



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777533.

PROviding Computing solutions for ExaScale Challenges

D2.2	Progress report (UC#1-5) - update		
Project :	PROCESS H2020 – 777533	Start / Duration:	01 November 2017 36 Months
Dissemination¹:	PU	Nature²:	R
Due Date:	31st October 2019	Work Package:	WP 6
Filename³	PROCESS_D2.2_ProgressReport-update_v1.0.docx		

ABSTRACT

This deliverable provides an overview of the development at the second year of the PROCESS project. The progress of use cases is described in relation to the previous deliverable. Significant progress has been made in the second year of the project, with KPIs being reached. In the next months, the use cases will focus on finalizing and exploiting the full integration with the PROCESS infrastructure.

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.

Deliverable Contributors:	Name	Organization	Role / Title
Deliverable Leader⁴	Müller, Henning	HES-SO	Coordinator
Contributing Authors⁵	Müller, Henning; Graziani, Mara	HES-SO	Writers
	Spreeuw, Hanno; Maassen, Jason	NLESC	Writers
	Höb, Maximilian; Schmidt, Jan; Heikkurinen, Matti	LMU	Writers
	Reichardt Janek; Pancake-Steeg, Jörg	LSY	Writers
Reviewer(s)⁶	Belloum, Adam; Cushing, Reggie	UvA	Reviewer
	Guggemos, Tobias	LMU	Reviewer
Final review and approval	Höb, Maximilian	LMU	Coordinator

Document History

Release	Date	Reasons for Change	Status⁷	Distribution
0.1	28.06.2019	Initial version	Draft	
0.2	15.09.2019	Section overviews	Draft	
0.3	16.10.2019	Section details	Draft	
0.6	28.10.2019	Internal review	In Review	
1.0	31.10.2019	Final version	Released	Public

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	3
Executive Summary	4
List of Figures.....	5
List of Tables	5
1 Introduction	6
2 Use Case 1: Exascale learning on medical image data.....	7
2.1 Status at M12.....	7
2.2 Progress.....	7
2.3 Challenges.....	9
2.4 Outlook.....	9
3 Use Case 2: Square Kilometer Array / LOFAR	10
3.1 Status at M12.....	10
3.2 Progress.....	10
3.3 Challenges.....	13
3.4 Outlook.....	14
4 Use Case 3: Supporting Innovation on global disaster risk data	15
5 Use Case 4: Ancillary pricing for airline revenue management	17
5.1 Progress.....	17
5.2 Challenges.....	19
5.3 Outline.....	19
6 Use Case 5: Agricultural analysis based on Copernicus data	20

Executive Summary

The objective of this deliverable is to provide an overview of the progress made with the use cases in the second year of the PROCESS project. More specifically, it is about how the use cases have progressed in relation to the Progress Report described in D2.1.

The use cases with direct developer involvement (UC1 and 2) have progressed steadily to run experiments and monitor performance on the infrastructure at Cyfronet in Krakow, Poland and the infrastructure at the University of Amsterdam (The Netherlands). Use case 4 made progress with its installation in the Slovakian University of Informatics (UISAV) environment. The use cases 3 and 5 focused on engaging existing communities, for which the progress is reported in this document.

The first use case, *Exascale learning on medical image data*, started from the first benchmark results on PROCESS infrastructure running at Cyfronet in Krakow, Poland. The progress at the time of writing shows cutting the computational time in half in comparison with running computations in loco.

The second use case, *Square Kilometer Array / LOFAR*, reused and extended the Web interface developed in EOSCPilot project and completed the implementation of the target pipeline defined in D4.1. The next step will focus on optimising the pipeline and integrating it with IEE.

The third use case on supporting innovation for global disaster risk data showed that communities have unique requirements on how their data should be made available and presented to the public. A new solution is proposed in this deliverable to address the problematics that its analysis raised at the beginning.

The fourth use case, *Ancillary pricing for airline revenue management*, generated a prototype data generation tool which is running at UISAV in Bratislava, Slovakia. Based on the generated data, we implemented a price calculator that provides revenue-optimal prices for the “first bag” ancillary, given several parameters of the request.

The fifth use case, *Supporting Innovation on global disaster risk data*, and *Agricultural analysis based on Copernicus data*, is starting to be actively approached, aiming at expanding the user base of the PROCESS platform.

Overall, important progress was made in the second year of the project, with KPIs being reached. In the coming months, the use cases will focus on finalizing and exploiting the full integration with the PROCESS infrastructure.

List of Figures

<i>Figure 1: Intermediate visualizations of heatmaps of cancer cell probability (with colormap from blue for low probability to orange for high probability) on the WSIs, compared to previous annotations (red borders).</i>	<i>8</i>
<i>Figure 2: Visualizations of feature importance inside the network training as heatmaps of network attention on the input images with Gradient Class Activation Mapping, GradCAM [1].</i>	<i>8</i>
<i>Figure 3: The prototype of the measurement set selection portal. This portal allows.....</i>	<i>11</i>
<i>Figure 4: The pipeline selection form, presented by the portal after the selection of a dataset.</i>	<i>11</i>
<i>Figure 5: The pipeline configuration form, parameters that work with default values are folded by default.....</i>	<i>12</i>
<i>Figure 6: Output before and after calibration.</i>	<i>13</i>
<i>Figure 7: Using machine learning models to predict how the probability of buying baggage depends on the ticket price as well as the time of ticket booking for a fixed first bag price.</i>	<i>17</i>
<i>Figure 8: Functions describing how the purchase probability depends on the first bag price. Linear dependency in yellow and Logistic curve in black.....</i>	<i>18</i>
<i>Figure 9: UC5 soil moisture simulation in South America.....</i>	<i>21</i>

List of Tables

<i>Table 1: CamNet: development status</i>	<i>8</i>
<i>Table 2: running time of the datagenerator.....</i>	<i>19</i>
<i>Table 3: running time of the machine learning training.....</i>	<i>19</i>

1 Introduction

In this document we provide an overview of the progress made by the use cases in the second year of the project, and relate this progress to the KPIs. The use cases will report about the first prototype implementation and about the progress made thanks to this.

