D5.8 Implementation Report on Pilot Applications



Date: 29 October 2026



Document identification			
Status	Final	Due Date	31/10/2026
Version	1.0	Submission Date	29/10/2026

Related WP	WP5	Document Reference	D5.8
Related Deliverable(s)	D5.1, D5.3, D5.4, D5.5, D5.6, D5.7	Dissemination Level (*)	PU
Lead Participant	UNISTRA	Lead Author	Christophe Prud'homme (UNISTRA)
Contributors	SZE, UNISTRA, PSNC, MTG, FAU	Reviewers	Jesús Gorroñogoitia (BULL)
			Harald Köstler (FAU)

Keywords:
pilot applications, implementation, pilot coupling, high-performance computing, final results

Disclaimer for Deliverables with dissemination level PUBLIC
This document is issued within the frame and for the purpose of the HiDALGO2 project. Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Poland, Germany, Spain, Hungary, France, Greece under grant agreement number: 101093457. This publication expresses the opinions of the authors and not necessarily those of the EuroHPC JU and Associated Countries which are not responsible for any use of the information contained in this publication. This deliverable is subject to final acceptance by the European Commission. This document and its content are the property of the HiDALGO2 Consortium. The content of all or parts of this document can be used and distributed provided that the HiDALGO2 project and the document are properly referenced.
Each HiDALGO2 Partner may use this document in conformity with the HiDALGO2 Consortium Grant Agreement provisions. (*) Dissemination levels: **PU**: Public, fully open, e.g. web; **SEN:** Sensitive, restricted under conditions set out in Model Grant Agreement; **EU-C: European Union Classified**, the unauthorised disclosure of this information could harm the essential interests of the Consortium.

Document information

List of contributors	
Name	Partner
Christophe Prud'homme	UNISTRA
Javier Cladellas	UNISTRA
Zoltán Horváth	SZE
Michał Kulczewski	PSNC
Wojciech Szeliga	PSNC
Krzysztof Kotecki	PSNC
Luis Torres	MTG
David Caballero	MTG
Ravi Kiran Ayyala	FAU
Harald Köstler	FAU
Vincent Chabannes	UNISTRA
Philippe Pinçon	UNISTRA
Gwennolé Chappron	UNISTRA
József Bakosi	SZE
Mátyás Constans	SZE
László Környei	SZE
Filip Depta	PSNC

Document history			
Version	Date	Change editors	Changes
0.1	08/09/2026	Christophe Prud'homme (UNISTRA)	Initial structure, contributors, drafting guidance and introduction
0.2	09/09/2026	Marcin Lawenda (PSNC), Shubham Kavane (FAU)	ToC, responsibilities and section ownership confirmed by consortium.

0.3	17/09/2026	Christophe Prud'homme (UNISTRA)	UB contribution, results and coupling updates; revised with solar-mask implementation, geometry and reproducibility evidence.
0.4	01/10/2026	Christophe Prud'homme	Consolidated UB, MTG and FAU contributions; UAP, RES and final coupling inputs pending.
0.5	06/10/2026	Christophe Prud'homme (UNISTRA)	Consolidated UB, PSNC and MTG updates; preCICE proof of concept; revised WP5 synthesis, introduction and conclusion; UB reading cues; fields refreshed.
0.6	07/10/2026	Christophe Prud'homme (UNISTRA)	Editorial updates following Jesús's first review; expanded summary, removed drafting procedures, completed acronyms and clarified coupling status labels. Technical confirmations remain open.
0.7	12/10/2026	Christophe Prud'homme (UNISTRA)	Consolidated SZE_KL additions, MTG tracked corrections and PSNC review changes; common pilot technical tables; updated synthesis and review replies.
0.8	15/10/2026	Christophe Prud'homme (UNISTRA)	Consolidated contributions; editorial/status corrections; synthesis, references and fields updated for review.
0.8-review-1	19/10/2026	Christophe Prud'homme (UNISTRA)	Editorial corrections and replies to Jesús's review; acronyms alphabetised and completed; fields refreshed.
0.8-review-2	21/10/2026	Harald Köstler (FAU); Christophe Prud'homme (UNISTRA)	Harald's review edits consolidated; MTW-WF not pursued; UAP-UBM initial tests and container preparation updated.
0.8-review-3	27/10/2026	Christophe Prud'homme (UNISTRA)	StationReplay demonstrator and Strasbourg visualisation added to the UAP-UBM coupling section; deployment status and quality-management handover updated. Review comments and partner revisions retained.

0.9	27/10/2026	Christophe Prud'homme (UNISTRA)	Clean version for quality and coordination review; peer-review changes accepted, comments removed, editorial confirmation requests replaced by factual scope statements, and navigation fields refreshed.
0.9.1	29/10/2026	Christophe Prud'homme (UNISTRA)	QA corrections: reference metadata, deliverable DOIs, active links and citation navigation; paragraph introductions harmonised; duplicate citation fields removed; Section terminology and navigation refreshed. BULL retained.
0.95	29/10/2026	Ravi Ayyala (FAU)	Quality assurance check
1.0	29/10/2026	Marcin Lawenda (PSNC)	Final check and improvements

Quality control		
Role	Who (Partner short name)	Approval Date
Lead author	Christophe Prud'homme (UNISTRA)	29/10/2026
Quality Manager	Ravi Ayyala (FAU)	29/10/2026
Project Coordinator	Marcin Lawenda (PSNC)	29/10/2026

Table of contents

Executive summary 11
1 Introduction 12
1.1 Purpose of the document 12
1.2 Continuity with previous deliverables and other project work 12
1.3 Structure of the document 12
2 WP5 implementation status in M46 13
3 Urban Air Project implementation status 16
3.1 Final implementation status and developments since D5.7 16
3.2 Main results and achievements 20
3.3 Remaining limitations and perspectives 21
4 Urban Building Model implementation status 22
4.1 Final implementation status and developments since D5.7 22
4.2 Main results and achievements 26
4.3 Remaining limitations and perspectives 27
5 Renewable Energy Sources implementation status 28
5.1 Final implementation status and developments since D5.7 28
5.2 Main results and achievements 32
5.3 Remaining limitations and perspectives 33
6 Wildfires implementation status 34
6.1 Final implementation status and developments since D5.7 34
6.2 Main results and achievements 37
6.3 Remaining limitations and perspectives 42
7 Material Transport in Water implementation status 44
7.1 Final implementation status and developments since D5.7 44
7.2 Main results and achievements 48
7.3 Remaining limitations and perspectives 48
8 Pilot couplings implementation status 49
8.1 UAP-UBM coupling 50
8.2 UBM-RES coupling 55
8.3 UAP-WF coupling 56
8.4 UBM-WRF coupling 56
8.5 RES-WRF coupling 58
8.6 MTW-WF coupling 59
8.7 Technical summary of coupling implementations 60
9 Conclusions 62
References 63

List of tables

Table 1. Consolidated WP5 pilot implementation status 13
Table 2. Urban Air Project technical implementation summary 18
Table 3. Urban Building Model technical implementation summary 24
Table 4. Initial UB objectives and implemented capabilities 26
Table 5. Renewable Energy Sources technical implementation summary 31
Table 6. RES objectives and implemented capabilities 32
Table 7. Wildfires technical implementation summary 36
Table 8. Post-processing utilities implemented in the thermal turbulent channel application 46
Table 9. Material Transport in Water technical implementation summary 47
Table 10. Summary of the envisaged pilot couplings 49
Table 11. Technical summary of coupling implementations 60

List of figures

Figure 1. Recent examples of urban scenes and roof geometries generated using Ktirio GUI. 23
Figure 2. Spatial partitions and bounding boxes used for solar-mask ray tracing 26
Figure 3. Electricity-production forecast generated by RES-energy 30
Figure 4. Simulation of a fire in Győr with secondary spots caused by the spread of embers 41
Figure 5. Temperature field in the flow-over-heated-plate demonstrator using kub-feel++ and OpenFOAM coupled through preCICE 52
Figure 6. Velocity magnitude in the flow-over-heated-plate demonstrator using kub-feel++ and OpenFOAM coupled through preCICE 52
Figure 7. Building-surface temperatures in Strasbourg exchanged through StationReplay. 54
Figure 8. Detailed photovoltaic-panel mesh for the UBM-RES interface 55
Figure 9. The Strasbourg metropolitan area overlaid on the temperature map generated by the WRF model. The simulated area is outlined in red. 57
Figure 10. Comparison of outdoor temperatures in downtown Strasbourg (WRF on the left, weather station on the right) 58
Figure 11. Coupled WRF-RES workflow in the QCG Portal 59

List of acronyms

Abbreviation / acronym	Description
AI	Artificial intelligence
API	Application programming interface
BC	Boundary condition (BCs: boundary conditions)

Abbreviation / acronym	Description
BDF2	Second-order backward differentiation formula
BESTEST	Building Energy Simulation Test
CFD	Computational fluid dynamics
CFDR	HiDALGO2 tool for CFD post-processing and web-based visualisation
CFL	Courant-Friedrichs-Lewy
CHT	Conjugate heat transfer
CI	Continuous integration
CI/CD	Continuous integration and continuous delivery
CPU	Central processing unit
CSV	Comma-separated values
DMS	Data Management System (WP4)
ECMWF	European Centre for Medium-Range Weather Forecasts
EESSI	European Environment for Scientific Software Installations
EOS	Equation of state
EULAG	Eulerian-Lagrangian model
FD	Finite difference
FDS	Fire Dynamics Simulator
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
FSE	Fire Spread Engine
FV	Finite volume
GPU	Graphics processing unit
GUI	Graphical user interface
HDF5	Hierarchical Data Format version 5
HPC	High-performance computing
HPDA	High-performance data analytics
I/O	Input/output
IAQ	Indoor air quality

Abbreviation / acronym	Description
JFNK	Jacobian-free Newton-Krylov
JSON	JavaScript Object Notation
KPI	Key performance indicator
KUB	Ktirio Urban Building
LB	Lattice Boltzmann
LES	Large eddy simulation
LHS	Latin hypercube sampling
LiDAR	Light detection and ranging
LSTM	Long short-term memory
MATHSO	HiDALGO2 portal for application configuration and HPC workflow execution
MPI	Message Passing Interface
MTW	Material Transport in Water
mUQSA	Web-based platform for uncertainty quantification and sensitivity analysis
MUSCL	Monotonic upstream-centred scheme for conservation laws
MWp	Megawatt peak
PCG	Procedural Content Generation
PDF	Probability distribution function
PID	Proportional-integral-derivative
preCICE	Precise Code Interaction Coupling Environment
PSM	Partially saturated cells
PV	Photovoltaic
QCG	QosCosGrid workflow and HPC execution tools
RANS	Reynolds-averaged Navier-Stokes
RES	Renewable Energy Sources
RLE	Run-length encoding
SDK	Software development kit
SRT	Single-relaxation-time
SVT	Sparse Volume Textures

Abbreviation / acronym	Description
TLS	Terrestrial laser scanning
TRT	Two-relaxation-time
UAP	Urban Air Project
UBM	Urban Building Model
UCAR	University Corporation for Atmospheric Research
UE5	Unreal Engine 5
UQ	Uncertainty quantification
VoB	Volume of body
VTK	Visualization Toolkit
WALE	Wall-adapting local eddy-viscosity
WF	Wildfires
WFDS	Wildland-urban interface Fire Dynamics Simulator
WPS	WRF Preprocessing System
WRF	Weather Research and Forecasting
WUI	Wildland-urban interface
YAML	YAML Ain't Markup Language

Executive summary

D5.8 presents the implementation advances of the five HiDALGO2 pilot applications and their couplings at M46. Continuing D5.6 [29] and D5.7 [2], it explains how research developments have been incorporated into software and HPC workflows. The main advances concern physical models, application data, and the connection of simulation with analysis and visualisation.

