



**UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH**

**Facultat d'Informàtica de Barcelona**



# **DEVELOPMENT OF A NEW ESM WORKFLOW FOR PRE-EXASCALE PRODUCTION EXPERIMENTS**

**LEO ARRIOLA MEIKLE**

**Thesis supervisor:** MARIO CESAR ACOSTA COBOS (Barcelona Supercomputing Center)

**Thesis co-supervisor:** MIGUEL CASTRILLO MELGUIZO

**Degree:** Bachelor Degree in Informatics Engineering (Computing)

**Bachelor's thesis**

**Facultat d'Informàtica de Barcelona (FIB)**

**Universitat Politècnica de Catalunya (UPC) - BarcelonaTech**

**18/10/2023**

# Contents

<b>1</b>	<b>Context &amp; Scope</b>	<b>5</b>
1.1	Context . . . . .	5
1.1.1	Introduction . . . . .	5
1.1.2	Concepts . . . . .	6
1.1.3	Implicated Actors . . . . .	7
1.2	Justification . . . . .	7
1.2.1	Problem description . . . . .	7
1.2.2	State of the Art . . . . .	8
1.2.3	Justification of choice . . . . .	13
1.3	Scope . . . . .	14
1.3.1	Objectives . . . . .	14
1.3.2	Risks . . . . .	14
<b>2</b>	<b>Software and System requirements</b>	<b>16</b>
2.1	The ICOsahedral Nonhydrostatic Model (ICON) . . . . .	16
2.1.1	Description . . . . .	16
2.1.2	Installation . . . . .	17
2.2	Workflow manager environment . . . . .	18
<b>3</b>	<b>Temporal planning</b>	<b>20</b>
3.1	Existing project planning . . . . .	20
3.2	Task description . . . . .	20
3.3	Resources . . . . .	22
3.3.1	Human Resources . . . . .	23
3.3.2	Material Resources . . . . .	23
3.4	Methodology and status reporting . . . . .	24
3.5	Gantt chart . . . . .	25
<b>4</b>	<b>Economic Management</b>	<b>26</b>
4.1	Personnel costs . . . . .	26
4.2	Generic costs . . . . .	27
4.3	Contingencies . . . . .	29
4.4	Unforeseen events . . . . .	29
4.5	Total Costs . . . . .	30
4.6	Managment Control . . . . .	30
<b>5</b>	<b>Project planning deviations</b>	<b>31</b>
5.1	Redefinition of tasks . . . . .	31
5.2	Estimated additional costs . . . . .	32
<b>6</b>	<b>Implementation</b>	<b>33</b>
6.1	Workflow specification and structure . . . . .	33
6.1.1	Workflow task design . . . . .	34
6.1.2	Workflow tasks . . . . .	35
6.2	First version of the workflow . . . . .	36
6.2.1	New configuration scheme . . . . .	36
6.2.2	HPC Compatibility . . . . .	38

<b>7</b>	<b>Testing and Key Performance Indicators (KPI's)</b>	<b>39</b>
7.1	Testing suite . . . . .	39
7.2	Workflow testcases . . . . .	40
7.3	Workflow KPI's . . . . .	40
<b>8</b>	<b>Conclusions</b>	<b>41</b>
<b>9</b>	<b>Sustainability report</b>	<b>42</b>
9.1	Self-evaluation . . . . .	42
9.2	Economic aspect . . . . .	42
9.3	Environmental aspect . . . . .	42
9.4	Social aspect . . . . .	42

## Acknowledgments

I would like to express my deepest gratitude to everyone who has helped me during the development of this project. Specially to my supervisors Miguel and Mario, which have managed to squeeze out some time to meet with me and to revise my work.

I also want to extend my thanks to my colleagues in the workflows team from Destination Earth. Without the team's collaborative spirit, I'm sure the end result of the project would look quite different.

# 1 Context & Scope

## 1.1 Context

This project documents the development of a new automated workflow for a state-of-the-art Earth system model in the context of a Digital Twin of the Earth. The workflow involves running simulations at very-high resolution, which output vast amounts of data. More specifically, the workflow provides the necessary steps for running different types of simulations for ICON (Icosahedral Nonhydrostatic Weather and Climate Model) in pre-Exascale HPC Platforms like LUMI or Marenostrum 5.

### 1.1.1 Introduction

To address the urgent climate crisis, we need to mitigate and adapt to our changing climate. If left unaddressed, the climate emergency will have severe consequences for the planet, including the loss of biodiversity, the displacement of people due to rising sea levels and extreme weather events, and the negative impacts on food security and public health [Comb]. In fact, it is already causing these consequences in many regions of the world, specially in the global south, regions that are generally the least historically responsible for emitting greenhouse gasses [Ass].

In the midst of this global emergency, the European commission has issued the flagship initiative [Comc], **Destination Earth 1**. Its purpose is to develop a highly accurate digital model of the Earth on a global scale. This model will monitor, simulate and predict the interaction between natural phenomena and human activities.

Up to date, **Destination Earth** is the most ambitious program that the European Union has approved and initiated in order to build a digital twin of the Earth system.



Figure 1: Destination Earth project showcase (source: [ESA])

It will unlock the potential of digital modelling of the Earth systems at a level that represents a real breakthrough in terms of accuracy, local detail, access-to-information, speed and interactivity. The main focus is on building the best possible tool to predict with as much accuracy any extreme weather events, and to adapt to them.

The **Destination Earth** initiative will implement these three key components in order to achieve the final Digital twin;

- The **Core Service Platform**, which is going to be operated and developed by the European Space Agency (ESA).
- The **Data Lake**, which will be operated by the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT).
- The **Digital Twins**, these will be digital replicas of the highly complex Earth systems, and will be developed by the European Center for Medium-Range Weather Forecasts (ECMWF)

Having described the components of the underlying project, this work has been developed under the third key component of **DestinE**, the **Digital Twins**. This part of the **DestinE** project is subdivided even further and counts with the collaboration of many institutions.

### 1.1.2 Concepts

In order to discuss the contents of the work I develop during my thesis, I will describe some of the main concepts in an attempt to create a common ground to ease comprehension.

**Climate Simulation** A climate simulation is generally a specific realisation of a climate model. These models and simulations attempt to replicate the behaviour of the Earth's climate system, including the interactions between the atmosphere, oceans, land, and as many components and processes as the model allows.

These simulations can be used to make projections about how the climate may change in the future under different scenarios of greenhouse gas emissions, as well as to understand past climate change and their causes. Some examples of climate simulations could be ;

- A historical climate simulation, where the purpose of such is to simulate and study the Earth's climate over a period of time in the past.
- A shared socio-economic pathway simulation is a simulation type which is actually being used by the Intergovernmental Panel on Climate Change (IPCC) which in essence tries to correlate socioeconomic drivers, such as population growth, economic development, and technological progress with the impact it may cause to the Earth climate systems.

**Earth System Model** An earth system model is a computational model that attempts to replicate the Earth climate systems; it achieves this via numerical modelling. Earth system models usually include separate coupled components or independent models that simulate the different actors in the Earth's climate.

These components could be the atmosphere, the ocean, the ecosphere, etc... Earth system models are then used to run climate simulations, as described above.

In order to do this, Earth System Models work on three spatial dimensions, which employ spherical grids which subdivide the model's spatial domains into cells. And the temporal dimension, which is added through internal model times stepping. These cells then apply their physics and chemistry modules on the available data, and then communicate and propagate the changes at short intervals.

**Workflow** In a more general sense, a workflow consists of a series of steps that need to be completed in a process. The workflow that is being developed in the **DestinE** is a climate simulation workflow, therefore it includes all the necessary actions needed to successfully run the different earth system models.

**Workflow manager** A workflow manager orchestrates a workflow. It ensures that all the steps are run correctly while adding extra features. Its purpose is to work as an interface between users and the models themselves, allowing enough flexibility and modularity for researchers to design and run their experiments.

These workflows may involve multiple computing systems for their completion, from HPCs to post-processing clusters or workstations.

### 1.1.3 Implicated Actors

As mentioned above, this work has been developed under the **Destination Earth** project. Eight or more different institutions are taking part in the required developments for the Climate DT, one of them is the Barcelona Supercomputing Center, which is mostly working with the IFS-NEMO model, although also contributing in other aspects like the workflow for the ICON model, the output handling and retrieval or porting everything to **Marenostrum 5**...

Regarding the implicated actors, whom I will work and interact during the developments of this thesis, will be researchers in the **Computational Earth Sciences (CES)** group in the Barcelona supercomputing Center.

Researchers in the **Max Planck Institute for Meteorology (MPI)** which take care of the developments for the ICON model. Moreover, researchers from the **Alfred Wegener Institute (AWI)** which will work in the other model's workflows specifics but with whom I'll develop collaboratively the general workflow structure.

And finally, which whom I'll work more closely on the specifics of this thesis will be my thesis Director and Co-director:

**Mario Cesar Acosta Cobos** Director of this thesis and co-leader of the Computational Earth Sciences group.

**Miguel Castrillo** Co-director of this thesis and co-leader of the Computational Earth Sciences group.

## 1.2 Justification

### 1.2.1 Problem description

The Climate Digital Twin is really computationally expensive, it has many components and has very high throughput. In addition, what really makes the climate DT workflow stand out in comparison with common climate prediction workflows is that the workflow has to manage extremely high resolution output and allow a great degree of interactivity and efficiency.

In this sense, it has to make use of the latest advancements in High Performance Computing (HPC) as it will run in many State of the Art HPC Platforms. The final workflow is intended to run on the **Marenostrum 5** supercomputer, which is hosted in the BSC, The LUMI Supercomputer, hosted by CSC in Finland, and others in the **EuroHPC** JU in Europe.

Apart from being computationally expensive, it also has to process great amounts of data from the climate models. Also, incorporated into the workflow will be applications that will feed on this data and generate case specific outputs.

In this general workflow, the primary focus of my thesis is to develop a comprehensive workflow for handling climate models. It has to permit that the Models (specially the ICON Model) can be transferred, compiled and run for experiments using different model settings

across various High-Performance Computing platforms. The goal is to develop a workflow that enables researchers to run different types of simulations with varying settings.

Moreover, it is imperative that the developed workflow is modular and allows for customization. The modularity allows researchers to easily adapt the workflow to their specific requirements, and the customization ensures that the workflow meets their needs.

### 1.2.2 State of the Art

#### EuroHPC Joint Undertaking

Resulting of complex international cooperation, the European High-Performance Computing Joint Undertaking (EuroHPC JU) is a private-public consortium that allows Europe to coordinate its efforts and pool their resources to develop top of the range exascale supercomputing and quantum computing.



Figure 2: EuroHPC JU Logo

EuroHPC JU was created in 2018 and includes a total of 33 Member states and associated countries. Spain is one of the associated members, with the **Barcelona Supercomputing Center** being one of the main implicated actors. The participation in EuroHPC JU has been key in the realization of **Marenostrum 5**.

Currently the EuroHPC JU has a total of eight supercomputers located across Europe:

HPC Platforms		
HPC Name	Location	Peak performance
LUMI	Finland	550 Petaflops
LEONARDO	Italy	323,40 Petaflops
MARENOSTRUM 5	Spain	314 Petaflops
MELUXINA	Luxembourg	18,29 Petaflops
KAROLINA	Czech Republic	15.69 Petaflops
VEGA	Slovenia	10,05 Petaflops
DEUCALION	Portugal	10 Petaflops
DISCOVERER	Bulgaria	5,94 Petaflops

Figure 3: EuroHPC JU HPC Platforms (source: [\[Und\]](#) )

This consortium is key in the digital transformations in science and innovation, which are key pillars in the European Digital strategy. The main research areas which will benefit from the EuroHPC JU are; fundamental physics, material sciences (drug design) and finally Earth sciences, with projects like the **Destination Earth** initiative.



