- GitHub and Python Jupyter Notebook
- Very High resolution data and potential of very high resolution data
- Different sensor data and their characteristics
- Data Searching and Few basic syntax
- Case study
- Discussion

GBDX Platform

How much Information DigitalGlobe Collects



GBDX APIs

amazon web services™

docker

AWS - Amazon Web Services
on demand cloud computing platform

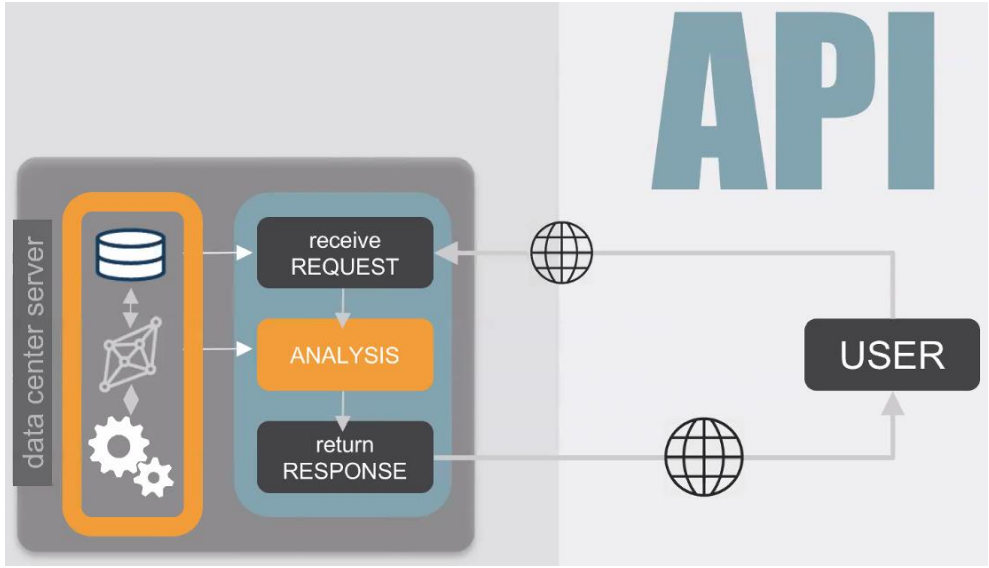
S3 - Simple Scalable Storage
store and retrieve any amount of data

EC2 - Elastic Cloud Compute
scalable computing capacity

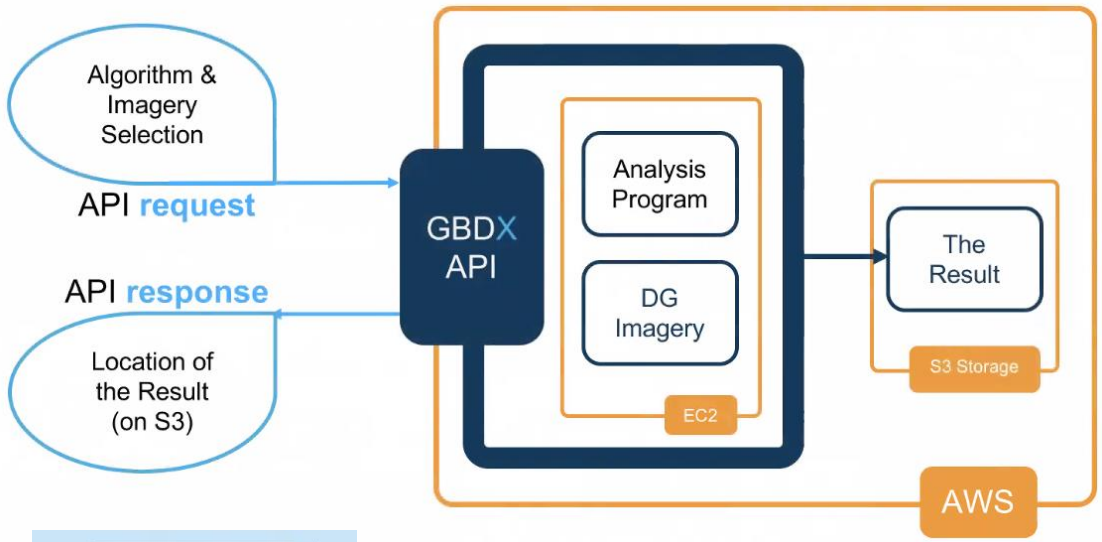
Docker
software containerization platform

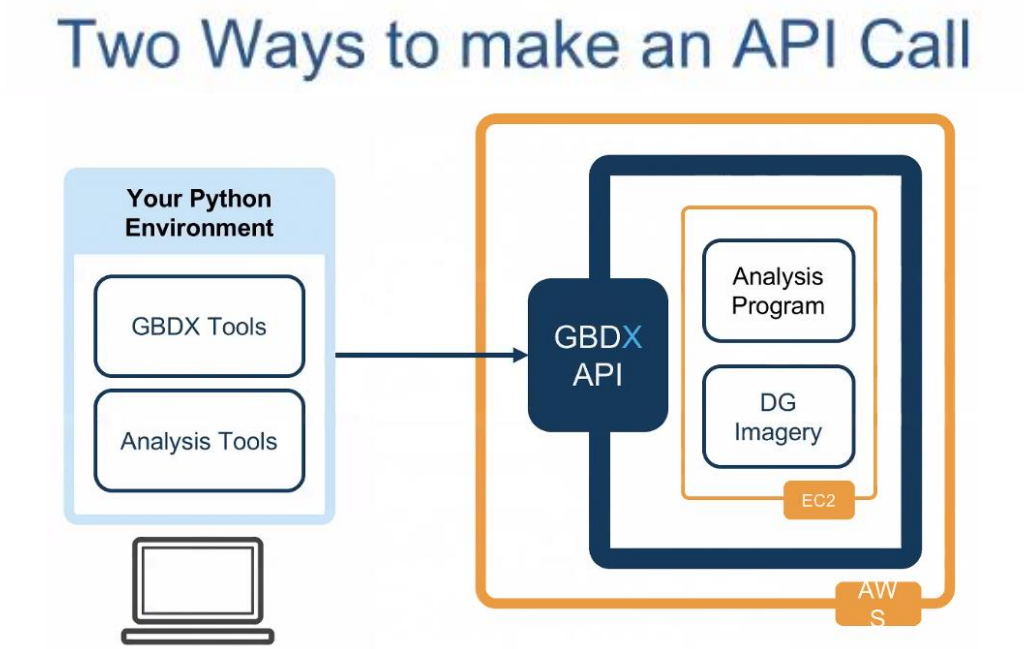
Docker Image
standardized wrapper for your algorithm

