http://www.everest-h2020.eu

dEsign enVironmEnt foR Extreme-Scale big data analyTics on heterogeneous platforms

EVEREST

D2.4 - Refined definition of the application use cases

The EVEREST project has received funding from the European Union's Horizon 2020 Research & Innovation programme under grant agreement No 957269

Project Summary Information

Project Title	dEsign enVironmEnt foR Extreme-Scale big data analyTics on heterogeneous platforms
Project Acronym	EVEREST
Project No.	957269
Start Date	01/10/2020
Project Duration	42 months
Project website	http://www.everest-h2020.eu

Copyright

Num.	Partner Name	Short Name	Country
1 (Coord.)	IBM RESEARCH GMBH	IBM	CH
2	POLITECNICO DI MILANO	PDM	IT
3	UNIVERSITÀ DELLA SVIZZERA ITALIANA	USI	CH
4	TECHNISCHE UNIVERSITAET DRESDEN	TUD	DE
5	Centro Internazionale in Monitoraggio Ambientale - Fondazione CIMA	CIMA	IT
6	IT4Innovations, VSB – Technical University of Ostrava	IT4I	CZ
7	VIRTUAL OPEN SYSTEMS SAS	VOS	FR
8	DUFERCO ENERGIA SPA	DUF	IT
9	NUMTECH	NUM	FR
10	SYGIC AS	SYG	SK

Project Coordinator: Christoph Hagleitner – IBM Research – Zurich Research Laboratory

Scientific Coordinator: Christian Pilato – Politecnico di Milano

Disclaimer

Deliverable Information

Work-package	WP2
Deliverable No.	D2.4
Deliverable Title	Refined definition of the application use cases
Lead Beneficiary	CIMA
Type of Deliverable	Report
Dissemination Level	Public
Due date	31/01/2023

Document Information

Delivery date	
No. pages	56
Version | Status	4 | Final
Responsible Person	Antonella Galizia (CIMA – CNR)
Authors	Antonella Galizia (CIMA – CNR), Antonio Parodi (CIMA) Riccardo Cevasco (DUF), Fabien Brocheton (NUM), Radim Cmar (SYG), Jan Martinovič (IT4I), Francesco Regazzoni (USI), Gianluca Palermo (PDM)
Internal Reviewer	Martin Šurkovský (IT4I)

The list of authors reflects the major contributors to the activity described in the document. All EVEREST partners have agreed to the full publication of this document. The list of authors does not imply any claim of ownership on the Intellectual Properties described in this document.

Revision History

Date	Ver.	Author(s)	Summary of main changes
16/12/2022	0.1	Antonella Galizia (CIMA – CNR)	First draft
20/01/2023	0.2	Riccardo Cevasco (DUF), Fabien Brocheton (NUM), Radim Cmar (SYG)	First update of the use cases, related workflows and related dataset
30/01/2023	0.3	Jan Martinovič (IT4I), Francesco Regazzoni (USI)	First check on requirements
15/02/2023	0.4	Riccardo Cevasco (DUF), Fabien Brocheton (NUM), Radim Cmar (SYG)	Second update of the use cases and related requirements
02/03/2023	1	Antonella Galizia (CIMA – CNR), and others	Complete draft
15/03/2023	2	Gianluca Palermo (PDM), and others	Second version after internal review
30/03/2023	3	All contributors	Final version
24/06/2024	4	All contributors	Revision after the final review

Quality Control

Approved by internal reviewer	30/03/2023
Approved by WP leader	30/03/2023

Approved by Scientific Coordinator	25/06/2024
Approved by Project Coordinator	25/06/2024

Table of Contents

1 Executive summary

This report updates the status of the use cases considered by the EVEREST project. In the lights of the delivery of baseline implementations (WP6.1 described in D6.1) at M18, the three EVEREST application use cases are revisited halfway through the project schedule. This impacts the requirements of the use cases that have to be revised and where needed, updated to include the occurred changes and modifications. This can be translated in the addition or removal of datasets, in a refinement of computational aspects and facilities, and in a clearer vision on how the EVEREST ecosystem can support the optimization of the use cases.

General considerations done for Deliverable D2.1 are still valid here.
The analysis of the use cases defines a set of requirements that drives the optimizations developed in the EVEREST design environment. Figure 1 shows how the process happens: use case requirements (D2.1 and the current deliverable D2.4), the language requirements reported (D2.2 and the refined D2.5), and the data requirements (D2.3 and the refined D2.6) together define the work on the EVEREST design environment in WP3-6.
WP2 provides the application use cases requirements for validating the EVEREST design framework (WP6). In addition, the data requirements, and language requirements, defined and refined in this WP, will be used in WP3 and in the compilation framework (WP4) to drive the optimization process implemented in WP4 and WP5.
The WP2 deliverables are closely linked and were carefully checked for consistency. Despite the many links between, e.g., application and data requirements, we attempted to make the three documents self-contained and easy to read. Therefore, some basic requirements (e.g., some numbers in the tables summarizing the input data of the use cases) are stated in all the deliverables, because cross-linking all of them would make the individual documents unreadable.
In the same line of being self-contained, this deliverable re-proposes the same structure and most of the text of D2.1, properly updated where needed.

1.1 Structure of the document

Section 2 briefly introduces the application use cases and the overall aim of the report, while Sections from 3 to 5 are dedicated to the discussion of their respective requirements.
Section 3 details the renewable-energy prediction, Section 4 the air-quality monitoring and Section 5 the traffic modelling use case. The pilots are analysed in terms of workflows, and for each model and/or workflow in the functional and non-functional requirements are reported and discussed.

Section 6 proposes a general vision on the requirements derived from the single use cases. The report ends (Section 7) with the list of the possible achievements for each use cases in terms of industrial goals.

In terms of changes/updates: the Introduction has been revised in order to report the updates done compared with the initial definition of the application requirements. Section 3 has been mostly modified since the renewable-energy prediction was implemented by scratch during the first 18 month of the EVEREST project: in D2.1 the state-of-the-art of wind energy prediction was presented, in D2.4 the actual description of the newly developed workflow, accomplished with new computational and data aspects, has been added.
As for the air-quality use case illegible modifications occurred and Section 4 is quite unchanged; while the traffic prediction model changed in different parts, as reported in Section 5.
Such changes in the use cases led to a refinement of the requirements reported in Section 6; actually, **REQ5** and **REQ6** have been updated for a tuned specification while **REQ15** and **REQ16** have been introduced as new requirements emerged in the first half period of the project.
Section 7 revises possible achievements for each use cases in terms of industrial goals: the section also introduces (new) methods, and provides priorities. Furthermore, the Sections associates metrics with KPIs to support the evaluation of the effectiveness of the EVEREST ecosystem.

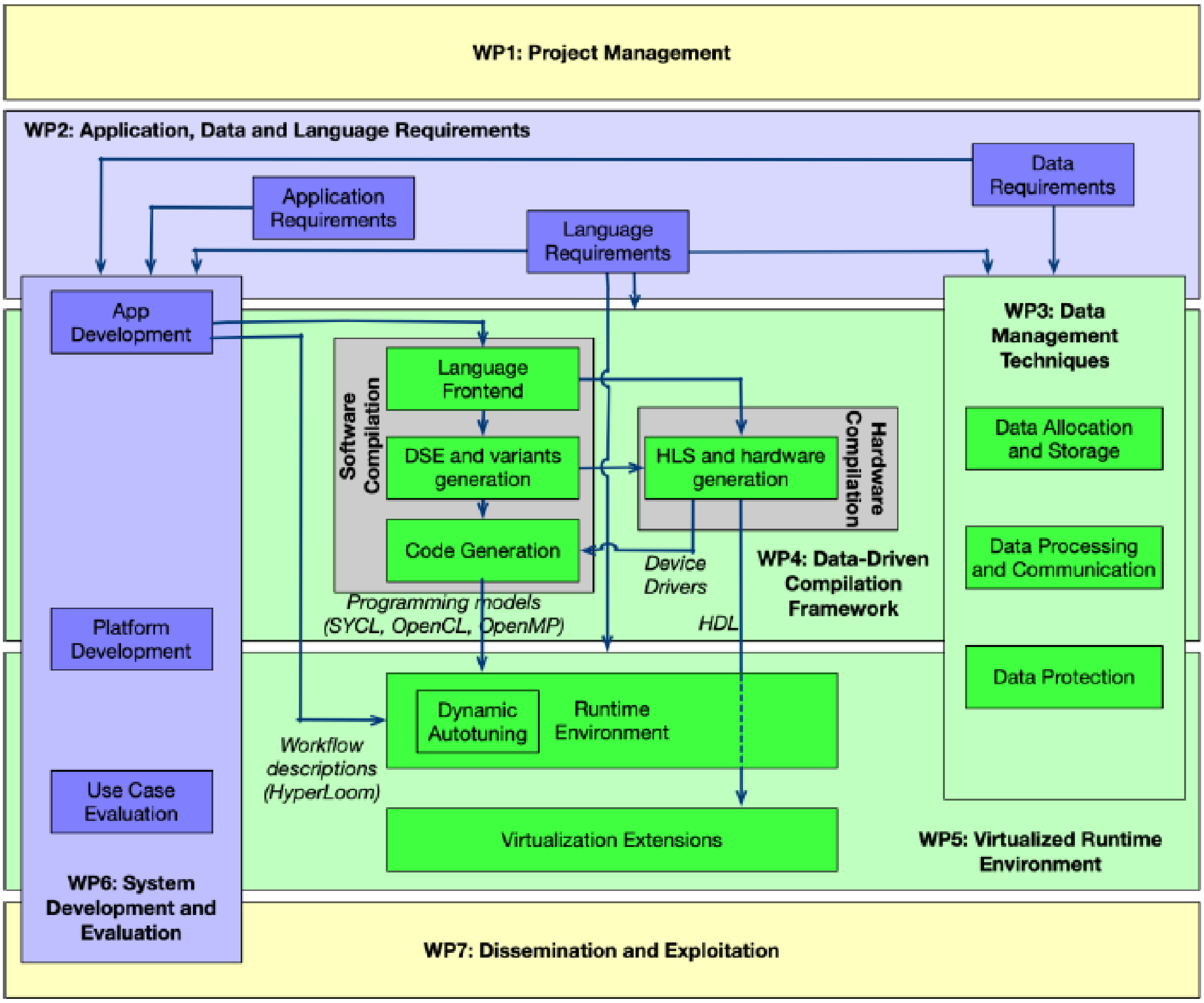

Figure 1 - Overview of the EVEREST work plan

1.2 Related documents

The report is closely related to:
D2.1 - Definition of the application use cases,
D2.2 - Definition of language requirements,
D2.3 - Definition of data requirements,
D6.1 - Preliminary version of the EVEREST applications,
D2.5 - Refined specification of language requirements,
D2.6 - Refined specification of data requirements.

2 Introduction

The aim of this report is to update the description of the applications, in terms of the functional and non-functional requirements, with respect to the beginning of the EVEREST project, i.e. the first analysis done (Month 6) and presented in D2.1. The report proposes the current status of the use cases, updating dataset, methodologies, and technical aspects in the light of the baseline implementation of all use cases deployed halfway through the project (Month 18). This also implies a refinement of the functional and non-functional requirements since they are meant to drive the co-optimization of computation, communication, and data management in the EVEREST programming environment.

In order to be self-contained, the deliverable proposes the entire definition of the EVEREST use cases, maintains the same structure (and most of the contents) of D2.1, properly updated where needed. With this goal in mind, the report describes and analyses the use cases and provides a basic feature of each use case to familiarize with it, to understand the actual data set utilized for the computation, the workflows designed and implemented to run the code as well as the target computational resources. The report acts as the baseline for the validation of their porting in the EVEREST framework.

For reader’s convenience, the three industry-relevant applications considered to validation are briefly mentioned:

- **Renewable-energy prediction**: a weather analysis-based prediction model going from the atmospheric scenario forecast to the energy trading market. The application is meant to support the prediction of the production of renewable wind energy reducing risks related to severe ramp-up/down events. This is obtained improving the prediction of the high-localized meteorological variations at hourly scale, and combining high-resolution weather forecast, including data assimilation, together with artificial intelligence tools.
- **Air-quality monitoring**: the application combines weather forecast with emission forecasts from an industrial site; this can be used to activate mitigation actions as to delay production activities or to activate emission reduction treatments. To improve the quality of the day-to-day forecasts an ensemble of meteorological simulations, including data assimilation, is run at cloud permitting grid spacing (2-4 km) and combined with machine learning approaches on local meteorological measurements.
- **Traffic modelling:** a framework for intelligent transportation in smart cities. The framework combines a traffic simulator, a traffic prediction model and intelligent routing methods to better characterize the road traffic based on combination of multiple data. To improve the quality of the model, the framework is extended to a calculate precise traffic model for all road elements at given time windows considering several data sources such as historical and real-time floating car data of moving

vehicles. It is used for computation of traffic prediction and truly intelligent routing using various machine learning techniques.

