



PROviding Computing solutions for ExaScale Challenges

D2.1	Progress Report (UC#1-5)		
Project:	PROCESS H2020 – 777533	Start / Duration:	01 November 2017 36 Months
Dissemination¹:	PU	Nature²:	R
Due Date:	31 October 2018	Work Package:	WP 2
Filename³	PROCESS_D2.1_Progress_Report_v1.0.docx		

ABSTRACT

The objective of this deliverable is to provide an overview of the progress made with the use cases in the first year of the PROCESS project. For each use case, we provide a brief overview of its goal, a detailed description of the progress that was made and any challenges that were encountered, and an outlook for the next year.

Overall, significant progress has been made in the first 12 months on 3 of the use cases. As a result 5 out of 6 KPIs have been met. Once the PROCESS services prototypes are operational, the two remaining use cases will engage their existing communities, with the aim of expanding the user base of the PROCESS platform in the second and third year.

This version is a draft of D2.1 and is under review.

¹ PU = Public; CO = Confidential, only for members of the Consortium (including the EC services).

² R = Report; R+O = Report plus Other. Note: all "O" deliverables must be accompanied by a deliverable report.

³ eg DX.Y_name to the deliverable_v0xx. v1 corresponds to the final release submitted to the EC.

This version is a draft of D2.1 and is under review.

Deliverable Contributors:	Name	Organization	Role / Title
Deliverable Leader⁴	Maassen, Jason	NLESC	Coordinator
Contributing Authors⁵	Graziani, Mara	HESSO	Writer
	Spreeuw, Hanno; Maassen, Jason	NLESC	Writers
	Heikkurinen, Matti; Hüb, Maximilian	LMU	Writers
	Biel, Michal; Reichardt Janek; Pancake-Steeg, Jörg	LSY	Writers
Reviewer(s)⁶	Belloum, Adam; Cushing, Reggie	UvA	Reviewer
	Heikkurinen, Matti	LMU	Reviewer
Final review and approval			

Document History

Release	Date	Reasons for Change	Status⁷	Distribution
0.0	24-09-2018	Structure	Draft	
0.1	12-10-2018	First draft	Draft	
0.2	15-10-2018	Text use cases complete	Draft	
0.3	16-10-2018	Merged D4.2 and restructured text	Draft	
0.4	17-10-2018	Finalization for internal review	In Review	
1.0	31-10-2018	Processed comments of review	Released	

⁴ Person from the lead beneficiary that is responsible for the deliverable.

⁵ Person(s) from contributing partners for the deliverable.

⁶ Typically person(s) with appropriate expertise to assess the deliverable quality.

⁷ Status = "Draft"; "In Review"; "Released".

Table of Contents

Table of Contents

- Executive Summary 4
- List of Figures 5
- List of Tables 5
- 1 Introduction 6
- 2 Use Case 1: Exascale learning on medical image data..... 8
 - 2.1 Overview 8
 - 2.2 Progress 8
 - 2.3 Challenges..... 10
 - 2.4 Outlook 11
- Use Case 2: Square Kilometer Array / LOFAR 12
 - 2.5 Overview 12
 - 2.6 Progress 12
 - 2.7 Challenges..... 15
 - 2.8 Outlook 15
- 3 Use Case 3: Supporting Innovation on global disaster risk data 17
 - 3.1 Overview 17
 - 3.2 Progress 17
 - 3.3 Challenges..... 18
 - 3.4 Outlook 18
- 4 Use Case 4: Ancillary pricing for airline revenue management 19
 - 4.1 Overview 19
 - 4.2 Progress 19
 - 4.3 Challenges..... 20
 - 4.4 Outlook 20
- 5 Use Case 5: Agricultural analysis based on Copernicus data 21
 - 5.1 Overview 21
 - 5.2 Progress 21
 - 5.3 Challenges..... 21
 - 5.4 Outlook 21

Executive Summary

The objective of this deliverable is to provide an overview of the progress made with the use cases in the first year of the PROCESS project. More specifically, how the use cases have progressed in relation to the Use Case Analysis described in D4.1 (Section 1, pages 11-47).

A distinction should be made between the use cases that have direct developer involvement in PROCESS (UC 1, 2 and 4) and the use cases which focus more at engaging existing communities (UC 3 and 5). While the former proceeded to actively develop their applications in conjunction with the development of the PROCESS services, the latter are dependent on having a running prototype of the PROCESS services implementation to demonstrate to their communities of interest. Only once these services deployed, can these communities be invited to evaluate the services for their own needs.

The first use case, *Exascale learning on medical image data*, has worked extensively with infrastructure developers to run initial pilot experiments on PROCESS infrastructure at Cyfronet in Krakow, Poland. At time of writing, they have started producing the first benchmark results on PROCESS infrastructure for this use case.

The second use case, *Square Kilometer Array / LOFAR*, has mainly focussed on creating a user portal for the easy selection of data and launching of processing pipelines. A prototype of this portal is available, but is not yet integrated into PROCESS infrastructure. Two processing pipelines are available, which can be used to process data, although neither are running on PROCESS infrastructure yet.

The fourth use case, *Ancillary pricing for airline revenue management*, has been focussing on generating a standard ancillary sales data set for airlines. This make testing new service concepts possible without having to obtain consent from actual users. A prototype data generation tool has been created and is running on Cyfronet in Krakow, Poland.

The third and fifth use cases, *Supporting Innovation on global disaster risk data*, and *Agricultural analysis based on Copernicus data*, have made limited progress in creating the roadmap and detailed plans for the integration of their application software in the PROCESS architecture. As explained above, they require the prototype of the PROCESS services to be operational in order to demonstrate its features to their user communities. Once the prototype is operational these communities will be actively approached, with the aim of expanding the user base of the PROCESS platform in the second and third year.

