



PROviding Computing solutions for ExaScale ChallengeS

D6.1	First prototype		
Project :	PROCESS H2020 – 777533	Start / Duration:	01 November 2017 36 Months
Dissemination¹:	PU	Nature²:	R+O
Due Date:	Month 12	Work Package:	WP 6
Filename³	PROCESS_D6.1_First_prototype_v1.0.docx		

ABSTRACT

The first PROCESS prototype comprises a first prototype of the proposed services with demonstration, initial prototype of the data orchestration system and tools of PROCESS, the first user interface portal of PROCESS as well as an initial benchmark set consisting of workflows, applications and data sets.

This version is a draft of D6.1 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.

Deliverable Contributors:	Name	Organization	Role / Title
Deliverable Leader⁴	M. Bubak	CYF	Coordinator
Contributing Authors⁵	Bobák, M., Dlugolinský, Š., Habala, O., Nguyen, G., Šeleng, M., Tran, V.	UISAV	Writers
	Pancake-Steeg, J., Spahr, S.	LSY	Writers
	Gentschen Felde, N., Heikkurinen, M., Höb, M., Schmidt, J.	LMU	Writers
	Graziani, M., Müller, H.	HES-SO	Writers
	Maassen, J., Spreeuw, H.	NLESC	Writers
	Belloum, A., Cushing, R., Rahmanian, A.	UvA	Writers
	Bubak, M., Meizner, J., Nowakowski, P., Rycerz, K., Wilk, B.	AGH / AGH-UST	Writers
Reviewer(s)⁶	Cushing, Reginald	UvA	Reviewer
	Graziani, Mara	HES-SO	Reviewer
	Habala, Ondrej	UISAV	Reviewer
	Höb, Maximilian	LMU	Reviewer
	Maassen, Jason	NLESC	Reviewer
	Meizner, Jan	AGH / AGH-UST	Reviewer
	Spahr, Stefan	LSY	Reviewer
Final review and approval			

Document History

Release	Date	Reasons for Change	Status⁷	Distribution
0.1	2018-09-17	Initial version	Draft	
0.4	2018-10-08	Revised version following changes to ToC agreed upon at the Berlin consortium meeting	Draft	
0.5	2018-10-21	Added references and information on future work	In Review	
1.0	2018-10-31	Final version following internal 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
1 Overview	6
2 Current status of the PROCESS platform	6
2.1 Implementation approach.....	6
2.2 PROCESS development environment.....	9
2.3 PROCESS from the user perspective.....	9
2.4 The PROCESS IEE workbench – structure and operations	10
2.5 Demonstration scenario	12
3 Future plans	15
4 Conclusion	15
References	16
Appendix A: IEEE eScience Workshop	17

Executive Summary

This deliverable constitutes a textual description of the initial prototype of PROCESS tools and services elaborated within WP6 of the project. It is meant to accompany preliminary versions of PROCESS software which will have been deployed on the infrastructural resources available to the project. The documentation presented herein will also serve as an introduction to the capabilities of the PROCESS computing environment, and will be extended in later WP6 deliverables.

The document is structured as follows:

- Section 1 provides a general overview of the goals and work in Work Package 6, and serves as an introduction to subsequent sections.
- Section 2 is devoted to the early implementation of platform components, as embodied by the prototype infrastructure. It also describes usage examples and is meant as the initial version of their respective user manuals.
- Section 3 discussed future plans related to development of the Interactive Execution Environment and integration with other components of the PROCESS infrastructure.
- Section 4 contains closing remarks.

This document should be read in conjunction with PROCESS deliverable D4.2 (Report on architecture evaluation and Dissemination), which presents the Project architecture and the rationale for the design choices, as well as describing the design and capabilities of each component of that architecture.

List of Figures

Figure 1: Schematic view of the PROCESS Interactive Execution Environment	7
Figure 2: Description of components available in the initial prototype of the IEE	8
Figure 3: Accessing the PROCESS Interactive Execution Environment - login screen	10
Figure 4: IEE user workbench.....	11
Figure 5: Setting up a new project	12
Figure 6: Configuring a new pipeline.	13
Figure 7: Starting another pipeline.....	14
Figure 8: Pipeline ready to run with input file provided.	14

1 Overview

The aim of the PROCESS project is to create prototypes that showcase the potential benefits of exascale data services. This goal will be achieved by creating tools and services that will support heterogeneous exascale use cases driven by both the scientific research community and the industry. Building on the architecture designed by Work Package 4, Work Package 6 implements, refines and validates the service-oriented computing infrastructure solution that is capable of supporting the development of the applications and novel usage scenarios supported and identified by the Work Packages 2 and 3.

The initial prototype of the PROCESS infrastructure implements key features and aspects of the platform's functionality, and is intended for validation based on the basic use case requirements [D4.1]. As such, it serves a major milestone in the Project's development lifecycle. While it is understood that the implementation will not be fully production-ready, it is nevertheless comprises core components and will be deployed in the subset of computing centres contributing resources to the project. The following key components of the PROCESS solution form this first prototype deployment:

- Initial version of the execution environment capable of deploying and executing PROCESS services on extreme large computational resources,
- A set of user interfaces for the former, enabling easy interaction by users who do not possess in-depth knowledge of HPC and distributed computing,
- A set of data services, which enable data to be supplied to the distributed computing resources in a timely fashion, thus facilitating computations
- A security infrastructure which restricts access to PROCESS services and tools to authorized individuals.

The following sections discuss the assumptions and challenges involved in designing and implementing each of the components mentioned above. They also show how the architecture, built from services that have already reached level TRL 6, will enable these services to be brought up to TRL 8.

The described prototype was demonstrated at the "Platform-driven e-infrastructure innovations" workshop at the IEEE eScience Conference 2018 in Amsterdam, The Netherlands. 35 scientists were attending the project's half-day workshop. A complete summary can be found in the Appendix A: IEEE eScience Workshop.