It is worth to remind that the use cases present different levels of maturity, besides diverse backgrounds, objectives and exploitation policies. For these reasons, the update of the use cases considers applications in a broad meaning and impacts also the related requirements. As for the renewable energy production, **DUF** has the ambition to implement by scratch a proprietary service, reflecting the state-of-the-art; at this stage of the project, the implementation of the (baseline) newly **DUF** workflow has been deployed with some modifications with respect to the initial vision, i.e. the definition of the data set actually used, the methodology to treat wind farm data accomplished with possible improvements on data integrity. As for air quality monitoring, **NUM** has worked on the solid in-house services based on the very latest Gaussian dispersion models and considering particularly complex effects such as hilly terrain, mixed land use patterns, obstacles, local weather conditions, etc. **NUM** has the ambition to make an additional step forward in technological and scientific advances in the field of atmospheric dispersion modelling to improve the model predictability while reducing execution time. At this stage of the project, the implementation of the (baseline) **NUM** workflow has been deployed with illegible modifications with respect to the vision proposed at Month 6, while improved with data integrity aspects. As for traffic monitoring, the **SYG** ambition is to provide any city with an optimized routing for a large number of cars simultaneously based on road network balanced principles, thus reducing traffic jams in an unprecedented way. Prerequisite to this is highly precise traffic model of a city and its traffic prediction functions, which are obtained by processing floating car big data while applying the state-of-the-art AI techniques. At this stage of the project, the implementation of the (baseline) specific **SYG** workflow has been deployed with slight modifications with respect to the first design of D2.1: the foreseen workflow has been reduced in the number of steps to be performed, and some implementation aspects emerged as necessary to support users in expressing details about the parallelism aspects of the single steps and/or components.

The revision of the baseline workflows led to refinement that impacted the workflows themselves and the functional and non-functional requirements that the industrial use cases pose to the EVEREST environment. Some of the requirements have been updated (e.g., REQ 4 and REQ5), while others have been newly defined (e.g., REQ 16 and REQ 17).

Lastly, for each use case (Section 7 respectively in Table 36, Table 37, Table 38) the list of the possible achievements in terms of industrial goals has been enriched with new information as a baseline starting point and priorities. Furthermore, metrics have been associated with KPIs and the methods to measure them. The lists will support the evaluation of the effectiveness of the EVEREST ecosystem, planned in Deliverable D6.5.

2.1 Methodology

The EVEREST project relies on a use-case-driven approach to systematically collecting requirements, as described in [1].

The process of refining the requirements is based on the definition of the EVEREST Working Groups (WGs). The WG organization was established to tackle specific technological aspects. This approach led to six WGs focused on exploiting the EVEREST SDK to develop the final version of the use cases.

Therefore, the first activity of WGs has been to map the baseline implementation of the three use cases onto the EVEREST technology, and as a consequence, the first result of the WGs discussion has been the refinement of the requirements that is summarized in the WP2 Deliverables.

3 Renewable-energy prediction

In 2017, the European Union generated more electricity from wind, solar and biomass than from coal for the first time [2]. The trend has been confirmed in the last years, in 2019 renewables rose to a new record supplying 35% of EU electricity. For the first time, wind and solar combined provided more electricity than coal, contributing 18% of EU electricity in 2019. This is more than a doubling of market share since 2013 [3]. Renewables rose to generate 38% of Europe’s electricity in 2020 (compared to 34.6% in 2019), overtaking fossil-fired generation for the first time, which fell to 37%. This is an important milestone in Europe’s clean energy transition. In fact, renewables are at the centre of the transition to a less carbon-intensive and more sustainable energy system: The European Green Deal has put the fight against the climate crisis at the very core of all EU policy work over the next five years: EU heads of state have endorsed Europe to become the first greenhouse gas neutral continent by 2050, and the EU commission is putting forward proposals to raise Europe’s 2030 greenhouse gas reduction target to -50% or -55% below 1990 levels. This implies power sector emissions will keep falling, even if electricity demand increases as transport, heating industry continue to electrify [4], [5]. Furthermore, the IEA forecast records wind and solar capacity growth in 2021 and up to 2024 [6].

The European energy market is strongly interconnected with significant interplay between the different energy sources, and the increasing demand for renewable energy sources has introduced significant challenges for power operators, traders and grid owners, due to the intermittent nature of renewables: e.g., a drought period can affect the hydroelectric production, demanding gas generation to compensate. Producers predict wind generation to schedule maintenance activities, but their goal has a different timescale from dispatchers and grid operators, focused on long term delivery (typically day ahead to 1 week or more ahead). Grid operators need to collect generation plans up to date, to ensure grid working and security, balancing demand, and generation in short-term or ultra-short-term timescale. Actually, renewable energy production forecasting becomes crucial; many systems already exist, and they have been adopted by several energy traders to buy or sell shares of energy at a given price to make a profit. This constraint pushes traders to improve the accuracy of their short-term forecasting to reduce imbalance costs due to power forecasting error. To counterbalance to this increasing demand in the market, many providers specialize on a service for site specific generation forecast. However, although many steps forward have been done, large uncertainties still exist [7].

This EVEREST use case aims to provide an accurate forecast system for the energy generated by renewables, in particular for wind power. The wind energy production use case is aimed to predict the short-range (one day ahead), very short-range (up to 6-12 hours ahead) and possibly nowcasting (up to 6 hours

ahead) wind power generation, to schedule on the energy intraday or day-ahead markets. The current energy market considers hourly updates as the most advanced and achieved only by few providers. This represents the actual barrier on the time resolution and is considered as real-time for the sector.
This pilot has been developed from scratch during the project; the use case is challenging in terms of input data quantity and resolution, timing of output delivery, and accuracy expected, so it's crucial to implement innovative hardware and software technologies to process data with the goal of improving the accuracy of the forecast of wind speed and power production. During the development, different architectures, models and approaches have been tested, the assessment of their performance permitted to design a system with a right cost-benefit ratio, suitable for industrial use.
Furthermore, it is to mention that at the current state of the EVEREST project, the WRF model is run once a day and this of course impacts the time-range of the forecast of the energy prediction, focused on the day ahead horizon. The possibility to execute the WRF more frequently leads to advancement also from this point of view.

3.1 Description of the application steps and related pipeline

This use case relies on three main steps for each energy production prediction simulation:

- Step 1: Compute deterministic (short range) and/or probabilistic (nowcasting) meteorological forecast with the Weather and Research Forecasting (WRF) model. Data assimilation procedures are applied to force computation through atmosphere observations to improve weather prediction. Due to the high computational and memory requirements, HPC resources are mandatory to run the simulations.
- Step 2: Pre-processing phase of the site-specific data (both WRF output and observation data) exploited in Step 3. This step mainly consists in the exploitation of a deterministic model discretizing the power curve of a Wind Turbine used as a validator for the aforementioned data. The pre-processing is applied on historical dataset, and more frequently on site-specific WRF output and observations that continuously feed the dataset.
- Step 3: Combine the deterministic weather forecast with the pre-processed data by Step 2 by the means of the Kernel-Ridge with Gaussian Kernel ML algorithm. Results are then post-processed to improve the shape of the energy forecast curve.

3.2 Data processed in the pilot

For each step just mentioned, a description of the input and output data presently utilized for the modelling/processing is provided. More detailed data information is provided in Deliverable D2.3 and D2.5.

It is to mention that all datasets are described at the present state of EVEREST practise, i.e., their delivery frequencies support the run of one workflow per day, more frequent updates of the data, in particular of the observations, are simply not beneficial. However, more run per day (e.g., every 12 or 6 hours, or even less) could be properly supported since the use case could provide data on hourly bases. This frequency of forecasts is considered the most advanced at the state of the art.

3.2.1 Input dataset

Step 1:

- Global weather forecast simulation from NCEP (GFS 0.25 Degree Global Forecast) available at https://www.nco.ncep.noaa.gov/pmb/products/gfs/ to initiate values and forcing at the border of the simulated domain.
- Provision of weather station data:
  - Wunderground as source of non-authoritative sensors data,
  - Italy authoritative weather stations.
- Remote sensing observation data - Radar data from the Italian Civil Protection radar mosaic.

Step 2:

- Output from Step 1, i.e., wind speed, air temperature and other atmosphere parameters forecasted by WRF model.
- Historical site-specific generation data (hourly basis [MWh]).
- Historical site-specific availability data (hourly basis [MW]).
- Historical site-specific curtailment data (hourly basis [MW]).
- Daily site-specific generation data on hourly basis [MWh].
- Daily site-specific availability data (hourly basis [MW]).
- Daily site-specific curtailment data (hourly basis [MW]).

Step 3:

- Output from Step 2, i.e.:
  - WRF model output - wind speed, air temperature and other atmosphere parameters forecasted;
  - Historical and daily site-specific generation data (hourly basis [MWh]);
  - Historical and daily site-specific availability data (hourly basis [MW]);
  - Historical and daily site-specific curtailment data (hourly basis [MW]);
- Daily site-specific availability prediction (hourly basis [MW]).

3.2.2 Output dataset

Step 1:

- Wind speed, air temperature and other atmosphere parameters forecasted by WRF model.

Step 2:

- WRF forecast related to the wind power site (coordinates and turbine hub height);
- Observational data, pre-processed to exclude bad data/anomalies.

Step 3:

- Power generation forecast by machine learning algorithms on hourly basis [MWh] with the timeline of the day ahead.

3.3 Computational resources required for the application

Analysing the pipeline defined in Section 3.1, we have three steps to detail; Figure 2 proposes a schematic representation of the corresponding workflow.

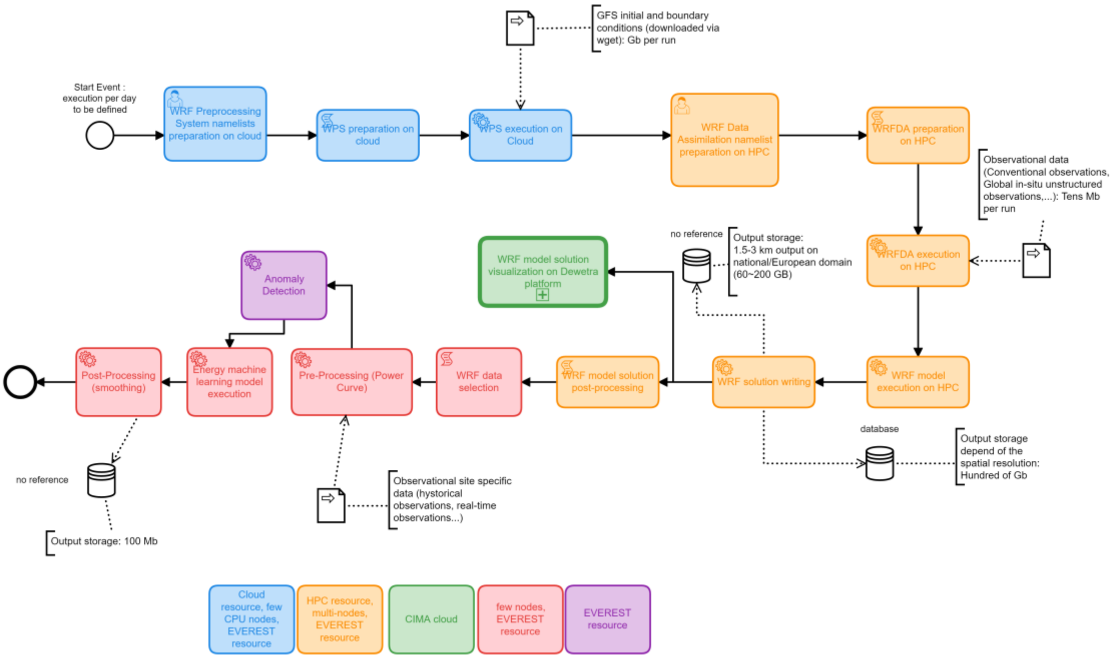

Figure 2 - Schematization of the workflow corresponding to the renewable energy use case

As for **Step 1**, it concerns the deterministic and/or probabilistic (ensemble mode) meteorological prediction exploiting the model WRF. WRF is a mesoscale forecasting system designed for both research and operational applications, capable of operating at spatial resolutions from hundreds of meters to hundreds of kilometres [13]. In Figure 3, the specific WRF workflow is introduced.

The WRF model is executed as a task on high performance computing facilities after the preparation of initial and boundary conditions provided by the WRF Preprocessing System (WPS); in the EVEREST project, the WRF model is run considering the data assimilation system – i.e., the WRFDA component. The WPS task is executed on local, or cloud computing facilities and it allows processing NCEP GFS global circulation model data to generate input fields for the WRF model itself.

The WRFDA task is a flexible, state-of-the-art atmospheric data assimilation system that is portable and efficient on available parallel computing platforms:

WRFDA is a task executed on HPC computational projects and/or cloud computing facilities.
The WRF task is a community, state-of-the-art atmospheric system that is portable and efficient on available parallel computing platforms: CIMA exploits WRF regional simulations for various applications, e.g., CIMA is currently running two different WRF model instances for operational purposes: WRF-1.5km Open Loop (without data assimilation) and WRF-2.5km 3DVAR both over Italy.

