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

D8.3	First packaged version of PROCESS software		
Project:	PROCESS H2020 – 777533	**Start / Duration:**	01 November 2017 36 Months
Dissemination[1]:	Public	**Nature[2]:**	**R**
Due Date:	31 July 2020	**Work Package:**	**WP 8**
Filename[3]	PROCESS_D8.3_FirstPackagedPROCESSSoftware_v1.0.docx		

ABSTRACT

This deliverable accompanies the first packaged version of the PROCESS software. It provides an overview of which services are released, and, for each of these released services, this deliverable will provide the relevant information on where the source code, binary release, documentation, etc. can be found.

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 Leader[4]	Maassen, J.	NLESC	Deliverable coordinator
Contributing Authors[5]	Cushing, R.	UvA	Writer
	Meizner, J.	AGH	Writer
	Nowakowski, P.	AGH	Writer
	Tran, V.	UISAV	Writer
Reviewer(s)[6]	Schmidt, J.	LMU	Reviewer
	Habala, O.	UISAV	Reviewer
Final review and approval	Höb, M.	LMU	Coordinator

Document History

Release	Date	Reasons for Change	Status[7]	Distribution
0.1	18.05.2020	Initial structure of document	Draft	Workgroup
0.2	10.07.2020	Abstract + introduction added	Draft	Workgroup
0.3	13.07.2020	Services data complete	Draft	Workgroup
0.4	17.07.2020	Version ready for review	Draft	Consortium
0.5	24.07.2020	Processed review	Draft	Consortium
0.6	28.07.2020	Processed second review	Draft	Consortium
1.0	28.07.2020	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

Executive Summary

This deliverable accompanies the first packaged version of the PROCESS software. As the PROCESS software stack is modular, this packaged version consists of individual releases of each of the services which can be found online. For each of the services we develop ourselves, we follow the FAIR software recommendations as published on https://fair-software.eu. This implies that next to providing a short description of each service and its role in the PROCESS and links to documentation, we also ensure the source code is available in a public repository (http://github.com or https://gitlab.com) with an open source license. Each release is archived and made citable through https://zenodo.org, and the software is made findable by publishing it via our online research software directory at https://software.process-project.eu. Next to the services developed within the PROCESS project, we also make use of several off-the-shelf open-source components developed externally. For these components, we refer to the relevant releases used within the PROCESS software stack. Finally, we also provide the technical readiness level (TRL) of each of the services.

List of Figures

List of Tables

1 Introduction

This document accompanies the first packaged version of the PROCESS software. This packaged release consists of individual releases of each of the services which play a central role in the final PROCESS architecture. This final architecture is shown in Figure 1 and described in detail in D4.5. The dashed red boxes in Figure 1 highlight the service components in PROCESS: IEE, Rimrock, Cloudify, LOBCDER (with its micro-infrastructure containers), and DataNet.

