D2.3 – Intermediate report of ACROSS co-design, system documentation and lesson learned together activities

Deliverable ID	D2.3
Deliverable Title	Intermediate report of ACROSS co-design, system documentation and lesson learned together activities
Work Package	WP2
Dissemination Level	PUBLIC
Version	3.0
Date	2022 – 08 – 31
Status	Final
Deliverable Leader	IT4I
Main Contributors	Svaton V. (IT4I)

Published by the ACROSS Consortium

The ACROSS project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955648. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Italy, France, Czech Republic, United Kingdom, Greece, Netherlands, Germany, Norway.

Document History

Version	Date	Author(s)	Description
0.1	2022-06-20	IT4I	ToC (Draft)
0.2	2022-07-11	IT4I, ATOS	Included input from CINECA and ATOS
0.3	2022-07-12	IT4I	Filled in the Executive summary and Introduction
0.4	2022-07-12	AVIO	Filled in WP5 LL
0.5	2022-07-13	IT4I	Filled in LL intro, methodology, general LL
0.6	2022-07-14	IT4I	Filled in System documentation, Deployment status
0.7	2022-07-14	ECMWF	Filled in LL
0.8	2022-07-15	LINKS	Provided inputs
0.9	2022-07-18	IT4I	Conclusion, architecture introduction
1.0	2022-07-19	IT4I	Finalized the architecture chapter
1.1	2022-07-19	IT4I	Final touches, preparation for the internal review
2.0	2022-07-29	IT4I	Incorporated inputs from the reviewers (LINKS, ATOS)
3.0	2022-08-25	LINKS	Chapter on co-design finalized

Table of Contents

Glossary

Acronym	Explanation
AI	Artificial Intelligence
API	Application Programming Interface
BD	Big Data
CPU	Central Processing Unit
CWL	Common Workflow Language
DL	Deep Learning
FPGA	Field Programmable Gate Array
FMLE	Fast Machine Learning Engine
GPU	Graphic Processing Unit
HPC	High-Performance Computing
HPDA	High-Performance Data Analytics
HW	Hardware
LL	Lessons Learned
ML	Machine Learning
NUMA	Non-Uniform Memory Access
SW	Software
VPU	Vision Processing Unit
WARP	Workflow-aware Advanced Resource Planner

List of figures

List of LL

Executive Summary

The ACROSS project aims at building an exascale-ready, HPC and data-driven execution platform, supporting modern complex workflows mixing HPC, BD and AI high-level tasks, by leveraging on an innovative software environment running upon advanced heterogeneous infrastructural components including GPUs, FPGAs and neuromorphic processors, as well as innovative smart resource allocation policies and job scheduling algorithms, up to the management of tasks inside jobs.

From the co-design perspective and in respect to the ACROSS's project timeline the initial pilot's requirements were gathered, consolidated and summarized in Deliverable D2.1 - Summary of pilots co-design requirements. Then the evaluation of existing solutions and the selection of tools that will become backbone of the ACROSS platform was performed and described in Deliverable D2.2 - Description of key technologies and platform design. At the current state of the project (M18) the first Alpha version of the ACROSS platform was deployed and is available to the pilots to execute and benchmark their use-cases. This deliverable summarizes the current state of the ACROSS platform, provides information about the existing system documentation and as a part of the Lessons Learned activity describes the various issues and challenges that were encountered during the implementation and integration processes.

Position of the deliverable in the whole project context

This deliverable is mainly part of Task 2.2 - User driven co-design and Lesson Learned. It summarizes the main components of the ACROSS architecture following the Milestone 3 ‘ALPHA version: ACROSS platform and technologies and Lessons Learned that were identified during the co-design processes, providing initial feedback to the technical selections and integrations.

The main contributors of this deliverable are both the ‘technical’ work packages (WP2, WP3, WP4) and the pilot’s work packages (WP5, WP6, WP7):

- IT4I as the WP2 Co-design leader and the HPC infrastructure provider
- CINECA as the HPC infrastructure provider
- ATOS as the WP3 Heterogeneous hardware and acceleration leader and experimental infrastructure provider
- LINKS as the WP4 Orchestration leader
- AVIO Aero as the WP5 Aeronautics pilot leader
- ECMWF as the WP6 Weather and Climate pilot leader
- SINTEF as the WP7 Energy and carbon sequestration pilot leader

Besides the main partners listed above also other partners (NP,CINI,INRIA,DELTA,MPI-M,MORFO) participating in a said work package contributed to this deliverable.

Description of the deliverable

Apart from the general introduction and the relation to the other deliverables this document is structures into several main chapters. Chapter 2 describes the evolved ACROSS architecture. It provides overview of the SW tools and HW technologies that were deployed as the first version of the platform and provides the information on where to find relevant documentation. Chapter 3 focuses on Lessons Learned that were identified during the previous milestones and mainly related to the Milestone 3 - Deployment of the first version of the ACROSS platform and Milestone 4 - Pilot’s use-cases integration. This document then closes with Chapter 4 containing a short summary of the achievements that have been made and an outlook on the development progress.

1 Introduction

The ACROSS project will co-design and implement a platform supporting the execution of HPC and data driven HPC workflows, by integrating innovative and advanced hardware and software technologies. To this end, ACROSS project will focus on the integration of heterogeneous accelerators aiming at efficiently executing complex workflows, as well as monitoring the status of underlying computing infrastructures. Data-driven application workflows will benefit from the integration of innovative HPDA solutions, ranging from user exposed API to hardware acceleration support. Co-design activity will ensure to build an integrated execution environment, where users will easily define and submit their workflows, by means of APIs that help hiding complex management procedures (e.g., selection of computing resources).
This deliverable is the outcome of the WP2 - Cross stack convergence & Co-design for HPC and Data driven HPDA software environment and mainly task T2.2 - User driven co-design and Lesson Learned. In the scope of this task the ACROSS project is designing an overall ACROSS architecture and defining a set of technical specifications to describe the platform. Based on the pilot's use cases, the ACROSS platform is being designed by co-design approach that involves several iterations with technical and pilot's work packages. Requirements and lessons learned from the different pilot's use cases serves as a base for the platform's architecture updates and, in the end, should form the final ACROSS platform. This deliverable is the mid-project report on the current state of the platform and the ongoing activities.

1.1 Scope

The scope of Deliverable D2.3 is to provide an intermediate report of ACROSS co-design, system documentation and lesson learned together activities. From the project's time-line perspective it is positioned in the middle of the project just between the two milestones M3 and M4 focusing on the deployment of the first version of ACROSS platform and integration of pilot's use-cases. At this stage of the project based on the initial pilot's requirements the main components of the platform were already identified and described, the first minimal version of the platform was deployed and is available to pilots, and pilots started with the integration and testing of their use-cases. This document summarizes the current state of the deployment and the issues encountered during these activities in form of a Lessons Learned documented experiences.

