



PUBLIC LECTURE

# Geospatial Big Data and Cloud Computing

Presenter: Dr. Eng. Serkan Girgin  
University of Twente, Faculty ITC  
[Center of Expertise in Big Geodata Science](#)

**FEBRUARY 2023**

# Geospatial data is getting **bigger and more difficult** to analyse

- Satellites, drones, vehicles, social networks, mobile devices, cameras, etc. generate **vast amount** of (open) geospatial data.
- Numerous methods and (open-source) applications have been developed to enable **discovery, delivery, analysis, and visualization** of geospatial data.
- However, large and complex geospatial data sets are difficult to handle using **conventional systems and methods**.
- Data processing and analysis tasks are **time consuming**, sometimes even not possible, if they are performed on laptops or local workstations.



# Solutions require expert **know-how and infrastructure**

- Local and regional studies with medium size data  
Analyses can be done faster by **parallel computing** on a workstation
- Machine learning and AI studies with medium size data  
Analyses require **special processing units** (e.g., GPU/TPU) due to computational complexity
- National, continental, and global studies with big data  
Analyses require **distributed computing** on a computing cluster due to computational complexity and/or large volume of data





**Cloud computing** is on-demand availability of computer system resources, especially **data storage** and **computing power**, without direct active management by the user

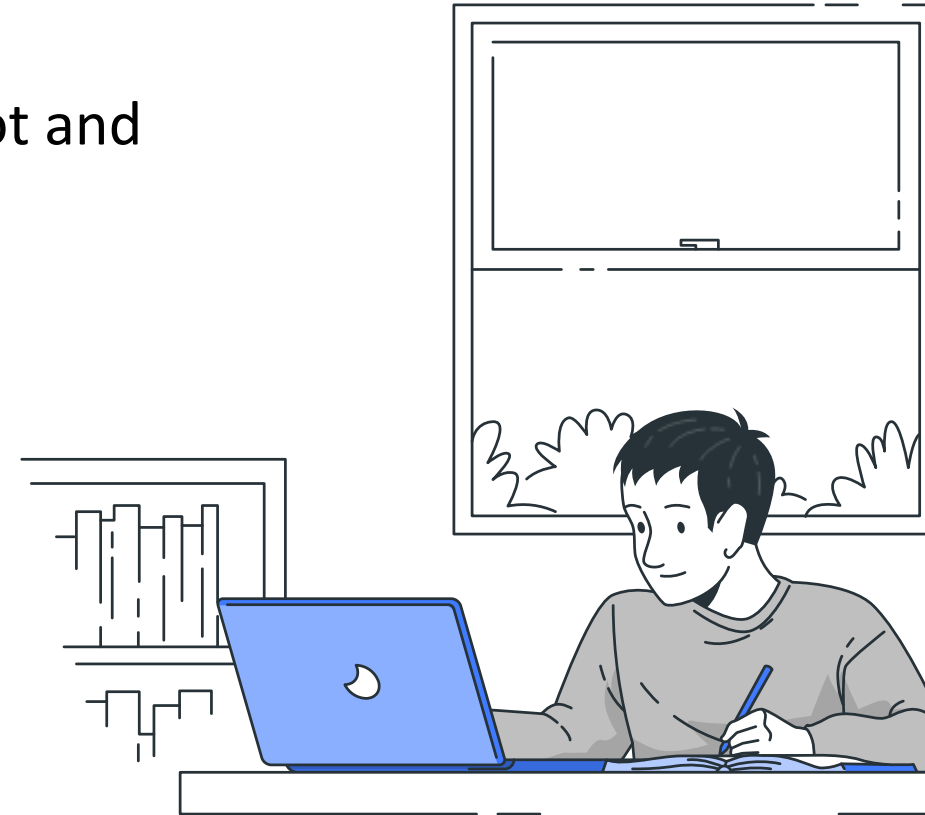
# Computing is **moving to the Cloud**, so is geocomputing

- Developments in infrastructure, both hardware and software, gave a **big push** to data processing and analysis capabilities.
- **Scalable and affordable** computing is available through:
  - Open-source systems that allow computing clusters on commodity hardware
  - Proprietary cloud-based data storage and computing services
- However, it is **challenging to choose** the right solution(s) depending on the nature of geospatial data and analysis needs.
- Using the solutions usually requires a transition in **modus operandi**.



