Large-scale EXecution for Industry & Society

Deliverable D2.1

Pilot needs & Infrastructure Evaluation Report

Co-funded by the Horizon 2020 Framework Programme of the European Union
Grant Agreement Number 825532
ICT-11-2018-2019 (IA - Innovation Action)

DELIVERABLE ID | TITLE	D2.1 | Pilots needs / Infrastructure Evaluation Report
RESPONSIBLE AUTHOR	Marc Levrier (Bull/Atos)
WORKPACKAGE ID | TITLE	WP2 | Requirements Definition and Architecture Design
WORKPACKAGE LEADER	Bull/Atos
DATE OF DELIVERY (CONTRACTUAL)	30/04/2019 (M04)
DATE OF DELIVERY (SUBMITTED)	30/04/2019 (M04)
VERSION | STATUS	V1.0 | Final
TYPE OF DELIVERABLE	R (Report)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	IT4I, LRZ, LINKS, O24, Avio Aero, CIMA, CEA, ECMWF, AWI
INTERNAL REVIEW	Stephan Hachinger (LRZ), Sean Murphy (CYC)

Project Coordinator: Dr. Jan Martinovič – IT4Innovations, VSB – Technical University of Ostrava
E-mail: jan.martinovic@vsb.cz, **Phone:** +420 597 329 598, **Web:** https://lexis-project.eu

DOCUMENT VERSION

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Creation	2019/02/27	Marc Levrier (Bull/Atos)
0.2	Updates according to Sean Murphy's review	2019/03/11	Marc Levrier (Bull/Atos) Sean Murphy (CYC)
0.3	Updates according to Sean Murphy's and Stefan Hachinger's reviews. Introduction and conclusion (new).	2019/03/12	Marc Levrier (Bull/Atos) Sean Murphy (CYC) Stefan Hachinger (LRZ)
0.4	Writing of section 4 (started).	2019/03/13	Marc Levrier (Bull/Atos)
0.5	Writing of section 4 and fixed stylesheet problems (from reviews using OpenOffice).	2019/03/14	Marc Levrier (Bull/Atos)
0.6	Updated use cases' workflow figures in section 2 and finished writing in illustrations in section 4.	2019/03/22	Marc Levrier (Bull/Atos), Donato Magarielli (AVIO), Antonio Parodi (CIMA), Emanuele Danovaro (CIMA), Thierry Goubier (CEA)
0.7	Updated list of WP7 application models in section 2, explanations about orchestration services in section 3 and EUDAT/iRODS, WCDA in section 4. This revision can be considered V1.0 candidate.	2019/03/26	Marc Levrier (Bull/Atos), James Hawkes (ECMWF)
0.7.1	Updated entire 2.12.1 section (from last WP5 team's update).	2019/03/28	Marc Levrier (Bull/Atos), Donato Magarielli (AVIO)
0.8	Included content updates from ECMWF, AVIO, CYC and CEA.	2019/04/05	Marc Levrier (Bull/Atos), Donato Magarielli (AVIO), James Hawkes (ECMWF), Sean Murphy (CYC), Thierry Goubier (CEA)
0.8.1	Updated content about TESEO gateway in section.	2019/04/05	James Hawkes (ECMWF)
0.8.2	Various fixes from main reviewer's final proof read.	2019/04/09	Sean Murphy (CYC)
0.9	Final fixes from project manager and codesign lead	2019/04/09	Olivier Terzo (LINKS), Jan Martinovic (IT4I)
0.9.1	Updated IT related figures from IT4I and LINKS	2019/04/10	Alberto Scionti (LINKS), Martin Golasowski (IT4I)
0.9.2	Updated IT related figures and additional minor fixes from IT4I, LINKS and AWI	2019/04/15	Alberto Scionti (LINKS), Martin Golasowski (IT4I) Natalja Rakowsky (AWI)
0.9.3	Updated sections 2.1, 2.1.1, 2.1.2 with details about Rotating parts case study and glossary	2019/04/24	Donato Magarielli (AVIO)
0.9.4	Updated all sections according to feedback from External Advisory Board (EAB)	2019/04/24	Marc Levrier (Bull/Atos)
1.0	Final release	2019/04/30	Marc Levrier (Bull/Atos)

GLOSSARY

AAI	Authentication & Authorization Infrastructure
AD	(Microsoft) Active Directory
CaaS	Container-as-a-Service
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
CH, core.h	Core.hour
CLI	Command Line Interface
DDI	Distributed Data Infrastructure
Gbps	Gigabit per second
GHz	Gigahertz
GPFS	Global Parallel File System (from IBM)
HPC	High Performance Computing
IAM	Identity and Access Management
IaaS	Infrastructure-as-a-Service
LDAP	Lightweight Directory Access Protocol
NFS	Network File System
NVME	Non-Volatile Memory Express (flash memory direct access from PCI Express)
PaaS	Platform-as-a-Service
RAM	Random Access Memory
SBB	Smart Burst Buffer
SSD	Solid State Drive
SME	Small & Medium Enterprises
WCDA	Weather and Climate Data API
WP	Work Package
YORC	Ystia ORChestrator

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The increasing quantities of data generated by modern industrial and business processes pose enormous challenges for organizations seeking to glean knowledge and understanding from the data. Combinations of HPC, Cloud and Big Data technologies are key to meeting the increasingly diverse needs of large and small organizations alike. Critically, access to powerful computing platforms for SMEs - which has been difficult due to both technical and financial reasons - may now be possible. LEXIS (Large-scale EXecution for Industry & Society) project will build an advanced engineering platform at the confluence of HPC, Cloud and Big Data which will leverage large-scale geographically-distributed resources from existing HPC infrastructure, employ Big Data analytics solutions and augment them with Cloud services. Driven by the requirements of the pilots, the LEXIS platform will build on best of breed data management solutions EUDAT[1] and advanced, distributed orchestration solutions, augmenting them with new, efficient hardware capabilities in the form of Data Nodes and federation, usage monitoring and accounting/billing supports to realize an innovative solution. The consortium will develop a demonstrator with a significant Open Source dimension including validation, test and documentation. It will be validated in the pilots - in the industrial and scientific sectors (Aeronautics, Earthquake and Tsunami, Weather and Climate) – where significant improvements in KPIs including job execution time and solution accuracy are anticipated. LEXIS will promote the solution to the HPC, Cloud and Big Data sectors maximizing impact through targeted and qualified communications. LEXIS brings together a consortium with the skills and experience to deliver a complex multi-faceted project, spanning a range of complex technologies across seven European countries, including large industry, flagship HPC centers, industrial and scientific compute pilot users, technology providers and SMEs. Federated Authentication and Authorization infrastructure across these HPC service and data providers will also be key.

Position of the deliverable in the whole project context

From a calendar standpoint, the present deliverable D2.1 is the first technical deliverable of the whole LEXIS project.

It is a product of WP2 (Requirements Definition and Architecture Design) in its Task 2.1 (Infrastructure Evaluation and Key Technology Identification for LEXIS).

This report document is to be delivered at the end of M04.

As shown in picture hereafter, WP2 needs to set the foundations for business cases (WP5 to 7) to benefit from advanced techniques brought in by WP3 (Data management using NVME Burst Buffers) and WP4 (Orchestration and cloud services). An access portal will also be a part of the LEXIS architecture to provide easy access from Industries, SMEs and Academia looking for high Performance computing and Data Management.

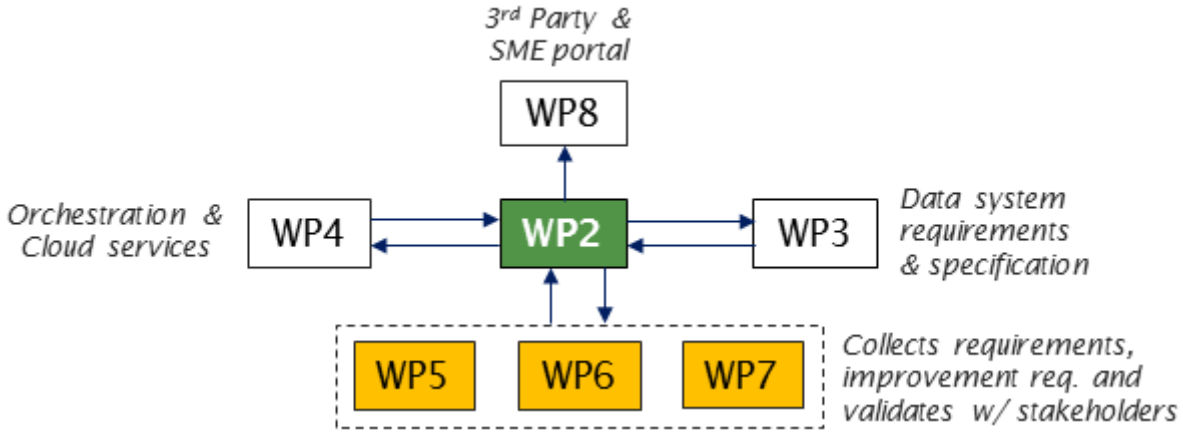

Figure 1: Position of WP2 in the LEXIS project

Contributors of this deliverable are:

- Bull/Atos as the work package leader (WP2),

[1] EUDAT - https://eudat.eu

- IT4I as one federated HPC service provider,
- LRZ as the other federated HPC service provider and work package leader (WP3),
- LINKS Foundation as the coordinator of codesign tasks and work package leader (WP4),
- OU24 for security related aspects,
- Avio Aero for the Aeronautics use case (WP5),
- CEA for the Tsunami Prediction use case (WP6),
- CIMA & ECMWF for the Weather Forecast use case (WP7),
- CYC for the portal related topics and work package leader (WP8).

Description of the deliverable:

The objectives of deliverable D2.1 are twofold:

Analysis of the pilot requirements: the 3 LEXIS pilot experiments (Aeronautics, Seismic/Tsunami prediction and Weather Forecast) are described in terms of problem type, software, workflow, currently used platforms, performance aspects and expectations. A detailed inventory of these characteristics is presented in Section 2.

Analysis of infrastructure available to LEXIS project: detailed analysis of the infrastructures of both LEXIS HPC service providers (IT4I and LRZ) with inventories of:

- What is already available and fits the pilots' requirements (Section 3),
- What is not yet available and required to reach the objectives, mostly user and system services required to make best possible use of the LEXIS hardware infrastructure (Section 4).

1 INTRODUCTION

About the present deliverable (D2.1)

WP2 (Requirements Definition and Architecture Design) deals with both application use cases requirements and overall system design of underlying infrastructures and services. Works started in early January 2019, kick-off meeting took place in Ostrava (Czech Republic) on the 15th of January, collaborative tools were setup a few days later and the first technical face-to-face and periodic web meetings we rapidly organized.

Our working method was basically to simultaneously:

- Collect technical information from HPC resource and data providers,
- Inventory other assets and skills,
- Learn from WP5, WP6 & WP7 about their application workflows and technical expectations in terms of platforms, services and performance,
- Start to inventory building blocks to be added to reach LEXIS's objectives (macroscopic view).

The present technical deliverable gathers and structures the contents collected during this initial project phase into three main parts:

- Analysis of the pilot requirements (software applications and workflows in Section 2),
- Analysis of infrastructure available to LEXIS (HPC and cloud systems in Section 3),
- First list of components or services to be added (in Section 4, to be detailed in deliverable D2.2 [1]).

2 PILOT REQUIREMENTS REPORT

This section provides detailed information about the three pilots of the LEXIS project.

They are respectively covered in WP5, WP6 and WP7 and technically analysed in the context of WP2, both in terms of basic hardware and software prerequisites, but also in terms of more advanced concepts such as compute and I/O acceleration, workload orchestration, remote visualization etc.

To make sure they will be able to execute in optimal conditions on the LEXIS federated infrastructure, an initial step has consisted in gathering all the technical aspects of these application workflows:

1. As they are today,
2. As they need to evolve and scale with LEXIS.

It is also important to note that we have started codesign sessions (Task 2.2) early in the project (in M02). The idea is to onboard representatives of all partner organizations right at the beginning of the technical talks, since the architecture definition is obviously on the critical path and is quite challenging.

The following subsections provide such information for each use case.

2.1 AERONAUTICS – AVIO AERO

Avio Aero pilot involves two main different aeronautics case studies, one regarding turbomachinery and the other one referring to rotating parts, both requiring computational-intensive and time-consuming c simulations but based on different application software.

Turbomachinery case study

Avio Aero has gathered significant experience in the past decades in turbomachinery design, development, testing and manufacturing. Among different product lines, unsteady fluid-dynamic simulations of an entire LP (low pressure) turbine (with multiple rows interacting each other) can represent one of the best-in-class use cases that can be selectable. This case study needs compute-intensive time resources. The goal is to carry a detailed performance assessment of the analysed configuration.

The simulation tackles moderately low, transitional Reynolds numbers flow regimes [6] [2] and faces critical phenomena like flow transition from laminar to turbulent occurring over the air foils, separation effects at blades' trailing edge, producing downstream strong mixing and complex wake structures impacting on neighbouring rows.

To limit the computing running time as much as possible, it will be important to enable the scalability of these URANS simulations (Unsteady Reynolds Averaged Navier-Stokes) to very large-scale HPC (thousand cores) so as to achieve a significant reduction in execution time and, at the same time, improve quality of the analyses, thanks to a dedicated quick post-processing of massive amounts (TB) of simulation output data.

Different milestones are envisaged to measure benefits arising from using the LEXIS solution. Initial baseline measurements will be taken at the start of the project and will be used to assess gains made from employing LEXIS technologies. Extra testing will be required in a second phase to validate and fine-tune HPC code optimizations.

The execution of mentioned Computational Fluid Dynamics (CFD) simulations, that Avio Aero will carry out in this case study and are framed in the discussed performance assessment, will rely on the application software "TRAF".

Turbomachinery computational approach

As essential preliminary remark, any CAE (Computer Aided Engineering) analysis includes the following three phases:

1. Pre-processing, defining the model, the set-up and environmental factors to be applied to it,
2. Analysis solver execution, usually performed on high powered computers,

[2] It is a good practice in engineering fluid-mechanics to use Reynolds number to classify different flow regimes. Three main types of flow regimes can be recognized: the laminar one (Low Reynolds number) typically characterized by low speeds with a very steepest boundary layer close to the wall, the turbulent one (high Reynolds number) characterized by flow vortices, eddies and wakes. In between these regimes, the transitional flow is laminar at the origin and then passing turbulent at a certain critical length. This structure can be characterized by high entropy contents, thus generating strong performance deficit if not well predicted and controlled.