1.2 Related documents

ID	Title	Reference	Version	Date
[RD.1]	Summary of Pilots co-design requirements	D2.1	3.0	2021-08-31
[RD.2]	Description of key technologies and platform design	D2.2	0.8	2021-11-30
[RD.3]	System Requirements Analysis for Orchestrator Design	D4.1	0.7	2022-02-28

2 ACROSS Co-design intermediate status assessment

At M18, corresponding to the MS4 milestone and half way through the project, the co-design process has defined all the requirements, both functional and technological, it has described the complex workflows to be build and executed by the pilots and it has consolidated the ACROSS platform's structure to achieve all the envisioned objectives. While some technical aspects are still subject to modification before the end of the project, the identification of the required technologies has been performed with the necessary detail and accuracy at this stage of the project.
In the first phase of the co-design process pilots have defined the specific wokflows to be implemented in the ACROSS platform for each use-case (Greener aero-engine, Weather, Climate, Hydrological and Farming, Energy and Carbon Sequestration), together with the related KPIs. From the analysis of such workflows, the hardware and software requirements for the platform have been identified. Deliverable D2.1 report all the details in this regard.
Based on the identified requirements, available technologies have been analyzed and selected, for both hardware (accelerators, optimal memory and CPU, etc.) and software (programming languages, libraries, tools, etc.). The full list along with the description of key technologies, as well as the platform design, are reported in the deliverable D2.2.
Finally, the ACROSS platform architecture, described in detail in Chapter 2, has been designed composing these technological bricks in order to fulfill both the pilots' requirements and the overall project objectives. This is the main outcome of the co-design process.
Figure 1 tries to summarize the full range of the co-design results in relation to the initial platform design provided in the DoA. As such, the figure highlights some major elements emerged during the co-design activity. In particular, the Common Workflow Language (CWL) has been identified has the mean for describing pilot workflows, while hardware acceleration (GPUs, FPGAs, etc.) will be extensively applied to *i*) maximize application performance along with the energy efficiency and *ii*) to support the emulation of innovative processing architectures (i.e., neuromorphic). In this regards, ACROSS consortium decided to leverage infrastructural compute, storage and networking resources provided by two HPC centers (each making available their resources in the form of HPC or Cloud instances), which are complemented by those provided by ATOS through its research infrastructure. Machine-/Deep Learning modeling as well as the support to neuromorphic software stack has been considered for the integration in the orchestration stack.

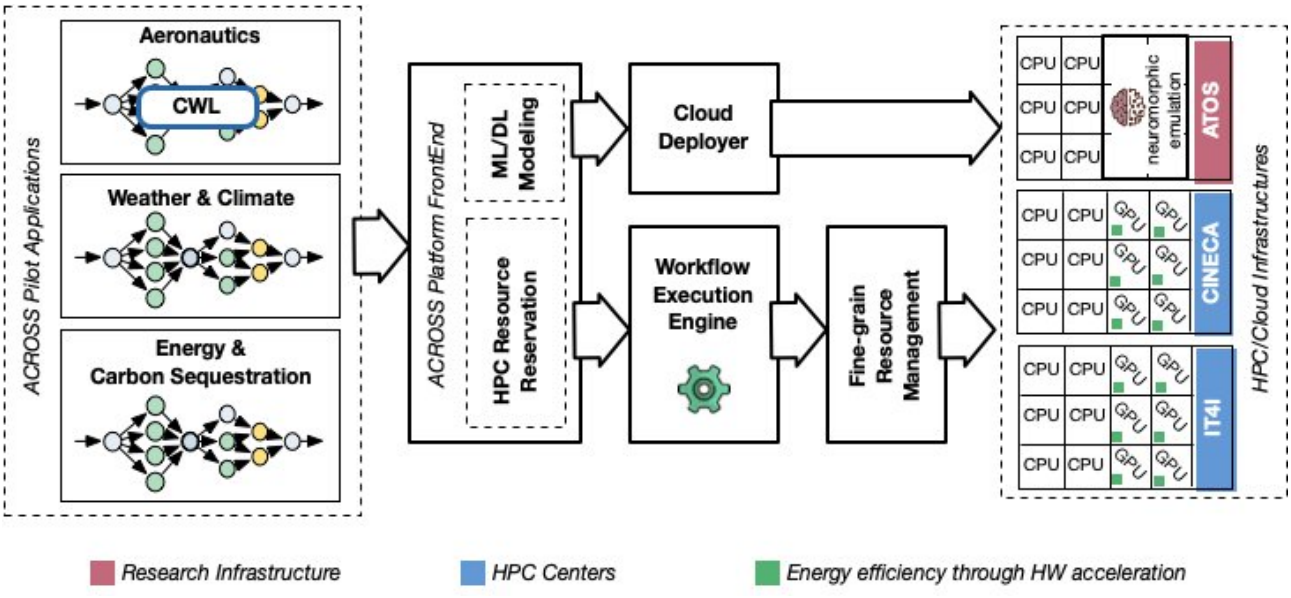

Figure 1 - ACROSS co-design solution

As mentioned before, during the remaining time of the project, the co-design effort will monitor the development of the platform and will identify potential issues and challenges in order to solve them or mitigate their impact. Therefore, additional Lessons Learned and possibly adjustments to the overall ACROSS platform design and architecture will possibly be considered. These additional LL and the final version of the ACROSS platform will

be reported in Deliverable D2.5 *"Final report of ACROSS co-design, system documentation and lesson learned together"* at M36, while Deliverable D2.6 *"Final validation for cross stack convergence"* will report the platform validation results.
In the following Chapter, the full detail of the ACROSS Architecture is provided.

3 ACROSS Architecture and system documentation

This chapter provides a description of the revised ACROSS architecture that was deployed as the first alpha version of ACROSS platform. The entire ACROSS platform consists of the HW stack represented by the available computing infrastructures and the SW stack deployed and available to platform users. Following chapters provides a more detailed explanation of this layered architecture together with the overview of the approved computational resources for the pilots, ending with the summary of the current deployment status.

3.1 ACROSS Architecture

The revised architecture of the ACROSS platform can be simply separated in three main layers: presentation layer, orchestration layer, and infrastructure layer. Figure 2 illustrates this layered architecture together with the main component that were identified during the requirements phase of the project and currently deployed as the alpha version of ACROSS platform (with the exception of the two grayed-out modules that will be deployed in the next phase of the project).