## Earth Sytem Models

The ICON (Icosahedral Nonhydrostatic) model is one of the several major forecast models in the world. It describes atmosphere, ocean and land by means of the component models ICON-A for the atmosphere, ICON-O for the ocean and ICON-L for the land. These components are coupled by fluxes of energy, momentum, water, and carbon dioxide at the surface. In order to couple the different components it makes use of the YAC coupler.

A more detailed description of the model is given in section 2.1.

## Workflows

This work benefits from the existing experience with State-of-the-Art workflows like the EC-Earth workflow, developed by the Earth Science Department at the BSC.

The EC-Earth model is a global climate model system which is based on the idea to use the world-leading weather forecast model of ECMWF (European Centre of Medium Range Weather Forecast) in its seasonal prediction configuration as the base of the EC-Earth climate model. The model system can be used in several configurations including the classical climate model (atmosphere, soil, ocean, sea ice) and Earth System configurations (adding atmospheric chemistry and aerosols, ocean bio-geo-chemistry, dynamic vegetation and a Greenland ice sheet).

Auto-EC-Earth is a workflow that is mostly maintained and developed by the Computational Earth Sciences group in the Barcelona Supercomputing Center. And, as its name indicates, it makes use of the Autosubmit Workflow Manager.

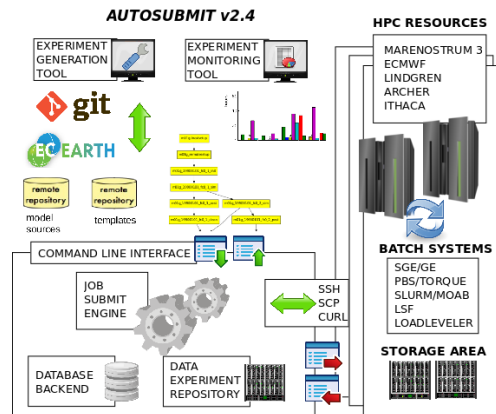


Figure 4: Autosubmit v2.4 and Ec-Earth Workflow (source: [Teaa])

In the Auto-EC-Earth workflow [24], Autosubmit, takes care of interactions with the HPC Platforms alongside with handling all user configuration.

## Workflow managers

Workflow managers are used to orchestrate workflows with ease. The most important workflow managers in the weather and climate community also have meta-scheduling features and work as an interface between the researchers and the HPC platforms.

They are usually responsible for submitting the tasks, taking care of their dependencies and allowing as much customization as possible. This usually translates into allowing users to set custom configurations for the platform specifics, Slurm, PBS directives or even model configuration variables.

Some notable workflow managers are `Cylc`[\[Con\]](#) and `Ec-Flow`[\[Tead\]](#), which in difference to `Autosubmit` use a client-server structure with dedicated TCP ports. In any case, the main workflow used in `Destination Earth` is `Autosubmit`.

**Autosubmit** Autosubmit [Teab] is a workflow manager designed to meet climate research necessities. It integrates the capabilities of an experiment manager, workflow orchestrator and monitor in a self-contained application. It allows defining and configuring experiments, supported by a hierarchical database that ensures reproducibility and traceability.

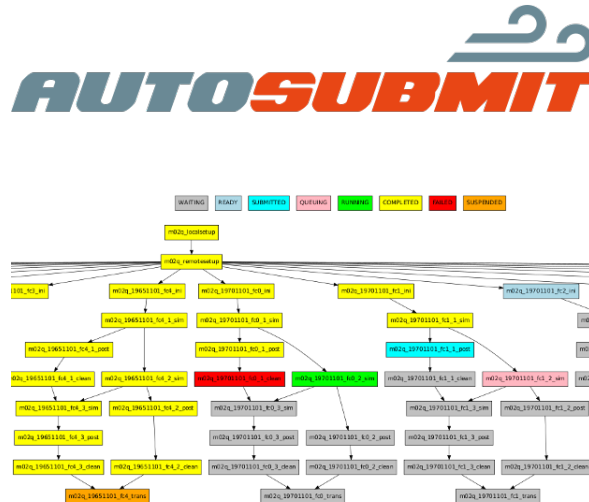


Figure 5: Autosubmit workflow viewed on the Autosubmit GUI (source : [Teac](#))

**Autosubmit** is designed to run complex workflows in research and operational mode by managing their dependencies and interfacing with local and remote hosts. These multiscale workflows can involve from a few to thousands of steps and from one to multiple platforms.

To manage experiment workflows, autosubmit makes use of a long list of commands, some of the most widely used are: `autosubmit expid`, `autosubmit create`, `autosubmit run...`

Autosubmit is widely used in the Barcelona Supercomputing Center. Apart from the BSC, Autosubmit is now expanding its user base in the context of the Destination Earth project.

## Experiment managers

Although most workflow managers include some degree of experiment customization capabilities, most experiment definition is edited manually and then incorporated into the workflow as part of a job. Usually this is done via name lists which include most of the configuration variables for the models.

In this sense, `Autosubmit` also allows a great degree of experiment definition. Specially in version 4.0, it allows doing so through `YAML` configuration files, which allow greater flexibility, change tracking and general workflow reproducibility.

The workflow for Destination Earth is actually the first workflow to make use of this new flexible configuration. The idea is that the new feature allows users to use and define as many custom configurations as desired.

Therefore, **Autosubmit** integrates the configuration, management, and the orchestration part of experiments. The management part is based on a database for the traceability of experiments and their subsequent analysis.

There are tools to facilitate the manual or automatic execution of models based on certain parameters. Although they have overlaps with **Autosubmit**, they all have different characteristics [Dir] and fulfil different functions:

- The **ScriptEngine** developed at the Swedish Meteorological and Hydrological Institute (SMHI) primarily for the EC-Earth community. Which uses YAML and python methods, but defines a Domain Specific Language which is then used to describe commands and simple routines.
- The **Modular Earth Submodel System**, MESSy. Which is more infrastructure oriented and incorporates ready-to-use implementations for most steps.
- The **Modular System for Shelves and Coasts**, MOSSCO. Which is focused on coastal modelling and applications, and provides an excellent modular system for coupling models and applications.
- The **Make Experiments tools**, Mkexp. Which is developed by the Max-Planck Institute for Meteorology, focuses on generating run scripts and post-processing scripts for ESM's. Although having many features, it generates pretty lengthy unintuitive shells scripts.
- The **Earth System Modeling Framework**, ESMF, provides a standard software platform with different structures for interconnecting model components. Although being generally generic, some couplings may require code adaptation.
- The **Earth System Modelling Tools**, ESM-tools, allows interconnecting different model components elegantly. It makes use of YAML configuration files, which include settings for a wide range of ESM components. It also allows a great amount of user customization through specific name-list changing commands

## HPC Platforms

The Climate DT will run on two Pre-Exascale Platforms from EuroHPC JU; On LUMI (Large Unified Modern Infrastructure), which is a petascale supercomputer located at the CSC (Finnish IT center for science) data center in Kajaani, Finland and MareNostrum 5, another preexascale supercomputer hosted by the BSC (Barcelona Supercomputing Center).

Apart from these two HPC Platforms, during the initial phases of the project development, other platforms have been used; Mainly, MareNostrum4 also hosted at BSC, Levante hosted at the DKRZ (German Climate Computing Center) and finally, Juwels from JSC (Jülich Supercomputing Center).

## LUMI

LUMI is one of the three European pre-exascale supercomputers. It's an HPE Cray EX supercomputer consisting of several hardware partitions targeted for different use cases. All the hardware partitions are connected via an HPE Slingshot 11 high-speed interconnect. As of 06/2023, LUMI ranks third on the top500.org list[\[Top\]](#).

Its hardware partitions are LUMI-C, which is the CPU partition, LUMI-G or the GPU partition and LUMI-{D,P,F} which include large memory GPU nodes and are more oriented to big data analytics and processing. The main hardware partitions that will be used during the project's development will be:

**The LUMI-C** hardware partition consists of 1536 dual-socket CPU nodes with 64-core 3rd-generation AMD EPYC™ CPUs, and between 256 GB and 1024 GB of memory. Each core has support for 2-way simultaneous multithreading (SMT) allowing for up to 256 threads per node. The 1536 CPU Nodes have an aggregated LINPACK performance of 5.63 petaflops.

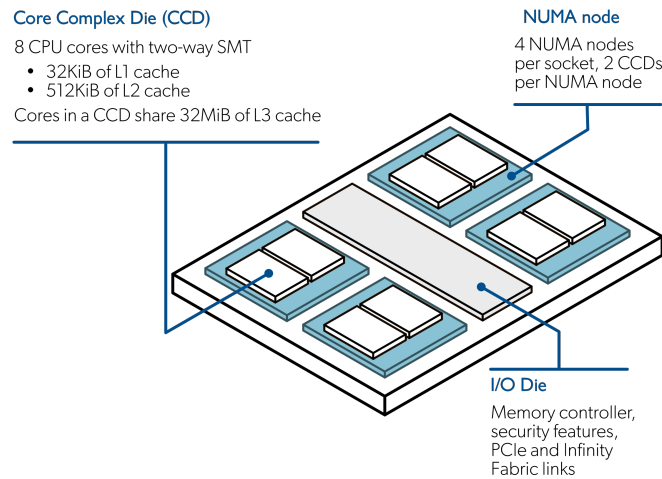


Figure 6: Lumi C node structure

**The LUMI-G** hardware partition consists of a total of 2560 nodes with 4 AMD MI250x GPUs and a single AMD EPYC "Trento" CPU. A MI250x GPU is a multi-chip module (MCM) with two GPU dies (GCD). Each GCD has a total of 110 compute units which have access to a 64GB, which means that each MI250x module has a total of 220 CUs and 128 GB of total memory. The CPU is configured as 4 NUMA nodes (NPS4), i.e., 128 GiB of DDR4 memory per NUMA node for a total of 512 GiB CPU memory.

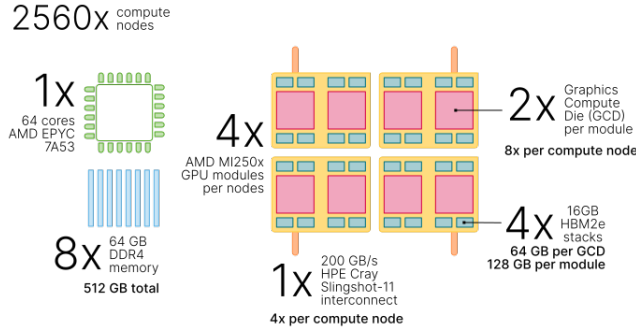


Figure 7: Lumi C node structure

Regarding networking, all LUMI compute nodes use HPE Cray Slingshot-11 200 Gbps network interconnect (NIC). The LUMI-C (CPU nodes) are equipped with single endpoints while the LUMI-G nodes (GPU nodes) have 4 endpoints - one for each AMD MI250x GPU module. Each endpoint provides up to 50 GB/s of bidirectional bandwidth.

Finally, in order to handle software environments, LUMI uses the Lmod Module environment. With it, users can load any software they need and change the environment variables accordingly. It also allows better maintaining, as system managers can modify the environment as they wish and make it available to everyone else in an easy manner.