2 Current status of the PROCESS platform

2.1 Implementation approach

While the overall PROCESS architecture is described in D4.2 [D4.2], in order to support discussion about implementation choices we will reiterate some of the assumptions upon which the PROCESS platform is based, given that it is the role of Work Package 6 to implement, refine and validate the architecture defined in Work Package 4, providing feedback where appropriate.

As the prototype deployment is centred upon the Interactive Execution Environment, we present a schematic description providing a base overview of the available components and tools in Figure 1.

D6.1 Current status of the PROCESS platform

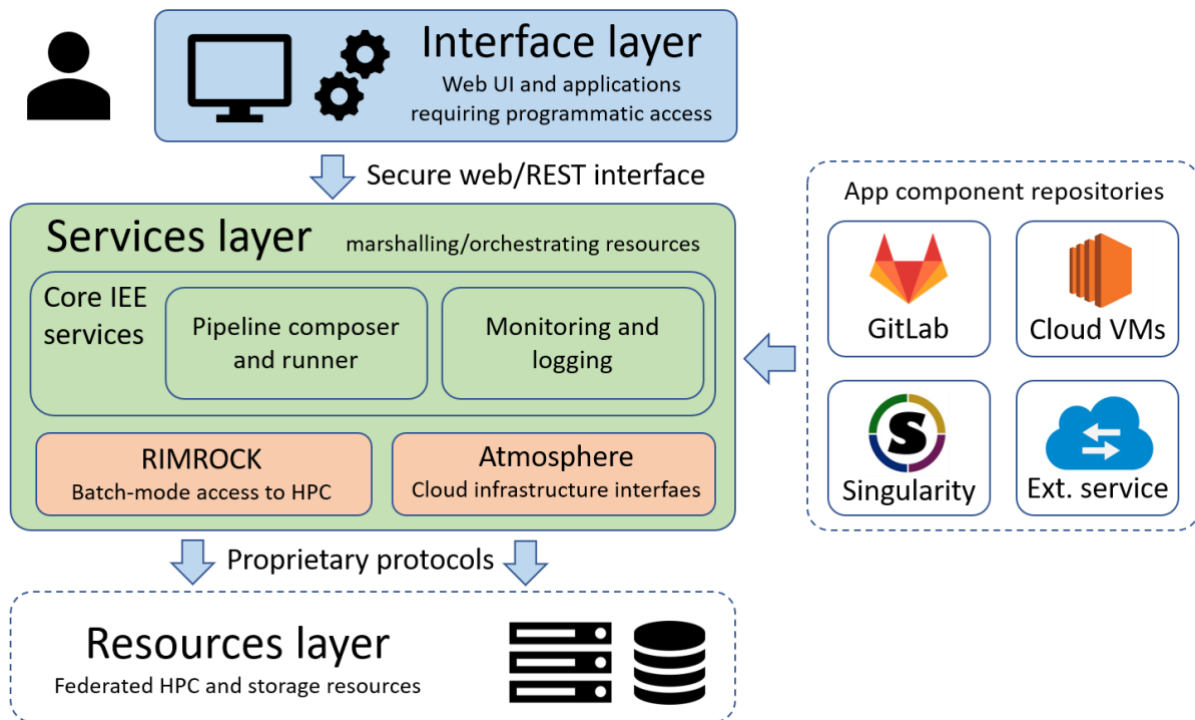


Figure 1: Schematic view of the PROCESS Interactive Execution Environment

The core of the Interactive Execution Environment, which integrates with other PROCESS services and tools, as described in the relevant deliverables, is based on a number of existing systems, of which the most notable ones are the Collage infrastructure [Collage] and the EurValve Model Execution environment, capable of executing HPC computations on distributed resources for the EurValve VPH collaboration [EurValve]. In line with the description which can be found in D4.2, the environment provides a bridge between the end users (who may not necessarily be skilled HPC programmers) and the distributed resources provided by participating HPC and data storage operators. The core assumption shaping the environment is that an exascale computational environment is unlikely to materialize as a monolithic installation (e.g. a single computational cluster), and will instead call for federation of diverse (and possibly geographically distributed) resources. The IEE (Interactive Execution Environment) addresses this challenge by providing the capability to manage and orchestrate such distributed resources to effectively act as a single computational infrastructure upon which PROCESS applications can be deployed.

Referring back to the above presentation, Figure 2 presents components of the IEE framework which have been implemented and deployed as part of the initial prototype.