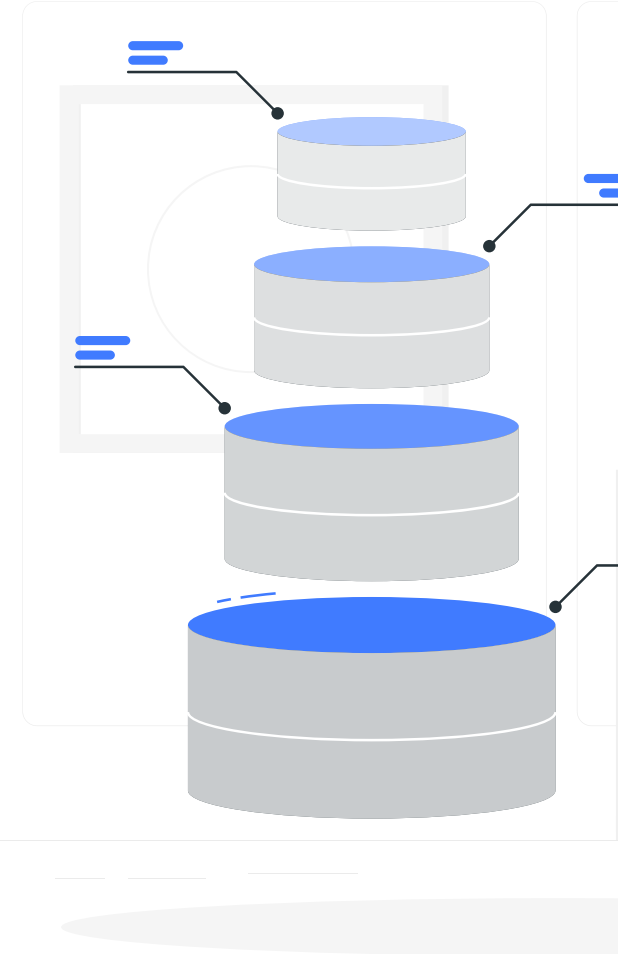
# Not everyone requires cloud computing and big data, but...

- Institutions are usually **heterogeneous** with respect to interests and needs.
- For some people cloud computing and big data are not and probably will **not be relevant or interesting**.
- Even if there is no apparent need or interest, it is still important to have at least a **basic understanding** of these topics, because they are becoming **key components** in the geospatial domain.
- This should be an **institutional priority**.



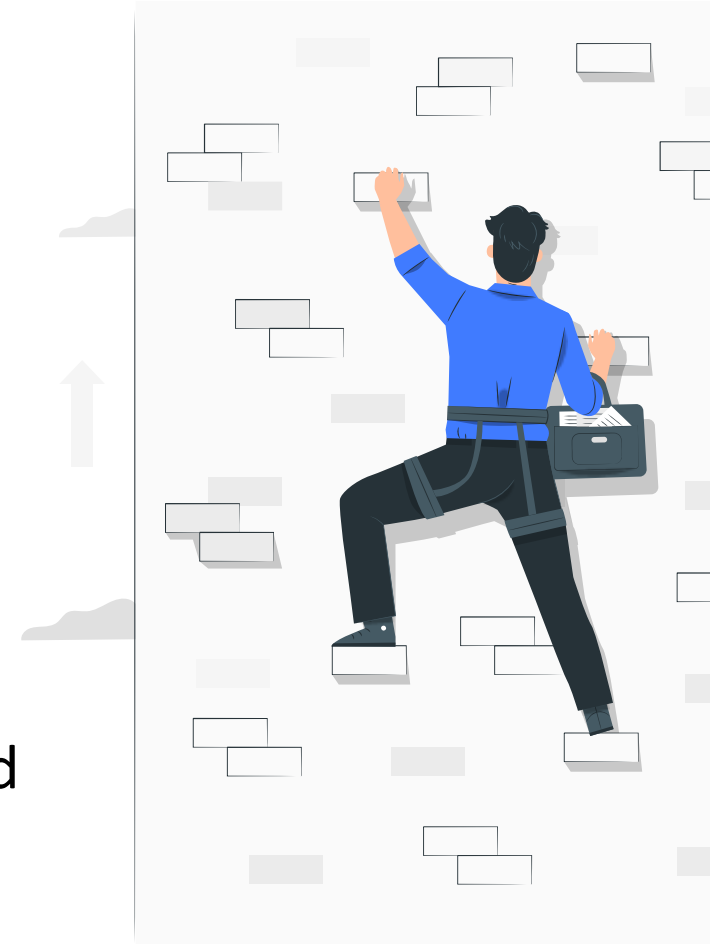
# Cloud computing has a few **distinctive features**

- **On-demand self-service:** provision of computing capabilities as needed without requiring human interaction.
- **Broad network access:** availability over the Internet with standard access mechanisms for different client platforms (e.g., tablets, laptops, mobile phones).
- **Resource pooling:** dynamic assignment and reassignment of physical and virtual resources according to consumer demand.
- **Rapid elasticity:** capability to scale rapidly outward and inward proportionate to consumer demand.
- **Measured service:** accurate monitoring, control, and reporting of resource and service utilization.