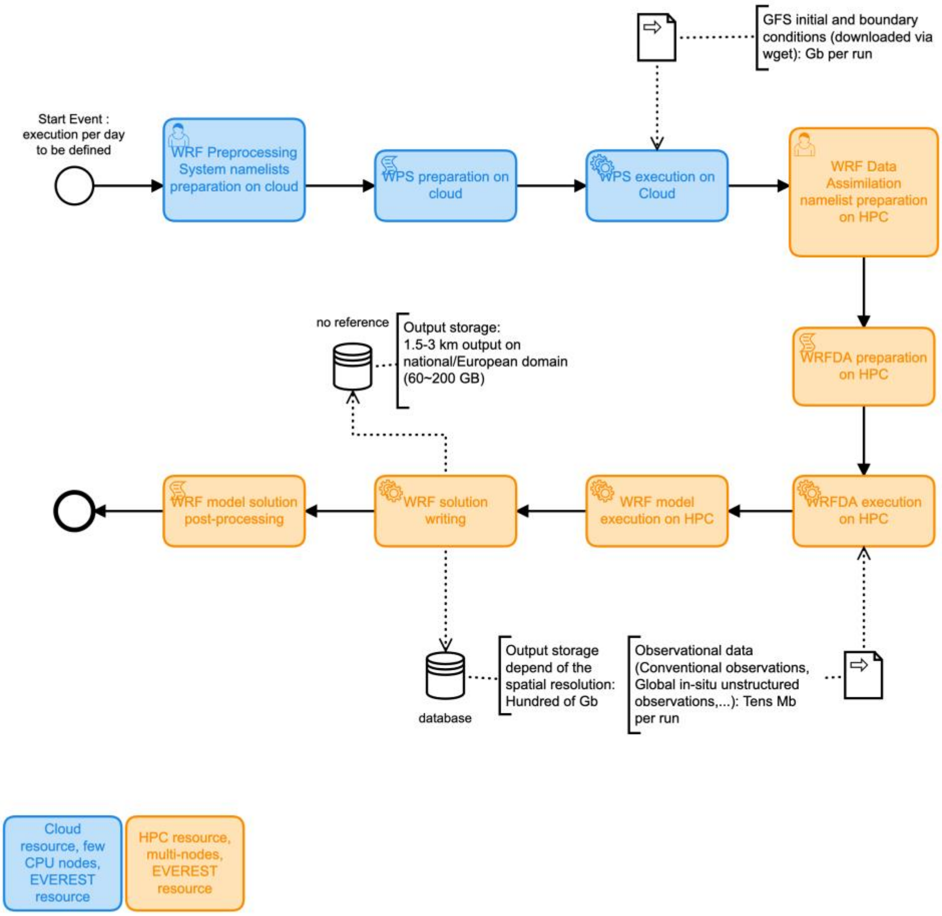

Figure 3 - Schematization of the WRF run

As for **Step 2**, it concerns with the pre-processing phase of the site-specific data mainly based on the deterministic prediction of hourly energy generation; the model implements turbine power curve by manufacturer to convert wind speed in power generation. The pre-processing is applied on all the data related to a specific-site, this means historical dataset, WRF output and observations. Indeed, the Turbine deterministic model concurs to maintain a high-quality level of the dataset used to train the ML algorithm – i.e., Step 3. Recently, to further reduce the data noise, smoothing techniques have been applied to pre-process the forecast wind speed.

As for **Step 3**, it concerns the implementation of the machine learning model to actually predict the energy production. The pre-processed site-specific data are used for training machine learning model, e.g. wind speed predictions, generation, availability etc. Several ML algorithms have been compared to identify the most suitable one for DUF use case, on the base of: dataset dimension, quality and features. Deep Learning, Recurrent Neural Network, XGBoost algorithms have been tested, but the best results in terms of prediction accuracy have been obtained by Kernel-Ridge with Gaussian Kernel.

Several training strategies of the ML algorithm, and different approaches have been tested and compared. Best results have been obtained using an incremental approach, adding new wind forecast data to increase the training period. The grid search cross-validation technique has been used to validate ML hyperparameters: the width of the Gaussian Kernel and the Ridge regularization parameter.

A post-processing phase has been also implemented thus to improve the shape of the energy forecast curve.

Figure 4 schematizes the workflow of both Step 2 and Step 3.

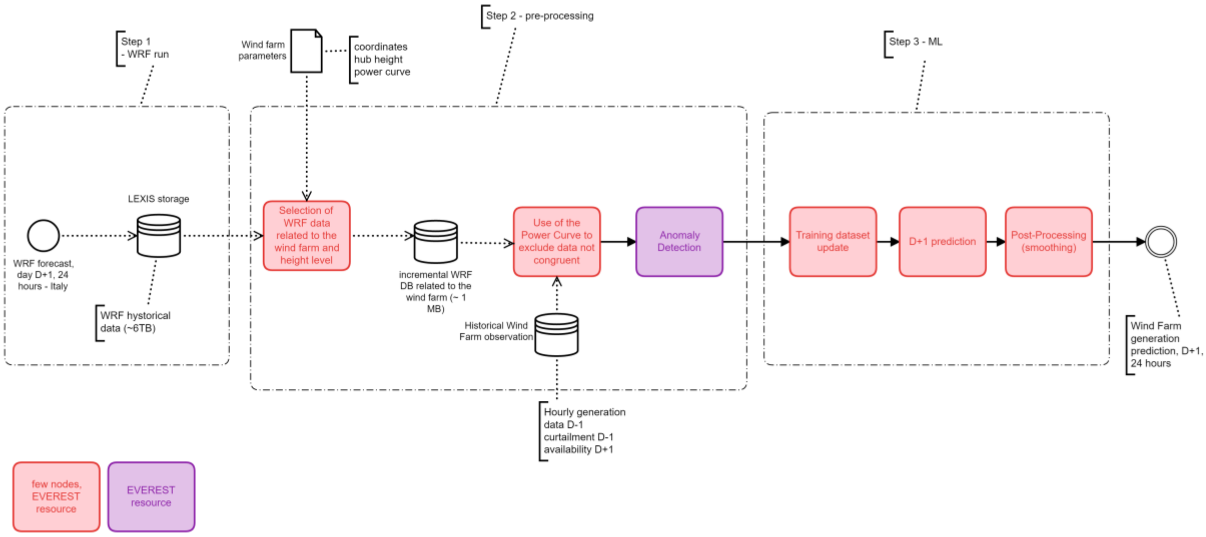

Figure 4 - Schematization of the energy production run

While managing the actual computation, the request for secure data transfer was further specialized and in particular anomaly detection has been selected because of its suitability. Techniques providing anomaly detection allow to identify data values which do not conform to the normal patterns of the data, which can help improve data validity. Anomaly detection will be applied to the wind farm data during pre-processing in order to help ensure data validity but also to improve the robustness of the system. On the detection of an anomaly, the technique will provide mechanisms to replace the anomalous value with one that would not be classified as anomalous. This replacement will ensure that there will not be missing data, thus it is expected to have a positive effect on the performance of the system.

The anomaly detection algorithm has been included in the Figure with a different colour since it represents a new requirement (Table 11).

Starting with data:

INPUT DATA

Table 1 ENERGY PRODUCTION use case – dataset GFS

ID	ITEM	DESCRIPTION
D1	Dataset name	Global Forecasting System (GFS)
	Dataset size	1 Gb (analysis and 48 hours boundary conditions at hourly temporal resolution and 25 km grid spacing)
	Dataset description	Initial and boundary conditions data
	Standards, format and metadata	Grib2 format

Table 2 ENERGY PRODUCTION use case – dataset Weather Underground stations

ID	ITEM	DESCRIPTION
D2	Dataset name	Wunderground stations
	Dataset size	Kbytes for one year for one station
	Dataset description	Personal weather stations data (2 m temperature, 10 m wind, 2 m relative humidity, surface pressure, rainfall intensity and rainfall depth) at hourly temporal resolution
	Standards, format and metadata	ASCII format

Table 3 ENERGY PRODUCTION use case – dataset Authoritative Weather Stations

ID	ITEM	DESCRIPTION
D3	Dataset name	Italy authoritative weather stations
	Dataset size	Kbytes for one year for one station
	Dataset description	Personal weather stations data (2 m temperature, 10 m wind, 2 m relative humidity, surface pressure, rainfall intensity and rainfall depth) at hourly temporal resolution
	Standards, format and metadata	ASCII format

Table 4 ENERGY PRODUCTION use case – dataset Radar mosaic

ID	ITEM	DESCRIPTION
D4	Dataset name	Italian Civil Protection radar mosaic
	Dataset size	12 Mb (24 hours data at hourly temporal resolution, 4 CAPPI levels)
	Dataset description	Radar reflectivity cappi at 2000, 3000, 4000 and 5000 m
	Standards, format and metadata	GeoTIFF format

Table 5 ENERGY PRODUCTION use case – dataset historical power generation

ID	ITEM	DESCRIPTION
D5	Dataset name	Historical power generation
	Dataset size	About 200 KB per wind farm (1 year)
	Dataset description	Wind farm specific on hourly basis [MWh] for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format

Table 6 ENERGY PRODUCTION use case – dataset historical power availability

ID	ITEM	DESCRIPTION
D6	Dataset name	Historical power availability
	Dataset size	About 2 MB per wind farm (1 year)
	Dataset description	Wind farm specific on hourly basis unavailability [MW or %] due to maintenance or failure, for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format

Table 7 ENERGY PRODUCTION use case – dataset historical curtailments

ID	ITEM	DESCRIPTION
D7	Dataset name and reference	Historical TSO curtailments
	Dataset size	About 100-200 KB per wind farm (1 year)
	Dataset description	Wind farm specific capacity limitations by TERNA (Italian Transmission System Operator); on hourly basis data [MW] for models training and calibration; minimum 1 year, possibly 2 years or more

	Standards, format and metadata	CSV format

OUTPUT DATA

Table 8 ENERGY PRODUCTION use case – dataset WRF model

ID	ITEM	DESCRIPTION
D9	Dataset name	WRF output
	Dataset size	2 Gb (cloud permitting grid spacing, 24 hours forecast at hourly time resolution, 500x500x50 grid points)
	Dataset description	2D and 3D meteorological fields, over different locations at hourly temporal resolution, for a selected period.
	Standards, format and metadata	NetCDF format

Table 9 ENERGY PRODUCTION use case – dataset energy prediction (ML)

ID	ITEM	DESCRIPTION
D11	Dataset name	Wind power forecast predicted with Machine Learning
	Dataset size	About 10 KB per wind farm (72 hr)
	Dataset description	Wind farm specific on hourly basis [MWh] for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format

Computational aspects:

Currently, computational aspects are well-known for models involved in Step1 (reported in the following Table), as models related to Step2 and Step3 are in design and/or development phases.

Table 10 WRF Model, including WPS and WRFDA

Name	Meteo prediction
Summary	The procedure required to compute a meteorological prediction adopting a Numerical Weather Prediction Model
Preconditions	Availability of a global circulation model

Basic course of the events	1. Select the initial and boundary condition from the global circulation model. 2. Prepare related ancillary data (DEM, land use, soil type, vegetation type) 3. Configure the model: grid spacing, time-step, timeframe, output intervals, physics, dynamics and numeric options, restart files, … 4. Download the observational data (e.g., radar, in situ weather stations) for subsequent data assimilation 5. Execute data assimilation through 3DVAR variational approach in rapid update cycle (RUC) mode 6. Start the weather forecast either deterministic (short range) or probabilistic (nowcasting).
Alternative paths	Single domain region or nested domains can be selected
Postconditions	Multiple rainfall, temperature, wind and pressure maps ready for analysis or further processing
Data accessed	Initial and boundary conditions from the global circulation model - Table 1 Observational data - Table 2, Table 3, Table 4
Data produced	Multiple rainfall, temperature, wind and pressure maps - Table 8
Local/remote resources	Local or remote HPC cluster
Resource capability	Hundreds of HPC cores for operational execution at cloud permitting / resolving grid spacing
Operating System and libraries	Linux, NETCDF, ZLIB, JASPER libraries

Table 11 - Pre-processing phase

Name	Step 2: pre-processing phase
Summary	The procedure processes the input data to feed the dataset related to a Wind Farm and to train the ML

	algorithms. The target is to maintain a high level of quality of this dataset to filter bad data.
Preconditions	The weather local forecast parameters from Step 1 and the observation data related to the wind farm and to the same day have to be available.
Basic course of the events	1. Select the wind farm. 2. Extract weather parameters forecast from the various source for this localization. 3. Apply the Power Curve filter to exclude outliers and apply smoothing methods to improve the shape of the timeseries.
Alternative paths	Anomaly Detection will be evaluated to filter input data and/or to substitute data excluded by pre-processing
Postconditions	Wind speed forecast timeseries and observation timeseries (i.e., hourly generation) ready for further processing
Data accessed	• Weather forecast from the WRF simulation – Table 8 • Dataset historical power generation - Table 5 • Dataset historical power availability - Table 6 • Dataset historical curtailments - Table 7
Data produced	A pre-processed dataset to be used for the training of the ML algorithms: wind speed timeseries and observation data related to the coordinates of the wind farm.
Local/remote resources	EVEREST resources
Resource capability	Few nodes
Operating System and libraries	Windows and Python Libraries

Table 12 Kernel-Ridge with Gaussian Kernel to predict energy production

Name	Step 3: Machine Learning prediction
Summary	The procedure requires to compute the prevision of the energy produced by a Wind Farm despatched by Duferco Energia on daily basis

Preconditions	Availability of the weather local forecast from Step 1 and the post-processed data from Step 2
Basic course of the events	1. Data collection from Step 2 2. Model training 3. Model forecast 4. Forecast smoothing
Alternative paths	-
Postconditions	WRF prediction related to D+1 will be integrated in the training set on the next day
Data accessed	• Weather forecast from the WRF simulation, pre-processed in Step 2 • Observation data pre-processed in Step 2
Data produced	24H point forecast of energy production - Table 9
Local/remote resources	Local or remote cloud server LEXIS (EVEREST HPC and FPGA accelerated resources)
Resource capability	Few nodes
Operating System and libraries	Windows and Python.