At this point in time, all the use cases make use of the hardware infrastructure available in PROCESS. Use cases 1 and 2 make use of the infrastructure in AGH and UvA. Use case 4 was integrated with Cloudify at UISAV. The TOSCA template and the corresponding set of deployment/execution scripts for the use case were developed and deployed on the Cloudify manager. Via a simple Cloudify REST API, users can deploy and execute jobs of UC#4 on IISAS-FedCloud site at UISAV, which is a part of EOSC-Hub Federated Cloud infrastructure. The Cloudify REST API is used for integrating the use case to IEE graphical portal and users now can execute UC4 jobs in Cloud via IEE.

The development has reached the target KPIs for month 24. In particular, the target KPIs 1 and 2 were reached by UC1 with the simulation of larger dataset sizes and an observed improvement over the baseline of approximately 9%. The pricing calculator for first bag pricing is currently in place, reaching KPI 3 for UC4.

In the remainder of this document, each use case gives a detailed report on its progress, elaborates on problems that were encountered, and provides an outlook for the next 24 months. The use cases are not 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 Status at M12

The pilot application for the use case was developed to tackle cancer detection and tissue classification on the Camelyon dataset⁸ (Camelyon16 and 17). Camelyon is still the largest and the most challenging dataset for histopathology research, with more than 1000 tissue Whole Slide Images (WSIs).

In D2.1 we introduced Camnet, a three-layer software architecture. The first layer preprocesses the raw BIGTIFF WSIs. 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. The third layer implements performance boosting and interpretability techniques. After twelve months from the start of the project, the development covered almost fully the functionalities of the first two layers. In the following, we report the progress made since then.

2.2 Progress

The initial prototype used random sampling of locations in the WSIs to extract patches from the WSIs sequentially. The first improvement was made to Layer I. We introduced the parallel processing of the data and the integration with the SLURM (Simple Linux Utility for Resource Management) system. The data preprocessing functionality of Layer I with random sampling is now handled by a SLURM job array where each task is configured with a different random generator seed. At each batch run a different patch set is extracted. Thus, the execution of 10 parallel tasks allows the extraction of 10 times the number of patches that were previously extracted. By running 10'000 processes at the same time, we observed a 1'000 times speed up, namely the collection of 5'000 patches in less than 5 minutes. Scaling is therefore possible by increasing the number of CPU nodes and patients, as shown in Figure 1 of D8.1.

A new sampling modality is available, which densely covers the WSI content. For instance, we extract patches by passing a sliding window on the whole WSI. This option was not possible with our local machines due to the computational and memory limitations. The new functionality uses a SLURM job array where each task is assigned a different WSI (a single "patient case") while the random seed is the same. On a testing batch of 5 patients (5 WSIs) we are now able to generate ~1TB of data in less than 2 hours, hence ~200 GB per patient. Note that patches with non-relevant information (e.g. white content, black pixels, background, etc.) are 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 with the pixel values in a set of intermediate HDF5 databases (one file per each batch).

As a further scaling step, we combine the two previous techniques by a SLURM job array where each task is assigned a different WSI (a single "patient case") and M parallel subtasks are run in parallel (the random generator seed is varied over M parallel subtasks). This double parallelization further reduced the time down to 1 hour for generating 1TB of data from 5 patient slides. Hence, under the hypothesis of infinite resources (i.e., available CPUs), this script achieves linear speed-up of #slides x #CPUs available.

The preliminary results of porting the network training workflow to AGH, Poland, shows nearly a 10% improvement in the validation performances (see D8.1, Table 3, page 9). The improvement is obtained with intense augmentation of the dataset of intermediate patches.

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

D2.2: Use Case 1: Exascale learning on medical image data

The parallel training on multiple GPUs with Docker was introduced in Layer II of the software. Results highlight at least a 20% save up in time by the parallelization on two NVIDIA K80, as shown in Table 5 of D8.1. One of the challenges for the future work is to parallelize the network training layer e.g. Layer II on the HPC clusters.

Layer III was substantially improved, with the initial generation of intermediate visualizations (see Fig. 1) and with analysis of feature importance inside the network (in Fig. 2).

