D6.1 Requirements on the eFlows4HPC software stack from Pillar III and evaluation metrics

Version 1.0

Documentation Information

Contract Number	9555558
Project Website	www.eFlows4HPC.eu
Contractual Deadline	30.06.2021
Dissemination Level	PU
Nature	R
Author	Marisol Monterrubio-Velasco (BSC)
Contributors	Juan E. Rodríguez (BSC), J. Selva (INGV), Jorge Macías (UMA), Marta Pienkowska (ETH), Steven J. Gibbons (NGI)
Reviewer	Alessandro D’Anca (CMCC)
Keywords	urgent computing, earthquakes, tsunamis, natural hazards

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

Change Log

Version	**Description Change**
V0.1	Proposed ToC and First Draft
V0.2	Second Draft
V0.3	Ready for internal review
V0.4	Changes from the internal review
V1.0	Final version

Table of Contents

1. Executive Summary

Earthquakes and tsunamis are unpredictable and devastating events that can have catastrophic socioeconomic impacts. Scientific progress together with the refinements of monitoring systems, as well as the developments of emergency protocols, contribute to the improvements in our capacity to act in the face of unforeseeable disasters. And yet, although we live in the era of a technological revolution, our susceptibility to natural disasters seems not to decrease. The continuous global population growth makes societies especially vulnerable to extreme events due to our increasing dependency on infrastructural services and our densely populated megacities.

Taking advantage of the rapid increases in computational power, the new technologies developed for data monitoring systems and advanced analytical techniques, the concept of Urgent Computing (UC) for Natural Hazards has emerged. UC links High Performance Computing (HPC), state-of-the-art simulation codes, readily available data and High Performance Data Analytics (HPDA) to provide insights into the impacts and potential damages immediately after the occurrence of an extreme event. In particular, in Pillar III of eFlows4HPC, the UC frameworks focus on providing solutions for mitigating potentially catastrophic earthquakes and tsunamis.

The strategy to develop UC workflows for earthquakes and tsunamis involves conceptual definitions and prototype workflow demonstrators in the first instance, followed by deployments of advanced tools and developments of complex tasks to ultimately bring them to an operational level.

This deliverable presents a detailed description of the current working versions of both earthquake and tsunami workflows and aims to establish a technical roadmap for envisaged further developments. In particular, the document includes a thorough analysis of the requirements at different levels of granularity, such as actions requirements, data flow analysis, execution specificity, and specific metrics definitions. The outlined requirements serve as a guideline for integrating the aforementioned workflows into the eFlows4HPC software stack in order to achieve a HPC Workflow as a Service (HPCWaaS) fulfilling the Quality of Service (QoS) and to bring the earthquake and tsunami workflows to an operational level.

2. Introduction

In a general sense, Urgent Computing (UC) refers to the use of HPC/HPDA during or immediately after emergency situations and typically combines complex edge-to-end workflows with capacity computing, where multiple model realizations are required (to account for input and model uncertainties) under strict time-to-solution constraints. Early decisions in rapid response to earthquakes have to be based upon interpretations of the best, yet often limited, data available at a given time immediately following the event to estimate the severity of shaking and, potentially, the impact of a subsequent tsunami. A combination of data analysis and numerical modelling, starting from a large ensemble of sources describing the uncertainty stemming from limited data, produces maps of strong ground shaking and/or tsunami inundation which can be employed to direct immediate relief measures (e.g. towards civil protection and first and second responders) and to forecast losses (e.g. towards the insurance industry). The temporal horizon for UC typically ranges from minutes to a few hours.

While numerical earthquake and tsunami simulations can model physical phenomena with great accuracy, the uncertainties in poorly constrained parameters result in uncertainties in computed outcomes. Elastic wave propagation and the resulting ground shaking are very sensitive to the prescribed source characteristics (earthquake location, size, and type) and to the underlying velocity and attenuation models (accounting for path and site effects), which may result in a large variability of impact estimates. Similarly, numerical propagation of tsunami waves and the resulting hazard is very sensitive to both the specification of the seafloor displacement and the assumptions made in modelling the inundation [1, 2].

Seismic and tsunami simulations for impact assessment therefore meet the three principal requirements of UC:

1. The computations need to be performed within a specific time frame required by decision makers. The results are only valuable if achieved within the given strict time constraint.
2. The earthquake onset is unpredictable, and thus so is the computation trigger.
3. The simulations are highly computationally intensive and require significant high performance computing (HPC) resources.

To meet these requirements of UC for Natural Hazards focusing on Tsunamis and Earthquakes in Pillar III, the objective is to improve and optimize workflows to provide a rapid response service to natural disasters. The main goal is to deliver faster end-to-end runs, with more robust and reliable workflows, as well as more usable outcomes for potential end-users. This implies a deep analysis of workflow dependencies, bottlenecks, and resource management. A strong focus will be put toward integrating multiple sources of data and a seamless interaction of the workflows into existing end-user decision flows. The intended developments of both workflows will capitalize on the capabilities of the eFlows4HPC software stack.

3. General description of Pillar III workflows and their components

The seismic and tsunami workflows share three main phases:

1. A pre-processing phase, in which an ensemble of possible sources is defined based on seismic data assimilation and earthquake parameter estimation, with uncertainty.
2. A simulation phase, in which individual scenarios are simulated and ground shaking / tsunami inundation are numerically quantified for all the scenarios in the ensemble
3. A post-processing phase, in which simulation results are processed to produce probabilistic forecasts including both source and modelling uncertainty, eventually updated through newly available observations from the monitoring networks.

Figure 1 provides an at-a-glance overview of the existing basic components of the seismic and tsunami workflows and how their common structure serves the shared challenges of source and model uncertainty, the need to perform very rapid numerical simulations, and the need to post-process and disseminate the results. The seismic workflow is called UCIS4EQ (Urgent Computing Integrated Services for EarthQuake) and the overall tsunami workflow is labelled PTF (Probabilistic Tsunami Forecast). Within PTF, there is a “mini-workflow” labelled FTRT (Faster Than Real Time) which sets up and performs the numerical tsunami simulations. In the scope of eFlows4HPC, the capabilities of Machine Learning (ML) and Data Analytics (DA) algorithms will also be explored and integrated at the relevant stages of each of the workflows (not depicted in Figure 1).