The Urban Air Project has extended its capabilities for simulating urban airflow and pollutant transport. Developments in physical modelling and the preparation of urban geometry, vegetation and emissions support applications addressing air quality and wind comfort.

The Urban Building Model has extended the preparation and representation of buildings, solar exposure and building physics. Scenario and ensemble workflows connect these models with building data and result analysis, supporting studies of energy use, indoor conditions and urban climate.

The Renewable Energy Sources pilot has extended wind and photovoltaic forecasting and the workflows used to prepare and execute these applications. The integration of meteorological simulation and data-driven prediction supports the assessment of renewable electricity production.

The Wildfires pilot has developed capabilities for landscape fire simulation and for representing fire behaviour near buildings. Advances in vegetation preparation, simulation and visualisation connect environmental data with the analysis of fire and smoke scenarios.

The Material Transport in Water pilot has extended particle-laden flow simulation to include heat exchange between moving particles and the surrounding fluid. CPU and GPU implementations, together with ensemble execution and analysis tools, support the investigation of coupled transport processes.

Across the pilots, application workflows connect data preparation, numerical models and computing resources. Build and testing arrangements, deployment mechanisms and result processing support their use. The technical summaries give a common description of the software components and interfaces.

Coupling work connects complementary models through meteorological, thermal and environmental data exchanges. File-based workflows and direct solver interfaces provide different ways to study interactions across application domains. The coupling section documents these mechanisms and their application examples.

These developments broaden the questions that the pilots can address and support further integration and reuse. D5.2, currently in preparation, will draw on this implementation work to assess the final synergies and benefits of integrated models, software and hardware.

1 Introduction

1.1 Purpose of the document

Work Package 5 develops five HPC applications: Urban Air Project (UAP), Urban Building Model (UBM), Renewable Energy Sources (RES), Wildfires (WF) and Material Transport in Water (MTW). They address complementary environmental and societal questions and provide settings in which HiDALGO2 methods and services are integrated into simulation workflows.

D5.8 concludes the Implementation Report on Pilot Applications series. It explains the capabilities developed since D5.7 and how the software components and coupling methods work together. In accordance with the amended project description, it consolidates research advances, synergies and coupled simulations into an account of WP5 implementation at M46 (October 2026).

The report follows a common perspective across the pilots: model capabilities, application data, HPC execution, and result analysis and visualisation. The sections provide implementation details, examples and validation, with shared technical summaries for comparison. Detailed scalability and speed-up assessments remain in WP3. D3.3 and D4.10 are currently in preparation.

1.2 Continuity with previous deliverables and other project work

D5.6 established the initial implementations and envisaged links. D5.7 described subsequent developments and coupling results. D5.8 continues this account through the capabilities added to the models, their workflows and their interfaces.

The report builds on the synergies and coupled simulations in D5.1 [25] and the research advances in D5.3 [26], D5.4 [27] and D5.5 [28]. It focuses on their incorporation into application software and workflows.

WP5 integrates contributions from WP2 on integration and workflow execution, WP3 on application assembly and ensembles, and WP4 on data management, AI, analysis and visualisation. D5.2 will draw on this implementation work to assess the final synergies and benefits of integrated models, software and hardware.

1.3 Structure of the document

Section 2 gives a common WP5 overview. Sections 3-7 present pilot developments since D5.7, application results, limitations and perspectives. Section 8 describes the coupling implementations. Section 9 draws together the main advances and directions for further use.

2 WP5 implementation status in M46

WP5 implementation has advanced along two complementary directions: extending what the models can represent and developing the workflows needed to use them. Across all five pilots, the work connects application data with simulation, execution on HPC resources and the interpretation of results. Table 1 provides a consolidated overview of the pilot software, computing environments and application examples.

Table 1. Consolidated WP5 pilot implementation status

Pilot and codes	Computing environment	Application examples	Implementation overview
UAP: UAP-RedSim, UAP-FOAM (OpenFOAM v2606), UAP-Xyst	Reported targets: LUMI, Karolina, MeluXina and Vega; MATHSO portal. Per-solver and per-service deployment coverage is not established by the reported results.	Urban wind comfort and air-quality applications; FAIRMODE inter-comparisons	Urban airflow and pollutant transport; atmospheric and radiative boundaries; vegetation and emissions processing. Validation and service coverage are detailed in Section 3.
UBM: Ktirio, Feel++, Python and Modelica/FMI; EnergyPlus epJSON model-generation work	MPI/EuroHPC execution; Docker and Apptainer packaging; documented Slurm workflows; MATHSO portal deployment	City-scale building energy and indoor-environment studies; weather and renovation scenarios	Building geometry and physics, distributed solar masks, scenarios and ensembles; benchmark validation. Technical summary in Section 4.1.

Pilot and codes	Computing environment	Application examples	Implementation overview
RES: RES-wind, RES-PV, RES-energy, RES-runner; RES-damages, RES.EULAG	HPC workflows through QCG Portal	Wind-energy scenarios and daily photovoltaic-production forecasting	forecasting and orchestration implemented for wind and PV scenarios; turbine data needed for wind-energy estimates.
WF: WRF-Sfire, FSE, FDS/WFDS; UE5/OpenVDB tools	EuroHPC systems; GPU implementation of FSE	Iberian Peninsula workflows implemented using Discoverer for WRF and WRF-Sfire and MeluXina for FSE.	Fire-risk, uncertainty and visualisation tools implemented; new FSE version deployed in MeluXina.
MTW: waLBerla, lbmpy, MESA-PD	CPU and GPU implementations running on EuroHPC systems	Heat transfer in particle-laden flows; thermal channel-flow ensembles	Conjugate heat transfer for moving particles; CPU/GPU channel-flow simulation, statistics and checkpoint/restart; benchmark validation.

The developments broaden the physical processes and application settings addressed by the pilots. UAP extends urban airflow and pollution modelling; UBM develops the representation and simulation of buildings; RES combines meteorological and data-driven approaches to energy forecasting; WF connects fire behaviour with environmental and vegetation data; and MTW adds thermal interactions to particle-laden flows. The pilot sections (Sections 3-7) present these advances through their implemented features and application examples.

Common implementation work supports these capabilities through data preparation, build and testing workflows, deployment and result processing. The pilot technical tables use the same categories to describe these arrangements and the inputs and outputs involved. Coupling implementations extend this approach across applications by connecting complementary models and exchanging the quantities needed by each

workflow. Section 8 presents the corresponding interfaces, mechanisms and examples; detailed validation and integration qualifications remain with the relevant pilot and coupling descriptions.

3 Urban Air Project implementation status

Contribution lead: Zoltán Horváth (SZE)

Contributors: József Bakosi, Mátyás Constans and László Környei (SZE)

The Urban Air Project (UAP) is an HPC-application developed by SZE for addressing urban-air challenges. The following six major UAP use cases, which can be accessed through the MATHSO portal, can be considered as standalone applications as well, introduced in this way in the reporting period: UAP.AQ for modelling air quality, UAP.Comf for computing the urban wind comfort exceedance values, UAP.Clim for computing the urban climate-related variables, UAP.Plan for supporting urban planning, UAP.Rep for reporting urban-air statistics, and UAP.AF for the urban air forecasting system.

The general UAP workflow consists of the following stages:

- preprocessing, which handles the urban geometry and processes the use-case-dependent parameters,
- core airflow computation, equipped with use-case-related HPC and AI modules, which is based on the three CFD solvers: RedSim, OpenFOAM, and Xyst; the UAP implementations are called UAP-RedSim, UAP-FOAM, and UAP-Xyst,
- post-processing, which includes both HPDA modules and visualisation.

The following sections detail the advancements in the past period.

3.1 Final implementation status and developments since D5.7

UAP-RedSim

Driven by stakeholder requirements, UAP-RedSim's features have been significantly extended in the following major steps since D5.7, making RedSim an advanced HPC CFD tool and UAP-RedSim an enhanced urban air simulation application. The current software, all running under MIT licence,

- models urban weather with solar radiation, heat propagation, and turbulence with large eddy simulation,
- provides user-defined functions for users to set their parameters and to enable couplings,
- performs optimal scalability for urban application relevant parameters for full EuroHPC machine partitions.

The main features of the development since D5.7 are as follows.

- **Wind computation with Smagorinsky** models of large eddy simulation (LES) with the full compressible Navier-Stokes equations and, optionally WALE (wall adapted-local eddy modelling) equipped with BCs and source terms, see below.
- **Implicit time stepping** with BDF2 and preconditioned matrix-free methods, using Jacobian-free Newton-Krylov methodology. These major developments enable UAP-RedSim to use for real applications as large parameters as CFL = 1000 thus, for example, runs 2 times faster than physical time for a 1.4M cell mesh of the Retiro park, Madrid.
- **Atmospheric boundary conditions**
    - Temperature: prescribed to vary linearly with altitude **according to a lapse rate**; the ground inputs are taken from local measurements.
    - Pressure: determined from **hydrostatic equilibrium.**
    - Density: from the gas EOS.
    - Wind velocity varies with altitude according to the prescribed atmospheric wind profile (log-profile with correction terms according to the stability status of the atmosphere, which is defined from the computed Obukhov length).
    - The resulting atmospheric state is applied through the characteristic far-field treatment.
- **Radiative wall boundary conditions**
    - Absorbed solar radiation (includes direct and diffuse irradiation, albedo, sky-view factor, and solar incidence angle) heats the wall, while bulk forced convection and linearised long-wave radiation cool it.
    - The resulting bounded equilibrium wall temperature is imposed through the ghost-cell thermodynamic state.
    - No transient wall-energy equation or conjugate wall-conduction equation is solved.
- **Source terms**
    - Gravity and resistance forces in the momentum equation for vegetation modelling (trees, etc.).
- **Pollution transport**
    - Advection-diffusion-reaction equation for NO, NO2, O3 (3 species).
    - For the advection and diffusion coefficients, it uses velocity and nu_t from wind computations, respectively, and numerically solved in the same implicit way as LES.
    - NO-NO2-O3 reaction:
        - NO-NO2-O3 reaction with radiation, and temperature-dependent coefficients. Concentrations immediately set to Leighton equilibrium.

- Computed by Strang splitting from the transport's advection-diffusion part.

UAP-FOAM

D5.7 reported support for OpenFOAM v2412 and temperature transport with Boussinesq buoyancy. Since then, our UAP-FOAM work has concentrated on preparing the Madrid application using OpenFOAM v2606. We completed the geometry preparation and a mesh of 7,822,643 cells, with local refinement around the measurement area.

The Madrid configuration uses RANS airflow modelling with the standard k-ε turbulence closure. Its atmospheric boundary conditions combine consistent wind, turbulence and ground-roughness treatments, with inflow calibration against measurements at 25 m. Deciduous and evergreen trees are represented through canopy drag and the Sogachev-Panferov turbulence-dissipation model, with seasonal leaf-area-density parameters. Low vegetation enters through ground roughness.

A dedicated utility has been prepared to map road-segment emissions onto CFD cells. Annual inventories define spatial distribution, while hourly profiles control emission strength. NOx is transported through Eulerian advection-diffusion, and a random-forest method estimates NO_2 using NOx, meteorological conditions and temporal predictors. Background concentrations are supplied separately. The modelling approach considers steady scenarios for longer-period reconstruction and transient simulations that retain transport history.

Table 2 summarises the available technical implementation information. The table distinguishes documented capabilities from information not established within the reported implementation.