# The features sound nice, but **status quo** is far from ideal

- Existing experience is **not widespread**, and difficulties exist in identifying the cases where cloud computing **can play a role**.
- Challenges exist in proper **selection and efficient use** of cloud computing methods, tools, and services.
- Available platforms and services are little used mainly due to **high cost** and limited **domain-specific technical support**.
- There is a high interest in **getting training** on how to (better) use cloud computing technology.
- There is also interest in **learning how** the technology is applied to solve domain-specific problems (e.g., what others do?)





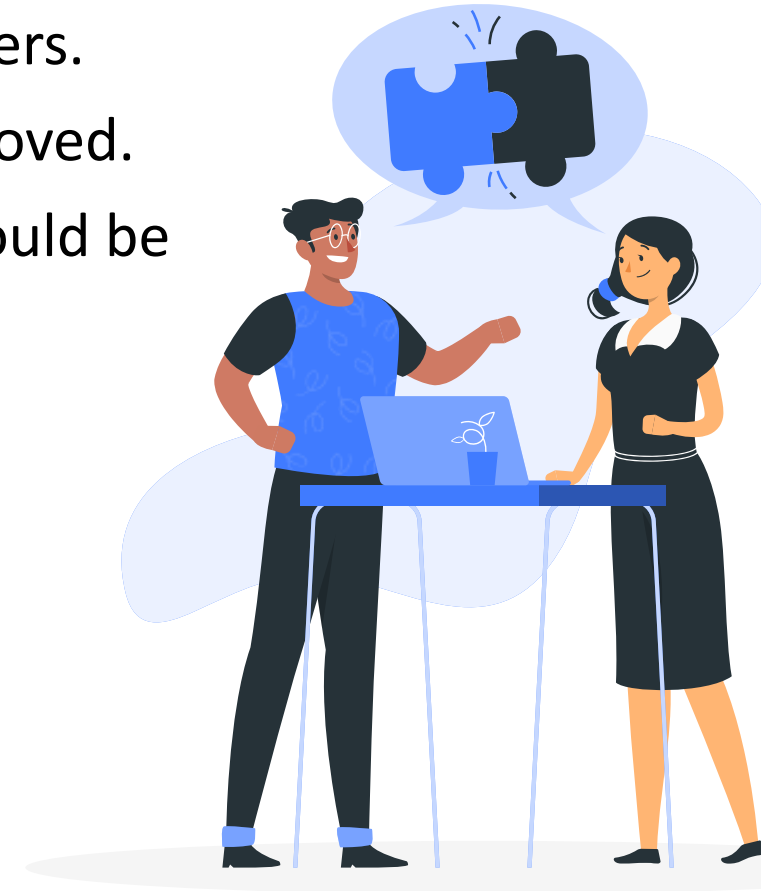
# The landscape is large and complex



## Principal needs are usually similar for the different user groups

- State-of-the-art should be **actively communicated** to the users.
- **Proficiency** of the users on cloud computing should be improved.
- Easy-to-use and efficient cloud computing infrastructure should be **made available** for training and work purposes.
- Workflows should be **enhanced and improved** with cloud computing technology where relevant.
- Ad hoc technical **support and advise** should be provided.
- Knowledge and good practices on better use of technology should be **transferred** between partner institutions.

**It is crucial to build a community that is self-motivated to learn, practice more, and share knowledge and experience!**



## **Rule One**

**Use the right tools!**

# Infrastructure as a Service (IaaS) – on demand virtual machines

- Provider supplies the infrastructure.
- User deploys and run arbitrary software, including operating system.
- Examples
  - [Amazon AWS](#)
  - [Microsoft Azure](#)
  - [Google Cloud](#)
  - [ESA DIASs](#)
  - National Research Clouds

**Low level:** Fine control on resources, custom system design, optimum performance, but difficult to manage, requires expertise!