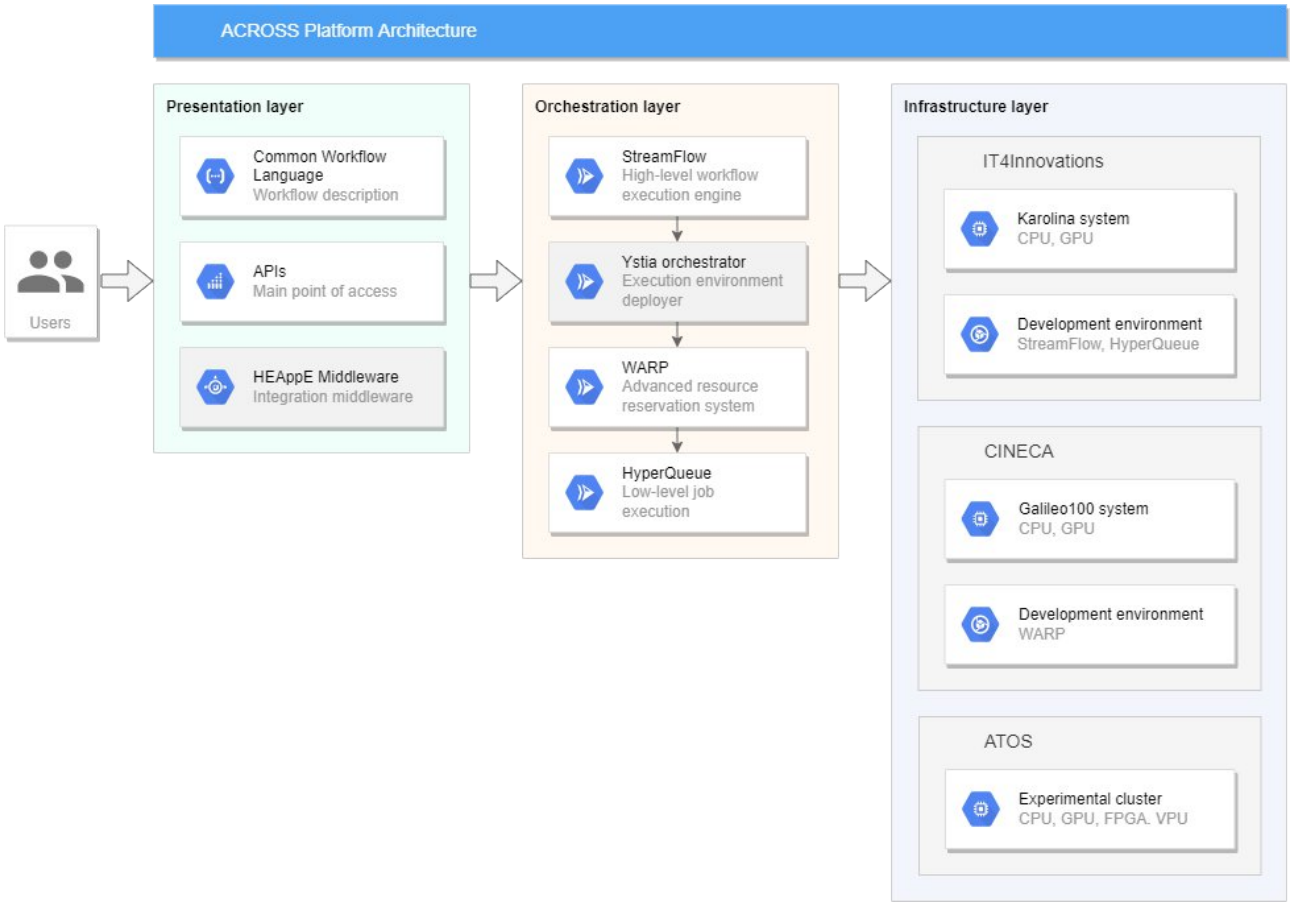

Figure 2 - ACROSS platform architecture

Presentation layer represents the platform's access points in terms of simple interaction and integration. Orchestration layer orchestrates the submitted workflow across multiple orchestration levels (from high-level point of view to the low one; focus on job vs task), and finally executing it on the selected computing environment that is represented as an infrastructure layer of the platform. Below is a more concrete explanation of the layered architecture.

1. **Presentation layer**

Presentation layer can be considered as platform API layer. It provides a set for APIs for the high-level interaction with the ACROSS platform. Using this API the users are able to submit a workflow to ACROSS platform to be orchestrated and executed by the underlying orchestration subsystem.
For HPC environments the pilot workflows are described using the Common Workflow Language (CWL) which serves as the main input for the high-level workflow execution engine represented by StreamFlow. For cloud environments Alien4Cloud will serve as the front-end and Ystia as the cloud execution engine.
To enable easy access and integration of ACROSS platform with other EU platforms the HEAppE Middleware is being considered as secure middleware with an easy-to-use REST API to serve as a bridge between the solutions and services developed in the scope of ACROSS project and other relevant service and platform providers.

2. Orchestration layer

The ACROSS orchestration architecture has been described in detail in D4.1. In short, starting from the identified system requirements, the key technological components, which have been identified and comprehensively described in D2.2, have been organized in three main architectural levels:

- *(High-level) workflow execution engine*: to allow proper execution of (complex) workflow's graph by easily matching with proper execution environments, StreamFlow has been chosen as workflow execution engine, as it provides the necessary capability and expressiveness.
- *Execution environment deployer*: to match the workflow's graph steps with the specific execution environments it is necessary to deploy these environments on the computing resources (both Cloud and HPC). Ystia, composed of the front-end named Alien4Cloud (A4C) and the back-end engine named YORC, has been selected for that purpose on the Cloud side.
- *Advanced (HPC) resource reservation system*: the WARP (Workflow-aware Advanced Resource Planner) module will leverage (and build on top of) the advanced reservation features exposed by current-generation batch schedulers to properly reserve (in advance) specific sets of HPC resources, with the goal to overcome the limitations such as the lack of workflow-awareness and limited capability to exploit HW heterogeneity of modern batch schedulers.
- *(Low-level) job execution on heterogeneous environments*: to exploit the heterogeneity of computing resources, HyperQueue will allow for finer grain control of the underlying resources by scheduling tasks contained in the jobs submitted to the batch scheduler.

Finally, an additional module, FMLE (Fast Machine Learning Engine) will provide a high-level interface for designing, training, and deploying ML/DL models as it leverages a direct connection with A4C to generate specific application templates and execute them through the YORC engine.
The scheme of the overall solution, composed of the various modules and their interaction is reported in Fig.3 of D4.1.

