Data Exchange and Management in INTAROS
using the integrated Arctic Observation System (iAOS)

INTAROS-MARIS Meeting
Nansen Environmental and Remote Sensing Center (NERSC)

30 October 2019, Bergen

Torill Hamre, NERSC
Contributions from INTAROS partners Terradue Srl, ARMINES and IMR

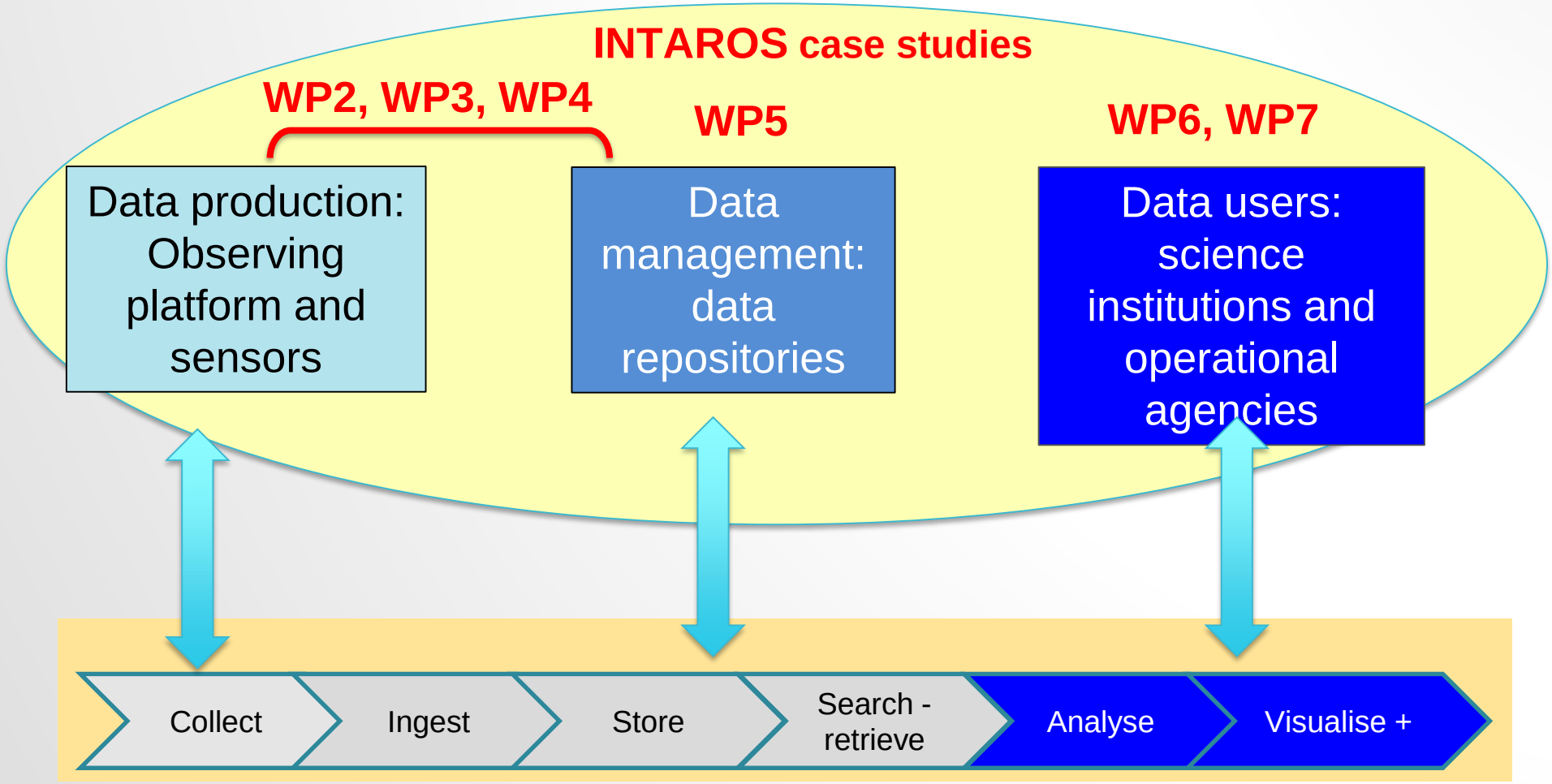
Agenda

- Objective of iAOS
- iAOS Cloud Platform
- Service development environment
- Examples of services
- Summary

Objectives of iAOS

Objectives of iAOS

Integrate multidisciplinary and distributed data repositories; offer tools to develop and deploy services for processing, analysis, visualization of multi-source data



Terradue Srl

iAOS Cloud Platform

iAOS Cloud Platform Objectives

- **Integrate data repositories** (*multidisciplinary* and *distributed*) into a scalable and resilient integrated Arctic observation system (iAOS)
  - Connect to observations and derived parameters together with EO data services (from partners, national and international data centres)
- **Develop processing services** for sea ice statistics, for integrated acoustics-remote sensing data analysis, and other geostatistics
  - Integrate a set of tools for data analysis, transformation and visualization
  - Support geostatistical methods for interpolation of spatiotemporal datasets
- **Support processing campaigns** of new observations from WP2-4
  - Enable users to run processing “within iAOS” (using iAOS-funded Cloud resources)
  - Store generated datasets in an iAOS-enabled repository