## Marenostrum 5

Marenostrum 5 will have an aggregated peak performance of 314 petaflops and aggregated Linpack performance of 205 petaflops. It will feature Nvidia's H100 Tensor Core GPUs and its Quantum-2 (aka NDR) InfiniBand networking. The system will also include Nvidia's Arm-based Grace CPU Super-chips, which feature tightly paired CPUs.

Marenostrum 5 will provide a storage of 248 PB net capacity based on SSD/Flash and hard disks. It will have an aggregated performance of 1.2 TB/s on writes and a 1.6 TB/s on reads. It will also include a long-term archive storage based on tapes of around 402 PB.

As of July of 2023 there isn't much available information on the Marenostrum5 architecture and structure. This information will likely be made public when MN5 is fully operative.

### 1.2.3 Justification of choice

After going through some of the most widely used tools used by workflow developers in computational earth sciences, this section explains and summarizes some of the reasons which have driven the current structure and strategy with regard to the climate simulation workflow.

One of the decisive points has been the existing knowledge and experience with the workflow for EC-Earth, which is a noticeable production workflow at the BSC. Taking into consideration its strengths and limitations, the workflow for the Climate DT has been developed by correcting some deficiencies the EC-Earth workflow has; increasing modularity, portability, configuration versioning and customization of the workflow configuration.

On the other hand, one of its strengths has been that the generic structure complies with most requirements for the ClimateDT project, this is likely because the workflow has been developed with Autosubmit in mind.

Regarding the workflow manager, Autosubmit, is able to run workflows on most HPC platforms while also providing features that usually an experiment manager would handle. Therefore, it provides a great foundation on which the workflow can be built upon.

## 1.3 Scope

### 1.3.1 Objectives

Although briefly introduced in the problem description section, in a more summarized way, the workflow solution will and should fulfil the following aspects :

**Be able to run the ICON model efficiently.** The workflow solution has to be adapted to the ICON model. Although the final workflow uses pre-compiled binaries, it should be able to compile the model, at least during development stages when optimizations are constantly applied to the models.

**Create a portable workflow, being able to run on various HPC platforms like LUMI or MN4/MN5.** This objective is central in the workflow development. Although connection handling with the various HPC Platforms is done by the workflow manager, all the central workflow steps have to be designed and implemented in a platform-agnostic way. All platform specifics have to be grouped in libraries that are then used depending on the platform where the workflow is being run.

**Allow running various types of simulations. Therefore, allowing a great amount of personalization.** This objective translates into parametrizing as much as possible all the workflow steps. In this way, users can configure their experiment and the workflow, with help of the workflow manager, and therefore change the model and simulation type accordingly.

**Periodically test various instances of the workflow with different configurations** The workflow solution includes a testing suite which periodically tests representative cases of the different workflow variations. This will likely be automatized and resemble a CI/CD pipeline.

**Allow great amounts of modularization in regard to base configuration, experiment definition and coupling with other essential components of the workflow.** The workflow solution has to be implemented in a modular way, so it allows users to use flexible and simple configuration files, to decide which model base configuration to use, which type of simulation to run, etc...

### 1.3.2 Risks

The work that is being developed in this thesis is dependent on many components and work-groups. The main dependencies are with the developments being made around the models, specially the ICON model.

Therefore, the main risks that could arise are closely related to model developments. These could be, difficulties when optimizing the model, difficulties when porting the model to other platforms, as usually many dependencies have to be met. And finally, consumption of all the available HPC resources.

The direct results of the issues I was commenting can be grouped in the following general risks :

- **Development time delays:** The ClimateDT project includes the work of many Work Packages, therefore work must be synchronized properly in order to converge properly. Issues in other interdependent work packages would likely translate to time-delays, and that will propagate through the dependant groups.
- **Insufficient HPC Resources:** Even though the HPC resources that the project will use have closely been analysed and quantified in the ClimateDT project proposal, unforeseen events that arise practically could likely increase HPC resource consumption.

## 2 Software and System requirements

In this section, a brief description of the system requirements will be given. The complete workflow interfaces mainly with LUMI and Marenosturm 5. Although during the initial phase of the project, specially in the more research driven phase, developments have also been made using Marenosturm 4, Levante [DKR] and Juwels [JSC].

Apart from the HPC platforms, the workflow also makes use of the ICON model (apart from the IFS based models, NEMO and FESOM, which are being developed by other researchers).

### 2.1 The ICOSahedral Nonhydrostatic Model (ICON)

#### 2.1.1 Description

The ICON (ICOSahedral Nonhydrostatic) modelling framework is a joint project between the Deutscher Wetterdienst (DWD) and the Max-Planck-Institute for Meteorology (MPI-M) for developing a unified next-generation global numerical weather prediction (NWP) and climate modelling system.

The main difference of ICON with other models, is its grid. The ICON grid, is based on icosahedrons, which are projected onto a sphere. The spherical icosahedron consists of 20 equilateral spherical triangles.



Figure 8: Icosahedron projected on a sphere

The edges of each triangle are bisected into equal halves or more generally into  $n$  equal sections. Connecting the new edge points by great circle arcs yields 4 or more generally  $n^2$  spherical triangles within the original triangle. ICON grids are constructed by an initial root division into  $n$  sections ( $R_n$ ) followed by  $k$  bisection steps ( $B_k$ ), resulting in a  $R_n B_k$  grid. Such grids avoid polar singularities of latitude-longitude grids and allow a high uniformity in resolution over the whole sphere.



Figure 9: Construction of an icosahedron grid



ICON is composed of component models for the atmosphere (ICON-A), the ocean (ICON-O) and the land (JSBACH), as well as chemical and biogeochemical cycles, all implemented on the basis of common data structures and sharing the same efficient technical infrastructure.

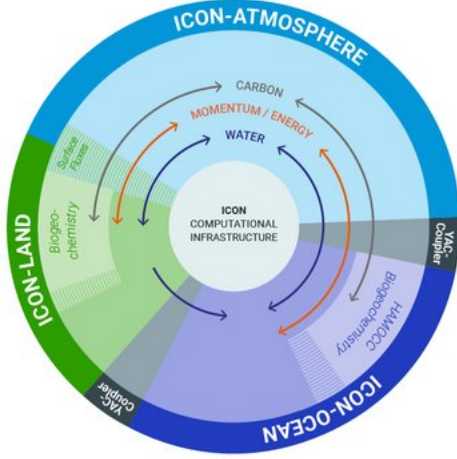


Figure 10: ICON Components (source: [Teaa])

The model coupler which interconnects and allows MPI communications between the different components is the Yet Another Coupler Coupler, the YAC Coupler [MY]. It allows the data exchange – e.g. of energy, momentum, water and important trace gases such as carbon dioxide – between the Earth system model components for ocean, atmosphere and land, and are essential for representing the climate system’s complex processes and feedbacks in the models

**The ICON Atmosphere component** ICON-A is an atmospheric general circulation model in a configuration using the Max Planck Institute physics package, which originates from the ECHAM6 general circulation model [ECHAM6], and has been adapted to account for the changed dynamical core framework. The coupling scheme between dynamics and physics employs a sequential updating by dynamics and physics, and a fixed sequence of the physical processes similar to ECHAM6.

**The ICON Ocean component** ICON-O solves the hydrostatic Boussinesq equations of large-scale ocean dynamics with a free surface, also referred to as the primitive equations of large-scale ocean dynamics [JW]. The ocean primitive equations are an evolution equation for a state vector  $\{v, \eta, \theta, S\}$  that consists of a horizontal velocity field  $v$ , the surface elevation  $\eta$  and the oceanic tracers, potential temperature and salinity  $\{\theta, S\}$ .

**The ICON Land Model** JSBACH (Jena Scheme for Biosphere-Atmosphere Coupling in Hamburg) [JSBACH] is a land surface model developed by the Max Planck Institute for Meteorology in Hamburg, Germany. It is able to simulate various processes occurring in the terrestrial biosphere, including vegetation dynamics, energy and water fluxes, carbon and nutrient cycling, and interactions with the atmosphere.

### 2.1.2 Installation

The process of building ICON consists of two parts: Defining, modifying and setting different model options in a wrapper configuration script (which produces a MakeFile), and then building the sources through make.

The configuration step is normally done by running the `./configure` script with command-line arguments, which, among other things, tells the script where to locate libraries and tools required for building. This configure script, essentially, makes use of GNU Autoconf to generate a MakeFile. It also includes an extensive set of low level tests that check the different Fortran model components for any incompatibilities.

As seen in Figure 11, The ICON model has many compulsory dependencies.

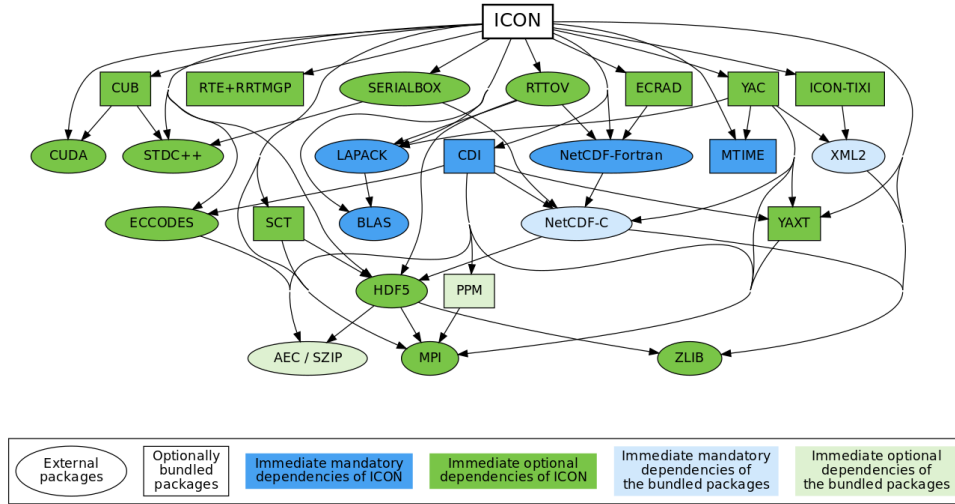


Figure 11: ICON Dependencies

The configure script makes sure that all these dependencies are properly configured and are compatible between each other. The list of arguments enabling a successful configuration might be quite long and difficult to compose, therefore, instead of running the generic `./configure` script directly, users are recommended to execute a corresponding platform- or machine-specific configuration wrapper that sets the required compiler and linker flags as well as the recommended set of configure options.

These configure wrappers allow over-writing the platform/machine agnostic base configure script with all the user and platform specific configurations. It mostly sets the different software dependency paths or in case of having a environment module system like Lmod, or similar, to load the correct modules.

Once the tests are done and the MakeFile has been generated, to build the ICON binaries, users simply have to access the root directory where the MakeFile is located, and run make.

## 2.2 Workflow manager environment

`Autosubmit` has been designed with collaboration in mind, it allows users to develop and execute experiments, but also, to share and monitor them between colleagues. Although `Autosubmit` is able to run experiments in most HPC Platforms, for simplicity, what is usually done, is to create a common installation between all members of a project. This is the case for the `Climate DT`, where a shared experiment database has been made available in a Virtual Machine to all researchers and engineers.

This Virtual Machine can access the main HPC Platforms used by the workflows through authorized hosts.