Based on the current knowledge, the meteorological model execution bottlenecks are located in the microphysical and radiation parameterizations, which deserve the main attention in terms of acceleration.
Acceleration would be considered also for Step 3, thus, to improve the time efficiency of the training phase of the ML algorithm.

4 Air-quality monitoring

Every year, new publications show the impact of air quality on the health over world. The latest figures from the World Health Organization show that air pollution kills an estimated seven million people worldwide every year. WHO data shows also that 9 out of 10 people breathe air containing high levels of pollutants, and that the economic impact of this pollution on health is estimated to 5.7K billions of dollars per year, [14], [15].

The use case concerns the forecast of air quality impact of atmospheric releases from an industrial site. In case of releases with an environmental or health impact around the site, the site can deploy different approaches to reduce or suppress this impact. On key element is the quality of the weather forecast in terms of wind direction, wind speed, etc. The objective of EVEREST is double: (i) increase the confidence of this forecast and (ii) reduce the simulation time in order to let time to site to adapt its process. For point (i) this will base on assimilation of observation data in the weather simulation and apply machine learning approach to correct errors of this deterministic weather forecast compared to local measurement on site. For point (ii) the use of hybrid computing with FPGA execution for weather simulation is expected.

The baseline implementation of the use case followed the path described in D2.1 without relevant changes to the workflow described in D2.1. The update of the use case only leads to remove one dataset, i.e., GFS has not been employed, due to the possibility to access the IFS dataset.

4.1 Description of the application steps and related pipeline

This use case relies on three main steps to simulate air quality impact prediction:

- Step 1: Compute deterministic (short-range) and/or probabilistic (very-short range and nowcasting) meteorological prediction with the WRF model. Data assimilation procedures are applied to force computation through atmosphere observations in order to improve weather prediction. Due to the high computational and memory requirements, HPC resources are mandatory to run the simulations. This step concerns a large domain simulation (for example France) and is then common to different industrial sites inside this domain.
- Step 2: Combine the deterministic weather forecast with another deterministic weather forecast and local measurement by the use of machine learning approach. This step is done for each industrial site inside the simulation domain of the step.
- Step 3: Compute air quality impact forecast for each industrial site, based either on the results obtained from the step 1 or the step 2.

Additionally, for Step 2, a training phase from a set of historical forecasts in past (3 or 6 months) is needed; therefore step 1 has been performed in such time range to provide a solid training set for the machine learning approach.

4.2 Data processed in the pilot

In the following, the description of data ingested and produced for each step is proposed.

4.2.1 Input dataset

For Step 1, the input data are:

i) Global weather forecast simulation from ECMWF (IFS 0.1 Degree Europe Forecast) available at https://www.ecmwf.int/en/forecasts/access-forecasts to initiate values and forcing at the border of the simulated domain

ii) Global observation weather data for assimilation procedure (e.g., weather underground stations)

iii) Some additional data such as soil/humidity temperature, etc.

For Step 2, the input data are:

i) The output from the step 1: weather forecast simulation.

ii) Other deterministic weather forecast simulations as the IFS forecast used as input of Step 1, but also GFS and NUMTECH operational weather forecast (3km resolution of France) at the location of the industrial site.

iii) Local observation weather data from the industrial site.

For Step 3, the input data are:

i) Either the output from the step 1 (weather forecast simulation) or the output from the step 2 (ensemble aggregation from various deterministic weather forecast downscaled and corrected by local observation of industrial site).

ii) Forecast of emission data of the industrial site for the air quality simulation.

iii) Fixed data to configure the air quality simulation for each industrial site (topography, surface roughness, buildings of industrial site, …).

4.2.2 Output dataset

There are two output datasets:

i) Weather forecast over a France domain from step 1.

ii) Air quality forecast for a limited domain (10km side) for each industrial site from step 3.

4.3 Computational resources required for the application

Figure 5 proposes a schematic representation of the workflow corresponding to the three steps introduced in Section 4.1. In the remaining part of this section, the three steps and related computational aspects are discussed.

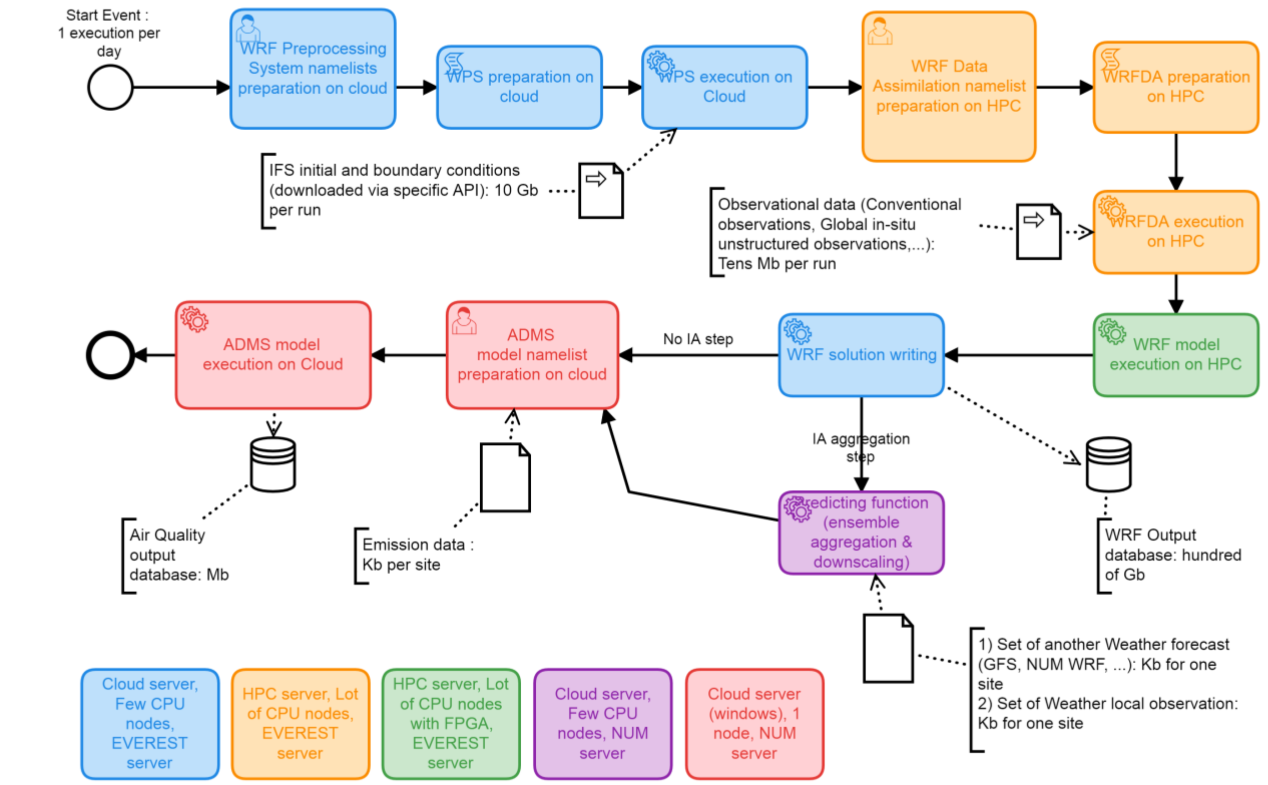

Figure 5 - Air Quality Monitoring use case workflow

As for **Step 1**, it concerns with the deterministic meteorological prediction exploiting the model WRF (Weather and Research Forecasting), the specific WRF workflow has been already discussed in Section 3.3 and Figure 3.

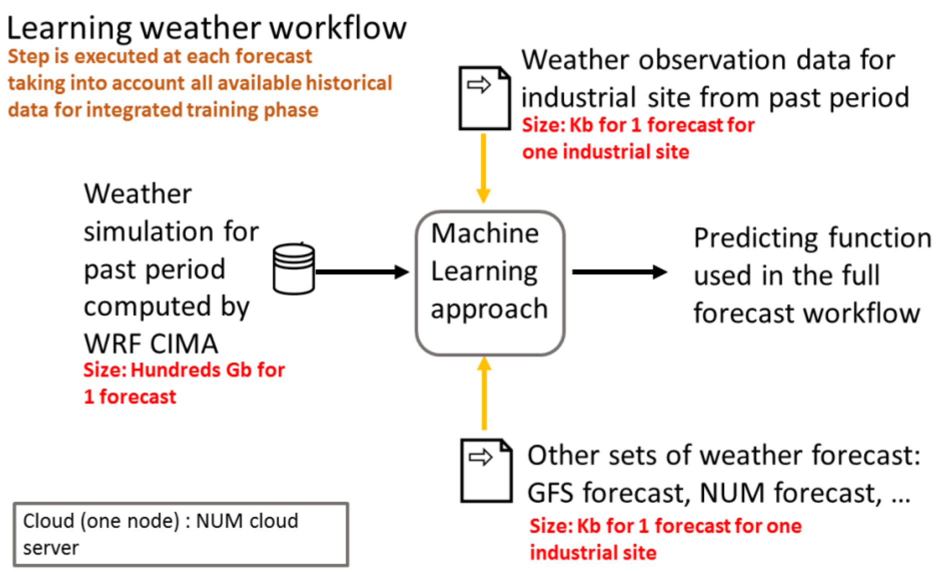

Figure 6 - Schematization of Step 2

As for **Step 2**, the NUMTECH machine learning method is executed in order to compute the correction prediction function to apply on deterministic weather forecast at the location of each industrial site. It requires a set of historical forecasts in past (3 or 6 months) obtained through the proper run of Step 1. Step 2 is schematized in Figure 6.

Step 3 is based on the commercial ADMS air quality model (https://www.cerc.co.uk/environmental-software/ADMS-model.html) two tasks are required:

- The preparation of weather data from Step 1 or Step 2
- The execution of ADMS (windows executable)

ADMS 5 [16] is a commercial code developed by CERC (http://www.cerc.co.uk) and NUM is the distributor of ADMS for the French speaking countries around the world and others, participates to its development and is an expert user of ADMS. ADMS 5 is an advanced dispersion model used to model the air quality impact of existing and proposed industrial installations. It is a new generation Gaussian plume air dispersion model, which means that the atmospheric boundary layer properties are characterised by two parameters: the boundary layer depth, and the Monin-Obukhov length.
It was originally developed for regulatory authorities in the UK. Its many features include allowance for the impacts of buildings, complex terrain, coastlines and variations in surface roughness; dry and wet deposition; NOx chemistry schemes; short term releases (puffs); calculation of fluctuations of concentration on short timescales, odours and condensed plume visibility; and allowance for radioactive decay including γ-ray dose.
Typical applications include:

- assessment of modelled air pollution concentrations against air quality standards and limit values including those from WHO, EU, UK, USA and China,
- planning/permitting, e.g., under Industrial Emissions Directive or Environmental Permitting Regulations,
- stack height determination,
- odour modelling,
- environmental impact assessments and
- safety and emergency planning.

As indicated in deliverable D2.1, WRF execution will be performed on EVEREST infrastructure (IT4I/IBM) and the ADMS execution on NUMTECH infrastructure.
Concerning the IA execution, there were two possible choices: the execution on EVEREST infrastructure or the execution on NUMTECH infrastructure, the latter solution has been preferred. Actually, this choice allows to exploit the developed API to exchange data between the EVEREST and the NUMTECH infrastructures, thus providing the demonstration that (and how) the EVEREST platform is able to scale to a global level by connecting rapidly and in secure way with external servers and resources.

Furthermore, the request for secure data transfer is stressed; as for the previous use case, anomaly detection techniques will be applied to improve the robustness of the system as well as to preserve data validity.

4.4 An analysis of requirements

In the following, data and models are described using a schematic approach.