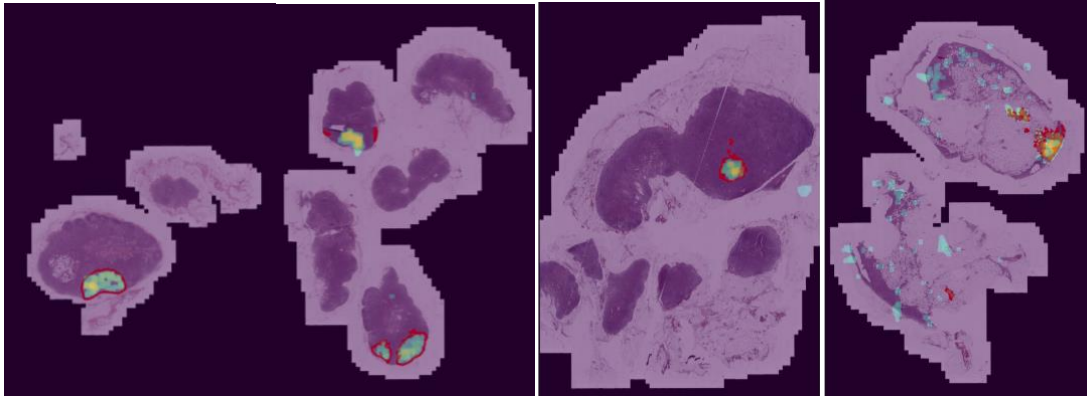


Figure 1: Intermediate visualizations of heatmaps of cancer cell probability (with colormap from blue for low probability to orange for high probability) on the WSIs, compared to previous annotations (red borders).

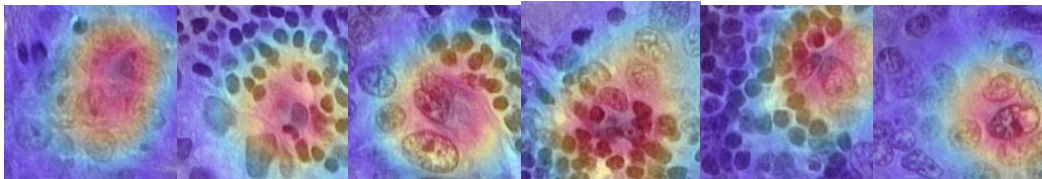


Figure 2: Visualizations of feature importance inside the network training as heatmaps of network attention on the input images with Gradient Class Activation Mapping, GradCAM [1].

Table 1 summarizes the progress in each of the software layers.

Layer I: data preprocessing and patch extraction	Layer II: local and distributed training	Layer III: Performance boosting and interpretability
Creation of normal and tumor tissue masks -- completed	Interchangeable network models -- 5 models available	Generation of intermediate visualizations -- heatmaps at the WSI and patch level
Sampling of patches at high resolution -- random and dense sampling completed	Local training -- benchmarks on Titan V, Titan X, V100, Tesla K80	Feature importance analysis -- gradCAM visualizations
Creation of H5DS database of patches -- up to ~200Gb of data per WSIs	Porting to HPC clusters -- benchmarks on Tesla K40	Perturbation robustness analysis
	Distributed training -- local scaling 2x Tesla K80	

Table 1: CamNet: development status

2.3 Challenges

An ideal configuration for running the use case pipeline was identified in Sec. 2.1.3 of D8.1., namely of running Layer I at AGH, Poland, and subsequently transfer the data to UvA, Amsterdam to execute Layer II. Running Layer I at AGH would in fact allow to use the extreme scale up in data extraction that we reached with the parallelization of the scripts. However, data transfer becomes more and more demanding as the size of the datasets increases. Considering that we are now able to extract up to 200 Gb per patient, transferring the data from AGH to UvA is one of the main challenges arising. Moreover, the FTP transfer protocol is not always possible in some infrastructures, which limits the transfer to the SCP protocol.

2.4 Outlook

Our coming work will focus on the challenges of training an exascale network that should process the amount of data of 10TB of data. We also plan to expand the performance boosting and interpretation layer of the architecture (Layer III).

3 Use Case 2: Square Kilometer Array / LOFAR

3.1 Status at M12

The goal of this use case is to build an easy to use and portable data reduction pipeline of archived LOFAR observations for astronomers. Following our thorough requirements analysis, an approach combining containerized workflows and Web interfacing has been decided.

Through the web interface, 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).

After requirements analysis from M1-M6, the interval M6-M12 was dedicated to developing containers for the different components of the reduction pipeline, mainly the so-called direction independent calibration ones. Some time has also been given for preparing the Web interface initially developed within the EOSCPilot⁹ AA-Alert¹⁰ projects for reuse in PROCESS. Indeed, the only pipeline then available was tailored made for specific hardware infrastructure and thus cannot be run on PROCESS infrastructure. Progress made during the second year is described in the next section.

3.2 Progress

First, the Web interface was improved with an additional selection criterion allowing the astronomer to specify an observation ID. Consequently, in addition to the usual fields specifying the observation(s) time period, the patch of the sky and the number of frequency subbands, the user can also search for very specific observation using its ID in the supporting database. Furthermore, when running a pipeline, the user may have to fill in several parameter values; consequently, to minimize user input, the interface is re-worked to only show the parameters which require settings, the other being hidden by default. However, all parameters can be modified if needed. The prototype of this web interface is shown in Figure 3.

⁹ <https://github.com/EOSC-LOFAR/ltacat>

¹⁰ <https://www.esciencecenter.nl/project/aa-alert>

D2.2: Use Case 2: Square Kilometer Array / LOFAR

PROCESS UC#2: SKA/LOFAR

The goal of this use case is to simplify the processing of archived data. 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 workflow infrastructure for processing of extremely large volumes of astronomical observation data.

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.