3. Post-processing of results, using visualization tools.

General information about used CFD solver

TRAF, an application software developed by the University of Florence, is the main CAE solver adopted by Avio Aero in CFD simulations on Turbomachinery. This application is used worldwide, but Avio Aero uses a special, customized version.
Specifically developed to assist turbomachinery designers, the TRAF code solves the unsteady, three-dimensional, Reynolds-averaged Navier-Stokes equations in the finite volume formulation on multi-block structured grids. A high level of parallelization is achieved by means of a hybrid OpenMP/MPI code architecture. The computational framework was extensively validated against several turbomachinery configurations. It runs on only CPU-based HPC infrastructure, but development activities for porting the code to GPU-accelerated computational platforms are currently ongoing.

Operations on LEXIS infrastructural building blocks

The Figure 2 next page below illustrates the workflow of CAE analyses supporting the Turbomachinery case study that will be implemented on the following three LEXIS infrastructural building blocks:

- **Aeronautics CFD simulation gateway**

The Aeronautics CFD simulation gateway will host the application services that are needed to support the CFD analyses during each of the three above mentioned phases, including the setup of TRAF solver parameters and the sending/reading of the CFD model input file at *pre-processing stage*, the monitoring of computational convergence and stability of simulations during *analysis solver execution*, and data analytics and visualization of simulation results at the *post-processing stage*. Moreover, the Aeronautics CFD simulation gateway will manage the input files (whose magnitude order size is 10^1 GB) that feed the execution of simulations on the HPC CPU/Accelerators Infrastructure. Output files are expected to be 1 or 2 orders of magnitude larger.

- **HPC CPU/Accelerators Infrastructure**

It will host the computational services supporting the batch iterations of TRAF execution, that includes a user-defined number of computational runs or periods. For this computational phase, all input files need to be uploaded and stored on a file system accessible from within the HPC cluster. At the end of each application run period, the output files representing the partial results of simulations will become the input files for next run period. Finally, once completed the last run period, the results of simulations are produced.

- **Data System Infrastructure**

It will host the storage of partial and final output files, whose order of magnitude in size is 10^2 GB. More specifically, each run period produces ~300 GB results at the end of its execution, including one ~160 GB file, two ~100 GB files, and one ~30 GB folder. The writing phase happens at each completed run period, that currently takes ~10 days depending on the complexity of the selected Turbomachinery input model and the current state-of-the-art CPU time. However, for backup and recovery purposes, a temporary writing phase based on a user-defined frequency will also occur on Data System Infrastructure to store an aerodynamics partial solution allowing to restart the simulation if any interruption. The average size of the scratch/temporary (single) file is 160 GB. It is produced according to a user-defined writing frequency and collects information that are spread over a set of temporary files that are overwritten from time to time, while big scratch files are always kept during TRAF execution. Temporary file size varies from 1 KB to 50 KB and their number up to 1000 (this number can change depending on input model).

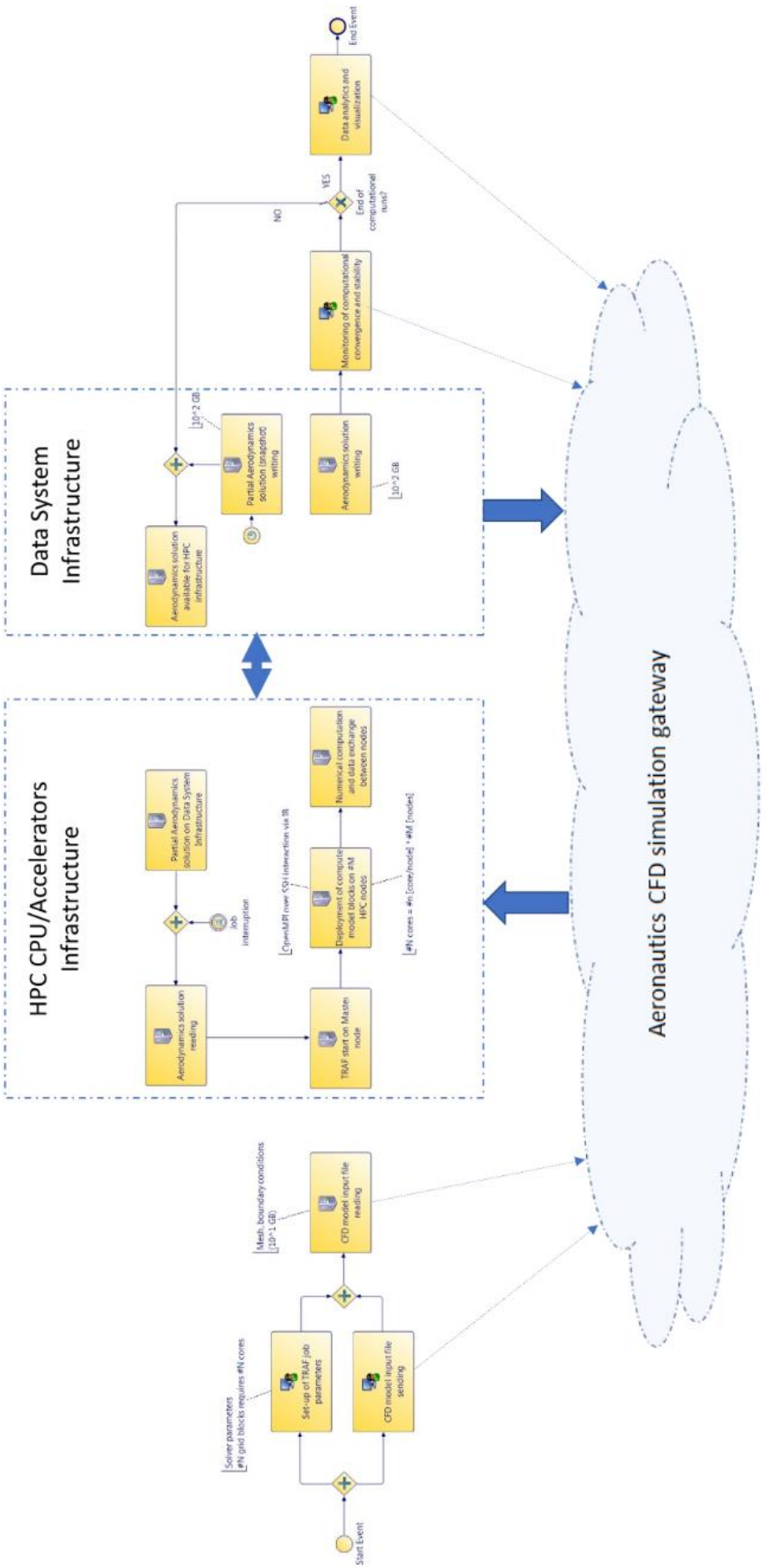

Figure 2: Turbomachinery CAE analyses on LEXIS infrastructural building blocks (source: AVIO)

Rotating Parts case study

Beyond standard CFD simulated products, today challenges are arising when studying complex flow fields in mechanical parts that rotate in the presence of air and lubricating oil. Today this kind of simulation is at the leading edge of numerical technology.

This problem arises in multiple scenarios, a good example of which is in the modelling and design of large gear box systems.

The need to design gearboxes capable to withstand high transmission efficiency, poses the challenge to predict and simulate the flow field operating inside with greater precision. Differently from most of the automotive applications, Aeronautics products are usually cooled and lubricated by oil jets instead of splash lubrication. The combination of jet lubrication and high tangential speeds precludes the possibility to neglect interactions between the liquid and gaseous phase. This represents the main challenge from a numerical analysis point of view.

Simulations of these phenomena require a large amount of compute resources and typically take considerable time; further, they are often part of a larger workflow which involves revisions to the design based on the output of the simulations. This workflow ultimately must converge to a solution which meets the design requirements. An integration with advanced HW solutions is envisaged to support and speed up the analysis process.

Avio Aero team will work to set up this new CFD methodology applied to gearboxes engineering, to deploy and test the related numerical solver on LEXIS infrastructure, and test and validate the benefits of this new engineering analysis approach.

The execution of mentioned CFD simulations, that Avio Aero will carry out in this case study, will rely on the application software “Altair nanoFluidX™”.

Rotating Parts computational approach

- **General information about used CFD solver**

Altair nanoFluidX™ is a particle-based fluid dynamics simulation tool to predict the flow in complex geometries with complex motion. This tool is based on a weakly-compressible Smooth Particle Hydrodynamics (SPH) formulation and contains several exclusive features which improve accuracy and make the code a unique particle-based solution on the market. The software is created and optimized for use on clusters of Graphical Processing Units (GPUs), making it extremely fast. It can be used to predict, for example, the oiling in powertrain systems with rotating shafts/gears and analyze forces and torques on individual components of the system or predict the sloshing in tanks with transient motions. For such typical gear-train applications, the code can run an order of magnitude faster than a Finite-Volume code while also including less geometry simplifications.

- **Operations on LEXIS infrastructural building blocks**

The Figure 3 next page below illustrates the workflow of CAE analyses supporting the Rotating parts case study that will be implemented on the following three LEXIS infrastructural building blocks:

- Aeronautics CFD simulation gateway
  The Aeronautics CFD simulation gateway will host the application services that are needed to support the CFD analyses during each of the three above mentioned phases, including the *pre-processing* in Altair SimLab™ application software for particle generation, the setup of nanoFluidX™ solver parameters (i.e. the user specifies the physical time he wants to simulate) and the sending/reading of the CFD input data at *pre-processing stage*, the assessment of computational results and the interaction with them during *analysis solver execution*, and data analytics and visualization of simulation results at the *post-processing stage*. Moreover, the Aeronautics CFD simulation gateway will manage the input files (whose order of magnitude in size is 10^1 GB) that feed the execution of simulations on the HPC CPU/Accelerators Infrastructure. Output files are expected to be 2 orders of magnitude larger (10^2 GB).

- HPC CPU/Accelerators Infrastructure
  It will host the computational services supporting the execution of nanoFluidX™. For this computational phase, all input files need to be uploaded and stored on a file system accessible from within the HPC cluster.
  Based on the user-defined parameters (such as physical time of simulation) and in accordance with required numerical criteria, the nanoFluidX™ code computes the time step so that the number of compute steps is determined from “physical_time/time_step”.
  Within the specified physical time, a set of output files representing the results of simulations is produced during the computation according to a user-defined frequency.
  For example, under the assumption of physical time = 1.5 s, a set of 150 output files is produced, meaning that 1 output has been generated every 0.01 s of physical time.
- Data System Infrastructure
  It will host the storage of restart files and final output ones, whose size is 10^2 GB. More specifically, the application job execution produces a set of ~400 GB of uncompressed data per compute job, including also all the restart files, that are relatively minor in size compared with the bulk output and are written for recovery purposes according to a user-defined frequency.

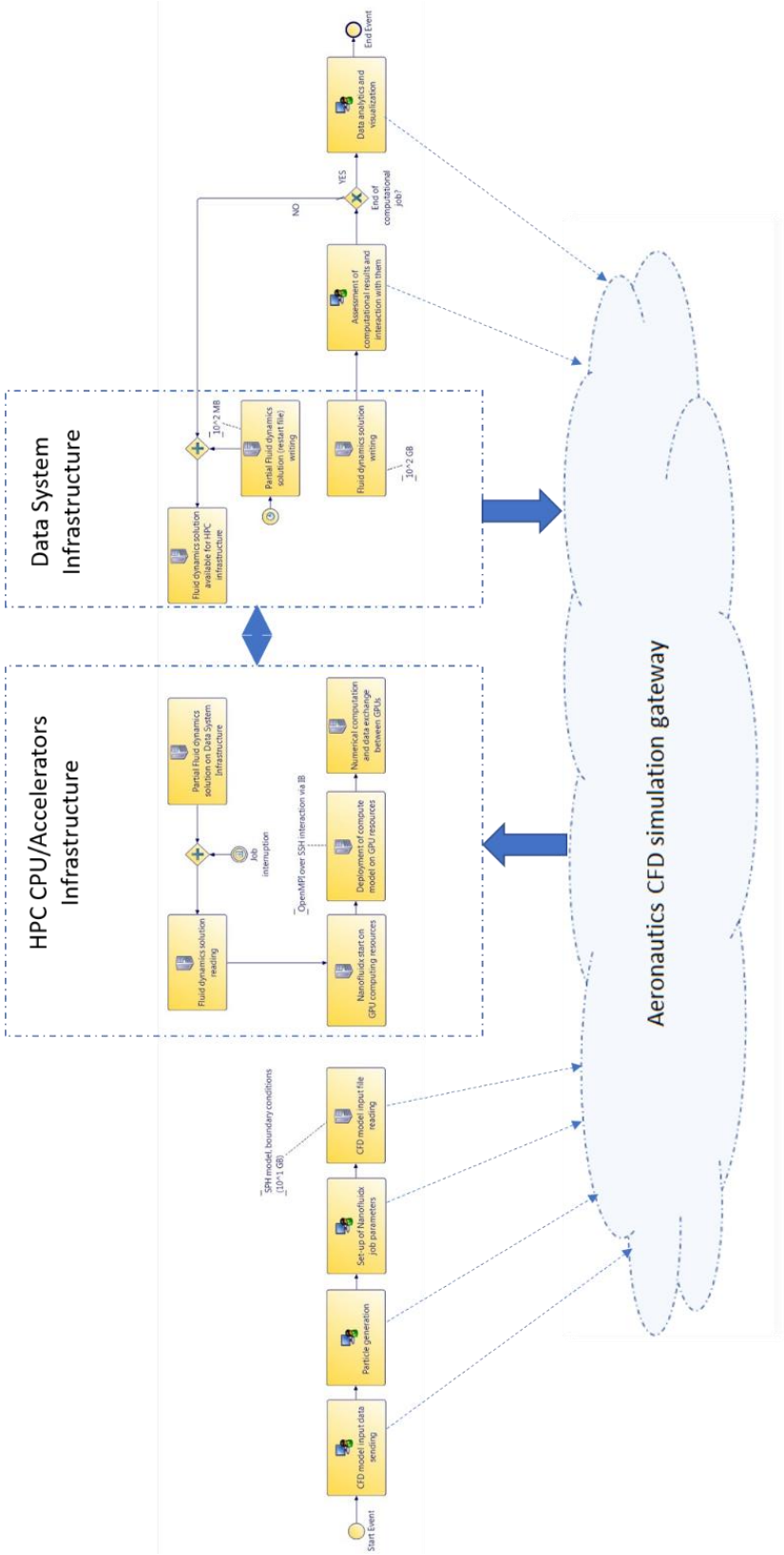

Figure 3: Rotating parts CAE analyses on LEXIS infrastructural building blocks (source: AVIO)

2.1.1 Hardware requirements

The hardware requirements needed to implement the two Aeronautic use cases will be hereafter described.

These use cases deal with CFD workflows (Computational Fluid Dynamics) which involve both batch computing (solvers implemented as parallel software applications running in background on many cores) and interactive 2D/3D sessions for pre- and post-processing of computing data.