# Platform as a Service (PaaS)

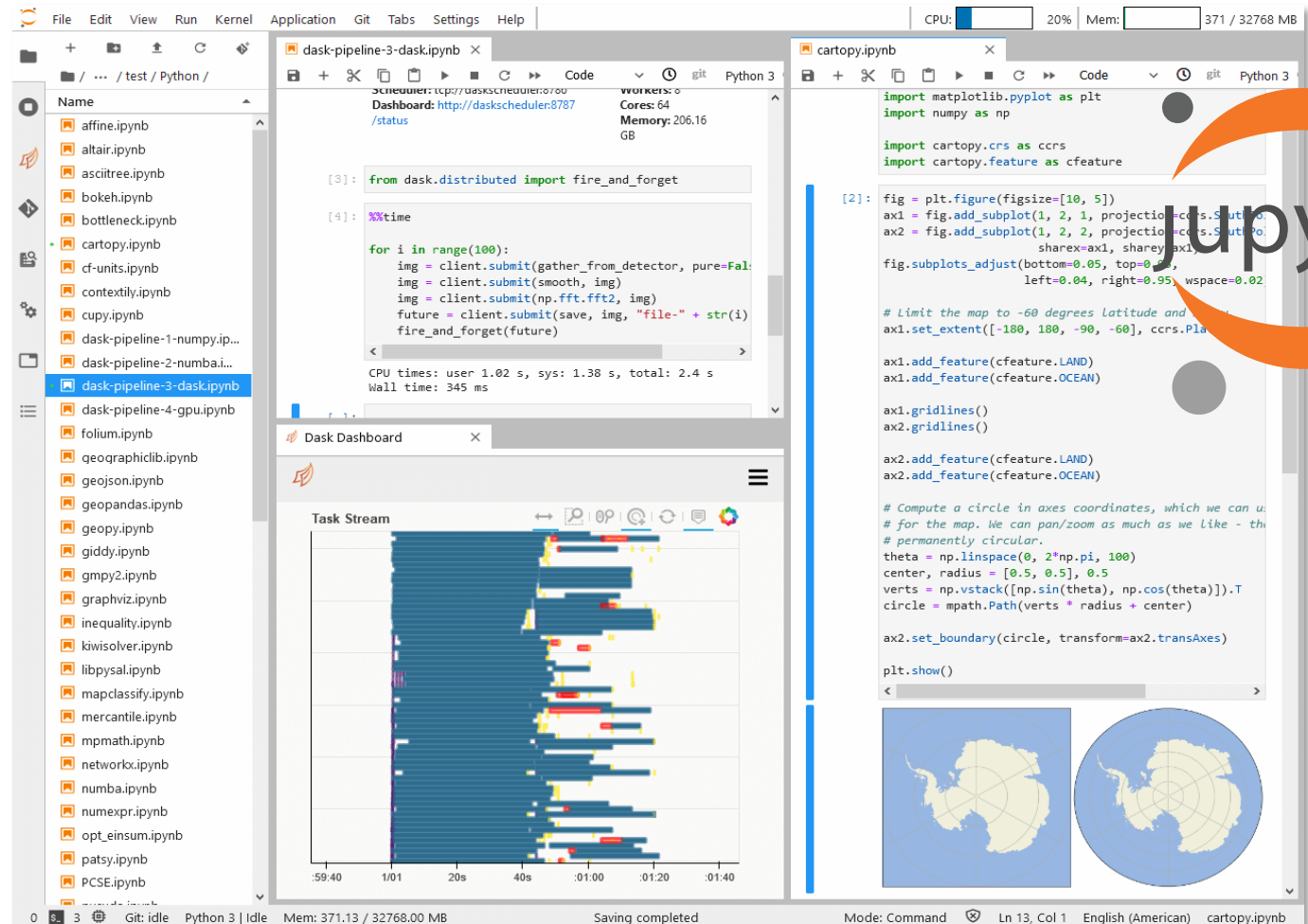
- Provider supplies the infrastructure, services, and tools that allow the user to deploy applications.
- User deploys applications and alters settings of the application hosting environment.
- Examples
  - [Google Earth Engine](#)
  - [Microsoft Planetary Computer](#)
  - [ITC Geospatial Computing Platform](#)
  - [Google Colab](#)
  - [Amazon SageMaker](#)

**Medium level:** Limited control on resources, custom workflow design, good performance, but requires programming skills!

# Project Jupyter is a gamechanger for interactive computing

Free software, open standards, and web services for interactive computing across various programming languages

[jupyter.org](https://jupyter.org)



The screenshot displays a JupyterLab environment with several open notebooks and a task stream. The top notebook, 'dask-pipeline-3-dask.ipynb', shows code for a distributed pipeline using Dask:

```

[3]: from dask.distributed import fire_and_forget
[4]: %%time
for i in range(100):
    img = client.submit(gather_from_detector, pure=False)
    img = client.submit(smooth, img)
    img = client.submit(np.fft.fft2, img)
    future = client.submit(save, img, "file-" + str(i))
    fire_and_forget(future)
    
```

The output shows CPU times: user 1.02 s, sys: 1.38 s, total: 2.4 s, and wall time: 345 ms. Below this is a 'Task Stream' visualization showing a Gantt chart of task execution over time.