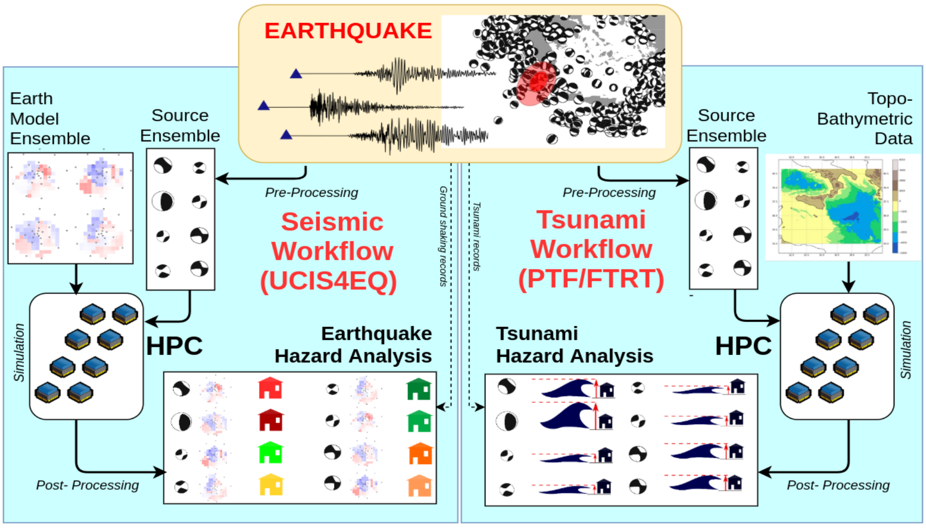

Figure 1. Coarse overview of the PTF/FTRT (tsunami) and UCIS4EQ (ground motion) UC Workflows with main components: Pre-processing, HPC simulation, and post-processing.

However, in spite of the similarities in the overall design, each workflow of Pillar III has been developed independently including its own particularities: making use of different HPC simulation codes, as well as pre- and post-processing strategies and approaches.

3.1 PTF/FTRT

Tsunamis are relatively rare phenomena but can have devastating consequences for coastal communities and infrastructure. Most tsunamis are caused by large offshore earthquakes and Tsunami Warning Centers (TWCs) have the challenging task of forecasting the tsunami threat immediately after such an event. The impact of a tsunami generated by a given earthquake on a given coastline stretch depends both on the source (i.e. the location of the earthquake: its size, and the exact form of the seafloor displacement) and on the topography and bathymetry between the source region and the coastline. The earthquake alarm is raised by automated systems processing near real-time data from seismic stations. The seismic waves travel with a finite speed and, when the first seismic waves arrive at the closest stations, there is enormous uncertainty surrounding the size of the earthquake and the details of the rupture. This uncertainty is reduced as more seismic data arrives, and we may have a very accurate estimate of the size of the earthquake after a few minutes. However, it may take a very long time to produce a complete picture of the rupture and TWCs will need to generate a forecast for the tsunami impact well before this time, within a few minutes after the event for alert purposes, and within a few hours for urgent computing purposes.

We therefore need to consider an ensemble of potential earthquake scenarios and generate a Probabilistic Tsunami Forecast (PTF) which assigns probabilities to the different possible outcomes. With explicit treatment of the uncertainties in the source, PTF will assign probabilities to each of these outcomes such that decision-makers will have a rigorous basis on which to decide the best risk-mitigation action. Figure 2 illustrates a PTF workflow which performs Faster-Than-

Real-Time (FTRT) numerical tsunami simulations for a large ensemble of earthquake scenarios before aggregating an estimate of the tsunami hazard from which Alert Levels can be determined.

At the start of the workflow, an ensemble of earthquake scenarios is determined from the incoming real-time seismic data and an existing knowledge base of tectonic data. The tsunami impact is then calculated for each individual scenario in FTRT calculations distributed over multi-GPU HPC facilities. The output from each of the simulations is then merged, incorporating the source probabilities, to produce a probabilistic hazard forecast.

The basic workflow for PTF/FTRT simulations has been developed by INGV, UMA, and NGI in the framework of the corresponding Pilot Demonstrators of Urgent Computing for Earthquakes within the Center of Excellence for Exascale in Solid Earth (ChEESE, https://cheese-coe.eu/results/pilots). Starting from this implementation, a number of innovations and modifications are planned to improve its performance within a UC framework, updating the workflow and its basic scientific methodology. The main changes consist of i) implementing a circular updating scheme, to guarantee the delivery time and to enable dynamic management of uncertainty; ii) enabling the delivery of results at different levels of convergence, depending on ensemble coverage; iii) exploiting machine learning (ML) techniques to improve performance.

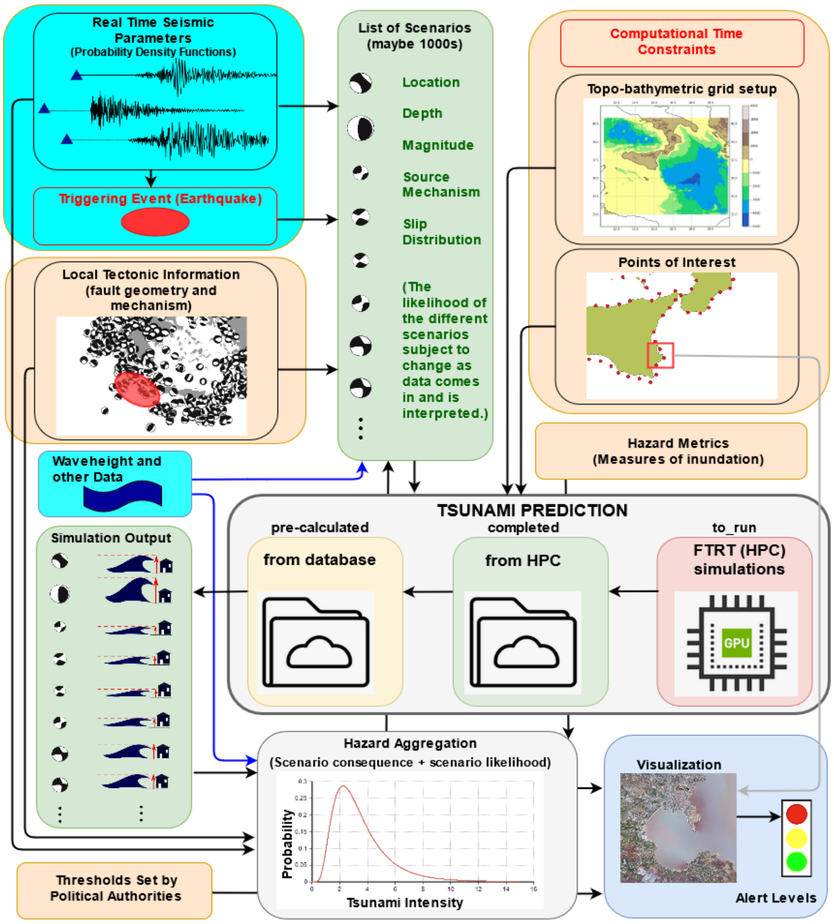

Figure 2. A schematic workflow for Probabilistic Tsunami Forecast (PTF). A more detailed breakdown of the requirements and building blocks of the provisional workflow is displayed in Figure 3.

Different options will be explored to integrate ML algorithms. They may be used to predict inundation results of a tsunami instead of performing full simulations with Tsunami-HySEA. In particular, neural networks can be trained to produce precise estimations of the maximum water height, arrival times or time series of the tsunami at a given set of points. Different options can be provided for the implementation. ML algorithms can be trained with either online data (using the simulations performed during the UC) or offline data (using pre-computed scenarios), or both. Also, different strategies can be implemented during the project, such as:

- Training a neural network (NN) from the results obtained with direct simulations of Tsunami-HySEA.
- Training a classifier to predict the inundation level of a tsunami at a certain area.
- Using a NN to increase the resolution of the simulations: a NN could be trained to produce high-resolution results (on fine grids) from low-resolution results.