INTAROS

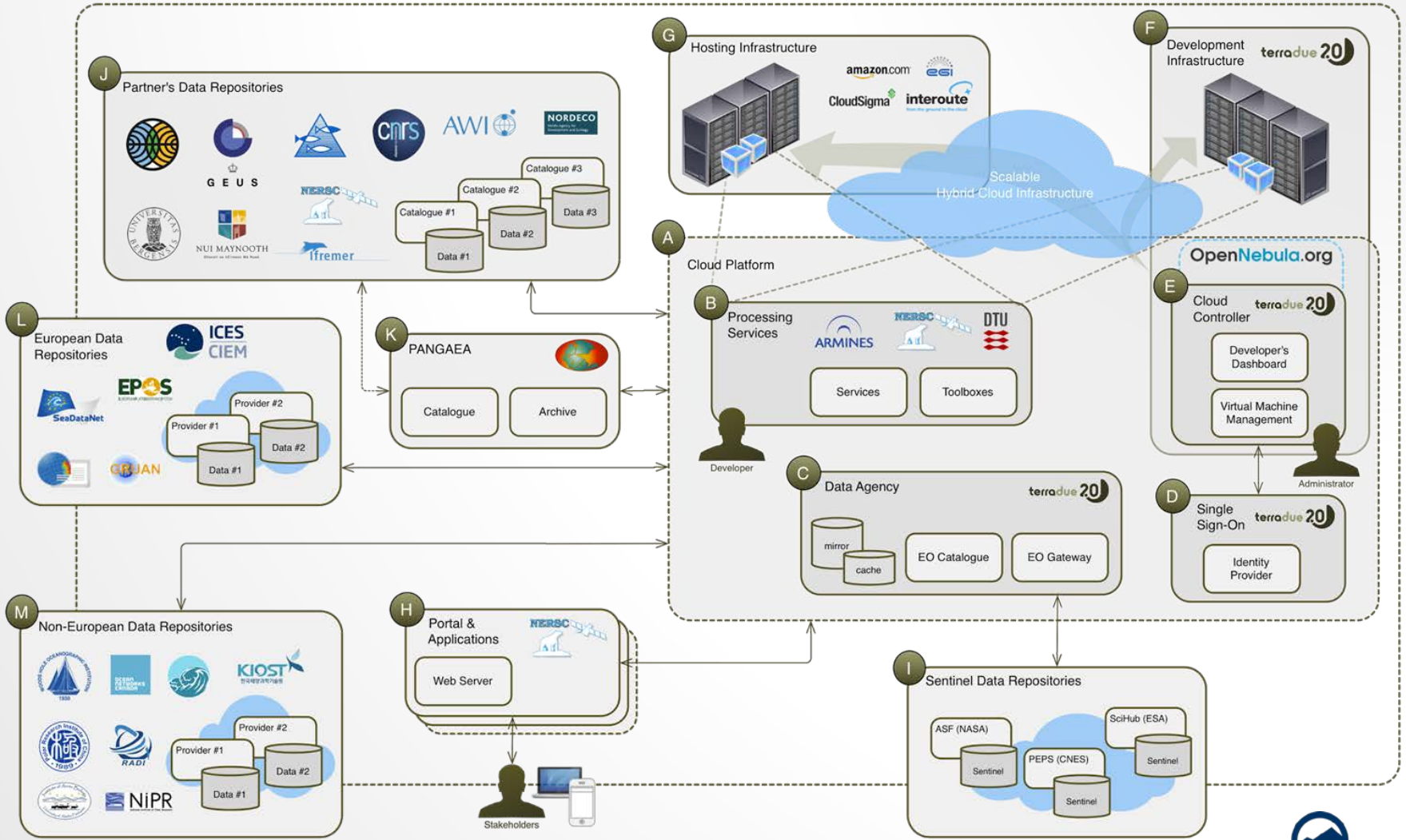
iAOS Cloud Platform

iAOS Platform Architecture

Cloud Platform is based on OpenNebula and development in several EC and ESA projects

Access to Sentinel data
Single sign-on
Service development
Deployment in several cloud infrastructures (e.g. EGI, AWS)

INTAROS integrates a set of Arctic data repositories



Terradue Srl

Service development environment

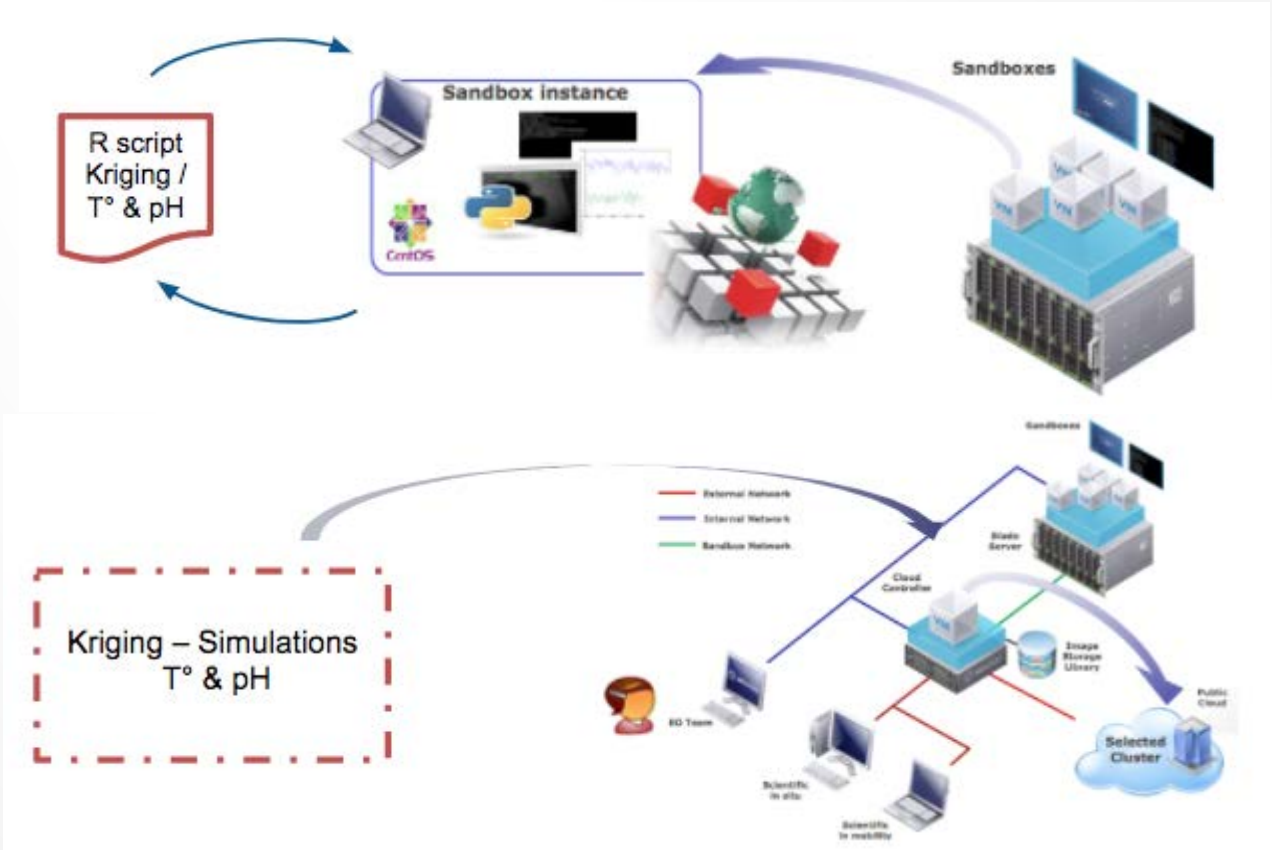
Service development environment

Cloud Sandbox:

- Job design: R/Python Script (data wrappers) integration on a Cloud Sandbox instance
  - Command Line Tools
  - Jupyter Notebook

Production environment:

- Job deployment: service scaling on a Production environment
  - Output as files on Cloud Storage