The bottom notebook, 'cartopy.ipynb', shows code for creating a map with cartopy:

```

[2]: fig = plt.figure(figsize=[10, 5])
ax1 = fig.add_subplot(1, 2, 1, projection=ccrs.SouthPole)
ax2 = fig.add_subplot(1, 2, 2, projection=ccrs.SouthPole)
fig.subplots_adjust(bottom=0.05, top=0.95,
                    left=0.04, right=0.95, wspace=0.02)

# Limit the map to -60 degrees Latitude and 180 degrees Longitude
ax1.set_extent([-180, 180, -90, -60], ccrs.PlateCarree)

ax1.add_feature(cfeature.LAND)
ax1.add_feature(cfeature.OCEAN)

ax1.gridlines()
ax2.gridlines()

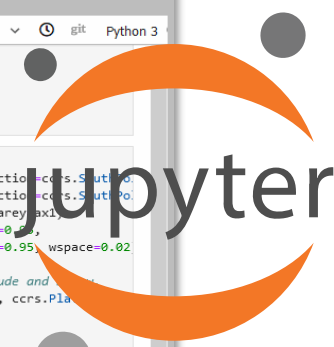
ax2.add_feature(cfeature.LAND)
ax2.add_feature(cfeature.OCEAN)

# Compute a circle in axes coordinates, which we can use
# for the map. We can pan/zoom as much as we like - the
# permanently circular.
theta = np.linspace(0, 2*np.pi, 100)
center, radius = [0.5, 0.5], 0.5
verts = np.vstack([np.sin(theta), np.cos(theta)]).T
circle = mpath.Path(verts * radius + center)

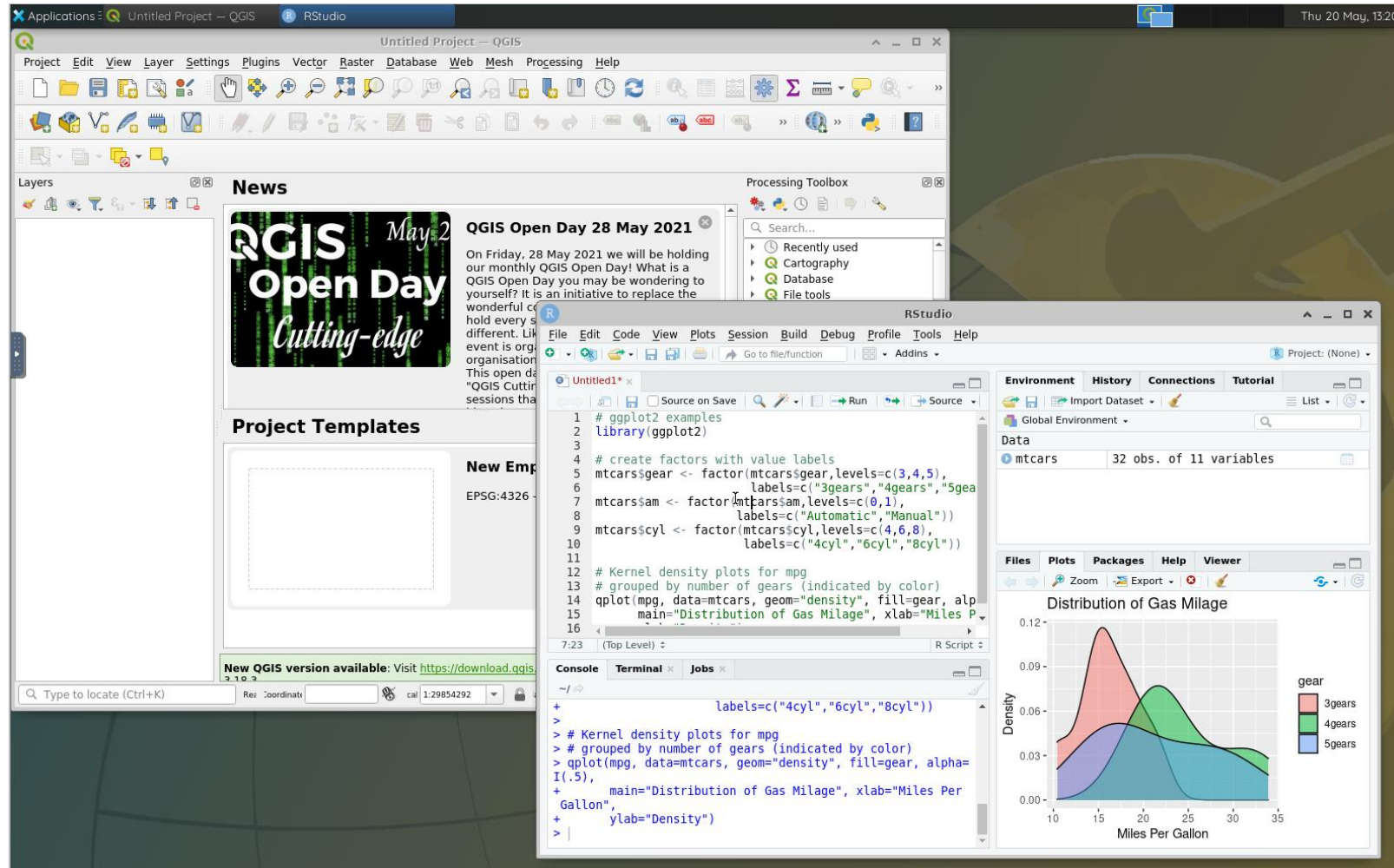
ax2.set_boundary(circle, transform=ax2.transAxes)

plt.show()
    
```

The output shows two maps of Antarctica: one in a rectangular projection and one in a circular projection with a boundary.