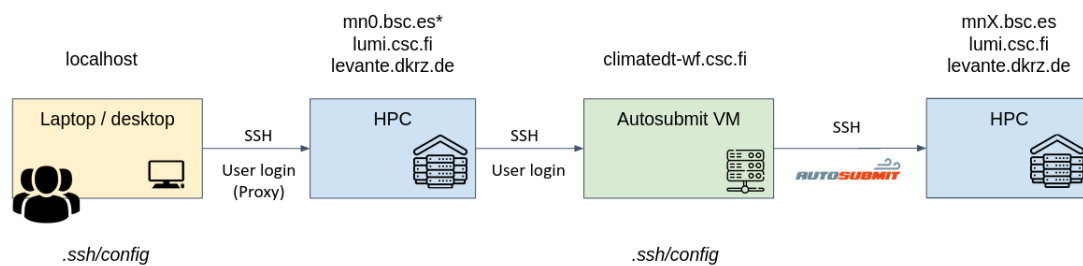


Figure 12: Autosubmit VM connection steps

Once all the configuration steps are done, and all the SSH keys are properly stored. Users are available to log in from their local computer and submit jobs to any of the available HPC Platforms that have been configured, mainly LUMI and Marenostrum 4/5.

### 3 Temporal planning

The work produced in this thesis is framed and will be used in the **Climate DT**, due to this, there are some predefined project timings which have to be respected. Although, having a predefined temporal planning, I've expanded on it.

To have a general temporal sense of this thesis, I'll start by reviewing the amount of hours to be invested into the project.

The thesis amounts to a total of 18 credits, ECTS. As each credit theoretically amounts to 25-30h of work, the total hours of work that have to be destined to it will be between 450-540h. As I'm working full time on this project, I should cover the amount of hours in 11-13 work weeks, or 3 months more or less. Therefore, ending the project between the end of May and the start of June.

Even though having an underlying project plan and organization, I will granularize my future work even further, having as reference points the different milestones setted in the original planning.

#### 3.1 Existing project planning

The way in which progress is tracked, is via **Deliverables** and **Milestones** 19, these include a brief description of what should be included, but are generic enough to allow variability and internal planning.

The **Deliverables** and **Milestones** that have been assigned to the work package on **Workflow of the climate simulations** are the following:

Milestones and Deliverables	
Deliverables	Due Date
Workflow specification and structure	December 2022
First version of the workflow	April 2023
Intermediate version of the workflow	October 2023
Final version of the workflow	April 2024

Figure 13: WP6 Important Deliverables

The first Deliverable that I worked on described in a more general sense the structure of the workflow. Which steps it would have, what tasks would they fulfil and how.

It didn't specify exactly how each task would be implemented, but it set a common ground for starting development.

The second Deliverable, includes a first prototype implementation of the workflow. This includes running actual simulations for the **ICON**, **IFS-Nemo** and **IFS Fesom** models, apart from also cloning the models, transferring them to the HPC Platforms and compiling them.

#### 3.2 Task description

Developments on my thesis, apart from the one developed in the **GEP** course (Gestió de Projectes) has been done in the context of the second and third Deliverable, which provides a First and Intermediate version of the climate Simulation Workflow.

Therefore the base ground for planning tasks, has been to use what was expected to be reached in the third Deliverable.

This deliverable should achieve the following:

- Run all models in a unique workflow.
- Run in a test configuration.
- Run the workflow in LUMI and MN4.
- Run different kinds of simulations.

Taking the work being done in the GEP course and these Deliverable achievements as start-points, the following tasks that have to be developed during this thesis are.

**[T1] Introduction, Context, and Reach** This task includes the definition and scope of the project and the tasks related to it, which are mostly the redaction of the initial drafts of this document. Although my thesis is already included into another project, defining a common ground and the scope of it for contextualization is still necessary.

**[T2] Temporal planning** The tasks related to this section include task identification, task grouping and analysis of the dependencies between them. As commented before, there exists an underlying project organization and planning, therefore this task expands on it.

**[T3] Economic management and Sustainability** The scope of this task is to identify and estimate the types and amount of resources that are needed during the development of this project. By doing so, it will also analyse the sustainability aspects of it.

**[T4] Documentation** This task will be developed during most of the span of the project. Its purpose is to generate documentation, likely by expanding this current document with the progress being made. This task includes working on the final document as well as working on the presentation for the defence of my thesis.

This task includes the work that is dedicated to the writing and reviewing of the Deliverables that have to be handed out in order to report the work done. In this case, it includes the time dedicated to write the deliverable D340.6.2.1: **First version of the climate simulation workflow**.

These following tasks are more directly linked to the already existing work I've done with regard to the Climate Simulation Workflow. It should be noted, that there has been a great amount of previous work done, which included a great amount of learning, as well as the work in previous Deliverables which weren't fully related to the work being done now, but which helped in having a general view on the project.

**[T5] Developments for the Deliverable D340.6.2.1** This task is used to account for the developments that have been done on the actual workflow during the period of the GEP course. These developments set an initial stage for the future developments, therefore it depends with most other tasks.

**[T6] Further parametrization of the workflow steps** The scope of this task is to adapt the workflow by reducing all the variables that are currently set in the workflow or "hardcoded" and make use of Autosubmit placeholders that automatically parse them and fill them with the previously defined values.

This task is very dependent on the variable definition and likely continues during future developments of the project. In any case, for the scope of this thesis, it is limited to allowing the development configuration and also, if provided by the other work packages, the definitions for test, spin up and historical simulations.

**[T7] Configuration and simulation specifics** This task consists in analysing and defining the variables for the ICON model workflow with regard to the simulation configuration, and the base configurations for the model.

Some of these specifics have been previously defined and are already being parametrized into the workflow, but most configurations are being studied by other work packages and are a work in progress.

**[T8] Treatment of Output and Input** The workflow solution has to handle model output using the FDB, think of it as the ClimateDT database, and **GSV** (Generic State Vector), which defines the output format, therefore specifying output variables like the grid type or general model variables like vertical pressure levels and so on.

The scope of this task, is to make use of the FDB interface and include in the workflow the necessary calls and steps in order to pass model output.

In this sense, most work focuses on studying and analysing how to set the model to output data in the necessary way. This includes parametrizing general output variables with Autosubmit.

The same has to be done with regard to model input. In order to handle model restarts and input data reading in the most optimum way.

**[T9] Development of the testing suite** In order to improve the quality and robustness of the development cycle with regard to the workflow, a testing suite has to be included which automatically test the latest modifications done to the workflow code.

There already exists a Python based tool, which was created in the BSC, to test other ESM workflows like auto-ECearth. This tool interfaces with the Autosubmit workflow manager and allows users to use most of the configurations and commands that Autosubmit has.

In any case, to be able to use the testing suite tool with the ClimateDT workflow, many modifications have to be made. The current testing suite is only compatible with the older versions of Autosubmit, 3.14 and lower.

Therefore, the testing suite tool has to be adapted to the recent version of Autosubmit, and then the most representative and important workflow variations have to be defined, in order to be tested regularly.

### 3.3 Resources

In this section, I'll describe in a general manner the resources that are likely going to be used in the different tasks. To ease comprehension and accounting, the resources will be separated into human resources, and material resources.

### 3.3.1 Human Resources

As human resources, I've identified two main figures which should be able to fulfil the tasks that I've described, these figures are the following;

- **A Developer (DEV)** Which would fulfil most work related to implementing the actual workflow solution and then test it. In this project, most tasks which aren't straightly related to the **GEP** subject would be fulfilled by the Developer.
- **A team lead (TL)** This figure should focus on project planning and organization, make sure all requirements are fulfilled. Most tasks that are done in the **GEP** course could be done by this figure.

### 3.3.2 Material Resources

The main material resources that are used are the **HPC Platforms (HPCP)** and my **Work Computer (PC)**. These are the resources that are more expensive and tangible.

Other resources that could be considered could be the proprietary software that has to be used during the development of this project.

### Summary Table

Task number	Resources	Hours	Dependencies
Task 1	PC, TL	14h	T4
Task 2	PC, TL	8h	T4
Task 3	PC, TL	8h	T4, T2
Task 4	PC, TL	20h	T2
Task 5	PC, DEV, HPCP	100h	T4, T2
Task 6	PC, DEV, TL, HPCP	150h	T2, T4, T5
Task 7	PC, DEV, TL, HPCP	110h	T2, T4, T5, T6
Task 8	PC, DEV, TL, HPCP	110h	T2, T4, T5, T6, T7
Task 9	PC, DEV, TL, HPCP	130h	T2, T4, T5, T6, T7

Figure 14: Task requirements and dependencies

Therefore, working a total of **650h** considering all the tasks.

Although most of the tasks developed during the scope of this project are interdependent, a dependency graph between the different tasks could be the following 15:

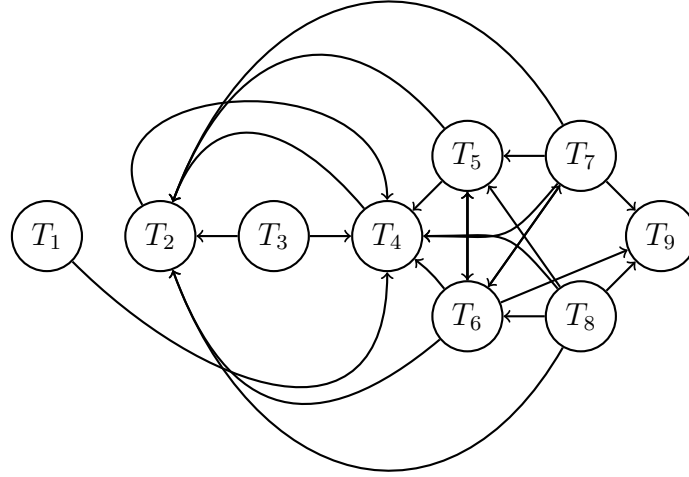


Figure 15: Dependency graph (created with `tikz`)

### 3.4 Methodology and status reporting

The development methodology that is used in the context of the ClimateDT is the Agile methodology, which has been adapted to the needs and nature of research. This methodology was the one proposed by the technical coordinator in the early stages of the project.

The way we organize and report the status of our work in the WP6: **Workflow of the climate simulations** is by meeting in a bi-weekly manner. Therefore, possible coding sprints could be done using the same time steps.

As we're meeting in a bi-weekly manner, having two-week long coding sprints could be a good sprint length in order to develop the project correctly.

Bi-weekly Sprints		
Sprint number	Tasks	Time period
Sprint 1	T1, T2, T4, T5	20 Feb – 6 March
Sprint 2	T3, T4, T5	6 March - 20 March
Sprint 3	T6, T4	20 March - 3 April
Sprint 4	T6, T4	3 April - 17 April
Sprint 5	T7, T4	17 April - 1 May
Sprint 6	T7, T4	1 May - 15 May
Sprint 7	T8, T4	15 May - 29 May
Sprint 8	T8, T4	29 May - 12 June
Sprint 9	T8, T4	12 June - 26 June
Sprint 10	T8, T4	26 June - 10 July
Sprint 11	T9, T4	07 Aug - 21 Aug
Sprint 12	T9, T4	21 Aug - 04 Sept

Figure 16: Possible sprint configuration

This is a proposal 16 of how work should develop.