	OBSERVATIONID	STARTTIME	ENDTIME	RIGHTASCENSION	DECLINATION	NR_SUBBANDS
	<input type="text" value="Enter OBSERVATIONID"/>	<input type="text" value="Enter STARTTIME"/>	<input type="text" value="Enter ENDTIME"/>	<input type="text" value="Enter RIGHTASC"/>	<input type="text" value="Enter DECLINATION"/>	<input type="text" value="Enter NR_SUBBANDS"/>
	50512	2012-03-03T04:55:03.000Z	2012-03-03T04:55:03.000Z	299.867916667	40.7339138889	80
	50509	2012-03-03T04:28:02.000Z	2012-03-03T04:28:02.000Z	311.42857125	90	1
	50508	2012-03-03T04:25:01.000Z	2012-03-03T04:25:01.000Z	299.867916667	40.7339138889	80
	51155	2012-03-09T15:13:00.000Z	2012-03-09T15:13:00.000Z	71.42857125	0	80
	51155	2012-03-09T15:13:00.000Z	2012-03-09T15:13:00.000Z	72.580645	11.54	80
	51155	2012-03-09T15:13:00.000Z	2012-03-09T15:13:00.000Z	123.400216667	48.217295	80
	51155	2012-03-09T15:13:00.000Z	2012-03-09T15:13:00.000Z	71.42857125	90	1

Figure 3: The prototype of the measurement set selection portal. This portal allows

Second, a new pipeline that the user can directly launch once a dataset has been selected, is made available alongside the existing LGPPP¹¹ pipeline. This is illustrated in Figure 4 below. Although it is called LOFAR_PREFACTOR_pipeline for the name of the LOFAR pre-processing pipeline, it actually implements all the components as described in D4.1, Section 1.2.5, page 26.

Observation overview

Product Parameters

Observation ID	456100
Start time	2016-04-30T03:54:01.000Z
End time	2016-04-30T03:59:21.000Z
Right Ascension	299.8681525
Declination	40.7339155556
Nr subbands	30

Data Processing

E-mail address:

Job description:

Select processing pipeline:

- LOFAR GRID Pre-Processing Pipeline
- PREFACTOR LOFAR pipeline

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

¹¹ https://github.com/EOSC-LOFAR/LGPPP_LOFAR_pipeline

D2.2: Use Case 2: Square Kilometer Array / LOFAR

Like most pipelines, the new pipeline has a number of configuration parameters which may be set by the user, such as those needed for data transfer from LOFAR LTA, those relative to the selected computing site. Once these parameters are set to the satisfaction of the astronomer, the workflow can be launched directly from the portal.

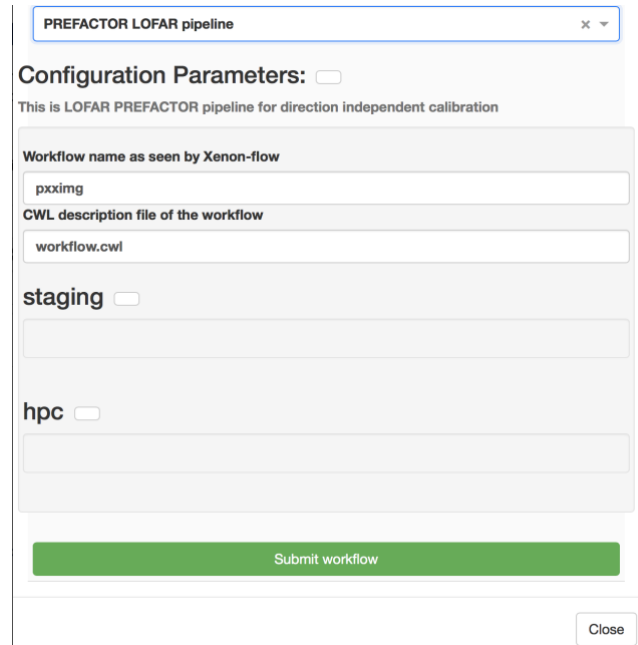


Figure 5: The pipeline configuration form, parameters that work with default values are folded by default.

When submitting a pipeline, a request must be sent 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 is implemented using the data services provided by LOBCDER,

Once a pipeline is submitted, it will both stage the data and retrieve the data from temporary storage and process it. Although we have shown the processing can run on any PROCESS computing infrastructure, including Prometheus in Krakow and CoolMUC in Munich, it runs currently by default on DAS5 in Amsterdam.

The PREFACTOR_pipeline¹² is an instantiation of the pipeline template¹³ and API¹⁴ developed for the EOSC Pilot for LOFAR project for the aim of allowing easy implementation of new pipelines. The pipeline implementation uses Xenon-flow¹⁵ to describe the work to be done as a common workflow language (CWL)¹⁶ workflow. The main steps of the workflow correspond to the commands required for executing a typical data reduction pipeline and are invocations of our Singularity containers. Various CWL files have been defined¹⁷. Once the workflow is specified, Xenon-flow relies on Xenon¹⁸ middleware for the actual submission and run of the specific steps.

¹² https://github.com/process-project/PREFACTOR_pipeline.git

¹³ https://github.com/EOSC-LOFAR/LOFAR_pipeline_template.git

¹⁴ https://github.com/process-project/lofar_workflow_api.git

¹⁵ <https://github.com/xenon-middleware/xenon-flow.git>

¹⁶ <https://www.commonwl.org/>

¹⁷ <https://github.com/process-project/PREFACTOR-XENON-CWL.git>

¹⁸ <https://github.com/xenon-middleware/xenon.git>

D2.2: Use Case 2: Square Kilometer Array / LOFAR

As an on-going benchmark, we have selected two datasets:

- A 25 GB dataset (L232873) for the calibrator, which is small enough to use in software development and realistic enough to serve as a demonstrator for UC2.
- A 433 GB dataset (L232875) for the target source, which is small enough to use for initial experiments on PROCESS infrastructure.