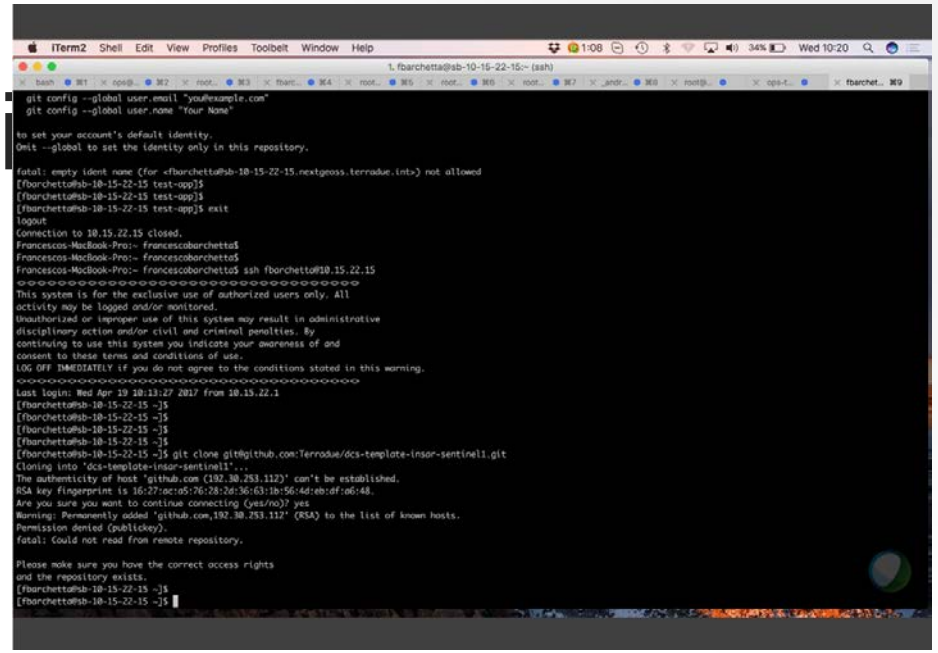
Service development envi

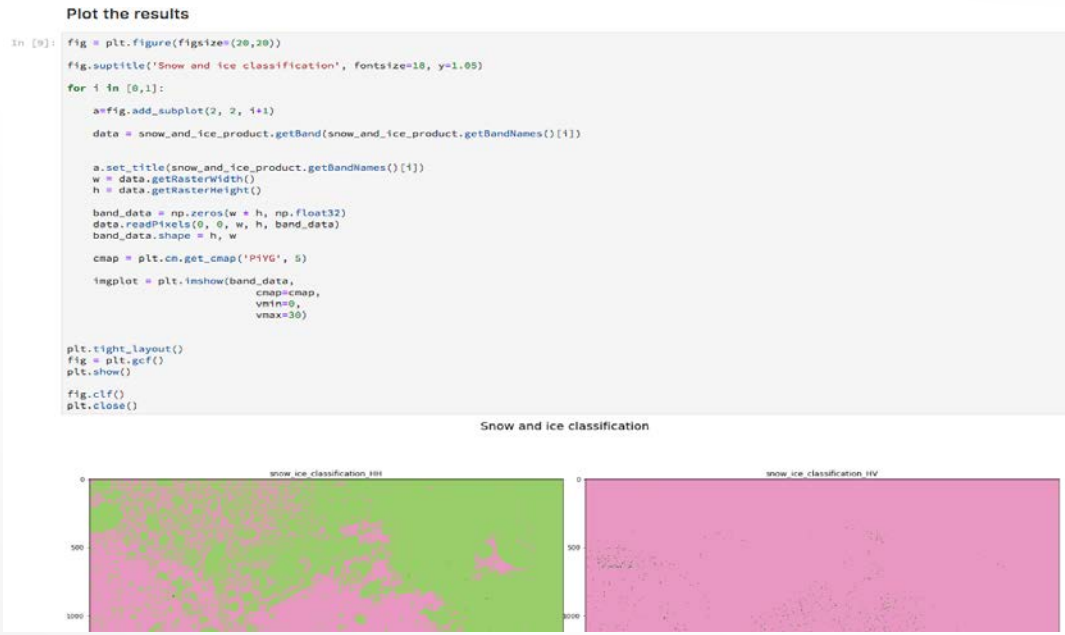
Linux Command Line Tool environment:

- Standard install commands in own VMs
- Code in git repos, maven to build
- Workflow monitoring, error logs
- RPM build and test for deployment

Notebook environment:

- Based on Jupyter Notebook
- Code in git repos
- Interactive testing of scripts and output
- Must be packaged for deployment