3. Infrastructure layer

Infrastructure layer is represented by a set of computing infrastructures available to users of the ACROSS platform. This layer consists of IT4I's Karolina system, CINECA's Galileo100 system and ATOS' experimental infrastructure. As these infrastructures were extensively described in the previous deliverable D2.2 below is just a brief overview.

IT4Innovations Karolina system

Karolina is the latest and most powerful supercomputer cluster built for IT4Innovations in Q2 of 2021.

- The Karolina cluster consists of 829 compute nodes, totaling 106,752 compute cores with 313 TB RAM, giving over 15.7 PFLOP/s theoretical peak performance and is ranked in the top 10 of the most powerful supercomputers in Europe.
- Nodes are interconnected through a fully non-blocking fat-tree InfiniBand network, and are equipped with AMD Zen 2, Zen3, and Intel Cascade Lake architecture processors. Seventy two nodes are also equipped with NVIDIA A100 accelerators.
- The user data shared file-system and job data shared file-system are available to users. The PBS Professional Open Source Project workload manager provides computing resources allocations and job execution.

CINECA Galileo100 system

Details about Galileo100 (G100) cluster hardware have been already described in the previous Deliverable 2.2 in which the ACROSS infrastructure have been extensively presented. Following with just a brief summary:

- The G100 system architecture provides computing nodes, each equipped with 2 CPU Intel CascadeLake 8260. Some data processing nodes consist of 3TB of Intel Optane memory. Additional nodes are equipped with 2 NVIDIA GPU V100. Servers OpenStack are available for cloud computing (ADA CLOUD), and there are 20 PB of active storage accessible from both Cloud and HPC nodes.

In this following part we want to detail more about the Hardware infrastructure exploited by the alpha version of ACROSS platform:

- INTEL OPTANE nodes on G100 cluster, these nodes are able to extend the memory till 3TB. This memory can be available as two main modes: “persistent memory” or “cache/app-direct” (here some additional details https://www.boston.co.uk/blog/2019/07/10/intel-optane-dc-persistant-memory.aspx
- INFINITE MEMORY ENGINE (IME) from DDN, caching system based on Flash, it is “located” among applications and filesystem in order to improve I/O operations. In the G100 actual configuration there are something like 720 TB of fast storage visible from the entire machine (compute nodes). See DDN IME Preso (uni-hamburg.de) for additional details. https://www.youtube.com/watch?v=gh6FLqt_038

ATOS experimental infrastructure

The cluster under construction at ATOS is composed of state-of-the-art computing technologies (Intel/SKX CPU, Nvidia/V100 GPU, Altera/Stratix10 FPGA, Intel/VPU MyriadX, etc.), memory access acceleration and large storage capacity. The software stack is characterized by the presence of modules dedicated to AI (e.g. TF2/Keras) operating on acceleration layers such as Intel/oneAPI and Intel/OpenVINO.

Computational resources provided to pilot’s

IT4Innovations

	WP5
Name	EU2010Temporary: ACROSS - WP5
Lifetime	2022-01-19 to 2023-01-19
Call type	EuroHPC JU Development Access Call
PI	Ennio Spano (Avio Aero)
Allocation	1,920,000 core-hours
System	Karolina

	WP6
Name	ACROSS-WP6
Lifetime	2022-06-06 to 2023-06-06
Call type	EuroHPC JU Development Access Call
PI	Emanuele Danovaro (ECMWF)
Allocation	1,920,000 core-hours
System	Karolina

	WP7
Name	ACROSS WP7 – Energy and Carbon Sequestration Pilot - CPU
Lifetime	2022-02-25 to 2023-02-25
Call type	EuroHPC JU Development Access Call
PI	Kjetil Olsen Lye (SINTEF)
Allocation	1,920,000 core-hours
System	Karolina

	WP7
Name	ACROSS WP7 – Energy and Carbon Sequestration Pilot - GPU
Lifetime	2022-02-25 to 2023-02-25
Call type	EuroHPC JU Development Access Call
PI	Kjetil Olsen Lye (SINTEF)
Allocation	384,000 core-hours
System	Karolina

CINECA

	WP5
Name	ACROSS WP5
Lifetime	2021-07-01 to 2021-10-31
Call type	CINECA
Allocation	120,000 core-hours
System	Galileo100

	WP6
Name	ACROSS WP6
Lifetime	2021-09-02 to 2022-10-31
Call type	CINECA
Allocation	195,000 core-hours
System	Galileo100

	WP7
Name	ACROSS WP7
Lifetime	2021-07-12 to 2021-12-31
Call type	CINECA
Allocation	100,000 core-hours
System	Galileo100

	ICEI
Name	icei_Giovann
Lifetime	2021-10-13 to 2022-10-12
Call type	EuroHPC JU ICEI Call
Allocation	2,400,000 core-hours
System	Galileo100

3.2 Deployment status summary

At the current stage of the project ACROSS platform consists of two available HPC infrastructures (IT4Innovations Karolina system and CINECA Galileo100 system), two development environments deployed for the purpose of developing and testing the platform's services and modules (IT4Innovations and CINECA) and Atos' experimental infrastructure. The SW stack of the platform includes the tools StreamFlow, HyperQueue, and WARP which is still under development.

IT4I's development environment

To provide a testing environment for the pilots to execute their workflows the StreamFlow and HyperQueue tools were deployed. Pilots are able to access the VMs and test the execution of the workflows described by CWL.

2 x Virtual Machine
- across1.it4i.cz, across2.it4i.cz
- 2 cpus
- 4 GB RAM
- 50 GB HDD
- CentOS

CINECA's development environment

Dedicated mini cluster for development and testing of Slurm plugin:

- 10 standard nodes of G100 system
- The system partition will be put out of production and dedicated to testing activity of a slurm plugin
- This plugin will be able to request reservations on demand on behalf of the WARP (WP4 activity)

Cloud resources to deploy development virtual machines. These VMs will be used as a testing environment for CWL workflows and for hosting some orchestrations tools provided by WP4. The total amount of available resources is:
- 12 vCPUs
- 90GB RAM
- 100 GB of storage

ATOS experimental infrastructure

The detailed specification of Atos' experimental infrastructure was provided in chapter 2.1. This infrastructure is available to users under Atos' standard access request procedure as described in a dedicated document 'Request to access Atos Nova platform', which has been already completed by relevant partners.