Turbomachinery HW requirements

For Turbomachinery, the parallel solver adopted during batch compute phase makes use of MPI libraries which require Infiniband interconnect to ensure good scalability up to 2400 cores, while interactive jobs for simulation results post-processing require nodes provided with accelerated graphics and ideally remote visualization services to seamlessly offer the same performance as a high-end 3D-capable local workstation. Specifically, Table 1 and Table 2 below list the needed TRAF requirements for compute and visualization phases:

Compute phase

	Turbomachinery	
Configuration	**Current**	**Planned**
#CPU cores	800	To be defined
Clock freq. (GHz)	2.5	
#Nodes	34 (*)	
RAM (GB/core)	4.0	
Bandwidth (Gbps)	56 (FDR)	
(*) estimation based on HW features of IT4I Salomon Cluster compute nodes		

Table 1: Aeronautics hardware requirements (source: AVIO)

Visualization phase

	Turbomachinery	
Configuration	**Current**	**Planned**
#CPU cores	16	To be defined
Clock freq. (GHz)	2.3	
#Nodes	1 (*)	
RAM (GB/core)	24 (384 GB total)	
Bandwidth (Gbps)	56 (FDR)	
(*) estimation based on HW features of IT4I Salomon visualization server		

Table 2: Aeronautics Turbomachinery hardware requirements (visualization phase)

Rotating Parts HW requirements

For Rotating Parts, the parallel solver adopted in the compute phase has been created and optimized for use on clusters of Graphical Processing Units (GPUs), requiring CUDA and OpenMPI libraries shipped with the binary, and standard PCIe bandwidth to ensure good scalability, while jobs for data pre-processing and simulation results post-processing require nodes provided with accelerated graphics and ideally remote visualization services to seamlessly offer the same performance as a high-end 3D-capable local workstation. Specifically, Table 3 and Table 4 below list the needed requirements for pre-processing, compute and visualization phases:

Pre-processing phase

	Rotating parts	
Configuration	**Current**	**Planned**
#CPU cores	2	To be defined
Clock freq. (GHz)	2.0	
#Nodes	1	
RAM (GB/core)	4.0	
GPU	Nvidia with minimum 512 MB, 3D acceleration enabled	
Bandwidth (Gbps)	N/A	

Table 3: Rotating parts hardware requirements for pre-processing phase (source: AVIO)

Compute phase

	Rotating parts	
Configuration	**Current**	**Planned**
#CPU cores	48	To be defined
Clock freq. (GHz)	2.7	
#Nodes	1 (*)	
RAM (GB/core)	128 GB total	
GPU	8 Nvidia Tesla V100 (PCIe or SXM2 connection)	
Bandwidth (Gbps)	standard PCIe bandwidth is sufficient (32 Gb/s)	
(*) estimation based on HW features of IT4I DGX-2 Cluster		

Table 4: Rotating parts hardware requirements for compute phase (source: AVIO)

Visualization phase

	Rotating parts	
Configuration	**Current**	**Planned**
#CPU cores	16	To be defined
Clock freq. (GHz)	2.3	
#Nodes	1 (*)	
RAM (GB/core)	24 (384 GB total)	
Bandwidth (Gbps)	56 (FDR)	
(*) estimation based on HW features of IT4I Salomon visualization server		

Table 5: Rotating parts hardware requirements for visualization phase (source: AVIO)

2.1.2 System prerequisites

- Pre-processing system based on SimLab™ application
- Batch system (background execution) for TRAF and nanoFluidX™ solvers
- Interactive mode for results post-processing and interaction with them based on Paraview [7]
- HW acceleration: use of GPU
- Linux (64 bit):
  - For nanoFluidX™ solver RHEL 7/CentOS 7 or Ubuntu 18.04 are recommended
  - Altair SimLab™ has been tested on RedHat 5.9, 6.4 and on SLEL 11 SP2, 12 SP3
- Required specific run-time libraries for TRAF:
  - Intel Fortran libraries (ver. >=11.1.075)
  - OpenMPI (ver. >=1.4.2) ones
- Required specific run-time libraries for nanoFluidX™:
  - NVIDIA CUDA 8.0 and OpenMPI 1.10.2 – shipped with the binary.
- Mesa OpenGL libraries are required to support Altair SimLab™ execution
- OpenMPI interaction over SSH between the HPC nodes to support TRAF execution
- OpenMPI interaction over SSH between the GPUs to support nanoFluidX™ execution
- GPU-accelerated HPC resources to support the application execution and reduce the application running time.

2.2 TSUNAMI AND EARTHQUAKE PILOT – CEA

In LEXIS, the tsunami-and-earthquake pilot is a first time, entirely new processing chain, build out of components belonging to various partners. So, a significant amount of work is going into defining precisely how those components will integrate, and how this will be deployed and enabled by the LEXIS technologies. At this stage, some of that information is tentative and may evolve during the project duration, as the partners experience and knowledge about the flow improves.

The current workflow is described in a simplified view in Figure 4. It is a time-constrained, event triggered, always-on workflow, orchestrated by a specific higher-level orchestrator. In a starting phase, it does not represent a very high compute load, but we expect the LEXIS infrastructure to allow for the exploration of heavier flows with better results within the time constraints.

This test bed involves 4 main applications or workflows:

- AWI TsunAWI (Unstructured Mesh Finite Element Model for the Computation of Tsunami Scenarios with Inundation),
- GFZ GIS and Loss calculation software,
- Ithaca Emergency mapping software,
- CEA HeScade Workflow manager

The overall workflow is running permanently for a part and is triggered by external events for other parts. The GFZ GIS is running all the time to update itself on every OpenStreetMap update (once every minute). Upon an earthquake event, shake maps are generated and trigger a loss calculation on the relevant area extracted from GFZ GIS, and a fast TsunAWI simulation is launched. An example of such relevant areas could be a focus of some Fast Tsunawi simulations in restricted geographical areas of most interests either in term of population (e.g. major impacted cities and towns) or of industrial relevance (e.g. Fukushima Dai-Ichi nuclear power station). This would give authorities and emergency squads relevant data and information to base their decision of intervention on. The loss calculation remains active from this moment on, to update itself on new data or updated results. A second event will provide the earthquake moment and trigger a slow, more precise TsunAWI simulation. Results from all those (TsunAWI simulations, shake maps, and loss calculations) will trigger the computation of an area of interest result, which will be used by the GIS processing step, including some remote sensing products requests. Finally, a map post-processing step will be executed on a desktop machine by an operator, to produce emergency mapping products. Along the way, alerts and estimates will be emitted for use by the relevant emergency and disaster relief stakeholders. Overall, the high-level workflow orchestrator will ensure scheduling of the different parts under the specified time constraints. This workflow will likely have to be split into several more basic, domain-focused sub-workflows to make it simpler to integrate in the YORC orchestrator.

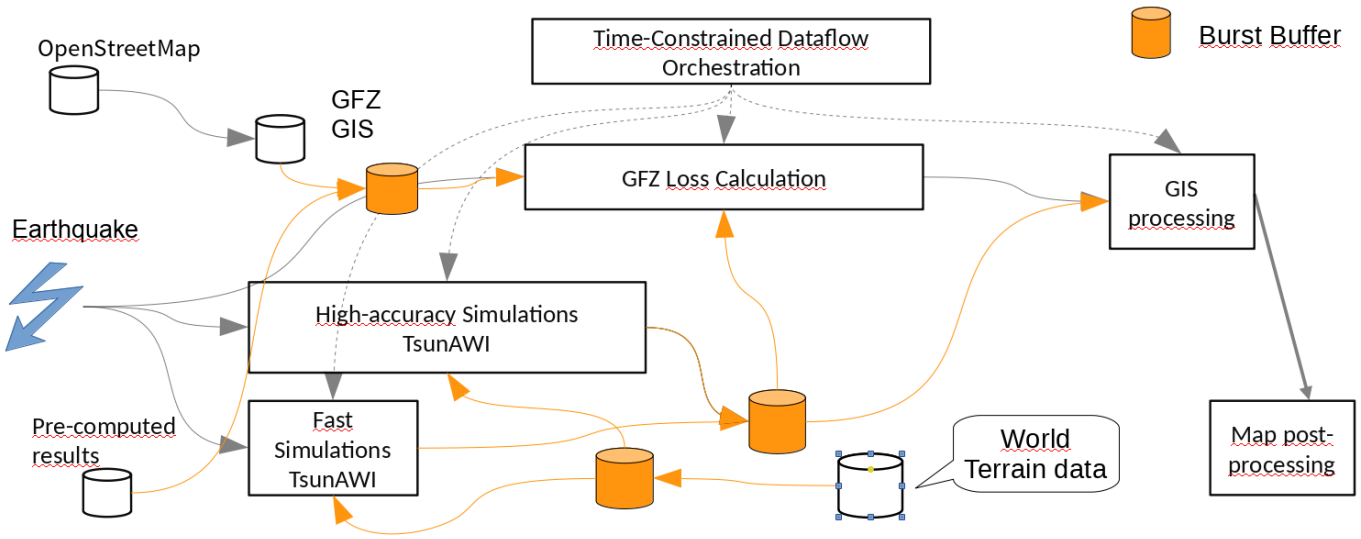

Figure 4: Tsunami & earthquake test bed infrastructure & data flow (source: CEA)

2.2.1 Hardware requirements

As currently under definition by the WP6 partners and lead (CEA), this complex workflow does not involve large parallel compute jobs. However, we're envisioning the possibility of running multiple simulations to get better early results, which would very significantly increase the computational requirements.

This also points out to the impact of I/O and burst buffers (displayed as orange cylinders on above picture) solutions to resolve its current performance issues with the GFZ GIS, and for allowing the test-bed to scale. We foresee then a significant role for burst buffers to hold the GIS data (so tied with the GFZ GIS update flow) and holding local datasets when running the TsunAWI simulations; as well holding area datasets for the loss calculations since those datasets may be updated multiple times during the processing of an event. The largest part is the mesh to read in. The output can also be written to the burst buffer for faster IO, and then copied elsewhere if necessary.

For now, TsunAWI writes Surfer grid raster data, because it is simple and does not need wrappers to *geotiff* or whatsoever. Netcdf is the default, but post processing this raw data is time consuming. It will likely be better to directly write the required data in the required format (to be assessed).

	TsunAWI		Ithaca		CEA WM		GFZ	
Configuration	**Current**	**Planned**	**Current**	**Planned**	**Current**	**Planned**	**Current**	**Planned**
#CPU cores	18	To be defined	18	To be defined	4	To be defined	64	To be defined
Clock freq. (GHz)	2.0		2.0		2.0		2.0	
#Nodes	1		1		1		1	
RAM (GB/core)	~1.8		0.9		4.0		2.0	
Bandwidth (Gbps)								

Table 6: Tsunami predictions hardware requirements (source: CEA)

2.2.2 System prerequisites

TsunAWI:

- HW acceleration: not required
- Advantage on accessing local SSD(s): likely a good candidate to use burst-buffers
- Compute + visualization (Matlab)
- Linux 64bits
- Intel Fortran
- OpenMP
- Not containerized

Ithaca (two parts):

- HW acceleration: use of GPU
  - Windows 64bits
  - Interactive within a GIS application
  - Runtime library: OpenGL (2.0) + Shader model 3.0
  - ArcGIS Desktop
- Without HW acceleration
  - Linux 64bits, batch
  - Software: QGIS + Python
  - Not containerized

GFZ GIS:

- Linux 64 bits
- Software and Runtime libraries: Python + PostgreSQL + GIS extensions to Postgis + osm2pgsql
- No HW acceleration
- Large amount of local, fast I/O resources (10+ TB)

CEA Workflow manager:

- Linux 64 bits
- Python / Java and Ystia / Yorc
- No HW acceleration
- Low latency access to resources

2.3 CLIMATE & WEATHER FORECAST – CIMA, ECMWF

The Weather and Climate Pilot will tackle the prediction of water-food-energy nexus phenomena and related socioeconomic impacts through the execution of complex, multi-leveled model stacks, including:

1. Global weather & climate models
2. Regional weather models
3. Domain-specific application models (e.g. hydrological, drought, wildfire)
4. Impact models providing information for key decision/policy makers.

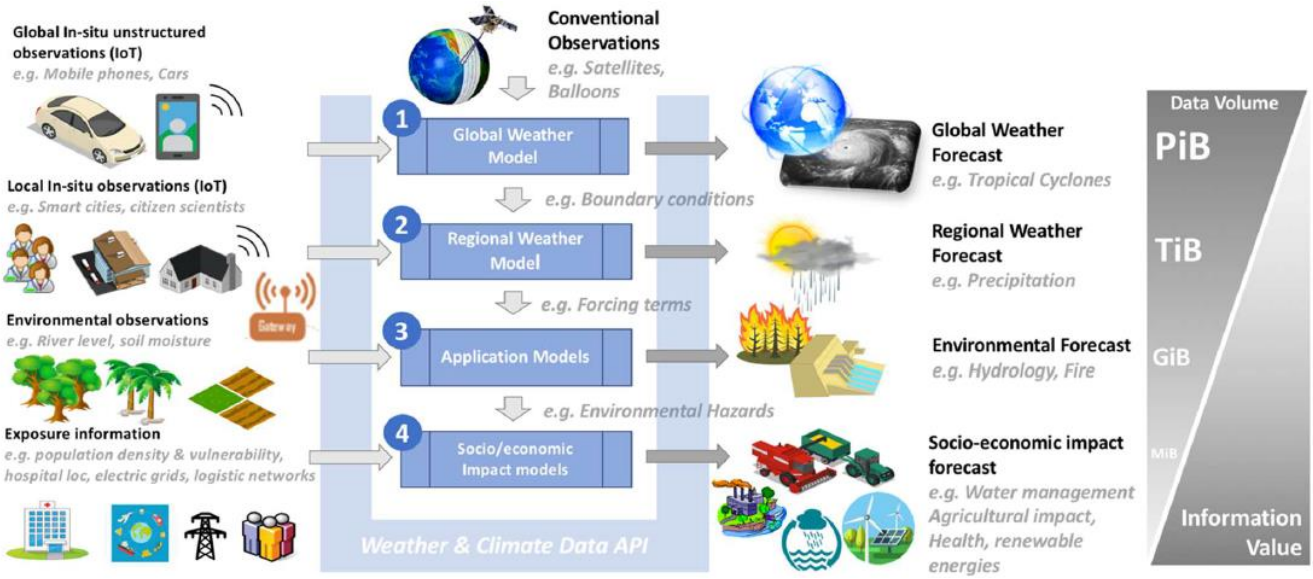

Figure 5: Weather forecast test bed infrastructure & data flow (source: CIMA)

This pilot will contain a test-bed which will develop innovative weather and climate predictions in two key aspects:

- Each model layer assimilates diverse types of weather and environment observation. Current operational systems are well suited to assimilate traditional observations (e.g. satellite, radio-probes, balloons…) but are not ready to handle observations as provided by emerging new technologies (e.g. IoT sensors, cell phones) and Edge-Computing data sources (e.g. self-driving cars, smart-city systems). The latter require new data analysis algorithms that deal with their unstructured nature and reliability (such as filtering mobile phones that are indoors). Moreover, data from many of these sources will often reside on heterogeneous Cloud systems, posing a data-collection problem.
- Each model layer produces its own model output to supply subsequent layers, as well as user-ready products (e.g. atmospheric temperature, precipitation rate or river flow). Until now, most of these models have run in dedicated computer clusters or HPC facilities. We will extend these models to interchange their data output and products with Cloud and HPDA environments. Given that the daily data volumes range up to hundreds of Terabytes (global ensemble forecasts), it remains an unsolved challenge how to efficiently transfer these massive data sets from the HPC to the Cloud systems, such that customers at the end of the data chain (e.g. SME's and small research centres) can build higher value products by post-processing and design business models around these data assets.