Table 2. Urban Air Project technical implementation summary

Item	Implementation
Software and availability	RedSim, and the UAP-RedSim software are available under MIT licence from https://github.com/sze-hidalgo-2/RedSIM . UAP-FOAM application uses OpenFOAM v2606 with a closed SZE licence. Xyst is available from https://xyst.cc/ under GPL v3 licence.

Item	Implementation
Build process and dependencies	RedSim: No external dependencies, apart from single header files integrated and modified such as stb_sprintf (https://github.com/nothings/stb/blob/master/stb_sprintf.h) The build process uses a single translation unit to optimise build time, the whole project builds in seconds (e.g. under one second on an Apple A18 Pro processor (MacBook Neo)). In order to build, we use a single bash script invoking clang, gcc or MPI with build flags. UAP-RedSim: Gmsh UAP-FOAM: OpenFOAM v2606
CI workflow and automated tests	RedSim: Integrated with GitHub through GitHub workflows: .github/workflows/ci.yml UAP-FOAM: Integrated to the portal CI. Xyst: not supplied in this contribution.
Main implemented features	RedSim: Smagorinsky large eddy simulation (LES) for the compressible Navier-Stokes equations by using 2nd order finite volume method and MUSCL-limited least square gradients. BDF2 implicit time-stepping with Jacobian-free Newton-Krylov method. UAP-FOAM: RANS airflow; vegetation canopy drag; road-emission mapping; NOx transport; and empirical NO2 estimation. Xyst: not supplied in this contribution.
Inputs and outputs	Inputs include building geometry, road-segment emission inventories, meteorological observations and background concentrations. Airflow and pollutant results are compared with observations. EnSight Gold is reported as a supported format; the remaining input/output formats and their coverage by solver remain to be specified.
Deployment and execution	SZE reports routine use of LUMI, Karolina, MeluXina and Vega with optimised Slurm scripts. The contribution does not establish deployment coverage for every solver and UAP service.

Item	Implementation
Validation and documentation	FAIRMODE comparisons for Madrid include winter and summer campaigns and full-year evaluations (Section 3.2); quantitative validation remains ongoing (Section 3.3).

3.2 Main results and achievements

UAP-RedSim

UAP-RedSim has been successfully applied to wind comfort simulation for Stockholm city, in a joint work with ENCCS (Swedish HPC competence network) and SLB Analys (unit of Stockholm city hall); see the EuroCC success story book [16]. A gallery of visualisations for applications to Stockholm and other cities can be found at https://memo.sze.hu/ . The flagship application of UAP-RedSim and UAP-FOAM is the FAIRMODE Intercomparison Exercises for Antwerp in 2024 (see [17]) and Madrid in 2026 (ongoing).

UAP-FOAM

The Madrid application addresses the original UAP objective of resolving local variations in urban airflow and pollution. It brings detailed building geometry, vegetation, meteorological observations and time-dependent traffic emissions into one modelling workflow. Concrete implementation outputs include the completed 7.82-million-cell mesh and road-segment inventory processing that preserves the total annual emissions.

Evaluation covers the winter campaign of 6 November-1 December 2014 (600 hours), the summer campaign of 15 June-14 July 2016 (696 hours), and the full years 2014 and 2016. Passive-sampler comparisons use NO_2 averages over the actual exposure periods and at the measurement locations and heights. Hourly comparisons use Escuelas Aguirre and Parque del Retiro, identifying observations used as model inputs separately from those used for independent validation.

Xyst

Xyst has been successfully applied in a full-machine run on LUMI in which Xyst was proven almost linearly scalable up to 196k CPU cores.

3.3 Remaining limitations and perspectives

RedSim and UAP-RedSim

Documentation and validation through publications remain ongoing.

The main perspectives of the codes are that RedSim is a native OpenMPI-CUDA CFD solver with LES options for the compressible Navier-Stokes equations on block-structured and unstructured polyhedral meshes that linearly scales up to 1 full partition of the LUMI-C under industrially relevant conditions. In this way, it provides the European industry and other communities with a true HPC CFD code with exactly the same features as for market-leader commercial codes.

UAP-RedSim can provide a great tool for city planners and other users for tackling urban air-related challenges, including studying climate scenarios.

UAP-FOAM

The Madrid inter-comparison uses flat terrain, although the UAP-FOAM workflow supports non-flat terrain. Remaining work includes completing the quantitative validation, verifying emission conservation throughout the workflow, and assessing sensitivity to meteorological forcing, background concentrations and vegetation parameters. These tasks, together with deployment documentation, are expected to be completed in the coming months. Initial implementations of chemical reactions and solar radiation could also be developed and verified against benchmark cases, subject to available resources.

After HiDALGO2, the intended continuation will focus on validating these additional physical models in urban settings, applying the workflow to further cities, and extending uncertainty assessment. Further development will support air quality, comfort, climate and planning applications, alongside maintenance of the software, documentation and reference cases. The scope and timing of this work will depend on follow-up funding and partner contributions.

Xyst

Xyst is a highly scalable CFD solver with adaptive load balancing and proven flexibility for implementing new CFD algorithms. This may be used by a larger set of developers as a testbed for numerical CFD methods, and is provided for application developers as a tool for CFD.

4 Urban Building Model implementation status

Contribution lead: Christophe Prud'homme (UNISTRA)

Contributors: Vincent Chabannes, Javier Cladellas, Philippe Pinçon and Gwennolé Chappron (UNISTRA)

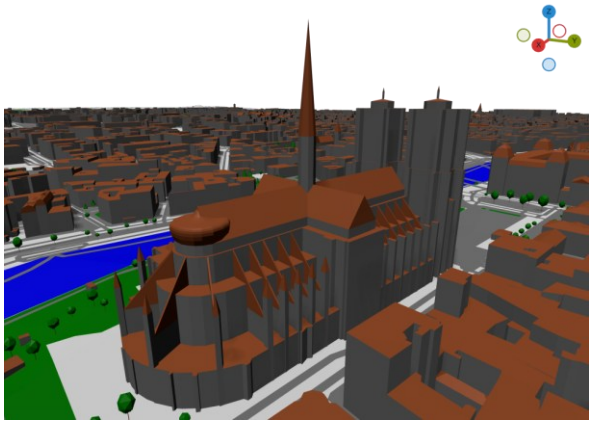
Ktirio Urban Building (KUB) simulates building energy, thermal comfort and indoor air quality at urban scale. Task 5.2 called for these models, improved solar shading, UAP exchanges and validation against monitored buildings [1]. The implementation snapshot underpinning this contribution is dated 17 September 2026; it is consolidated here with the subsequent M46 updates and coupling assessments.

4.1 Final implementation status and developments since D5.7

D5.7 reported multi-storey models, heating controls, parallel solar masks and city deployments, including more than 160,000 buildings in Paris and its suburbs [2]. Subsequent work has extended the geometry pipeline and distributed solar-mask calculation, and implemented the scenario and ensemble workflows then under development.

Urban geometry and building data. Ktirio-geom also adds roof forms, solar-panel meshes, tetrahedral meshing and refinement. Multithreading, incremental constrained Delaunay triangulation and a correction to tile merging improve preprocessing. Refinement is restricted to triangles touching tile boundaries [3].

CemDB, the KUB database layer, separates location data (geometry, weather and usage profiles) from simulator data (FMUs and model metadata). A scenario resolves these inputs before execution. Ktirio-geom urban meshes are connected to the Feel++ C++ simulation core, which distributes buildings across MPI processes and executes Modelica models through FMI [4] [3]. Figure 1 presents recent urban scenes and roof geometries generated using Ktirio GUI.



(a) A close-up of Notre Dame de Paris, showing different roof types



(b) A wide view of Grenoble, France, showcasing terrain elevation.

Figure 1. Recent examples of urban scenes and roof geometries generated using Ktirio GUI.

Building physics and validation. Building models use explicit constructions, glazing, ground conditions and schedules. Solar radiation, window optics, heat exchange and time integration have been corrected and checked through analytical tests and BESTEST. Model generation and execution use the same resolved inputs. The indoor air-quality model tracks CO2, NOx and particulate matter using ventilation and outdoor concentrations [4] [5].

Interchangeable simulation backends. Ktirio maintains the building representation independently of the underlying thermal simulation engine. An API defines the interface between this representation and the selected backend, taking account of the capabilities supported by that engine. This architecture supports the existing Modelica/FMU workflow and prepares the integration of EnergyPlus and other simulation backends.

For EnergyPlus, a lightweight API was developed around its JSON schema to generate epJSON models and run simulations in memory. This extends the model-generation workflow while keeping the building representation under Ktirio's control.

Distributed solar-mask calculation. Solar-mask calculations use spatial partitioning and distributed ray tracing. Terrain, buildings and vegetation are partitioned while keeping each building on one rank. A global bounding-volume hierarchy identifies the relevant partitions, and local hierarchies contain the detailed geometry. Rays are exchanged through asynchronous MPI communications. This implementation is complete [3].

Scenario and ensemble execution. The scenario framework uses YAML (YAML Ain't Markup Language) to describe its configuration. It organises components into bundles and scenario sets, then materialises a locked configuration for execution. It covers weather events, envelope renovation and ventilation strategies. Versioned datasets,

manifests and checksums identify the inputs used in heatwave, cold-wave and renovation studies [4] [3].

Ensembles support Monte Carlo and Latin hypercube sampling of global or building-specific parameters. MPI distributes buildings within each simulation and runs samples concurrently. Statistical summaries, uncertainty intervals and risk indicators implement the ensemble capability envisaged in D5.7 and contribute to the WP3 work [4].

Results and visualisation. Building-level energy, comfort, temperature and indoor air-quality outputs feed dashboards, ParaView and Ktirio.GUI. Manifests link results to their source data. These capabilities provide the UB contribution to WP4 data management and visualisation [4].

Packaging and deployment. For WP2 integration and deployment, the repository provides GitHub Actions workflows for C++ tests, Python packaging, Docker images and Apptainer conversion. Containers are published through the Feel++ registry, and documented Slurm workflows support HPC execution. The KUB documentation describes the software capabilities and usage [4] [6]. UB is also deployed on the MATHSO portal developed by SZE; a HiDALGO2 user can run KUB simulations on HPC machines by configuring their connection to the machines of their choice and monitor the results of the various instances they run from the dashboard.

Table 3 summarises the software, build and testing workflows, data formats and deployment arrangements described in this implementation section.

Table 3. Urban Building Model technical implementation summary

Item	Implementation
Software and availability	Ktirio Urban Building (KUB) combines a Feel++ C++/MPI simulation core with Python tools and Modelica models executed through FMI. Documentation: https://kub.ktirio.fr/. The repository declares dual licensing: GPL-3.0-or-later or a commercial licence.
Build process and dependencies	The C++ application is configured and built using CMake presets against the Feel++ SDK. Python packaging uses scikit-build-core; Python bindings use pybind11. The workflow includes Modelica/FMU generation tools. HDF5 supports time-series export; Apache Arrow/Parquet support is optional.