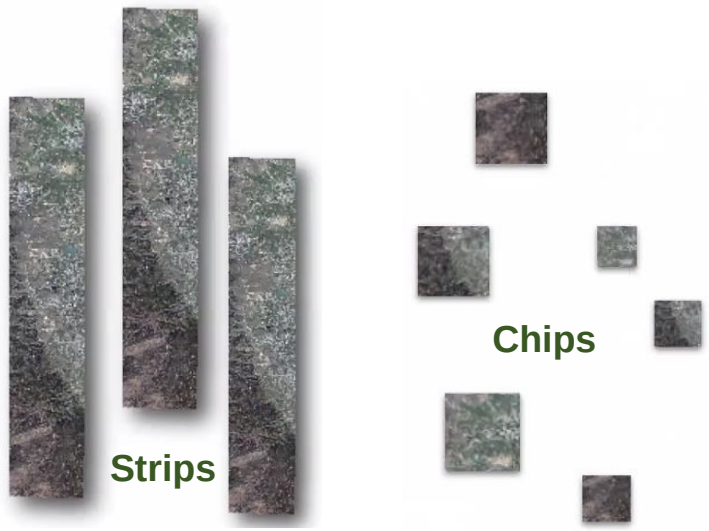
Docker Hub
a repository of images



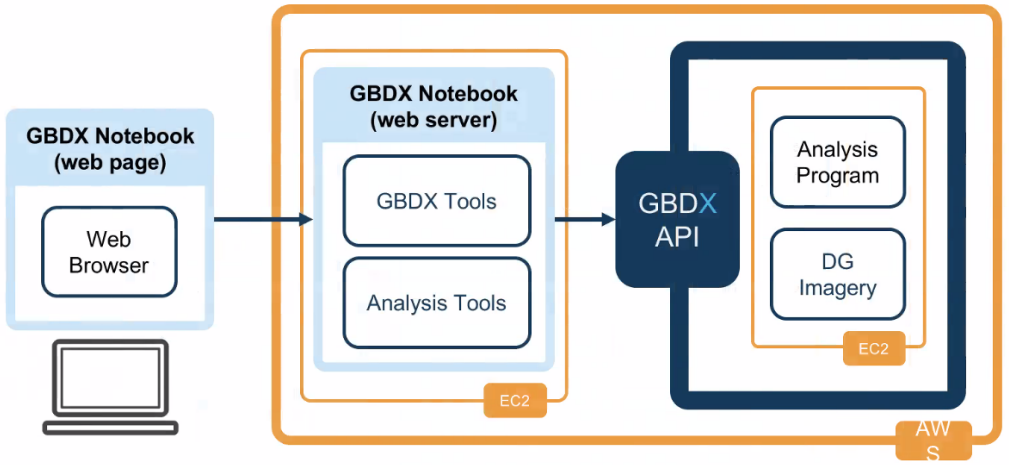
How it works

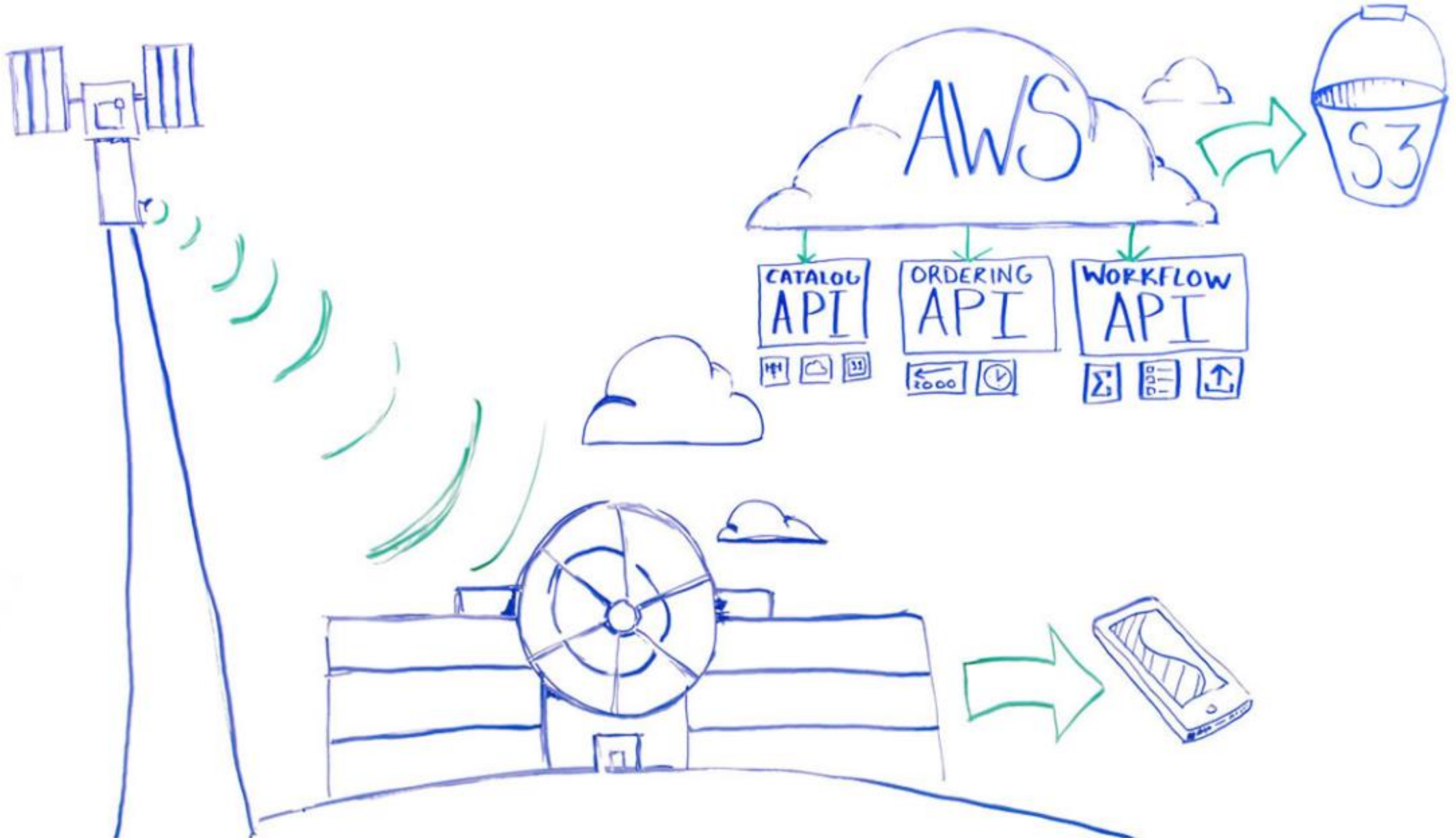






Strips and Chips: Processing time and Data volume



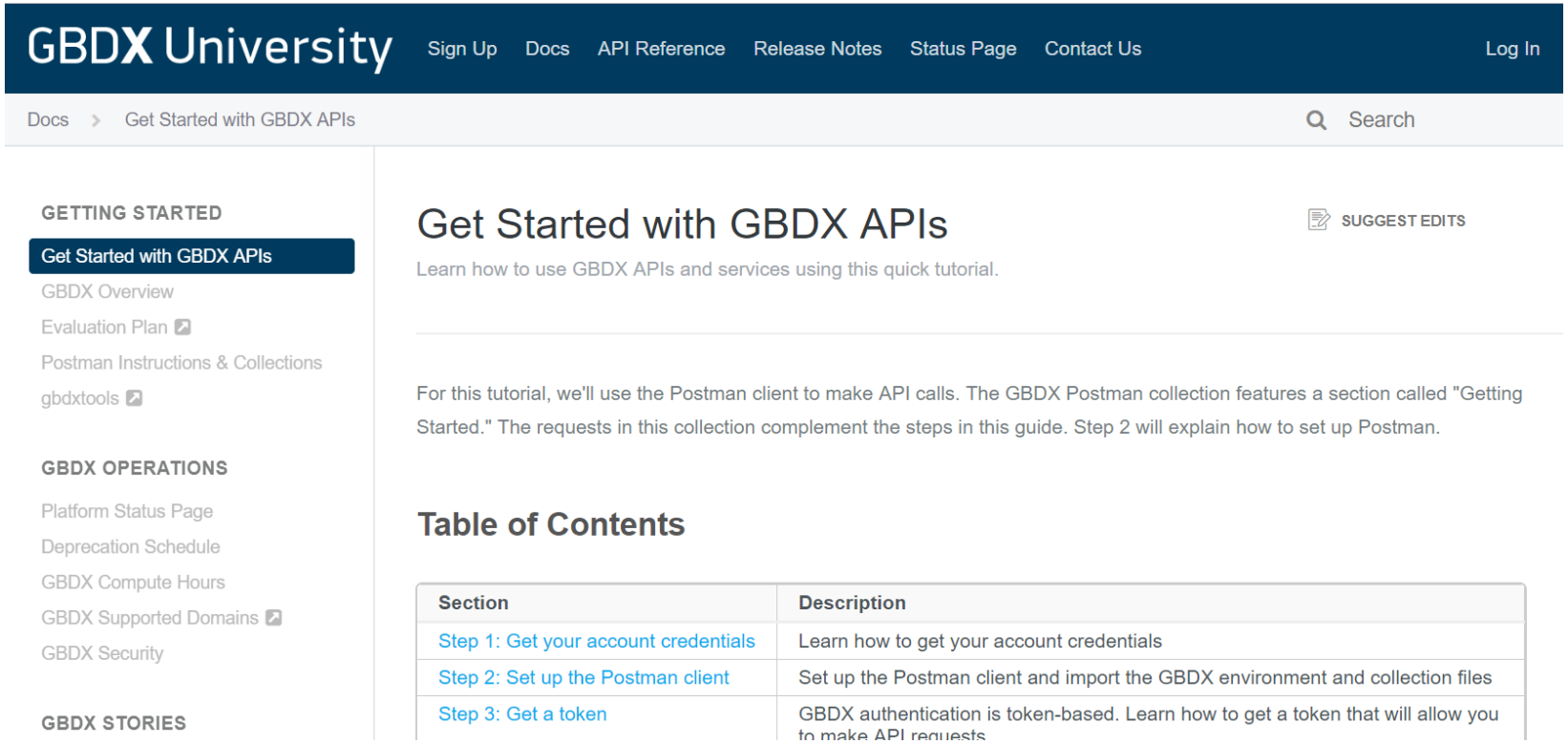

AWS
S3
CATALOG
API
ORDERING
API
WORKFLOW
API

Available data

DigitalGlobe
- WorldView-3
- WorldView-2
- WorldView-1
- GeoEye-1
- IKONOS
- QuickBird

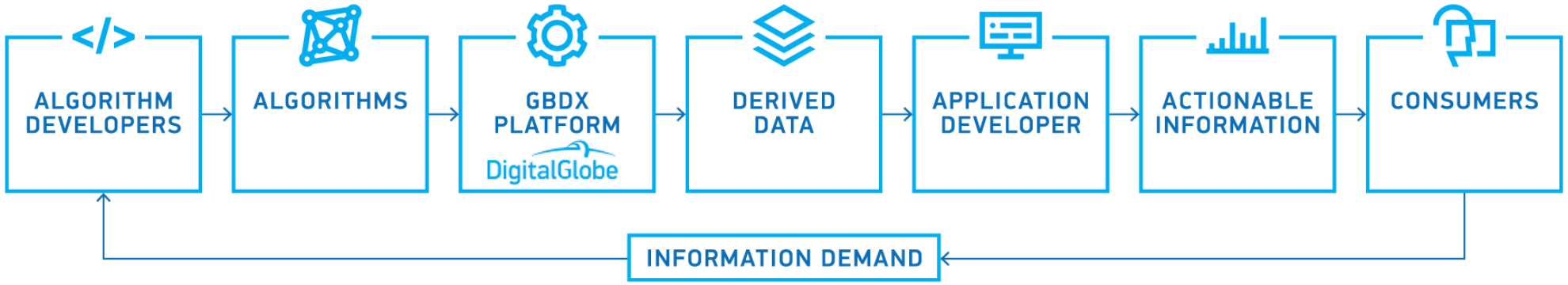
MDA
- RADARSAT-2

NASA
- Landsat-8



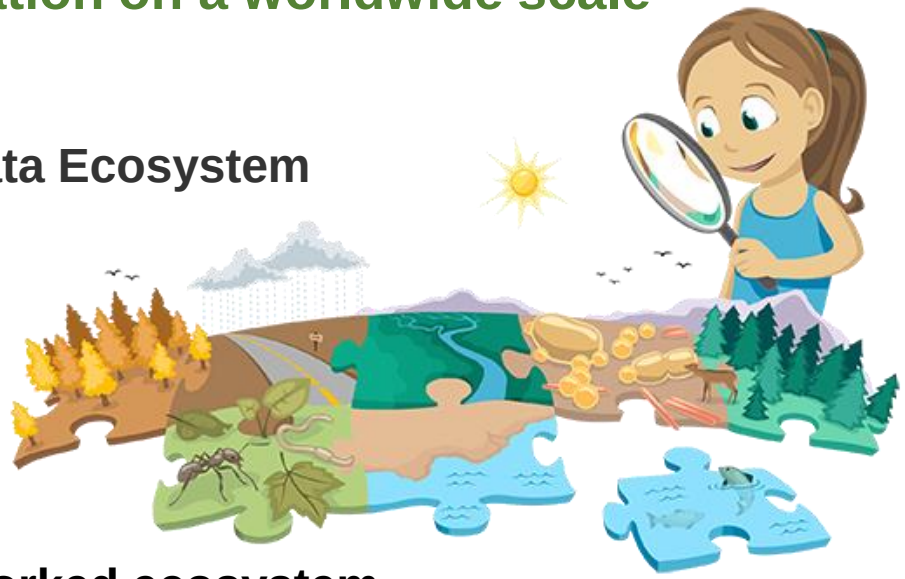
https://gbdxdocs.digitalglobe.com/docs/getting-started