More details on the innovative components of the workflow are discussed in Section 4.

3.2 UCIS4EQ

Deadly earthquakes are unpredictable and relatively rare, but have a high socioeconomic impact. Estimating this impact is crucial for hazard preparedness and mitigation, yet doing so from historic events alone remains difficult. With large earthquakes being so rare we often lack sufficient data to constrain any empirical relations, rendering extrapolations to future events imprecise [3]. Physics-based numerical simulations and synthetic data analysis, on the other hand, are powerful tools that can provide detailed shaking information and contribute to outlining the impacts of such events [4]. The information from simulations is very rich in terms of spatial resolution, thus complementing existing live data records in countries boasting dense seismic networks. Moreover, simulations can provide a suitable mitigation to the lack of live data in regions less covered by seismic networks, such as developing countries. A better early assessment of the impact of earthquakes implies both technical and scientific challenges. A novel HPC-based urgent seismic simulation workflow was proposed in [5], hereafter referred to as **Urgent Computing Integrated Services for EarthQuakes (UCIS4EQ)** which has the potential to deliver more accurate short-time reports of the consequences of moderate to large earthquakes.

The UCIS4EQ is a collection of virtualized services developed at the Barcelona Supercomputing Center specifically in the framework of the Pilot Demonstrator of Urgent Computing for Earthquakes within the Center of Excellence for Exascale in Solid Earth (ChEESE, license pending, https://pypi.org/project/ucis4eq/). UCIS4EQ differentiates two systems: a front-end and a business logic implementation or back-end. In general terms, the back-end consists of a set of processes that need to be executed in an orderly and automated manner and orchestrated by a workflow manager (WM). We differentiate three back-end services:

1. **Event Detection.** This service is external to the UCIS4EQ workflow. It is an automatic service which queries external repositories for detecting and identifying worldwide earthquakes with urgent computing requirements. In particular, the query accesses the web services provided by the federated agencies of the International Federation of Seismograph Networks (FDSN). Once an earthquake with certain properties (magnitude, location, etc.) that are indicative of a potentially destructive event) is detected, the urgent computing workflow is triggered. By means of this service, users can also activate the urgent

computing workflow manually for studying specific use cases (e.g. to simulate well-known past earthquakes for research purposes).

2. **Urgent Computing workflow.** The workflow orchestrates the set of processes needed for the successful execution of an urgent event. In the design adopted for the workflow, each process is encapsulated to work as a specialised micro-service acting as access point for the execution of a specific task. In the very rare scenario where two (or even more) EQ events require urgent computing, more than one instance of this service could run simultaneously. In those cases, the target HPC environments could be different for each event simulation as well as the UC policies and designated resources for solving them. For each workflow execution request, the WM must coordinate and execute the set of tasks needed for giving the required results to the final user, including the mechanisms to run solid ground simulations in Exascale HPC environments. Such a coordination includes, among others:

   - dealing with the tasks' input/output,
   - solving task dependencies,
   - task parallelisation,
   - data transfer from external servers/cloud to HPC environments and vice versa,
   - managing conditional paths,
   - job submission/cancelling/restarting/monitoring, and
   - running parallel simulations on different HPC clusters.

   The set of tasks have been grouped into a set of building blocks for a better understanding of the overall requirements of this service and an optimal implementation of the workflow using the workflow manager solution proposed by the eFlows4HPC WP1 software stack. Section 5.1 (ii) provides further details and outlines the specifics of each building block, including the envisaged deployment of Machine Learning (ML) tools. It should be noted that not all components have been developed or will be developed specifically for the UCIS4EQ, as some processes of the workflow may require the use of third-party software to solve some functional requirements.

3. **Data Access Layer (DAL)**. A service that currently provides a simple interface for interacting with:

   - **intermediate** information generated by both Event Detection and Urgent Computing workflow which is stored in non-relational databases (e.g. MongoDB). Such information could include event primal data, the state of the workflow execution, components setup, execution metadata, large data location (e.g. data maps), etc.
   - **long-term data storage services**. It implements the mechanisms to access (in both read and write modes) different storage services such as EUDAT B2SAFE. The DAL service provides support for enabling data access through different communication and authentication protocols. The data that UCIS4EQ needs to store is, for example, seismic velocity models, simulation meshes, off-line trained models, historical past earthquake events information databases, or socio-economic databases. Additionally, the use-case data provenance will be stored following FAIR standards in order to ensure the reproducibility of the urgent computing scenario. For

example, the simulation input/outputs will be stored long-term in order to be analysed and to guarantee its reproducibility.

This service should be accessible to many processes from both Event Detection and Urgent Computing workflows. Therefore, non-functional requirements such as high-availability and redundancy are necessary.

The implementation of a front-end solution which provides the human-machine interface is still pending for UCIS4EQ. A production service must meet this requirement for ensuring such human-machine interaction but it is not in the scope of the eFlows4HPC. The interface needs to support:

- Interactive decision-making during the UCIS4EQ life cycle (e.g. Urgent Computing resources requests)
- Event simulation monitoring
- On-the-fly intervention (stop, restart, pause, etc.)
- Post-processing analysis

4. Innovative components of the Pillar III workflow

Pillar III workflows boast several innovations that will be deployed to enhance present-day workflows and bring them to Workflows as a Service (WaaS).

I. Urgent Computing for Natural Hazards

The general scope of this Pillar is a major innovation and proposes a novel approach complementing the short-term seismic and tsunami hazard assessment.

The UC concept is at the forefront of scientific, technical, and political challenges that need to be tackled to include workflows in services that offer a solution to relevant problems (that is, under prescribed time-constraints and requiring large computational resources). UC within the context of disaster mitigation explores the role that simulations and data analysis can have in rapidly outlining the impact of potentially destructive events to help risk managers in the first phase of post-event intervention. UC relies on accurate and scalable simulations and on data analysis software, powerful computational resources and fast means to access such resources and deliver results in time for decision making. The main challenges toward enabling UC are the production of reliable results, the time to solution, and rapid access to computing resources.

Therefore, beyond the back-end challenges, establishing protocols or policies for urgent access and usage of high-performance computing (HPC) and high-performance data analytics (HPDA) resources is required to achieve fast and accurate results. To date, HPC infrastructures do not provide systematic support for UC and are mostly managed by job management methods that optimize the occupancy of the system above other requirements. Urgent access to resources is thus a highly-relevant and a key innovative component in Pillar III.