3.5 Gantt chart

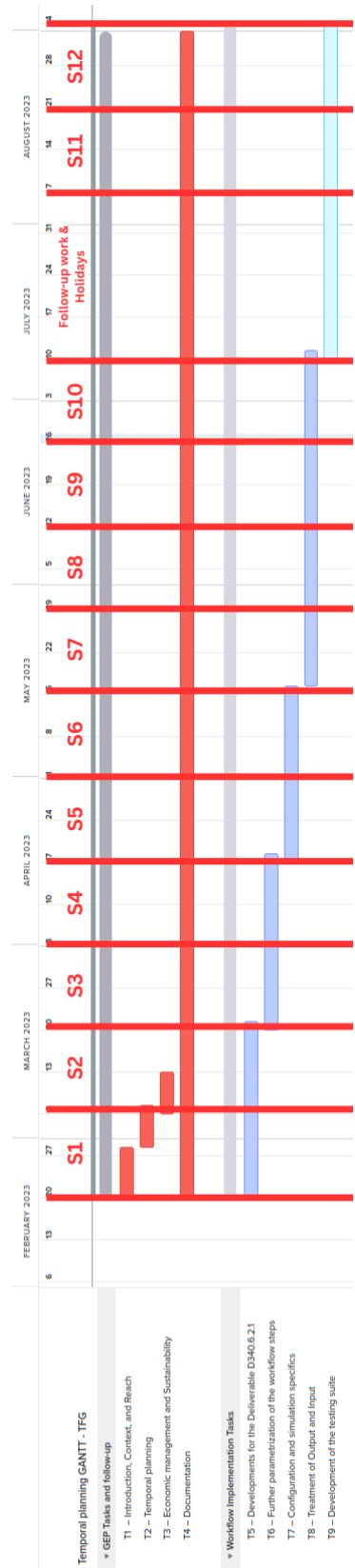


Figure 17: Gantt chart (created with teamgantt)

## 4 Economic Management

This section provides a comprehensive analysis of the resources needed, including human resources, equipment, infrastructure, and materials for successfully developing this project. These resources can be grouped in the following manner, personnel costs, generic costs, contingencies and finally, unforeseen events.

### 4.1 Personnel costs

To estimate the cost of personnel, I will first identify the main actors in the development of the project. In the case of the developments of the climate simulations workflow, the work is somewhat subdivided in the following manner. There are the main developers for the specifics of each climate model workflow, and then, the team lead, which leads the work package which guides and assists the developers.

Help and collaboration from the other model workflow developers will likely occur further on (specially as our work converges). But most of the workload is independent, therefore I wouldn't estimate the cost of their contributions as far as this project goes.

On that account, the personnel costs will be attributed to a main developer, and a team lead who's worked is focused on work planning, assisting development, etc. . .

In the following table 22, I include an estimate of the total cost per personal. The human resource costs are personally elaborated, they are based on my wages. This cost calculation includes the social security, which is calculated in the following manner:

$$\textit{Social Security} = \textit{Common contingencies} (24, 10\%) + \textit{Professional contingencies} (7, 80\%)$$

Where the shown percentages are taken from the gross salary. Then, the total cost is the result of the addition of the gross salary and the social security contingencies.

$$\textit{Total} = \textit{Social Security} + \textit{Gross Salary}$$

Position	Gross salary	Social security	<b>Total</b>
Developer	11,5 (€/h)	3,6685 (€/h)	15,1685 (€/h)
Team Lead	20 (€/h)	6,38 (€/h)	26,38 (€/h)

Figure 18: Total personnel costs (Own elaboration)

Having an estimate for the personnel costs, we can now proceed to calculate the CPA (Cost per activity) making use of the Gantt chart that was elaborated in the temporal planning section 3.

## Costs per activity

Task	Developer	Team Lead	Cost
T1	0h	14h	369,32 €
T2	0h	8h	211,04 €
T3	0h	8h	211,04 €
T4	0h	20h	527,6 €
T5	120h	0h	1820,22 €
T6	130h	20h	2499,505 €
T7	80h	30h	2004,88 €
T8	120h	40h	2875,42 €
T9	80h	30h	2004,88 €
Total CPA			
	<b>530</b>	<b>170h</b>	<b>12.523,905€</b>

Figure 19: Total personnel costs

## 4.2 Generic costs

In this section, I will go over the main generic costs. I will try to estimate the costs associated to the environment that is necessary for deploying the Climate Simulation Workflow, this will include, hardware, software, etc.

For the scope of this project, I've considered the following resources.

- **Computing environments**

By Computing environments I mean the different HPC platforms that are going to be used during the developments of my project. The whole Climate DT project has already been assigned a limited amount of CPU/GPU hours apart from the necessary disk space for the model's input and output.

As the main HPC platforms that are going to be used are LUMI and Marenostrum 4, I will estimate the CPU/GPU hours that I expect to use and then calculate the costs associated to them.<sup>1</sup>

Platform	Price CPU/Hour	Price GPU/Hour
LUMI	0.008 €	0.35 €
Marenostrum 4	0.010 €	—

Figure 20: Estimated HPC Core/GPU computing costs (source : [\[Tar\]](#))

The workflow is being developed to be run in MN4 and on LUMI, although that being the case. Most developments are going to be done in LUMI, as MN4 will stop being operative as we transition to Marenostrum 5. Therefore, a good platform usage estimate could be 40% time in MN4 and the rest in LUMI.

The amount of cores that the testing will require will vary as we test different configurations and model resolutions. Some tests have been done in the context of the

---

<sup>1</sup>The computing costs for Marenostrum4 have been extrapolated from the information obtained for the LUMI supercomputer. Information regarding these estimates doesn't seem to be publicly shared.

DYAMOND project [DCa], and it seems that for the configurations used in the ClimateDT on the ICON model, a good ceiling for the number of cores is around 10000.

Taking into consideration a 1:4 ratio of development and testing, we amount to a total of 132,5 h from the 530h destined to development.

In conclusion, we achieve a computing cost of :

$$\begin{aligned}
 \textit{Total Computing Costs} &= (10000 \textit{ cores} * 132,5h * 0.4 \textit{ (BSC)} * 0.010 \textit{ Euro/Core/hour}) \\
 &+ (10000 \textit{ cores} * 132,5h * 0.6 \textit{ (LUMI)} * 0.008 \textit{ Euro/Core/hour}) \\
 \textit{Total Computing Costs} &= 5300 + 6360 \\
 \textit{Total Computing Costs} &= 11660 \textit{Eur}
 \end{aligned}$$

The total use of computational resources could amount to **11.660 €**.

- **Personal hardware**

In this case, I will include the price of the computer that I'm currently using, and will most likely be used during the development of this project.

The computer model is the following, *Dell Latitude 5420, Intel Core i5-1135G7, 16 GB DDR4, 512GB SSD* with a total cost of **1.177,39 €**.

If we suppose a median lifespan of 60 months (5 years), then the amortization of this cost is of :

$$\begin{aligned}
 \textit{Total amortization} &= (12 \textit{ months}/60 \textit{ months}) * (1.177,39 \textit{ Eur}) \\
 \textit{Total amortization} &= 235,478 \textit{ Eur}
 \end{aligned}$$

Amounting to a total of **235,478 €**

- **Proprietary software**

Finally, although much of the software that is being used is free, we are still making use of some proprietary software in the context of this project, mostly for document writing.

We make use of Microsoft Word for writing documents in a collaborative way. We do this in virtual machines where Windows is installed, therefore the added costs if we buy a one time licence for Windows 10 and Microsoft 365 are:

Software	Price
Windows 10	145 €
Microsoft 365	149 €
Total	
	<b>294 €</b>

Figure 21: Proprietary software costs (source : [Win] & [Mic])

## Total Generic costs

Type	Price
Computing environments	11.660 €
Personal Hardware	235,478 €
Proprietary software	294 €
Total	
	<b>12.189,478 €</b>

Figure 22: Total generic costs

### 4.3 Contingencies

In this section, I will discuss and include a margin for unexpected cases or situations that could increment costs. Usually in software development projects, the percentage that is destined for contingency measures oscillates between 10-30

As this project is research oriented, I think a contingency percentage that could fit the case could be **20%**. Therefore, amounting at a total of:

$$\text{Contingency Cost} = (\text{CPA Costs} + \text{Generic Costs}) * 0.20$$

$$\text{Contingency Cost} = 16356,854 * 0,20$$

$$\text{Contingency Cost} = 3271,3708\text{Eur}$$

Therefore, considering the actual cost estimation, a sensible contingency cost should be approximately **3452,0528 €**.

### 4.4 Unforeseen events

These section will discuss possible events that were not foreseen during project planning. The idea is to estimate and predict these events and calculate how they could increment the cost of the project development.

Considering the existing project planning of the Climate DT project, the only possible events that I think could increase costs, could be:

- **Development time delays:** The ClimateDT project includes the work of many Work Packages, therefore work must be synchronized properly in order to converge properly. In view of how the project is developing currently, I think there could of up to a maximum of 30h for the Climate Simulations Workflow.
- **Insufficient HPC Resources:** This case describes the event where the HPC resources of LUMI and/or the Marenostrum 4 are insufficient for the CLimateDT project. I think a ceiling of 10% of added HPC resources is a good estimate in this case.

Therefore, amounting to:

$$\begin{aligned}
\text{Development time delays} &= \text{Dev Total Salary} * 30h \\
\text{Development time delays} &= 15,1685/h * 30h \\
\text{Development time delays} &= 455,055\text{Eur}
\end{aligned}$$

$$\begin{aligned}
\text{Insufficient HPC Resources} &= \text{Total Computing Env} * 0.1 \\
\text{Insufficient HPC Resources} &= 6.600 * 0.1 \\
\text{Insufficient HPC Resources} &= 660\text{Eur}
\end{aligned}$$

In total, considering the insufficient HPC Resource case and the development time delays case, unforeseen events could increment the cost of the project by a total of **455,055 € + 660 € = 1115,05€**

## 4.5 Total Costs

In the table 23 we can find the total amount of costs that this project could produce.

Cost Type	Amount
CPA	12.523,905 €
GC	12.189,478 €
Contingencies	3.271,3708 €
Unforeseen events	1.115,05 €
Total	
	<b>29.099,8038 €</b>

Figure 23: Total costs

## 4.6 Managment Control

In this section, I will discuss how I will keep track of the work development in order to correct deviations in an attempt to reduce added costs. In this sense, the idea is to keep revising the estimated costs and actualize them if there are any divergences from the initial planning.

In any case, the estimations for the unforeseen events and possible contingencies have been calculated with a great margin. In order to parametrize the possible deviations, I will use the following formulas:

$$\begin{aligned}
\text{Cost Deviation} &= (EC - RC) \cdot RHC \\
\text{Cosumption Deviation} &= (EHC - RHC) \cdot EC
\end{aligned}$$

Where  $EC$  and  $RC$  represent the estimated cost and the real cost, respectively. And  $EHC$  represents the estimated hour consumption and  $RHC$  represents the real hour consumption.

## 5 Project planning deviations

The main deviations that have occurred during the development of the project have been caused by delays in the readiness of the climate models to perform climate simulations. The development of the workflow, is strictly dependent on the availability of the ICON model, the setbacks in its readiness, have influenced and propagated into the workflow development.

In this case, the main issues which conditioned the development of the ICON workflow have been adapting the ICON model to compile in the environment used in the LUMI HPC Platform, Cray. Finally, additional delays stacked up while porting the atmospheric component of the model to GPU's.

More directly related to the workflow, the initial plans for incorporating the ICON model run scripts changed considerably, as the initial run script format was changed long due in the project, having to adapt the existing workflow to make it compatible with the new run script structure.

These delays have added considerable overheads in refining of the workflow. As only low resolution, short term simulations have been used and tested thoroughly. And not the actual high resolution development and production configurations.

Finally, it is important to mention, that the availability of computing resources has also induced a significant delay in the readiness of the models, and subsequently in the development of the workflow. This was considered as a high risk factor in the calculation of costs in the first document of this thesis, which was done during the GEP course.

### 5.1 Redefinition of tasks

Due to the delays and changes in the original project planning, the tasks that we're identified initially changed considerably. The tasks that we're more affected were from task *T6* to *T9*.