To maximize the interoperability of the 4 model layers, both observational data and model output data will be accessed with a unified Weather and Climate Data API (WCDA), also serving as data-transfer API between the model layers. This abstraction provides the interoperability paramount to the success of this use case, as third-party users will describe their data requirements directly in scientific terms without technical overhead. With similar abstraction, the Meteorological Archival and Retrieval System (MARS) by ECMWF handles petabytes of observations and model output. To maximize the success and the impact of the test-bed, we will leverage MARS Technologies as a starting point to quickly deliver the WCDA.

This test bed involves 5 main applications in a 3-step workflow. Between each step, some sort of data management and processing is involved:

Step #1: IFS [8] (Global Weather model)

Step #2: WRF [9] (Regional Weather model)

Step #3 (application models):

a. CONTINUUM (Hydrology Simulation)
b. RISICO (Wildland Fire Risk Simulation)
c. LIMAGRAIN (Agricultural Impact)
d. ERDS (extreme rainfall detection system - ITHACA)
e. An additional one on renewable energy production is being considered

Need for HPC resources and data volumes are indicated in the following diagram (only 3 application models are represented):

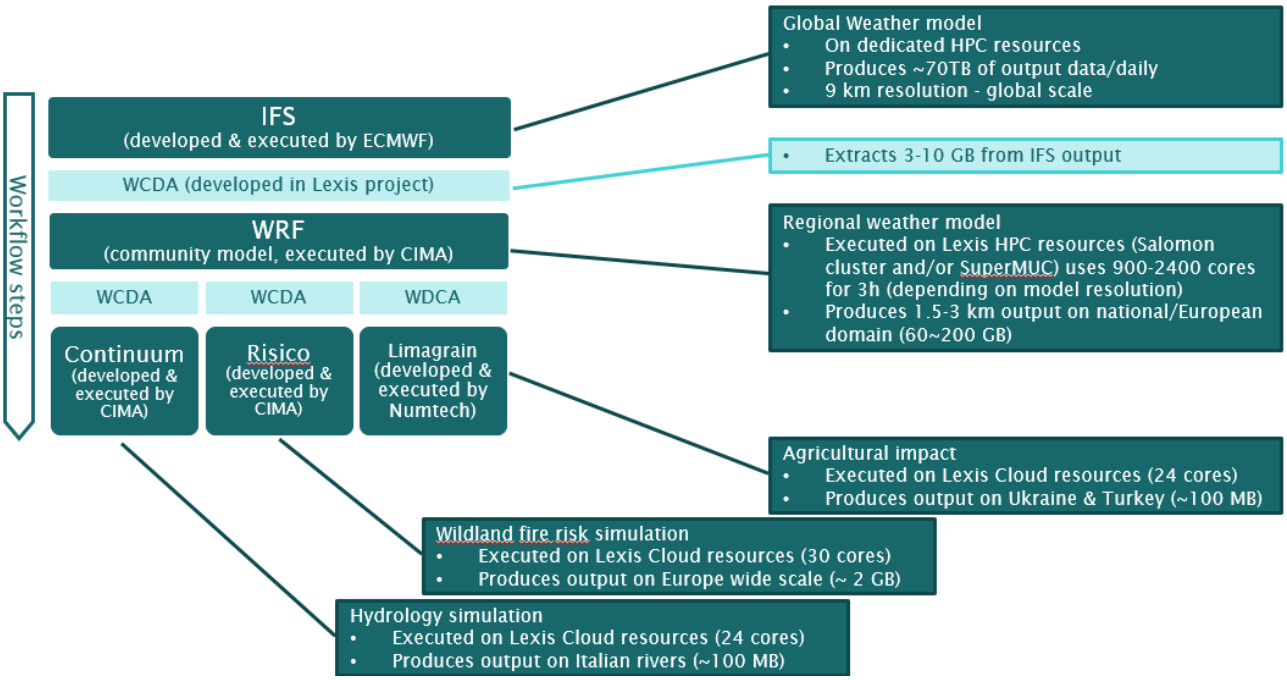

Figure 6: Weather forecast software workflow, data volumes and use of HPC resources (source: CIMA)

2.3.1 Hardware requirements

WRF, Continuum or ERDS are parallel (MPI) software that have been designed for intensive use of HPC clusters and can scale over thousands of processors. The air quality software is designed to run on a SMP (shared memory) Windows platform. Application models that come with specific requirements are shown in the table below.

	WRF		Continuum		Air Quality Forecast		Extreme Rainfall Detection System	
Configuration	**Current**	**Planned**	**Current**	**Planned**	**Current**	**Planned**	**Current**	**Planned**
#CPU cores	250	To be defined	1	To be defined	2 Xeon E3-1220L V3	To be defined	4	To be defined
Clock freq. (GHz)	2.0		2.0		2.0		3.2	
#Nodes	-		-		1		1(?)	
RAM (GB/core)	1.5		1.5		1.0		0.5	
Bandwidth (Gbps)	40		40		-		0.2	

Table 7: Climate & Weather hardware requirements (source: CIMA)

2.3.2 System prerequisites

WRF / Mesoscale weather simulations:

- HW acceleration: not required
- Linux 64bits
- Intel Fortran + MPI
- Batch execution
- HDF5, NetCFD, libpng, libjasper

Continuum:

- HW acceleration: not required
- Linux 64bits
- Intel Fortran + MPI
- Batch execution
- NetCFD

Air quality forecast:

- HW acceleration: not required
- Advantage on accessing local SSD. Candidate for SBB
- Windows 64bits
- Batch execution
- Not containerized
- NUMTECH licensed software
- Walltime metrics: <1h (industrial case), <3h (urban case)

Extreme rainfall detection system:

- HW acceleration: not required
- Linux 64bits (Ubuntu 18.04 LTS)
- Intel Fortran + MPI
- Python3, H5Py, GDAL
- Batch execution
- Supports Docker containerization
- Metrics to optimize: current execution 5-30 min

The test bed will also include a gateway for collection, filtering and transmission of in-situ observations (developed by LEXIS partner TESEO).

2.4 GLOBAL VIEW OF PILOTS' REQUIREMENTS

The following histograms are intended to provide the reader with single view of the main HPC hardware resource requirements for all the pilot software at once:

- Number of cores
- Amount of RAM per core
- Interconnect bandwidth

This will impact the infrastructure on which these applications will run (HPC versus cloud parts) as well as the sizing of the various kinds of compute and visualization nodes.

2.4.1 Infrastructure (number of cores)

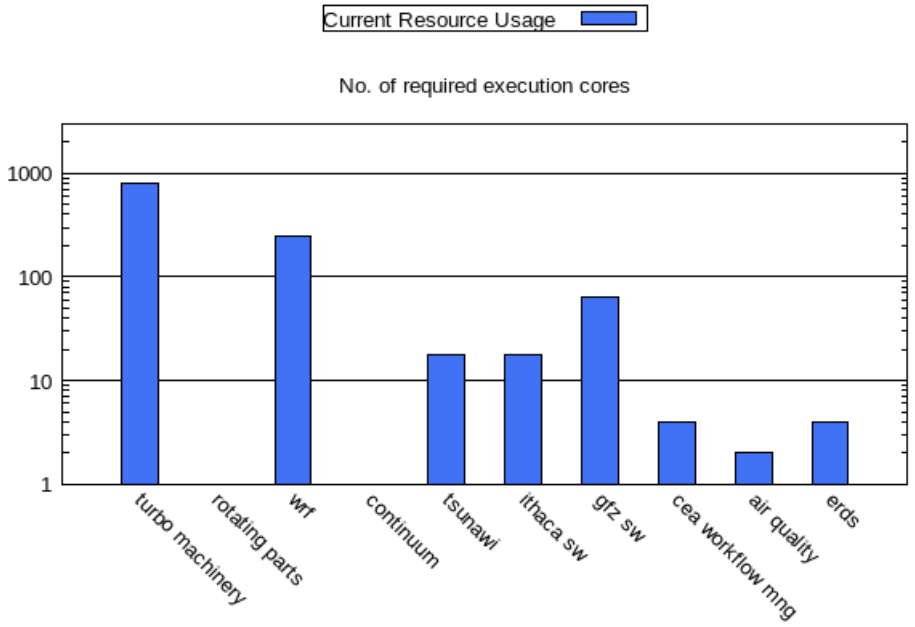

Figure 7: Number of cores required by each software

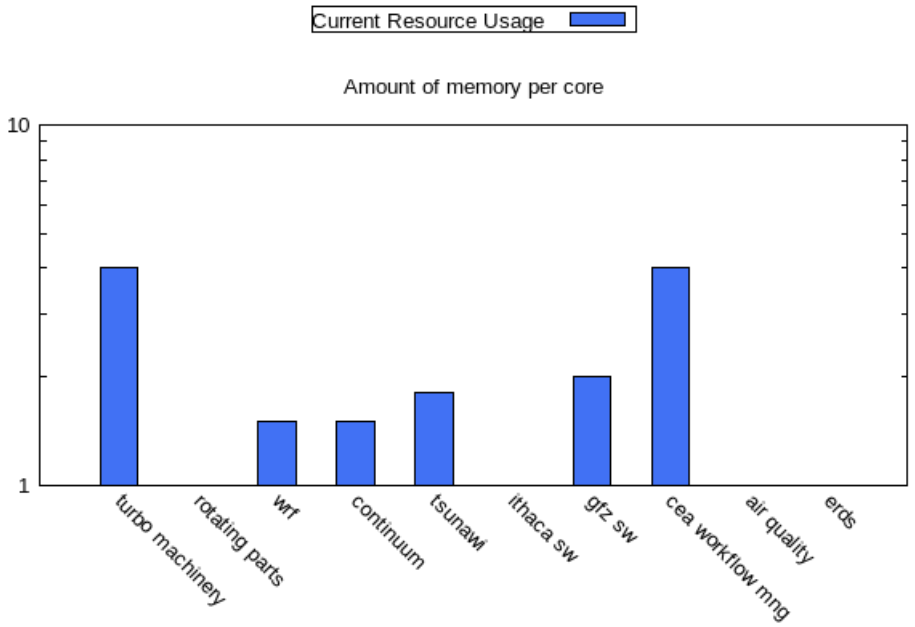

Figure 8: Amount of RAM per core required by each software (in GB)

2.4.2 Interconnect bandwidth

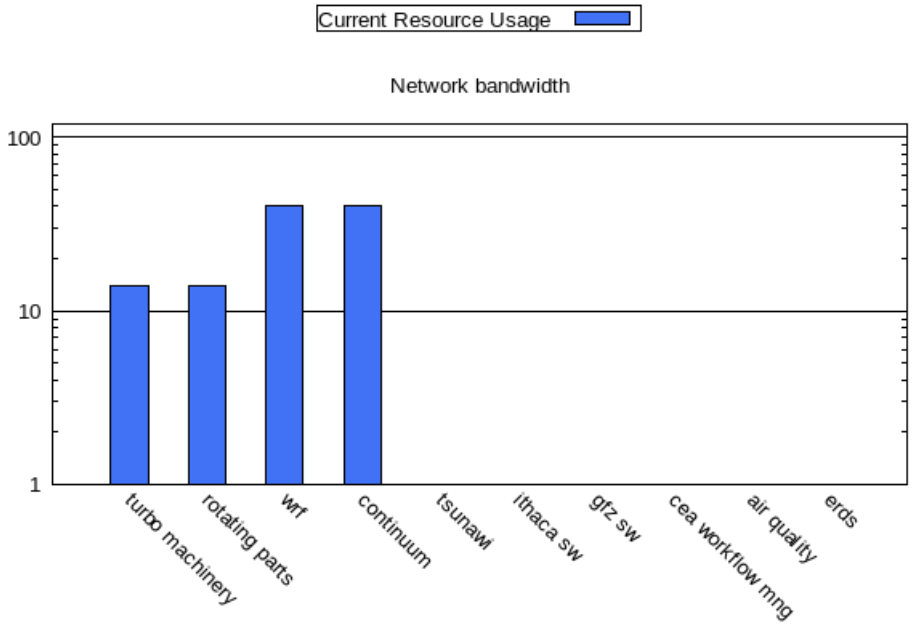

Figure 9: Interconnect bandwidth required by each software (in Gbps)

3 INFRASTRUCTURE EVALUATION REPORT

The LEXIS solution currently being designed consists of federating 2 HPC compute resource providers (IT4I in Czech Republic and LRZ in Germany) and 1 very large data provider (ECMWF in UK, specialized in Climate and Weather forecasts).

This section presents the infrastructure as well as the data and security services provided by these 3 organizations, as they are today. Additions, extensions or changes needing to be made to these within the context of LEXIS are briefly presented in this document. They will be described and technically studied in detail in the next deliverable D2.2 [1] of the present task T2.1.

3.1 IT4I (COMPUTING RESOURCE PROVIDER)

3.1.1 Overview of HPC and Cloud computing power

The following figure gives an overview of the HPC clusters available at IT4I as well as the planned Cloud infrastructure, together with parallel file systems available on each HPC cluster (Lustre) as well as in the cloud part (Ceph object storage).