# Remote desktop connection allows conventional access



The screenshot displays a remote desktop environment with two primary applications: QGIS and RStudio.

**QGIS Interface:** The top window shows the QGIS desktop environment. The menu bar includes Project, Edit, View, Layer, Settings, Plugins, Vector, Raster, Database, Web, Mesh, Processing, and Help. The Layers panel on the left is empty. The main canvas displays a news article titled "QGIS Open Day 28 May 2021" with a "Cutting-edge" graphic. The Processing Toolbox on the right lists "Recently used" tools: Cartography, Database, and File tools.

**RStudio Interface:** The bottom window shows the RStudio IDE. The script editor contains the following R code:

```

1 # ggplot2 examples
2 library(ggplot2)
3
4 # create factors with value labels
5 mtcars$gear <- factor(mtcars$gear, levels=c(3,4,5),
6 labels=c("3gears", "4gears", "5gears"))
7 mtcars$am <- factor(mtcars$am, levels=c(0,1),
8 labels=c("Automatic", "Manual"))
9 mtcars$scyl <- factor(mtcars$scyl, levels=c(4,6,8),
10 labels=c("4cyl", "6cyl", "8cyl"))
11
12 # Kernel density plots for mpg
13 # grouped by number of gears (indicated by color)
14 qplot(mpg, data=mtcars, geom="density", fill=gear, alpha=I(.5),
15 main="Distribution of Gas Milage", xlab="Miles Per Gallon", ylab="Density")
16

```

The Environment pane on the right shows the loaded data frame: "mtcars" with 32 observations and 11 variables. The Plots pane displays a kernel density plot titled "Distribution of Gas Milage" showing the density distribution of Miles Per Gallon (MPG) for three gear types: 3gears (red), 4gears (green), and 5gears (blue). The x-axis is labeled "Miles Per Gallon" and ranges from 10 to 35. The y-axis is labeled "Density" and ranges from 0.00 to 0.12.

# Software as a Service (SaaS) – on demand application software

- Provider supplies the infrastructure that run the applications.
- User uses provided applications through an interface.
- Examples
  - [ArcGIS Online](#)
  - [CartoDB](#)
  - [Mapbox](#)
  - [R Studio Cloud](#)

**High level:** Easy to use, (usually) optimum performance, but no control on resources, usually paid!



# There are also many **other ..aaSs!**

- Function as a service (**FaaS**)
- Data as a service (**DaaS**)
- Data Processing as a service (**DPaaS**)
- ...