D6.1 Current status of the PROCESS platform

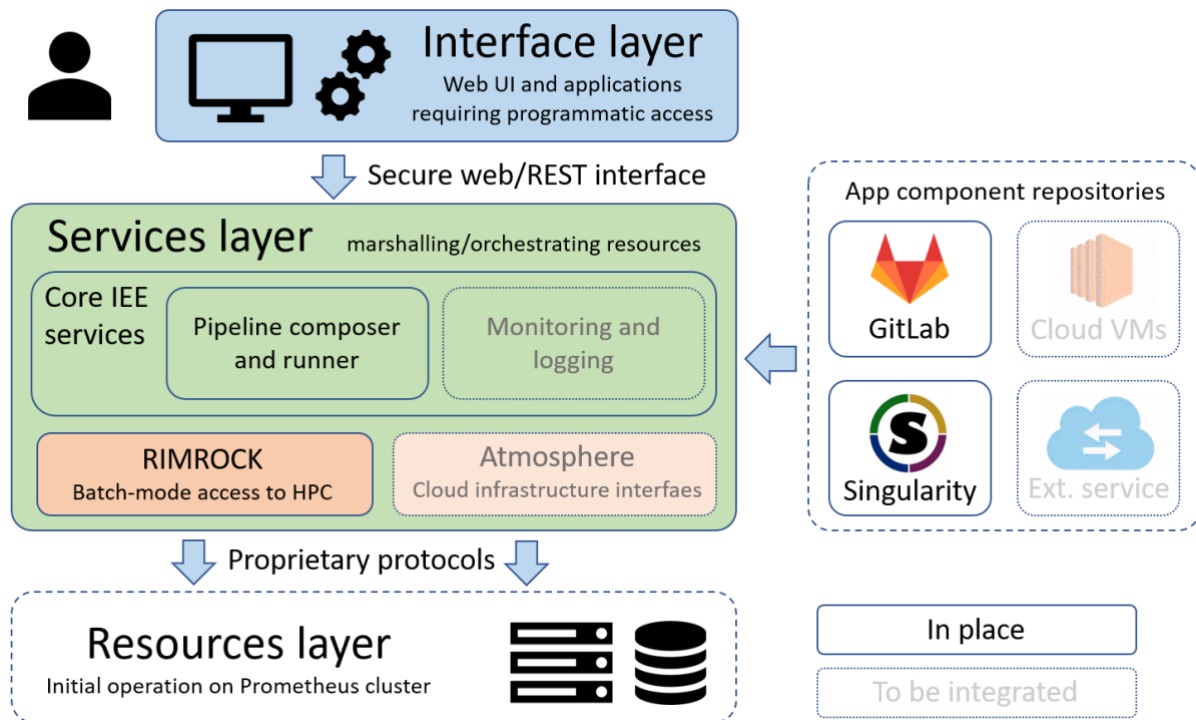


Figure 2: Description of components available in the initial prototype of the IEE.

For the initial demonstrator, we have decided to implement and deploy the basic components which would permit us to proceed with the execution of a simple use-case scenario in the single-site operation mode, using the resources available at a single computational center. This setup should be treated as a simplified version of the PROCESS operation flow, which, eventually, will encompass multiple sites and be capable of deploying container-based computations with common usage policies and resource discovery capabilities. For operational reasons, the decision was made to make use of the Prometheus supercomputing cluster at ACC Cyfronet AGH, and to proceed with deployment of a GPU-enabled Singularity container hosting components of the HESSO use case. For more information regarding the specific nature of this use case, please refer to section 2.5. In order to permit this use case to proceed, the following operations were performed:

- An initial stub of the Interactive Execution Environment was created as a Ruby on Rails application stack, repurposing components of the EurValve Model Execution Environment framework.
- The IEE prototype was extended to accommodate deployment of Singularity containers and access to GPU-enabled computing resources.
- The IEE prototype was deployed as a PL-Grid Service, and registered with the appropriate PL-Grid authorities.
- A set of computing grants were negotiated with PL-Grid, facilitating use of HPC resources at Cyfronet (both GPU-enabled and otherwise).
- Storage space was reserved on the Lustre filesystem at Cyfronet to accommodate PROCESS use case data.
- A PROCESS group was set up within the PL-Grid infrastructure, to enable Consortium partners to log into the framework and execute computations.
- Tests were carried out involving various means of deploying Singularity containers on the Prometheus cluster.

2.2 PROCESS development environment

In accordance with good software development practices, the PROCESS environment is expected to be provided as two distinct deployments:

- A production environment, based upon tools and platform components which have been deemed sufficiently stable to enable integration of application use cases. This environment will be available at <https://process.cyfronet.pl>.
- A development environment, which will host the latests (unit-tested) versions of platform components and provide a means of testing their functionality in a real-world setting. This environment will be available at <https://process-dev.cyfronet.pl>.

As part of our work on the initial prototype, we have secured the necessary computational infrastructure, including hosting resources, certificates etc. to provide support for both environments. However, only the development environment is currently in operation as the initial prototype is not yet sufficiently robust to constitute a “true” production environment, and is liable to frequent changes necessitated by ongoing integration with external tools and services; particularly the data storage services.

Given that the entity responsible for deploying and managing the PROCESS Interactive Execution Environment is ACC Cyfronet AGH, a Polish entity, the environment itself is implemented and set up as a service of the PL-Grid infrastructure [PL-Grid]. PL-Grid is a Polish national initiative which unites major scientific research and HPC centers and enables them to share resources, including computational resources. Deploying PROCESS tools in such a way brings several major advantages, the most important of them being that Polish HPC resources – including the Prometheus cluster – can be easily accessed by all members of the PROCESS consortium. Note that PL-Grid resources are provisioned free of charge to Polish scientists as well as their foreign collaborators (which includes PROCESS project partners).

2.3 PROCESS from the user perspective

In order to acquire access to the PROCESS IEE, each prospective member must first register with the PL-Grid project, either as a Polish scientist or a foreign collaborator thereof. This is done by creating an account at <https://portal.plgrid.pl> and requesting to join the PROCESS user team. Details of the registration process are out of scope of this deliverable; however, ACC Cyfronet AGH, in its capacity as a Project partner, provides ongoing assistance and help in case of registration problems, particularly for members of use case teams.

Once an account has been set up, it can be used to access the PROCESS development IEE at <https://process-dev.cyfronet.pl>, as illustrated in Figure 3.