We followed the steps of typical reduction pipeline for producing images, which consists in pre-processing the data, including some averaging and flagging of bad data, in calibrating the calibrator, then calibrating the target using the solutions from the previous step. The pipeline up to this step is run using the container defined in the first year providing the direction independent calibration through PREFACTOR. Then, to check and validate the processing up to this step, we added two imaging steps, one before the beginning of the processing before any calibration and the second after direction-independent calibration. The outputs of both steps are shown side-by-side in Figure 6.

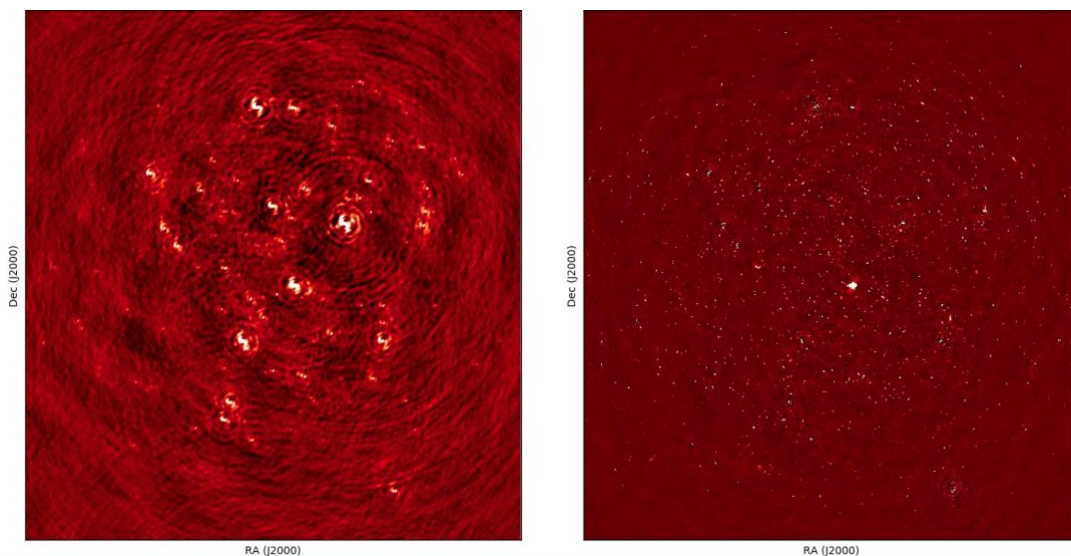


Figure 6: Output before and after calibration.

To get the images back into the portal, we also implemented an ad hoc Python step responsible for converting the actual FITS images produced by the direction independent calibration step into JPEG format that can easily be shown on the Web. There is an approach with Javascript showing interactively the actual FITS images but, of course, this is unwieldy.

Third and lastly, we completed the data reduction pipeline with direction dependent calibration using ddf-pipeline¹⁹. This pipeline performs several rounds of self-calibration and imaging to produce high-quality images of the sources in the target patch of the sky.

3.3 Challenges

One of the challenges during the second year of the project was the implementation of a LOFAR pipeline which would run on PROCESS infrastructure given that the then-only pipeline, LGPPP, is tightly integrated to SURF Sara systems. Another challenge was how to actually reach PROCESS computing sites from the Web interface once we discard the previous solutions. This is how Xenon-flow and Xenon came into the figures again as the latter was already in the initial PROCESS proposal and the former is just CWL wrapper around it. The last challenge is relative to containerizing the direction dependent calibration as all available

¹⁹ <https://github.com/mhardcastle/ddf-pipeline.git>

algorithms are heavy-weight software, complex to setup and run, lacking clear documentation or requiring expert knowledge.

3.4 Outlook

The next steps in this use case will be fourfold.

First, reduce manual work in running the pipeline. Currently, substantial manual work is required to adapt various configuration files to the selected observation datasets. This can easily be done using our approach based on Xenon-flow by inserting intermediary Python steps in the CWL description.

Second, continue working on containerizing a more reliable direction dependent calibration algorithm. One direction we are looking into is work done in the distributed radio astronomical computing (DIRAC) project where the more robust calibration software, SAGECal²⁰, is being pipelined with imaging steps into a product similar to ddf-pipeline.

Third, make the pipeline process multiple observations on multiple computing sites at the same time. While the pipeline is shown to be capable of reducing observations from any LTA location on any of the currently available computing sites, it does not do so on multiple sites in parallel. Although this capability is built in PROCESS architecture design and the pipeline implementation, it has not been tested so far. The problem here mainly lies 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 PRACE²¹ (PaRtnership for Advanced Computing in Europe) 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.

Fourth and finally, the adapted EOSC portal needs to be integrated into PROCESS ecosystem. Discussion has been launched with Cyfronet on ways of achieving this. Beyond the Web interface, it would be desirable to have the current pipeline submission infrastructure composed of Xenon-flow and Xenon either interact with IEE and Rimrock or have their equivalent functionality into IEE.