Item	Implementation
CI workflow and automated tests	GitHub Actions workflows define Python package builds and pytest execution, containerised C++ builds and CTest execution, version-consistency checks, and Debian, Docker and Apptainer packaging. Publication steps depend on the workflow context.
Main implemented features	Urban geometry preprocessing; building energy, thermal comfort and indoor air-quality simulation; distributed solar-mask calculation; configurable weather, renovation and ventilation scenarios; and parallel ensemble execution with statistical analysis. Ktirio maintains the building representation independently of the thermal backend. The existing Modelica/FMU workflow is complemented by EnergyPlus model-generation work using epJSON.
Inputs and outputs	Inputs include JSON building data and metadata, Gmsh meshes, FMUs, CSV weather and schedules, and YAML scenario definitions. Outputs include HDF5 time series, CSV results, JSON metadata and statistics, and EnSight Gold files for visualisation in ParaView. Parquet analytics output is available when the corresponding build option is enabled.
Deployment and execution	MPI execution is supported through containerised workflows, with Docker and Apptainer packaging and documented Slurm execution. Deployment through the SZE MATHSO portal is reported in the UB contribution: users configure their HPC connection, launch simulations and monitor runs through the dashboard.
Validation and documentation	The UB contribution reports analytical and ventilation/indoor air-quality benchmark tests, together with annual heating and cooling results meeting ANSI/ASHRAE Standard 140 acceptance criteria for the BESTEST cases evaluated. KUB documentation covers installation, execution and data organisation: https://kub.ktirio.fr/.

4.2 Main results and achievements

Table 4 links the main capabilities to the original Task 5.2 objectives. The added geometry, scenario and ensemble infrastructure extends the initial pilot. These implementation results are distinct from the assessment of formal project KPI targets.

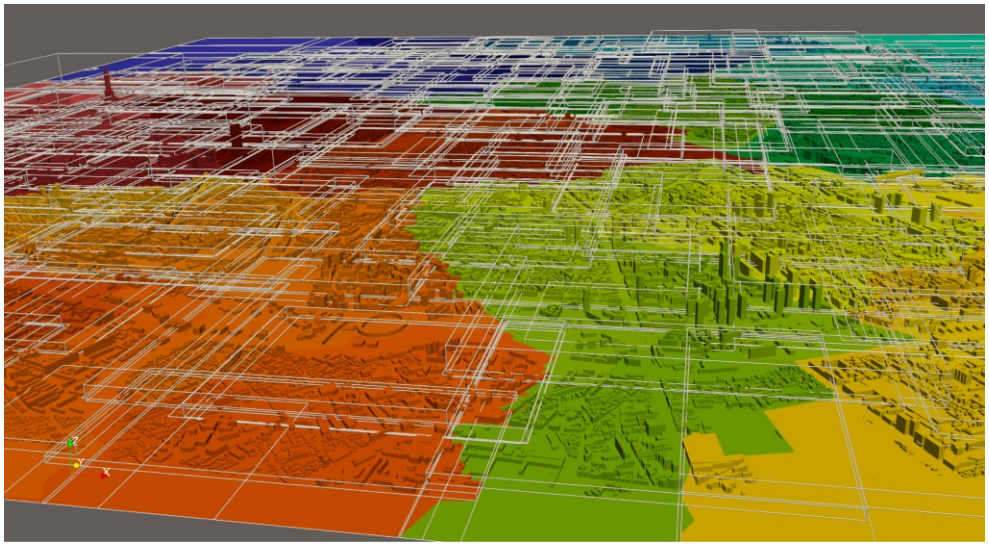
Table 4. Initial UB objectives and implemented capabilities

Initial objective	Delivered capability	Evidence and boundary
Energy and comfort at urban scale	Multi-storey thermal models, heating systems and parallel city simulation	City deployments in D5.7; subsequent physics and scenario improvements.
Improved solar shading	Partitioned urban geometry, distributed ray tracing and tested solar-radiation treatment	Distributed ray tracing with spatially partitioned geometry and asynchronous MPI exchanges.
Ventilation and indoor air quality	Ventilation and multi-species IAQ models	Tested against benchmark cases to verify ventilation flows and indoor concentration behaviour.
Model validation	Six-case BESTEST workflow and independent execution checks	Selected annual-load criteria pass; monitored-building validation is a separate requirement.
Additional integrated capabilities	Geometry tools, reusable scenarios, parallel ensembles and traceable analysis	Executable workflows for renovation, heat-event and uncertainty studies.

For the BESTEST cases evaluated, the building models' annual heating and cooling results meet the acceptance criteria of ANSI/ASHRAE Standard 140 [5] [7].

Figure 2 illustrates the urban geometry and bounding boxes used in the solar-mask calculation. Colours distinguish spatial partitions.

Figure 2. Spatial partitions and bounding boxes used for solar-mask ray tracing



4.3 Remaining limitations and perspectives

The UB pilot has achieved its main HiDALGO2 implementation objectives and extended the initial scope through additional modelling and workflow capabilities. The main remaining challenge is the availability and quality of building information: geometry, construction properties, occupancy and local outdoor conditions. These uncertainties affect how closely simulations represent individual buildings. BESTEST provides building-physics validation for the cases evaluated; comparison with measured building data remains important for real-world applications.

The perspective is to develop increasingly representative building models and deploy them in practical studies with social housing providers, local authorities and energy providers. Emerging datasets, including LiDAR surveys, offer opportunities to improve building geometry, while information available from partners can refine construction properties, usage profiles and operating conditions.

The implemented UB platform supports the monitoring, simulation and assessment of buildings and energy or renovation scenarios. Further enrichment and validation can draw on new observations, corrected properties and monitoring data, with updates recorded so that simulation results remain traceable.

5 Renewable Energy Sources implementation status

Contribution lead: Michał Kulczewski (PSNC)

Contributors: Wojciech Szeliga, Krzysztof Kotecki and Filip Depta (PSNC)

This section presents the final status of the Renewable Energy Sources pilot and the progress achieved since D5.7.

5.1 Final implementation status and developments since D5.7

Renewable Energy Sources (RES) pilot supports renewable-energy modelling in several areas: assessing probability of the damages, weather forecasts tailored for the renewables, and electrical energy production forecasts. The pilot comes with different components tailored to different needs: i) RES-damages for assessing the probability of the damages to the infrastructure due to extreme weather events, ii) RES-wind for weather forecasts and flows specific to wind farms, iii) RES-PV for weather forecasts specific to photovoltaics systems, iv) RES-energy, coupled to RES-PV, to forecast energy production from PVs. On top of these components there is RES-runner for orchestrating each scenario setup and execution.

In the reporting period the focus was on finalising the RES-wind and RES-PV components, providing a suitable user interface and finalising the visualisation of results for users and stakeholders.

RES-wind

As described in D5.7 in terms of modelling renewable energy systems, the first stage of the framework development was focused on modelling photovoltaic systems. Since then, a significant effort has been made in order to tailor the RES pilot, using RES-wind module, for weather predictions dedicated to wind farms. First, there are two different physical setups of the solver - one is dedicated for the wind farms located outside the cities where orography is taken into account. The second is for the wind turbines located in the cities, e.g. on the roofs, where geometry of the buildings is taken into account. The user can define the location and height of each turbine and automatically receive wind vector components for each turbine's rotor centre point, which then can be used for energy production estimation based on turbine's characteristics. The output data visualisation has been improved with the option of marking turbines' location in autogenerated pictures depicting 2D view of solver's results. The framework development has been integrated with EULAG, the second-stage solver of the RES framework. In EULAG a dedicated workflow has been implemented for wind-turbine scenarios. It combines using WRF solver output data for setting up boundary conditions as well as introducing terrain-following mesh deformation for capturing local topography. In order to enrich the wind farm-oriented weather simulations flexibility, apart from using WRF results as input data, there have

been added several ways of defining vertical wind profile in EULAG, especially valuable for analysis of hypothetical scenarios, for example, at the stage of planning a new wind farm. Now the user can use four approaches of modelling wind profile: 1) constant, 2) linear, 3) log wind profile (based on surface roughness length), 4) wind profile power law (based on exponential coefficient). [22]

RES-PV and RES-energy

The RES-energy module, known also as RES.AI, has been prepared for the photovoltaics in a way that it can be trained and later used for inference for any location. Any historical weather data source can be used to train the model, as long as at least one year of the required data is provided. For this pilot, RES-energy was additionally trained with the data coming from RES execution for the past 1.5 years, and it is now used on a daily basis for the energy production of 1 MWp PSNC's photovoltaics farm. This component has been substantially extended in three directions: i) enrichment of the weather inputs, ii) quantification of their impact on prediction accuracy, and iii) extension of the forecast horizon to a full day ahead, and the details can be found in deliverable D4.5 [31]. Moreover, it is now part of the whole workflow available to be run within QCG Portal, providing immediate forecasts for the photovoltaics and energy production, see Figure 3.

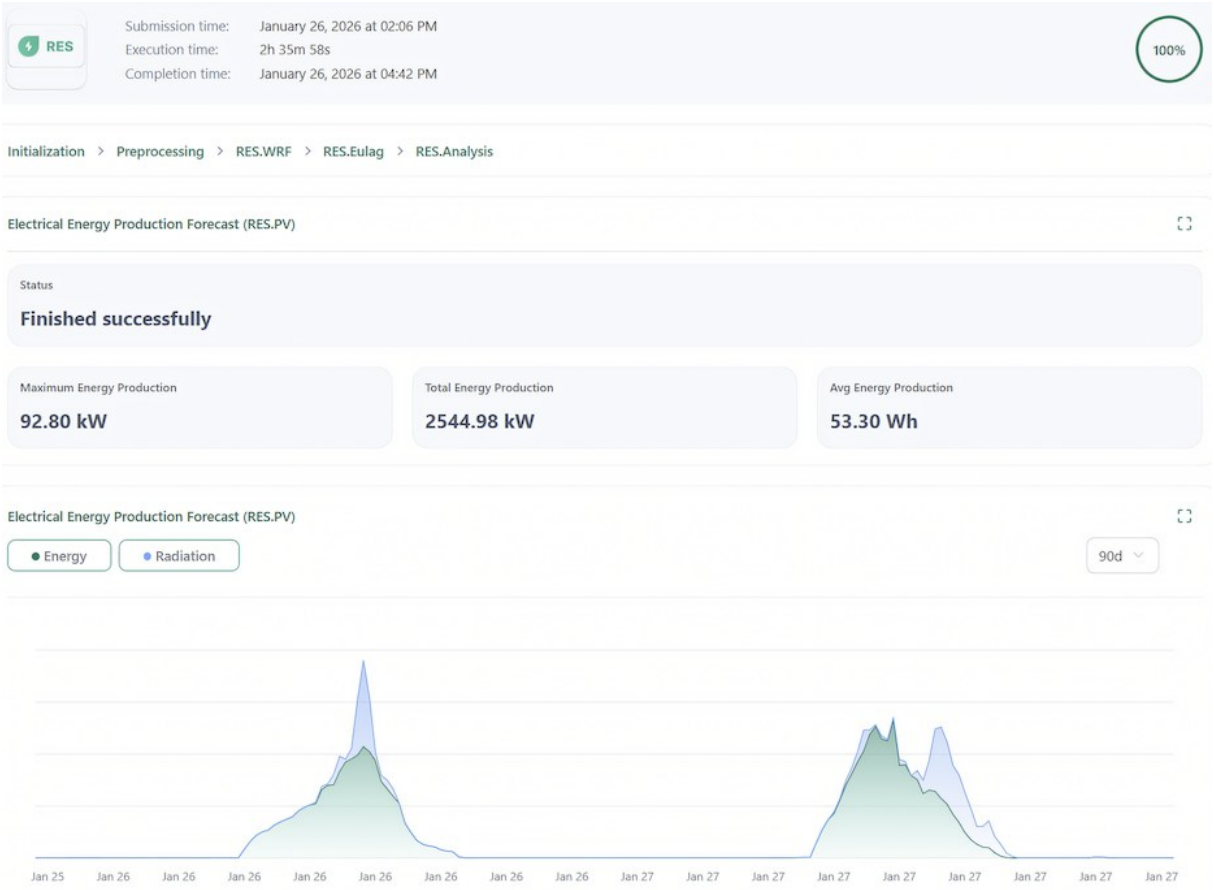


Figure 3. Electricity-production forecast generated by RES-energy

RES-runner

The final version of RES-runner supports the execution of the whole pilot workflow within the QCG Portal (GUI), as described in deliverable D2.9 [30]. The user provides details of the scenario via the GUI, including region of interest, required data, scope of the analysis, which is then processed by RES-runner. In return, RES-runner automatically enlarges or shrinks computational domain, provides the best domain decomposition and time-step in terms of time-to-solution, prepares the initial data to be downloaded, forms boundary conditions, and dictates optimal hardware resources to be used. The execution of the workflow is then orchestrated via the QCG Portal.