Overall, significant progress has been made in the first 12 months, and 5 out of 6 KPIs have been reached. In the next months, the use cases will focus on integration with the initial prototype of PROCESS services which will become available in month 12.

List of Figures

List of Figures

Figure 1: Camnet: development status 9
Figure 2: MNIST benchmarking on HESSO infrastructure 10
Figure 3: The prototype of the measurement set selection portal. This portal allows the user to search for data based on observation time, right ascension and declination, subbands, or any combination. 13
Figure 4: The initial pipeline selection form, presented by the portal after the selection of a measurement set. 14
Figure 5: The airline setting generated by the datagenerator 19

List of Tables

Table 1: Month 12 KPIs 6
Table 2: Camnet benchmarking and troubleshooting report 11

1 Introduction

In this document we will provide an overview of the progress made by the use cases in the first year of the project, and relate this progress to the KPIs. As the initial PROCESS prototype implementation will not be deployed until M12, the use cases could not yet make use of this prototype implementation in this first year (progress made on this prototype implementation can be found in D4.2). Nevertheless, several of the use cases already make use of the hardware infrastructure available in PROCESS and will start integration of the PROCESS services prototype as soon as they become available.

KPI ID	Description	Target at M12	Measured Aspect	Achieved
KPI1	Data volume in medical use case UC#1	Data from 1000 persons	The amount of data that is available to us in the medical UC for training the tools, first from existing cohorts and scientific challenges.	yes
KPI2	Performance increase of the medical use case UC#1	Baseline of performance	The measure of increased effectiveness when evaluating medical data in our developed solution as opposed to the original method.	yes
KPI3	Increase of revenue in the Ancillary Pricing UC#4	Historical data for ancillary sales is simulated	Measuring the revenue increase for an airline is tricky, since a flight can obviously only take-off once. Therefore we will create a simulation that simulates customer behaviour with respect to buying ancillaries according to the simulated customer base. With this simulation we are able to compare different methods of ancillary pricing.	yes
KPI4	Active users of the UNISDR data (UC#3)	20 users from more than one country	The basic measurement is a proxy indicator of the impact on global scale and visibility of European contributions in the disaster risk reduction domain. The country-information will be deducted from the network addresses	yes
KPI5	Examples of complex simulations based on preprocessed Copernicus data (UC#5)	Enabling 3 different Earth system simulations	Each of these simulations targets different aspects of earth system modeling and requires different preprocessing approach for producing Earth System model input.	no
KPI6	Processing workflows active on LOFAR/ SKA archive data (UC#2)	Processing single workflow on selection of LOFAR archive (order of 100GB)	Measures the scalability of the proposed solution, both in amount of data processed by a single workflow and number of workflows concurrently handled.	yes

Table 1: Month 12 KPIs

Introduction

Table 1 list the M12 targets for each of the KPIs and if this target has been achieved or not (see 777533 PROCESS – Part B – v1.1, page 36 for the full table). As the table shows, the targets have been reached for 5 out of 6 KPIs.

In the remainder of this document, each use case will give a detailed report on its progress, describe its current pilot scenario and/or datasets used, describe if and how it is using the PROCESS infrastructure, elaborate on any problems which were encountered, and provide an outlook for the next 12 months. The use cases will not be described in detail here. They can be found in the *Use Case Analysis* section of D4.1 (Section 1, pages 11-47).

2 Use Case 1: Exascale learning on medical image data

2.1 Overview

Developing better tools for detection, localisation, stage grading and treatment planning is a current need in medical imaging research. The application of machine learning to medical imaging allows to analyse large amounts of patient data and introduces objectivity in the diagnostics. Exascale learning extends traditional approaches by increasing the number of trained parameters and dataset sizes. More details about the use case are given in D4.1 (see use case analysis, Section 1.1, pages 11-22).

2.2 Progress

We developed the first pilot application of the use case (e.g. Camnet: Interchangeable network architectures for Camelyon), which tackles cancer detection and tissue classification for Camelyon dataset⁸ (Camelyon16 and 17).

Camelyon is currently the largest and the most challenging dataset for histopathology research. The two datasets contain more than 1000 tissue Whole Slide Images (WSIs), gathering data from more than 200 patients. We applied data augmentation to the models to simulate a 3 fold increase in the number of patients. It is important to notice that from a computational perspective using 1000 WSIs from 200 patients is equivalent to using data from 1000 separate patients with 1 WSI each. By using additional data augmentation it is possible scale up the dataset further and simulate the computational requirements of having a larger number of WSIs per patient. In this way it is possible to focus on the processing challenges instead of on having to gather data from extremely large cohorts.

The data have been transferred to the PROCESS storage. While doing so, the Data Upload workflow as proposed in D4.1 has been analysed and integrated in the PROCESS architecture development and initial data load and storage functionalities have been developed. Future work will focus on the integration of the data pre-stage and pre-processing services described in the D5.1 (pages 19-22).

A three-layer software architecture for training different deep neural network models has been developed as a prototype of the use case functionalities⁹. The Model training workflow guidelines presented in D4.1 (see Figure 4, page 18) have been followed. Figure 1 presents a summary of the workflow components. The first two layers have been completed in the first 12 months of the project, namely the patch extractor and the patch classifier. The application was ported on the following PROCESS resources: the Machine Learning platform (LRZ), Prometheus (AGH), UISAV and UVA.