Starting with data:

INPUT DATA

Table 13 Air quality use case – dataset IFS

ID	ITEM	DESCRIPTION
D15	Dataset name	Integrated Forecasting System (IFS)
	Dataset size	3 Gb (analysis and 48 hours boundary conditions at hourly temporal resolution and 9 km grid spacing)
	Dataset description	Initial and boundary conditions data
	Standards, format and metadata	Grib2 format

Table 14 Air quality use case – dataset Weather Underground stations

ID	ITEM	DESCRIPTION
D16	Dataset name	Wunderground stations
	Dataset size	Kbytes for one year for one station
	Dataset description	Personal weather stations data (2 m temperature, 10 m wind, 2 m relative humidity, surface pressure, rainfall intensity and rainfall depth) at hourly temporal resolution
	Standards, format and metadata	ASCII format

Table 15 Air quality use case – dataset local weather observation from industrial site

ID	ITEM	DESCRIPTION
D17	Dataset name	Local weather observation from industrial site
	Dataset size	Kbytes for one year
	Dataset description	Weather parameters data (2m temperature, 10 m wind, …) at hourly temporal resolution measured at local site of each industrial site

	Standards, format and metadata	Binary format

Table 16 Air quality use case – Emission data from industrial site

ID	ITEM	DESCRIPTION
D18	Dataset name and reference	Emission data from industrial site
	Dataset size	Kbytes for one year
	Dataset description	Emission parameters data (pollutants concentrations at emission, emission temperature, velocity of release, ...) at hourly temporal resolution measured for the different sources of each industrial site
	Standards, format and metadata	Binary format

Table 17 Air quality use case – NUMTECH weather forecast at industrial site

ID	ITEM	DESCRIPTION
D19	Dataset name	Deterministic NUM weather forecast for each industrial site
	Dataset size	Mbytes
	Dataset description	Weather parameters forecast data (2m temperature, 10 m wind, ...) for each industrial site at hourly temporal resolution for a selected period
	Standards, format and metadata	Binary format

OUTPUT DATA

Table 18 Air quality use case –WRF model output

ID	ITEM	DESCRIPTION
D20a	Dataset name	WRF output
	Dataset size	2 Gb (cloud permitting grid spacing, 24 hours forecast at hourly time resolution, 500x500x50 grid points)
	Dataset description	2D and 3D meteorological fields, over different locations at hourly temporal resolution, for a selected period.
	Standards, format and metadata	NetCDF format for full dataset, and Binary for extraction for one site

Table 19 Air quality use case – Aggregated weather forecast output

ID	ITEM	DESCRIPTION
D20b	Dataset name	Aggregated Weather forecast
	Dataset size	Kbytes
	Dataset description	1D meteorological field, over one location, at hourly temporal resolution for forecast of 48 hours.
	Standards, format and metadata	Ascii format

Table 20 Air quality use case - ADMS model output

ID	ITEM	DESCRIPTION
D21	Dataset name	ADMS output
	Dataset size	1 Mb (24 hours forecast for a 10km*10km grid and various pollutants)
	Dataset description	2D air-quality concentrations over different locations (France and Italy) at hourly temporal resolution, for a selected period.
	Standards, format and metadata	Ascii format

Computational aspects:
As for the Step 1, the computational requirements of the WRF model have been already presented in Table 10.

Table 21 Air Quality use case – Machine Learning model

Name	Step 2: machine learning prediction
Summary	The procedure required to compute an ensemble local weather forecast at a specific localisation from various deterministic weather forecast datasets
Preconditions	Availability of a weather local forecast from Step 1 and others weather forecast
Basic course of the events	1. Select the industrial localisation. 2. Extract weather parameters forecast from the various source for this localisation. 3. Download local weather observation at this localisation.

	4. Execute machine learning approach.
Alternative paths	Number of deterministic weather forecasts available
Postconditions	Multiple weather parameters ready for further processing
Data accessed	• Weather forecast from the global forecast simulation - Table 13 • Weather forecast from the WRF simulation - Table 8
Data produced	1D values of various weather parameters for the localisation of an industrial site
Local/remote resources	Local or remote cloud server
Resource capability	1 node
Operating System and libraries	Linux

Table 22 Air Quality use case – ADMS Model

Name	Step 3: air quality forecast
Summary	The procedure required to compute an air quality prediction using a numerical air quality dispersion model
Preconditions	Availability of a weather local forecast from Step 1 or Step 2
Basic course of the events	1. Select the local weather forecast data from Step 1 or Step 2 2. Prepare emission forecast data 3. Configure the execution for the period from fixed configuration - grid output, parameter to produce, …) 4. Start the execution.
Alternative paths	Use of weather data from Step 1 or Step 2
Postconditions	Visualization of the outputs and further processing (KPI evaluation)
Data accessed	Weather forecast from the WRF simulation - Table 8
Data produced	2D concentrations maps of different pollutants according to the industrial site - Table 16
Local/remote resources	Local or remote cloud server

Resource capability	1 node
Operating System and libraries	windows

The most intensive part of the air quality use case is the WRF execution on HPC servers, moving kernels presenting the main bottlenecks on accelerated resources improves the overall performance. Further improvements would be achieved speeding up machine learning prediction.

5 Traffic modelling

Cities of today face ever-growing problems with traffic congestions. In Europe, urban traffic congestion costs the EU nearly 1 percent of its annual GDP and accounts for 40 percent of all CO_2 emissions. Statistically, driving in congested conditions increases CO_2 emission by 80%. Travel times in mega cities during rush hours are on average increased by 50 to more than 100%. And yet, the concentration of people in cities continues to grow. In Europe, the percentage of citizens living in urban areas will rise from 73% in 2010 to 80% in 2050 [17]. Smart cities should therefore continually validate their transport infrastructure whether it satisfies the mobility needs of their residents at a sufficient level of comfort. To evaluate the quality of the transport infrastructure, traffic models are used to characterize the road network by quantification of the following macroscopic parameters: traffic flow, density, and average speed (so called 3D traffic model), ideally for all road segments and for each time interval (e.g., every 10 minutes) in a day, week, and year. This model needs to be constantly updated in time and refined based on newly obtained statistics and measurements.

Traffic modelling use case is the eco-system comprised of a mixture of processing and storage components, which pose challenges both on big data and AI computational processing. Big data sets need to be transferred across components, efficiently stored, and readily available for processing. AI computations need to be executed fast and economically so that it is cost efficient, and it meets a daily processing cycle because each day new data enter the eco-system. Moreover, eco-system needs to be permanently responsive for routing requests.
The baseline implementation slightly modified the workflow designed in D2.1; the foreseen workflow has been reduced in the number of steps to be performed: actually, two steps have been merged in a single one, while previous Step 4 has been removed since it is qualitatively inferior to current Step 4, while technologically equivalent. The updated version is discussed in the remaining of the section.
Furthermore, during the FPGA mapping of the first experiments, it has been recognized (in compilation process) the necessity to support users in expressing details about the parallelism aspects of the single steps and/or components. To facilitate this elicitation in flexible and convenient manner, high-level forms have designed as discussed in D2.5; the general goal of the requirement in to lower the barrier in the exploitation of the EVEREST SDK while achieve targeted performance goals.

5.1 Description of the application steps and related pipeline

In Figure 7, the main traffic monitoring eco-system is presented. It is composed of 4 processing steps. Each step represents a workflow consisting of data input, processing component(s), and data output. The workflows communicate over data sets. At the boundary of the eco-system is the **SYG** backend client, which first, enters data into system, and second, uses the routing service.

The steps in place are:

- **Step 1** Big data collection: processes daily incoming floating car data (FCD) and stores them in a suitable form for a subsequent use.
- **Step 2** Traffic simulator: produces reality close simulation scenario helped by road speed data output from Step 1 and origin-destination matrix. As an output it calculates traffic view for each road element for an arbitrary time instance (3D model). At the same time generates training sequences for learning traffic prediction on city intersections.
- **Step 3** Model training: ML Training of traffic prediction model based on training sequences generated by Step3.
- **Step 4** Routing: process, which uses the model knowledge learned in Model training to provide efficient route calculation for real-time user requests.

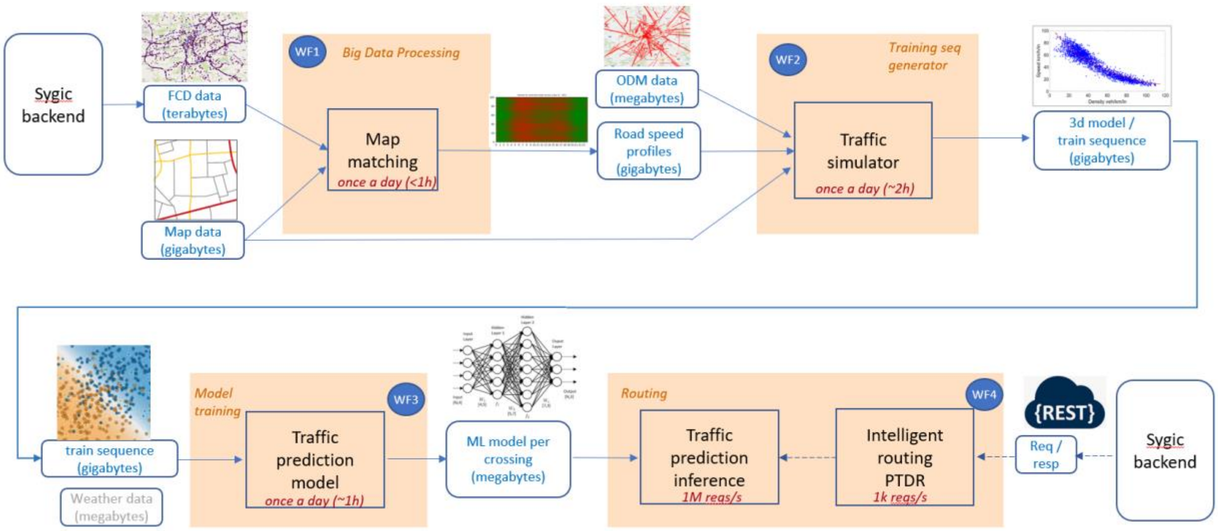

Figure 7 - Traffic monitoring eco-system

5.2 Data processed in the pilot

Traffic modelling eco-system operates with several data sets, which can be classified as big data (requiring short-term or long-term storage), configuration data, results, or real-time data.

5.2.1 Input dataset

The following defines input data for all steps.

Step 1:

- **Raw FCD data input**: FCD big data arriving daily from an external system, i.e., **SYG** backend, defined in terms of vehicle speed at a given time instance and GPS position.

- **Map data input**: these data define road network topology of a given city; it is subject to occasional updates.
  Shared input for Step 1 and Step 2

Steps 2:

- **Road speeds**: pre-processed sequence of (envisioned) last 100 days of raw traffic FCD data. It is the output of the step 1, which is a big data requiring long term data storage.
- **ODM data input**: data characterizing traffic flow situation in a city between critical points of traffic infrastructure, defined in terms of number of vehicles on a route A-to-B within a time-period, it is subject to occasional updates.
- **Map data input**: these data define road network topology of a given city; it is subject to occasional updates.
  Shared input for Step 1 and Step 2
- Various configuration data.

Step 3:

- **Traffic training sequence**: generated sequence of traffic data (speed, density) per each road element of the traffic network produced by Traffic simulator, i.e., the output of the step 3, which is a big data requiring short term storage.

Step 4:

- **ML traffic model**: coefficients representing traffic prediction models of a large set road elements.
- **Routing requests:** routing start and destination requests, which trigger the execution of the Intelligent routing algorithm.

5.2.2 Output dataset

The following are relevant output data.

Step 1:

- **Road speeds**: pre-processed sequence of raw traffic FCD data, meant primarily as intermediate data.

Step 2:

- **Traffic training sequence**: time sequence of traffic data per each road element, meant primarily as intermediate data.
- **Extended road speeds and densities - 3D model**: average characterisation of road segments in terms for speeds and densities for the 15-minutes time instances across the day.

Step 3:

- **ML traffic model**: coefficients representing traffic prediction models of a large set road elements, meant primarily as intermediate data.

Step 4:

- **GPS output routes**: defined as GPS trajectories calculated from system input requests given as origin and destination. It is the output of the Step 6, meant as system output resulting from real-time calculation of Intelligent routing.

Intermediate data are typically generated by some workflows and consumed by others, yet they can be treated as public output too.

5.3 Computational resources required for the application

There are four key computations in the system shared by 6 steps described in previous chapter:

Step 1:

- **Big data processing**
  processes daily incoming FCD data from an external world and stores them into a structure suitable for filtering and a quick multi-pass access scheme for other components in the traffic modelling eco-system. Next, it will execute map-matching algorithm to convert the FCD data into road speeds on an underlying road network. It is a batch processing, runs daily, and should be executed in less than an hour.

Step 2:

- **Traffic simulator**
  runs the microscopic traffic simulation of a configured city by moving all vehicles on roads based on calculated routes from origin-destination matrix. The simulation generates 3D traffic data for each major road element of a road network as well as training sequence for speed predictions. Is it a batch processing, runs daily for multiple conditions.