D6.1 Current status of the PROCESS platform

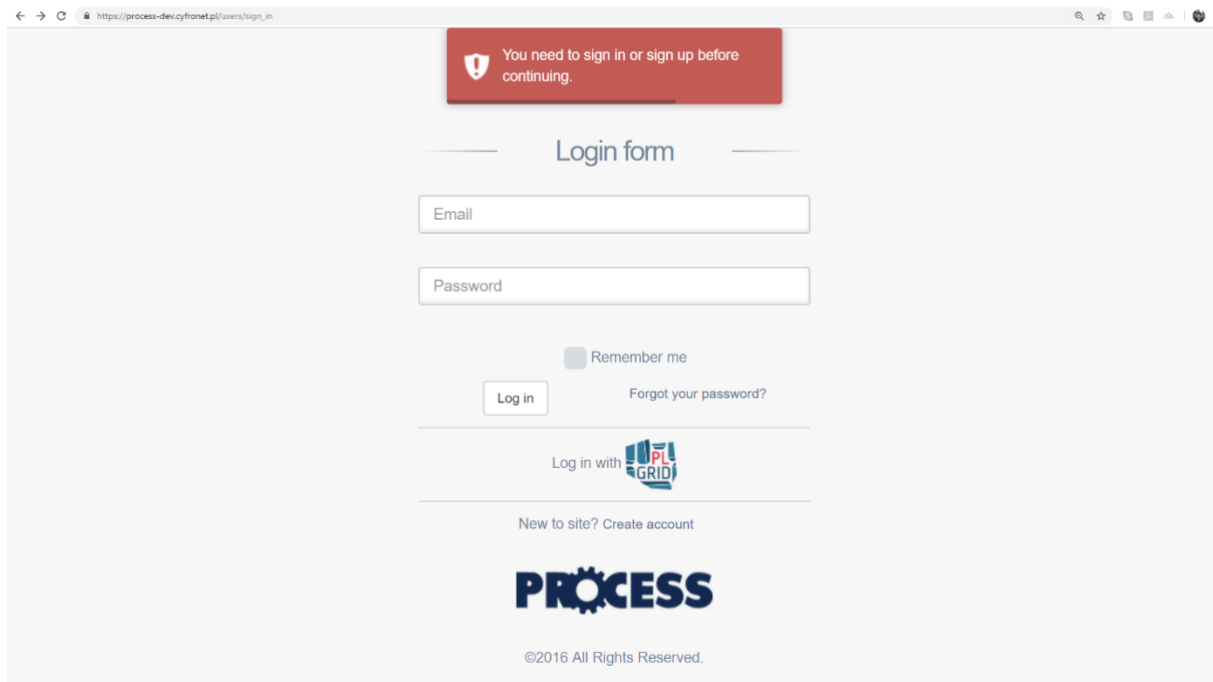


Figure 3: Accessing the PROCESS Interactive Execution Environment - login screen

Rather than directly typing account credentials in the provided form (which is a development-only feature meant to facilitate access for platform maintainers), the user should instead select the “Log in with PL-Grid” option, which will automatically redirect them to the PL-Grid login site.

PL-Grid provides single sign-on functionality for all services registered with the infrastructure (which now includes PROCESS, since the service has been officially included on the list of PL-Grid services). It is important for end users to access the platform with their PL-Grid credentials, since this enables the IEE to automatically generate a user proxy certificate, which will subsequently be used to authorize any requests to HPC resources integrated by PROCESS. Without this step, it would not be possible for computations launched on behalf of individual users to be run on the resources managed by PL-Grid.

Once the login process is successfully completed, the user is redirected to their workbench where they can launch computations on the PROCESS infrastructure, and monitor their status.

2.4 The PROCESS IEE workbench – structure and operations

The basic view of the IEE user workbench, which is presented to the user upon login, is shown in Figure 4.

D6.1 Current status of the PROCESS platform

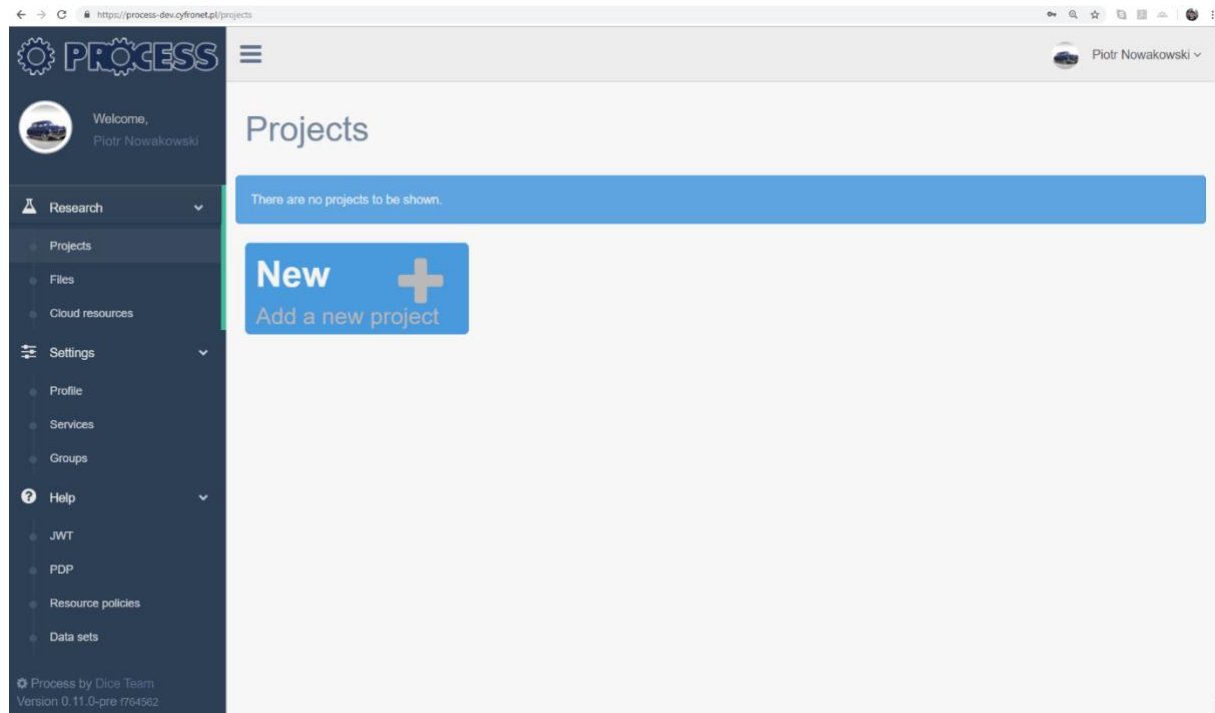


Figure 4: IEE user workbench

Conceptually, the PROCESS execution environment treats each use case as a separate **Project**. The user may create an arbitrary number of projects, and each computational workflow is executed in the context of a separate project.

A project is defined as a distinct repository of data which the user can operate on. In the context of PROCESS use cases, this may mean a specific LOFAR observation, a specific medical image (or image set) or any other subset of the exascale data pool available to the application. Upon this dataset arbitrary computations may be performed on the available HPC resources. These operations are defined in terms of **pipelines** – essentially workflows – which include:

- Arbitrary computations submitted to HPC via the Rimrock [REF] tool,
- Singularity containers downloaded from an external repository (e.g. SHub) and deployed to the available HPC resources
- External services invoked by calling their REST APIs (this concerns services which are not developed within PROCESS but supplied by outside providers)

A **pipeline** consists of a number of **steps**, each of which defines a specific computation by specifying:

- the required list of input files which must be present in order for the step to be *runnable*
- the computational payload of the step (e.g. HPC script, container etc.)