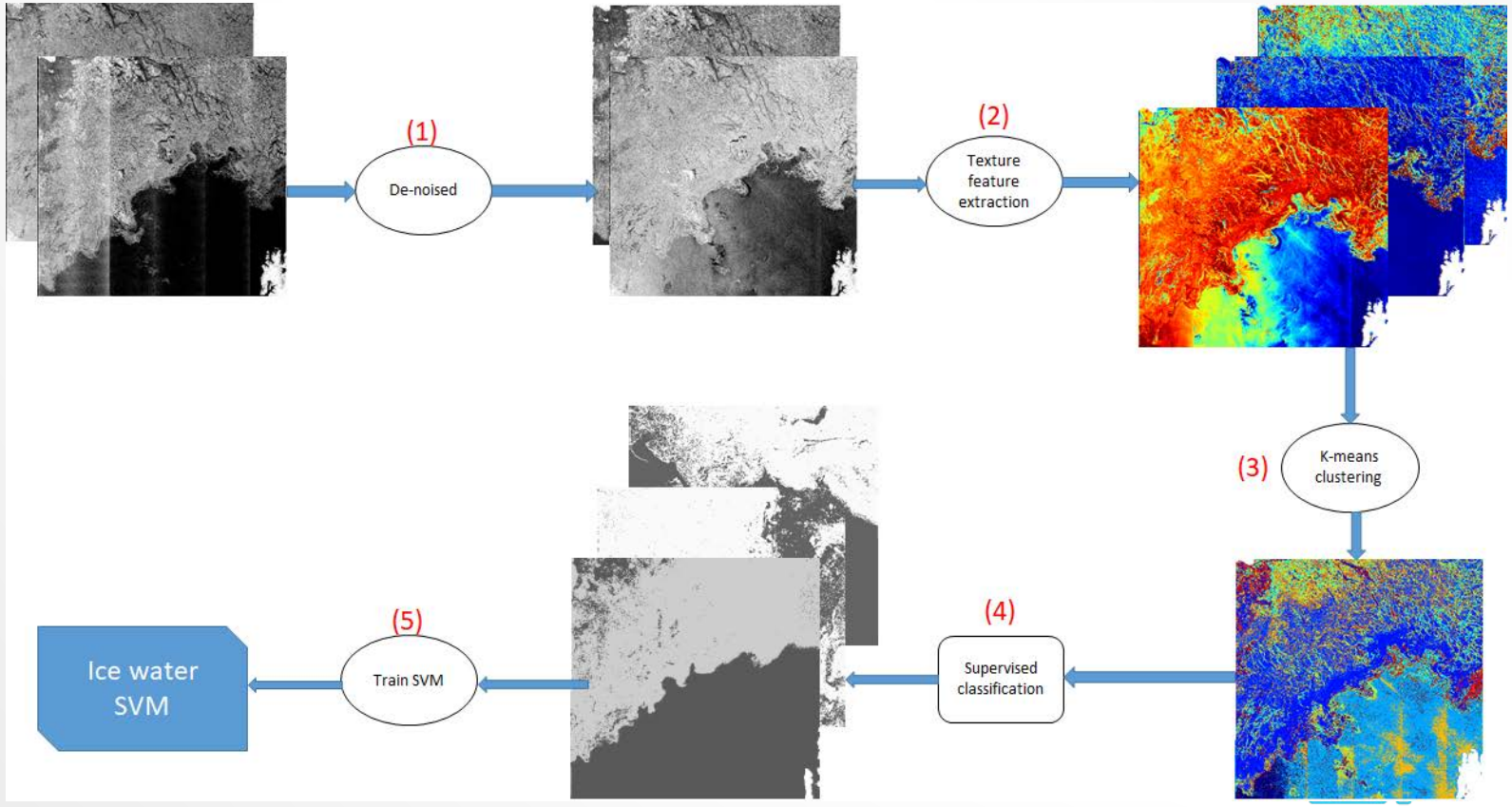
NERSC, ARMINES & IMR

Examples of services

Examples of services (#1)

NEXTGEOSS
Contributing to the Vision of GEO

- Sea ice classification service
  - Sentinel-1 SAR, EW, dual polarisation
  - Algorithm uses ML techniques
  - Thermal Noise reduction
  - Coded in Python
  - Open source libraries Nansat, GDAL, …
  - Outputs GeoTIFF map
  - Wrapped as cloud service in iAOS



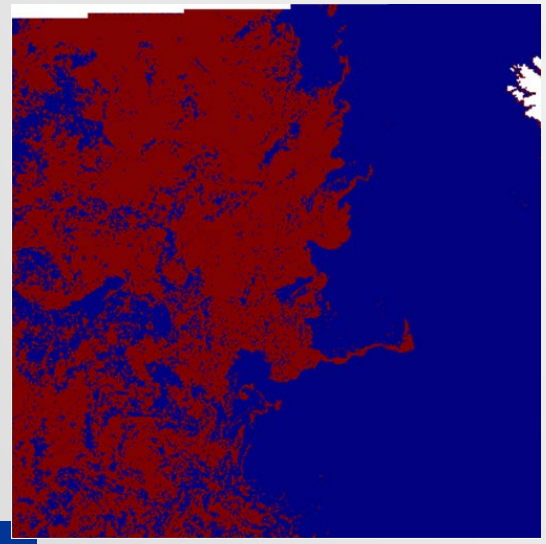
INTAROS

Examples of services (#1)

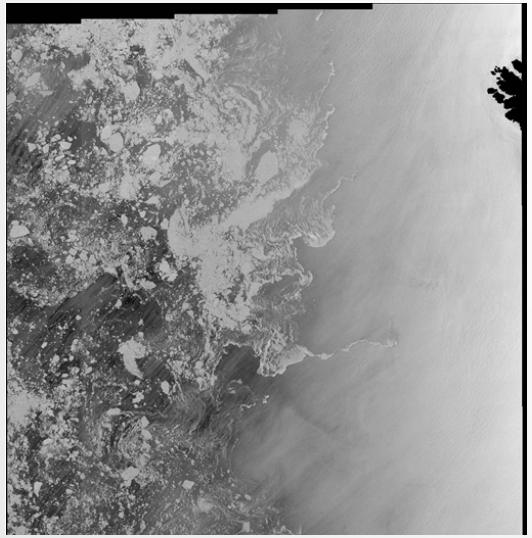
NEXTGEOSS
Contributing to the Vision of GEO

- Sea ice classification – for INTAROS 2018 Cruise (29 Jul – 20 Aug)
  - Service deployed on EGI cloud infrastructure
  - Classified 200 Sentinel-1 SAR images from Fram Strait & North of Svalbard

29 July 2018

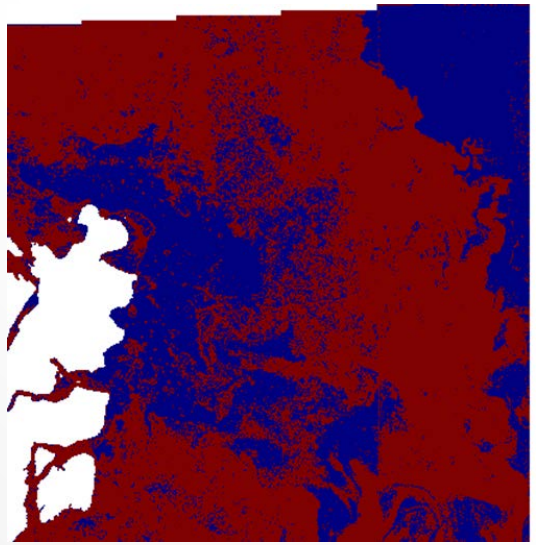


Blue - open water.
Red – sea ice.

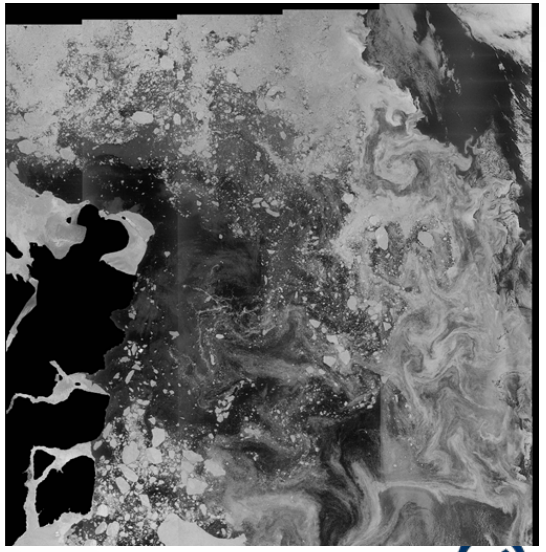


S1-A SAR, EW, HH polarisation

11 Aug 2018



Blue - open water.
Red – sea ice.