Moving Beyond the Pixel



The GBDX platform enables to **build, access and run advanced workflows and tools** that extract **actionable information on a worldwide scale**

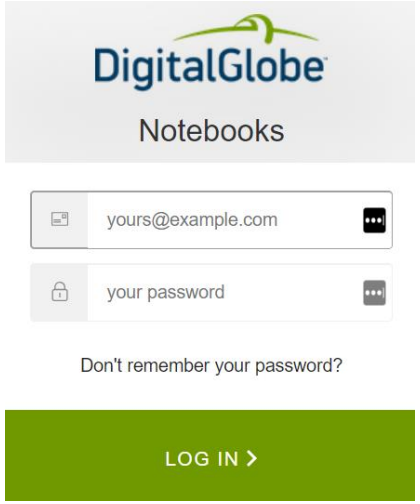
Geo Big Data Ecosystem

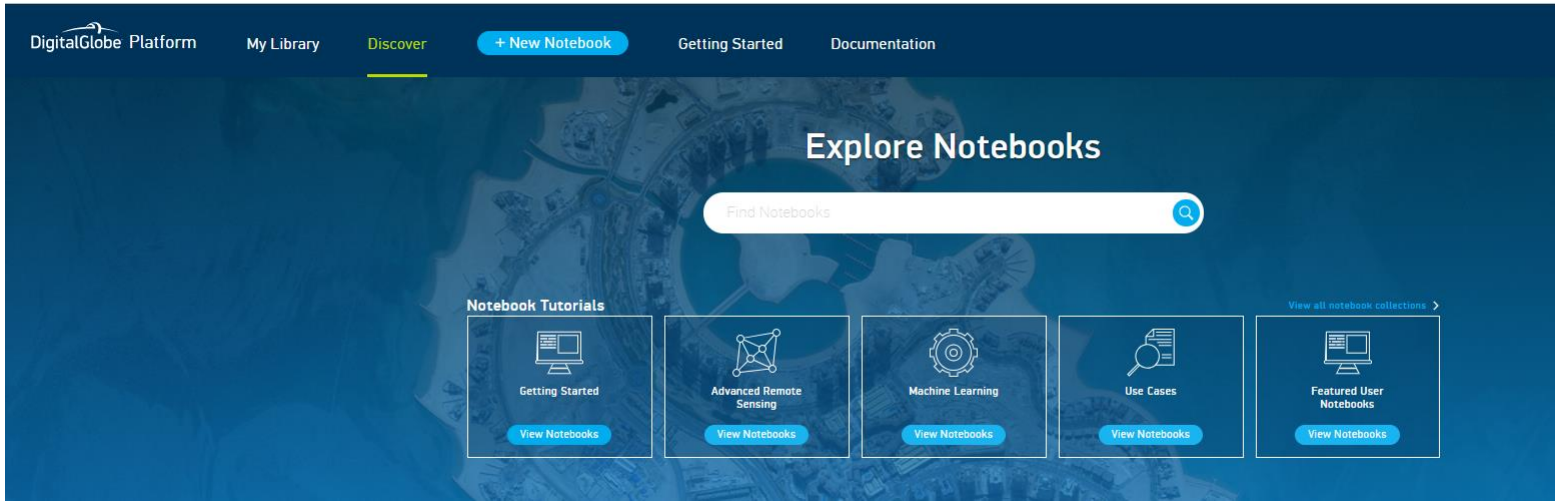


Networked ecosystem

Enables data Producers and Developers to **build solutions**

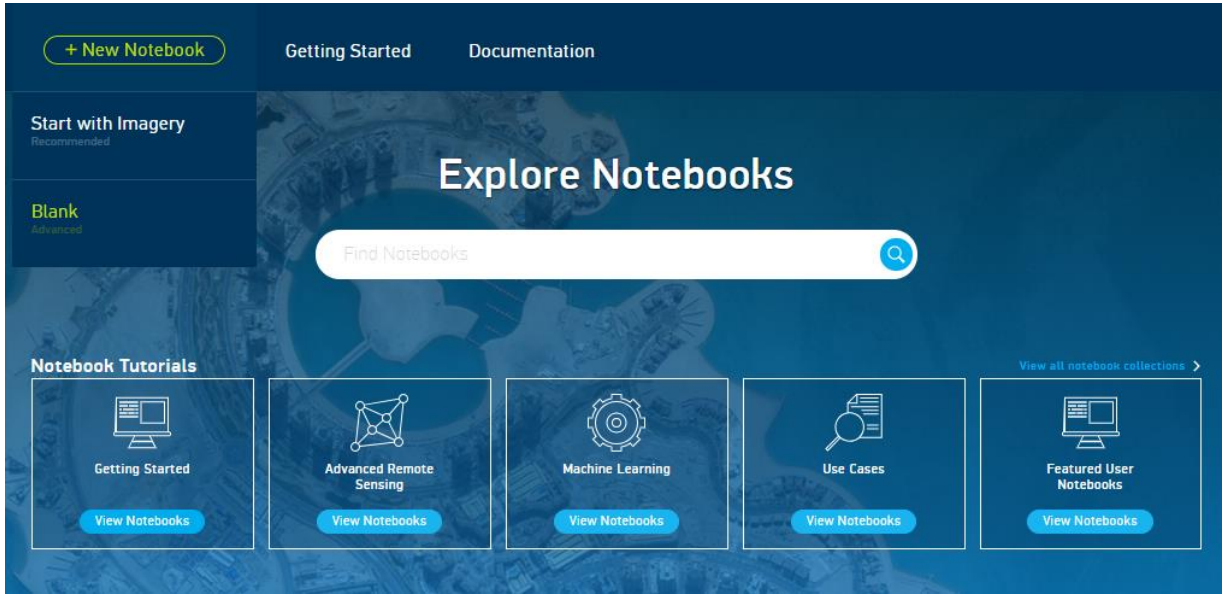
GBDX Notebook

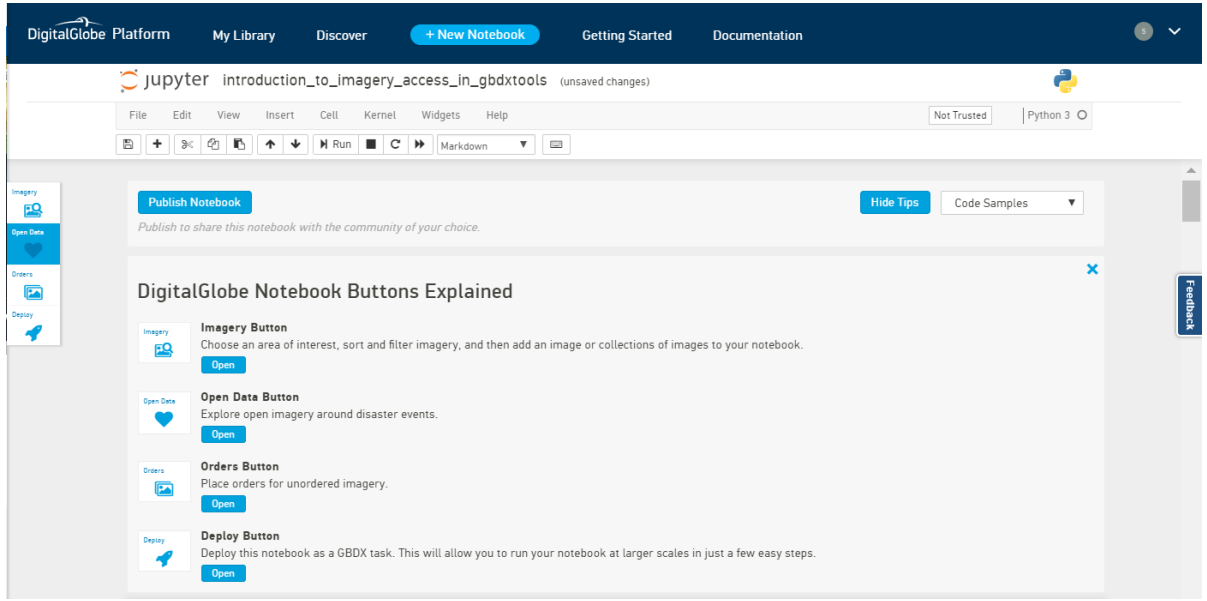


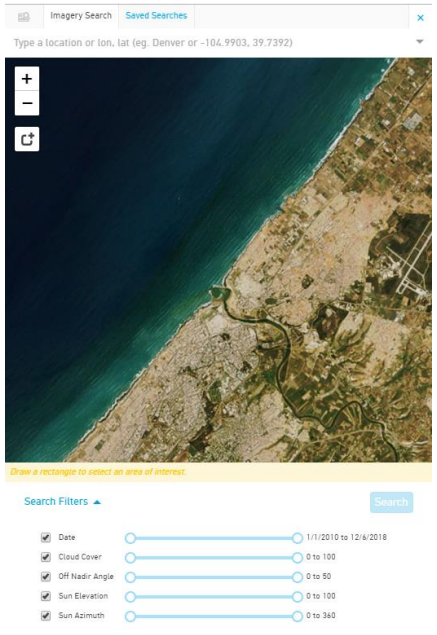


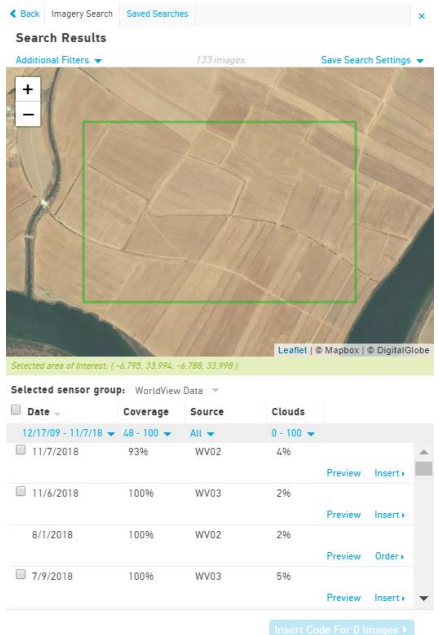
https://notebooks.geobigdata.io/

Image Searching



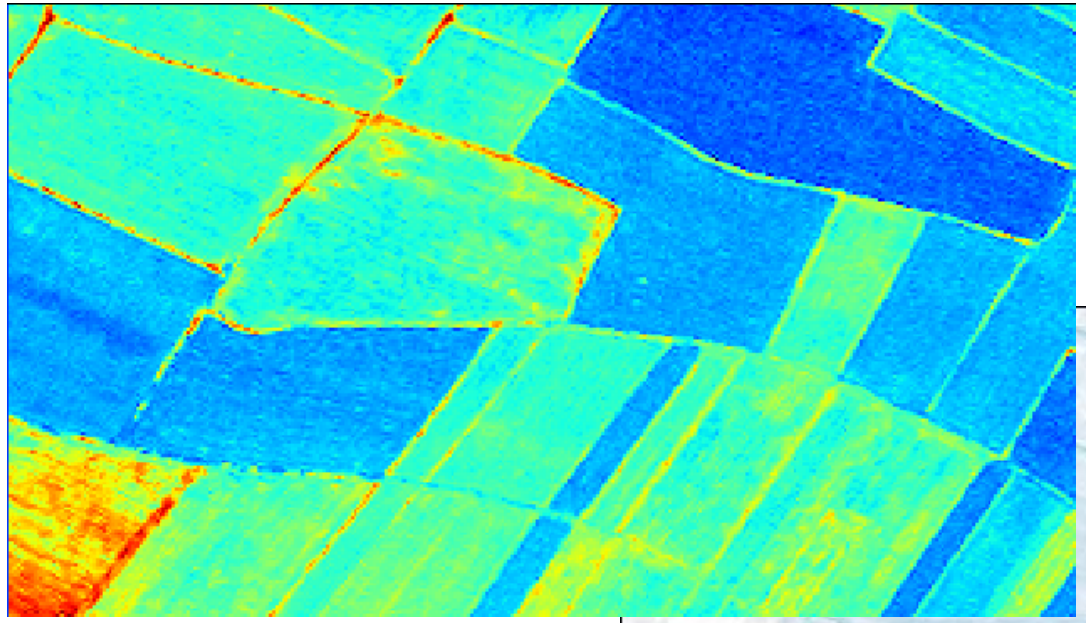


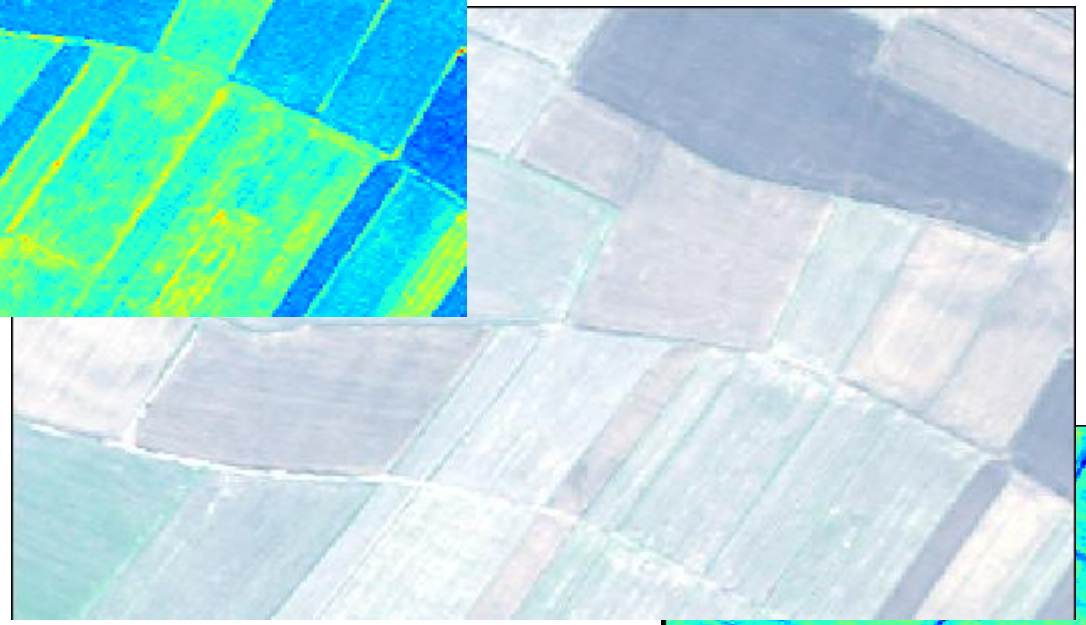