⁸ <https://camelyon17.grand-challenge.org/Data/>

⁹ https://github.com/medgift/PROCESS_UC1/

Development status

Camnet: Interchangeable networks for Camelyon

Layer I: Data preprocessing and patch extraction	Layer 2: Local and Distributed Training	Layer III: Performance Boosting and Interpretability
Creation of normal and tumor tissue masks — completed	Interchangeable network models — 2 models developed	Generation of intermediate visualisations
Random sampling of patches at high resolution — completed	Local training — benchmarks for Titan V, Titan X, V100	Feature importance analysis
Creation of H5DS database of patches — completed	Training on HPC clusters	Perturbation robustness analysis
	Distributed training — local scaling up to 2 GPUs K80	

Figure 1: Camnet: development status

The first layer focuses on data preprocessing and patch extraction. The pre-processing steps of the raw BIGTIFF WSIs include: the scan of the filesystem, the retrieval of patient-related metadata, the extraction of normal and tumor tissue binary masks by thresholding methods which are standard in digital pathology. Different sampling strategies can be adopted to extract high-resolution patches from the tissue masks. For the initial prototype random sampling has been chosen. Patches with non-relevant information (e.g. white content, black pixels, background, etc.) have been filtered out and discarded. Information about the patient, the lymphnode, the hospital which handled the acquisitions, the resolution level of the patch and the patch location in the WSIs are stored together with the pixel values in an intermediate HDF5 database. Moreover, the doctor annotations are stored in the HDF5 as a binary label on the patch, which discriminates between tumor and non-tumor patches.

The second layer loads the intermediate HDF5 dataset and focuses on training deep learning architectures for the binary classification between tumor and non-tumor patches. A modular system has been developed to allow the training of multiple architectures and to test their scaling to a larger number of parameters

Several state-of-the-art neural network architectures (pre-trained on Imagenet) have been fine-tuned on the intermediate dataset and they can be selected for the classification. Two modules have been developed for benchmarking the ResNet50 and ResNet101 deep neural network architectures¹⁰. At testing time patch-level classification accuracy was used for performance evaluation. Heatmaps of the probability of the presence of tumorous tissue were generated for visual inspection. Feature importance was investigated through network interpretability.

For the initial experiments we had access to several different GPUs (K40, K80, Titan V, Titan X and V100). To create an initial performance base line, we used the existing MNIST model¹¹ on these different architectures. The results, shown in Figure 2, provides us with a relative performance of the different GPUs and allow us to compare this performance to well known

10 He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

11 LeCun, Yann. "The MNIST database of handwritten digits." <http://yann.lecun.com/exdb/mnist/> (1998).

Use Case 1: Exascale learning on medical image data

benchmarks. These include results for distributed training using multiple K80 GPUs in a single node.

Next, we performed experiments with the use case application, following Scenario n.1 of D4.1 (page 14). The initial results are shown in Table 2. During these experiments, we have run into several issues for which further analysis is required. Especially, the optimisation of CPU memory to GPU memory transfer emerged the main bottleneck while porting the software to the different architectures. Such optimisation will be the focus of future development. The baselines reported will be used to measure the progress of KPI2 during the lifetime of the project.

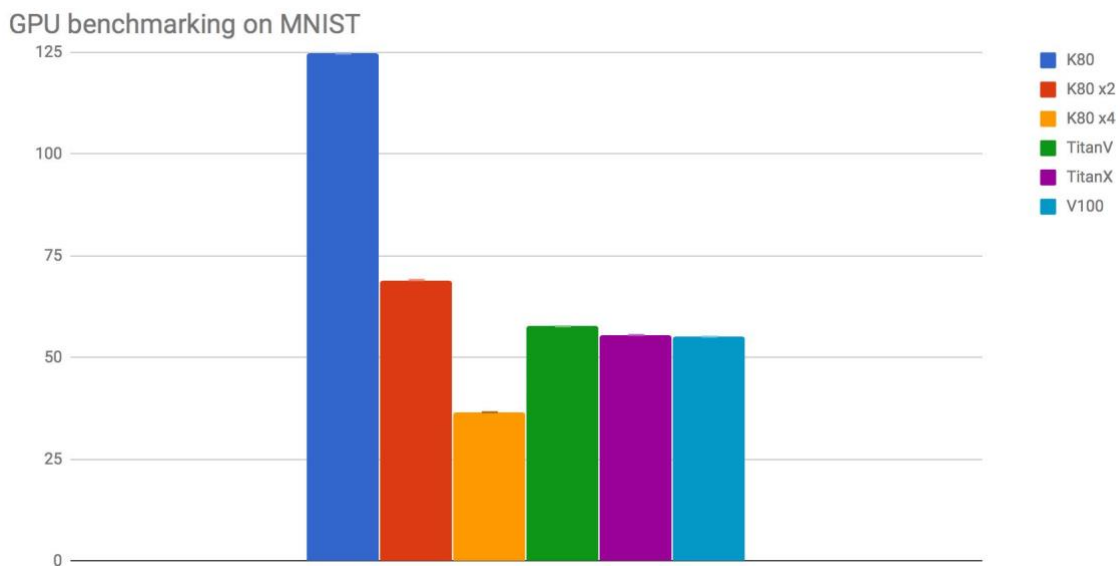


Figure 2: MNIST benchmarking on HESSO infrastructure

2.3 Challenges

Data movement was identified as a key challenge. The time required to transfer the raw data can be drastically reduced by integrating the patch extraction layer (software layer I) with the PROCESS data services proposed in D5.1 (pages 21-22). This will allow the extraction of selected resolutions and selected rectangles from the pyramid TIFF file and help to reduce dataset sizes and make the transfer efficient and paving towards exascale.