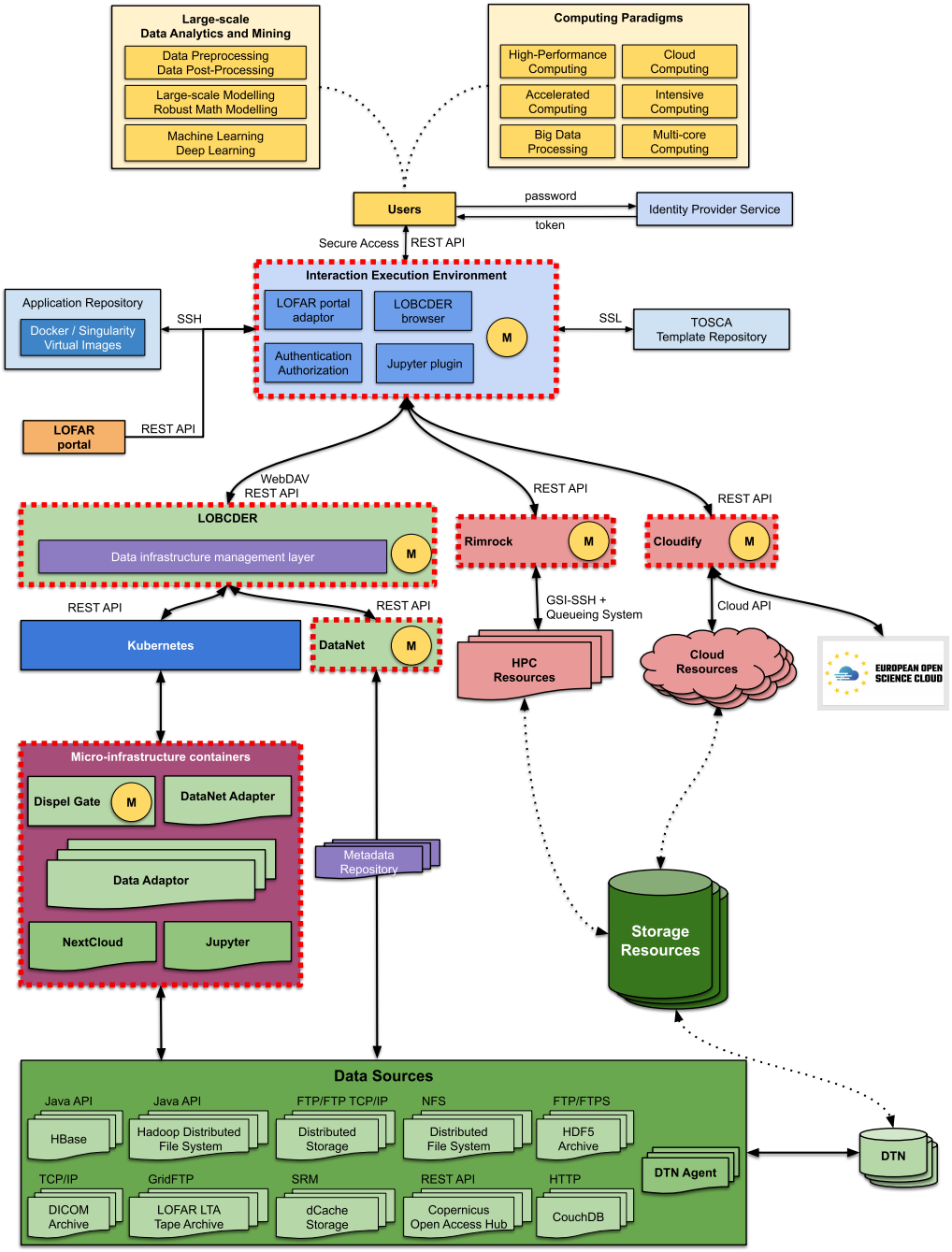

Figure 1: PROCESS architecture

The IEE, RIMROCK and LOBCDER services have been developed independently in projects predating PROCESS. In PROCESS, they have been further developed and modified to meet the PROCESS requirements and improve their TRL level. For each of these services, we have released a version specifically for the PROCESS production prototype. These releases are described in more detail below.

DataNet is not yet used by any of the services of use cases in PROCESS. Therefore, while some modifications and development are planned, its release is not part of this production prototype. Therefore, we will more details on DataNet below, but no version information yet. Future releases of the PROCESS software stack will may a version do DataNet, once it becomes available.

Cloudify is an off-the-shelf component which is developed externally and is used extensively in commercial cloud datacenters. It is developed and supported by Cloudify Ltd, and available in both a commercial and open source community version. In the process software stack, we use the unmodified community version, whose details are described below.

For all of the software which is developed or modified in the PROCESS project, we follow the FAIR software recommendations for releases, as published on https://fair-software.eu. This implies that next to providing a short description of each service and its role in the PROCESS and links to documentation, we also ensure the source code is available in a public repository (http://github.com or https://gitlab.com) and uses an open source license. Each release is also archived and made citable through Zenodo (https://zenodo.org). The software is made findable by publishing it in our project specific research software directory which can be found at: https://software.process-project.eu. We also estimate the technical readiness level (TRL) of each of the released services.

For the externally developed off-the-shelf components (currently Cloudify), we refer to the relevant releases used within the production prototype of the PROCESS software stack.

In this release, we do not yet release the necessary orchestration and configuration files needed to run the integrated PROCESS software stack on the testbed. These files will be included in the final release of the PROCESS software, along with documentation on how to deploy this software stack on other infrastructures.

2 The Interactive Execution Environment (IEE)

The Interactive Execution Environment (IEE) is a graphical web interface (shown in Figure 2), which provides a one-stop entry point to the PROCESS infrastructure and execution of computational tasks on the underlying HPC and cloud resources. The IEE manages the required delegation of security credentials and abstracts away all details related to interaction with various types of computational resources.