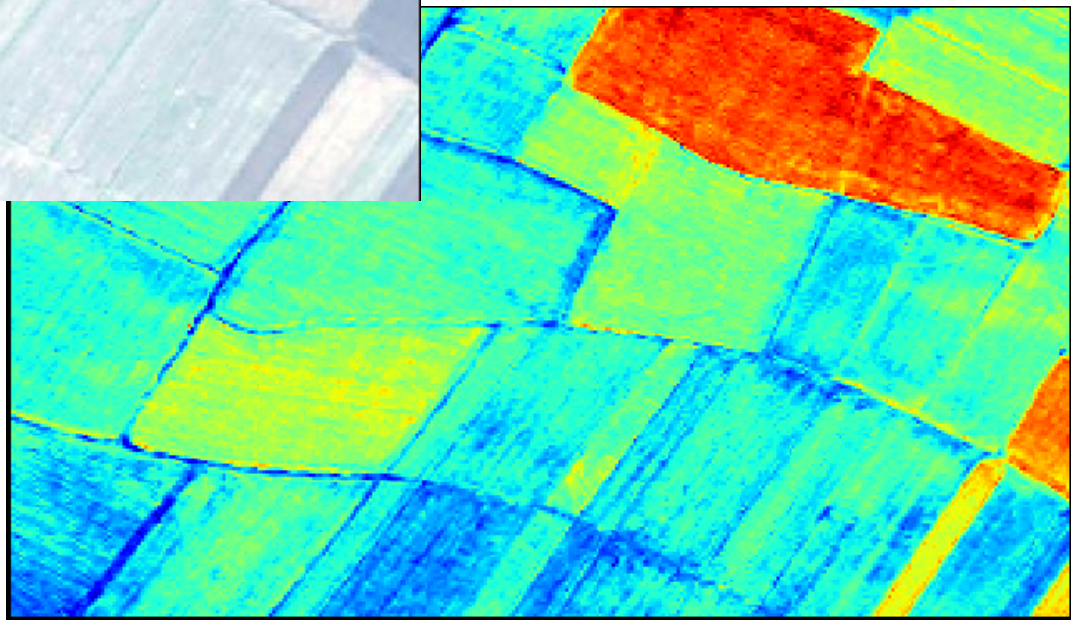



```
In [1]: from gbdxtools.task import env
        from gbdxtools import CatalogImage
        %matplotlib inline

        catalog_id = env.inputs.get('catalog_id', '1030010088B8E800')
        bbox = env.inputs.get('bbox', '-6.795043945639919, 33.99424688250682, -6.787877082497289, 33.99817847460809')

        image = CatalogImage(catalog_id, band_type="MS", bbox=map(float, bbox.split(",")))
        image.plot(w=10, h=10)
```

Case Study
Hands-on







aWhere

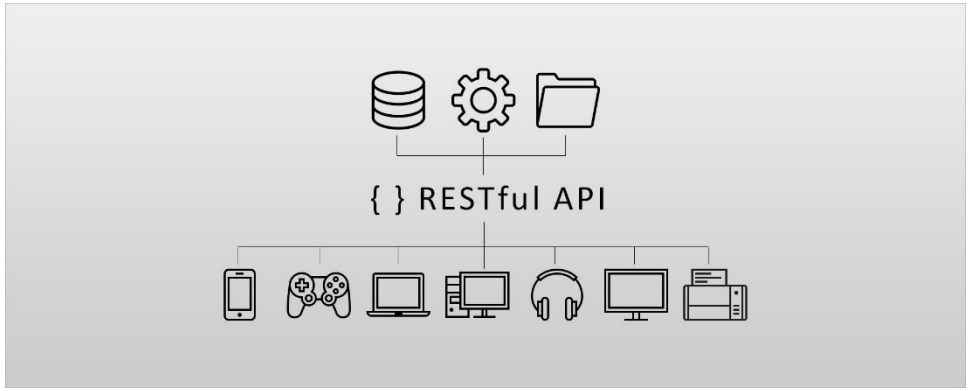
- Platform and Workflow
- R and RStudio
- Libraries and API fetching
- Case study
- Discussion