²⁰ <https://github.com/nlesc-dirac/sagecal.git>

²¹ <http://www.prace-ri.eu>

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

Use Case 3 aimed at supporting innovation based on global disaster risk data in close collaboration with UNISDR that was recently renamed "UN Office for Disaster Risk Reduction (UNDRR)". In addition to change of the brand, the organisation remodelled its operations and restructured its internal departments, shifting the focus from offering in-house modelling expertise to developing a policy forum similar. In additions to delays due to ramp-up phase, this resulted in the loss of contact with the key persons of UNISDR, as already reported in the deliverable D2.1 (section 3.3).

As attempts to gain a visible position in the UNDRR landscape (e.g. by approaching research organisations active in the "new global risk assessment" model) were not fruitful, the PROCESS executive meeting in September 2019 decided to shift its focus away from the UNISDR data sets. The use of legacy datasets was deemed to be of little or no value: the dataset does not provide a meaningful technical challenge for the PROCESS platform, nor is it expected to lead to any meaningful reuse of the disaster risk datasets by new users and user communities. The solutions related to the high-resolution datasets that are currently of interest for UNDRR and other organisations are not yet mature enough to support "long tail of science" approach with the raw data, and attempts to serve specialists would duplicate efforts of other projects the consortium partners are in contact with. Nevertheless, while the specific community engagement/validation of the solution through observation of independent, "long-tail of science" efforts is not possible in the UNDRR context, the technical development made by the PROCESS for these purposes will be exploited in a different way.

The main aim of Use Case 3 is to "enable more efficient data management of [...] data in the face of increased amounts of data that will be produced in a more dynamic fashion by a distributed, heterogeneous collaboration". To do so in a more generic manner, we developed a containerized and generalisable approach with a user friendly web-portal, which can be deployed on any storage site to expose data sets together with their meta data description. This approach was already reported in D2.1 - Section 3.

Supported by additional meta-data modules from PROCESS, scientists and communities are able to easily make their data sets available to the broad public. This encourages the goals of the open data initiatives and contributes to support reproducibility of computations made with those data sets. As a vision, this could be seen as extending the workflows supported by PROCESS to automating publishing the results as an open data product - a step towards a "FAIR data in a container" module.

The Use Case itself has shown that different communities have unique requirements on how their data should be made available and presented to the public. With this containerized module that has initially been developed to expose the UNISDR/UNDRR data sets, we can develop an additional building block to the PROCESS software stack that enables other communities to customize the publication of their data sets to their needs.

The developed solution consists of a containerized web-portal that can easily be configured and integrated with the PROCESS ecosystem. The deployment of the container will be integrated with the PROCESS data service LOBCDER. Exploiting its capabilities of data placement the container can be efficiently deployed near the data that the projects want to publish. Therefore the user communities only need to focus on their desired presentation of the data.

D2.2: Use Case 3: Supporting Innovation on global disaster risk data

This container will be integrated with the IEE as an optional feature for every project in a way that makes it easy for the user to enable this feature and customize the presentation within the container. Once the customization is completed, the configured container will then be deployed by LOBCDER and the user will get access to the exposed web-portal that can be made available to their community.

This solution can and will be used for different projects and communities. PROCESS is especially in contact with another EU-funded project, which is very interested in the offered solution to expose their data from tape archives to make them public. This new collaboration will be initiated by the end of 2019 with a goal of formalising it in an MoU. The validation of the ease of use features can be done and demonstrated considerably more convincingly based on the data sets and the new user communities accessible through this new collaboration.

While the PROCESS consortium regrets that the direct collaboration with UNISDR/UNDRR could not produce the expected outcome regarding the risk data sets, we expect the new approach to provide results that are of at least equal value. The PROCESS project will continue to disseminate the technical solution to other communities following the sustainability aim for all its developed modules - including the research organisations actively supporting the work of UNDRR in its current form.

5 Use Case 4: Ancillary pricing for airline revenue management

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. In particular we will concentrate on the purchase of a first checked-in bag in addition to hand luggage which is contained in purchased fare, called “first bag”.

5.1 Progress

In the past year we implemented a pricing calculator for first bag pricing using the machine learning framework h2o.