**Redefinition of task T6: *Further parametrization of the workflow steps*** The initial idea for this task was to *adapt the workflow by reducing all the variables that are currently setted in the workflow or "hardcoded" and make use of Autosubmit placeholders that will automatically parse them and fill them with the previously defined values.*

This task has been under development during the whole span of the project, as changes in the models and other aspects of the workflow have been constant. Considering the evolution of the work that has been done, this task should be redefined to: **Adapting the ICON workflow to include the new simulation runscripts**, which is more granular and has been the source of most workflow parametrizations.

**Redefinition of task T7: *Configuration and simulation specifics*** Conditioned by the overheads in the climate readiness, the *definition of the variables for the ICON model with regard to the simulation configuration, and the base configurations for the model* were delayed considerably.

In any case, as agreed with developers of ICON of the Max Planck Institute for Meteorology, the run scripts that were provided included some base configurations for the development and production runs. Therefore, only "minor" modifications and name list variables should change considerably.

**Redefinition of task T8: *Treatment of Output and Input*** As for the treatment of output and input in the workflow, due to the propagation of the delays that were specified earlier, no considerable changes have been made. Therefore, shifting these associated tasks out of the scope of this project. The only considerable changes that have been done regarding the input and output, have been reorganizing the input file structure into a more homogenous one.

**Redefinition of task T9: *Development of the testing suite*** Regarding the development of the testing suite, it was assigned to a new position that was incorporated lately in the development of the DestinE project. As most aspects of it were strictly related to the Autosubmit workflow manager, the initial learning curve was challenging, if the development of the ICON workflow continued parallelly.

## 5.2 Estimated additional costs

During the planning of the economic management, in the GEP course. The unforeseen or risks that were identified were the following ones (These are also described in more detail in 1.3.2) :

- **Development time delays:** The ClimateDT project includes the work of many Work Packages, therefore work must be synchronized properly in order to converge properly. In view of how the project is developing currently, I think there could of up to a maximum of 30h for the Climate Simulations Workflow.
- **Insufficient HPC Resources:** This case describes the event where the HPC resources of LUMI and/or the Marenostrum 4 are insufficient for the CLimateDT project. I think a ceiling of 10% of added HPC resources is a good estimate in this case.

Both events have been given during the development of the project. In any case, the associated costs to these unforeseen events, were taken into consideration with a relative high margin, therefore not inducing more unplanned costs.

In total, considering the insufficient HPC Resource case and the development time delays case, unforeseen events could increment the cost of the project by a total of **455.055 € + 660 € = 1115.05€**



## 6 Implementation

This section describes the development steps, including preliminary tests that have been done up to current dates in regard to the workflow. As these developments have led to results reported in different project deliverables, I'll use them as development milestones. The workflow structure and implementation, have been evolving during all stages of development, therefore some definitions or features will likely change in future versions of the workflow.

### 6.1 Workflow specification and structure

During the initial phases of the project, an initial workflow structure was created to start planning the work, enable discussion and ease collaboration.

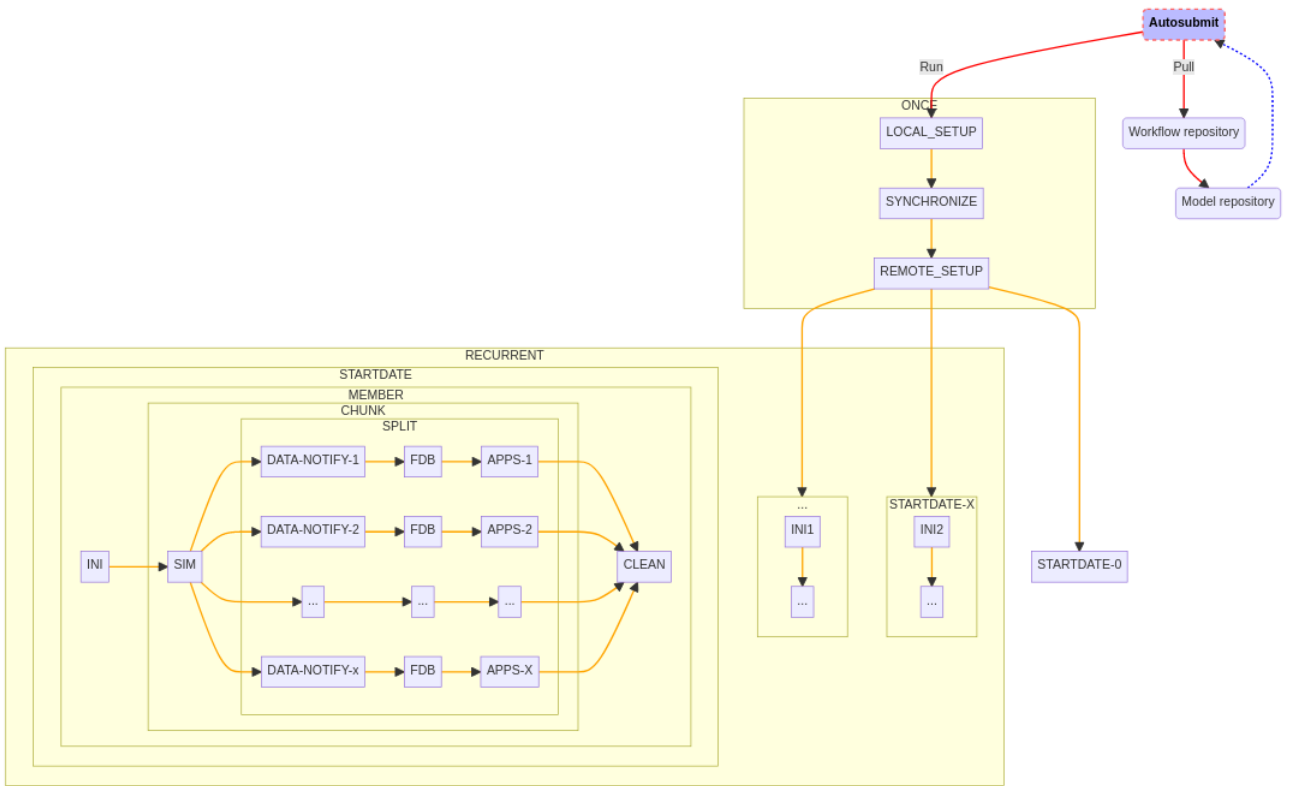


Figure 24: Preliminary general workflow structure

The workflow is split into two main kind of tasks, the tasks that only have to be run once, these being `LOCAL_SETUP`, `SYNCHRONIZATION`, `REMOTE_SETUP` and other mantainment tasks, like cleaning up unused output files. And on the other hand, there are the tasks that are more closely related to the actual realization of the climate simulations, the recurrent tasks.

Before explaining the tasks, I'll explain top down the job hierarchy:

- **Startdate** is the first aggregation level in the Autosubmit workflow hierarchy. It is used to group different instances of the simulation cycle only distinguished by the date of the initial conditions. It has as many elements as start dates in the configuration.

- **Member** is second aggregation level in the Autosubmit workflow. It is used to group different instances of the simulation cycle only distinguished by relatively small variations (called perturbations) in the initial conditions. It will have as many elements as members, per start date.

Actually, ensemble simulations use sets of members to have an overall better approximation to actual observations. This is basically done by joining the simulation output through means and other statistical methods.

- **Chunks** are the third aggregation level. They basically allow long simulations to be split into shorter segments. This allows having a more modular workflow, which improves the workflows resilience when using an unstable environment. Each chunk has a start date and an end date, these two are always included in the bigger original start and end dates of the whole simulation.
- **Splits** Lowest aggregation level in the Autosubmit workflow hierarchy. They are used when the work executed at a chunk level needs further subdivision. Splits have no dependencies between them.

Finally, the tasks that are included in this recurrent block, are **INI**, **SIM**, **DATA-NOTIFIER**, **FDB**, **APPS** and finally **CLEAN**.

The later tasks, from the **SIM** step (not included) onwards, have been and are being developed by other colleagues at the **BSC**. This part of the workflow, has an added layer of complexity, as the applications will have to feed from model output in a streamed fashion.

This division of the general workflow, separating the core climate simulation part of the workflow, from the coupling and processing of the output with applications, was done to modularize development and to allow greater flexibility. The final outcome of the workflow will merge these two sections, allowing to test the complete flow of information through all workflow steps and states.

### 6.1.1 Workflow task design

An important feature for the Climate DT workflow design and structure is interoperability. To achieve model and platform abstraction in the workflow, a set of shell scripts were developed as a backbone for the workflow structure.

These scripts do not call the models and applications directly, but instead, they are templates for the automatically generated scripts that the workflow tasks executes. They are written in shell script and differ from the workflow scripts in the sense that they contain placeholders that are filled with the workflow parameters by the workflow manager.

## Listing 1: Workflow code design example

```
#!/bin/bash
set -xv

#####
This is a template for deploying DT propagators in any cluster
#####

# Interface
HPCARCH=%HPCARCH%
SCRATCH_DIR=%SCRATCH_DIR%
HPCPROJ=%HPCPROJ%
HPCUSER=%HPCUSER%

# Include
. ${PROJDIR}/lib/${CURRENT_ARCH}/configure.sh

# Main code
STAMP=$(date +%Y_%m_%d_%H_%M)

# Load NEMO configuration
get_configuration_nemo
# Load dependencies
load_modules
# Compile
compile
```

The templates include function calls to general functions that are implemented in purpose libraries, including lower-level code, as seen in the code block on the left.

Different versions of the functions are provided for the different platforms which, using overloading principles, are not tightly coupled with the workflow manager or workflow format.

### 6.1.2 Workflow tasks

**Local Setup** The primary goal of this task is to prepare the necessary workflow files such as configuration, platform dependent settings, and other relevant files before the execution. All workflow files (including workflow libraries, model, and application code) will be tar'ed to be uploaded for the first time to the remote platforms.

**Synchronize** This part of the workflow is responsible for uploading the files required for executing the workflow in the remote platform. These files include everything needed for models, applications, and libraries, and are usually uploaded to the scratch partition of the remote platform. The first time this task is executed, the full set of files are sent in a tarball file. If the same task is executed multiple times (usually during development) then only the files that have changed will be synchronized with the remote platform.

This synchronization task allows for workflow libraries to be synchronized. It also allows model sources to be transferred, which in later development stages may not be necessary as pre-compiled binaries are used.

**Remote Setup** After sending the source and configuration files to the remote platform, it is necessary to have a task to compile binaries and to prepare the necessary files used during the workflow execution. It is important that this task takes into consideration the architecture of the platform, choosing the correct compiler and settings used for the model (e.g. optimizations).

This task can be simplified if the models or applications used have already been compiled in the remote platform. Users are able to choose the pre-compiled binaries and application versions to be used instead.

**Ini** This is the first task after the environment setup stage. This task's purpose is to set up the necessary initial data for the different simulation members. This initialization step usually consist of deploying the initial conditions and necessary files for every member of the simulations.

**Sim** This task runs the actual model simulation. It corresponds to one of the sequential iterations in the simulation cycle. More precisely, this task is a chunk task. It is still being

studied (as it will be subject to subsequent deliverables) the frequency of the model output data and its processing with regard to the GSV, specifically at which time steps the data will be streamed, and which dependencies may arise in the process.

**GSV** This task's main purpose is to preprocess output model data, the Output State Vector, and convert it to the GSV format. It makes use of the MultiIO pipelines. This task will not be necessary once the models are able to write GSV data directly to the FDB.

**Apps** This task relates to communication with the different applications that read the data created by the models. These applications include other models (hydrologic, urban, energy use), the one-pass algorithms, and other applications that are under development for the Climate DT project.