Step 3:

- **Traffic prediction model training**
  through machine learning algorithms it calculates prediction models for many road elements exploiting the training sequence generated by Traffic simulator.
  Is it a batch processing, runs daily for multiple training sequences.

Step 4:

- **Routing**
  computes efficient route based on the origin-destination request. It is

tied to traffic prediction inference computation, which uses road trained models to calculate prediction for a next hour. Routing is a real-time processing of throughput minimally 1000 requests per second. Traffic prediction is the macro-request calculating the prediction for all the roads in a city at once.

5.4 An analysis of requirements

Starting with data:

The following are important data sets, which imply some storage requirements.

Table 23 Traffic monitoring use case – FCD data input

ID	ITEM	DESCRIPTION
D22	Dataset name	**FCD data input**
	Dataset size	gigabytes
	Dataset description	Raw data of vehicle speeds and positions
	Standards, format and metadata	CSV format

Table 24 Traffic monitoring use case – Map data input

ID	ITEM	DESCRIPTION
D23	Dataset name and reference	**Map data input**
	Dataset size	gigabytes
	Dataset description	Road network topology
	Standards, format and metadata	Annotated graph

Table 25 Traffic monitoring use case – ODM data input

ID	ITEM	DESCRIPTION
D24	Dataset name	**ODM data input**
	Dataset size	megabytes
	Dataset description	Traffic flow characterization on a set of origin-destination points
	Standards, format and metadata	CSV format

Table 26 Traffic monitoring use case – organized FCD data storage

ID	ITEM	DESCRIPTION
D25	Dataset name	**Road speeds**
	Dataset size	Up to terabytes
	Dataset description	Road speeds organized in a temporal and geo hierarchical structure

	Standards, format and metadata	Nosql database

Table 27 Traffic monitoring use case – Traffic training sequence

ID	ITEM	DESCRIPTION
D26	Dataset name	**Traffic training sequence**
	Dataset size	megabytes
	Dataset description	Training vectors for ML processing
	Standards, format and metadata	CSV format

Table 28 Traffic monitoring use case – ML traffic model output

ID	ITEM	DESCRIPTION
D27	Dataset name and reference	**ML traffic model**
	Dataset size	gigabytes
	Dataset description	Coefficients of multiple ML components
	Standards, format and metadata	CSV format

Table 29 Traffic monitoring use case – Traffic 3D profile output

ID	ITEM	DESCRIPTION
D28	Dataset name and reference	**Traffic 3D profile**
	Dataset size	gigabytes
	Dataset description	Speed & density & intensity of road segments
	Standards, format and metadata	CSV format

Computational aspects:

The characterization of the key four processing components envisioned during the project lifetime follows.

Table 30 FCD data processing

Name	**FCD data processing**
Summary	Convert raw data into well-organized structure
Preconditions	Availability of FCD data Access to suitable data storage
Basic course of the events	1. Open input data stream 2. Read chunk and store it 3. Filter out 4. Process through map-matching 5. Store into hierarchy
Alternative paths	-
Postconditions	-

Data accessed	Raw FCD data - Table 23 Map data –Table 24
Data produced	Road speeds - Table 26
Local/remote resources	EVEREST HPC and FPGA accelerated resources
Resource capability	Expected tens of cores
Operating System and libraries	Independent of OS

Table 31 Traffic monitoring use case – Traffic simulator

Name	**Traffic simulator**
Summary	Generates traffic 3D model and training sequence for learning traffic prediction model
Preconditions	Availability of FCD data, ODM data and map network Access to fast routing algorithm
Basic course of the events	1. Select the initial and boundary conditions for a particular time window 2. For each O-D pair calculate route 3. Simulate vehicle movements for a given time period on a given route 4. Save average flow statistics for each road
Alternative paths	Vehicle routes can be dynamically recomputed
Postconditions	-
Data accessed	ODM data - Table 25 Map data - Table 24 Road speeds - Table 26
Data produced	Traffic training sequence - Table 27 Traffic 3D profile – Table 29
Local/remote resources	EVEREST HPC and FPGA accelerated resources
Resource capability	Expected hundreds of computational cores, optionally with combination of thousands of accelerated cores
Operating System and libraries	Unix like OS

Table 32 Traffic monitoring use case – Traffic prediction training

Name	**Traffic prediction training**
Summary	trains prediction models from training traffic flow vectors
Preconditions	Training vectors available
Basic course of the events	1. Load latest model state 2. Train with updated training sequence for a number of epochs 3. Save new model state

Alternative paths	Bayesian inference ML model instead of CNN
Postconditions	Model provides prediction function
Data accessed	Traffic training sequence - Table 27
Data produced	ML traffic model coefficients - Table 28
Local/remote resources	EVEREST HPC/Cloud (GPUs) and FPGA accelerated resources
Resource capability	Expected thousands of accelerated cores
Operating System and libraries	OS with TensorFlow and Python support

Table 33 Traffic monitoring use case – Intelligent routing

Name	**Intelligent routing**
Summary	Calculates route from A to B on request
Preconditions	Availability of map network Access to traffic prediction model
Basic course of the events	1. Update cost functions of edges 2. Request prediction if not cached 3. Run routing algorithm
Alternative paths	-
Postconditions	-
Data accessed	Map data - Table 24, Traffic prediction model - Table 28
Data produced	GPS route trajectory per request
Local/remote resources	EVEREST FPGA accelerated resources
Resource capability	Automatic scaling based on current needs from one to thousands of accelerated cores or hundreds cloud cores allocated dynamically
Operating System and libraries	Unix like OS

All processing components are candidate to be accelerated.

6 Requirement analysis

Starting from the description of the specific use cases, it is possible to derive a set of common requirements to be taken into account in EVEREST framework.

Requirements are distinguished on computational and data aspects, classified in terms of priority levels and connected with EVEREST infrastructure and technologies. Actually, different priority levels have been defined for the development: "Must have" are the mandatory requirements to obtain the first working prototypes to have a baseline for optimization; "Should have" includes the requirements that are expected to target for optimizing the baseline and to achieve the overall EVEREST KPIs; finally, "Could have" are additional requirements needed to trigger extra functionalities that can further improve the results.

In order to preserve/improve the readability of the Section, newly defined requirements use an incremental ID starting from the ones defined in D2.1, i.e. starting from REQ15. They are listed in proper group of requirements although the ID is not consecutive – as was at M6.

EVEREST target platform has to be considered as the set of resources that the project accesses through collaboration with other projects, the participation to IT4I Open Access Competition, https://www.it4i.cz/en/for-users/computing-resources-allocation, partially also for Directors' discretion EVEREST project, and the IBM FPGA resources on premises. EVEREST technologies can be considered the set of tools and solutions that are currently supposed to be employed in the lifetime of the project.

REQ1: Resources for data and computationally intensive scientific computing - HPC computational resources corresponding to at least 200-300 cores, with at least 2Gb per core. This supports the execution of WRF predictions at cloud-permitting (2-3 km) grid spacing over synoptic scales domains (1000 km wide). WRF is at the base of two up to three use cases.

REQ2: Priority on the workflow(s) execution on target computing resources - Management of the execution of a set of EVEREST workflows on predefined times: one execution every day, four executions every day, by triggered event. Computation has a prescribed deadline in order to ensure the effectiveness and the practical impact of predictions. This is a common requirement to all the use cases.

REQ3: Resources for data intensive scientific computing - Execution of ensemble of cloud-permitting grid spacing simulations. At least 10 members are needed corresponding overall to 2000-3000 cores. At the moment, this is a requirement for two out of three EVEREST use cases.

REQ4 (updated): Processing elements with big data storage - Data storage elements provided with sufficiently large space (GBs) and flexible access scheme to computation elements. Supporting efficient multiple passes as well as random access over data as specifically required by traffic use case.

REQ5 (updated): Machine Learning runtime able to run on heterogeneous hardware - Envisioned are the functions to create deep neural network structure, inference calculation and gradient descent optimization. Supported within the scope of tensor flow python coding.

REQ6: Services to ingest real-time input data from external sources - Real-time data from external sources concur in the definition of the input dataset. As an example, **SYG** external system will need to feed in a workflow with big chunks of data on a daily basis. Suitable protocol and storage should be prototyped. This is a common requirement to all the use cases.

REQ7: Services for real-time request/response scheme - Some calculation will require real-time interaction with external subsystems, such as the traffic monitoring use case where there is the need to take into account client requesting route calculation. It is necessary to improve the throughput.

REQ8: Orchestration at a single location with heterogeneous hardware - Traffic eco-system will comprise of several cross-dependent workflows. The workflows execution needs to be orchestrated within one location so that the system operates smoothly and uses efficiently the available hardware, e.g., FPGAs.

REQ9: Processing deployment flexibility - The deployment mechanism should provide flexibility for targeting any of the technology. With reference to the weather modelling-based use cases, the WRF model will be potentially executed on HPC or FPGA system. The same argument holds for the FPGA execution of components of Machine Learning inference workflows. This is a common requirement to all the use cases.

REQ16 (new): Compilation flow with kernel parallelisation - Mapping to target architecture with the convenience and flexibility on parallelism hinted by user. Applicable for C++, Python source inputs.

Table 34 Classification of computational requirements in terms Type, Priority, Technologies and Methodologies

Id	Type	Priority	Technologies/ methodologies used in EVEREST
REQ1	HPC WRF	Must have	• EVEREST HPC and Cloud Infrastructure; • EVEREST FPGA nodes; • EVEREST DSLs and compilation-synthesis flow.

REQ2	Timing constraints	Must have	• EVEREST DSLs and compilation-synthesis flow (for supported kernels, cf. D2.5).
REQ3	HPC Ensemble	Should have	• EVEREST HyperQueue - if we can describe this as DAG of specific tasks (e.g., simulations, etc.); • EVEREST virtualisation of compute nodes (includes API for HyperQueue) for different accelerators support.
REQ4	Optimization	Must have	• EVEREST HPC and Cloud infrastructure; • EVEREST FPGA nodes;
REQ5	Optimization	Must have	• EVEREST HPC and Cloud infrastructure; • EVEREST FPGA nodes; • EVEREST SDK (integration with ML framework and co-optimization of selected kernels); • EVEREST virtualisation of compute nodes (includes API for HyperQueue) for different accelerators support.
REQ6	Streaming dataflow	Could have	• EVEREST HPC and Cloud infrastructure; • EVEREST FPGA nodes; • EVEREST SDK for Traffic use case (potential extension to dataflow languages as syntax on top of streaming frameworks).
REQ7	Optimization	Should have	• EVEREST HPC and Cloud infrastructure; • EVEREST FPGA nodes; • EVEREST DSLs and compilation-synthesis flow to accelerate computation. • EVEREST virtualisation of compute nodes (includes API for EVEREST HyperQueue) for different accelerator support;
REQ8	Optimization	Should have	• EVEREST HPC and Cloud infrastructure; • EVEREST FPGA nodes; • EVEREST HyperQueue; • EVEREST dataflow abstractions for optimizations across workflows (depends on REQ6); • EVEREST virtualisation of compute nodes (includes API for EVEREST HyperQueue) for different accelerator support.
REQ9	Programming convenience/ Optimization	Should have	• EVEREST HPC and Cloud resources; • EVEREST FPGA nodes;

			• EVEREST compilation-synthesis tool flow for transparent programming of different targets (for selected components, cf. D2.5); • EVEREST virtualisation of compute nodes (includes API for EVEREST HyperLoom) for different accelerator support.
REQ16	Programming convenience/ Optimization	Should have	• EVEREST DSL, C++, Python compilation-synthesis flow • EVEREST DOSA and Olympus synthesis towards FPGA

REQ10: Data anonymity - Locations and seasonal periods of observational data related to the pilot use cases (e.g., wind farms positions and compositions, air quality data, traffic data) must be anonymized for sake of industrial IPRs. This is a common requirement from all the use cases.

REQ11: Data ingestion - Observational data either for assimilation purposes and for workflows validation scope have to adhere to a minimum set of principles (format, size, time and spatial resolution, quality control and quality assurance) to be further ingested and used into the different workflows (e.g., new radar data must be in GeoTIFF etc). At the moment, this is a requirement from two out of three EVEREST use cases.

REQ12: Data transfer among the different model instances/steps - Efficient data transfers among workflows – i.e., different software components and computational nodes. As an example, large data volume at cloud permitting grid spacing must be moved for subsequent post-processing using ML algorithms. This is a common requirement from all the use cases.

REQ13: Access to historical data - Data storage needed for historical data at the base of different modelling steps: initial and boundary conditions for the weather modelling instances, to train and calibrate the ML steps, to validate the overall workflow(s). This is a common requirement from all the use cases.

REQ14: Mechanism for uploading and downloading data - Workflow's output such as weather prediction, training sequences and model characterization (GBs) will be of interest for subsequent download, e.g., dissemination purposes, additional post-processing. Similarly, some computations will require handy data output, e.g., map topology (MBs). This is a common requirement from all the use cases.