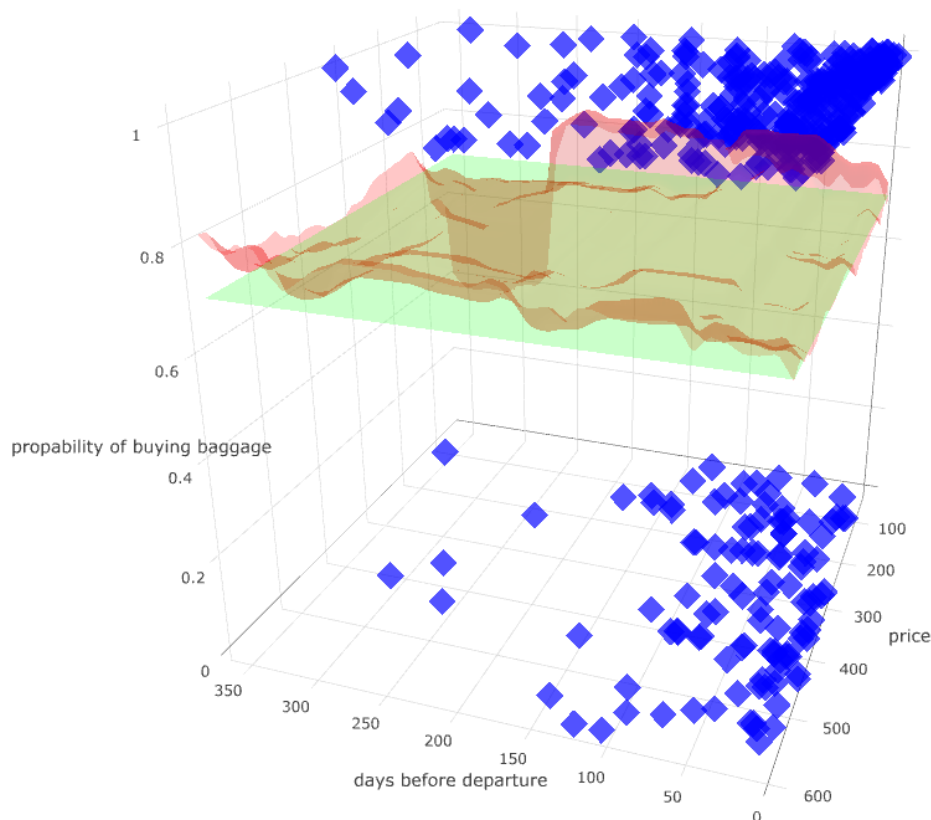


Figure 7: Using machine learning models to predict how the probability of buying baggage depends on the ticket price as well as the time of ticket booking for a fixed first bag price.

Figure 7 shows an early example of how we used h2o machine learning methods. Note that we always assume the same fixed price for a first bag purchase as this is the current state in the airline business. The blue points display some of the training data, hence they either have a probability of zero (no first bag purchased) or one (first bag purchased). The red graph interpolates the prediction of the random forest while the green graph corresponds to the prediction of the neural network.

D2.2: Use Case 4: Ancillary pricing for airline revenue management

In the next step we improved the accuracy of our predictions by using more independent variables such as the location from where the ticket was booked or the booking channel. The next intermediary step was to assume how the purchase probability depends on the bag price at each point in time. One way to do this is to predict the purchase probability for the observed price (15€ in the example below) using one of the trained machine learning models and to fit a curve through this point. In the example below (Figure 8) this was done using a linear dependency (yellow) as well as a logistic curve (black).

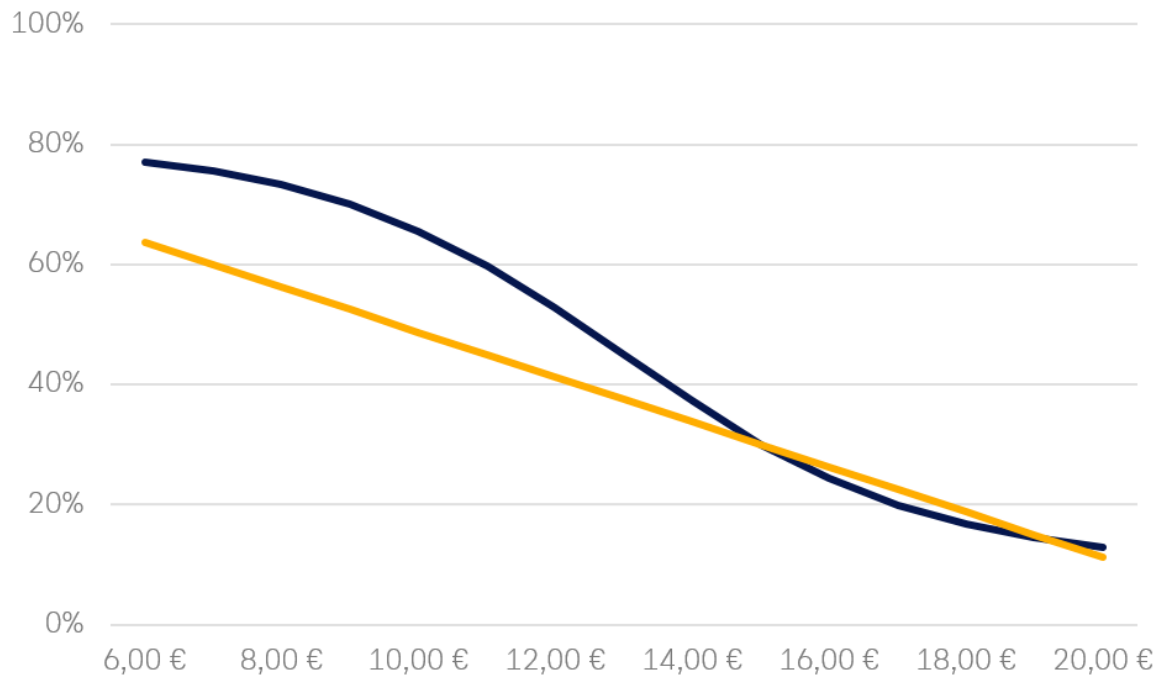


Figure 8: Functions describing how the purchase probability depends on the first bag price. Linear dependency in yellow and Logistic curve in black.

Afterwards we used those results to set up a mathematical function that models how the expected monetary value of a customer depends on the bag prices from the point of their booking until departure. The resulting nonlinear optimization problem is then solved using for instance the BOBYQA algorithm²² or a gradient optimization method which are implemented in the math component of the apache commons project.