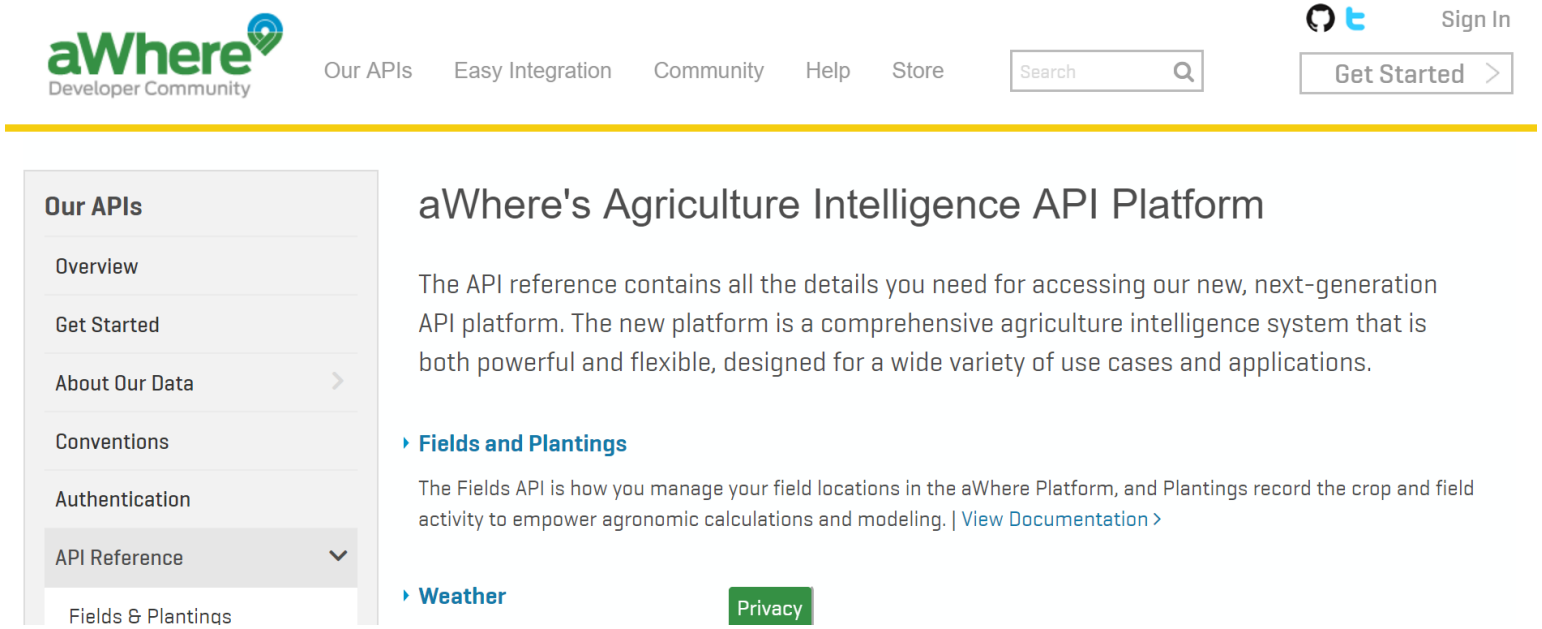
aWhere Services

aWhere provides Ag applications and big data insights.

RESTful APIs

aWhere's API platform is built as REST-based web services.



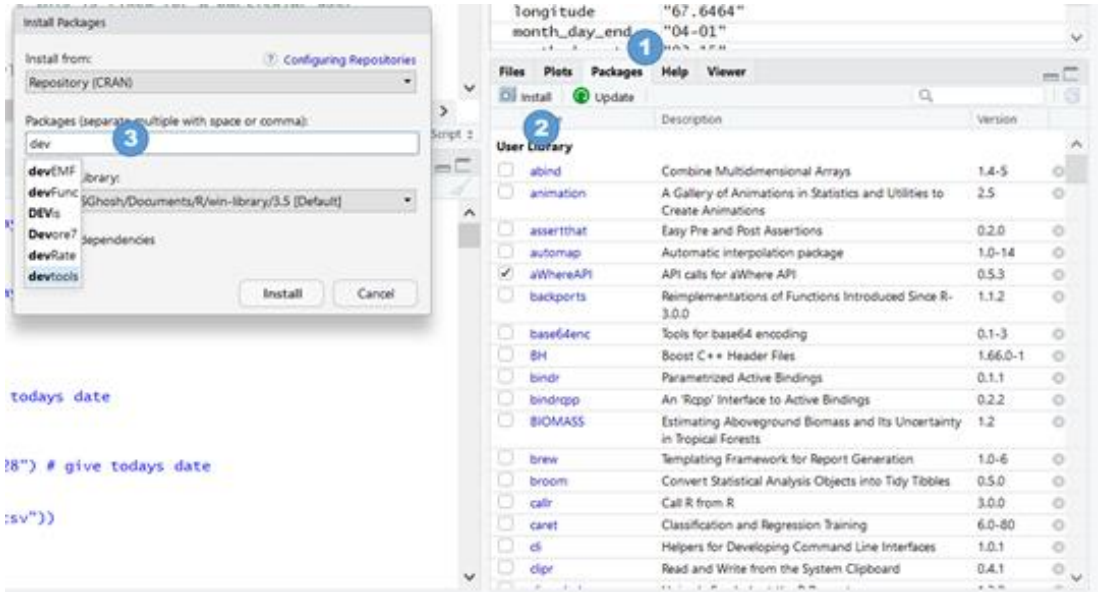


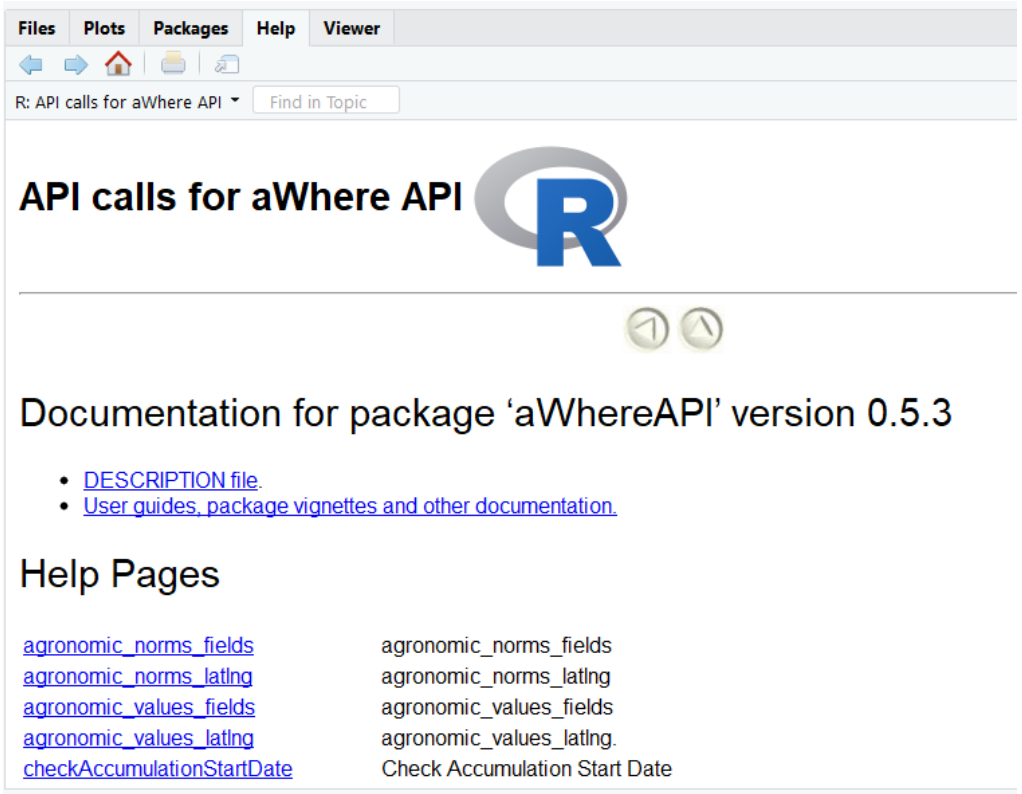
https://developer.awhere.com/api/reference