II. AI-assisted rapid characterisation of impact

Emulating near-instantaneously the tsunami and earthquake simulation results with the assistance of AI is a high-value development that could provide fast-generated information coming from online and/or offline trained models. In the context of UC, Machine Learning models become a relevant tool both to provide preliminary information within minutes after the event (prior to the completion of the urgent simulations), as well as to complement the final outputs of the urgent service with ample extra information. It is worth noting that the generation of a high quality, dense and clean database of earthquake/tsunami simulations necessary to train different ML algorithms is crucial to ensure more accurate results. Since synthetic earthquake and tsunami simulations are numerical solutions to highly computationally demanding problems, substantial resources are needed to create a large database discretizing the parameter space in our target region(s) (e.g. seismic velocity models, seismogenic faults, bathymetry, cities and towns, and critical infrastructures). To face this challenge, and cover as much data variability as possible, it is important to produce a large number of simulation events to densely sample the parameter space.

III. Data integration and management

Both workflows generate large amounts of real (observed) and synthetic (simulated) data. Efficient integration and management of data, storage, and transfer in a timely manner is a major bottleneck that needs to be addressed. Several data-related innovations can be envisaged in urgent computing for natural hazards.

A. Assembling real and simulated data

Enriching simulated datasets with observations could contribute to improving the management of uncertainties, eventually reducing them through time as new information is available. This may maximize the potential usefulness of the outputs of the service. Employing HPDA in this component would allow for an intelligent tool that feeds into and improves the ML models in real-time and that guides the uncertainty treatment in the simulation ensembles. However, on top of the purely scientific challenges, assembling data in real-time from different data providers is a difficult technical problem. It not only requires high data availability from monitoring centers or agencies, but also fast-transfer services, a large capacity storage, and a smooth and efficient integration and processing of a range of data formats.

B. Interaction of different computational environments

The functional requirements of workflows in this Pillar include interactions of different computational environments, such as remote or external servers, HPC facilities, and the cloud, with different workflow stages implemented for the most suitable environment. It is fundamental to manage the integration of the different environments. This state-of-the-art approach must incorporate technical modules that allow data, metadata, and information transfer in real-time considering the UC tight time constraints. External servers and HPC facilities must interconnect high-efficiency data flows. This may require changes in HPC facilities transfer-policies related to internal security and connection protocols.

C. Efficient storage and post-processing of synthetic results

The large amount of data produced, both in size and in quantity, needs efficient storage and post-processing solutions. In particular, the required short-time to solution imposes tight time constraints on the data movements, and thus the services are very sensitive to where and how the data is stored and processed. Data logistics is thus essential to provide an effective service for UC. In addition, to guarantee data availability and reproducibility, a key requirement is enabling efficient storage for UC results. It is essential to carry out thorough short-term and long-term analyses to identify potential features that may help interpret and/or improve UC outputs. Moreover, UC simulation results can enrich existing databases of synthetic solutions, contributing to the long-term improvements of UC performance.

5. Requirements & Constraints

This section describes the requirements of both workflows at different levels: the building blocks (5.1), the data (5.2) and the execution requirements (5.3).

5.1 Building blocks requirements

The requirements to develop and to deploy the complex workflows of Pillar III are detailed in the so-called building blocks. Each building block is an independent piece of the system that carries out a specific task in the workflow. The data flows through each building block, thus interconnecting them in a complex data creation-consumption pattern. Moreover, the deployment of each building block could occur in different computing infrastructures such as HPC, external servers, or in the cloud. Furthermore, both workflows of Pillar III combine HPC, HPDA, and ML frameworks in their implementation.

The data flows through the specified blocks and their interconnectedness are represented in charts, with each block subsequently described in a table. Note that the numbering of the building blocks does not necessarily correspond to a sequential execution in time.

5.1.1 PTF/FTRT

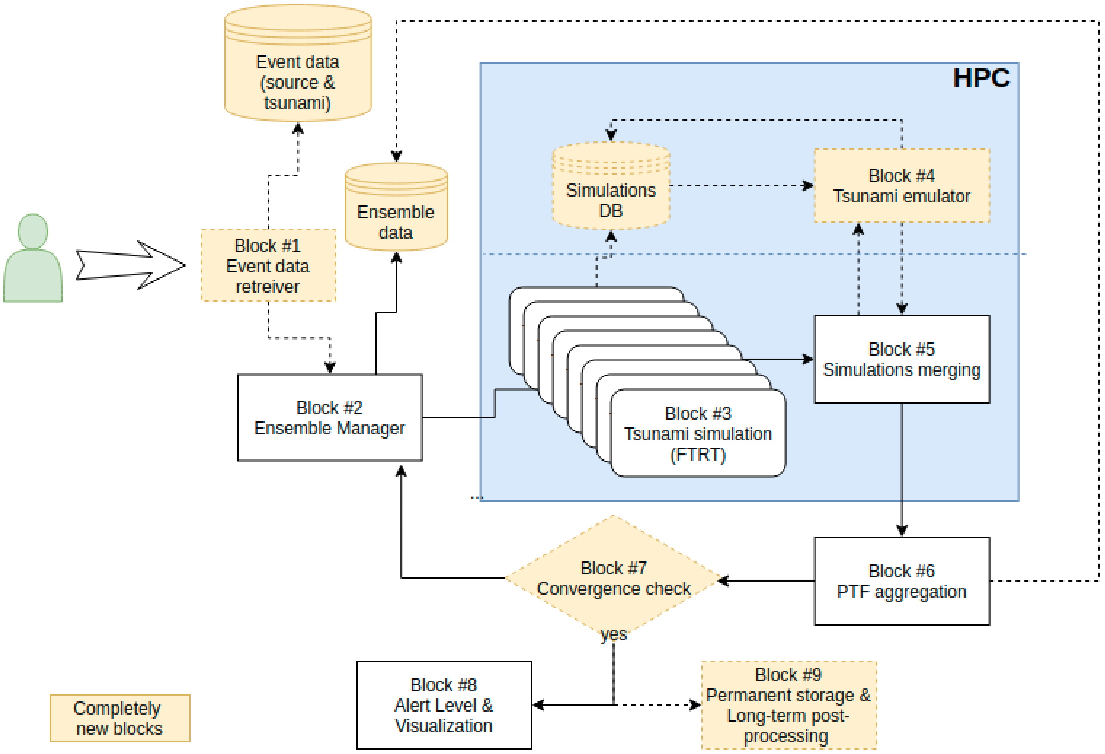

Figure 3. Schematic representation of PTF/FTRT workflow. White boxes are the building blocks already developed and in yellow the completely new components to be developed. The description of each building block is detailed in Table 1.