Further research has been made in the area of checkpointing that is creating a *new* machine learning model using *new* data based on an *old* already trained machine learning model. It is supported in the h2o framework. This would allow us to update the models with incoming new data in regular intervals. One way to define these intervals could be to set a threshold for some quality measure, a simple example for such a rule could be: “retrain if the mean absolute error rises above 0.2”. However we discovered some impediments that suggests that a complete new initialization from time to time may be more suitable for our use case than using the checkpointing framework.

Additionally we made some tests analysing the running time of training the models and the running time of the datagenerator for various amounts of data, both on .

²² http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf

D2.2: Use Case 4: Ancillary pricing for airline revenue management

Records generated	Generation time [ms]	Generation time [min]	Records generated / s
10,000,000	20031	0.33	499,226.20
20,000,000	51032	0.85	391,910.96
50,000,000	107346	1.79	465,783.54
100,000,000	147153	2.45	679,564.81
200,000,000	279626	4.66	715,241.07
250,000,000	351051	5.85	712,147.24
300,000,000	415639	6.93	721,780.20
350,000,000	485307	8.09	721,192.98

Table 2: running time of the datagenerator.

Records amount	(Baggage bought)	(Baggage bought)
	Random forest model	Deep neural network model
10,000,000	0:00:05	0:01:47
20,000,000	0:00:07	0:06:22
50,000,000	0:00:08	0:14:38
100,000,000	0:00:08	0:31:32
200,000,000	0:00:09	1:04:48
250,000,000	0:00:12	1:13:48
300,000,000	0:00:16	0:50:21
350,000,000	0:00:21	1:15:36

Table 3: running time of the machine learning training.

5.2 Challenges

Currently we see no major challenges.

5.3 Outline

In the next few months, we mostly want to finish two tasks: complete the integration in the Slovakian environment and test the performance of the pricing calculator. For the former task, we want to improve the automation of the technical set-up in the Slovakian environment. For the latter task, we have to run some performance tests to see whether we can answer pricing requests in real-time and how many we can answer within a second and a day. To fulfill the requirement of 300 million requests per day we need to reduce the time it takes to compute an optimal pricing solution as far as possible.

6 Use Case 5: Agricultural analysis based on Copernicus data

Use Case 5 has an ambiguous aim while including a German SME with their closed source application. This application, as displayed in D2.1 and D8.1, focuses on supporting the European agriculture industry with PROMET.

The connection between PROCESS and PROMET has been finalized and the following actions are available:

- Configuration of input variables within the PROCESS portal, the IEE
- Deployment of a PROMET run from the IEE on the LRZ infrastructure
- Stage-Out of the output data via LOBCDER for the end-user

These functionalities are realized with a generic and configurable API-container, which is deployed at the Use Case site is the communication point for the IEE. The API itself can trigger the computation of the PROMET HPC workflow, currently at the LRZ in Garching, control its execution via status updates and request LOBCDER to transfer the output data, which is then made available for the initial end-user in the IEE.

This solution has shown no overhead since all deployment requests are directly passed forward. The data transfer is called with the same semantic from the API-container than from the IEE.

In this stage, for several regions data sets are available:

- Europe*
- Central America*
- South America
- Australia
- West South Asia*

For PROMET data sets for the complete earth are available, so far only South America and Australia are included in the PROCESS workflow.

The potential computation focus on many different output variables, for example biomass, leaf area index or soil moisture. Those can be computed together with other parameters set within the IEE, e.g. crop, nutrition factor or irrigation.

Several of those computations were successfully deployed by the IEE on the LRZ site. Among those one run was computed on the region of South America. Here, the development of the soil moisture was simulated, including the parameters crop (maize) and nutrition factor (0.60). The following Figure 9 shows exemplary the development for 3 time steps May, August and October 2017:

D2.2: Use Case 5: Agricultural analysis based on Copernicus data

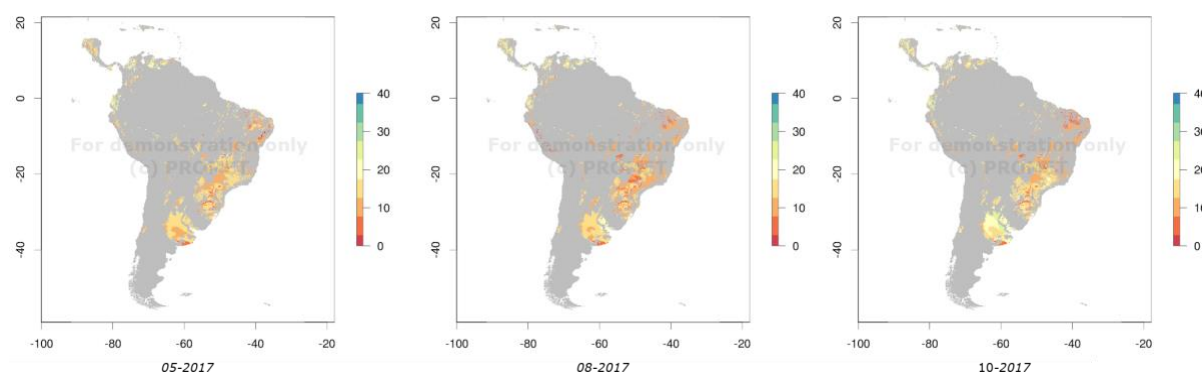


Figure 9: UC5 soil moisture simulation in South America.

The next phase will be used to increase the number of regions and parameters, the end-user can choose, and to include an efficient download option for the output.