3.3 System documentation

In the scope of the previous activities, milestones, and deliverables the main software components of the ACROSS platform were identified, the overall ACROSS architecture was devised and the first 'minimal' version of the platform was deployed. In relation to the system documentation activity the SW module

providers/maintainers of these components provided a unified description of the component/tool that contains the following information:

- Components name
- Developed by
- License
- Website
- About
- Source codes
- API
- HW requirements
- SW requirements
- Deployment/Installation instruction

The provided information was aggregated into a dedicated document 'SW catalogue' that serves as a first-level system documentation containing all of the necessary information on where to find the component's source-codes, what are the requirements, how to deploy them, etc.

The list of the main components of ACROSS platform from SW catalogue document:

– please note that not all of the listed components have been deployed in the first version of the ACROSS platform (M3)

Name	**Workflow-aware Advance Resource Planner (WARP)**
Developed by	LINKS Foundation
License	Not yet decided
Website	Not yet available
About	Smart planning and on-demand acquisition of (HPC) compute resources supporting the execution of workflows mixing numerical simulations, ML/DL and high-performance data analytic
Source codes	(Front-end): https://git.mycloud-links.com/across/wp4/warp-frontend (Back-end): https://git.mycloud-links.com/across/wp4/warp-backend
API	Command Line Interface (CLI) through the front-end module
HW requirements	(Minimum) 2 cores, 8GB RAM, 5GB Storage; (Recommended) 8 cores, 16GB RAM, 10 GB Storage
SW requirements	(Front-end) Python >= 3.7 and Python <=3.10; (Back-end) MySQL (latest release with standard InnoDB engine set up), Rust compiling toolbox (latest release)
Deployment/Installation instructions	Will be made available on the website/code repositories

Name	**StreamFlow**
Developed by	Università di Torino (UniTO)
License	LGPLv3
Website	https://streamflow.di.unito.it/
About	Container-native workflow manager for hybrid infrastructures
Source codes	https://github.com/alpha-unito/streamflow
API	Command Line Interface
HW requirements	1 core, 2 GB RAM
SW requirements	MacOS / POSIX OS, Python >=3.8
Deployment/Installation instructions	https://streamflow.di.unito.it/documentation/latest/install.html

Name	HyperQueue
Developed by	IT4Innovations
License	MIT
Website	https://it4innovations.github.io/hyperqueue/stable/
About	HyperQueue (HQ) lets you build a computation plan consisting of a large amount of tasks and then execute it transparently over a system like SLURM/PBS. It dynamically groups jobs into SLURM/PBS jobs and distributes them to fully utilize allocated notes. You thus do not have to manually aggregate your tasks into SLURM/PBS jobs.
Source codes	https://github.com/It4innovations/hyperqueue
API	https://it4innovations.github.io/hyperqueue/stable/cli/shortcuts/
HW requirements	min. VM, 2cpu, 4GB ram, 50GB HDD
SW requirements	https://it4innovations.github.io/hyperqueue/stable/installation/
Deployment/Installation instructions	https://it4innovations.github.io/hyperqueue/stable/deployment/

Name	Fast Machine Learning Engine (FMLE)
Developed by	Atos (BDS R&D)
License	Atos proprietary (subject to change)
Website	Part of Codex AI Suite : https://atos.net/fr/solutions/codex-ai-suite
About	ML/DL toolbox for HPC, Hide complexity of HPC jobs management for AI Model Management (training...)
Source codes	internal
API	available in fmle distribution
HW requirements	gateway node : 8 cpus, 16G ram and a Posix (lustre/nfs) file system shared between all compute and gateway nodes
SW requirements	nginx/lua, postgreSQL, bindfs, docker, yorc and a4c stacks
Deployment/Installation instructions	available in fmle distribution

Name	Alien4Cloud
Developed by	Atos
License	Open Source, Apache 2.0
Website	https://alien4cloud.github.io/
About	UI - Yorc front-end (TOSCA catalog, editor to create TOSCA applications, relies on Yorc for Applications Lifecycle and workflows management)
Source codes	https://github.com/alien4cloud/alien4cloud
API	REST API https://alien4cloud.github.io/documentation/3.4.0/rest/overview.html
HW requirements	java application, at least 2 CPUS, 8 GP of RAM
SW requirements	https://alien4cloud.github.io/#install-launch-and-configure-alien4cloud
Deployment/Installation instructions	Ansible playbooks at https://github.com/alien4cloud/alien4cloud-spray or yorc bootstrap at https://yorc.readthedocs.io/en/latest/bootstrap.html

Name	YSTIA / YORC
Developed by	Atos (BDS R&D)

License	Open Source, Apache 2.0
Website	https://ystia.github.io/
About	TOSCA based orchestrator (incl. TOSCA Forge, Application lifecycle and workflows management, hybrid Cloud/HPC support).
Source codes	https://github.com/ystia/yorc
API	REST API https://github.com/ystia/yorc/blob/develop/rest/http_api.md
HW requirements	https://yorc.readthedocs.io/en/latest/install.html#host-requirements
SW requirements	https://yorc.readthedocs.io/en/latest/install.html#packages-installation
Deployment/Installation instructions	https://yorc.readthedocs.io/en/latest/bootstrap.html

Name	**Damaris**
Developed by	Inria (KerData team, Inria Rennes Bretagne-Atlantique)
License	Open Source, LGPL
Website	https://project.inria.fr/damaris/
About	Middleware for asynchronous I/O and data management targeting large-scale, MPI-based HPC simulations.
Source codes	https://gitlab.inria.fr/Damaris/damaris
API	https://project.inria.fr/damaris/instrumenting-a-simulation/ (C/C++, Fortran, XML)
HW requirements	https://project.inria.fr/damaris/environment-preparation/
SW requirements	https://project.inria.fr/damaris/environment-preparation/
Deployment/Installation instructions	https://project.inria.fr/damaris/documentation/

Name	**High-End Application Execution Middleware**
Developed by	IT4Innovations national supercomputing center (www.it4i.cz)
License	Open-Source GPL-3.0 License
Website	http://heappe.eu/
About	HPC remote execution middleware. Implementation of an HPC-as-a-Service concept. Provides secure remote access to HPC without the need to manually register within the HPC center. For more information see the official website.
Source codes	https://github.com/It4innovations/HEAppE
API	REST API (Open API Specification v3) API example: https://heappe.it4i.cz/presentation/swagger/index.html
HW requirements	see HEAppE Middleware deployment manual (attached in pdf)
SW requirements	see HEAppE Middleware deployment manual (attached in pdf)
Deployment/Installation instructions	see HEAppE Middleware deployment manual (attached in pdf)

4 Lessons Learned on technical development and integration

In general Lessons Learned (LL) is an effective tool to prevent repeated mistakes by an effective and targeted transfer of information and experience not only in production areas of industrial enterprises. The process is based on the principle of learning from one's own mistakes and from one's own experiences. LL is an input for improvement and standardization, for further development of processes and methods and regulations, work procedures, product design process etc. LL methodology is often used to convey information about causes and proven and successful corrective measures to known errors so that they do not occur again within the given project, but also for similar work processes. LL should not be focused only on negative aspects, but it makes sense to include positive cases as well.