Table 1. Building blocks description of PTF/FTRT workflow

Building Block	Name	Included actions	Input/Output data structure	ML/DA	Deployment	Description	Status/Phase
1	Event data retriever	Source parameter retrieval Tsunami data retrieval	Input: XML-like, key value, and ASCII files (< 100 MB) Output: key value files (< 10 MB)	no	External host	It regularly updates available event data. Seismic and tsunami data will be retrieved for both further analyses and for check, from international data transmission services (e.g. SeedLink).	To be developed (high priority)
2	Ensemble manager	Source data retrieval and uncertainty quantification Ensemble definition / update Scenario sampler	Input: key value files (< 10 MB) Output: ASCII and raw files (< 10 MB)	no Optional: DA to improve sampling	External host	It integrates available data (from block 1) and defines / updates the event ensemble (scenario list and relative probabilities). From the ensemble, it samples the set of scenarios to be simulated at each round.	Developed in MATLAB. Porting to Python ongoing. Automatic data retrieval (from block 1) to be developed (high priority) Updating of scenario probability to be developed
3	Tsunami simulation	Jobs definition GPU Tsunami simulations (FTRT) Data extraction from single scenarios Simulation DB population	Input: netCdf GRD, ASCII files (40-350 MB) Output: netCdf files (< 200 MB)	no	HPC-GPU	It performs parallel tsunami simulations on predefined topo-bathymetric grids and it extracts the information required for PTF from simulated time series to produce smaller output files. TBD: It populates the Scenario DB	Fully developed. Scenario DB population to be developed
4	Tsunami emulator	Training on available computed scenarios Testing ML performance	Input: netCdf files (< 200 MB) Output: TBD	ML	HPC	Based on available simulations build a tsunami emulator for the rapid characterization of tsunami intensity at sites as a function of source parameters	To be developed
5	Simulation merging	Retrieval of simulation results Trigger/retrieval of ML Aggregation of simulations & ML emulations	Input: netCdf (< 100 MB) Output: netCdf (< 100 MB)	no	HPC	It aggregates simulation results in a single netCDF file. TBD: In the first round, it triggers ML for tsunami emulator (block 4). When ML convergence is reached, it merges emulator results with available simulation results.	Simulation aggregation developed, without ML functionalities. Integration with ML to be developed
6	PTF aggregation	Probabilistic combination	Input: netCf (< 100 MB) Output: netCDF and raw files (< 10 MB)	DA	External host or HPC	It quantifies the probabilistic forecast at target points, based on source ensemble and simulations.	Developed in MATLAB. Porting to Python ongoing.
7	Convergence check	Evaluation of convergence	Input: netCDF and raw files (<	DA	External host or HPC	It checks if the tsunami ensemble is	To be developed(high priority)

			10 MB)			enough populated to produce stable results. The method has still to be developed.	
8	Visualization	Alert level definition Automatic Visualization	Input: netCDF and raw files (< 10 MB) Output: TBD	no	External	It defines alert levels (if required) and it produces the visualization of the results.	Alert level definition and automatic visualization developed in MATLAB. Porting to Python ongoing. Data logistics to be developed
9	Permanent storage & long-term post-process	Permanent storage of all data secondary analyses Hazard disaggregation Pattern extraction	Input: all types above (> 50 GB)	No Optional: ML (?)	External host/ HPC (?)	It moves all the results toward permanent storage, for second use. and post-event analysis	To be developed

5.1.2 UCIS4EQ

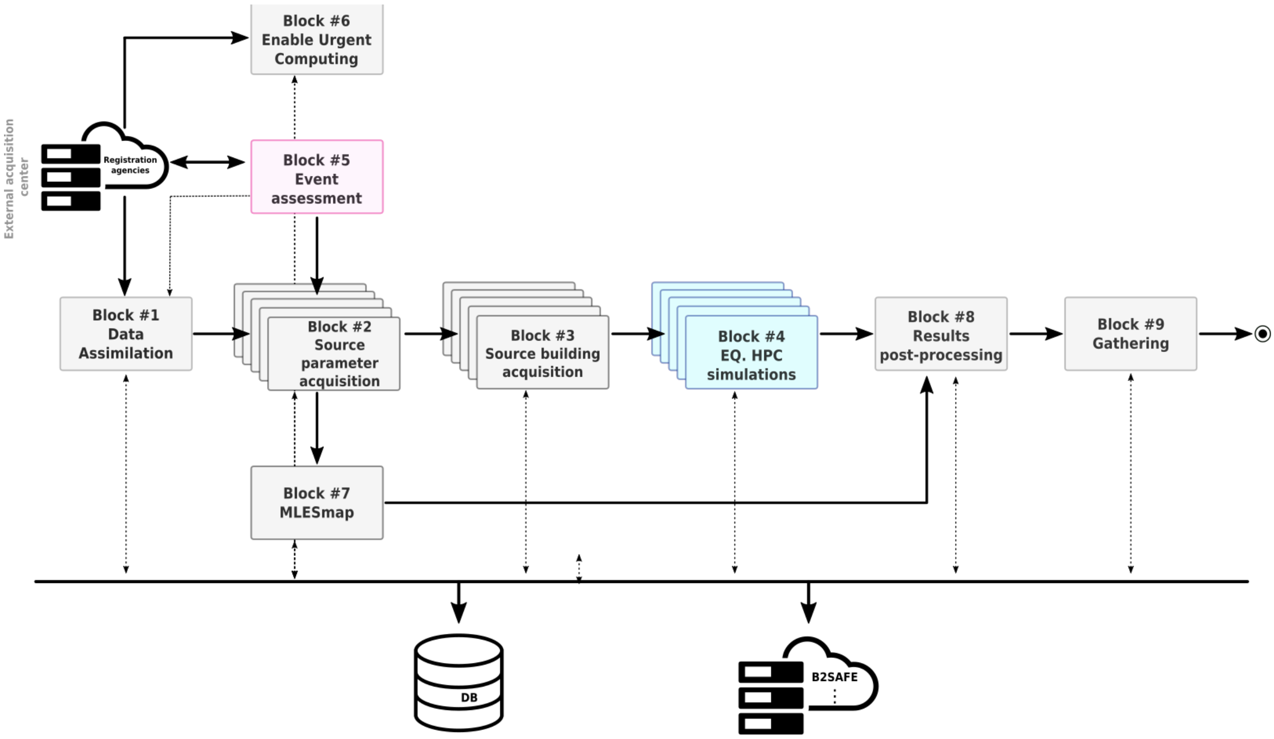

Figure 4. Schematic representation of UCIS4EQ workflow. Gray boxes correspond to building blocks deployed in external servers or in the cloud. Blue box is the building box deployed in HPC facilities. And the pink box indicates the building box that needs data streaming deployment. The description of each building block is detailed in Table 2.