Users on the HPC side connect using their IT4I credentials, all administrated by IT4I staff. Alternatively, the Cloud part of the IT4I infrastructure will be accessed using LEXIS credentials (identity federation across the LEXIS project):

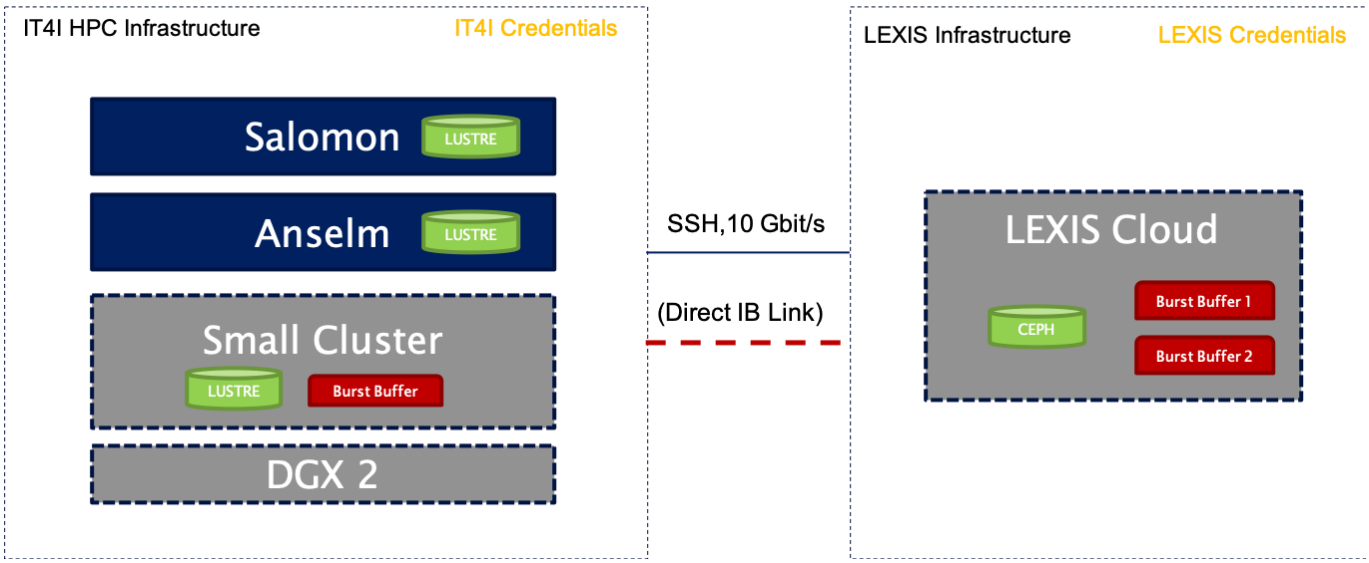

Figure 10: IT4I infrastructure overview (source: IT4I)

The cloud infrastructure that is planned for LEXIS is still under evaluation. IT4I indicates that the following is a good approximation of what they will be able to provide:

- Cloud computing nodes (6-8),
- 2 Burst Buffer server nodes,
- Fully redundant 100 Gbps Ethernet connections to the HPC part of the infrastructure,
- Fully redundant 10 Gbps Ethernet connections to NAS (file) storage systems,
- Current available WAN bandwidth is 4 x 10 Gbps, with planned upgrade to 100 Gbps,
- A view of such a cloud environment is presented in Figure 11.

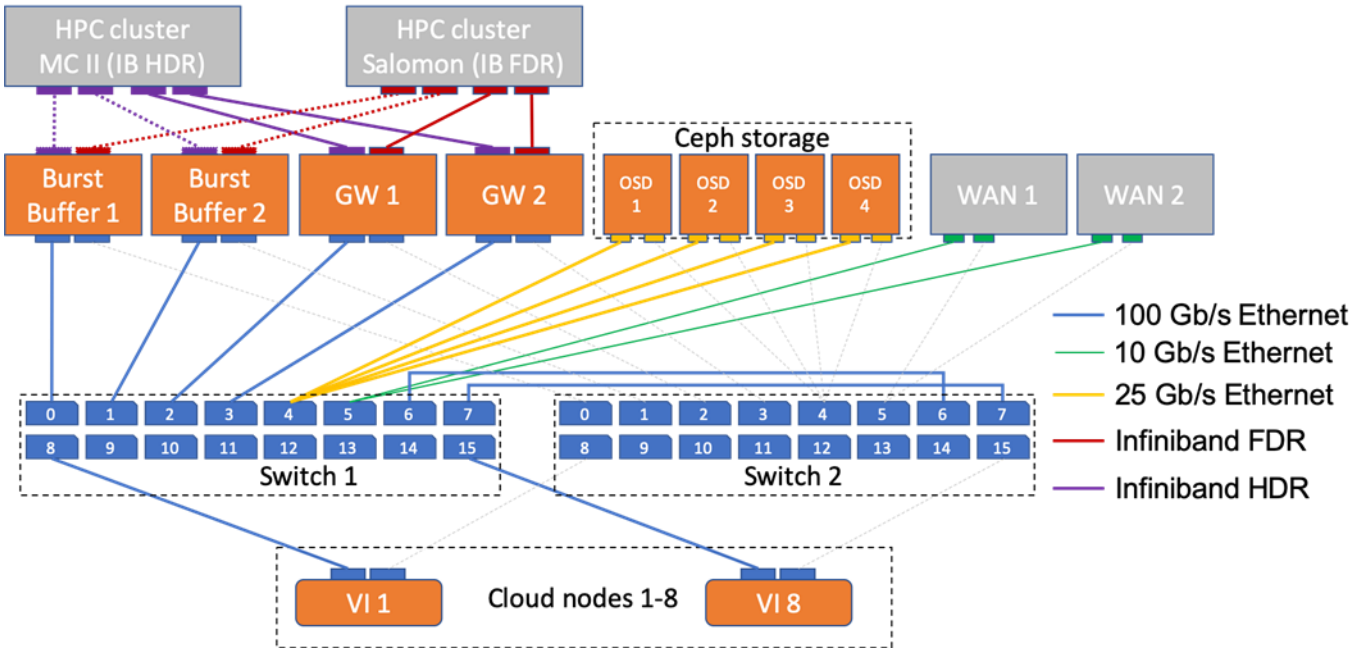

Figure 11: IT4I cloud infrastructure (proposal. Source: IT4I)

	Salomon cluster	Anselm cluster	Small cluster II	DGX2	Cloud (planned)
RPeak PFlops	2.00	0.09	0.84	0,13	N/A
#Nodes	1008	209	198	1	8-10 + 2 SBB
Node types (CPU, GPU)	Bi-E5-2680V3-12c Bi-Phi 7120P-61c UV2000 SMP112c No local HD	180x2xE5-2470-8c 23xNvidia K20m 4xPhi 5110P	189x2xCL-18c 8x4xNvidia V100 1x4x12c fat node	2x24c w/ AVX-512 16x Nvidia V100	Not chosen yet + 2 burst buffer nodes (possibly w/ FPGA or GPU)
Interconnect	IB FDR56 7D enhanced HC	IB QDR40 non-blocking fat-tree	IB HDR200 fat-tree	NVLink (12xNVSwitch, 2.4Tbps/bissection) 8x100Gbps IB	Eth. 100 Gbps Possibly IB
Ethernet	4x10g WAN	4x10g WAN	Not mentioned	Not mentioned	
Storage	1.7 PB Lustre 0.5 PB NFS Ramdisk on nodes	0,15 PB Lustre 0,3 PB Lustre (homes)	Burst buffer scratch 200 TB Home 25 TB	30 TB NVMe SSD	100 TB CEPH
Software	Linux CentOS7 LMOD tools PBS Pro	Linux CentOS7 LMOD tools PBS Pro	Linux RHEL7 PBS Pro	Linux distro packaged by NVidia	Openstack, Kubernetes, VMWare ESXi or KVM, Yorc, HEAppE, LEXIS support services

Table 8: IT4I infrastructure technology and sizing (source: IT4I)

3.1.2 Users' authentication

- Authentication protocol: SSH with RSA private keys,
- Obtaining access:
  - E-mail request signed by a trusted certificate (self-signed certs are not allowed) with name, affiliation, Project ID, written agreement with AUP policy,
  - Internal authorization process,
  - Obtain credentials by encrypted e-mail response: username, SSH private key + passphrase, internal LDAP password,
  - User has access to various other services: VPN, internal SCS portal for accounting and other.

3.1.3 Accounting

- Wall-clock core.h: 1 WCH = 1 CPU core used for 1 hour,
- Normalized core.h:
  - 1 NCH = w • WCH
  - Core.h consumed on a cluster are weighted by their state of obsolescence
- Queues:
  - Limited by max. walltime, number of nodes per job
  - Min. allocation size = 1 node
- Allocation:
  - Open Calls issued every 6 months
  - Submissions evaluated by an internal commitee
  - Director's discretion - individual

3.2 LRZ (COMPUTING RESOURCE PROVIDER)

3.2.1 Overview of HPC and Cloud computing power

The following figure gives an overview of the main HPC clusters available at IT4I as well as the planned Cloud infrastructure, together with parallel file system available on each HPC cluster (GPFS) as well as in the Cloud part (Ceph object storage).

Users on the HPC side connect using their LRZ credentials, all administrated by LRZ staff. Reversely, the Cloud part of the LRZ infrastructure will be accessed using LEXIS credentials (identity federation across the LEXIS project):

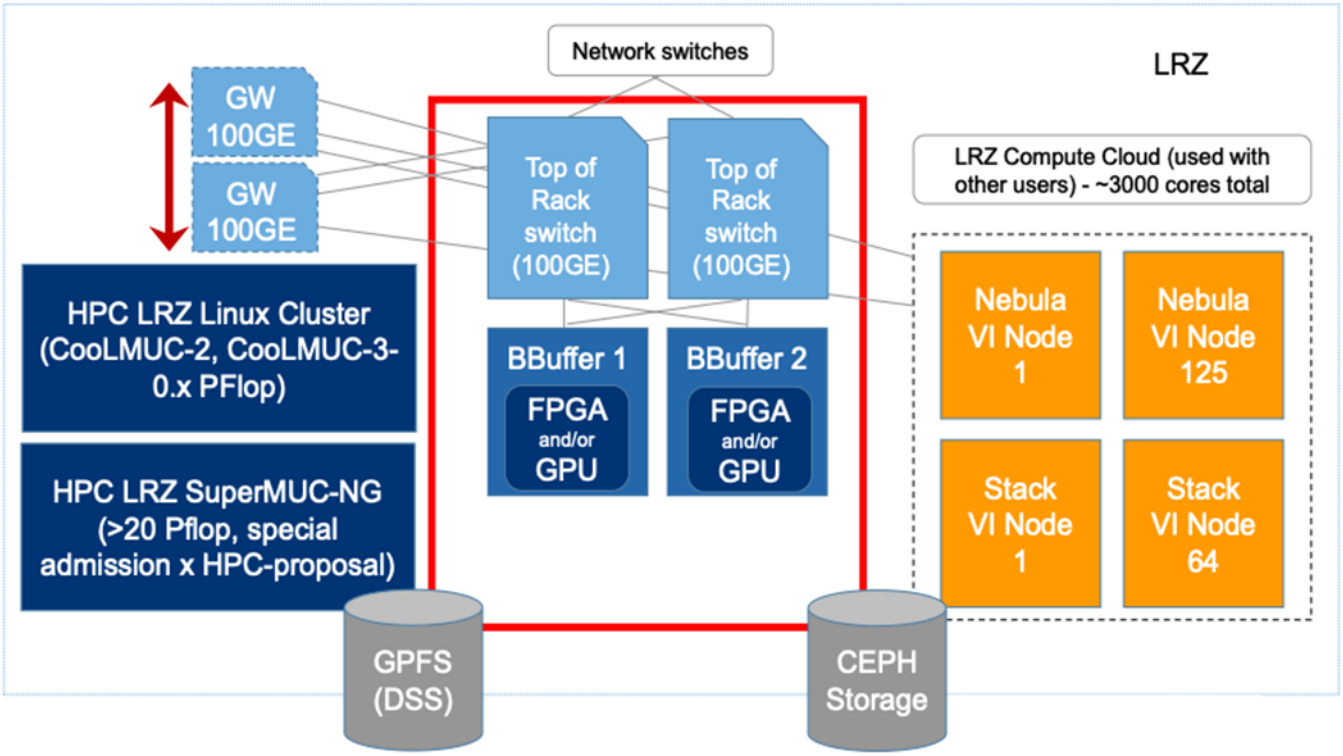

Figure 12: LRZ infrastructure overview (source: LRZ)

The Cloud infrastructure that is planned for LRZ is still under evaluation. LRZ indicates that the following is a good approximation of what they will be able to provide:

- Cloud computing nodes (8 to 10),
- 2 Burst Buffer server nodes,
- Fully redundant 100 Gbps Ethernet connections to the HPC part of the infrastructure,
- Fully redundant 10 Gbps Ethernet connections to NAS (file) storage systems.

	SuperMUC-NG cluster	SuperMUC cluster	Smaller clusters	VMWare IaaS/PaaS	Cloud
RPeak PFlops	20	3	1	N/A	N/A
#Nodes			198		
Node types (CPU, GPU)	X86	X86	10000 KNL cores 10000 HSW cores 640 IVB cores 96 SMP cores	Not mentioned	1000-2000 cores, GPUs
Interconnect	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
Ethernet	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
Storage	100s TB GPFS		GPFS	Not mentioned	CEPH backend + NFS IP-filtered access
Software	Linux SLES12 Slurm	Linux SLES11 IBM LSF	Linux SLES12 IBM LSF	Not mentioned	OpenStack

Table 9: LRZ infrastructure technology and sizing (source: LRZ)

3.2.2 Identity and Access Management (IAM)

As shown in the picture hereafter, LRZ have developed their LRZ-SIM identity management system to record and track their users' identity as well as various authentication services based on:

- LDAP proxies (used for all clusters),
- Grid user administration layer (used for smaller clusters/projects),
- Microsoft Active Directory (used for office automation and messaging systems).

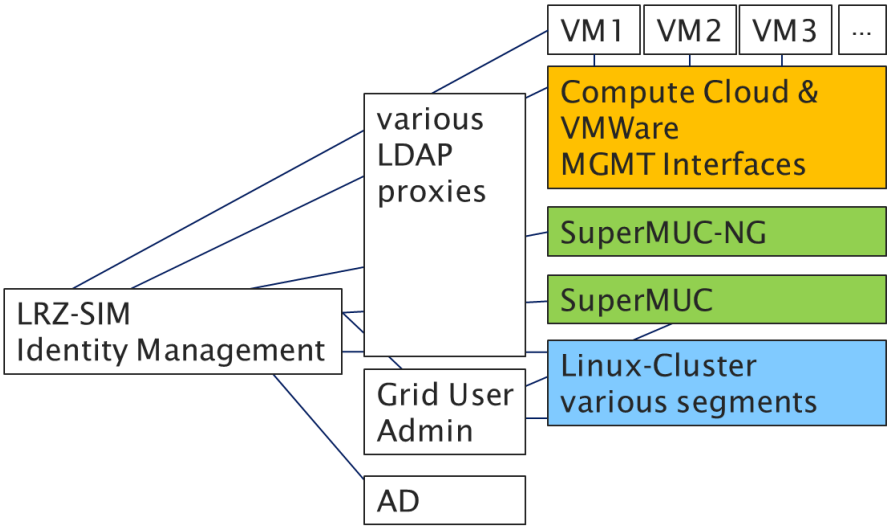

Figure 13: LRZ AIM structure (source: LRZ)

X.509-based authentication allows automatic login on HPC machines. Password-less keys are rejected.