4.1 LL methodology

This methodology chapter is focusing on the three important questions related to the LL - *why* are we collecting LL, *what* are we collecting and *how* to organize the collected information.

Why the LL collection is important?

Collecting and monitoring LL is an essential activity of any project throughout the project's life cycle. As described in the general introduction of this chapter it gives us a tool to learn from the previous mistakes and achievements to implement good practices and maximize the success of the entire project in terms of improving the efficiency and quality of developed services and platform.

Objectives of LL:
- improving implementation approaches
- preventing and minimizing the risk of failures
- improve the planning of project's phases

What LL are collected?

In general, the LL is a documented information knowledge gained from the process of conducting a project's objectives. With regards to the ACROSS project the following type of information is being collected as LL:

- best practices
- challenges and issues
- differences between expected and achieved objectives

LL structure:
- General information: Author, LL title, LL brief description
- Activities leading to this LL
- Challenges or issues
- Envisioned solutions

How are the LL organized?

Based on the type of the information that we want to collect and the clear separation of 'technical' and pilot WPs in the ACROSS project the LL were separated into three parts:

- Technology related LL
  - contains LL related to the technology and infrastructure providers, and orchestration service developers that deals with the setup of the infrastructures, access and security policies, preparation of the ACROSS platform etc.
- Pilot related LL
  - focuses on LL that relates directly to the pilots in terms of pilot's code deployment, code benchmarking, license issues etc.
- General LL
  - contains LL that relates to the entire project consortium and not only a selected WP or a specific activity of a single WP/partner.

4.2 Technology related LL

Author	LL title	LL brief description
ATOS (WP3, WP5, WP7)	**Understanding and Tuning the Performance of HPC Applications**	Deep understanding of HPC architectures and theory behind algorithms parallel computation are crucial to guide the developers for an optimal implementation of their applications on a heterogeneous HPC architectures.
Questions		**Answers**
Activities leading to this LL		Assistance to WP5 (ANSys Fluent simulation) and WP7 (OPM flow) for a better understanding and then an improvement of performance of their respective pilots.
Challenges or issues		Requires a good technical knowledge of HPC architectures with its variety of computing and acceleration technologies as well as parallelization techniques in function of these architectures including programming models and their combination with placement and scheduling. Domain experts typically lack this level of understanding, while HPC experts lack the necessary understanding of the application details.
Envisioned solutions		Acquisition of a broad knowledge for the development of an effective working method based on the profiling and measuring of criteria affecting performance.

Lesson Learned 1

Author	LL title	LL brief description
IT4I (WP2, WP5)	**Respecting Karolina system CPU architecture**	This LL relates to AVIO LL 'Karolina performance analysis'. Users utilizing the new IT4I's Karolina HPC system usually do not know the exact CPU architecture in terms of NUMA domains that is different from the CPUs previously used in IT4Innovation's older system.
Questions		**Answers**
Activities leading to this LL		Initial benchmark of WP5 pilot use-case on IT4I's Karolina system showed better results when not fully utilizing the entire compute nodes.
Challenges or issues		The most likely cause for this behavior is a specific NUMA architecture of Karolina system CPUs. Not respecting this architecture might cause a noticeable slowdown when executing memory bound applications while 'overloading' the compute nodes with the hope of maximizing the utilization of compute nodes.
Envisioned solutions		As this issue is a general one concerning all Karolina system users and not related only to ACROSS project the decision was made that the concrete explanation of NUMA architecture and 'how to' guidance for the users will be crated as a part of official IT4Innovations documentation.

Lesson Learned 2

Author	LL title	LL brief description
IT4I & CINECA (WP2)	Different access & security policies across multiple infrastructures	Different computational infrastructures often implement different security policies, user registration processes and infrastructure access methods. ACROSS project integrates three different computing infrastructures thus there was a need to simplify access to these infrastructures for pilots.
Questions		**Answers**
Activities leading to this LL		ACROSS project partners requesting access to computing infrastructures to deploy, test and benchmark their pilot use-cases or develop new ACROSS platform services.
Challenges or issues		To access CINECA resources users have to register into a dedicated UserDB portal. To access IT4Innovations resources users have to submit a registration request via a signed email using a digital certificate. To access ATOS infrastructure users have to submit a request access form. Each process with its unique requirements and specifications which might be confusing for the regular users.
Envisioned solutions		To provide support for ACROSS project partners and ease up the registration process the infrastructure providers created a dedicated documents explaining the registration process and the necessary requirements. These documents are located in the project's document storage under the 'Access to infrastructures' folder and are available to all project partners. In a medium perspective, the federated ICEI/FENIX infrastructure (in which G100 is inserted) can be seen as a valid point of reference of many relevant aspects (authentication and authorization infrastructure, VMs, scalable and interactive computing services...). The feasibility of FENIX integration will be evaluated in Task 2.5 as well as its integration/interaction with the LEXIS Platform.

Lesson Learned 3

Author	LL title	LL brief description
LINKS (WP4)	Access to "in advance allocation" feature on HPC infrastructures	Allowing HPC users and/or specific services, such as the WARP, to reserve computational resources in advance presents a number of challenges; the main ones being the compatibility with the batch schedulers in use in the HPC centers and with their specific policies. The creation of a SLURM plugin has been identified as the solution of this types of issues.
Questions		**Answers**
Activities leading to this LL		During the design phase of the orchestration architecture, the need for a module dedicated to the in advance provisioning of compute resources has been identified as one of the enabling features for the execution of complex workflows, such as the ones defined by the pilots. This led to the definition of the WARP module and of its required functionalities.
Challenges or issues		The WARP should be able to reserve computational resources in a point in time in the future in a deterministic way by interacting with the batch scheduler of the HPC center involved. However, these schedulers (mainly SLURM and PBS) allow a similar reservation feature only to system admins in the context of system testing and maintenance.

Envisioned solutions	As developing this type of feature directly in the WARP, i.e., outside of the scope of the batch schedulers, poses more challenges than it solves; thus, the creation of a specific plugin (SLURM at first) has been identified as the preferred solution. The plugin will exploit and/or extend the original advanced reservation feature of SLURM to allow a defined set of normal HPC users to access the information on the available resources as a function of time and to request reservation with specific characteristic. The idea is to allow this feature only for dedicated queues and resource partitions in order to facilitate the adoption in the HPC centers and to avoid the disruption of standard queues and scheduling policies.