Table 2. Building blocks description of UCIS4EQ workflow

Building Block	Name	Included actions	Input/Output data structure	ML	Deployment	Description
1	Data Assimilation	Unify and Register event Obtain EQ region information Event assimilation	Inputs: QuakeML's (XML like format) Outputs: key-value set	No	External host or cloud	Receives the first event data or the updated information and decides which (if any) information needs to be calculated.
2	Source Parameters acquisition	Statistical CMT Invert CMT Finite Fault Inversion	Inputs: key-value XML (<10MB) Raw Data (<100MB) Outputs: key-value RawData (<1MB)	Yes	External host or cloud	Computes the source parameters using different methodologies according to the data provided by the Data Assimilation building block.
3	Source building	Generate rupture Build Input Parameters	Inputs: Plain text Raw Data (<100MB) Outputs: key-value Plain text (<1GB)	No	External host, cloud or HPC	Performs the finite fault kinematic source description by means of the Rupture Generator code either from CMT solutions or from Finite Fault inversions computed in Source Parameters acquisition building block. All input parameters are built using the information provided by the external data repositories.
4	EQ. HPC simulations	Building Model EQ simulation Post-Process raw data	Inputs: key-value Plain text (<2GB) Outputs: HDF5 (<15GB)	No	HPC	Performs all the functions that require HPC facilities such as mesh building or gathering a precomputed mesh, earthquake simulation and simulated raw data processing.
5	Event assessment	Time series retrieval Update event Event evaluations	Inputs: key-value Outputs: Raw Data (<100 MB) key-value	Possible	External host or cloud (Stream)	Queries the updated information from external agencies and international seismic networks. Subsequently, the changes and updates are evaluated in order to decide if they call for a start of a new instance by the data assimilation building block or for aborting a task in progress.
6	Enable Urgent Computing	Compute Index Priority Set Computing Resources Activate UC protocol	Inputs: key-value ML offline trained models Historical EQ.catalogs (<400MB) Outputs: key-value	Yes	External host or cloud	Enables the UC policies considering the index priority computation. This building block could require external (human or automatic) interaction. In parallel to the decision to activate or not to activate the UC policies the required computer resources are calculated.
7	MLESmap	Real-time Intensity Maps from ML	Inputs: ML models trained offline Outputs: Ground Shaking Maps	Yes	HPC (precomputed offline)	Computes ground shaking maps based on ML models which are offline trained. This information assists the uncertainty quantification stage and also provides real-time affectation information prior to the availability of the simulated results.

8	Results post-processing	Unify results Uncertainty quantification	Inputs: HDF5 (<100GB) Outputs: key-value	Possible	External host or cloud	Post-processes the information from the EQ simulations, and the other observations such as local GMPE's, times series, MLESmap in order to unify results and quantify uncertainties.
9	Gathering	Build data provenance Generate final outcomes	Inputs: key-value RAW data Outputs: Maps and documents	no	External host or cloud	Generates the Workflow outputs such as maps, reports, and time series, among others.

I. Programming languages

Programming language	**Versions/distributions**	**Role in the PTF/FTRT workflow**
CUDA		Tsunami-HySEA
Python	>= 3.6 (in miniconda environment)	Retrieval of seismic and tsunami data, post-processing of Tsunami-HySEA and tsunami emulator results
Matlab	> 2016	Definition of the ensemble and probabilistic aggregation
Programming language	**Versions/distributions**	**Role in the UCIS4EQ workflow**
Python		Main programming language of UCIS4EQ including the actions of Table 2

II. Specialised software/licenses

Library	**Versions/distributions**	**License**	**Used in PTF/FTRT building block**
Tsunami-HySEA	3.8.1		Block 3
WMS light	0.1	Apache Licence v2	Workflow manager (implementation ongoing in the framework of ChEESE)
Bash	>=4.1.2	GNU GPL v3	Simplified workflow manager
Early Est	>= 1.2.4	GNU GPL v3	Block 1
Library	**Versions/distributions**	**License**	**Used in USIC4EQ building block(s)**
Docker	> 19.03	Apache License 2.0	BB #1, BB #2, BB #3, BB #5
Mongo DB	>= 4.2	GNU AGPL v3.0	BB #1, BB #2, BB #3, BB #5
WebDAV client	>=3.14.5		BB #1, BB #2, BB #3, BB #5
Scikit Learn	0.24.2	BSD License	BB #2, BB #6, BB #7
BayesISOLA	3.0	GNU Lesser General Public v3.0	BB #2 (aiming to used)
Seiscomp3	1.2	SeisComP Licenses	BB #2, BB #5 (aiming to used)
Graves-Pitarka rupture generator	>= 5.4.1	Apache License	BB #3
Salvus	1.2.3	Software License Agreement for Academic Use	BB #4
Parmetis	4.0.3	Educational and research purposes	BB #4
Intel MPI	2017.4		BB #4
Obspy	1.2.2	GNU Lesser General Public License v3.0	BB #5 ((aiming to used)
CyberShake plattform		Open source	BB #7

5.2 Data requirements

In the context of the definition of data consumption patterns we classify data in three different groups depending on the use of such datasets during the workflow execution: sources (inputs), intermediate (temporal) or outputs. Moreover, the persistency of the data is indicated, where persistent means that we need to store the information used for the simulation to ensure reproducibility. Non-persistent describes the datasets with no need of storage in the data repositories after execution.

5.2.1 PTF/FTRT

In Table 3, we describe the main data produced in the PTF/FTRT workflow. Most of the produced data are thought to be persistent, to allow second order analyses, while only a part of them is strictly used to produce the urgent computing results. The optimization of file size is considered only for those data that should be moved from HPC systems to external resources, which happen at the interface between blocks #2 and #3, and blocks #5 and #6 (Figure 3).

Table 3. Data requirements for PTF/FTRT workflow

Data sets	**Data Flow**	**Persistency**	**Data type/structure**	**Consumption pattern**	**Read or Created**
Earthquake source data	Source	Persistent	key value file (JSON like format), XML files (QuakeML, YALM), other plain text (CMT, ad hoc, etc.)	1 to 1	Block 1
Tsunami data	Source	Persistent	plain-text	1 to 1	Block 1
Ensemble data	Intermediate	Persistent	plain-text and raw file	1 to N	Blocks 2
Topo-Bathymetric data	Source	Persistent	netCDF GRD format (netCDF3 CLASSIC 32-bit offset)	1 to 1	Block 3
Tsunami-HySEA Input (Earthquake source and run parameters)	Intermediate	Persistent	plain-text files	1 to N	Blocks 2, 3
Tsunami-HySEA output (maxima from time series, arrival time, time series in the gauges)	Intermediate / Output	Persistent	NetCDF	1 to N	Blocks 3,4
Tsunami-HySEA compressed output for PTF	Intermediate	Non-persistent	NetCDF	N to 1	Blocks 3
Tsunami-HySEA compressed output for PTF	Intermediate	Persistent	NetCDF	N to 1	Blocks 5,6
Tsunami forecast	Output	Persistent	NetCDF and raw	1 to N	Block 6
Figures & reports	Output	Persistent	various figures and ASCII files	1 tot 1	Block 8

5.2.2 UCIS4EQ

The data requirements of USCIS4EQ consider two different levels of granularity. First, the data creation-consumption pattern of the UCIS4EQ workflow at a building block level is represented as a Directed Acyclic Graph (DAG) in Figure 5. In this figure each node is a specific building block from

Table 2, the edges represent the interaction between them and node dependencies are defined by the arrow's direction.

Figure 5 shows one of the simplest cases that could occur in the UCIS4EQ where only one interaction is done by the Event Assessment building block.