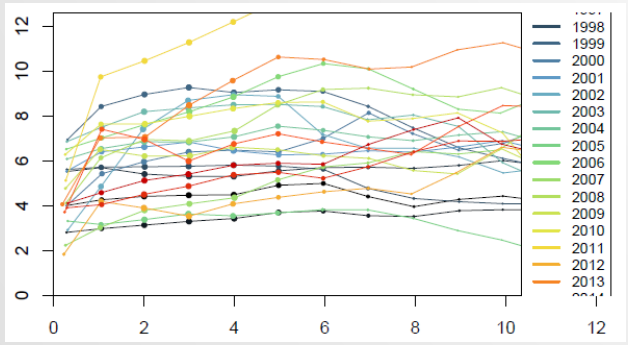


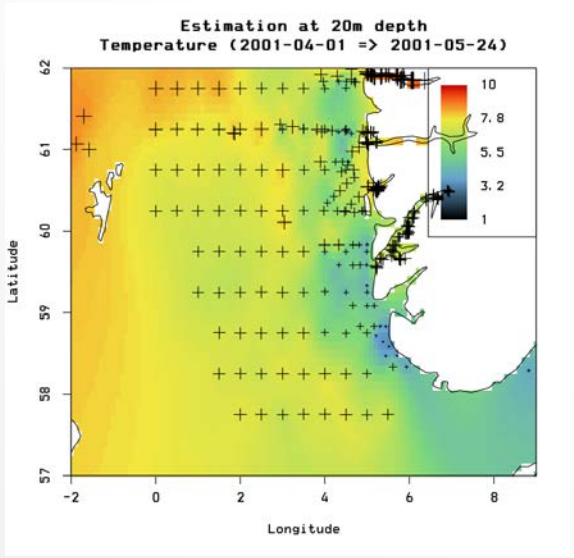
S1-A SAR, EW, HH polarisation

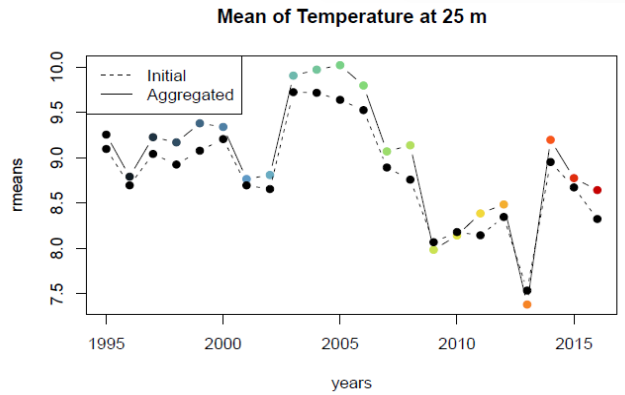
INTAROS

Examples of services (#2)

- Use geo-statistics library **RGeostats** developed by ARMINES
- Service for interpolating in situ data to a (model) grid
  - Describe the spatial characteristics of the variable (variogram): classification, spatial correlation
  - Estimation, interpolation (kriging)
  - Simulations: possible alternative scenarios
  - Appraisal of uncertainty
  - Risk assessment

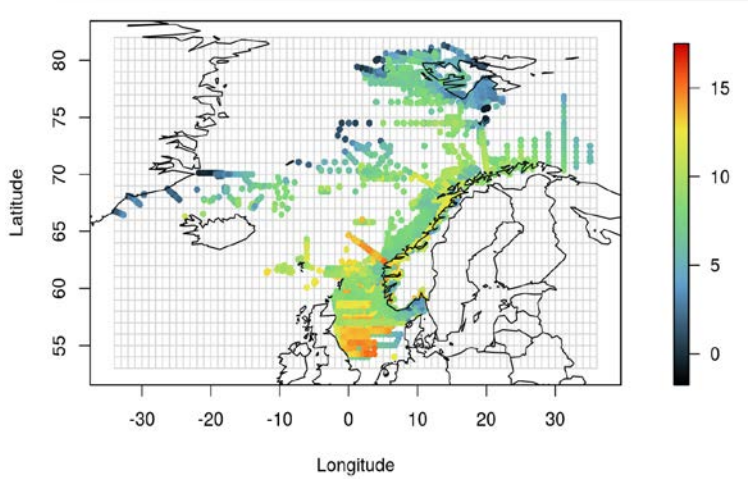


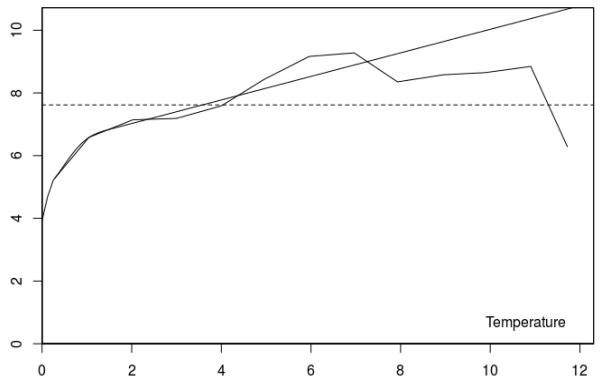


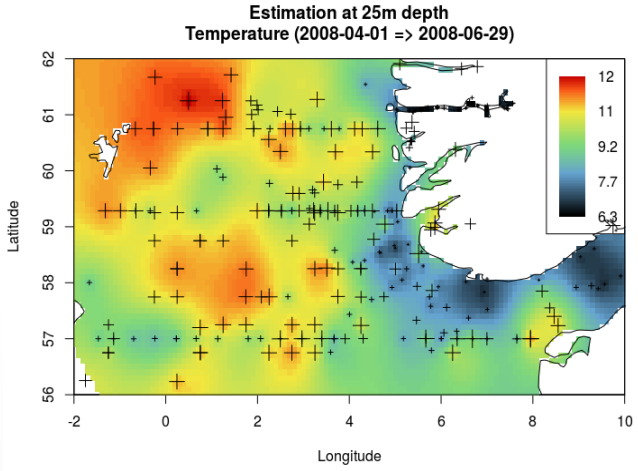


Examples of services (#2)

- Service implemented in Notebook environment
  - Test data from IMR (7 vessels from 1995 to 2016)
  - 3 variables measured:
    - Temperature
    - Salinity
    - Conductivity
  - 63 500 positions {long, lat}
  - 63 500 vertical profiles (in depth)
  - 5 billions samples
  - 84 NetCDF files (~60 Mb each)
  - Interpolated to grid by Kriging
  - Input to WP6 tasks