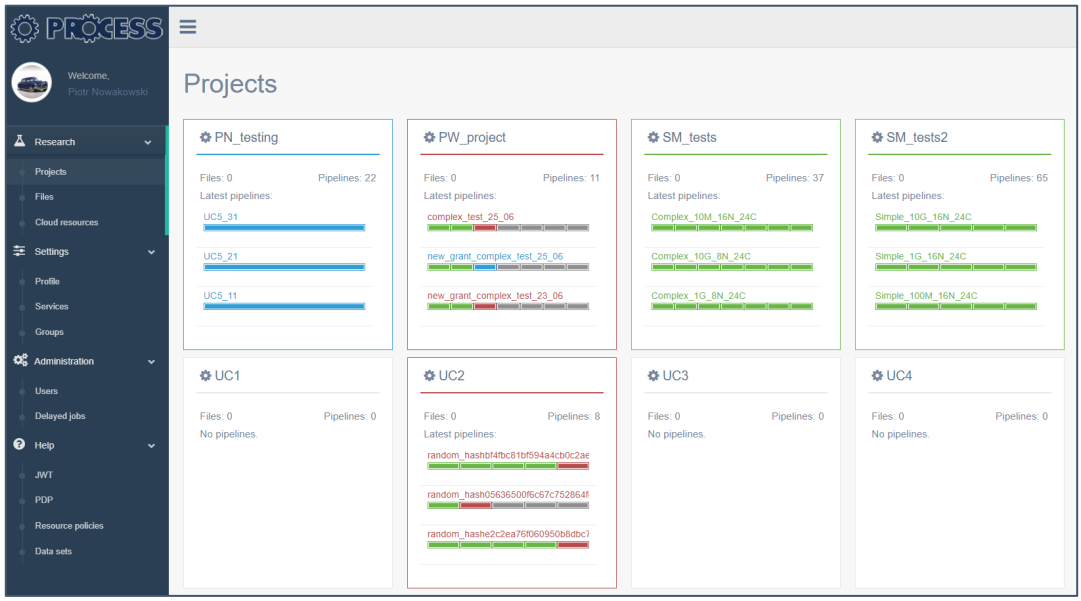

Figure 2: IEE user interface

The IEE provides access to configurable computational pipelines which represent individual use cases. Each pipeline may be launched an arbitrary number of times, with varying parameters. The IEE takes care of acquiring the required resources, scheduling data transfers (via LOBCDER) and executing the computational steps which make up the use cases. IEE can schedule computations on traditional HPC compute clusters (via RimRock) and cloud resources (via the Cloudify extension). It can also communicate with bespoke REST interfaces offered by providers of external computational services - such as with the AgroCopernicus use case (UC#5). In addition, it also allows access through a REST API for use cases that require a use-case-specific user interface, such as the LOFAR use case (UC#2).

The IEE was originally developed in the EuroValve project and is still being used there. For PROCESS, it was adapted and extended to meet the requirements of our use cases. We therefore estimate its TRL as 8.

Table 1: IEE - PROCESS production prototype release

Process website	https://www.process-project.eu/prototypes/
Software catalogue	https://software.process-project.eu/software/iee
Source code repository	https://gitlab.com/cyfronet/iee/-/tree/1.0.0
Install documentation	https://gitlab.com/cyfronet/iee/-/blob/1.0.0/README.md
User documentation	Available within GUI environment
DOI	10.5281/zenodo.3950336
Version	1.0.0
License	MIT
TRL	8

3 LOBCDER

Moving data in the exascale era is very time consuming because moving data is bound by the underlying limitations of the network infrastructure. One exabyte of data takes around 3 years to transfer on the faster, state of the art interconnects. At the same time projects like the Square Kilometre Array are expected to generate an exabyte of data every day or so. This puts a lot of strain on how to handle and process the data which is globally distributed and difficult to move.

Our approach, in PROCESS, is that the data services cannot be viewed as just software components. Instead a more holistic approach is needed which takes into consideration the underlying infrastructure, the flow of data, the processing and the services. This leads to a smart infrastructure where we can, for example, move the computation to the data instead of moving the data or split the processing of data into several parts, each being executed on different parts of the network.

LOBCDER introduces the concept of a micro-infrastructure which uses container technologies to combine data services with infrastructure to create a smart data infrastructure. This means that development of services now consists of two parts; the description of the underlying container infrastructure (micro-infrastructure) and the development of the different software services that fit into the micro-infrastructure.

In PROCESS, LOBCDER is used by the IEE to deploy use case specific data services, which allow use cases to adapt the data access and transfers to their needs. LOBCDER has significantly extended to fit the requirements of the use cases and the testbed. Therefore, we estimate it to be at TRL 7.