REQ15: Shared data storage - There is a need to share data among the overall workflow components. An example, for the weather-based use cases, the observational data are required both for the WRF model and the AI models. The same happens for the traffic eco-system that has to share global state between particular components running on a single location. It is required an effective distribution of the shared global state targeting different architectures.

REQ17 (new): Data integrity - The integrity of the data must be ensured. To this end, the use of appropriated technique such as the ones based on anomaly detections has been proposed, since such techniques will also allow to catch spurious data which could alter the performance in some use cases (for example, in the used case of the energy production).

Table 35 Classification of data requirements in terms Type, Priority, Technologies and Methodologies

Id	**Type**	**Priority**	**Technologies/ methodologies used in EVEREST**
REQ10	Data Security	Must have	• Data must be passed to the EVEREST environment already anonymized, it could be however interesting to explore the possibility to integrate into the parser for loading the data a check to verify the anonymity.
REQ11	Data interoperability	Could have	• Control of the rules on input data could be integrated into EVEREST parser for loading the data.
REQ12	Efficient data transfer	Coould have	• EVEREST HPC and Cloud infrastructure connected with appropriated communication channels
REQ13	Large dataset/ HW capability	Must have	• EVEREST HPC and Cloud infrastructure; • EVEREST FPGA nodes.
REQ14	Data transfer	Must have	• EVEREST HPC and Cloud infrastructure should support access to data form authorized parties.
REQ15	Optimization	Could have	• EVEREST HPC and Cloud resources; • EVEREST FPGA nodes; • EVEREST HyperQueue.
REQ17	Data Security	Must have	• Data integrity will be ensured with appropriated techniques such as anomaly detection

7 A roadmap to exploit the EVEREST technologies

Leveraging on the specific requirements posed by the use cases, it is possible to derive the results achievable by the means of the EVEREST programming environment and computational platform. In the following, for each use cases, a set of objectives are defined and discussed in terms of methods required to achieve such goals, related priority, and the of metrics/KPIs that can possibly support the evaluation of the effectiveness of the EVEREST ecosystem.

7.1 Renewable-energy prediction

Although renewable energy is considered to be the most promising alternative to fossil fuels, it also brings additional unpredictability which threatens the reliability and stability of energy systems, especially with the increasing large-scale integration of renewable energy. On the one hand, renewable energy exhibits strong volatility, intermittency and randomness, which will undoubtedly increase the reserved capacity of the electric energy systems, thereby increasing the cost of power generation. On the other hand, the use of renewable energy involves a large number of power electronics, which reduces the rotational inertia of the power system and thus reduces the stability margin of the system. Therefore, renewable energy forecasting as an effective measure is essential for mitigating related uncertainties, which is conducive to planning, management and operation of electrical power and energy systems. However, accurate renewable energy forecasting remains a challenging task due to the intermittent, chaotic and random nature of renewable energy data intrinsically associated to the corresponding highly fluctuating predictability of the atmospheric forcings, namely wind speed and direction as well as solar radiation.
Forecasting models are used across the entire value chain of wind power, in generation, transmission, and load. For generation purposes, the forecasts can be used for turbine control, management of wind farms, feasibility studies, maintenance planning, and electricity trading. In the transmission network, forecasting can be used by the wholesale market for market clearing and electricity trading, and by TSO/DSO for grid operation, economic dispatch, reserve requirement decisions, and other grid decisions.

In this complex context and based on the state-of-the-art, **DUF** has the ambition to implement **a proprietary service** aligned and possibly outperforming the state of the practice in the prediction of wind power. In the following we provide a list of possible steps at the base of the design phase currently in place.
Firstly, the choice of forecasting models is not trivial and depends on the intended applications and hence the time horizon, [9]. Most of the literature analysed is focused on medium-term forecasting from 1 to 72 hr ahead, and the majority of the literature in this horizon has been published in recent years. Combining models across all forecasting horizons, most of literature shows how

the timescale or time horizon affects the use of certain types of methods such as timeseries based models are only present in nowcasting (1-6 hr ahead) and short-range forecasting (6-48 hr ahead). **The design of the application is predominantly focusing on the first-time range.**
Timescale of the forecast has an obvious impact on the processing times required by the model; current and future energy markets require a continuous update of the wind power forecast of the hour ahead, so this is a basic constraint in timing of assimilation and processing data. **The exploitation of the computational heterogeneous resources shared in the EVEREST project represents the enabling infrastructure to tackle such intensive application**.
A further and mandatory consideration relates with the accuracy of the wind power forecast: its estimation and a clear path to evaluate it are necessary steps to validate the entire application, [11]. According to **DUF** experience, NMAE (Mean Absolute Error normalized by power capacity) is the most used to evaluate the accuracy of the forecast, in comparison with the other commercial services available in the market. NBIAS (normalized bias) could be useful to intercept systematic errors. **The target is to achieve, with respect to such metrics, values that are comparable with the best prediction used by DUF.**

Many experimentations have been performed to actually develop and deploy the baseline version of the energy production use case. However, the main objectives of the use case are not changed and only few more methods have been added. In particular, in Objective O1.2, the experimentation of ensemble approaches has been decoupled from the selection of algorithms and methods employed in the workflow. This choice is motivated by the fact that their priorities are deeply and strongly different, as emerged during the development phase.
Furthermore, it has been enforced the requirement about secure data transfer with the check of the possible anomalies occurred in the observation collection/transfer. This has been translated in a new method of the Objective O1.4.

Table 36 Desiderata envisioned of the Renewable Energy use case

Id	Objective	Method	Priority	Baseline	KPI	Notes
O1.1	The implementation of a DUF proprietary application for energy production prediction	Studying the state-of-the-art of scientific literature to understand the most interesting and promising trends	Must Have	N/A	Availability document with state of the art analysis	No baseline available at the beginning of the project, an assessment of the state of the art and SWAT analysis was needed
		Summarizing the state of the practice in industrial sector to be aligned and possibly outperform the current scenario	Must Have	N/A	Presence of a comparative analysis for the different methods and different providers	
		The design of a specific application workflow	Must Have	N/A	Presence of code deployed within the Everest environment	No baseline available at the beginning of the project, the use case have been developed from scratch
O1.2	The focus is on daily day ahead forecast (24 hrs day ahead) timeline *	Understanding pros and cons of different models and approaches	Must Have	ML algorithm Kernel Ridge with Gaussian Kernel without pre-processing and post-processing	Quality improvement of the Wind Farm Energy prediction	Power generation forecast accuracy
		The selection of the proper methods, metrics and algorithms to be implemented in the workflow for this temporal scale	Must Have	Comparison among different ML algorithms	Quality improvement of the Wind Farm Energy prediction	Comparison involved dataset dimension, quality and features; Deep Learning, Recurrent Neural Network, XGBoost algorithms have been tested, besides Kernel-Ridge.
		The evaluation of probabilistic ensemble approach	Could Have	Baseline prediction without ensemble model	Quality improvement of the Wind Farm Energy prediction	The execution of a small ensemble run of WRF simulation requires a quite huge amount of computational power. At this stage of the project, it was not possible to obtain such a number of FPGA-accelerated cores
O1.3	Effective run of the service in terms of computational performance efficiency with a specific focus on the WRF workflow	Evaluating the most intensive models and kernels (e.g. physics parameterizations) selected for WRF modelling for cloud-resolving (1-3 km grid spacing) applications	Must Have	WRF CPU version	% of WRF execution to be accelerated with EVEREST technology	Not all WRF can be accelerated, profiling of typical executions will be used
		The optimization of such kernels and models within the EVEREST platform	Must Have	Selected WRF-Kernel CPU version	Performance and energy Improvement	
		The exploitation of the EVEREST platform to support meteorological workflow	Must Have	WRF CPU version	Availability end-to-end WRF execution with accelerated modules	
O1.4	Prediction accuracy comparable to the state of the practice	Compare the EVEREST forecasting performance over selected periods and for selected wind farms in terms of deterministic and probabilistic forecasts	Must Have	N/A	Availability of End-to-end Workflow including data assimilation, WRF execution and Wind farm energy prediction	No complete workflow was available at the beginning of the project
		Implementation of adequate pre-processing components for observational data to be assimilated into the procedure	Must Have	N/A	Availability of the component	No data-assimilation component was available at the beginning of the project
		Implementation of adequate security policy during observation collection/transfer	Should Have	Workflow without Anomaly Detection module	% of injected anomalies that are detected and count of anomalies detected in historical data	The module will be used for detecting possible malicious data ingestion, and for cleaning historical data
		Exploiting of the EVEREST computing platform in support of the overall modelling workflow	Must Have	N/A	Availability of the End-to end execution of the energy prediction workflow using EVEREST hw/sw platform	FPGA exploitation at least of one component, complex orchestrator of the different component, data transfer of the daily in-situ data

* The WRF model is run once a day and this of course impacts the time-range of the forecast of the energy prediction, focused on the day ahead horizon. The possibility to execute the WRF more frequently leads to advancement also from this point of view.

7.2 Air-quality monitoring

Manage an industrial process in order to reduce its atmospheric impact is generally not possible in real-time and requires several hours to be adjusted. Moreover, identify meteorological conditions (in terms of wind, atmospheric stability, etc.) not suitable for air dispersion is not easy, since it is a complex phenomenon which depends on numerous factors, such as the height of the emission, its temperature and velocity release compared to the atmosphere, just to name a few.

Starting from meteorological conditions, they have a local nature. A condition can be then good for one source but not for another one, or good when we consider a distance below 1km from the source but not beyond. Anticipation of risky situations is then performed by numerical forecast simulations. **One key point is then the forecast of meteorological conditions.** Simulation at local scale in the vicinity of emissions sources means to simulate air dispersion at a distance which generally do not exceed 10 km. Most of the standard weather forecasts provided by National Weather Centers have a 5-10km spatial resolution. In addition, most of the observations assimilated in these forecasts correspond to surface observations from national weather network with a sparse spatial cover. For example, in France, the automatic stations are installed on airfield locations which are naturally far away from industrial sites, while local weather observations may exist at an industrial site level. **NUM** has the ambition to face against different points to reduce uncertainties related with meteorological predictions, with a focus on quality of the forecast, especially on wind direction and wind speed which are important for air-quality dispersion.

To do that, it was decided to: 1) Improve the spatial resolution to better represent local effect such as slopes, sea breeze, etc in the initial WRF execution. 2) Improve the assimilation part in the initial WRF execution by switching to a simple nudging of the observation to 3DVar approach. 3) In a second step, to apply a machine learning method consisting on an ensemble aggregation of various weather forecasts forced by local observations.

Quality first of the WRF forecast and in second time of the IA approach will be performed based on RMSE (Root Mean Square Error) for different weather parameters for a selection of industrial sites for which local weather observations are available.

Concerning the WRF execution, the consequence of the applied choices is naturally an increase of execution time. The exploitation of EVEREST platform with the support of the WRF **acceleration by FPGA is thus important to maintain a reasonable execution time and even to reduce it** (and to reduce it strongly on initial WRF workflow with no change on assimilation and spatial resolution). Provide forecast as soon as possible in the morning is a key point for industrial sites in their management procedures.

The impacts on air-quality forecasts will be evaluated through two metrics over a selected period: the reduction of false forecasted pollution events, and the increasement of true forecasted pollution events

Table 37 Desiderata envisioned of the Air-quality Monitoring use case

Id	Objective	Method	Priority	Baseline	KPI	Notes
O2.1	Improved spatial resolution, initialization and forcing of weather forecast	Pushing to the limits of meteorological predictions with a fine resolution (at least 1-3 km) to be more representative of local predictions, add assimilation procedure, add IFS forcing	Must have	NUM forecast at 3km without data assimilation and GFS forcing	- Availability of data assimilation procedure - Availability of IFS data - Stability of the new execution - Improvement of the prediction: comparison of RMSE on wind direction, wind speed and air temperature between baseline and EVEREST execution	Due to the execution time, we will not have an EVEREST simulation with only assimilation, one with only IFS, one with only improvement on spatial resolution, ...: Evaluation of performance will concern all combined effects
O2.2	Effective run of the service in terms of computational performance efficiency with a specific focus on the WRF workflow	Evaluating the most intensive models and kernels (e.g. physics parameterizations) selected for WRF modelling for cloud-resolving (1-3 km grid spacing) applications	Must Have	WRF CPU version	Percentage of the WRF execution to be accelerated with EVEREST technology	same as O1.3 since this objective is related to WRF that is a shared component across the two UCs
		The optimization of such kernels and models within the EVEREST platform	Must Have	Selected WRF-Kernel CPU version	Performance and energy Improvement	same as O1.3 since this objective is related to WRF that is a shared component across the two UCs
		The exploitation of the EVEREST platform to support meteorological workflow	Must Have	WRF pure CPU	End-to-end WRF execution with accelerated modules	same as O1.3 since this objective is related to WRF that is a shared component across the two UCs
O2.3	Moving towards ensemble prediction and AI machine learning approach	Design IA methods to exploit fine and coarse meteorological prediction forced by local observation	Must have	No ensemble model available in the baseline implementation	Availability of an ensemble models for prediction	Ensemble model is built considering NUM forecast, EVEREST WRF forecast, MeteoFrance Forecast and GFS
		Implementation of ensemble aggregation method for predictions at local scale with downscaling approach	Must Have	No ensemble model available in the baseline implementation	Improvement of the quality for prediction	Comparison of RMSE on wind direction, wind speed and air temperature between baseline and EVEREST execution
		Exploiting of the EVEREST computing platform in support of the ensemble prediction	Could have	No ensemble model available in the baseline implementation	Capability of the EVEREST platform to generate WRF ensemble models	Running also multiple WRF forecast for the ensemble requires more resources than what are available from the prototype platform of the project
O2.4	Predictive capability better than the current service	Exploiting of the EVEREST computing platform in support of the WRF workflow	Must Have	Original Forecasting on CPU on NUM HPC server	Speedup of the forecasting with possible use of FPGA acceleration	
		Exploiting of the EVEREST computing platform in support of the overall modelling workflow	Must have	Original Forecasting on NUM resources without anomaly detection	Availability of the end-to-and air quality prediction workflow that exploits the EVEREST distributed computing platform, including an anomaly detection module to secure transfer of data	
		Full statistical assessment of the added value of EVEREST forecasting and ensemble forecast on simulated air quality	Should have	Original Deterministic Forecasting based on NUM forecast	Reduction of number of forecasts of false pollution events and increase of number of forecasts of real pollution events	Large set of historical data and runs should be used within a selected period for a full statistical assessment