**Clean** Finally, this step might be repeated in multiple stages of the workflow. Usage of similar dedicated steps has been implemented in other climate model workflows like EC-Earth. Its purpose is to clean up unnecessary chunk-related files.

## 6.2 First version of the workflow

In this section, I will describe the incorporations and features that were developed and included in the first delivered version of the workflow. This deliverable included the first release of the workflow, **v1.0.0**, which was made available to the ClimateDT consortium.

This deliverable was an important milestone in the development of the complete workflow. Therefore, most of the features that were developed are likely definitive. The main features that were developed have been; **A new configuration scheme**, **Increased HPC Compatibility** and **Climate DT VM (CSC)**.

### 6.2.1 New configuration scheme

In **Autosubmit 4**, the YAML language was introduced in order to configure experiments. By doing so, it increased the readability and management of the different configuration files.

These latest developments in Autosubmit allowed to provide a configuration scheme that could configure the complexity at stake. There were many possible variations of the workflow depending on the experiment that the model will develop, and therefore many layers and combinations of configurations.

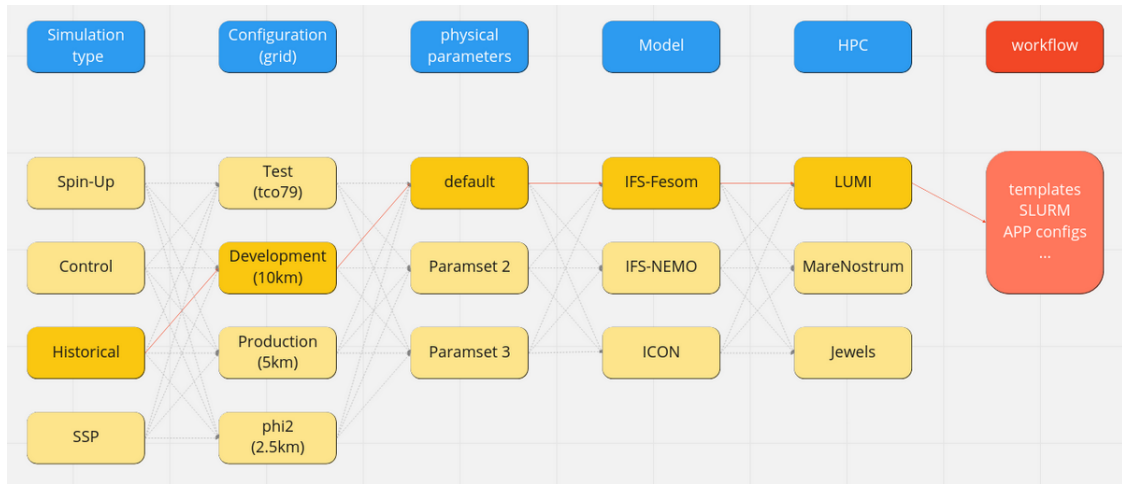


Figure 25: workflow complexity (source: Sebastian Beyer - AWI)

As seen in Figure 25 the configuration scheme had to allow configuration files for each model (textttIFS-Fesom, IFS-Nemo & ICON), for each possible configuration **test**, **development**, **production**, for each platform LUMI, Marenostrum, Levante and other more specific configuration files.

At the same time, we identified that we needed to separate the configuration files in different layers of “interaction”. These layers also somehow reflected the intended “user-cases”, being **workflow/application developers**, **advanced users** & **users**. As seen in the following figure.

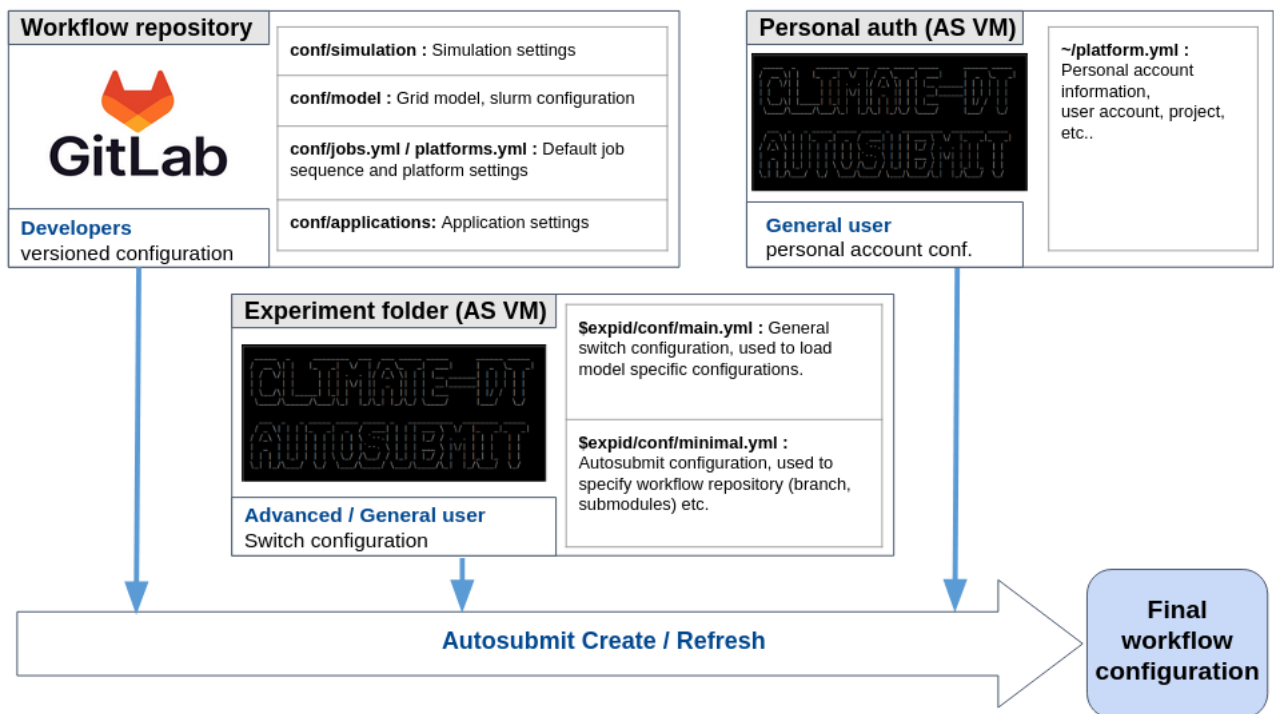


Figure 26: workflow configuration files

In order to define these different layers, we defined a general `main.yml` file which included the switchers, which would then load the desired configuration files.

```
RUN:
  WORKFLOW_TYPE = MODELS
  EXP_TYPE = SPINUP
  GRID = PRODUCTION
  MODEL = IFS-NEMO
```

---

These variables, defined in `main.yml`, (alongside with `minimal.yml`), inform `Autosubmit` about which files and sections it must load, allowing to represent the different variations of the workflow. Additionally, users have the chance to override any parameter in the experiment configuration, inside new or multiple YAML files. Therefore, allowing the different layers of configuration files, usually being the `main.yml` file the entry point for most users, and then deeper in the tree, the more specific configuration files.

---

Listing 3: Extract of `minimal.yml` file example

---

```
- %PROJDIR%/conf/experiments/%RUN.EXP_TYPE%.yaml
- %PROJDIR%/conf/platforms/%DEFAULT.HPCARCH%.yaml
- %PROJDIR%/conf/grids/%RUN.GRID%.yaml
- %PROJDIR%/conf/models/%RUN.MODEL%/%RUN.GRID%.yaml
```

---

As seen in the extract of the minimal file above, through variable interpolation, we allow the workflow manager `Autosubmit` to load the desired configuration files, therefore allowing the multiple configurations of the complete workflow.

## 6.2.2 HPC Compatibility

Once the new configuration scheme was defined, adaptations we're made to run on the LUMI HPC Platform. At the time being, the ICON workflow was only compatible with Marenostrum4. In order to adapt the workflow to the new configuration scheme, changes in the `REMOTE_SETUP` and in the simulation run scripts had to be made.

Firstly, the compilation environment had to be set, these resulted in the creation of new functions which included the loading of the needed modules for the model.

Then for the environment of the runscripts, other functions were added to include the `MPI`, `OPENMPI` environments alongside other needed modules. These efforts basically focused on parametrizing the actual runscripts, and making them as more platform-agnostic as possible.

Finally, as a measure of precaution, the workflow was also made available to run on `Levante`, another HPC platform which ICON developers, used.

## 7 Testing and Key Performance Indicators (KPI's)

In this section, I will describe the process of testing the workflow, including some insight into the use of the testing suite. Additionally, an overview of the Key Performance Indicators (KPI) will be given. The final implementation of the workflow is underdevelopment, therefore, these tests and KPI's reflect performance of the intermediate versions.

The tests done by testing suite and the calculation of the KPI's are done automatically by the testing suite scripts and Autosubmit, the work that has been done is to select a significant set of cases of the ICON workflow that should be tested regularly.

### 7.1 Testing suite

In order to increase the development rate of the workflow, late in the project, a testing suite was created. The use of similar testing suites is a common practice in the Earth Science department at the BSC.

A testing suite includes the most representative cases or variations of the software that has to be tested. In this case, it includes a small set of workflow variations which are needed to be as stable as possible. These cases should be scientifically relevant. Variations could include; changes in the models resolutions, changes in the amount of resources that it uses, changes in the compiler settings or even component variable changes to, for example, run ensemble simulations, etc.

Essentially, the testing suite is a Python tool which interfaces with the Autosubmit API. It submits jobs in a similar manner as a user would do manually, and it exposes some Autosubmit commands ( `create`, `clean`, `run...` ) to be executed in batches.

t0n8 - T255L91-ORCA1L75-LPJG-PISCES-TM5 co2,co2fb				
VERSION	CHUNKS	MODEL_RES	PRODUCTION_EXP	NUMPROC
trunk	3 (12 month)	LR	FALSE	528

OUTCLASS	CMIP6/EC-EARTH-CC/cmip6-experiment-CMIP-piControl
----------	---------------------------------------------------

COMPONENTS	PROCS	INI
IFS	256	a2t4
LIM3	-	a2t4
LPJG	8	a2t4
NEMO	192	a2t4
PISCES	-	a2t4
TM5:CO2	4	-

SYPD	ASYPD	RSYPD	JPSY	CHSY
7.4203	2.4045	2.5383	82643333.0	1707.74

Figure 27: Testing suite example EC-Earth (source: Eric Ferrer - BSC)

As a final result (via the `--report` command), the testing suite provides a report with a brief description of the experiments being run as seen in 27.

## 7.2 Workflow testcases

In order to test the latest developments of the workflow, a small set of scientifically or technically significant workflow variations will be provided. The development of the workflow, and the climate readiness of the models, is something that is under constant change. In the scope of this thesis, due to delays in the model development, the configuration that has been tested in more depth has been the R2B4 (160km) coupled configuration.

In any case, a short temporal list of significant test cases will be provided:

ICON Workflow test cases			
Model version	Resolution	Compilation	Hetjobs
feature-destine-ngc3-wo-hiopy	R2B4	Compilation	None
feature-destine-ngc3-wo-hiopy	R2B4	Precompiled	None
feature-collection-destine-v.0.1	R2B4	Precompiled	None
feature-collection-destine-v.0.1	R2B4	Precompiled	ATM (GPU), OCE (CPU)

Figure 28: Provisional workflow testcases for the testing suite

## 7.3 Workflow KPI's

In order to test and analyse empirically how a workflow performs, key performance indicators (KPI's) are used. The main KPI's used in the evaluation of workflows are the following:

**SYPD** Simulated years per day, measures the model speed by providing the number of years the model could simulate within a 24-hour period, given a certain configuration and computational platform, this case being LUMI.