Installation aWhere API

Requirement: R and/or Rstudio

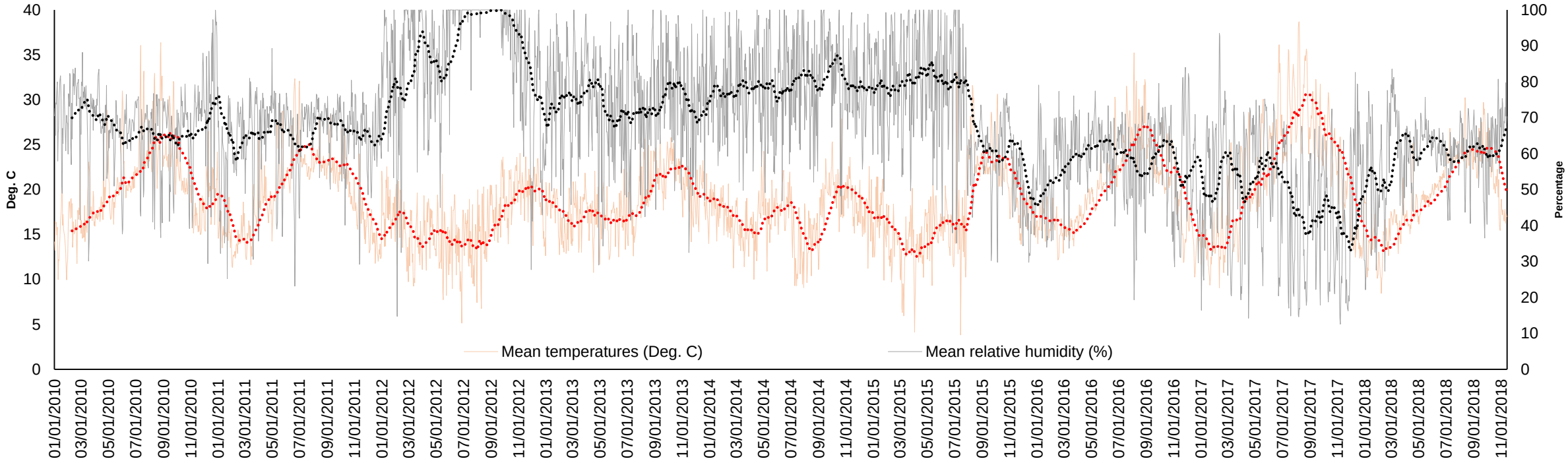
Install **devtools**

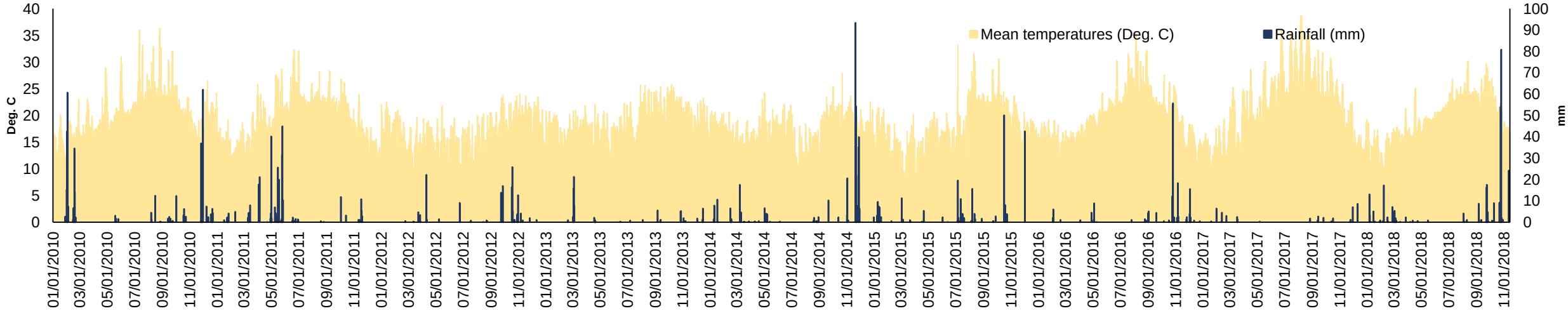




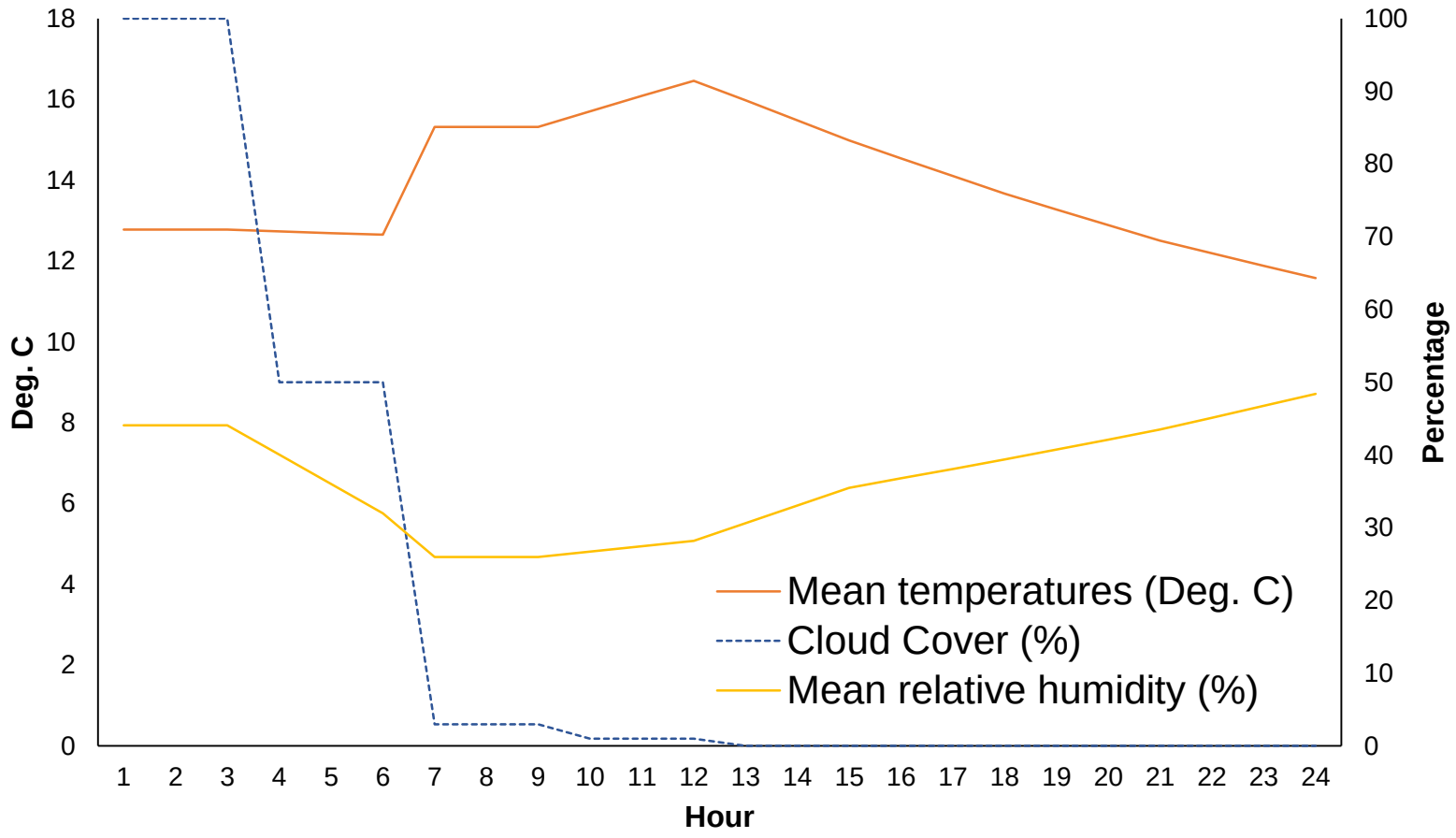
devtools::install_github("aWhereAPI/aWhere-R-Library")

Historical data (30.53,-9.34)



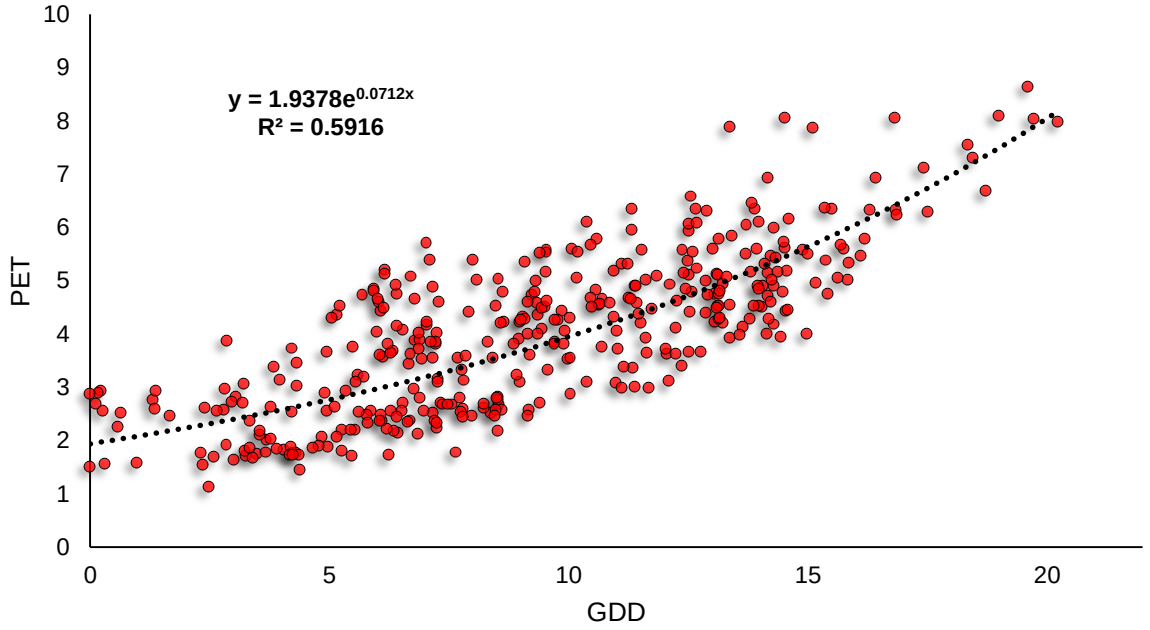


Now casting (15/11/2018)



Growing Degree-Days (GDDs)

Growing Degree-Days are a standard metric for ambient heat critical to plant growth.