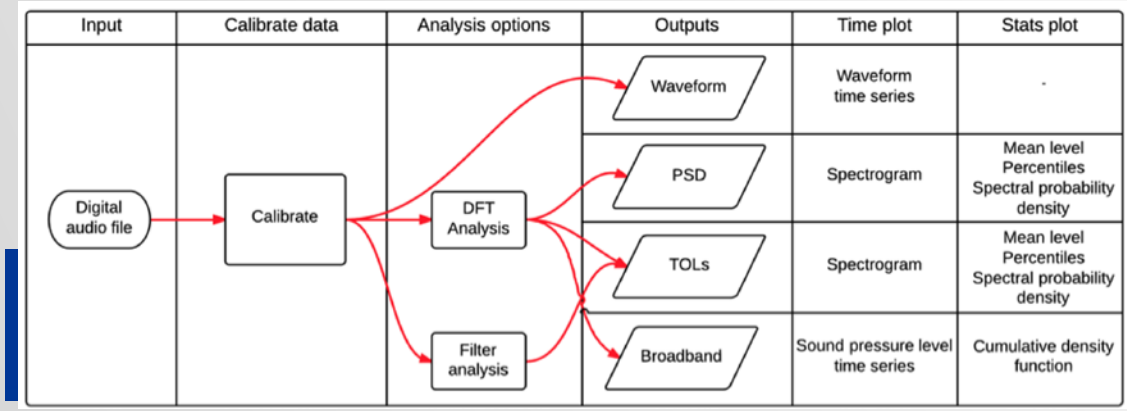


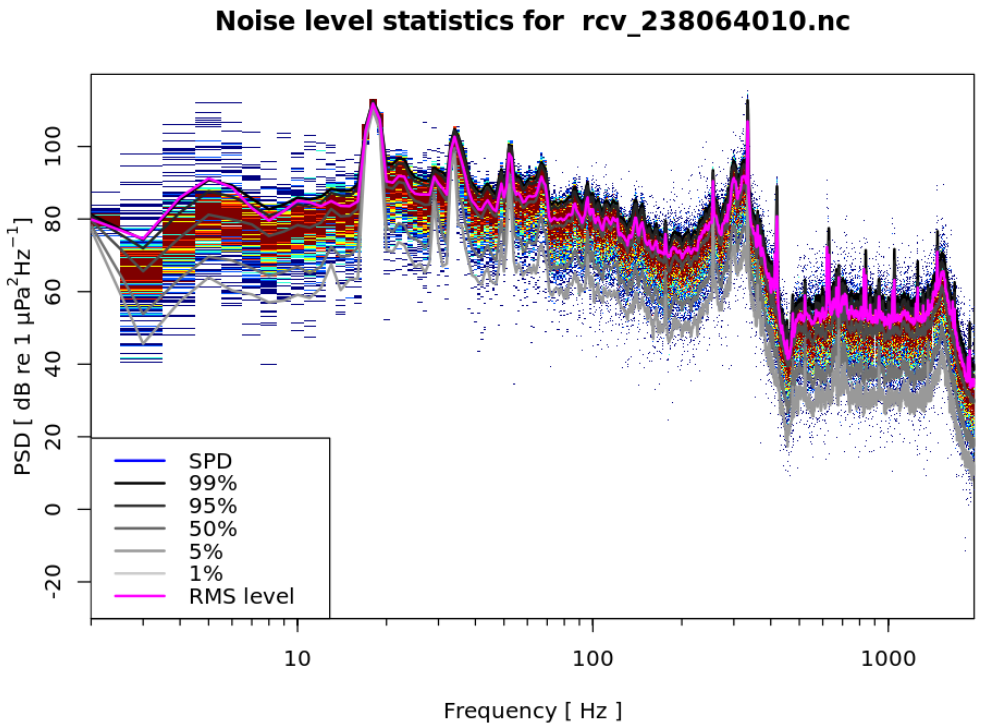


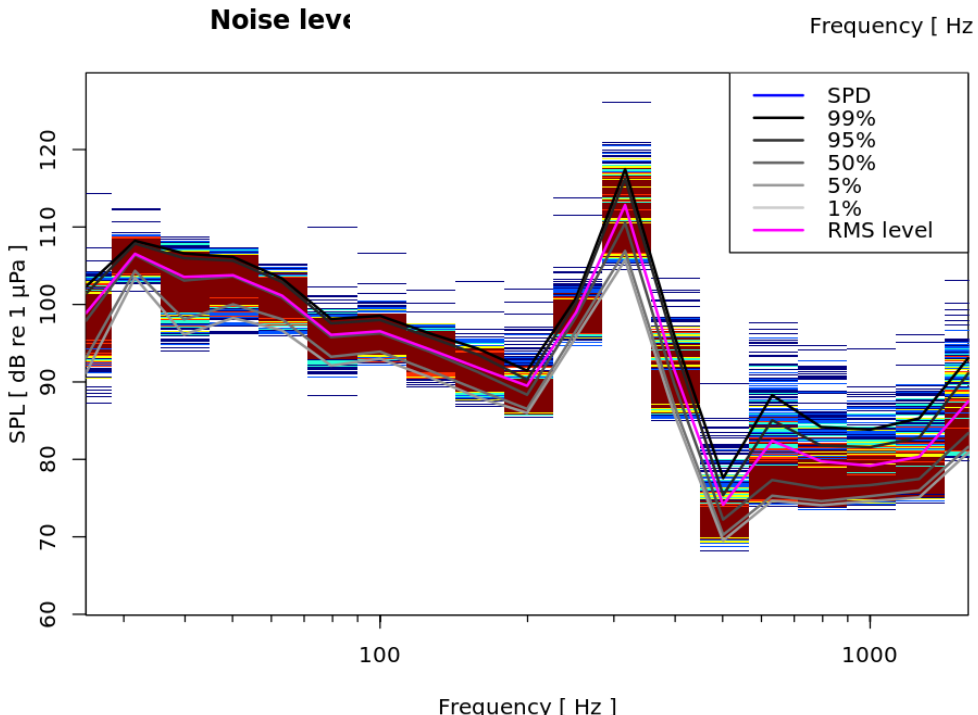


Service #3

- **Acoustic data processing & analysis**
  - PAMGuide tool (R version)
  - Calibration and processing
  - Different plots to investigate the data (spectrograms, noise statistics)
  - Supports WAV, NetCDF, …
  - Testing with WIFAR data