- Petaflops-range clusters are accessible only through incoming IP filtering,
- Other clusters use regular SSH connections (with valid keys),
- Cloud VM security is user-configurable.

3.3 ECMWF (WEATHER & CLIAMATE RELATED DATA PROVIDER)

The Weather & Climate Use Case focuses on a complex system, to provide a diverse set of forecasts: weather, flood, fire, energy, air pollution.

- This system will exploit **Copernicus** and **GEOSS data services**, supplemented and complemented by global and local in-situ unstructured observations,
- Data will be filtered and pre-processed before assimilation to ensure quality. Each layer in the use case will produce large data set of forecast data (from 100+TB global weather models, to MB at the decision maker level),
- Both the assimilated data (inputs) and data produced by each layer (outputs) will be made available through a **Weather and Climate Data API (WCDA)**.
  - between layers (model chains),
  - for further analysis and industrial exploitation.

The system will be applied to several test cases, spanning all the available forecasts, to fine tune the system and demonstrate the innovative value.

The pilot will also include a gateway for collection, filtering and transmission of in-situ observations (developed by LEXIS partner TESEO).

3.4 ADVANCED TECHNOLOGIES UNDER EVALUATION IN LEXIS

Two recent disruptive technologies are being significantly studied and used in LEXIS:

- (Smart) Burst buffers (I/O acceleration based on NVMe devices [2]), referred to as SBB in the present document and on which WP3 is focused,
- Cloud orchestration using the Ystia Orchestrator [3] software, referred to as YORC in the present document and on which WP4 is focused,
- LEXIS service catalogue and Cloud portal for SMEs (WP8).

3.4.1 Smart Burst buffers (SBB)

The Smart Burst Buffers technology proposed by Bull/Atos is a state-of-the-art approach to benefit from ultra-fast I/O capabilities (faster than the HPC parallel file systems themselves). It can be seen as a cache in front of the parallel file systems involving NVME (RAM-like) devices allowing unprecedented I/O performance.

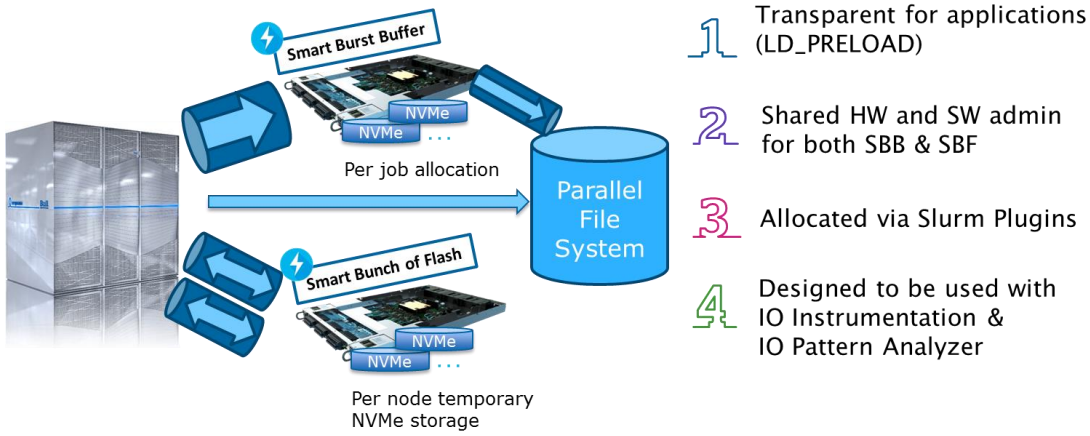

Figure 14: SBB modes: temporary storage or per-job allocation (source: Bull/Atos)

Smart Burst Buffer architecture (SBB):

- iolib intercepts IO calls (LD_PRELOAD)
- libsbb
  - read/write calls redirection
  - fd – file handle mapping
- Client-server communications via IB RDMA
- sbbd daemon
  - Cache IOs on fast storage (RAM & NVMe)
  - Destaging IOs on Lustre file system
-
- **NVME over Fabrics:**
- Export storage spaces over network
  - Network protocol indep. from device type
  - Supports HDD, SSD SATA and NVMe
  - Supports Infiniband RDMA
- Works at block level
  - No namespace sharing
  - No distributed FS
- Possibility to export
  - NVME name spaces (partition)
  - Logical volumes
  - Other block devices

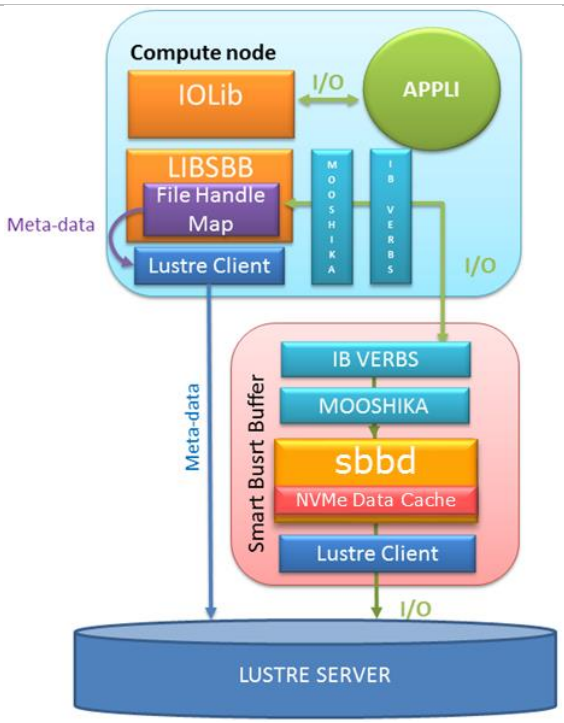

Figure 15: Smart Burst Buffer architecture (source: Bull/Atos)

SBBs are seamlessly integrated with Slurm, resilient and instrumented.

Smart Bunch of Flash use case:

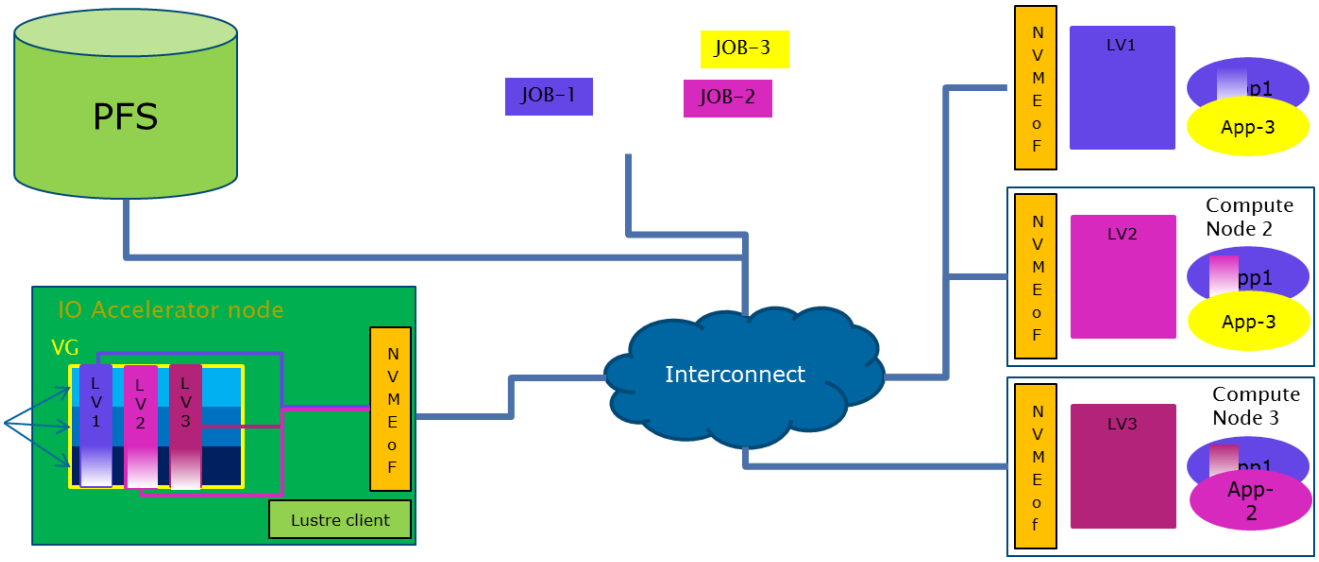

Figure 16: Smart Bunch of Flash sample use case (source: Bull/Atos)

3.4.2 Ystia orchestrator (YORC)

Ystia Suite is an open source software developed in Bull/Atos around other open source components for which Bull is a major or main contributor. The 3 main components are:

- **Orchestrator:** applications lifecycle management over hybrid infrastructures,
- **Forge:** repository for analytics components and application templates,
- **Studio:** Alien4Cloud (**A**pplication **Li**fecycle **En**ablement **for Cloud**), to easily explore, develop fast and deploy to multi-platforms.

Orchestrator overview:

- Objective: support applications lifecycle management over hybrid infrastructures
  - IaaS (AWS, Google Cloud, OpenStack, Hosts Pool),
  - CaaS (Kubernetes),
  - HPC (Slurm, PBS planned).
- Can be extended through plugins,
- Applications modelized in TOSCA [10] (**T**opology and **O**rchestration **S**pecification for **C**loud **A**pplications). OASIS consortium standard language to describe a topology of cloud-based web services, their components and relationships, portable across infrastructures,
- It exposes a REST API, usable directly (like curl), by YORC CLI or via a plugin in Alien4Cloud or UI/Studio companion for YORC.

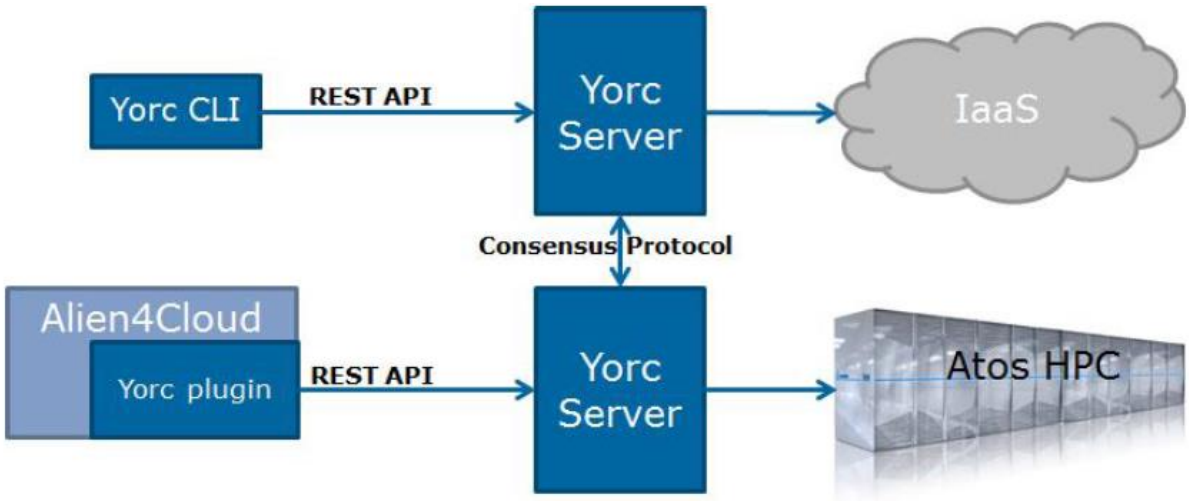

Figure 17: Yorc available user interfaces (source: Bull/Atos)

High availability and entry point configuration:

- The backend (YORC) part supports active-active mode and can be configured to provide a single endpoint (that will be used by the Alien4Cloud frontend), using a DNS domain name managed by Consul (which resolves the address via a round robin method and skips YORC unhealthy instances). Details are available here: https://yorc.readthedocs.io/en/latest/ha.html
- The frontend (Alien4Cloud) supports active-standby mode. There is one endpoint per instance, but a reverse proxy can be configured to expose a single endpoint. Details are available here: http://alien4cloud.github.io/#/documentation/2.1.0/admin_guide/ha.html

4 NEW COMPONENTS REQUIRED FOR LEXIS FEDERATED HPC SOLUTION

During working sessions, especially codesign ones (involving WP2-to-8 representatives), we strived to identify which building blocks were missing or incomplete at the 3 resource and data providers of the LEXIS federation project.

The main role of WP2 is to design, build the target architecture and ensure operations will run smoothly. Even though these building blocks will be extensively presented and studied in deliverable D2.2, they are noted here to highlight some of the main challenges and solutions under consideration within LEXIS.

4.1 FEDERATED IDENTITY, ACCESS AND DATA MANAGEMENT

Due to the focus of the LEXIS project on large scientific data management across 3 European HPC resource and data providers, the notion of **federation** is a key driver of the LEXIS system architecture to be designed. To ensure secure management of data within a federation of service providers, we also need to federate **identity and access management (IAM)** using up-to-date standards.

These aspects are well known and covered in the EUDAT European project. This project already implements advanced collaborative data management services over iRods as well as OpenID Connect (identity management), OAuth2 (Authorization framework) and SAML (for SSO-like support).

4.1.1 AAI - Identity management

This section contains description of the AAI infrastructure and user account lifecycle in the LEXIS platform.

- User is assigned to a certain group/project based on the pilot use-case,
- The user is identified by a login and password and RSA key pair (or similar PKI),
- The user can login to the portal, run a specified workflow and upload/download data using the DDI,
- Its identity is unique across the entire LEXIS platform, including the individual centers connected to the platform.

4.1.2 Federation of users' identity across HPC centers

- Each center will run its own instance of AAI server (LDAP server) which will contain local copy of the platform user list,
- The local copy will contain only users which can run workflows in the given center,
- The local AAI server will be used for authentication to all the platform services running locally (OpenStack, iRODS, Ystia, HEAppE, etc.),
- The local AAI server will be synchronized in a hierarchy with a master AAI server and will be read only from the local services
    - It will synchronize the entire user account, including salted hash password, public key and relevant metadata (flags).
- Each center will run a part of this hierarchy, some centers will be the *providers,* and some will *consumers.*

For example, OpenLDAP supports replication using *provider* and *consumer* paradigm which can be successfully used for this case: https://www.openldap.org/doc/admin24/replication.html

Several technical variants are being made, two of them being presented in Figure 18 and Figure 19. Both are based on single-sign-on (SSO) paradigm, using strong authentication protocols with gateways to local AAI systems via the HEAppE middleware developed by IT4I.