Table 5 summarises the available technical implementation information. The table distinguishes documented capabilities from information not established within the reported implementation.

Table 5. Renewable Energy Sources technical implementation summary

Item	Implementation
Software and availability	The pilot provides EULAG solver available for RES-wind, RES-PV, and RES-damages scenarios, supplemented by RES-energy module, and packed within RES-runner framework. RES components are available under AGPLv3 licence.
Build process and dependencies	The build process is orchestrated through RES-runner. Detailed package build instructions are planned for a README by the end of M48; they are not reported as available at the M46 reporting point.
CI workflow and automated tests	RES uses GitLab runners for the CI/CD pipelines - each commit triggers a solver test case to be run on an HPC machine to validate the update.
Main implemented features	Detailed weather forecasting for wind farms and photovoltaics; PV energy-production forecasting; infrastructure-damage assessment; ensemble and uncertainty quantification workflows, and a graphical interface for configuration, execution, monitoring and post-processing via the QCG Portal
Inputs and outputs	Inputs include weather data for the initialisation of the model, topography, building structure, land cover, overhead network topology for the RES-damages scenario, location of wind turbines for the RES-wind scenario. Outputs include detailed weather and energy forecasts, uncertainty results, and visualisations. Section 8.5 describes file-based WRF-to-EULAG exchange
Deployment and execution	HPC workflows are available through the QCG Portal. RES-energy is used daily for the PSNC 1 MWp PV installation. The MTG-specific WRF integration and complete UBM-RES workflow remain under evaluation.

Item	Implementation
Validation and documentation	PV forecasting has been trained using RES-generated data and is used for the PSNC installation. RES-damages has been evaluated on existing infrastructure. RES-wind has been used to select location for the forthcoming PSNC's wind farm

5.2 Main results and achievements

Table 6 summarises the initial RES objectives and the corresponding implemented capabilities.

Table 6. RES objectives and implemented capabilities

Objective	Capability	Evidence
Damages assessment due to extreme weather events	Probability of damages applied to the overhead electrical network, applicable to the wind farms, using uncertainty quantification	RES-damages with mUQSA and uncertainty quantification (WP4)
Support for urban and non-urban wind farms	Two different solver setups for orography and immersed boundary method	RES-wind
Radiation forecasts for the photovoltaics	1-minute radiation forecasts, computed per specific PV location	RES-PV
Energy production from RES	Forecasting electrical energy production from PV	RES-energy, validated against PSNC's 1 MWp PV installation
User interface	Running RES pilot workflows via GUI with setup, simulation monitoring, results post-processing	QCG Portal (WP2), visualisation (graphs, CFDR - WP4)
Ensembles	Running ensembles for improved weather forecasts, study how much model is prone to differences in the initial conditions	Uncertainty quantification with mUQSA (WP4), ensemble (WP3) with HPDA post-analysis (WP4)

5.3 Remaining limitations and perspectives

While the functionality of RES-wind module is ready to perform wind farm-tailored simulations and data generation, technical data for the turbines are still required to be able to predict the energy generation. At the moment RES-wind delivers weather forecasts for wind farms (outside urban areas) and wind turbines in the cities, it is up to the user then to use this data for energy production forecasts. Such tailored forecast can be used to find the best location for the new wind infrastructure.

RES-energy is a generic model for forecasting energy production from photovoltaic installations. Users can adopt the model and train with their datasets, keeping in mind that some tweaks may be necessary, in particular for PV infrastructure with peak power lower or higher than 1 MWp. The RES solution may be used to provide 1+ year of required data for any location of the PV infrastructure. Extensions beyond the reported implementation include providing a generic model for forecasting energy production from wind farms and wind turbines. Another perspective is to account for factors which lower the efficiency of energy production from the photovoltaics, such as dust, poor air quality, or snow cover.

RES-damages provides a solution to assess the probability of the damages to the overhead electrical network due to the extreme weather events related to the wind. This solution is applicable to the wind farms and wind turbines. Extensions beyond the reported implementation include providing model dedicated to the photovoltaics infrastructure, but also inclusion of other events such as hail storms, excessive heat, or icing.

6 Wildfires implementation status

Contribution lead: Luis Torres (MTG)

Contributors: David Caballero, Leydi Laura Salazar and Jaime Ribalaygua (MTG)

As described in the HiDALGO2 deliverable D5.7, two different use cases have been defined within the WF Pilot. A first use case using WRF-SFIRE focused in atmosphere and wildfire interactions at landscape scale, this is several km of fire spread in the wildland area; and a second use focusing on the fire behaviour around houses in the WUI scale (wildland-urban interface), comparing OpenFOAM against WFDS. [18] [19]

6.1 Final implementation status and developments since D5.7

Landscape scenario

In addition to solving WRF bottlenecks, MTG has adapted its wildfire simulation engine (FSE) for CPU and GPU execution on HPC systems. The GPU version has been prepared to support large-scale sensitivity analysis and uncertainty quantification campaigns. These activities, carried out jointly with other HiDALGO2 partners (namely ICCS, Eviden and PSNC), leverage HPDA techniques to process the outputs of thousands of simulation runs for territory vulnerability assessment and are helping to define AI solutions applied to fire behaviour.

Important advances have been made in both target scenarios. At the landscape scale, improvements are being made in uncertainty quantification (UQ) through ensemble simulations. For this study, several tests were carried out with groups of 50 or 350 members. In order to test the ability to run members simultaneously using up to 700 nodes, two different configurations were considered and three tests were carried out. We are grateful to the technical team at Discoverer for enabling us to run these tests by increasing the project QoS and the resources available.

Progress has also been made in facilitating the compilation of WRF on various HPC architectures using EasyBuild and by developing a solution based on EESSI. EESSI is currently available on several of the HPCs in the Euro-HPC network (Karolina, Vega, Deukalion and MareNostrum5). A contribution to the EESSI software repository, with a WRF-SFIRE EasyBuild recipe has been submitted and we are waiting for feedback. We would like to thank the EPICURE project for the support they have provided us in this effort and for their offer to serve as liaisons with the EESSI team to ensure that this solution is adopted in their repository.

Wildland-Urban scenario

A study has been done to evaluate the suitability of several OpenFOAM selected solvers (fireFoam, porousGasificationFoam) as well as FDS-WFDS for combustion simulation through a comparative study, including their scalability, in order to identify the workflow that provides the closest results to reality and is sufficiently practical for

use in real case scenarios. Although the solvers used were designed for modelling pyrolysis processes in biomass furnaces, their suitability for a new environment, pyrolysis and biomass combustion processes in the wildland urban interface, is being evaluated. As a result, the WFDS-based solution was selected to develop a workflow that facilitates the analysis of fire situations at the wild-urban interface.

To this end, the first step has been the development of an innovative solution for the description of vegetation as a porous medium, the first tests have been carried out for its incorporation into the workflow and its portability to the HPC environment is planned. This allows the incorporation of vegetation masses at individual level in the WUI environment.

The central scientific contribution is a pipeline with no prior published equivalent: UE5 Procedural Content Generation (PCG) vegetation → fractal barycentric discretisation of foliage geometry → FDS/WFDS Lagrangian particle initialisation → EuroHPC-scale simulation → OpenVDB post-processing of SMOKE3D output → real-time volumetric rendering in UE5 via Sparse Volume Textures (SVT). Two papers built on this pipeline have passed peer review at the X International Conference on Forest Fire Research (Advances on Forest Fire Research, 2026, in press), and a third, addressing FDS/WFDS scalability on EuroHPC infrastructure, was submitted to PPAM 2026.

Two lines of work were carried out during this period. First, software development: the FDS Vegetation Exporter UE5 plugin (stable at v3.2, with proposed v3.3 extensions), a LiDAR-to-FDS converter for terrestrial laser scanning data (las2fds_meluxina.py), and a parallel FDS-to-OpenVDB converter (fds2vdb_parallel.py) validated at production scale. Second, a source-code-level finding of scientific significance: FDS 6.10.1 does not support heterogeneous bulk-density assignment at the individual-particle level, a structural limitation identified by direct inspection of read.f90 and part.f90 and confirmed empirically by binary output parsing, which motivates a proposed solver fork (feature/binary-particle-init) developed in coordination with the WFDS author community.

Scientific context: the UE5 → FDS → OpenVDB → UE5 pipeline

The work reported here supports a single coherent scientific pipeline with four stages, each of which was developed, validated, and in most cases published or submitted for peer review during the reporting period.

Stage 1 - Procedural vegetation and fractal barycentric discretisation

Vegetation scenes are authored procedurally in UE5 using the Procedural Content Generation (PCG) framework, producing architecturally explicit tree crowns composed of trunk, branch, twig and foliage geometry, with biomass allocated between woody and foliar components according to published allometric relationships per species. A representative one-hectare wildland-urban interface (WUI) scene contains several million triangular facets across dozens of individual trees. The FDS Vegetation Exporter plugin converts this geometry into FDS Lagrangian particle initialisation files

through a fractal barycentric discretisation method: each foliage triangle is recursively subdivided into four congruent sub-triangles until the leaf-level sub-triangle area approaches the target FDS cell face area (dx^2); the barycentre of each leaf-level sub-triangle is mapped to its containing FDS voxel, which is then flagged as active and assigned a Lagrangian particle initialisation block. This is, to our knowledge, the first method to couple WFDS initialisation directly to the exact geometry of a procedurally generated, architecturally explicit tree crown, rather than to a simplified geometric proxy.

Stage 2 - EuroHPC simulation (FDS/WFDS)

The resulting case files, together with automatically generated FDS mesh layouts, terrain rasterisation (via Möller-Trumbore ray casting against a digital terrain actor), the validated Mell et al. vegetation combustion chemistry, and SLURM job scripts, are executed on EuroHPC clusters. [20]

Stage 3 - OpenVDB post-processing

FDS SMOKE3D binary output (soot density and temperature, run-length-encoded per the FDS source format) is decoded, dequantised against physical reference constants, assembled across all MPI meshes, and written as named OpenVDB volumetric grids (density, temperature, and scatter for excess water vapour) using the fds2vdb_parallel.py converter. [21]

Stage 4 - Real-time volumetric rendering in UE5

The OpenVDB grid sequences are imported into UE5 as Sparse Volume Texture (SVT) animations and rendered through the UE5 Heterogeneous Volume actor, closing the loop between procedurally authored vegetation and physics-based fire visualisation within a single environment. This is, to our knowledge, the first published pipeline coupling FDS SMOKE3D output to a real-time game-engine volumetric renderer. [23]

Table 7 summarises the available technical implementation information. The table distinguishes documented capabilities from information not established within the reported implementation.

Table 7. Wildfires technical implementation summary

Item	Implementation
Software and availability	WRF-Sfire, FSE, FDS/WFDS and UE5/OpenVDB tools. The supplied account identifies FDS 6.10.1, UE5.7, the FDS Vegetation Exporter v3.2 and OpenVDB 12.0.0. Component repository links and licensing remain to be consolidated by MTG.
Build process and dependencies	EasyBuild and EESSI are used to facilitate WRF deployment; a WRF-Sfire EasyBuild recipe has been submitted. The visualisation pipeline uses UE5 tools and Python-based FDS-to-OpenVDB conversion. Component-specific build instructions are not consolidated in this contribution.