Potential Evapotranspiration (PET)

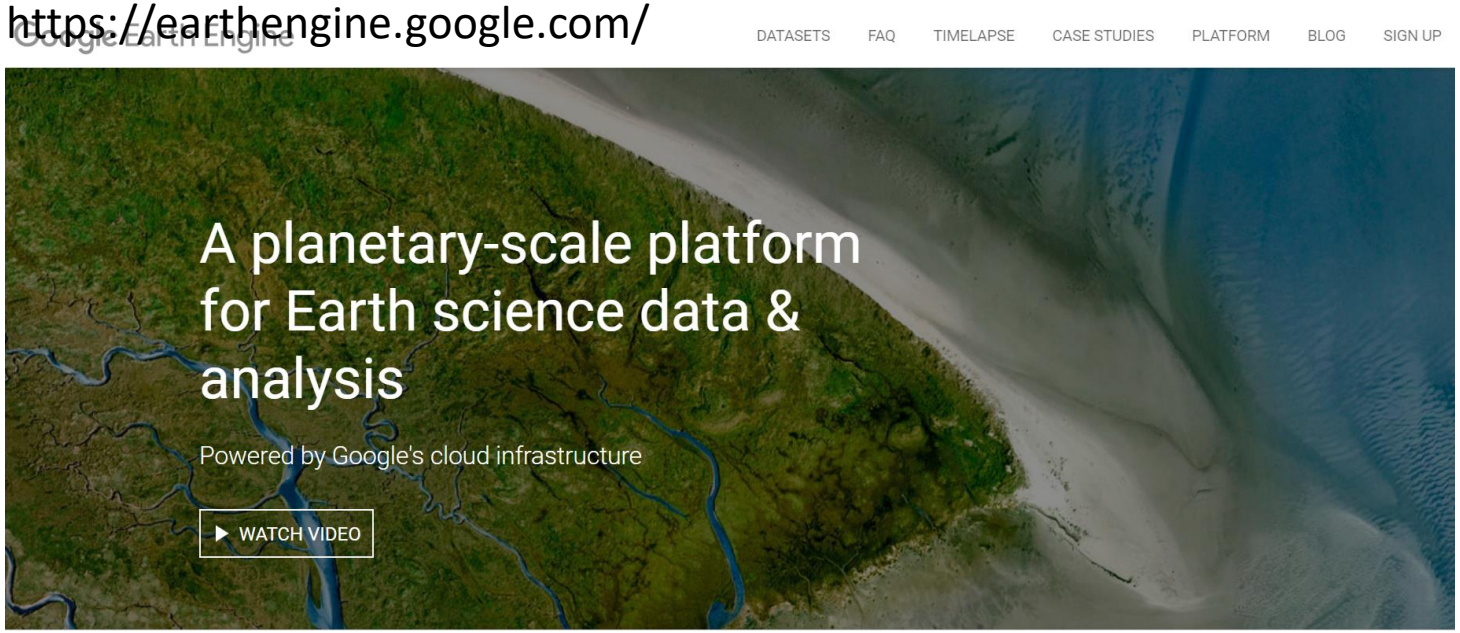
Potential Evapotranspiration is an metric in agronomy that can drive some powerful insights especially around irrigation decisions.

Case Study
Hands-on

- **Platform and Workflow**
- **Libraries and API fetching**
- **Case study**
- **Discussion**

Google Earth Engine

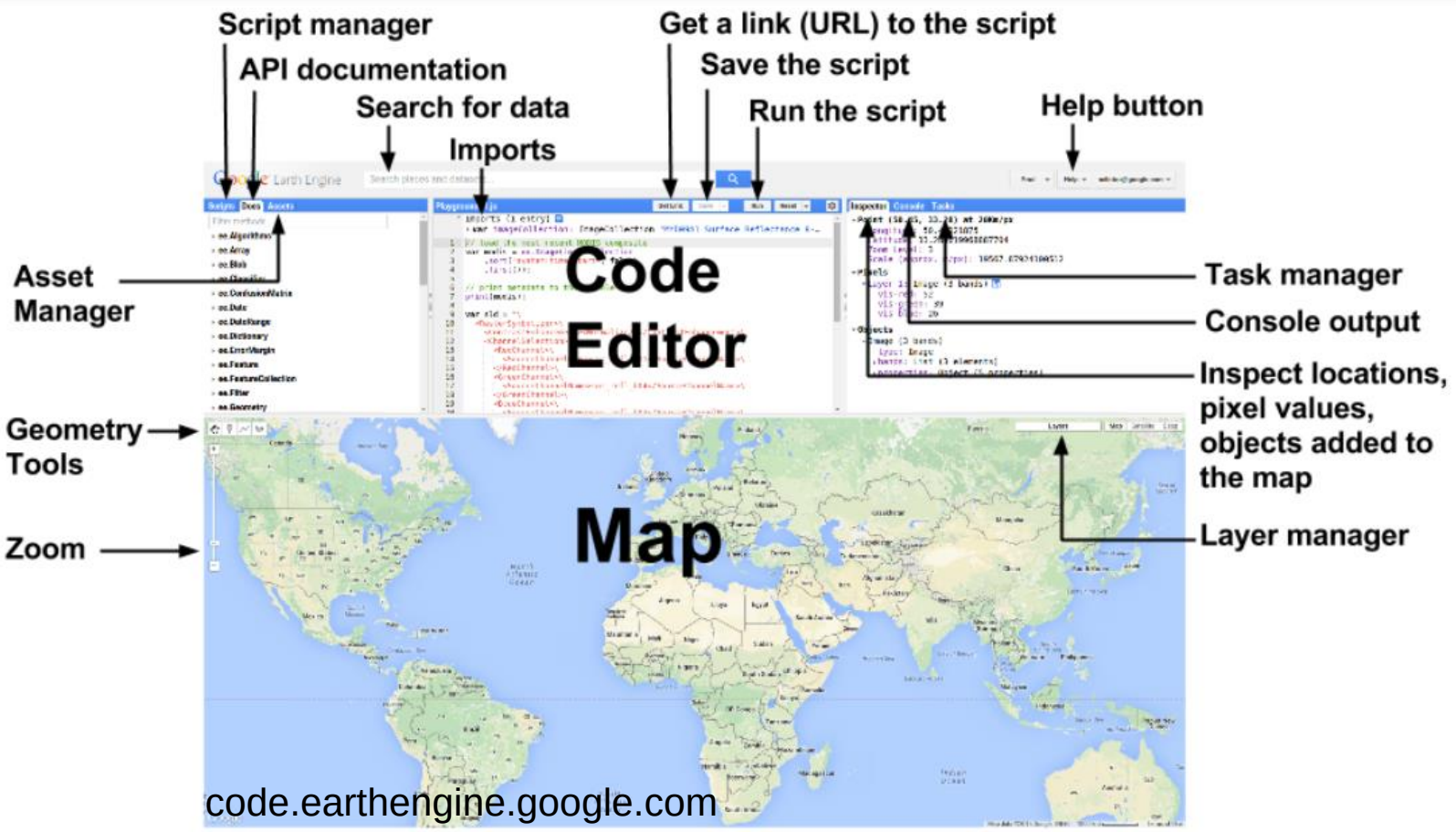
https://earthengine.google.com/



Google Earth Engine is a cloud-based platform for planetary-scale environmental data analysis.

- **Datasets:** A petabyte-scale archive of publicly available remotely sensed imagery and other data.
- **Compute power:** Google's computational infrastructure optimized for parallel processing of geospatial data.
- **APIs:** APIs for JavaScript and Python (hosted on GitHub) for making requests to the Earth Engine servers. These docs focus on JavaScript.
- **Code Editor:** An online Integrated Development Environment (IDE) for rapid prototyping and visualization of complex spatial analyses using the Javascript API.

Earth Engine Code Editor



- JavaScript code editor
- Map display for visualizing geospatial datasets
- API reference documentation (Docs tab)
- Git-based script manager (Scripts tab)
- Console output (Console tab)
- Task manager (Tasks tab) to handle long-running queries
- Interactive map query (Inspector tab)
- Search of the data archive or saved scripts
- Geometry drawing tools

Thank You

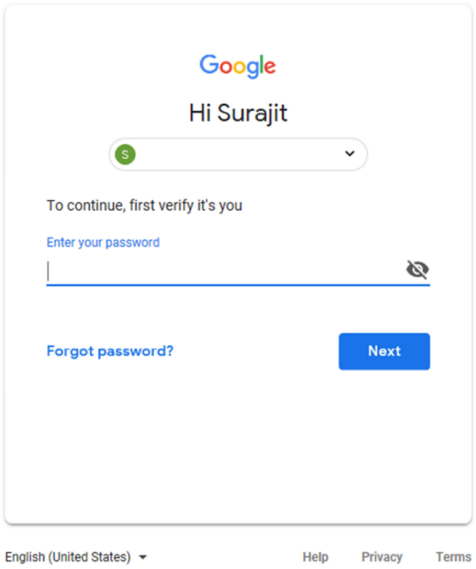
GEOAGRO ICARDA CGIAR

Sources of images used in this presentation

https://depositphotos.com/43981793/stock-photo-word-cloud-data-analysis.html
http://rkinsight.com/insights-discovery/
https://www.vutlan.com/blog/new-api-made-public-b34.html
https://bigdata.cgiar.org/
https://github.com/GeoBigData/gbdx-training/blob/master/gbdx_notebooks_module
https://www.youtube.com/watch?v=s7wmiS2mSXY
https://gbdxdocs.digitalglobe.com/docs/getting-started
https://developer.awhere.com/api/reference
https://earthengine.google.com/

Module 1. Step by step instructions for Google Earth Engine Setup

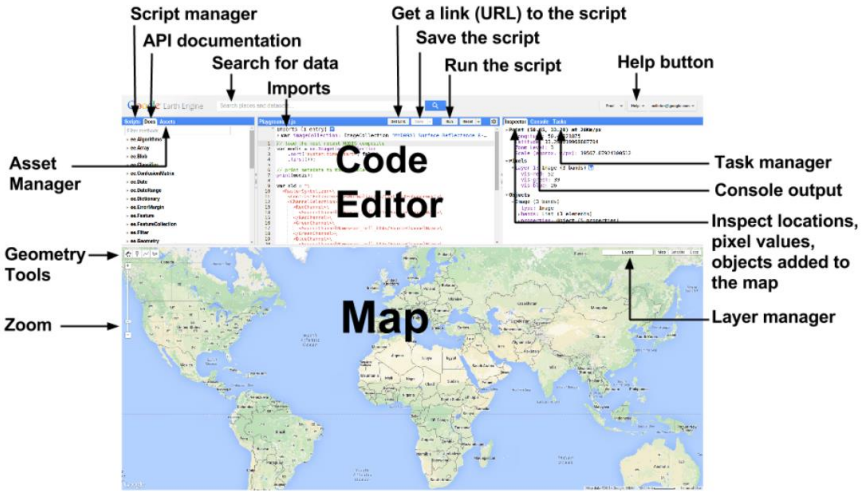
Step 1.
Signup for Earth Engine (https://signup.earthengine.google.com).