Lesson Learned 4

Author	**LL title**	**LL brief description**
LINKS (WP2, WP4)	**Matching orchestration modules installation with the infrastructures**	The high-level orchestration architecture is composed of modules whose installation and functioning requires the access to infrastructural resources in a way that may not fit with the specific policies of the HPC centers. For instance, some orchestration module (FMLE, WARP) may require root access and the installation on the login node, or the execution in background as a service. HPC centers (e.g., IT4I) have restricted security policies that do not allow root access to login nodes, while in other cases the maximum allotted time for a process is limited (e.g., CINECA allows processes on the login node to last for at most 10 min). Installation of such components on a virtual machine living on the Cloud partition with standard access to the login and worker nodes has been identified as the solution. Also, the creation of a small, dedicated testbed (with few worker nodes, login node and a separated installation of SLURM) has been identified as a solution.
Questions		**Answers**
Activities leading to this LL		The design of the orchestrator architecture led to the integration of specific components that have been designed to work as service or require to run as daemons. Specifically, FMLE and WARP modules have these features; the analysis of the characteristics of the infrastructures at the IT4I and CINECA centers led to the identification of specific solutions for the deployment of these components, in order to test them and being able to demonstrate their functioning.
Challenges or issues		Supercomputing resources made available at IT4I and CINECA centers are managed with specific policies. These generally translate into no access as root, limitations on the placement of ssh keys, or a limited amount of time allotted to the processes for running (i.e., on the login node). As such, the deployment of FMLE and WARP module is affected since their installation and functioning require to cope with these policy restrictions. The main challenge is thus that of finding a proper configuration of the infrastructural resources made available, such as the modules can be properly deployed and integrated with the other orchestration components.
Envisioned solutions		To cope with these restrictions, a couple of solutions have been identified. The first one is that of using virtualized resources (VMs) hosted on the cloud partitions and with access to the HPC nodes (login and workers). As such, on IT4I, we devised a situation where the FMLE can be installed on a VM get access to other VMs as worker nodes. On CINECA side, the VM can access to the HPC worker nodes. WARP module has been devised to be installed on a VM too, since the above-mentioned limitations. A second solution has been also considered: on CINECA side, a small group of resources will be made available for creating a small

	testbed. As such, the testbed will comprise a service node and small group of worker nodes. The service node will be used for the installation of the SLURM environment, and since there will not be subject to previously mentioned restrictions, it will be possible to deploy and test the whole orchestration solution without the need for passing on the cloud partition.

Lesson Learned 5

Author	LL title	LL brief description
LINKS (WP2, WP4)	**Interaction of different SW components / modules**	The management of workflow's graphs as those coming from the Pilots required the identification of a set of components able to reserve resources, execute workflows' steps on top of them, and the fine control over the available resources. Once the components have been identified, there is need to properly connect them each other. This requires the definition of sequence of actions that the orchestrator components have to perform, based on which will be possible to implement proper software interfaces.
Questions		**Answers**
Activities leading to this LL		The co-design of the orchestrator led to the identification of a set of components to be integrated each other. The execution of a workflow requires a specific interaction between the components that has been defined.
Challenges or issues		Different components or ACROSS orchestration solution have capabilities that allow them to be used for multiple purposes. As such, there were an overlap of these capabilities among the components. Identifying the most suited one required to analyze their specific features; then, a proper execution sequence have been identified, clarifying the role of each component and the set of other components to interact with.
Envisioned solutions		Based on the analysis of the capabilities exposed by each component and their flexibility to fulfil specific purposes (e.g., the degree of expressiveness and easiness of describing a workflow using TOSCA (Ystia) or using CWL (StreamFlow)) led to the identification of well-defined roles for each component: • StreamFlow is the main workflow execution engine, and CWL format has been chosen for the workflow description. • WARP provides resource reservations; as such there will be a strict interaction between WARP and StreamFlow, and between WARP and the batch scheduler. • Fine control over reserved resources is achieved through HyperQueue. As such, StreamFlow will interface with HyperQueue to submit jobs. • ML/DL modeling and cloud operations are managed through FMLE and Ystia respectively. As FMLE provides features to ease the process of creating ML/DL models, we see the role of FMLE as the main tool for managing the training of ML/DL models as they will become available in the pilot applications. We also see FMLE as the main target for integrating a tool pipeline for the transformation of conventional DL models into Spiking Neural Network models that will fit with neuromorphic processing devices as developed in WP3. On the other hand, Ystia will support the management of cloud resources.

Lesson Learned 6

4.3 Pilot related LL

Author	LL title	LL brief description
AVIO (WP5)	**Ansys license issues**	The ANSYS Fluent solver used within the combustors task of WP5 is a commercial software. The licences used are proprietary and belong to the DIEF research group of the University of Florence. In order to be able to use these licences, calculation nodes of the HPC in use must be able to connect with the UNIFI licence server. This requires the opening of special ports to enable the connection.
Questions		**Answers**
Activities leading to this LL		Issue was identified during the preliminary activities for the optimisation of the multiphysics and multiscale tool U-THERM3D, main goal of the combustor part of the WP5.
Challenges or issues		Once the environment had been prepared for the use of proprietary licences, it was possible to exploit the computing resources of IT4Innovations.
Envisioned solutions		Solution obtained during the first phase of the project. ANSYS proprietary licences can be used correctly after the set of the UNIFI server ports.

Lesson Learned 7

Author	LL title	LL brief description
AVIO (WP5)	**Karolina performance analysis**	During a scalability test, errors occurred when using ANSYS Fluent code on more than 9 nodes of the IT4I Karolina cluster. In particular, the numerical test case cannot be opened and therefore the desired job is not performed correctly.
Questions		**Answers**
Activities leading to this LL		The issue was identified during the preliminary activities for the optimisation of the multiphysics and multiscale tool U-THERM3D, main goal of the combustor part of the WP5.
Challenges or issues		Errors occurred when using ANSYS Fluent code on more than 9 nodes of the IT4I Karolina cluster. After an investigation carried out by IT4I support, the problem was identified in a lack of memory/memory bandwidth but still needs to be studied in detail. Most likely cause is the NUMA architecture of Karolina CPUs which was not taken into account when tuning the application for the Karolina system.
Envisioned solutions		The performance evaluation of the Karolina cluster is currently under analysis. At the moment, the problem is solved by requesting more nodes than are actually used during the job, in this way a fixed number of computing resources is used but with greater availability in terms of memory. The current procedure has a beneficial effect on simulation speedup but actually requires more nodes than are actually used during the calculation. New solutions will be developed after the detailed analysis of the performance of the Karolina cluster. The idea is to perform the performance analysis with dedicated software and have WP3 also analyse them as the leader of the hardware architecture and tuning the application in respect to the NUMA architecture. Also this could be an interesting activity for profiling tools such as likwid, to determine the underlying cause of the different behavior.