Table 2: LOBCDER – PROCESS production prototype release

Process website	https://www.process-project.eu/prototypes/
Software catalogue	https://software.process-project.eu/software/mini-lobcder
Source code repo	https://github.com/micro-infrastructure
Install documentation	https://github.com/micro-infrastructure/documentation#installation-documentation
User documentation	https://github.com/micro-infrastructure/documentation
DOI	10.5281/zenodo.3900943
Version	0.1.0
License	MIT
TRL	7

4 RIMROCK

Rimrock provides a REST API to existing scientific compute infrastructure, such as HPC clusters. By using REST, access to compute services, applications and advanced scripts deployed on the infrastructure becomes independent of the local software stack. The main advantage of the Rimrock is the ability to connect to any underlying software technology and provide an integrated solution for secure access to computing infrastructures.

By using REST, the Rimrock service allows to use its functionalities independently of any programming language chosen to build applications on top of the computing infrastructure. It is therefore possible to create web and desktop applications as well as prepare advanced computation scripts. An interesting approach, also supported by the service, is the ability to

develop web applications, which can be run solely in the user's web browser, minimizing the role of server-side software.

All data exchanged with the Rimrock service is fully secured with an encrypted connection and for user authorization a temporary user certificate is used, ensuring that long-lived credentials would not leak, giving permanent access to the attacker.

In PROCESS, Rimrock is used by the IEE to access the HPC compute infrastructure of the PROCESS testbed. To do so, it was extended with support for container-based applications and workflows, which is used by UC#1 and UC#2.

Before PROCESS, Rimrock had already been successfully used during development of a web application in the domain of energy sector, allowing for harnessing of the computing power of the PLGrid infrastructure for the analysis of different scenarios when building a national power grid and the influence on the environment and human health. We therefore estimate its current TRL level to be 8.

Table 3: RimRock - PROCESS production prototype release

Process website	https://www.process-project.eu/prototypes/
Software catalogue	https://software.process-project.eu/software/rimrock
Source code repo	https://gitlab.com/cyfronet/rimrock/-/tree/2.0.3
Install documentation	https://gitlab.com/cyfronet/rimrock/-/blob/2.0.3/doc/install/installation.md
User documentation	https://rimrock.plgrid.pl/jobs
DOI	10.5281/zenodo.3949651
Version	2.0.3
License	MIT
TRL	8

5 Cloudify

Service orchestration is a process for automated configuration, deployment and management of services and applications in the cloud. It can automate the execution of different service workflows including deployment, initialization, start/stop, scaling, healing of services based on standardized descriptions of composed services, relations between components and their requirements. In the PROCESS project, we use the OASIS TOSCA standard for service description and use Cloudify for orchestration.

Cloudify is an externally developed open-source cloud orchestration platform, designed to automate the deployment, configuration and remediation of application and network services across hybrid cloud and stack environments. It uses OASIS TOSCA templates written in YAML (called blueprints in Cloudify) for defining applications, services and dependencies among them. These blueprint files describe the execution plans for the lifecycle of the application for installing, starting, terminating, orchestrating and monitoring the application stack. Cloudify uses the blueprint as input that describes the deployment plan and is responsible for executing it on the cloud environment.

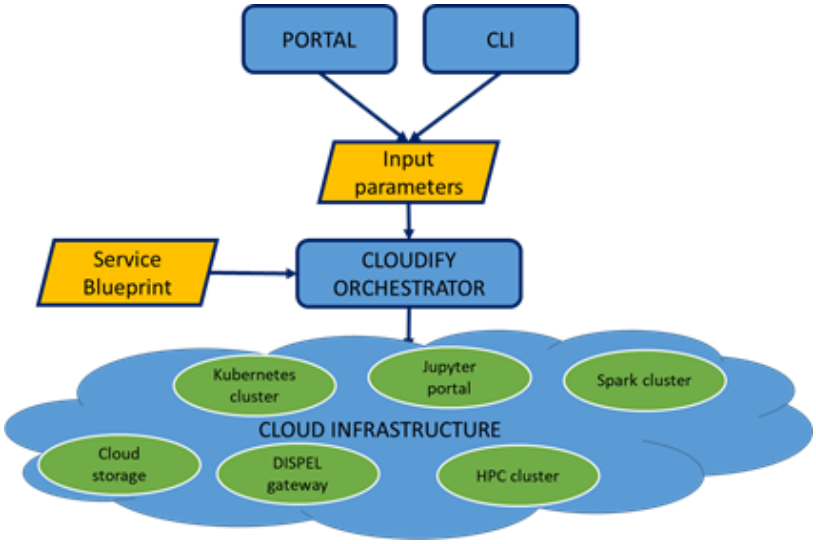

Figure 3: Cloudify orchestration

In PROCESS, Cloudify is used by the IEE to access the cloud infrastructure available as part of our testbed. The role of Cloudify is similar to that of Rimrock. Where Rimrock provides unified access to the HPC compute infrastructure, Cloudify provide this access to the cloud infrastructure. This cloud infrastructure is used in UC#4.