Use Case 1: Exascale learning on medical image data

System details	Resource Location	Notes/Issues	Time per epoch	Expected tot. Training time	Performance training accuracy/ validation accuracy (%)
No.1 GPU K80	HESSO	-	~ 2400 s (40 minutes)	~ 7 h	98.83/85.18
No.1 GPU V100	HESSO	-	1800 s (30 minutes)	~ 6 h	81.21/86.19
No.1 GPU TitanV	HESSO	CPU-memory to GPU-memory resource exhaustion	-	-	-
No. 2 GPUs K80	HESSO	CPU-memory to GPU-memory resource exhaustion	-	-	-
No.1 GPU K40	AGH	CPU-memory to GPU-memory resource exhaustion	-	-	-

Table 2: Camnet benchmarking and troubleshooting report

The optimization of the neural network code to be system-specific is also a key impediment to performance. The link of CPU memory to GPU memory has been identified as a major bottleneck in the execution of the scripts on cluster machines.

2.4 Outlook

The next year will focus on troubleshooting the optimization requirements due to the specific hardware architectures and on the development of Layer II and Layer III of the software.

The use case will use the data services for pre-processing the WSIs and facilitating data transfer as soon as they become available. The compute services will be used to train different models independently on each data centre, as proposed by Scenario n.1 in D4.1 (page 14). This scenario simulates the hospital conditions of limited data sharing permissions and limited computing resources.

Statistics about network training will be collected and visualized with the compute services. Data services will also take care of storing weights, checkpoints and statistics about the trained network models. The results will be validated in terms of performance increase with respect to the baseline performances of the first software architecture.

Use Case 2: Square Kilometer Array / LOFAR

2.5 Overview

LOFAR is a state-of-the-art radio telescope capable of wide field imaging at low frequencies. It serves as a testbed for the Square Kilometer Array (SKA) since it is similar to the part of SKA that will be built in South Africa and Australia. LOFAR has been producing data at a rate of approximately 5-7PB/year since 2012, resulting in a long-term archive (the LOFAR LTA) of over 30 PB.

Currently expert knowledge is needed to process observations stored in the archive. The goal of this use case is to simplify this processing. Astronomers should be able to select a dataset on a portal, select a workflow, and then launch the processing pipeline from there. For this we need an easy to use, flexible, efficient and scalable infrastructure for processing of extremely large volumes of astronomical observation data.

PROCESS will provide a mechanism to run containerized workflows, thereby improving the portability and ease of use. A suitable portal is needed to select datasets and workflows. Through this portal, the astronomer must be able to browse through the available datasets and available workflows, and launch processing directly from there to the hardware infrastructure available in the project. Data should then be transferred from the LTA to the processing infrastructure, processed, and the results made available in the portal. A more detailed description of this use case is given in D4.1 (Use case analysis, Section 1.2, pages 23-30).

2.6 Progress

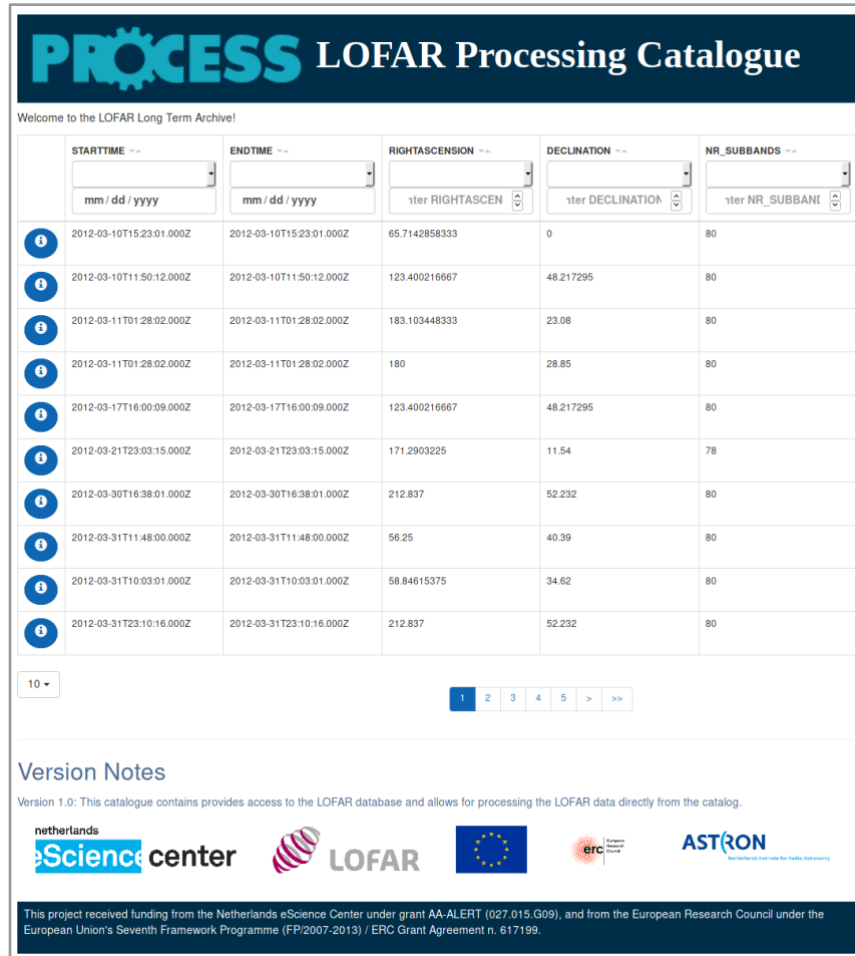
In the first year we have focussed on creating the portal¹² astronomers will use to select the required data sets and launch a workflow. This is the first requirement as described in the requirements analysis of D4.1 (Section 1.2.8, page 29). The prototype of this web interface is shown in Figure 3. This portal was created in cooperation with the EOSC-pilot for LOFAR¹³ and AA-Alert¹⁴ projects.