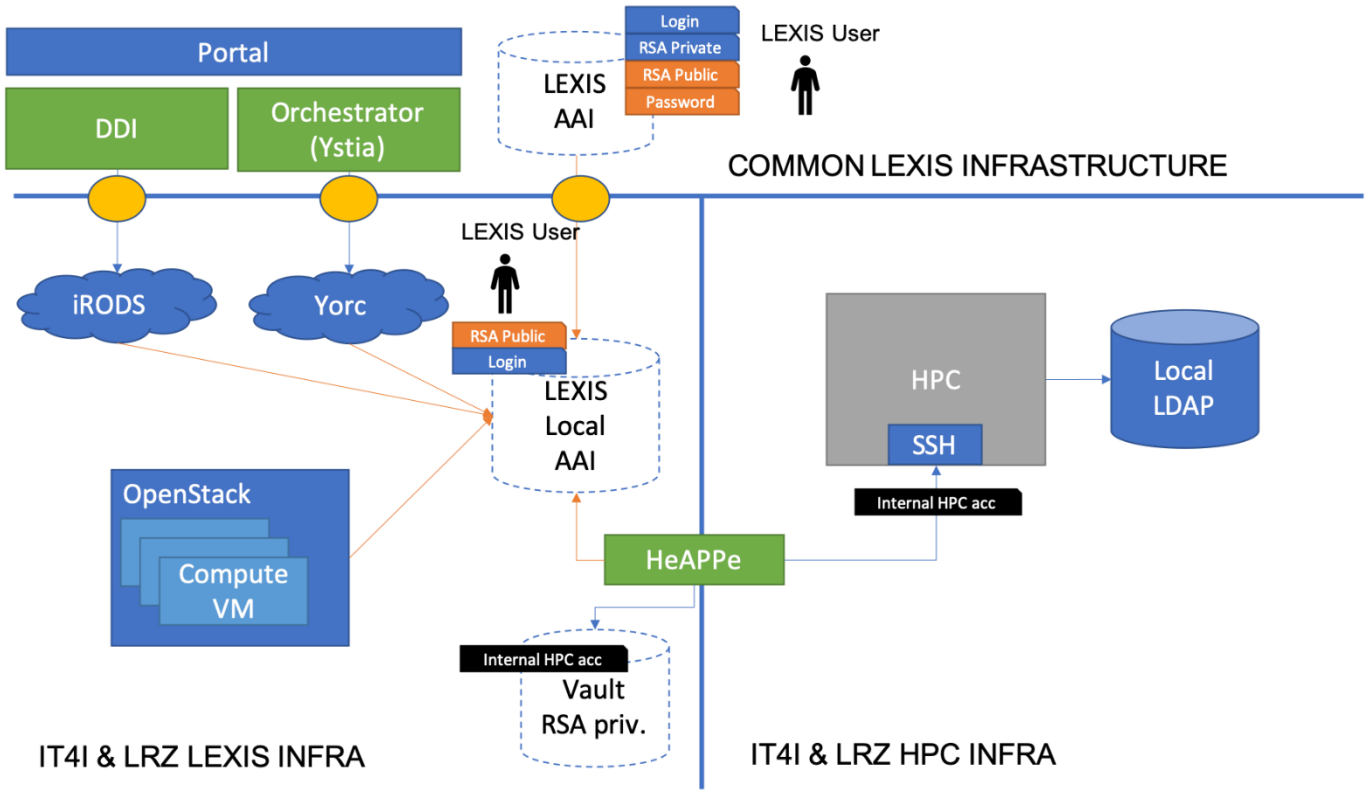

Figure 18: Proposal #1: hierarchical LEXIS AAI (version showing DDI interface)

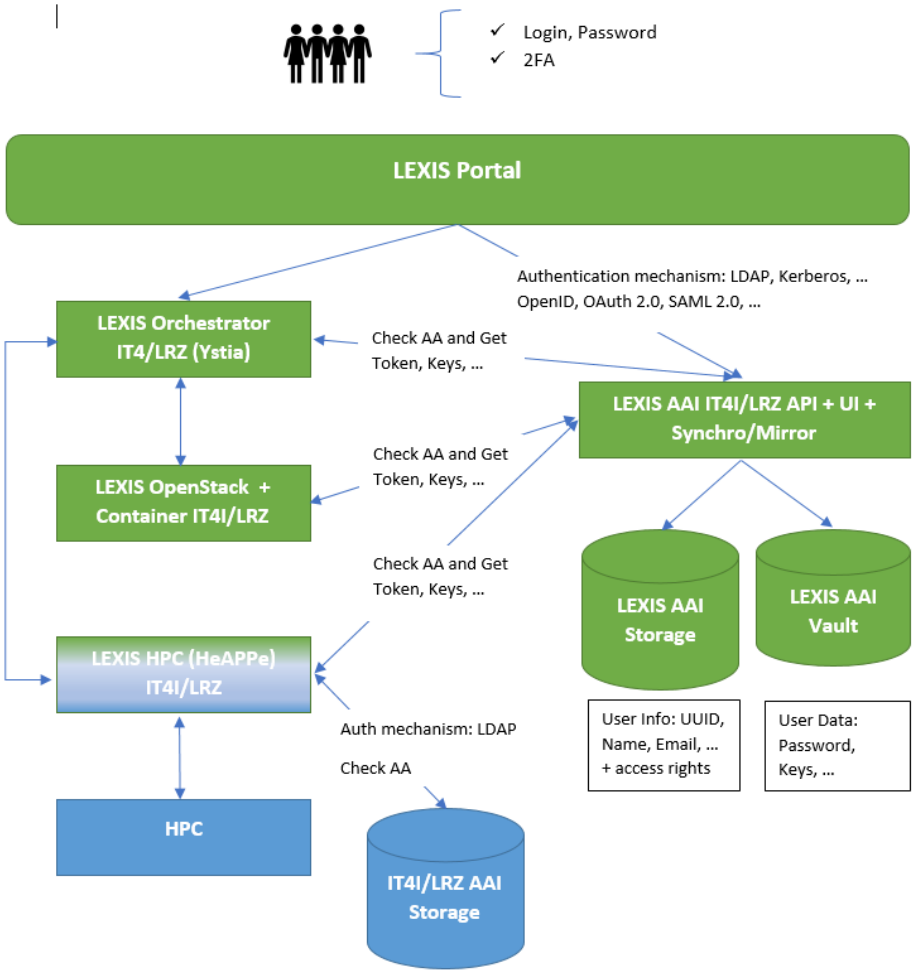

Figure 19: Proposal #2: symmetric/mirrored LEXIS AAI infrastructure (orchestration focus)

4.1.3 User account life cycle

4. **Registration**
   a. User will be able to register on the portal using an external authentication provider (e.g. EUDAT, EduGAIN, Shibboleth or Google and similar),
   b. Once the user is authorized using the following services, the portal will create a user account and write all relevant information to the AAI server,
   c. The user either can set its own password and upload its own RSA public key, or the portal will automatically generate one for it – this can be determined later.
5. **Update**
   a. User changes her/his password/RSA key or login,
   b. This change is automatically propagated to the AAI server hierarchy.
6. **Termination**
   a. User account is deactivated by the system administrator by setting a flag,
   b. User is no longer able to log in to the portal and use the LEXIS infrastructure.

4.2 FEDERATED DATA MANAGEMENT

4.2.1 DDI (LEXIS Distributed Data Infrastructure)

The LEXIS general-purpose federated data infrastructure (for all projects but Weather & Climate curated use cases) will be implemented and managed by replicating a subset of EUDAT and iRODS services. This will be a new system, especially designed for LEXIS, and for which some key EUDAT and iRODS services currently being tested in LRZ.

4.2.2 WCDA (Weather & Climate Data API)

The WCDA is the Weather and Climate Data API that WP7 users will need to retrieve, access and manage curated weather & climate data. The WCDA is not a general-purpose storage; it only deals with curated weather & climate data (for which metadata and content have been checked and accepted). Not all the data used by pilot will be stored and transferred via the WCDA. The WCDA will be mostly be for large, structured, input and intermediate weather & climate data.

The WCDA will be a distributed system, facilitating data transfer between components of weather & climate workflows. The WCDA storage backend, using existing technology (FDB5 -- fields database v5) at each data site can simply be hosted via a directory in a file system. There could, for instance, be a WCDA repository at ECMWF, and "cache" data spaces at LRZ & IT4I for storage close to the HPC. The WCDA could do data transfer between these sites on its own, on instruction from the orchestrator; or the task of data movement could be delegated to the orchestrator (as a staging job, for instance). The WCDA will be responsible for serving data to any application irrespective of its physical location.

LRZ indicates that the WCDA requirements in terms of data volume should be known in advance and that GPFS volumes could only be visible through NFS exports.

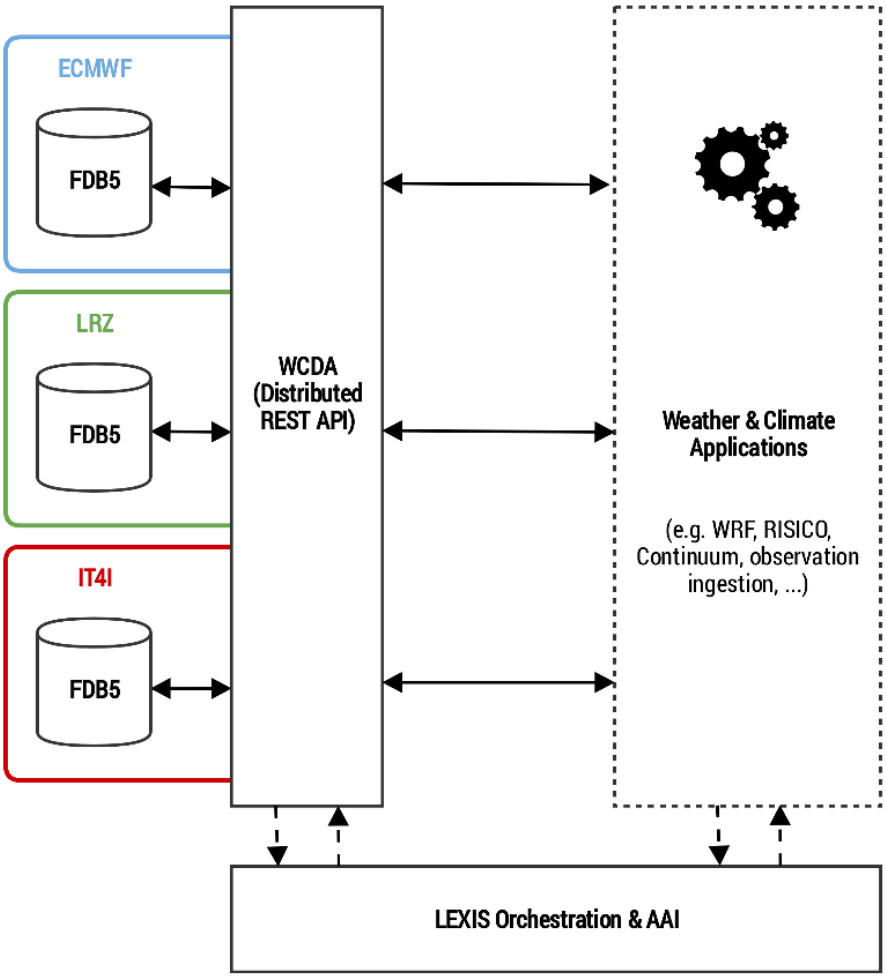

Figure 20: Preliminary system design of the WCDA with distributed REST API and FDB5 storage.

Details on LEXIS orchestration, DDI, AAI and computing are intentionally omitted as they do not impact the core WCDA design.

4.3 CLOUD-TO-HPC SYSTEM INTERFACES

4.3.1 Multi-tenancy

This isolation and privacy between Cloud tenants is a common but key practice to which Cloud software stacks bring lots of tools (like pre-packaged firewalling, VPN wrappers, software-define networking). Reversely, this constraint had never been considered in HPC (until very recently) for many technical and cultural reasons we will not detail here.

Therefore, LEXIS must design secure system interfaces between the Cloud and the HPC parts in a way both parts keep relying on their own, proven security model. These system interfaces must at the minimum, cover:

- Data sharing,
- Job submission and control,
- LEXIS-to-HPC service provider user mapping (see Section 4.1),
- Remote visualization.

4.3.2 Interconnects

Interconnect such as Infiniband are HPC clusters' private backbones allowing very large scalability for parallel applications, and typically MPI applications. These interconnects not only form a fabric to connect compute nodes but also parallel file system I/O nodes (such as Luster or GPFS) and often 3D-accelerated visualization nodes.

IT4I, for instance, will configure a gateway server running both Lustre client software and an iRods service connected to its counterpart on the Cloud side.

4.3.3 NFS

For these reasons, and to be able to connect these 2 incompatible worlds, the decision was made to implement data sharing between burst buffers (on the cloud side) and parallel file systems (on the HPC side) using NFS mounts as a data gateway. We will use ability from both GPFS (in LRZ) and Lustre (in IT4I) to support NFS-compatible exports to achieve this goal while keeping the data management architecture homogeneous.

4.3.4 Overview of the resulting data management system architecture

Application workflows will be dispatched over both LEXIS (cloud) and HPC infrastructures which implies that different interfaces (iRODS for long term, federated data storage and NFS to use burst buffer acceleration from HPC nodes) will be used to connect both worlds as shown in the figure below (IT4I example).

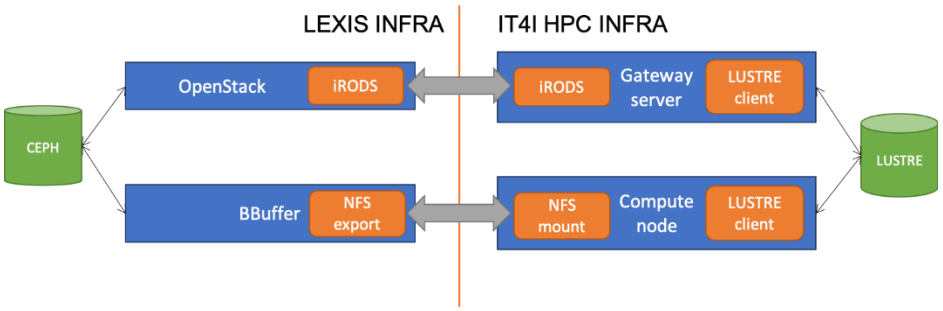

Figure 21: Data management system architecture overview (source: IT4I)

4.3.5 Overview of the resulting job management system architecture

The following picture depicts how the proposed approach is to be applied at IT4I.