# There are also many cloud **service providers!**



- Common features

- Virtual machines
- Cloud storage
- Open-source software
- Open datasets

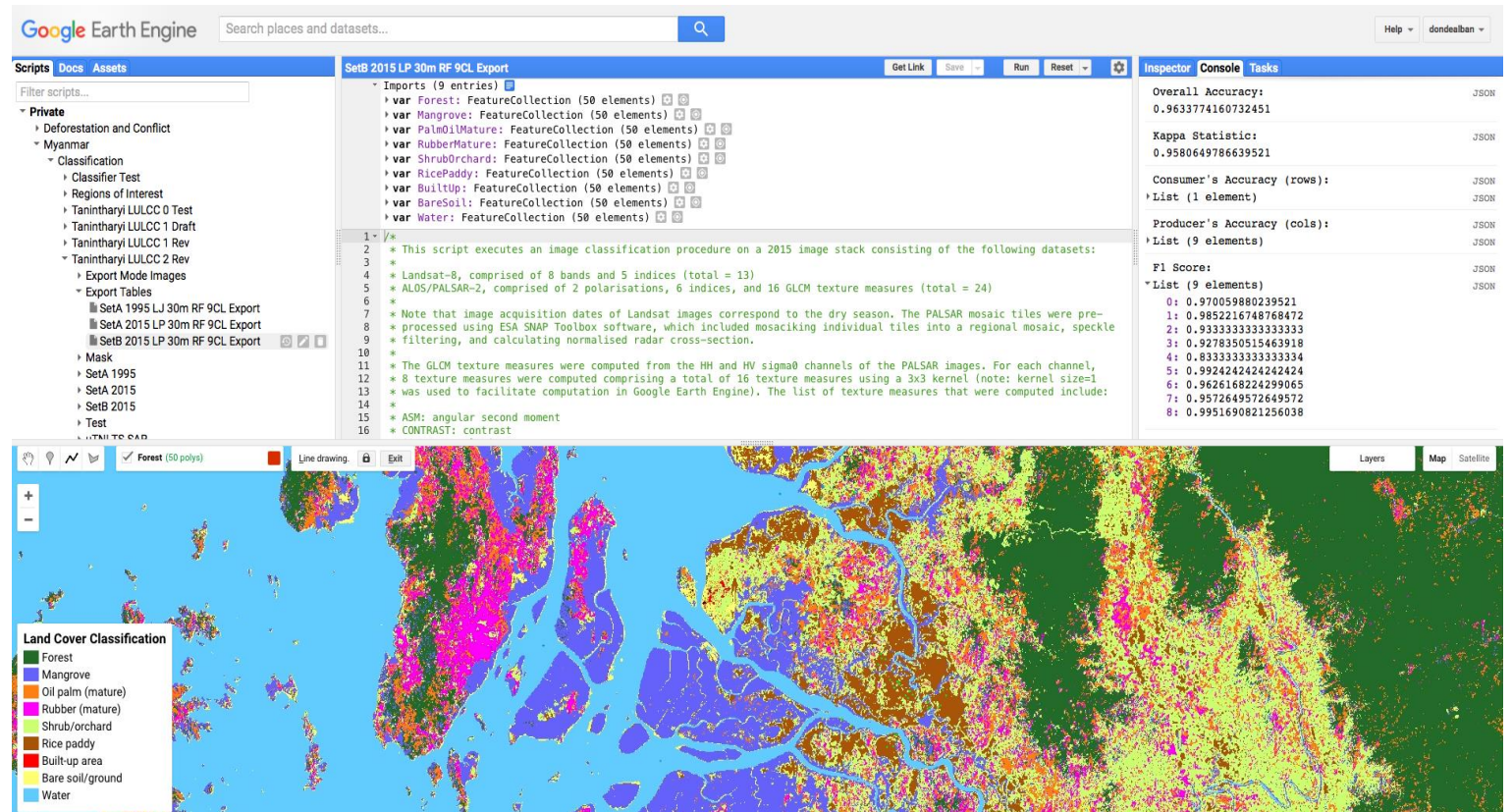
- Different features

- Azure Machine Learning Platform  
Cloud-based environment to train, deploy, automate, manage ML models
- Azure Data Science Virtual Machines  
Geo AI Data Science VM with ArcGIS
- EMR Cloud-native Big Data Platform  
EC2 + S3 clusters without provisioning, with OSS (Hadoop, Spark, etc.)
- Google Compute Engine  
Cloud TPU (eg. ResNet-50, 90 ep.: 8 V100 GPU: 216 min, Cloud TPU V2: 7.9 min)
- BigQuery  
BigQuery ML: create and execute ML models using standard SQL  
BigQuery GIS: analyze and visualize geodata by using standard SQL

# Google Earth Engine is a gamechanger for geospatial computing

Combination of a multi-petabyte catalog of EO imagery and geospatial datasets with planetary-scale analysis capabilities available for free\*.

[earthengine.google.com](http://earthengine.google.com)



# Geocomputing on **local cloud** can be efficient and cost effective

- **ITC Geospatial Computing Platform** provides GPU-enabled general purpose (8 vCPU, 32 GB RAM) and big data (72 vCPU, 768 GB RAM) units with large storage, analysis ready data, ready-to-use interactive and desktop software (1500+ packages), and shared workspaces.
- Currently serves **850+ registered users**.
- Provided **225,000+ hours** of computing since January 2021.
- Already returned **15+ times** the investment costs.
- Monthly cost is **< 200 Euro**.

The platform has also been used by IDEAMAP SUDAN  
<https://crib.utwente.nl>



# Overall, cloud computing has **many benefits**

- Better computing **infrastructure** (e.g., more CPUs, GPUs, RAM)
- Better **storage** (e.g., large, replicated)
- Better **scalability** (e.g., more resources on-demand)
- Improved workflow **performance** due to co-location of data and computing (i.e., no download)
- Improved **collaboration** (e.g., direct access to same assets)
- Improved **resource utilization** (e.g., less idle time)
- **No cost** for investment and maintenance (if remote cloud)
- **Low cost** for extensive use (if local cloud)