¹² <https://github.com/EOSC-LOFAR/ltacat>

¹³ <https://eoscpilot.eu/science-demos/lofar-data>

¹⁴ <https://www.esciencecenter.nl/project/aa-alert>

Use Case 2: Square Kilometer Array / LOFAR



The screenshot shows the 'PROCESS LOFAR Processing Catalogue' website. It features a search interface with dropdown menus for STARTTIME, ENDTIME, RIGHTASCENSION, DECLINATION, and NR_SUBBANDS. Below the search fields is a table listing measurement sets with columns for STARTTIME, ENDTIME, RIGHTASCENSION, DECLINATION, and NR_SUBBANDS. The table contains 10 rows of data. At the bottom of the table, there is a pagination control showing '10' items and a set of navigation buttons (1, 2, 3, 4, 5, >, >>). Below the table is a 'Version Notes' section with the text: 'Version 1.0: This catalogue contains provides access to the LOFAR database and allows for processing the LOFAR data directly from the catalog.' At the bottom of the page, there are logos for the Netherlands eScience Center, LOFAR, the European Union, ERC, and ASTROn. A footer text states: 'This project received funding from the Netherlands eScience Center under grant AA-ALERT (027.015.G09), and from the European Research Council under the European Union's Seventh Framework Programme (FP2007-2013) / ERC Grant Agreement n. 617199.'

	STARTTIME --	ENDTIME --	RIGHTASCENSION --	DECLINATION --	NR_SUBBANDS --
1	2012-03-10T15:23:01.000Z	2012-03-10T15:23:01.000Z	65.7142858333	0	80
2	2012-03-10T11:50:12.000Z	2012-03-10T11:50:12.000Z	123.400216667	48.217295	80
3	2012-03-11T01:28:02.000Z	2012-03-11T01:28:02.000Z	183.103448333	23.08	80
4	2012-03-11T01:28:02.000Z	2012-03-11T01:28:02.000Z	180	28.85	80
5	2012-03-17T16:00:09.000Z	2012-03-17T16:00:09.000Z	123.400216667	48.217295	80
6	2012-03-21T23:03:15.000Z	2012-03-21T23:03:15.000Z	171.2903225	11.54	78
7	2012-03-30T16:38:01.000Z	2012-03-30T16:38:01.000Z	212.837	52.232	80
8	2012-03-31T11:48:00.000Z	2012-03-31T11:48:00.000Z	56.25	40.39	80
9	2012-03-31T10:03:01.000Z	2012-03-31T10:03:01.000Z	58.84615375	34.62	80
10	2012-03-31T23:10:16.000Z	2012-03-31T23:10:16.000Z	212.837	52.232	80

Figure 3: The prototype of the measurement set selection portal. This portal allows the user to search for data based on observation time, right ascension and declination, subbands, or any combination.

The portal presents the measurement sets stored in the LOFAR archive. As explained in D4.1 (Section 1.2.7, page 29), each measurement set consists of an observation performed by the telescope for a certain time period and certain patch of the sky (expressed in right ascension and declination) and for a number of frequency subbands. Using any combination of these five fields, the user can search through the ~16000 measurement sets available (28 PB in total).

Once a dataset has been selected, the user can directly launch one of the available processing pipelines, as shown in Figure 4. For the current pilot only a single pipeline is available in the portal, the “LOFAR grid pre-processing pipeline” (LGPPP)¹⁵, which is based on the GRID_LRT¹⁶ pipeline developed at Leiden University and SURFsara (described in D4.1, Section 1.2.5, page 26). This LGPPP pipeline performs the initial calibration of the data (i.e., correcting for atmospheric disturbance) and but does not yet perform any imaging.

Like most pipelines, the pre-processing pipeline has a number of configuration parameters which may be set by the user, such as desired frequency or time averaging of the measurement samples (which increases signal to noise ratio, but decreases resolution). Once these parameters are set to the satisfaction of the astronomer, the workflow can be launched directly from the portal.

¹⁵ https://github.com/EOSC-LOFAR/LGPPP_LOFAR_pipeline

¹⁶ https://github.com/apmechev/GRID_LRT

Use Case 2: Square Kilometer Array / LOFAR

Observation overview

Product Parameters

Start time	2012-03-11T01:28:02.000Z
End time	2012-03-11T01:28:02.000Z
Right Ascension	183.103448333
Declination	23.08
Nr subbands	80

Data Processing

E-mail address:
h.spreeuw@esciencecenter.nl

Job description:
test 2

Select processing pipeline:
LOFAR GRID Pre-Processing Pipeline

Configuration Parameters:

This is the LOFAR GRID Pre-Processing Pipeline. Here we print a description of the pipeline.

search

avg_freq_step
2
corresponds to .freqstep in NDPPP .type=average , or in case of .type=demixer it is the demixer.freqstep

avg_time_step
4
corresponds to .timestep in NDPPP .type=average , or in case of .type=demixer it is the demixer.timestep

do_demix
 true false
if true then demixer instead of average is performed

demix_freq_step
2
corresponds to .demixfreqstep in NDPPP .type=demixer

demix_time_step
2
corresponds to .demixtimestep in NDPPP .type=demixer

demix_sources
CasA

select_nl
 true false
if true then only Dutch stations are selected

parset
lba_npp

Submit workflow

Close

Figure 4: The initial pipeline selection form, presented by the portal after the selection of a measurement set.

To inform the portal of the parameters of each workflow, a template library was created¹⁷ that uses a simple JSON file to generate the appropriate parameter form for each pipeline in the portal. This allows new pipelines to be added to the portal without the need of any web development skills.