As Cloudify is developed commercially and used extensively in commercial cloud data centres, we estimate its TRL level to be 9.

Table 4: Cloudify – PROCESS production prototype release

Process website	https://www.process-project.eu/prototypes/
Software catalogue	https://software.process-project.eu/software/cloudify
Source code repo	https://github.com/cloudify-cosmo
Install documentation	https://docs.cloudify.co/4.5.0/install_maintain/
User documentation	https://docs.cloudify.co/4.5.0/working_with/
DOI	N/A (externally developed)
Version	18.10.4-community (also known as 4.5.0)
License	Apache-2
TRL	9

6 DataNet

DataNet enables lightweight metadata management backed by a flexible database which allows convenient access to the stored objects. One of the main goals of DataNet is to make it usable from the largest set of languages and platforms possible. That is why it uses HTTP as a basis for transferring data between backing servers and the service, and − for improved compatibility and automation− we applied the REST methodology to structure the messages sent to and from the repositories, which makes the integration process straightforward.

DataNet also features in-transit encryption to ensure the security of the metadata while being moved between components. The pluggable security mechanism also ensures that access to the platform is restricted to the appropriate group of people to prevent leakage of stored data. DataNet's API enables both the integration with other PROCESS components such as the IEE portal as well as direct access from external components.

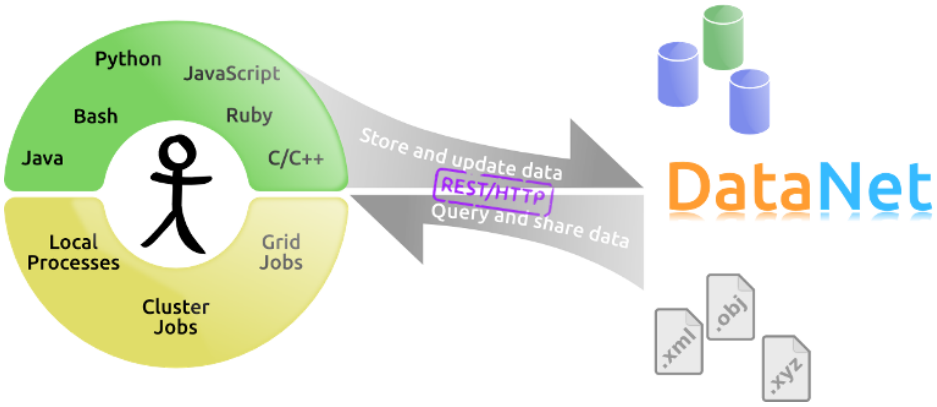

Figure 4: DataNet lightweight metadata management

Originally the DataNet was developed and tested in the scope of the PLGrid project. At that point it utilized the state of art Platform-as-a-Service component which ensures scaling and database service provisioning for structured data. In the scope of the PROCESS project the solution has been reengineered based to reflect on the rapid progress in development of platform-as-a-service infrastructure. This new implementation is based on Restheart, a REST API on MongoDB in combination with an orchestration of Docker containers. This ensures the ability of DataNet to run on a wide range of infrastructures. The platform also has been extended to support non-structured metadata.

While part of the PROCESS architecture, DataNet is not currently use by the use cases or services yet. Its metadata storage and lookup capabilities are currently not exploited by other PROCESS components. DataNet's development is ongoing but the integration with the PROCESS software stack is not complete. It is therefore not yet released as part of the production prototype.

7 Conclusion

In this deliverable we have provided an overview of the first packaged version of the PROCESS software. For each of the services developed or modified in the PROCESS project (IEE, RIMROCK and LOBCDER), we have created a release which adheres to the FAIR software guidelines, and provided the relevant information on this release in this document. For Cloudify, and externally developed service, we provide the relevant information on the version used. DataNet is not yet included in this release, as its development and integration into the PROCESS software stack is not finished.

We do not yet release the necessary orchestration and configuration files needed to run the integrated PROCESS software stack on our testbed. These files will be included in the final release of the PROCESS software, along with documentation on how to deploy this software stack on other infrastructures