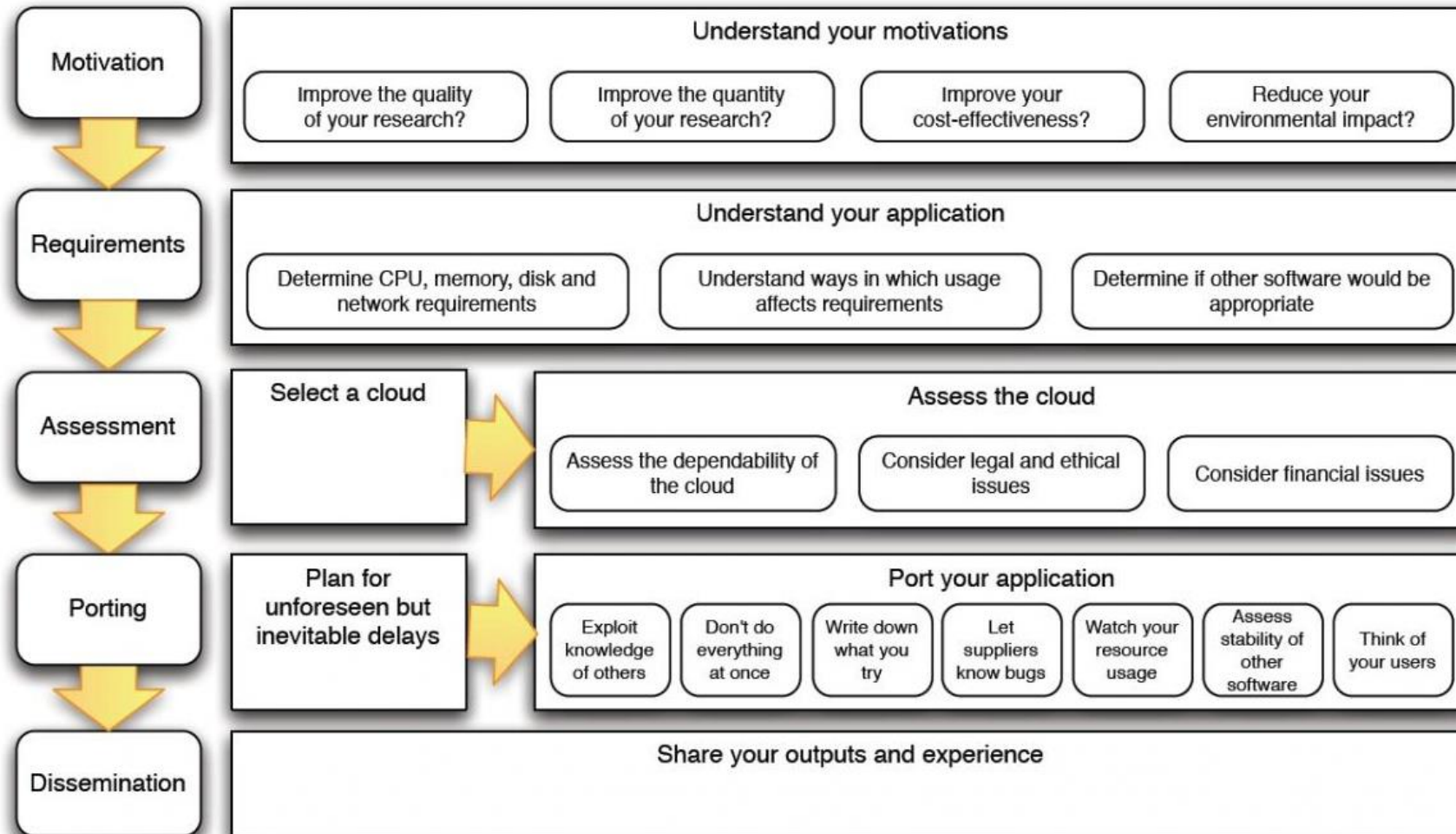


## A few **suggestions** for newcomers

- **Ensure familiarity** with the cloud computing technology through short talks and lectures.
- **Improve know-how** by participating tool- and technology-specific training
- **Try and use** the infrastructure and platforms available for free or through partner organizations.
- **Follow** a hybrid approach (local + cloud) to maximize the benefits.
- **Ask for technical and scientific support** for better implementation and integration of the technology.
- **Ask for guidance** for the planning of future activities.
- **Share your knowledge** and good practices with your colleagues (e.g., for cost-effective and efficient use).



# Following **best practices** facilitates moving to the Cloud



# Subscribe to our newsletter to stay informed!

## Big Geodata Newsletter

<https://itc.nl/big-geodata/newsletter/>



**Subscribe Now**





## Follow us to stay informed!



<https://itc.nl/big-geodata>



[crib-itc@utwente.nl](mailto:crib-itc@utwente.nl)

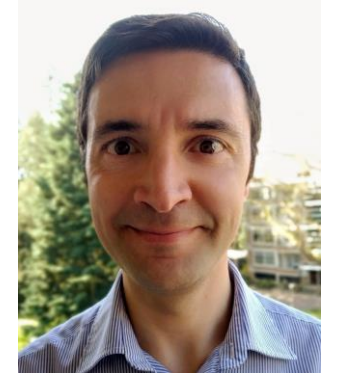


[@BigGeodata](https://twitter.com/BigGeodata)



[CRIB YouTube Channel](#)

## Contact



dr.ing. Serkan Girgin MSc  
Senior Researcher  
Head of Center of Expertise in  
Big Geodata Science  
[s.girgin@utwente.nl](mailto:s.girgin@utwente.nl)  
+31 53 489 55 78