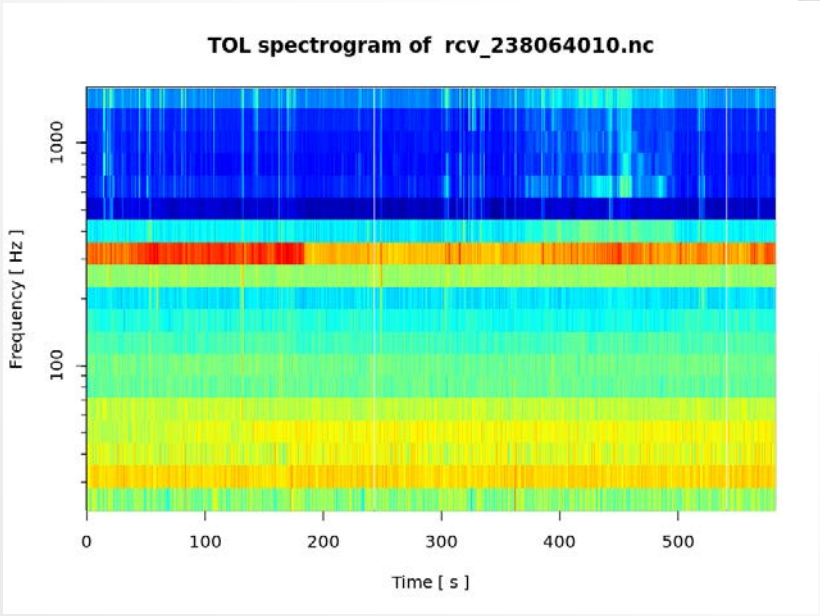


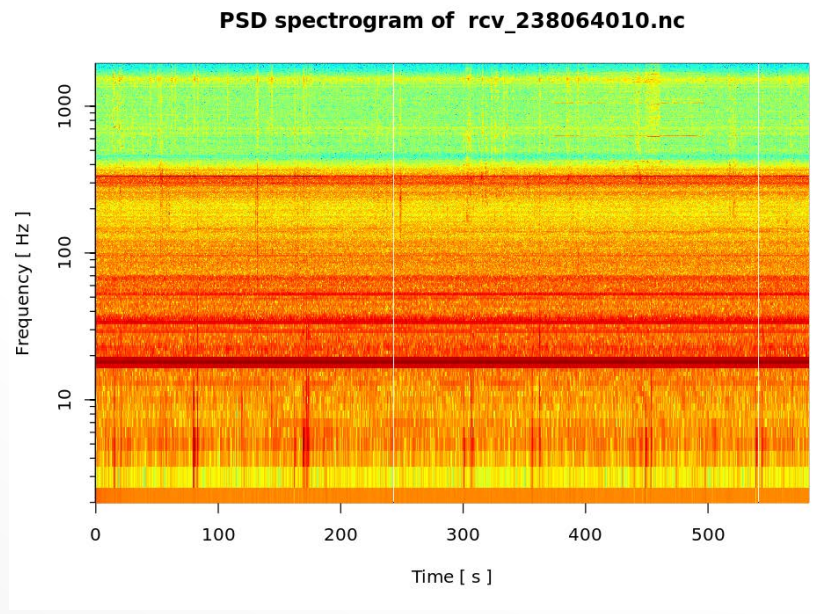
INTAROS

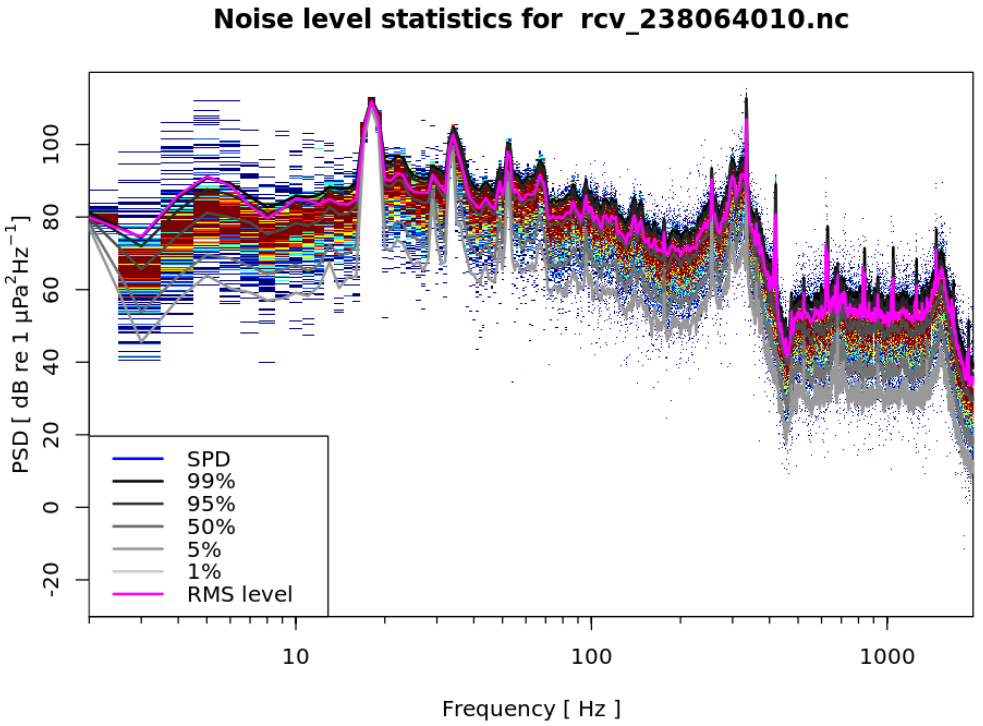
Service #3

- **Acoustic data processing and analysis**
  - Developed using Jupyter Notebook
  - Initial version ready
  - Including OPeNDAP support
  - For INTAROS WP6 (PAMGuide installed, extended, packaged as cloud service)
  - To be used in training