In addition, each pipeline may be instantiated as *automatic* or *manual*. In an automatic pipeline all steps are launched automatically when the system detects the required input conditions are met (input data is present and computational payload is available). In a manual pipeline, each step must be explicitly triggered by the user by selecting the appropriate option in the GUI. This enables the user to review intermediate results of pipelines and also prevents launching of unnecessary computations when these intermediate results are unsatisfactory.

A pipeline may produce output in the form of files, which can also be retrieved directly from the IEE thanks to its integration with the PROCESS data storage subsystem.

D6.1 Current status of the PROCESS platform

The following section presents an example of pipeline execution using the current prototype of IEE.

2.5 Demonstration scenario

As a demonstration scenario, we provide a simple pipeline which is based on the HESSO use case. While not yet supporting all the features associated with this use case, it nevertheless includes all the core building blocks of the IEE architecture and makes use of the available HPC resources. The pipeline is defined as a script which uses the TensorFlow library to perform computations on input data (supplied as files in the PROCESS file store) and output results (also as files).

In order to be able to launch the scenario, we begin by creating a new **project** which will contextualize our computations. This is done by clicking the **New** button seen in Figure 4.

This will bring up the project window, which you can see in Figure 5.

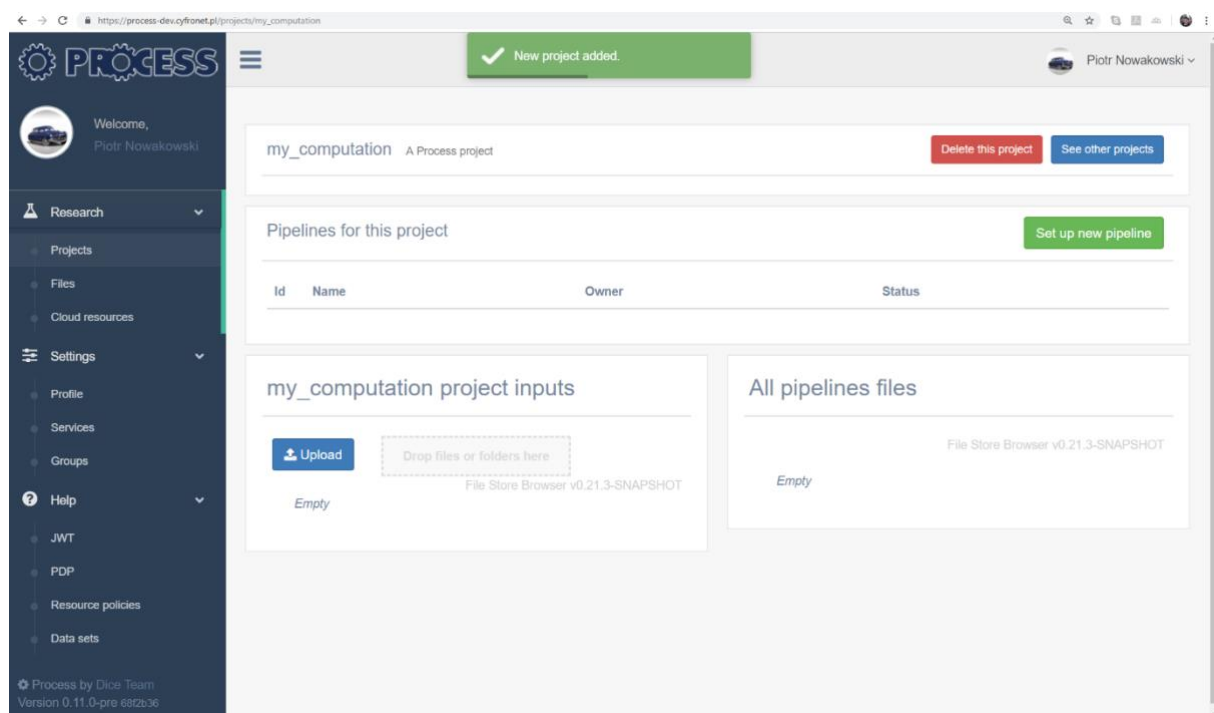


Figure 5: Setting up a new project

Within the context of the project, you can launch any number of **pipelines** which represent individual computations carried out on input data. The IEE also shows the data items assigned to your project, as well as any outputs produced by the project. All these are stored in the PROCESS file storage, which provides a WebDAV extension for the benefit of IEE users (naturally, extremely large files are not expected to be uploaded or downloaded by this channel – rather, this is a measure of convenience, enabling retrieval – where feasible – of project results).

By clicking **Set up new pipeline** you can start a new pipeline – this is shown in Figure 6.