**ASYPD** Actual simulated years per day, is used to measure a long-running experiment and includes queueing time, taking into account the sharing of HPC resources. This metric is significant when running complex workflows, that include many tasks.

In the case of the ICON workflow, in order to see if the different steps in the workflow add significant overheads, to the whole simulation running. A simple test that could be done is to compare the **SYPD** and the **ASYPD**.

CPUSs	Queuing (s)	Runtime (s)	Q + R	SYPD	SYPD (queues)
1024	4	61	65	27.24	25.56
1024	3	60	63	27.69	26.37
1024	2	60	62	27.69	26.80
1024	2	61	63	27.24	26.37
1024	1	61	62	27.24	26.80

Figure 29: 5 Week ICON R2B4 simulation KPIs



## 8 Conclusions

Since the beginning of the DestinE project, my focus has been to develop a workflow for the ICON model. In order to do so, during the initial phases of the project and my thesis, I had to familiarize with a highly complex HPC environment and with some of the general software that is used in production at the Earth Science Department in the BSC.

In order to schematically review the work that has been achieved, I'll go through the objectives that were defined early on during the project definition.

- **Be able to run the ICON model efficiently.** The workflow for ICON model is able to run the model efficiently. The workflow allows setting the necessary configuration variables, being the amount of nodes, process mapping or other slurm variables, which increase the general model benchmarking.

It also includes the latest changes in the compiler and model sources, allowing to run the ICON model on the GPU partition.

- **Create a portable workflow, being able to run on various HPC platforms like LUMI or MN4/MN5.** The ICON workflow has been run on LUMI, Marenostrum 4 and in Levante, therefore, achieving a great amount of portability. In any case, the latest workflow developments have focused on maintaining compatibility with LUMI, extra work would be required to upgrade the workflow and model sources with other HPC Platforms.
- **Allow running various types of simulations. Therefore, allowing a great amount of personalization.** The workflow allows running many kinds of simulations. Currently, it's able to run with a development configuration; control, historical and low resolution test simulations.
- **Allow great amounts of modularization in regard to base configuration, experiment definition and coupling with other essential components of the workflow.** Through Autosubmit place-holders, the workflow allows setting most recurrently changed model and simulation variables. As for the latest versions of the workflow, it allows running the model with its components plus a separate out process in different partitions.

Regarding the project deviations, the development of the workflow has not been steady nor linear. This has been mostly caused by the issues and delays of the climate readiness of ICON. Adding also, that the availability of computing resources, has not been regular, and that queuing times could sometimes make development unviable.

It is also noteworthy that as a result of the workflow for DestinE framing, the workflow structure, configuration scheme implementation has been developed in shared effort between other workflow developers. Although diverging in the model specific parts, we have closely collaborated in the definition of some of the more general steps.

Finally, the end result of this project, which is still underway, is and will be a workflow which orchestrates the simulations for SoTA models in the most advanced supercomputers in Europe. Which will help and allow producing more accurate climate predictions in order to adapt to our changing climate.

## 9 Sustainability report

### 9.1 Self-evaluation

In general, I think that Destination Earth of which this work is part of, is a great climate adaptation initiative. It is probably the biggest initiative from the European Union to create a digital replica of Earth system, to be used to predict and adapt to the current climate crisis. Also, as it has been issues by the European commission, it is financed by public funds. Therefore, during the next development stages, and when fully implemented, if open science philosophy remains central, and all the data that is generated by the models and later on transformed by the applications is available to the general public, then the project has a remarkable potential to have great impact in adapting to the climate emergency.

On the other side, even though the project has a lot of potential, I'm always uncertain about the need to generate endless amounts of knowledge and information about an already critical situation. Academia appears to operate under the assumption that if academics generate information, then society's leaders will use that information to make wise decisions that promote the public good [Gar]. I think this way of operating is dangerous.

In the current situation we are, I personally think that climate science which is focused on mitigating and adapting to the climate emergency, should be accompanied by brave policies and climate action by our political leaders. If not, it partly loses its purpose.

### 9.2 Economic aspect

- The general costs of the project have resulted higher than I expected. I think this is due to the fact that usually the costs of personnel are greater than the others. This is most likely a result of its use of HPC platforms.
- This project will be used further on by the scientific community, it's making use of the latest improvements in HPC, and on climate modelling, therefore its lifespan will be considerable, allowing many other projects to make use of it. Future phases of the project will This project is really new in its kind, therefore It can't really be compared with previous ones.

### 9.3 Environmental aspect

- To some extent, I've tried to determine the resource cost of the HPC platforms. With more time, I would've liked to investigate further on its impacts, and even calculate the amount of energy used by the project, maybe including its carbon footprint.
- HPC platforms are really resource hungry, therefore the way to reduce the impact of the project is by actively reducing the amount of runs, by making sure the submitted jobs are essential.

### 9.4 Social aspect

- The fact that this project is focused on climate adaptation and the climate emergency, is really fulfilling. Computer Science is usually seen as something completely abstracted from nature. By doing this project, I can apply my technical knowledge to fight the climate emergency.

- This project, and the project it's included in, will improve knowledge about how our changing climate will evolve. This knowledge allows governments to make better policies for adapting to the climate crisis.

## References

- [Ass] United Nations Assembly. *Effects of climate change on the global south*. URL: <https://press.un.org/en/2019/gaef3516.doc.htm>.
- [Avi] Sándor Kertész Avi Bahra Iain Russell. *ecFlow features*. URL: <https://www.ecmwf.int/en/newsletter/166/computing/ecflow-5-brings-benefits-member-states>.
- [Coma] EU Comission. *EU Digital Strategy: Destination Earth*. URL: [https://digital-strategy.ec.europa.eu/en/policies/destination-earth#tab\\_2](https://digital-strategy.ec.europa.eu/en/policies/destination-earth#tab_2).
- [Comb] European Comission. *Effects of climate change*. URL: [https://climate.ec.europa.eu/climate-change/consequences-climate-change\\_en](https://climate.ec.europa.eu/climate-change/consequences-climate-change_en).
- [Comc] European Comission. *Flagship Initiave: Destination Earth*. URL: <https://digital-strategy.ec.europa.eu/en/policies/destination-earth>.
- [Con] NIWA British Crown (Met Office) Contributors. *Cylc Documentation*. URL: <https://cylc.github.io/cylc-doc/stable/html/>.
- [DCa] Daniel Klocke DCathy Hohenegger Luis Kornblueh. *Climate Statistics in Global Simulations of the Atmosphere, from 80 to 2.5 km Grid Spacing*. URL: [https://pure.mpg.de/rest/items/item\\_3055640\\_8/component/file\\_3220094/content](https://pure.mpg.de/rest/items/item_3055640_8/component/file_3220094/content).
- [Dir] Miguel Andrés-Martínez ... Dirk Barbi Nadine Wieters. *ESM-Tools Version 5.0: A modular infrastructure for stand-alone, and coupled Earth System Modelling (ESM)*. URL: <https://gmd.copernicus.org/preprints/gmd-2020-100/gmd-2020-100-manuscript-version4.pdf>.
- [DKR] DKRZ. *Levante HPC Platform*. URL: <https://docs.dkrz.de/doc/levante/index.html>.
- [ESA] ESA. *Desination Earth showcase*. URL: [https://www.esa.int/Applications/Observing\\_the\\_Earth/Destination\\_Earth](https://www.esa.int/Applications/Observing_the_Earth/Destination_Earth).
- [Gar] Charlie J. Gardner. *From Publications to Public Actions: The Role of Universities in Facilitating Academic Advocacy and Activism in the Climate and Ecological Emergency*. URL: [https://www.researchgate.net/publication/351994595\\_From\\_Publications\\_to\\_Public\\_Actions\\_The\\_Role\\_of\\_Universities\\_in\\_Facilitating\\_Academic\\_Advocacy\\_and\\_Activism\\_in\\_the\\_Climate\\_and\\_Ecological\\_Emergency](https://www.researchgate.net/publication/351994595_From_Publications_to_Public_Actions_The_Role_of_Universities_in_Facilitating_Academic_Advocacy_and_Activism_in_the_Climate_and_Ecological_Emergency).
- [JW] Roger Temam Jacques-Louis Lionst and Shouhong Wan. *On the equations of the large-scale ocean*. URL: [https://www.researchgate.net/publication/240152131\\_On\\_the\\_equations\\_of\\_large-scale\\_ocean](https://www.researchgate.net/publication/240152131_On_the_equations_of_large-scale_ocean).
- [JSC] JSC. *JEWELS HPC Platform*. URL: <https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels>.
- [LTD] CSC – IT CENTER FOR SCIENCE LTD. *CSC Lumi Billing units*. URL: <https://research.csc.fi/billing-units>.
- [Mic] Microsoft. *Price for Microsoft 365 Licence*. URL: <https://www.microsoft.com/es-es/microsoft-365/buy/compare-all-microsoft-365-products>.
- [MY] Teresa Holfeld Moritz Hanke Dr René Redler and Maxim Yastremsky. *The new coupling software YAC for Earth system models*. URL: <https://www.dkrz.de/en/communication/news-archive/en-der-neue-yac-koppler-fuer-erdsystemmodelle>.

- [Pri] Dell Prices. *Prices for Dell Laptops*. URL:  
[https://www.dell.com/es-es/shop/port%C3%A1tiles-dell/port%C3%A1til-empresarial-latitude-5420/spd/latitude-5420-laptop/gxcto1542014emea?redirectto=SOC&configurationid=e48be14c-9b0c-4c41-866e-65f8a1ae22fd#features\\_section](https://www.dell.com/es-es/shop/port%C3%A1tiles-dell/port%C3%A1til-empresarial-latitude-5420/spd/latitude-5420-laptop/gxcto1542014emea?redirectto=SOC&configurationid=e48be14c-9b0c-4c41-866e-65f8a1ae22fd#features_section).
- [Res] MPI Researchers. *ICON Earth System Model*. URL:  
<https://mpimet.mpg.de/en/science/models/icon-esm/>.
- [Tar] HPC Center - University of Tartu. *High Performing Platform pricing*. URL:  
<https://hpc.ut.ee/pricing/>.
- [Teaa] Autosubmit Team. *Auto-EcEarth workflow*. URL:  
<https://earth.bsc.es/gitlab/es/autosubmit/-/wikis/home>.
- [Teab] Autosubmit Team. *Autosubmit Documentation*. URL:  
<https://autosubmit.readthedocs.io/en/v4.0.0b/introduction.html>.
- [Teac] Autosubmit Team. *Autosubmit gitlab*. URL:  
<https://earth.bsc.es/gitlab/es/autosubmit>.
- [Tead] ecFlow Team. *ecFlow Documentation*. URL:  
<https://confluence.ecmwf.int/display/ECFLOW>.
- [Top] Top500. *Top500 Supercomputer List*. URL:  
<https://www.top500.org/lists/top500/2023/06/>.
- [Und] EuroHPC Joint Undertaking. *EuroHPC Joint Undertaking Supercomputers*. URL:  
[https://eurohpc-ju.europa.eu/supercomputers/our-supercomputers\\_en](https://eurohpc-ju.europa.eu/supercomputers/our-supercomputers_en).
- [Win] Windows. *Price for Windows 10 Licence*. URL:  
<https://www.pccomponentes.com/microsoft-windows-10-pro-32-64-bit-licencia-completa-fpp-digital>.