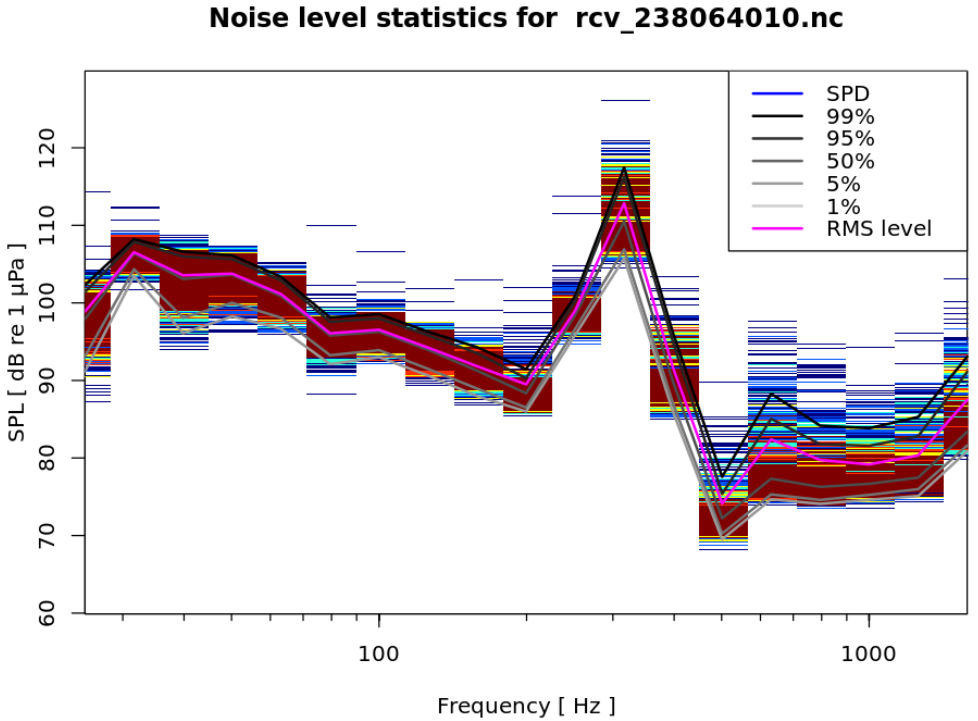
```
In [401]: # Get attributes:
          nc_atts <- ncatt_get(ncin,0)
          #print(nc_atts)

          print(nc_atts$summary)
          print(nc_atts$data_assembly_center)

[1] "WIFAR/UNDER-ICE acoustic recording from an integrated ice station in the Fram Strait marginal ice zone"
[1] "CONSORTIA/INSTITUTIONS>>>>NERSC >Nansen Environmental and Remote Sensing Centre>http://www.nersc.no/main/index2.php"

In [402]: # Sampling frequency [Hz]:
          fs <- ncvar_get(ncin, "sample_rate", verbose = F)

          # Time at start of recording:
          t0 <- ncvar_get(ncin, "start_time", verbose = F)

          # Samples:
          x <- ncvar_get(ncin, "samples", verbose = F)

          # Get the number of samples in the recording.
          Nsamp <- length(y)

          # Vertical bit resolution:
          Nbit = 24

In [403]: # Sensitivity of the hydrophone:
          HySens <- -168 # dB re 1 V/uPa


          # Gain of the voltage amplifier:
          Gain <- 12 # dB re 1 V/V

          # Digitizer information:
          B2V <- 2.5/2^(Nbit - 1) # Volt/Bit

          B2 <- 1/2^(Nbit - 1)

In [413]: # From samples to bit ...
          ybit <- x*B2

          # Want units in Pa => must add 120 dB
          pressure <- x*B2V/10^((HySens+120)/20)/10^(Gain/20) # Unit: Pascal [Pa]
          # Set up the time vector from t0 with dt
          dt = 1/fs
          t <- seq(0, Nsamp-1)*dt

In [405]: source("PAMGuide.R")

In [412]: PAMGuide(fullfile="rcv_238064010.nc", atype="TOL", plottype="Both", r=over1, lonlog="Lin",isvector=1, y=x,vADC=2
```

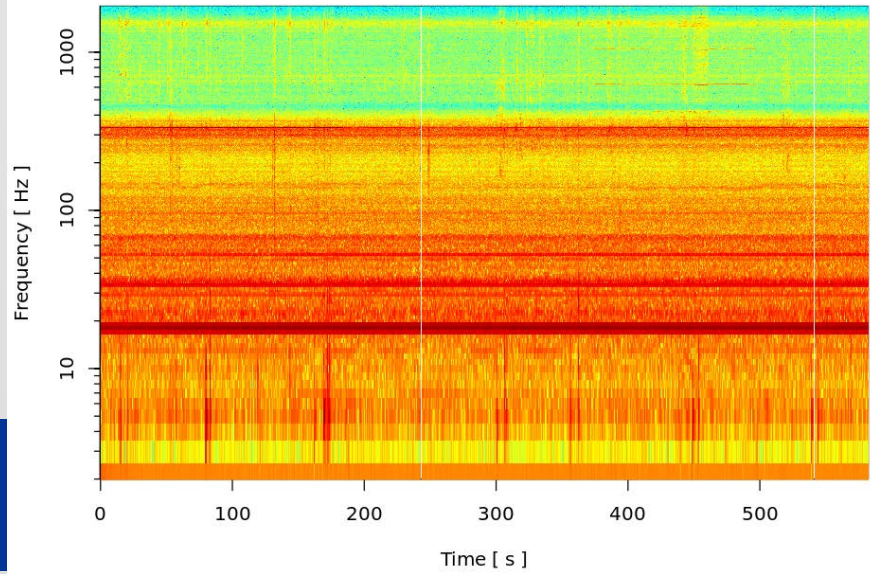


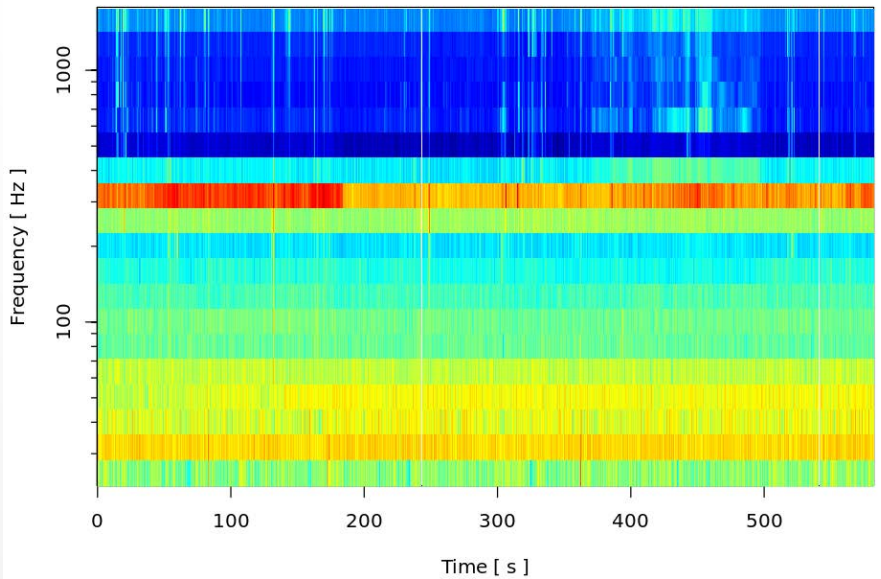



Noise level statistics for rcv_238064010.nc
PSD [dB re 1 µPa²Hz⁻¹]
Frequency [Hz]
SPD
99%
95%
50%
5%
1%
RMS level

a


Noise level statistics for rcv_238064010.nc
SPL [dB re 1 µPa]
Frequency [Hz]
SPD
99%
95%
50%
5%
1%
RMS level


PSD spectrogram of rcv_238064010.nc
Frequency [Hz]
Time [s]


TOL spectrogram of rcv_238064010.nc
Frequency [Hz]
Time [s]

Summary

Summary

- iAOS in INTAROS is based on proven technological cloud platform; extensive functionality and support for cloud service development
- iAOS provides access to Arctic data in distributed databases – *no duplicaiton of data* – only metadata is harvested
- Data can be downloaded or used in services
- Complexity of services affects the time needed to "cloudify" them (e.g. many dependencies, pre-installations of libraries, extensions of code, etc.)
- Services will be further developed in INTAROS, and made available through the iAOS Portal