When submitting a pipeline, a request must be send to the LOFAR LTA to copy the data from the tape archive to temporary storage (so called staging), as shown in D4.1, Figure 6 (page 27). This has not yet been implemented in the current prototype portal, Therefore the staging must be triggered manually *before* submitting the pipeline.

¹⁷ https://github.com/EOSC-LOFAR/LOFAR_pipeline_template

Use Case 2: Square Kilometer Array / LOFAR

Once a pipeline is submitted, it will retrieve the data from temporary storage and process it. Currently, this processing does not yet take place on PROCESS infrastructure. Instead, the SURFsara Grid cluster (located in Amsterdam) is still used, as was done in the original implementation of the LGPPP pipeline.

We currently have two pilot pipelines available. The LGPPP pipeline (described above), which is already integrated into the portal, and a separate *prefactor-CWL pipeline*¹⁸ developed for the EOSC Pilot for LOFAR project. The advantage of LGPPP is that it is based on production codes and offers more complete calibration of the data. Its downside is that it is designed to run on the SURFsara Grid environment and is not easily portable to other infrastructures.

The *prefactor-CWL pipeline* was developed from scratch to be portable. It offers a pipeline implementation that uses containers (both Docker and Singularity) for portability and uses the common workflow language (CWL)¹⁹ as a workflow language to connect components together. However it offers a less complete calibration than LGPPP and has not yet proven itself in practice.

As an initial benchmark, we have selected two datasets:

- A 387 MB dataset (L570745) which is small enough to use in software development and offline demos (which typically takes place offline on laptops).
- A 114 GB dataset (L429550) which is small enough to use for initial experiments on PROCESS infrastructure.

We have processed the 387 MB dataset offline, using the *prefactor-CWL pipeline* to perform an initial calibration of this data, including some averaging and flagging of the calibrated solution. Flagging masks the data polluted by terrestrial interference or corrupted by runaway calibration solutions. The 114GB dataset was processed using the LGPPP pipeline via the PROCESS portals, but still using the existing SURFsara Grid environment.

2.7 Challenges

The major challenge during the first year of the projects was the tight integration between the current implementation of the LOFAR pipelines and the infrastructure it currently runs on (both the software and hardware component). This makes porting the workflows to PROCESS infrastructure non-trivial. We therefore decided to focus on the user portal first. In parallel, a more portable version of the pipeline was developed, the *prefactor-CWL pipeline* (described above), which we will most likely use in our first experiments on the PROCESS infrastructure.

2.8 Outlook

The next steps in this use case will be the integration with the PROCESS infrastructure and services and adding additional processing pipelines. To achieve this, we can directly use the *prefactor-CWL pipeline*. Additionally, we may also adapt the LGPPP processing pipeline to no longer depend on the SURFsara grid infrastructure, and containerize it using singularity to be portable. A start has already been made with the latter.

Next, we can start integrating the pipelines with the PROCESS services. Initially, we will focus on the compute service and assume the data is already available on site. This can easily be done by using a small test dataset, such as the L570745 or L429550 datasets mentioned above.

¹⁸ <https://github.com/EOSC-LOFAR/prefactor-cwl>

¹⁹ <https://www.commonwl.org/>

Use Case 2: Square Kilometer Array / LOFAR

To integrate into the PROCESS compute service, the current portal implementation needs to submit its pipeline submissions to the Model Execution Environment described in D4.1 (Section 3.2, page 57). This Model Execution Environment will then start the pipeline on the PROCESS compute infrastructure, monitor its status and return the results to the portal.

Once the pipelines can be deployed using the compute services, we will shift our focus to integrating the LOFAR LTA archive into the data services. Although relatively straightforward from a software engineering perspective, the problem here mainly lie with the network infrastructure between the sites. The amount of data that need to be moved are very large, and therefore high bandwidth connections are essential. We will investigate the option of using the PRACE network infrastructure for this, as it already connects LRZ, Cyfronet and SURFsara resources, and uses gridFTP for data transport which is also used by the LOFAR LTA.

Once both the compute and data services are integrated, we can proceed to parallelize the processing pipeline. This can be achieved by splitting a single 16TB observation into smaller chunks representing different wavelength. Initially these chunks can be processed independently, although the results must be integrated further along the pipeline.

The micro-infrastructure of the data service can support parallelisation of these pipelines. There are two options for this: the data can be distributed over multiple sites (provided all data is available at the same time) allowing it to be processed in parallel. Alternatively, data can be transferred one chunk at a time, as soon as it has been read from tape. This allows the data to be processed in a streaming fashion, reducing the overall wait time for the user.

3 Use Case 3: Supporting Innovation on global disaster risk data

3.1 Overview

The role of Use Case 3 is to demonstrate that the PROCESS platform can support the new, emerging users of extreme data services. As such, in terms of technological features and performance requirements, the use case hasn't presented any specific challenges to the platform. Due to the dramatic organisational changes in the UNISDR²⁰ working processes, the key has been to engage with the data owners and organisations acting as potential conduits to users in the disaster risk/emergency response domain.

As part of this charting of the future requirements, the project has contacted the following actors relevant to the Use Case 3:

- CIMA research foundation (mandated by UNISDR)
- UNOSAT
- WMO Hydrohub
- Citizen Cyberscience Centre

The technical activities have concentrated on two streams of activity: supporting CIMA in their high-resolution flooding risk mapping pilot (Sub-Saharan Africa pilot) and maintaining the UNISDR Global Assessment Report²¹ (GAR) flooding data portal as a mechanism to gauge baseline interest in the statistical flooding risk data. Since its launch on September 12th 2017 the data portal²² has had over 9000 unique visitors.