Lesson Learned 8

Author	LL title	LL brief description
ECMWF (WP6)	**Large scale allocation of cluster system**	Timely execution of Numerical Weather Prediction is crucial in order to provide valuable model output data to users (such as forecasters, and civil protection). On the other side, to improve the accuracy and reliability of NWP, there is a rising request of large ensemble of high-resolution simulations (NP for regional downscaling purposes for smart farming necessities). This results in large allocations of HPC resources for a limited amount of time (usually ~1h), which is not ideal from the scheduling point of view.
Questions		**Answers**
Activities leading to this LL		ACROSS WP6 aims at demonstrating global-scale NWP at cloud-resolving resolution (5km or better), which requires in excess of 200 HPC nodes for each model ensemble. A full model ensemble (50 members) is thus requiring up to 10,000 HPC nodes.
Challenges or issues		Most HPC resources are configured as shared resources with best-effort scheduling policies. This approach is excellent for concurrent execution of a large number of rather small jobs (few nodes with respect to the total size of the cluster), while larger jobs (i.e., requesting almost the entire machine) might wait in queue for hours/days before execution. This is unfortunately not acceptable for time-critical activities like NWP.
Envisioned solutions		For the time being, in cooperation with the user support staff of the supercomputing centres, we are managing this requirement through reservations, that allow us to test the workflows developed in the context of ACROSS, and demonstrate the capabilities of the EuroHPC computational resources. We are looking forward to adopting, whenever possible, policies supporting “urgent computing”.

Lesson Learned 9

Author	LL title	LL brief description
NEUROPUBLIC (WP6)	**Benchmark / test case**	Some test cases have been selected for benchmarking. Those test cases concern weather forecast simulations with a spatial resolution up to 1km. The above forecasts are computationally very ambitious procedures. Hence, the aim is to produce high resolution forecasts (e.g. 1km) in a relatively small period of time (e.g. 84h forecast in less than 4h)
Questions		**Answers**
Activities leading to this LL		Benchmarking concerning MPI scalability has been made using various nodes of Galileo 100. The best solution is to use 48 nodes with 48 tasks per node (approx. 4h are needed for an 84h forecast).
Challenges or issues		A solution could be found, in order to reduce further the time needed for the 84h forecast (e.g. less than 4h) (e.g. asynchronous I/O capability).
Envisioned solutions		The asynchronous I/O capability that is built into the model (Weather Research and Forecasting/WRF) has been also turned on, through the use of “quilting” in order to speed up the writing of the models outputs. One node (among the 48) will be used only for this procedure.

Lesson Learned 10

Author	LL title	LL brief description
SINTEF (WP7)	Benchmark case size WP7	The concrete test cases used for benchmarking must reflect real expected usage, while at the same time pushing the boundaries in terms of size and complexity sufficiently to reveal issues that must be dealt with.
Questions		**Answers**
Activities leading to this LL		Benchmark testing for MPI scalability have been run using the Sleipner benchmark case (~2M cells) on 1-4 nodes on Karolina (up to 512 cores).
Challenges or issues		While this is a real and important case, it does not have the size required to reveal scaling issues beyond the tested number of nodes.
Envisioned solutions		Use refined Sleipner testcase (~18M cells) for future benchmark runs.

Lesson Learned 11

4.4 General LL

Author	LL title	LL brief description
IT4I (WP2)	Inner-project communication channels	What is the best communication channel within the project's consortium.
Questions		**Answers**
Activities leading to this LL		At the start of the project the consortium agreed to use as the main communication channel the chat function of Nextcloud platform that serves as the project's main document storage and a tool for a collaborative writing. The idea was mainly to minimize the number of mail messages being sent to/from a project's mailing groups.
Challenges or issues		Throughout the project it became apparent that the most people don't check or just forget to check the chat messages on Nextcloud platform and as the platform itself does not send the notifications to the users the most messages are read with a noticeable delay.
Envisioned solutions		Standard email communication has proven to be still the most reliable form of communication within the consortium. Nowadays most people are using dedicated email clients that notifies the users immediately upon receiving the new email message, are able to automatically organize the incoming messages to a predefined groups or enable to 'flag' the important messages for later.

Lesson Learned 12

Author	LL title	LL brief description
IT4I (WP2)	Harmonize technical terminology	Need to agree on the technical vocabulary within the project consortium to avoid the misunderstanding during the mutual discussions.
Questions		**Answers**
Activities leading to this LL		Mutual discussions in the early stages of the project regarding the ACROSS platform architecture and the orchestration subsystem.

Challenges or issues	In the scope of the co-design activities and related discussions there was a need to clearly explain or specify the meaning of certain technical terms in a specific context. As the ACROSS project is developing the orchestration subsystem that spans across a several layers e.g. high level job vs low level task orchestration thus the different people with different background and knowledge bases understand these terms differently in a given context.
Envisioned solutions	Create a inter-project dictionary for a technical vocabulary that will be agreed on by all partners.

Lesson Learned 13

5 Conclusions

The goal of deliverable D2.3 is to provide intermediate report on co-design activities, lessons learned acquired throughout the project so far and give the overview about current system documentation. This deliverable marks the end of milestone M3 where the first version of ACROSS platform is deployed and ready to use by the pilots and start of milestone M4 that focuses on the first pilot integration into a said platform.
First part of the deliverable describes the general introduction, position of the deliverable in the project, its relevance and reference to the other deliverables.
Second chapter of the deliverable deals with the overview of the devised ACROSS platform, explains its separation into a layered architecture and describes them in more detail. It provides an overview of the underlying HW infrastructures, SW stack that was deployed deployed in terms of the development environments to test the first pilot integrations and newly developed services, and contains a list of the computational resources assigned, mostly under the EuroHPC JU development access, to pilot WPs to deploy, benchmark, and update their use-cases on ACROSS platform and the available infrastructures.
The last chapter describes the Lessons Learned on technical development and integration. The acquired LL are divided into several categories based on the area of interest whether it concerns the whole project, technical development or just a specific pilot use-case.
In summary, all of the activities described in this deliverable are in line with the current phase of the project and the monitored activities and focusing on the next milestone which is the beta version of ACROSS platform and technologies.

References

There are no sources in the current document.