7.3 Traffic modelling

Traffic prediction often suffers from serious imprecisions due to, first, being built on top of **imprecise, simplified, outdated traffic models** (often reduced to so-called speed-profiles), second, **not exploiting big data and real-time data** at its full potential. The reason is the computational complexity of such calculation, which needs streaming of large chunks of data and an application of expensive machine learning (ML) operations in training, in real-time and in near real-time. Another problem is that some critical data are just missing or are very approximative such as origin-destination matrix. Another further improvement to the overall system precision could be obtained taking into account weather conditions. The methods, which can synthesize or refine such data boil down again to the category of high computational demand, such as Traffic simulator which learns missing traffic patterns over multiple runs to reflect reality based on reference sequences from observations (using AI components such as Bayesian inference, neural networks, gradient descent, maximum likelihood).

SYG envisions EVEREST as an enabler for the replacement of critical components of the traffic modelling framework with the most powerful counterparts (typically of AI origin), as well as a tooling for an optimization of data flow in and out of the processing elements. This has to consider not only improvements in terms of predictions but also costs and programmability efforts.

The objective is to create the traffic modelling eco-system comprised of several critical computation components, such as FCD processing, Traffic simulator, Prediction model training and inference, Intelligent routing. The conjunction of these components, accelerated as in specification, defines a product which can update traffic model and its services on a daily basis, overnight. This is what with today's market offer takes painfully long and with limited quality. Our eco-system benefits from tightly coupled real time data acquisition, followed by immediate data processing accelerated with EVEREST and finally directly turning into services.
Current market traffic view is represented with statistical speed profiles and statistical prediction. With the acceleration of critical kernels, we incorporate real-time input into the game making traffic view and prediction more precise. Use of ML algorithms, now enabled thanks to EVEREST, will further increase the precision of our solution.
SYG envisions EVEREST as an enabler for the replacement of critical components of the proprietary traffic modelling framework with the most powerful alternatives being of AI origin, as well as a tooling for kernels' accelerations. This also relates to the improvements in terms of saving energy costs and programmability effort.

The refined objectives follow. In comparison to D2.1:

- O3.3 has been alleviated in terms of "could have" annotations due to prioritization of computation optimization to data exchange optimization.
- O3.4 has been added, due to emergence of challenges in FCD big data processing

Table 38 Desiderata envisioned of the Traffic Modelling use case

Id	Objective	Method	Priority	Baseline	KPI	Notes
O3.1	Improve the overall performance of traffic simulation model	Optimized stream of queried data from big data sets into processing elements	Must Have	N/A	Availability of the module	
		Exploitation of an external traffic prediction service for each road	Should have	Traffic simulator without the neural network Traffic Prediction	Availability of the integration with a neural network Traffic Prediction	Before the project, the non-deterministic version of the traffic simulator used only PTDR algorithm. The new deterministic version of the simulator has to integrate both PDTR and Traffic Prediction.
		Evaluation of performance for traffic simulation test case	Must Have	50K vehicle on a single instance of the original traffic simulator	Number of vehicles possible to simulate thanks to EVEREST improvements	The baseline refers to the maximum amount of vehicles that was possible to simulate before the improvements carried out within the project
		Improve overall system precision by incorporating weather data from WRF model	Could Have	Traffic simulator without weather forecast model connected	Availability of a weather forecast module connected to the simulator	This objective is not part of the EVEREST project, however it will be evaluated the effort and impact in having it
O3.2	Improve the overall performance of neural network traffic prediction model training	Employment of accelerated AI computation kernels as enabler for a fast loop calculation. The utilization of heterogeneous architectures with efficient data management	Must have	CPU execution (Python)	Availability prediction kernel on FPGA with small programmer effort	
		Evaluation of computational performance, precision and energy cost with	Must Have	CPU based sequential execution of 10K	Speedup w.r.t. CPU based, % energy saving	
O3.3	Improve data management and computational services and reduce programming effort	Exploiting of an effective EVEREST computing platform to support distributed computation and optimized data flows	Must Have	N/A	Availability of end-to-end WFs for the Traffic use case	Before the project, the modules where available as separate instances
		Availability of EVEREST module that manages data exchange from and to accelerated kernels	Must Have	N/A	Availability of almost transparent accelerator integration in the application code	
		Heterogenous resources to be exploited in a simple way, e.g. with minimum number of lines of code	Must Have	No FPGA accelerators in the code	#kernels mapped to FPGA using the Everest Framework	Evaluated kernels are Traffic Prediction, MapMatching and PTDR
O3.4	Improve the overall performance of map matching kernel	Acceleration of Map-match kernel comprised of GPS-projection, Dijkstra and Viterbi components forming Hidden Markov model.	Must Have	Map Matching on CPU	Availability of FPGA accelerated kernel	
		Evaluation of computational performance and energy cost	Must Have	CPU based sequential execution of 20M FCD points	Speedup w.r.t. CPU based, % energy saving using FPGA acceleration	
O3.5	Improve the overall performance of the probabilistic time dependent routing Kernel	Acceleration of PTDR kernel Including Montecarlo Sampling for time to destination estimation	Must Have	PTDR Montecarlo on CPU (C++)	Availability of FPGA accelerated kernel	
		Evaluation of computational performance and energy cost	Must Have	CPU based sequential PTDR execution of several thousand requests	Speedup w.r.t. CPU based, % energy saving using FPGA acceleration	

8 Conclusions

The deliverable has revised the status of the EVEREST use cases, after the baseline implementation. As a first step, we updated each use case as a workflow with individual schematic components. Next – for each of the use cases – we specify the data used at the input, the data produced at the output, and the computational resources required to execute the application(s) on the EVEREST target system.

Based on the analysis of the individual use cases, we refine the overall list of requirements; such list contains the ones already elicited at M6, the refined ones (i.e., REQ4 and REQ5), and the new ones (i.e., REQ16 and REQ17).

The requirements have been divided into computational requirements and data requirements, which simplifies the association of those requirements with the EVEREST workpackages that address them.

In order to drive focus and execution within the project, we classified the requirements in terms of priority levels and their connection with EVEREST infrastructure and technologies. Three priority levels have been defined for the development: "Must have" are the mandatory requirements to obtain the first working prototypes to have a baseline for optimization; "Should have" includes the requirements that are expected to target for optimizing the baseline and to achieve the overall EVEREST KPIs; finally, "Could have" are additional requirements needed to trigger extra functionalities that can further improve the results. The renewable energy case has 11 "must have" requirements out of a total of 13, the air quality case has 8 "must have" requirements of out of a total of 10, and finally the traffic modelling has 11 "must have" requirements out of a total of 13.

As a final step, we stated the overall objectives in each of the use cases and which EVEREST technologies are used to accomplish these objectives; we also translated such objectives in a set of KPIs to support the evaluation task in WP6.

The renewable-energy prediction use case has been developed from scratch; the proper use of ML methods over large (observational and forecasted) datasets and the effective run of the WRF model represent very promising requirements to achieve a competitive **DUF** proprietary service. Furthermore, data integrity has been recognized as new requirement in order to improve the robustness of the system as well as to replace suspicious data (REQ17). This will ensure that there will not be missing data, thus improving the overall performance.

The air-quality monitoring instead has the ambition to face against uncertainties related with meteorological and pollutant predictions. ***NUM*** aims to exploit the state-of-the-art artificial intelligence methodology to merge data from different sources and at different scale, while considering the advanced use of data assimilation, ensemble and down-scaling techniques, and high-resolution weather forecast to improve their service for local air quality prediction. In this case, heterogeneous resources and complex orchestration tool represents key requirements. The use case exploits resources external to the EVEREST infrastructure; therefore, a high priority has been confirmed to REQ14 (data transfer) and data integrity has been recognized as new mandatory requirement to improve data validity.

The traffic modelling use case moves towards the replacement of critical components of the proprietary traffic modelling framework with the most powerful alternatives being of AI origin feed with real time and near real time stream of data. This has to be accomplished with the possibility to access further data (again observed or computed in a more accurate manner) while guarantee the access to resources in a simple way. Furthermore, due to the complexity of some components, the possibility to express convenience and flexibility for kernels' accelerations has been recognized as a mandatory requirement (REQ16). This also relates to the improvements in terms of saving energy costs and programmability effort. **SYG** highlights such enabling requirements to update their proprietary service.

References

[1] Jacobson, I. Christerson, M. Jonsson, P. Object-Oriented Software Engineering. A Use Case Driven Approach. ACM Press 1992, ISBN 978-0201544350.

[2] Agora Energiewende and Sandbag: The European Power Sector in 2017. State of Affairs and Review of Current Developments, 2018.

[3] Agora Energiewende and Sandbag: The European Power Sector in 2019: Up-to-Date Analysis on the Electricity Transition, 2020.

[4] Agora Energiewende and Ember (2021): The European Power Sector in 2020: Up-to-Date Analysis on the Electricity Transition, 2021.

[5] COM(2019) 640 final. European Commission. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions - The European Green Deal. Available online at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52019DC0640

[6] International Energy Agency (IEA), Power Systems in Transition, IEA, Paris https://www.iea.org/reports/power-systems-in-transition, 2020.

[7] Nazir, M. S., Alturise, F., Alshmrany, S., Nazir, H., Bilal, M., Abdalla, A. N., ... , M Ali, Z. Wind generation forecasting methods and proliferation of artificial neural network: a review of five years research trend. Sustainability 2020, 12(9), 3778.

[8] Lydia, M., Kumar, S. S., Selvakumar, A. I., Kumar, G. E. P. Wind resource estimation using wind speed and power curve models. Renewable Energy, 2015, 83, 425-434.

[9] Roungkvist, J. S., & Enevoldsen, P. Timescale classification in wind forecasting: A review of the state-of-the-art. Journal of Forecasting, 2020, 39(5), 757-768.

[10] Santhosh, M., Venkaiah, C., & Vinod Kumar, D. M. Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review. Engineering Reports, 2020, 2(6), e12178.

[11] González-Sopeña, J. M., Pakrashi, V., Ghosh, B. An overview of performance evaluation metrics for short-term statistical wind power forecasting. Renewable and Sustainable Energy Reviews, 2020, 110515.

[12] Yang, B., Zhong, L., Wang, J., Shu, H., Zhang, X., Yu, T., & Liming, S. State-of-the-art One-stop Handbook on Wind Forecasting Technologies: An Overview of Classifications, Methodologies, and Analysis. Journal of Cleaner Production, 2020, 124628.

[13] Powers, J. G., Klemp, J. B., Skamarock, W. C., Davis, C. A., Dudhia, J., Gill, D. O., ... , Duda, M. G. The weather research and forecasting model: Overview, system efforts, and future directions. Bulletin of the American Meteorological Society, 2017, 98(8), 1717-1737.

[14] World Health Assembly, 71. Health, environment and climate change: road map for an enhanced global response to the adverse health effects of air pollution: report by the Director-General. World Health Organization.https://apps.who.int/iris/handle/10665/276321, 2018

[15] Manisalidis, I. Stavropoulou, E. Stavropoulos, A. Bezirtzoglou, E. Environmental and Health Impacts of Air Pollution: A Review. Frontiers in public health, 2020, 8, 14

[16] Hood, C., MacKenzie, I., Stocker, J., Johnson, K., Carruthers, D., Vieno, M., and Doherty, R.: Air quality simulations for London using a coupled regional-to-local modelling system, Atmos. Chem. Phys., 18, 11221–11245, https://doi.org/10.5194/acp-18-11221-2018, 2018.

[17] D. Schrank, B. Eisele, and T. Lomax, 2019 urban mobilityreport,https://mobility.tamu.edu/umr/report/