3.2 Progress

The progress of the use case 3 has been limited to:

- Maintaining the existing proof-of-concept portal²³ to maintain contact with the current user community of disaster risk data, and
- Supporting the UNISDR-mandate flood risk modelling activities that complement the datasets available through the UNISDR portal²⁴. The modelling work was originally planned to be completed by Spring 2018, but has been delayed several times due to reasons unrelated to PROCESS project.

The interest of the target community in the pilot dataset is at the moment steady, but relatively low. The main reason for this is the change of the monitoring methods related to risk reduction activities of the member states. The planned change in the GAR approach that has been described in the ISESS conference paper "UNISDR Global Assessment Report - Current and Emerging Data and Compute Challenges"²⁵ seems to be taking longer than expected. For example, the 2017 UNISDR Risk Atlas²⁶ was based on the modelling data of the 2015 Global assessment report.

Thus, at the moment the use of data services is straightforward (the portal service), while the future use of the compute service for high-resolution modelling is unclear. Adapting the portal to the PROCESS services (once they are available) will be a straightforward task. The high-resolution modelling-related work does present two functional requirements that may require

20 <https://www.unisdr.org/>

21 <https://www.unisdr.org/we/inform/gar>

22 <https://gar.mnm-team.org>

23 <http://unisdr.mnm-team.org/>

24 <http://www.cimafoundation.org/cima-foundation/news/unisdr-africa.html>

25 https://link.springer.com/chapter/10.1007/978-3-319-89935-0_26/

26 <https://www.preventionweb.net/english/hyogo/gar/atlas/>

Use Case 3: Supporting Innovation on global disaster risk data

attention when deploying the PROCESS solutions and building the service management and user support approaches surrounding the technical services:

- The software solutions used by the emerging user communities may lack basic checkpointing functions and exhibit long runtimes even with high-end server solutions. In the high-resolution simulation work some of the jobs didn't complete within the 48-hour runtime limit imposed by the underlying infrastructure.
- The software used in the simulation generated a large number of files (order of millions), which can be a challenge for the underlying infrastructure.

3.3 Challenges

The main challenge in this use case is caused by the large organisational changes in the UNISDR working processes. This has caused a delay in modelling work that was originally planned to be completed by Spring 2018.

3.4 Outlook

The use case will likely need to deal with a relatively high degree of uncertainty in terms of engagement of the current primary user community. In case the Global Assessment Report process will kick off in 2019, the datasets that provided the foundations of the 2015 and 2017 editions will naturally be one of the key foundations of the 2019 work. As the Global Assessment Report process is planned to be opened up to a considerably larger number of research groups. A clear commitment by the UNISDR to the production of a new Global Assessment Report would be sufficient to both reach the planned KPIs as well as identifying interesting opportunities for data reuse and deployment of higher-level PROCESS services in the service of disaster risk reduction activities.

To mitigate risks related to scenarios where the UNISDR focus will remain outside the formal GAR process, the project will continue dialogue with the parties mentioned above, as well as proactively seeking other stakeholders interested in using and/or generating natural disasters related datasets. Cross-linking (and possibly integrating) these datasets with the existing portal will increase the visibility of the GAR dataset and the likelihood of new innovative approaches based on it. The project will also investigate opportunities to cross-linking and potentially reusing tools developed in the use case 5 with the GAR data portal.

The formal validation mechanism is based on the KPI 4 values, which are either very conservative (in case of the start of a new UNISDR GAR process in 2019) or somewhat challenging (in case they need to be met based on new collaborations with developers of new services with limited existing user base).

Use Case 4: Ancillary pricing for airline revenue management

4 Use Case 4: Ancillary pricing for airline revenue management

4.1 Overview

Ancillaries is in the airline world a broad term for any services that goes beyond simple transportation from A to B. Ancillary in this sense can be anything from being able to check-in an additional bag to booking an “Uber” that transports the customer from the airport to his hotel.

The goal of this use case is to derive a promising machine learning algorithm for pricing of offered ancillaries. As an important first step we will generate a standard ancillary sales data set for airlines. A more detailed description of this use case can be found in D4.1 (Use case analysis, Section 1.4, pages 36-42).