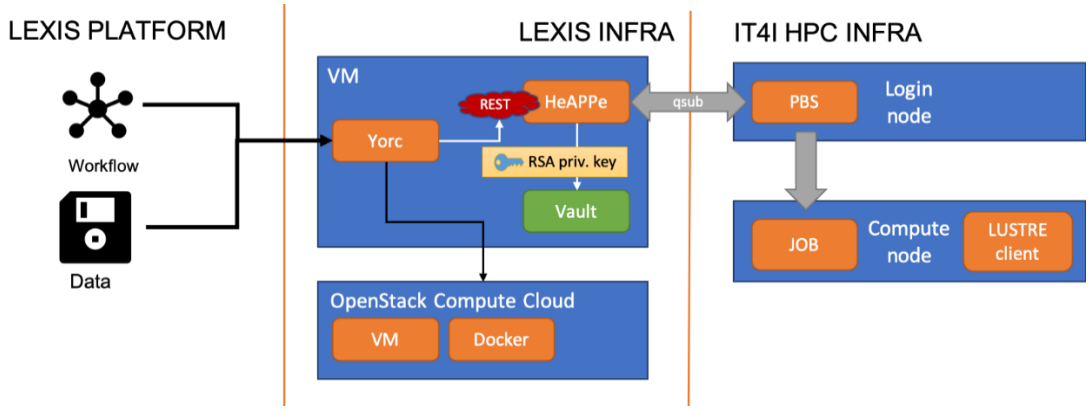

Figure 22: Job management system architecture overview (source: IT4I)

4.4 WORKLOAD TYPES

4.4.1 Virtualized (cloud stack and hypervisor)

Both IT4I and LRZ have the OpenStack open source Cloud software deployed to manage their Cloud infrastructure. Open Stack, is, in other words an open set of IaaS building blocks able to operate several kinds of hypervisors.

This leaves room to both IT4I and LRZ to select their favourite hypervisor(s) while still hiding this complexity behind OpenStack. OpenStack orchestration is itself triggered from the YORC orchestrator (which understands OpenStack orchestration, as well as other Cloud solutions). This way, all these Cloud infrastructure and platform related technical concepts will not be exposed to the end users. In addition to this, communities expecting to use LEXIS through the portal, may not even have to be aware of YORC but would transparently configure it from web forms published in the portal.

OpenStack has primarily been designed for virtual machines (VM) even though it can, indirectly handle containerized and even bare metal workloads.

However, the bare metal mode is somewhat opposite to the Cloud paradigm and would be extremely heavy to handle. We have no interest in opting for this approach and this is not where HPC is going globally.

4.4.2 Containerized

Even though virtualization can be considered as the main technical building block of Cloud computing, VMs are progressively leaving way to lighter, faster and easier-to-manage concepts such as containers. Containers can be used to package applications or complete workflows in a way that they can be immediately executed on a Cloud platform, no matter what the Cloud platform uses in term of hypervisor or operating system.

The most popular container technologies are Docker, Singularity, OpenShift. Singularity is the most promising and popular in the HPC community.

As for VMs, containers also require orchestration to handle their life cycle (creation, run, stop, deletion…) and make it possible to handle many of them at once. The most popular container orchestration technology is Kubernetes.

The YORC orchestrator supports Kubernetes too. It will be interesting to identify which applications or workflows can take advantage of being containerized in the context of LEXIS (as opposed to being installed in regular VMs). However, containerized workflows are not mandatory for LEXIS.

4.4.3 Handling of windows workloads

Some Windows-only application software have been identified in the project. Windows containers are still immature technology. Windows platforms also come with rather significant constraints in terms of Microsoft licensing and context of use. So far, only virtual machines seem realistic to run Windows workloads on the cloud side of LEXIS, but this will be explored further.

4.4.4 Workloads supported by the Ystia Orchestrator (YORC)

Ystia already supports TOSCA applications / jobs over the following infrastructures or platforms:

- OpenStack,
- Kubernetes,
- Bare metal HPC cluster running SLURM job scheduler,
- Cluster of SSH-enabled hosts,
- Various public Cloud APIs.

LEXIS will explore orchestration with OpenStack (the IaaS existing in IT4I and LRZ) and possibly Kubernetes (if the project shows interest in container deployments). In case an application is tagged as “containerized”, a compute node is allocated, system dependencies (e.g. Docker) are installed and the container is seamlessly executed. Bare metal HPC applications will be handled through YORC’s interfaces with job schedulers and/or clustered SSH commands.

4.4.5 Features to be developed in the LEXIS context

- Ability to deploy an application over a hybrid infrastructure (as of today, application components can only be deployed in a single infrastructure),
- Ability to define placement policies to indicate where to deploy the application (as of today, the user must manually set the target location),
- Ability to upgrade a deployment (works in progress),
- The LEXIS federated AAI will require a new orchestrator plugin to interface with the HEAppE authentication service, used as a gateway between LEXIS accounts and local HPC accounts on each premise, ideally in the frame of the OAuth2 framework. YORC backend supports the addition of new plugins and a custom development for HEAppE support has been planned.

4.5 LEXIS PORTAL

WP8 focuses on the LEXIS portal for SMEs. As WP8 does not start until M4 of the project, many of the specifics relating to the design of the portal have not been detailed. The full set of requirements of the portal are still under discussion, but the following basic requirements clearly need to be implemented:

- Support for registration with the LEXIS system,
- Support for listing available data sets,
- Support for deploying workload to the LEXIS Cloud resources,
- Support for deploying workload to the LEXIS HPC resources,
- Support for monitoring the status of the deployed job,
- Support for determining the incurred cost of running a given job,
- Support for estimating in advance the cost of running a given job.

A basic release plan has been devised, with an indicator of the specific functionalities to be provided within the different releases as follows:

- Release 1 (R1), delivery M9: limited system capabilities including register/login, list available data sets, see T&Cs of use of the data sets, link to documentation describing the data sets and download samples of the dataset,
- Release 2 (R2), delivery M15: increasing capabilities including basic support for submitting a job using limited resources on the LEXIS Cloud, support for accounting and billing integration,
- Release 3 (R3), delivery M24: support for deploying workflow to HPC infrastructure, basic job cost estimation and monitoring supports,
- Release 4 (R4), delivery M30: enhanced views of job status and interface support for downloading results.

The design of the portal is still under discussion with a number of issues requiring clarification, mostly around how the portal will interface with the other elements of the system including the AAA mechanisms, the data store mechanisms and the orchestration/job control mechanisms. These interfaces are becoming clearer as WP2 evolves and WP8 expects to have a initial system design in a M4-M5 timeframe such that R1 can be delivered in M9.

4.6 TESEO'S SMART GATEWAY

TESEO's smart gateway (for WP7 / Weather Forecast use case) will:

- Amalgamate required IoT data sources,
- Connect to/provide a service which can be used to retrieve said data.

This gateway will be accessed by CIMA, who will handle the conversion and pre-processing of the IoT observations and put these observations in the Weather and Climate Data API. This complements other observations pipelines e.g.:

- Meteonetwork has a gateway/API from which CIMA will fetch, convert and provide the data to the WCDA,
- Moji (MoWeather mobile app) will provide a collated data service to ECMWF, who will convert and provide the data to the WCDA.

These pipelines should be independent of each other (e.g. Meteonetwork data should not be amalgamated in TESEO's gateway) and the WCDA will provide the unification of these data sources.

4.7 USE OF FPGA TECHNOLOGY IN LEXIS

Some LEXIS partners have FPGA expertise and two FPGA stakeholders (in Bull/Atos) have been named and assigned the coordination of the various possible FPGA use case studies.

At this (early) stage of the project, three topics may show some potential and will require further technical investigation:

- **Use cases:** check if and how WP5 (maybe WP6) applications could use them efficiently (Navier-Stokes differential equations) and compare results to CPU and GPU implementations,
- **Smart Burst Buffers:** study FPGA-accelerated encryption to secure SBBs with without compromising performance,
- **Orchestration:** expose the use of FPGA in TOSCA application modelling and YORC orchestration heuristics.

4.8 LARGE DATA SHARING, TRANSFERS AND REMOTE VISUALIZATION

Following a general trend in Scientific computing, LEXIS use cases will use and generate very large amounts (hundreds of gigabytes to terabytes) in several locations. To accelerate or even avoid prohibitive transfer times, across LEXIS locations and / or between Cloud and HPC parts, one enabler is an efficient network backbone between locations, other ones are 3D remote visualization or bringing the workload to the data location (using containers or virtual machines).

Network performance

ECMWF, IT4I and LRZ are planned to connect through large scientific networks (federated in the Géant [11] project). Available bandwidth and mode of access are being assessed. Test results will be presented in D2.2 [1].

Remote visualization for interactive tasks

On the visualization side, many scientific workflows require human, interactive data preparation (pre-processing) such as the building or tuning of 3D geometric models, meshing, the application of constraints to models or the selection of an interest area in a wider dataset. It goes the same for post-processing to validate computation results or make decisions about cancelling running jobs.

Several LEXIS partners have experience in using or design remote visualization solutions based on classical desktop or application remoting technologies, providing transparent GPU acceleration to Cloud users. also known as 3D API Intercept such as TurboVNC, XPRA, NX, Nice DCV or Bull XRV etc.

Other approaches such as 3D VDI (Citrix XenDesktop, VMWare Horizon, etc.) use GPU virtualization and 3D remoting APIs or, more recently, software-defined visualization (SDViz pushed by Intel for instance).

Although the interest in such techniques has been clearly identified in the project, the use cases' stakeholders admitted little or no experience (for most of them).

Including 3D remote visualization in the LEXIS service portfolio with seamless integration in the LEXIS portal could bring significant added value in terms of usability, efficiency and resource consumption (network bandwidth and storage) thanks to some partner's experience like LRZ, IT4I and Bull/Atos.

5 CONCLUSION

After 3 months of active codesign (involving all partners: application use cases stakeholders, HPC service providers and technology vendors), the required high-level system architecture assessments have been made and LEXIS now has a global and precise view of its current assets. Some areas still require some investments and development, but the objectives have been met on time:

- The requirements of the pilots are detailed in Section 2,
- The review of available infra is detailed in Section 3.

Throughout the work, the components/functional blocks required to build the federated LEXIS solution have been investigated further but the detailed architecture of the LEXIS solution is still ongoing.

Major missing components to reach the project's objectives were identified early enough like Smart Burst Buffers for I/O acceleration, Cloud and HPC resources orchestration including data locality related rules or heuristics, FPGA technology, remote visualization, LEXIS user portal and so on.

The fruitful codesign sessions motivated the analysis of a few new ones, like federated security and data management, the choice of one or more Cloud stacks and hypervisors, the use of containerized workloads. Some turn out to be challenging, but several tracks, methodology and tools have already been discussed in codesign to handle them efficiently and in a modern way. They will be presented in detail in the next deliverable D2.2 [1].

At this stage, we estimate that most of the macroscopic target LEXIS architecture and data management is ready on paper (existing assets and components to be added), as reflected Figure 23:

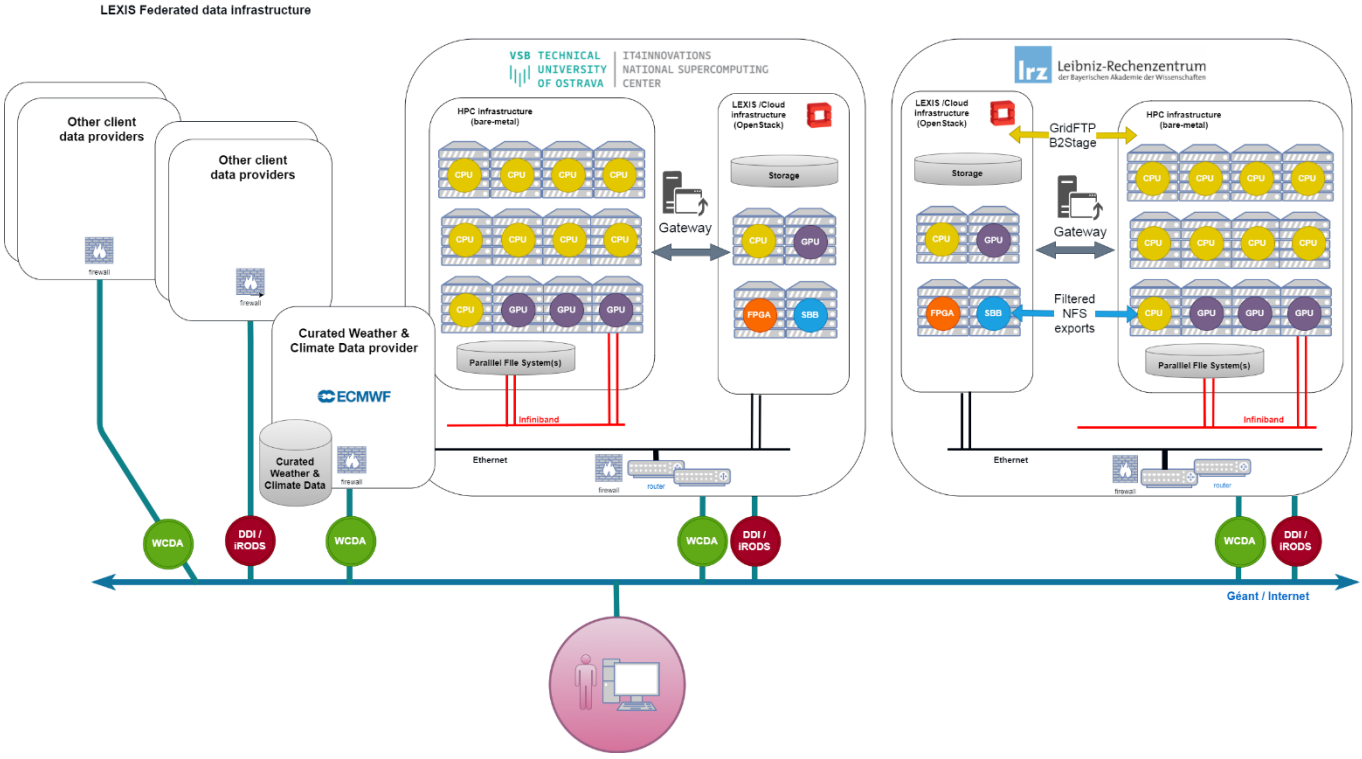

Figure 23: LEXIS federated data infrastructure overview

REFERENCES

[1] LEXIS Deliverable, *D2.2 Existing technological assets and required technology.*

[2] https://insidehpc.com/2019/02/atos-steps-up-with-nvmeof-flash-accelerator-solutions/

[3] https://ystia.github.io/

[4] https://www.lrz.de/services/compute/

[5] https://docs.it4i.cz/salomon/introduction/

[6] The Role of Laminar-Turbulent Transition in Gas Turbine Engines, Robert Edward Mayle of Rensselaer Polytechnic Institute, New York – ASME paper 1991, Orlando

[7] www.paraview.org

[8] https://www.ecmwf.int/en/research/modelling-and-prediction

[9] https://en.wikipedia.org/wiki/Weather_Research_and_Forecasting_Model

[10] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

[11] http://www.prace-ri.eu/IMG/pdf/PD16-11-UF-4-3-PRACEdays-CHEVERS.pdf