Item	Implementation
CI workflow and automated tests	Not documented in the reported implementation.
Main implemented features	Landscape fire-risk and uncertainty workflows; CPU/GPU FSE; procedural vegetation and LiDAR-based scene preparation; FDS/WFDS simulation; and OpenVDB-based volumetric visualisation in UE5.
Inputs and outputs	Inputs include meteorological and terrain data, vegetation scenes and LiDAR point clouds in LAS format. The urban pipeline generates FDS input files and SMOKE3D binary outputs, which are converted to OpenVDB grids for visualisation in UE5.
Deployment and execution	MTG reports Iberian Peninsula workflows using Discoverer for WRF/WRF-Sfire and MeluXina for FSE. The supplied account also reports FDS/WFDS and visualisation-pipeline validation cases on EuroHPC resources.
Validation and documentation	Section 6.2 describes ensemble studies, vegetation-validation cases and end-to-end visualisation results. Section 6.3 distinguishes remaining software limitations and continuation work; tight MesoNH-FSE coupling is not reported as implemented. Core technology references are provided.

6.2 Main results and achievements

Landscape scenario

A landscape-scale uncertainty study was conducted to assess the variability in wildfire spread predictions under different sources of uncertainty, namely: model, meteorological and detection/terrain uncertainties. In order to assess the combined uncertainty a set of one-dimensional and areal variables were analysed including the main axis length, main axis direction, moment of inertia, total burnt area, perimeter, probability of burnt and the Jaccard index. The methodology adopted and the results obtained have been submitted to the X International Conference on Forest Fire Research, Coimbra November 2026.

In this context, the MeteoGrid FSE (Fire Spread Engine) solution has been implemented on GPUs. This approach is computationally less demanding and enables HPDA and UQ workflows but we still are in the final development phase. This work not only addresses the development of the wildfire and smoke dispersion simulator but also the adaptation of input variables from ECMWF, WRF, or MesoNH to the resolution required for the simulation and the tile management necessary to enable the process to run efficiently in HPC or GPU environments.

A process pipeline is currently being developed to conduct operational tests that evaluate the feasibility of running fire-spread simulations at a speed faster than real

time. This involves running WRF for the Iberian Peninsula at a resolution of one kilometer on Discoverer, transferring the resulting forecasts to the HiDALGO2 DMS (WP4), and then downloading them to another computer running FSE. A simpler version that takes advantage of the recent deployment of a GPU server on Discoverer is also planned; a new grant has been requested for this, which extends beyond the end of the project.

Publications and dissemination

Two papers have passed peer review at the X International Conference on Forest Fire Research (Advances on Forest Fire Research, 2026, in press), a first one addressing WRF/Sfire Uncertainty Quantification and a second one on surrogate models for speed-up WRF/Sfire simulations:

- Salazar, L.L., Caballero, D., Torres, L., Rivera, A. (2026) Uncertainty Quantification in Landscape-Scale Wildfire Spread Modelling: An Ensemble Approach Using WRF-SFIRE and a Reduced-Complexity Fire Spread Engine. Advances on Forest Fire Research - 2026 D. X. Viegas & L.M. Ribeiro (Ed.).
- Filandrianos G., Dimitriou A., Caballero D., Torres L., Stamou G. (2026) A Retrieval-Based Approach for Real-Time Wildfire Analysis Using LSTM-Driven Similarity Matching. Advances on Forest Fire Research - 2026 D. X. Viegas & L.M. Ribeiro (Ed.).

Another paper addressing deep learning for geographical risk analysis from ensemble WRF/Sfire simulations was submitted to PPAM 2026:

- Lawenda M., Krasicka A., Caballero D., Torres L. and Szustak L. Interpretable Patch-Based Deep Learning for Wildfire Spread Prediction from Ensemble Simulations. Workshop on Advancements of Global Challenges Applications. PPAM 2026, Poznań, Poland, September 2026.

Wildland-Urban scenario

FDS Vegetation Exporter (UE5 plugin)

The FDS Vegetation Exporter is a C++ editor plugin for UE5.7 that constitutes the first stage of the pipeline described before. It is stable at version 3.2, which is the current production anchor point.

Core capabilities (stable, v3.2)

- Fractal barycentric discretisation of UE5 static-mesh foliage triangles into FDS &INIT Lagrangian particle blocks.
- Automatic FDS mesh layout calculation with configurable target cells per MPI rank, updated from the previous empirical value of 170,000 cells/core to the empirically derived sweet spot of 12,500 cells/core following the strong scaling study that will be reported in HiDALGO2 D3.3 Scalability, Optimization and Co-Design Activities.

- Full FDS case file generation (.fds), including vegetation chemistry (Mell model via the external vegetation_model.txt), ignitor configuration, SLCF diagnostic slices, and SMOKE3D volumetric output.
- Terrain rasterisation via Möller-Trumbore ray casting against a terrain actor (FDSTerrainSampler).
- SLURM job script generation, with multi-cluster support for LUMI and MeluXina.
- Viewport voxel preview via a transient ProceduralMeshComponent actor.

Validation cases include Picea abies grove and picea_medium on LUMI (32 MPI cores, 20 cm resolution, 18,202 occupied voxels, 3.21 kg dry mass).

Known limitation and planned v3.3

The plugin's &SM3D output block does not currently request QUANTITY='SOOT DENSITY', which is required by the downstream OpenVDB converter; this is scheduled for v3.3. Additional diagnostic outputs planned for v3.3 are: &DEVC entries for Umax/Vmax/Wmax with SPATIAL_STATISTIC='MAX' for CFL diagnostics; &SLCF VELOCITY and TEMPERATURE slices on the central PBY plane and at PBZ=10; and a PARTICLE MASS FRACTION quantity in &PART QUANTITIES to support progressive foliage combustion rendering.

LiDAR-to-FDS converter (las2fds_meluxina.py)

This converter transforms voxelized terrestrial laser scanning (TLS) point clouds, in LAS format, into FDS case files for execution on MeluXina, providing an independent, field-measured route into the same Lagrangian particle initialisation used by the UE5 plugin. It has been validated on Pinus pinaster and Pinus sylvestris plots (plot_121-plot_134) supplied by Universidad de Alcalá de Henares, giving the pipeline a physically measured reference dataset against which the UE5-procedural route can eventually be cross-validated.

FDS-to-OpenVDB converter (fds2vdb_parallel.py)

This converter decodes FDS SMOKE3D binary output and produces OpenVDB grid sequences for import into UE5. Three named grids are written per frame: density (soot, normalised to a fixed reference of 0.001 kg/m³), temperature (normalised to 1800°C), and scatter (excess water vapour above ambient, normalised to 0.10 kg/kg). Using OpenVDB 12.0.0 compiled from source, the converter processes a 171-frame sequence (144 MPI meshes, 9.2 million cells, CSIRO Case F19 reference simulation) in under 3 minutes on a single MeluXina CPU node using Python's multiprocessing.Pool across 32 worker processes. Output file sizes range from approximately 5 MB per frame at ignition to 12 MB at peak fire spread, reflecting the sparse nature of the OpenVDB format.

The next version of the converter will replace the water-vapor scatter channel with a PARTICLE MASS FRACTION channel derived from the FDS .prt5 Lagrangian particle output (0 = fully burned, 1 = intact), enabling progressive foliage combustion to be

rendered directly in the UE5 material graph rather than only inferred from smoke and temperature.

Source-code finding

Direct inspection of the FDS 6.10.1 source code installed on LUMI (master branch) shows that read.f90, line 13410, assigns LPC%POROUS_VOLUME_FRACTION = IN%MASS_PER_VOLUME / LPC%DENSITY inside the loop that processes all &INIT namelists during the input-reading phase. LPC (Lagrangian Particle Class) is a class-level object shared by every particle of a given PART_ID; each successive &INIT block sharing that PART_ID overwrites the previous value, so only the last &INIT processed determines the class's porous volume fraction for the remainder of the run. When individual particles are subsequently created (part.f90, line 1682), LP_VOLUME = LPC%POROUS_VOLUME_FRACTION × LP%DX × LP%DY × LP%DZ, so every particle of a given class receives an identical volume and mass regardless of the MASS_PER_VOLUME specified in its originating &INIT block.

The architecturally correct solution requires a per-particle (not per-class) mass field, populated via parallel binary I/O rather than the namelist mechanism. The proposed fork adds a BINARY_FILE keyword to &INIT (e.g. &INIT PART_ID='foliage picea', BINARY_FILE='foliage_init.bin' /), with each MPI rank reading its own particles' pre-computed mass via MPI_File_read_at, bypassing LPC%POROUS_VOLUME_FRACTION entirely.

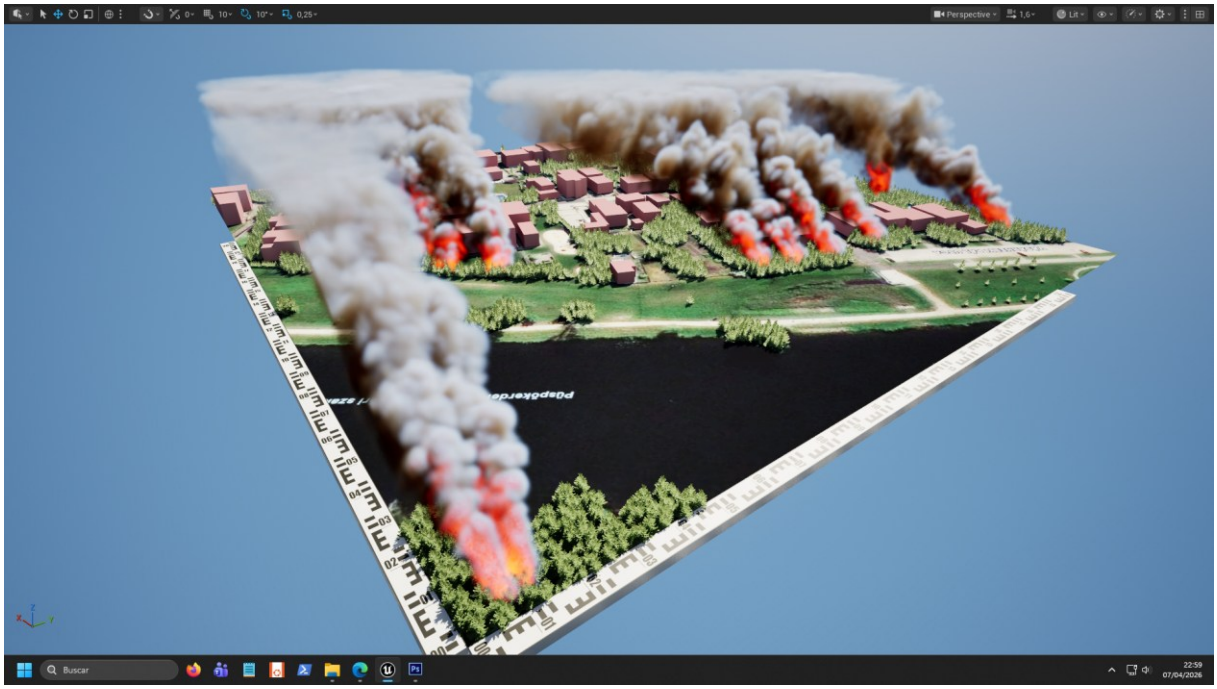
This work is currently underway in partnership with William (Ruddy) Mell (USDA Forest Service, WFDS author) and constitutes the scientific motivation for Paper 2 of the pipeline . This fork will not materially improve weak-scaling memory behaviour, since Lagrangian particles are not the dominant memory cost; its value is exclusively physical and scientific.