D6.1 Current status of the PROCESS platform

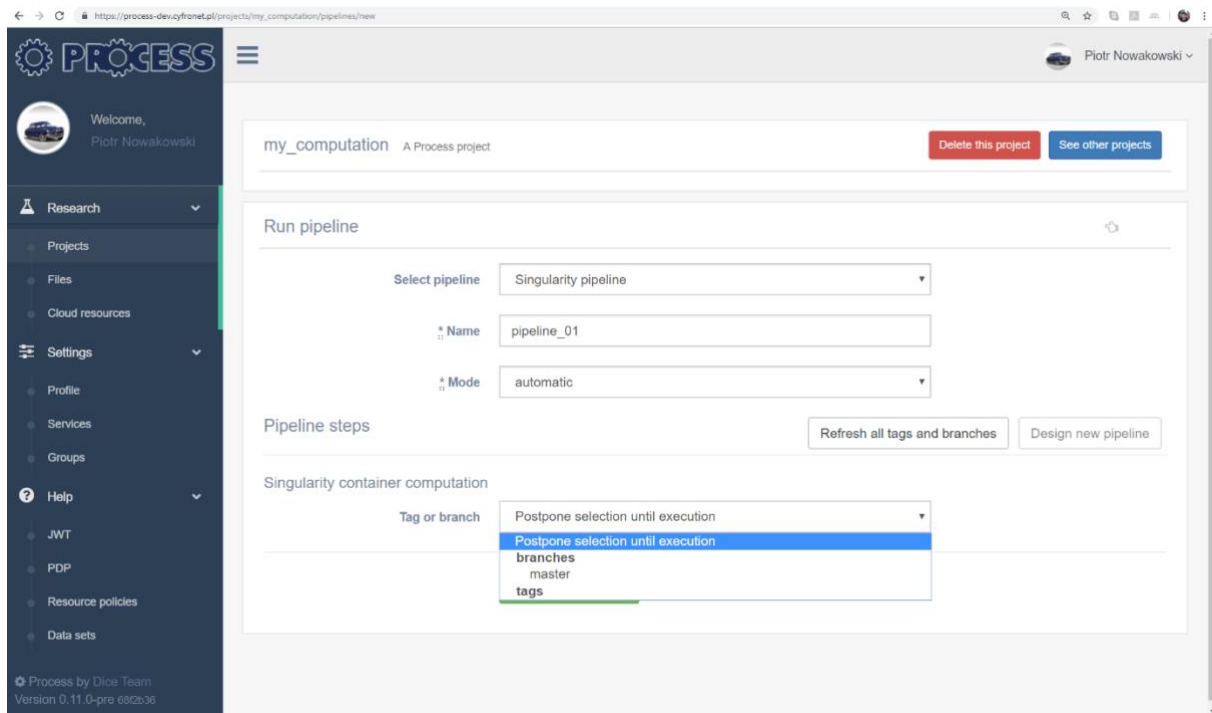


Figure 6: Configuring a new pipeline.

IEE enables the user to select the pipeline in question (the prototype provides a number of sample pipelines which are executed using HPC resources at ACC Cyfronet AGH), and determine whether it is to be launched in automatic or manual mode.

In addition, note the **Tag or branch** selector displayed for each step of the given pipeline (in our case, the pipeline consists of only one step). Since the computational payload of each step is stored in a dedicated Git repository (we use Gitlab for the prototype, but any Git-compliant repository will do), we acknowledge that the project tree (which includes the code to be executed) may be branched by application developers (i.e. the domain scientists responsible for developing PROCESS use cases). Consequently, we provide the ability to select the version of code which will be executed within the given pipeline. This approach brings a number of benefits, the most important of them being *reproducibility* – a given branch or tag may be persisted for later use even when a new version of the step code is released. In addition, the presented feature also helps test new versions of pipeline steps before they are released to end users.

Note that multiple pipelines can be started within each project – for example, we may launch a second pipeline while the first one is being configured, as shown in Figure 7.

D6.1 Current status of the PROCESS platform

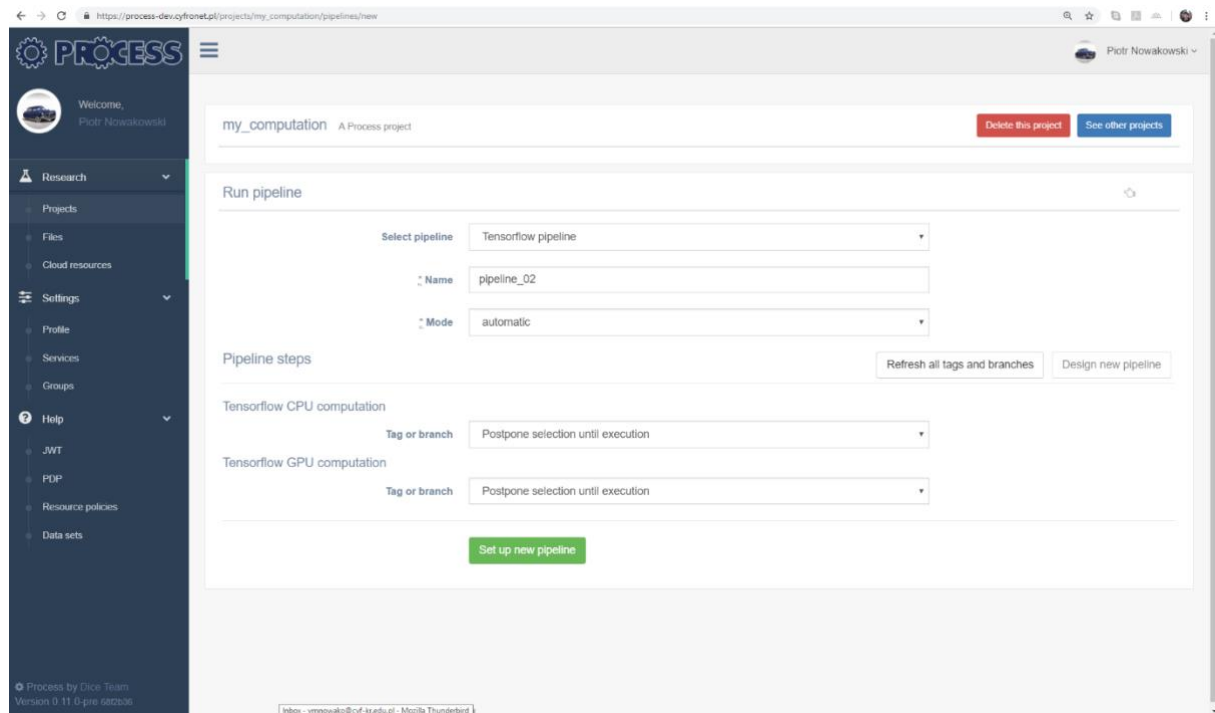


Figure 7: Starting another pipeline

Each step can be run if the required input files are present. While in future versions of the IEE platform we will expect the required input to be staged in by the PROCESS data management components (or, alternatively, by a dedicated pipeline step preceding the actual computation), in this case, we can simply upload an input file from the local PC by using the **pipeline inputs** widget – see Figure 8.