Once you have been accepted, you will receive an email with additional information.

Step 2.
Go through JavaScript for Earth Engine.
https://developers.google.com/earth-engine/tutorial_js_01

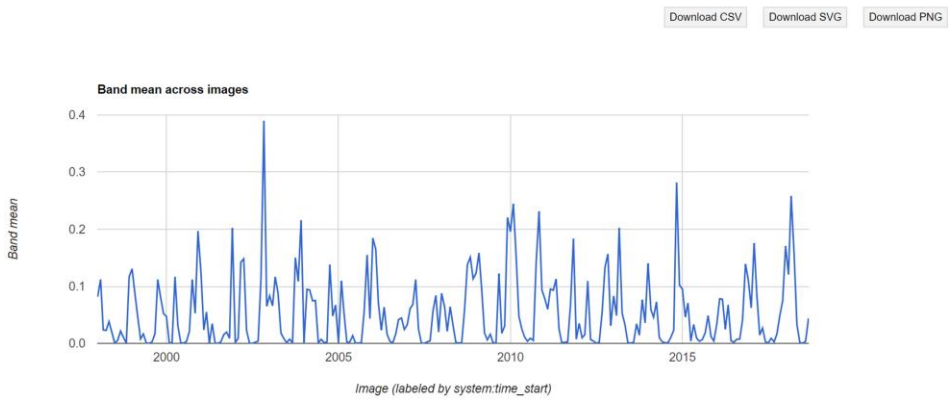
Step 3.
Explore Earth Engine Code Editor (the IDE for writing Earth Engine JavaScript code in a web browser)
https://code.earthengine.google.com/



Module 2. TRMM time series data analysis

```
//Author: GU, ICARDA

//https://developers.google.com/earth-engine/datasets/catalog/TRMM_3B43V7
var dataset = ee.ImageCollection('TRMM/3B43V7')   //call the data set
                .filter(ee.Filter.date('1998-01-01', '2018-12-31')); //set the time range


//precipitation  unit mm/hr
var precipitation = dataset.select('precipitation');
//Set you area of interest (region)
var geom = ee.FeatureCollection('users/point/Rabat');
//Set you area of interest (point with a buffer around a point.)
//var geom = ee.Geometry.Point(-6.8604577,33.9803603).buffer(2000); //(longitude, latitude)
format

Map.addLayer(ee.Image().paint(geom, 0, 2), {}, 'Area of Interest');
Map.setCenter(-6.8604577,33.9803603, 7);


// Create and print time series chart.
print(ui.Chart.image.series(precipitation, geom, ee.Reducer.mean()));

/*
var precipitationVis = {
  min: 0.1,
  max: 1.2,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};

Map.addLayer(precipitation, precipitationVis, 'Precipitation');
*/
```

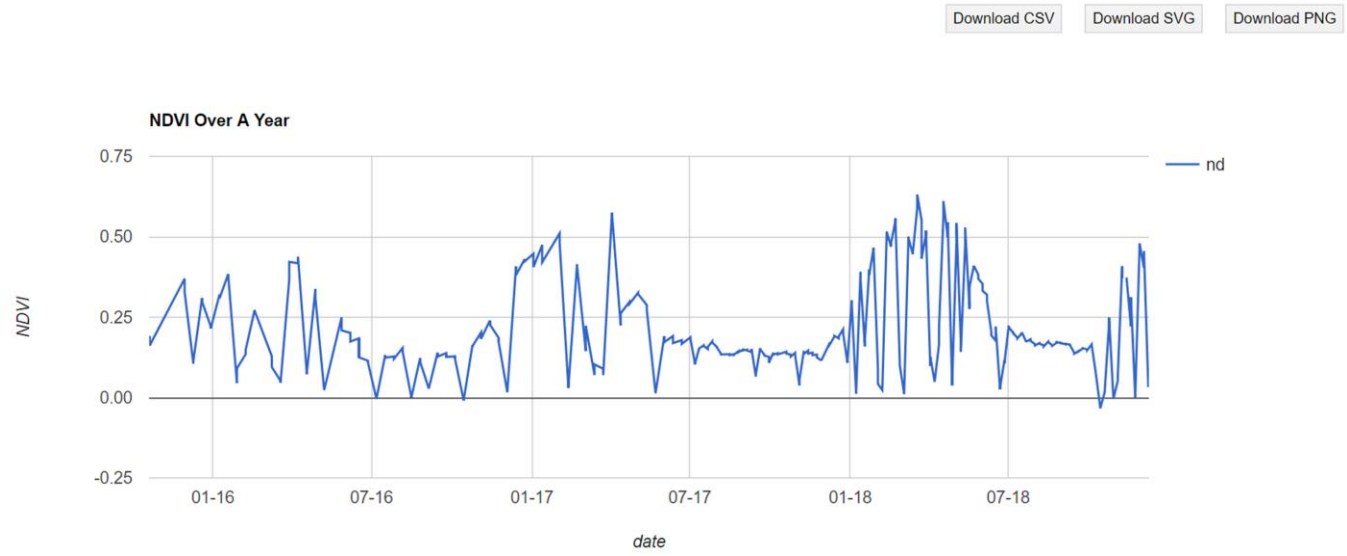
Output

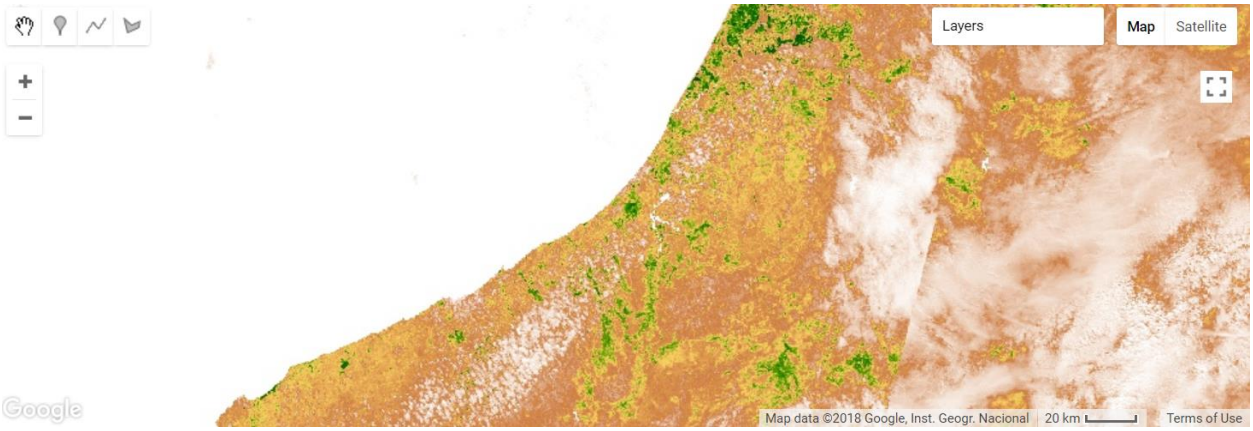


Module 3. Vegetation analysis

```
//Author: GU, ICARDA

//Getting the S2 image collection of a year
 var S2 = ee.ImageCollection('COPERNICUS/S2')
 .filterDate('2015-10-01', '2018-12-11'); //

//Computing the NDVI using band8 and band4
var ndvi = S2.map(function(image) {
  return image.select().addBands(image.normalizedDifference(['B8', 'B4']));
});

var vis = {min: 0, max: 1, palette: [
  'FFFFFF', 'CE7E45', 'FCD163', '66A000', '207401',
  '056201', '004C00', '023B01', '012E01', '011301'
]};

Map.addLayer(ndvi, vis, 'NDVI');
Map.setCenter(-6.8857609, 33.9167924, 8);

// Create a panel to hold our widgets.
var panel = ui.Panel();
panel.style().set('width', '300px');

// Create an intro panel with labels.
var intro = ui.Panel([
  ui.Label({
    value: 'Chart',
    style: {fontSize: '20px', fontWeight: 'bold'}
  }),
  ui.Label('Click on the map to generate chart')
]);
panel.add(intro);

var lon = ui.Label();
var lat = ui.Label();
panel.add(ui.Panel([lon, lat], ui.Panel.Layout.flow('horizontal')));

Map.onClick(function(coords) {
  lon.setValue('lon: ' + coords.lon.toFixed(2)),
  lat.setValue('lat: ' + coords.lat.toFixed(2));

  var point = ee.Geometry.Point(coords.lon, coords.lat);
  var dot = ui.Map.Layer(point, {color: 'FF0000'});
  Map.layers().set(1, dot);

  // NDVI chart.
  var ndviChart = ui.Chart.image.series(ndvi, point, ee.Reducer.mean(), 500);
  ndviChart.setOptions({
    title: 'NDVI Over A Year',
    vAxis: {title: 'NDVI'},
    hAxis: {title: 'date', format: 'MM-yy', gridlines: {count: 7}},
```

```
  });
  panel.widgets().set(2, ndviChart);

   //Get the NDVI time series from the clicked point region
   var y = ndvi.getRegion(point, 250).slice(1)
      .map(function(item) { return ee.Number(ee.List(item).get(-1)).divide(1e4) })

});

Map.style().set('cursor', 'crosshair');
// Add the panel to the ui.root.
ui.root.insert(0, panel);
```





Useful links

https://developers.google.com/earth-engine/
https://youtu.be/4E6yQLoGO2o
https://americaview.org/program-areas/education/google-earth-engine-tutorials/
https://geoscripting-wur.github.io/Earth_Engine/