4.2 Progress

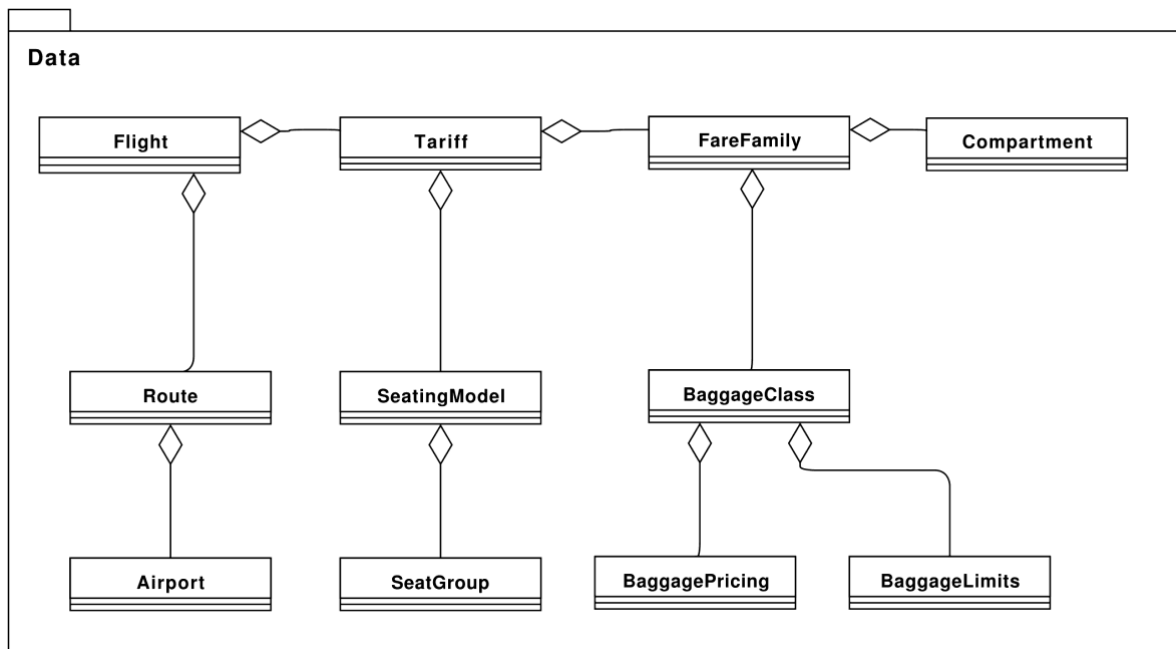


Figure 5: The airline setting generated by the data generator

During the first year we developed a data generator as a first step to implement our use case. This data generator creates bookings for flights with all the necessary information regarding baggage, seat information etc. Even though the used airports and flight routes can be built up on historical flight schedules (using SSIM files), the bookings are artificial itself and do not contain any personal data and are not related to existing passengers. This allows us to generate a standard ancillary sales data set, which can be used without violating passengers’ privacy or revealing airline business knowledge.

Several considerations were made for a corresponding airline ancillary data generator. On the one hand, the generator should be able to describe current airline ancillary offerings accurately, while on the other hand the generator should be able to generate data that fits more than only one airline.

Figure 5 depicts the class model for the ancillary generator. The two considered kinds of ancillaries, Seating and Baggage, are two of the most common ancillaries where the airline

Use Case 4: Ancillary pricing for airline revenue management

has full pricing control. Other ancillaries like travel insurance, hotel rooms, or rental cars are in general offered by third party companies and are not under pricing control of the airline. Tariffs are organized in Products, which are again organized in compartments (i.e. the Product “Super Economy” might belong to the compartment “Economy” and offer multiple tariffs).

As displayed in Figure 5 all Tariffs belonging to one Product which shares the same baggage classes defined by their prices as well as their size and weight limits. Seating models are attributes of tariffs and define how many seats of a seat group are available for which price. Flights are defined by their route, time of flight as well as a number of tariffs, which are bookable on this flight. From this list of offers the generator can construct artificial bookings by choosing a flight, a corresponding tariff, seats for all passengers, as well as a number of bags.

The source code for the generator is currently only privately available in the project’s Gitlab. The code requires Java 8 and Maven in version 3 or higher. The application uses Spring Data²⁷ for access to the data. Therefore, any kind of relational database can be used. For now a H2 database²⁸ is used. It is not needed to create a model schema explicitly. If none is present, the application will create one by itself.

The generator is configured by a yaml file that stores default values of generator options. It allows to set the amount of each generated entity as well as ranges for flight departures. This file can be also passed as first program argument during execution with corresponding parameter.

For huge airline network carriers about 50 million bookings can be expected over the course of a year and historic data of 2-3 years is required for proper data analysis. Based on the current implementation this translates to a H2 database of roughly 65GB per (simulated) year. This size may vary depending on the database used as well as some generation parameters.

The ancillary data generator uses the computing resources at Cyfronet. The next step is storing the computed data with the data management services provided by PROCESS (see Report D4.2, Section 2, pages 9-22).

4.3 Challenges

So far, no mayor challenges were encountered in this use case.

4.4 Outlook

With the base of the data generator implemented, we will next focus on storing the data within the PROCESS environment directly in Prometheus. Furthermore, we will scale the data, i.e. generating the standard ancillary sales data set with realistic amounts of data for two or three simulated years.

Afterwards, as described in D4.1 (Use case analysis, Section 1.4, pages 36-42) we will evaluate some machine learning algorithms provided in the PROCESS environment with the generated data.

²⁷ <http://projects.spring.io/spring-data>

²⁸ <http://www.h2database.com>

5 Use Case 5: Agricultural analysis based on Copernicus data

5.1 Overview

Use Case 5 was presented in detail in D4.1 (Use case analysis, Section 1.5, pages 43-46). Based on Sentinel satellite data sets from the Copernicus project²⁹. This use case will add an end-user solution for accessing such large data collections from the PROCESS framework. The use case application itself uses image and radar files to detected changes in the agricultural usage of the land.

5.2 Progress

The pre-processing software specified in D4.1 (Section 1.5.5, page 44) is still in development and was not executed on a PROCESS resource so far. Therefore, it could not be used as a pilot use case for the available infrastructure. In the last month, the software was enhanced (including containerization) and that interim version is already executed on local HPC resources.

The workflow components will need to fetch large data sets from the Copernicus service and store it to a PROCESS storage resource. So far, all requirements are met by the proposed and PROCESS architecture. From the storage resource, the data needs to made available at the computing resource. The selected data sets are before specified by a user within the interactive portal. The source code of the PROMET software needs to be available on the designated computing resource.

The workflow components will be written in python and C++. Both are usually available at all included computing centers and all following. All dependencies will be met with self-compiled libraries packed with the application.

5.3 Challenges

The main challenge was the delay in the development of the pre-processing software (which is being created by the PROMET developers). As a result, integration into the PROCESS platform was delayed, as was the work to pre-process the data for 3 different earth system models.

5.4 Outlook

The release of the running workflow on a PROCESS resource is expected in the first half 2019. The input size for one computation will include approximately 10-20 terabyte in a first stage, but will be extended to several hundred terabytes at the end of the project.

²⁹ <https://scihub.copernicus.eu>