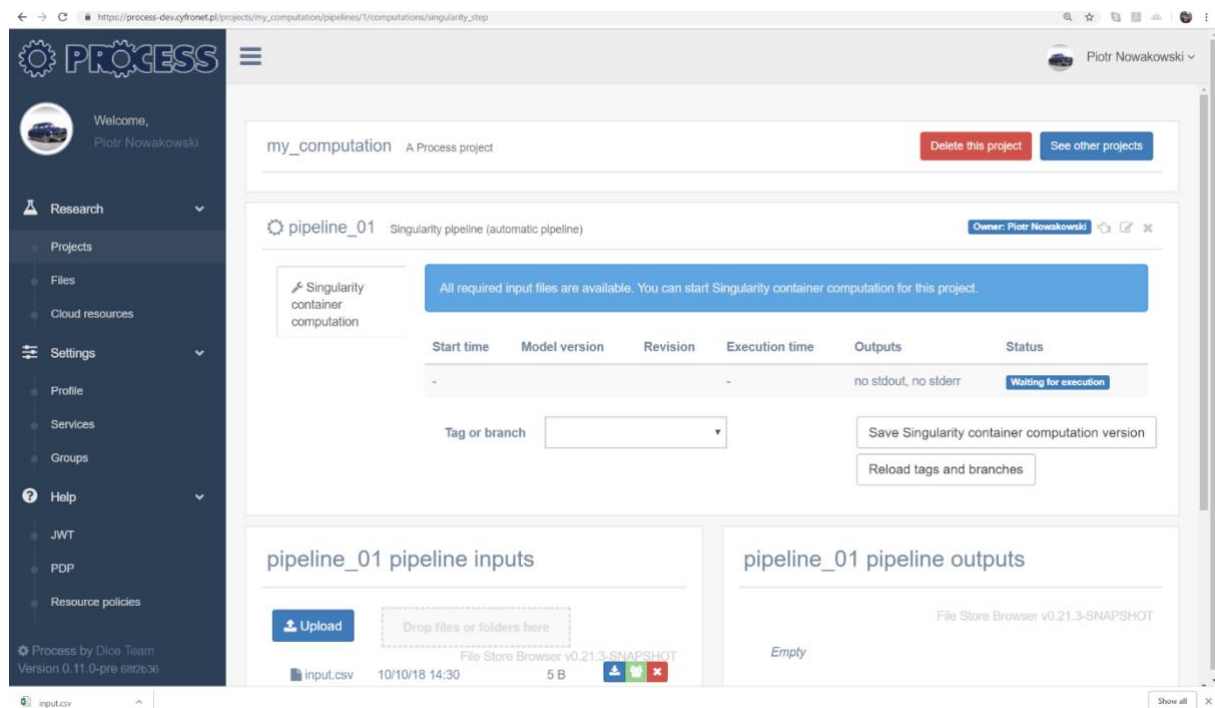


Figure 8: Pipeline ready to run with input file provided.

D6.1 Future plans

Assuming the given pipeline step is ready to run (the presence of an input file is checked, and also – if the pipeline is manual – the user will need to manually click the launch button), the computation will be submitted by IEE to the appropriate computing resources and the user will be informed of the status of the given step (this information is automatically updated in the IEE pipeline window).

A pipeline step may be in the following states:

- **Waiting for input** – the required input data is not available and must be uploaded or staged into the pipeline input directory
- **Ready** – the step is ready for execution (i.e. all required input is present) but has not actually been submitted to HPC
- **Waiting for execution** – the step has been submitted to the computational resources and we are awaiting acknowledgement of submission
- **Queued** – the computation has been accepted by the queuing system deployed on the underlying HPC resources
- **Running** – the computation has been assigned computational resources and is ongoing
- **Finished** – the computation finished successfully, and its output has been stored in PROCESS file storage
- **Error** – an error has occurred during execution of the computation (the user may then download and review the corresponding error log).

The IEE also provides a widget where pipeline outputs may be retrieved directly through a Web browser (assuming they are of reasonable size). In addition, the full contents of the PROCESS file storage (filtered according to user access permissions) may be browsed by selecting the **Files** option in the left-hand panel.

3 Future plans

The prototype described in this deliverable is, of necessity, a rudimentary version of project components. In the run-up to its deployment, we focused on integration of platform components, procurement of the required computational resources and registration of PROCESS services.

Following deployment of the presented prototype, work will focus on enabling multisite operation, primarily by integrating IEE with the LMU and, later on, SARA computing clusters. In addition, work on supporting PROCESS use case scenarios is currently ongoing and will be reported upon accordingly. We also intend to work on integration with the Cloudify platform, which will enable us to automatically stage in elements of the PROCESS computational workflows so that pipelines can be executed automatically at multiple sites in parallel.

Note that further details regarding implementation of each component of the PROCESS architecture can be found in Deliverable 4.2, which is released in parallel with this document.

4 Conclusion

The environment presented in this deliverable is a prototype, supporting execution of initial versions of the PROCESS use cases. As further features are introduced, we expect to be able to expand upon the set of available pipelines, eventually encompassing the entire set of PROCESS applications, while retaining sufficient genericity to provide support for arbitrary extreme large scale applications, deployable in parallel on multiple HPC infrastructures. This work is already ongoing and progress will be reported upon in future WP6 deliverables.

References

[Collage] Nowakowski P, Ciepela E, Hareźlak D, Kocot J, Kasztelnik M, Bartyński T, Meizner J, Dyk G, Malawski M. The collage authoring environment. *Procedia Computer Science*. 2011 Jan 1;4:608-17.

[D4.1] PROCESS Deliverable D4.1, Initial state of the art and requirement analysis; internal document

[D4.2] PROCESS Deliverable D4.2, Report on Architecture Evaluation; internal document

[EurValve] Bubak M, Bartyński T, Gubała T, Hareźlak D, Kasztelnik M, Malawski M, Meizner J, Nowakowski P. *EurValve Model Execution Environment in Operation*. Editors: Marian Bubak Michał Turała.:65.