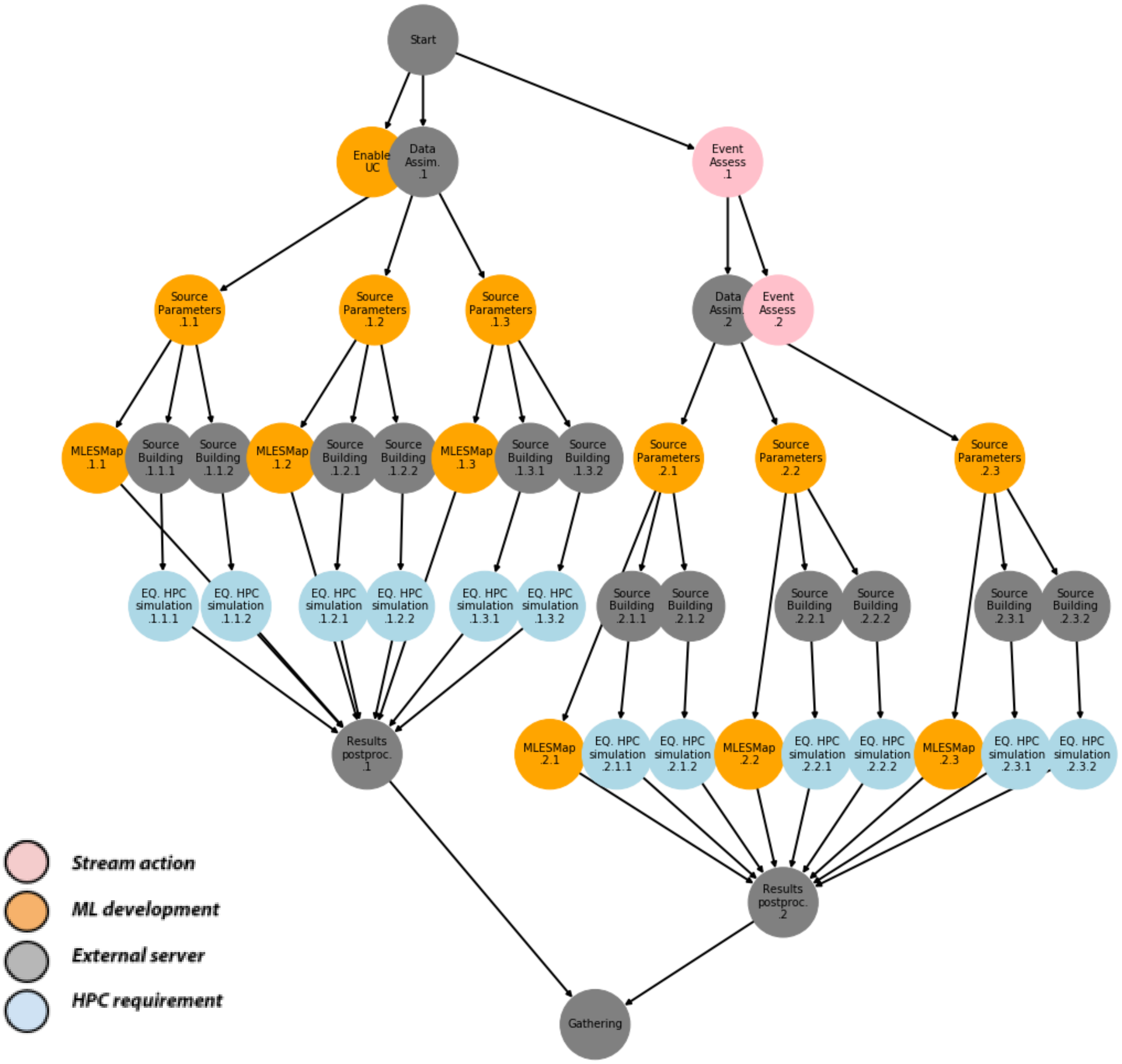

Figure 5 Directed Acyclic Graph of the UCIS4EQ showing a simplest data creation-consumption pattern at building block level.

This example shows how, after evaluation of the new data queried from streaming, the Event Assessment precedes a second Data Assimilation process, thus beginning a second sequence of building block interactions. This DAG is just an example to show the workflow consumption pattern. However, in a production service, the complexity of this pattern of interactions can increase. The colours at each node indicate particular characteristics. Pink nodes are building blocks with data streaming actions. Orange nodes are building blocks where ML tools are used or aimed to be developed. Grey nodes indicate building blocks without any specific requirement but deployed in external servers. Last, blue nodes show the building blocks that require HPC facilities to be deployed.

In Table 4 a general description of the datasets read or created in the UCIS4EQ is shown.

Table 4. Data requirements in UCIS4EQ

Data sets	Data Flow	Persistency	Data type/structure	Consumption pattern	Read or Created
QuakeML’s	Sources	persistent	XML	1 to 1	Read in BB #1
Trained ML models	Sources	persistent	RAW	1 to 1, 1 to N,	Read in BB #1, BB #2, BB #6, BB #7
Past EQ catalogs	Sources	persistent	csv, XML	1 to 1	Read in BB #6, BB #2
Velocity models	Sources	persistent	HDF5	1 to N	Read in BB #3, BB #4
Regional bathymetry and topography	Sources	persistent	HDF5	1 to N	Read in BB #4
Socio-economic database	Sources	persistent	plain-text	1 to 1	Read in BB #6
Receivers & critical infrastructure location	Sources	persistent	plain-text	1 to N	Read in BB #4, BB #8
Time series	Sources	non-persistent	HDF5	1 to N	Read BB #2, BB#8
CMT solutions	Intermediate	persistent	key-values	1 to N	Read in BB#5 and/or Created in BB #2
Fault rupture models	Intermediate	persistent	plain-text	1 to 1	Created in BB #3
Computational meshes	Intermediate	non-persistent	HDF5	1 to 1	Created in BB #4
Full-field	Intermediate	non-persistent	HDF5	1 to 1	Created in BB #4
Shake maps of ground motion proxies (PGA, PGV, PSA, shaking duration)	Intermediate	persistent	png, jpg images	N to 1	Created in BB #4, and BB #7
Waveforms at specific sites	Intermediate	persistent	HDF5	N to 1	Created in BB #4
Uncertainty quantification analysis	Output	persistent	document, png or jpg images	N/A	Created in BB #9
Data provenance report	Output	persistent	document	N/A	Created in BB #9

5.3 Workflow deployment / execution requirements

The deployment of both workflows depends on specific execution requirements; the innovative character of the tsunami and earthquake workflows in this project requires state-of-the-art deployment and execution parameters. The requirements described below could improve the workflows ensuring a Quality of Service (QoS) once they are in production. Although all of the execution requirements listed below are highlighted, we stress the priority of each following the RFC 2119 convention [6].