Real-time volumetric visualisation pipeline

FDS SMOKE3D output is stored per-mesh as run-length-encoded binary sequences; the RLE scheme was recovered directly from the FDS source (smvv.f90, SUBROUTINE RLE) and independently re-implemented in fds2vdb_parallel.py. Physical values are recovered via channel-specific dequantisation - soot density from a per-frame linear scale stored in the .s3d.sz header, temperature from a fixed [0°C, 2,500°C] mapping - and assembled across all MPI meshes into a single global volumetric array per frame using the spatial transforms stored in the SMOKE3D headers, ensuring correct metric placement. The assembled fields are normalised to fixed physical constants for cross-case comparability (density to [0, 0.001 kg/m³], temperature to [0°C, 1800°C]) and written as named OpenVDB grids using the OpenVDB 12.0.0 Python binding.

Figure 4 illustrates a fire simulation in Győr, including secondary spots caused by the spread of embers.

Figure 4. Simulation of a fire in Györ with secondary spots caused by the spread of embers



Production-scale validation converted the 171-frame active-propagation window (t = 29.3-60.0 s) of the CSIRO Case F19 reference simulation in under 3 minutes on a single MeluXina CPU node using 32-way process parallelism. The resulting VDB sequence is imported into UE5 as a Sparse Volume Texture animation and assigned to a Heterogeneous Volume actor, with density mapped to channel R, temperature to channel G, and (in the current production configuration) scatter to the R channel of Attributes B, all at 32-bit float precision. The UE5 volumetric renderer operates with single scattering and a single directional light; indirect illumination and Skylight are not supported in this render path. Within the material, absorption is driven by the density channel scaled by a configurable extinction coefficient, and thermal emission by a remapped temperature ramp transitioning from transparent below approximately 600°C to orange-red emission above it, yielding a physically informed, hardware-accelerated animation at 60+ fps on a mid-range desktop GPU. To our knowledge this is the first published pipeline coupling FDS SMOKE3D output to a real-time game-engine volumetric renderer.

Publications and dissemination

Three peer-reviewed or submitted works document this pipeline and its scalability characterisation, together with formal technical reports maintained as internal HiDALGO2 records:

- Paper 1 (in press, X International Conference on Forest Fire Research, Advances on Forest Fire Research, 2026): real-time physics-based fire visualisation via OpenVDB and UE5 Sparse Volume Textures.

- Paper 2 (in press, same venue and volume): the fractal barycentric method for Lagrangian particle initialisation from UE5 procedural vegetation, with the homogeneous-MPV limitation documented as motivation for the feature/binary-particle-init fork.
- Paper 3 (submitted, PPAM 2026): FDS/WFDS scalability on EuroHPC infrastructure, consolidating the strong- and weak-scaling studies and the cross-platform evaluation into a single peer-reviewed account, including some EuroHPC operational recommendations.

6.3 Remaining limitations and perspectives

Software

- The main limitation encountered with WRF-Sfire are the compilation complexity as well as the difficulty in running resource-intensive configurations due to the use of shared memory. As a result, simulations that run well on some servers may not run well on others.
- One possible future solution to the aforementioned limitation would be to replace WRF-Sfire with MesoNH-FSE. Integrate FSE with MesoNH to provide a solution better suited to the HPC/GPU environment, enabling the solutions that have been tested on the Iberian Peninsula to be scaled up across all of Europe. A tight coupling of these systems was not part of the objectives of HiDALGO2 and has not yet been addressed, but it shows promise.
- FDS Vegetation Exporter v3.3: add QUANTITY='SOOT DENSITY' to the &SM3D block; add Umax/Vmax/Wmax CFL diagnostics; add central-plane VELOCITY/TEMPERATURE slices; add PARTICLE MASS FRACTION output.
- fds2vdb_parallel.py: add the PARTICLE MASS FRACTION VDB channel from .prt5 output, replacing the water-vapor scatter channel, to support progressive foliage-combustion rendering in UE5.
- Continue development of the feature/binary-particle-init FDS fork for MPI-IO per-particle mass initialisation, in coordination with William Mell.

Scientific development

- Validate, using real data, the solution designed to automate the quantification of meteorological and terrain uncertainty.
- Digital arboretum of Mediterranean species (Pinus halepensis, P. pinaster, Quercus ilex, Q. suber, Arbutus unedo): architecturally correct 3D models for dual use in UE5 visualisation and FDS parameter extraction, using a SpeedTree surrogate model trained on synthetic data and fine-tuned with TLS field data, in

collaboration with CIRAD-AMAP (Frédéric Boudon, Christophe Pradal) and the Ghent TLS group (Kim Calders).

- Longer-term strategic assessment of migration to OpenFOAM/fireFoam with fvDOM radiation for landscape-scale work, given FDS's structural scalability limitations (fixed geometric decomposition, FFT-based pressure solver, absence of dynamic load balancing).

Conclusions

Two new tools considered promising for risk management have been fully developed during this period and documented in scientific articles: risk analysis based on hundreds of simulations and the measurement of uncertainty associated with meteorological uncertainty. Furthermore, having a process in place to conduct fire-spread simulations across the Iberian Peninsula opens up the possibility of conducting operational simulations across Europe based on satellite detection of hot spots.

As for the wildland urban scenario the reporting period produced a validated, end-to-end, and to our knowledge novel pipeline from procedurally authored vegetation to physically-based fire simulation on EuroHPC infrastructure and back to real-time volumetric visualisation, together with the formal strong- and weak-scaling characterisation required as a HiDALGO2 project deliverable, and a source-code-level scientific finding (the homogeneous bulk-density limitation) that will shape the next phase of solver development in direct coordination with the WFDS author community. Two papers building on this work are in press and a third is under review.

The technical conclusion that carries the most weight for future EuroHPC resource requests is that FDS's own computational behaviour is well characterized, predictable, and efficient within a clearly defined operating envelope, and that the principal obstacle to exploiting larger EuroHPC allocations was, throughout this reporting period, not the physics code but the operational policies of the platforms themselves.

7 Material Transport in Water implementation status

Contribution lead: Ravi Kiran Ayyala (FAU)

Contributors: Harald Köstler (FAU)

The Material Transport in Water (MTW) pilot develops fully resolved simulations of particle-laden flows in water. The target applications are the transport of sediment, ash and other particulate material, together with the associated heat transfer, resolved at the scale of individual particles. The pilot is implemented in the open-source waLBerla framework [8]. Lattice Boltzmann (LB) kernels are generated with lbmpy [9], and particle dynamics are handled by the MESA-PD [14] module. This section reports the implementation reached at M46 and the developments completed since D5.7 [2].

7.1 Final implementation status and developments since D5.7

D5.7 [2] reported the MTW implementation at two levels. At the fluid level, the flow and temperature fields were coupled in both directions and this coupling was validated. At the particle level, the hydrodynamic coupling between fluid and resolved particles was implemented using the partially saturated cells method (PSM), and thermal particles were incorporated for the first time, with each particle held at a uniform temperature. Since D5.7, the uniform-temperature treatment of the particles has been extended to a total-enthalpy-based conjugate heat transfer (CHT) model for moving, fully resolved particles. This model was implemented, validated and tested in waLBerla, and the results have been published [10]. Building on this work, a thermal turbulent channel application for particle-laden flow has been implemented. It adds new collision operators, controller-based flow forcing on both CPU and GPU, and a set of hand-written components for statistics, wall quantities and restart-files.

Total-enthalpy based conjugate heat transfer in particle-laden flows

Model: The model couples heat conduction inside the particles, convection and conduction in the fluid, and heat exchange across the moving interface. It follows the smoothed-profile LB approach of Hu et al. [11], reformulated with a volume-of-body (VoB) virtual-fluid description.

A single energy equation is solved over the whole fluid-solid system. The effective heat capacity, thermal conductivity and velocity in this equation are each weighted by the local solid fraction. The hydrodynamic PSM already enforces no-slip at the particle surface, so the thermal interface conditions are satisfied without a separate boundary treatment.

The energy equation is solved with a second set of distribution functions on a D3Q19 lattice. Its equilibrium distribution uses a reference heat capacity as a numerical stabilisation parameter for large solid-to-fluid heat-capacity contrasts. Two choices of

the reference specific heat are implemented: the harmonic mean of the solid and fluid values, or their minimum. A uniform volumetric heat source inside the particles and Boussinesq buoyancy forcing of the fluid are also implemented

Collision operators: The thermal populations can use either a single-relaxation-time (SRT) or a two-relaxation-time (TRT) operator, the latter with magic parameter $\Lambda = 1/4$ and the antisymmetric relaxation time is set from the prescribed thermal diffusivity.

Code generation: All kernels are generated with lbmpy for CPUs (C++) and AMD GPUs (HIP) and integrated into waLBerla. This covers the hydrodynamic and thermal LB updates, boundary handling, communication and field operations. The hybrid CPU-GPU arrangement of Kemmler et al. [12] was extended to carry the thermal distribution function. The code and data are openly available [13].

Application and extensions of the validated models on turbulent channel flows

The MTW pilot is aimed at simulating thermal turbulent channel flows with fully resolved particles, as a way to study how temperature and sediment are transported in water. The CHT model described above was a necessary step towards this goal. It provided a validated treatment of heat transfer between the fluid and moving particles, and the turbulent channel scenario builds on it. The final objective is to simulate complex particle-laden thermal turbulent channel flows at a scale that requires large HPC infrastructure. The application has a CPU version and a GPU version. Both are built from the same lbmpy-generated kernels and share the hand-written components described below.

Generated kernels: One Python generation script produces all LB, boundary and statistics kernels, for CPU (C++) or GPU (CUDA and HIP) depending on the build. It emits the kernels in single or double precision, and GPU thread-block sizes are exposed as runtime parameters.

Collision operators: The fluid uses the cumulant collision operator on a D3Q19/ D3Q27 stencil, combined with the PSM coupling and Guo forcing. The cumulant operator was chosen for its stability at the low viscosities required for turbulent flow. Similarly for the thermal populations, a dedicated thermal method builder was implemented on top of lbmpy for the advection-diffusion equation on D3Q27. The equilibrium is a second-order continuous Maxwellian: temperature is its zeroth moment. The thermal populations are further extended with the following collision operators:

Central-moment-srt: all the relaxation rates of all the moments are set according to the thermal diffusivity of the fluid.

Central-moment-1st: Only the first order moments are relaxed with the thermal diffusivity of the fluid and all other moments are relaxed with 1.

Central-moment-trt: The odd moments are relaxed based on the thermal diffusivity and the even moments are relaxed with either 1 or based on a certain magic parameter as defined in the LBM literature.

For all the central-moment operators, each moment order can be given its own relaxation rate as described above, up to sixth order. The zeroth-order moment is conserved. The builder rejects configurations it does not support, such as zero-centred storage or thermal forcing, so that inconsistent kernels cannot be generated.

Post-processing utilities: The application includes post-processing utilities for analysing the turbulent flow and heat transfer during the simulation. Some are hand-written CPU routines while some of them are generated with lbmpy, so the same kernel definition can produce kernels for both CPU and GPU architectures. The utilities are summarised in Table 8.

Table 8. Post-processing utilities implemented in the thermal turbulent channel application

Routine / Kernel	Architecture (manual or code-generated)	Functionality
Wall-normal shear-stress profiles	CPU (manual)	Computes the stress profiles across the wall-normal directions
Wall-normal phase-resolved profiles of average and RMS quantities	CPU (manual)	Computes phase-specific average and root mean square velocities and temperatures of fluid and particles
Heat-flux budgets	CPU (manual)	Computes the heat flux budgets to analyse the contributions of solid, fluid convective and conductive contributions of heat transfer
Welford mean and variances	CPU/GPU (code-generated)	Computes running time averages and variances of velocity and temperature at every cell in a single pass, without storing the time series.