[PL-Grid] The PL-Grid National Computing Infrastructure, www.plgrid.pl.

Appendix A: IEEE eScience Workshop

Workshop summary

Platform-driven e-infrastructure innovations

Organised in conjunction with eScience 2018 conference, Amsterdam, The Netherlands

On the 29th of October, 35 eScientists joined a half-a-day workshop “Platform-driven e-infrastructure innovations” to discuss extreme data services and their impacts, development paths and the road towards sustainability. The workshop was organised by the PROCESS project, with contributions from ESCAPE-2⁸, EPEEC⁹, EOSC-hub¹⁰ and DEEP-HybridDataCloud¹¹ projects. One of the main goals of the workshop was to bring together projects that had potential synergies and common interests that could serve as starting point for future joint activities that could be both deeper and broader in nature.

The workshop was opened by Nils Gentschen Felde (PROCESS project’s deputy coordinator), who presented the PROCESS project’s basic goals and organisational structure. This was followed by an opening talk “Towards Exascale Computing” by project coordinator Dieter Kranzlmüller. He raised several points related to how translation of HPC system benchmark figures into accelerated science is a multifaceted challenge that requires paying attention to several organisational, technical and operational issues that may not be obvious in the abstract. As a concrete, recent example of the scientific impact of increasing available computational capabilities by orders of magnitude, Dieter Kranzlmüller reported the initial findings of genetic mapping related to breast cancer therapies. A research team was given the access to full 6.8 PFLOP system for the duration of a weekend (between the end of a maintenance break and start of the normal, scheduled use). In addition to the expected quantitative benefits, this activity produced immediate results that were qualitatively fundamentally different from ones achieved before.

The vision of the Exascale Landscape was followed by a data-driven literature review presented by Stijn Helden from NLeSC. Due to the large volume of the exascale-related publications a traditional literature review would be difficult (if not impossible). To overcome this limitation, full texts of the exascale-related publications were downloaded from Elsevier Scopus and analysed and clustered e.g. based on keywords and citation networks. Already a preliminary analysis could identify some issues that are relevant to the identification of essential competences for successful exascale deployment that may not be immediately obvious. For example, the fault tolerance of the system was a recurring theme in the publications analysed, which might allow us to identify communities with relevant, complementary skills and knowledge that could speed up the exascale deployment. On the other hand, the applications used in the domain sciences to be supported by the next generation HPC were typically not discussed (at least in detail) in the analysed exascale publications.

These overview presentations were followed by the presentation of the PROCESS use cases, system architecture and its components and finally a live demonstration of the first prototype implementation of the PROCESS platform. The SKA/LOFAR pilot¹² was presented in detail by

8 https://cordis.europa.eu/project/rcn/215821_en.html

9 https://cordis.europa.eu/project/rcn/215832_en.html

10 <https://eosc-hub.eu/>

11 <https://deep-hybrid-datacloud.eu/>

12 <http://www.process-project.eu/projects/square-kilometre-array-lofar-ska/>

Jason Maassen (NLeSC) with Matti Heikkurinen (LMU) presenting the overview of the use cases and some details of the four other pilots.

After a coffee break, the contributing projects presented their approaches and goals. ESCAPE-2 project (Energy-efficient SCalable Algorithms for weather and climate Prediction at Exascale) was presented by Andreas Mueller (European Centre for Medium-Range Weather Forecasts, ECMWF). Project's goal is to make it possible to increase the resolution of the global weather model applications from the current 18km to 5km. This requires a comprehensive rethinking of the software and hardware solutions – both in order to reach sufficient performance and to remain within the constraints imposed by the energy consumption. The work presents also a formidable data management challenge, with the current archives of weather prediction data exceeding 290PB and growing at the rate of 200TB per day.

The EPEEC project (European joint Effort toward a Highly Productive Programming Environment for Heterogeneous Exascale Computing) was presented by Antonio J. Peña. The project aims at creating a parallel programming environment that supports application developers to utilise exascale supercomputers efficiently. This support needs to cover heterogeneous application mix: compute-, data-intensive, and extreme data solutions. The EPEEC solution includes advanced analysis of the application code, including automatic code annotation and analysis of the code patterns (often-repeating software components that can have a fundamental impact on the reliability and efficiency of the application).

The DEEP project (Designing and Enabling E-infrastructures for intensive Processing in a Hybrid DataCloud) was presented by Martin Bobák (UISAV). The project aims at creating modular, reusable components for deep learning, post-processing and on-line analysis of data streams. The component library approach¹³ aims at providing – in addition to performance benefits – interoperability between existing and emerging IaaS solutions.

As a final contributed presentation, Mark van de Sanden (NLeSC) presented the vision and current status of the EOSC initiative (European Open Science Cloud). The topic of special interest was the EOSC service portal – the ways the projects participating in the workshop could become service providers was discussed in some detail. The final set of presentations before the panel presented the PROCESS architecture in detail, including the containerised approach and principles of the service orchestration. The presentations can be found from the workshop page¹⁴ on the PROCESS website.

A panel discussion with all the speakers finalised the workshop. The discussion was kicked off by a provocative question: how the current exascale and extreme data initiatives are different from the initiatives that have preceded them during the last two decades. The discussion concluded that while ideas themselves may have reached maturity earlier on, “things start to work” only once the surrounding infrastructure and interest from key stakeholders reach a sufficient level. On the other hand, it was noted that some of the working solutions (such as GridFTP) from the earlier efforts tend to survive and evolve in the context of new efforts.

In conclusion, there was a strong consensus that further cross-project events like this initial workshop would be very useful tools for speeding up both to increase awareness of the solutions being developed and to encourage reuse and uptake of the existing solutions. Thus, in addition to possible joint publications and bilateral collaborations, the participants agreed to study possibilities to organise a similar event in roughly 6 months' time, ideally with a larger number of projects representing a broader range of key stakeholders.

¹³ Which could be seen as being similar, but complementary, approach to code pattern approach used by the DEEP project

¹⁴ <http://www.process-project.eu/workshop/platform-driven-e-infrastructure-innovations-workshop-agenda/>