Table 5. Execution requirements in the Workflows of Pillar III

ID	Name	Description	Priority (must, should, may)
1	Urgent computing access	Priority access to HPC computational resources	must
2	Data accessibility	Some data required in HPC computations is stored in external repositories and some computational results must be required for post processing in external services. So, some High-performance data management mechanism between external repositories and HPC facilities is required in order to make data accessible in the infrastructure required by the computation.	should
3	Data replication	Redundancy of data is required in different phases of the workflow execution. Source data must be replicated in different locations to assure a high-availability computation as well as avoiding time consuming data transfers (e.g. computational meshes). Intermediate data generated by large computation must be also considered in order to avoid losing data in case of failures.	must
4	Execution Robustness	Support for the management of fault tolerance during the workflow execution including checkpoints or retries. For example, during a large execution if a node fails, the workflow must be able to recover and continue to the end.	must
5	Infrastructure interoperability	Interoperability between computations performed in different infrastructures used in the Pillar (e.g., HPC clusters and external servers)	must
6	Portability and Reusability	Workflow and its components must be portable and reusable to several infrastructures and users.	may
7	Streaming Data Source	Management of streaming data sources in real-time from external agencies or servers	must
8	Integrated workflow manager	Support for the management of task dependencies, execution of parallel simulations on different HPC infrastructures, management of batch jobs (submission, monitoring, cancellation), management of conditional paths, and coordination of microservices invocations	must
9	Integration with permanent storage	Support for access to external data repositories (R/W) such as EUDAT - Data Storage services (e.g. B2SAFE). Support for final storage in long-term storage for second use and/or to satisfy FAIRness policies	must
10	Inference with online/offline ML	Support to the use of inference from online and/or offline trained ML models by Earthquake and Tsunami emulators as	must

	models	steps in its workflows	
11	DA integration	Predictive and prescriptive data analytics to assist some building blocks in analysis and decision tasks	may
12	Workflow malleability	Capability to cancel and add new component invocations at run-time	should

6. Metrics to be used in the evaluation of the developed workflows

The evaluation of workflows during the eFlows4HPC software stack implementation needs the definition of metrics to help quantify the improvements. Some of these metrics are general for all the Pillars and they are provided by the WP1 and described in D1.1. However, in addition to those metrics, Pillar III proposes four specific metrics in order to evaluate its particular requirements.

Table 6. Specific metrics for Pillar III

Acronym	Specific metrics for WP6	Description
RT	Response time	End-to-solution time constraints in an urgent computing context. This metric is defined as the clock time measured from the event reception until a first valid solution for the event is delivered to the stakeholders such as civil protection agencies. RT must be normalized with respect to the specific amount of computational resources used for a given instance in order to make the metrics comparable between different workflow instances.
UAR	Urgent Assignment of Resources	Time to obtain necessary resources for an urgent computing execution. The inclusion of this metric quantifies the QoS in HPC facilities that provide UC services. Moreover, it is a measure to evaluate if the adopted policies are adequate for an UC execution.
RUQ	Results Uncertainty Quantification	High-fidelity and high-accuracy results. This metric is proposed to fulfill the specific UCIS4EQ workflow. RUQ metrics is related to the uncertainty of the service outputs, as it is crucial to constrain and reduce the uncertainty of provided impact estimates. The uncertainties must be calibrated and assigned, among others, via the available GMPEs in the study region, observations from monitoring systems, information of past earthquakes, and results from ML algorithms.
Conv	Convergence	This metric is proposed to evaluate the specificity of the PTR/FTRT requirements in particular the convergence of the results based on a reference solution

7. Conclusion

Urgent Computing workflows in Pillar III combine both technical and scientific state of the art and developing them as WaaS has a very high social impact. The potential use of tsunami and earthquake workflows at an operational level, however, requires specific innovative deployments and particular metrics to evaluate successful execution. Following an exhaustive analysis, this document provides details about the requirements and constraints of each workflow. Each building block is defined including its definition, actions, input/output data structure, and deployment infrastructure. The data creation-consumption patterns are also included. Finally, the execution requirements and the specific metrics are defined. Integrating the workflows into the eFlows4HPC software stack ecosystem (including Data Logistics, Advanced Data Analytics, Machine Learning algorithms, among many others) will deploy both Use Cases as HPCWaaS bringing them to an operational level.

8. Acronyms and Abbreviations

Term or abbreviation	*Description*
AI	Artificial Intelligence
ChEESE	Center of Excellence for Exascale in Solid Earth
D1.1	Deliverable 1.1Requirements, metrics and architecture design
DA	Data Analytics
DAG	Directed Acyclic Graph
DAL	Data Access Layer
EQ	Earthquake
FAIR	Findable, Accessible, Interoperable, Reusable
FTRT	Faster-Than-Real-Time (FTRT)
GMPE	Ground Motion Prediction Equations
HPC	High Performance Computing
HPDA	High Performance Data Analytics
HPCWaaS	HPC Workflow as a service
ML	Machine Learning
MLESmap	Machine-Learning based Estimator for ground motion Shaking maps
NN	Neural Network
PTF	Probabilistic Tsunami Forecast
QoS	Quality of Service
TWCs	Tsunami Warning Centers
UC	Urgent Computing
UCIS4EQ	Urgent Computing Integrated Services for EarthQuakes
WM	Workflow Manager
WP1	Work Package 1

9. References

[1] Grezio, A. Babeyko, M. A. Baptista, J. Behrens, A. Costa, G. Davies,E. L. Geist, S. Glimsdal, F. I. González, J. Griffin, C. B. Harbitz, R. J.LeVeque, S. Lorito, F. Løvholt, R. Omira, C. Mueller, R. Paris, T. Par-sons, J. Polet, W. Power, J. Selva, M. B. Sørensen, H. K. Thio, Prob-abilistic Tsunami Hazard Analysis: Multiple Sources and Global Applications, Reviews of Geophysics 55 (4) (2017), 1158–1198.

[2] G. Davies, J. Griffin, F. Løvholt, S. Glimsdal, C. Harbitz, H. K. Thio,S. Lorito, R. Basili, J. Selva, E. Geist, M. A. Baptista, A global probabilistic tsunami hazard assessment from earthquake sources, Geological Society, London, Special Publications 456 (1) (2018), 219–244.

[3] E. Daniell James, M. Schaefer, and F. Wenzel, Losses associated with secondary effects in earthquakes. Frontiers in Built Environment 3 (2017), 30.

[4] Wald, D. J., Worden, C. B., Quitoriano, V., & Goltz, J. (2002). Shakemap: its role in pre-earthquake planning and post-earthquake response and information. In Proceedings SMIP02 Seminar on Utilization of Strong-Motion Data (pp. 1-20).

[5] de la Puente, J., Rodriguez, J. E., Monterrubio-Velasco, M., Rojas, O., & Folch, A. (2020, June). Urgent Supercomputing of Earthquakes: Use Case for Civil Protection. In Proceedings of the Platform for Advanced Scientific Computing Conference (pp. 1-8).

[6] Bradner, S. O. (1997). Key words for use in RFCs to Indicate Requirement Levels. RFC 2119. Request for Comments. https://rfc-editor.org/rfc/rfc2119.txt