Checkpointing and restart capabilities: Checkpointing restart capability was implemented for simulations that exceed the wall-time limits of production queues. Checkpoints contain:

- the block structure;
- fluid and thermal PDFs;
- the particle storage, including contact history;
- the Welford mean fields;
- the phase-resolved time-averaged profiles;
- a metadata record with the time step, current PID forcing, Welford sample count, stationarity flag and statistics-reset state.

Table 9 summarises the available technical implementation information. The table distinguishes documented capabilities from information not established within the reported implementation.

Table 9. Material Transport in Water technical implementation summary

Item	Implementation
Software and availability	The pilot uses waLBerla, lbmpy and MESA-PD. The contribution provides references for waLBerla/lbmpy and the published heat-transfer implementation and data. waLBerla is licensed under GPLv3.
Build process and dependencies	Python-based kernel generation produces C++ CPU and CUDA/HIP GPU implementations, integrated with waLBerla. LB, boundary and statistics kernels are generated; manual routines provide additional post-processing. The waLBerla build instructions are available at https://www.walberla.net/doxygen/setup-instructions.html. More detailed application build instructions are planned for a README by the end of M48; these are distinct from the existing framework build instructions.
CI workflow and automated tests	waLBerla uses gitlab runners for the CI/CD pipelines. The pipeline has several tests written and upon pushing codes to gitlab, the pipelines trigger automatically and the CI/CD pipeline is fully automated.
Main implemented features	Total-enthalpy conjugate heat transfer for resolved moving particles; thermal turbulent-channel simulation; CPU/GPU code generation; runtime statistics and wall quantities; and checkpoint/restart.
Inputs and outputs	The model uses fluid/particle and thermal parameters. Outputs include velocity/temperature statistics, shear-stress profiles, heat-flux budgets and restart data. Exact configuration, result and checkpoint file formats remain to be specified.
Deployment and execution	CPU and GPU implementations are described. Checkpoint/restart supports runs exceeding production-queue wall-time limits.
Validation and documentation	Conjugate heat transfer has been evaluated against three 3D benchmarks at Prandtl numbers 0.7, 7 and 20. At the reporting point, 25 ensemble simulations have been completed and 10 further runs are expected by the end of M46. Detailed uncertainty-quantification results are planned for D4.10.

7.2 Main results and achievements

- The CHT implementation was verified and validated against three 3D benchmark problems. Together they cover Prandtl numbers of 0.7, 7 and 20, and a range of solid-to-fluid conductivity and heat-capacity ratios.
- Ensemble simulations of laminar and turbulent particle-laden heat transfer in channel flows are ongoing at M46. At the reporting point, 25 simulations have been completed and 10 further runs are expected by the end of M46, for an expected total of 35. Detailed results will be included in D4.10 on uncertainty quantification, due at the end of M47.

7.3 Remaining limitations and perspectives

CHT model:

- Heat transfer through particle-particle and particle-wall contacts is not included.

Turbulent channel application:

- The application has currently only been implemented and tested for spherical particles and extension towards other particle shapes would be an interesting future direction.
- Hybrid LB-FV (lattice Boltzmann - finite volume) or LB-FD (lattice Boltzmann - finite difference) approaches would be another interesting direction as they are gaining more popularity in the recent years and incorporating such strategies can help solve complex parameter ranges where the current thermal Lattice Boltzmann solver can struggle with.

8 Pilot couplings implementation status

This section updates the six envisaged pilot couplings from D5.7. Each assessment identifies the implemented exchange mechanism, the results demonstrated and the remaining limits. A one-way exchange is not automatically incomplete: its status depends on whether it meets the intended workflow and has been demonstrated or used operationally.

Each reported coupling is classified as operational, demonstrated, partial or not pursued. Table 10 summarises the partners and status at M46. The classification distinguishes implemented components and demonstrations from end-to-end operational workflows.

Operational: the reported coupled workflow runs end to end on HPC resources and its routine use or results are documented.

Demonstrated: the implemented exchange has been tested on an identified case; this does not establish routine production use.

Partial: an interface or component is available, but the intended end-to-end integration or validation has not been established.

Not pursued: the envisaged coupling was not implemented, with the reason stated.

Table 10. Summary of the envisaged pilot couplings

Coupling	Partners	Status at M46
UAP-UBM	SZE, UNISTRA	Demonstrated: KUB-StationReplay interface exchanges. Partial: complete two-way integration with the RedSim/OpenFOAM UAP solver is not demonstrated in this report.
UBM-RES	UNISTRA, PSNC	Partial: RES-PV and UBM generate data in separate runs; combined RES-energy consumption and end-to-end QCG execution are not demonstrated.
UAP-WF	SZE, MTG	Partial: WRF-Sfire outputs generated; end-to-end UAP ingestion and concentration validation are not demonstrated.
UBM-WRF	UNISTRA, MTG	Demonstrated for Strasbourg; end-to-end target-system deployment is not demonstrated.

Coupling	Partners	Status at M46
RES-WRF	PSNC, MTG	Operational for the existing RES-WRF workflow; end-to-end MTG-specific integration is not demonstrated.
MTW-WF	FAU, MTG	Not pursued (physical coupling); water visualisation available.

The reported implementations include file-based data exchange, the Feel++-OpenFOAM solver-adapter benchmark and the KUB-StationReplay demonstrator. Joint UAP-UBM district feedback, the complete UBM-RES chain, UAP-WF ingestion and the MTG-specific RES-WRF configuration are outside the demonstrated end-to-end scope. MTW-WF was not pursued; its section records the rationale and the available water visualisation.

8.1 UAP-UBM coupling

Contributors: Zoltán Horváth and Mátyás Constans (SZE), Christophe Prud'homme and Vincent Chabannes (UNISTRA)

Deliverable D5.7 had previously demonstrated a one-way transfer of surface temperatures from the UBM model to the UAP/OpenFOAM workflow on the Lingolsheim case study. This initial coupling was carried out through disk-based file read and write operations, which represented a major drawback in terms of performance and execution time. The direct preCICE interface extends this approach to exchanges between the Feel++ building simulation core (written in C++) and an external participant. To orchestrate this complex interaction, our approach relies on integrating the preCICE coupler. [2]

preCICE (Precise Code Interaction Coupling Environment) is a state-of-the-art, open-source coupling library specifically designed for partitioned multi-physics simulations. One of its fundamental strengths is being fully adapted to high-performance computing (HPC) environments. It can run efficiently on multi-node supercomputers, managing communications in a robust, distributed, and direct manner (via MPI or TCP/IP). Using preCICE thus enables peer-to-peer data exchange in memory, completely eliminating the major disk I/O bottleneck of the previous version. Furthermore, it handles spatial interpolation between non-matching meshes and integrates advanced coupling schemes to guarantee numerical stability. [15]

To connect our domain-specific environment based on the Feel++ library to the preCICE ecosystem, we have undertaken the development of a dedicated software adapter. This bidirectional interface, which is exercised by the StationReplay

demonstrator, has been entirely written using the modern C++23 standard, allowing us to optimise performance and integration with existing data structures.

Although the full-scale coupling between KUB and UAP at the urban scale is not yet complete, the adapter's operation has been validated through a proof of concept. To do this, we implemented the classic conjugate heat transfer benchmark, flow-over-heated-plate. This benchmark aims to replicate and validate on a small scale the exact mechanics of the final intended coupling between KUB and UAP. In this test configuration:

- The thermal dynamics of the solid part are solved by our kub-feel++ solver.
- The fluid dynamics are handled by OpenFOAM.

This test successfully verified the direct exchange of fields at the interfaces via our preCICE adapter. Figure 5 shows the temperature field and Figure 6 the velocity magnitude in the flow-over-heated-plate demonstrator.

The current benchmark effectively demonstrates the essential interface quantities required for the coupling: temperature and heat flux. While additional variables could technically be exchanged, these two core quantities perfectly illustrate the two fundamental types of data handling required for the final urban-scale coupling:

- Continuous data (Temperature): Transferred via a consistent exchange method, such as linear interpolation, to ensure profile continuity across different mesh resolutions.
- Flux data (Heat Flux): Transferred via a conservative element-by-element exchange method to guarantee strict physical conservation of energy across the interface.

Furthermore, the quality of this data transfer must be carefully monitored and evaluated to ensure an accurate and reliable coupling

The heated-plate case validates the Feel++-OpenFOAM coupling infrastructure. Its application to the KUB building-model workflow requires a separate pilot-level integration, described below.

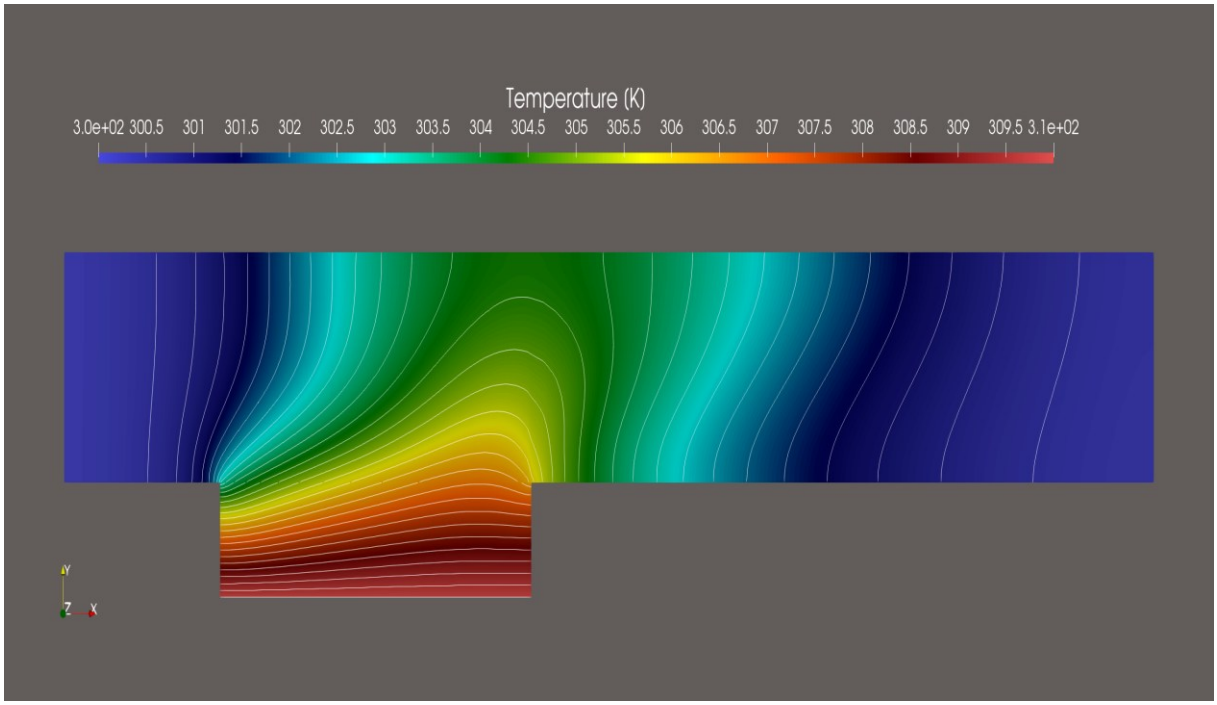


Figure 5. Temperature field in the flow-over-heated-plate demonstrator using kub-feel++ and OpenFOAM coupled through preCICE

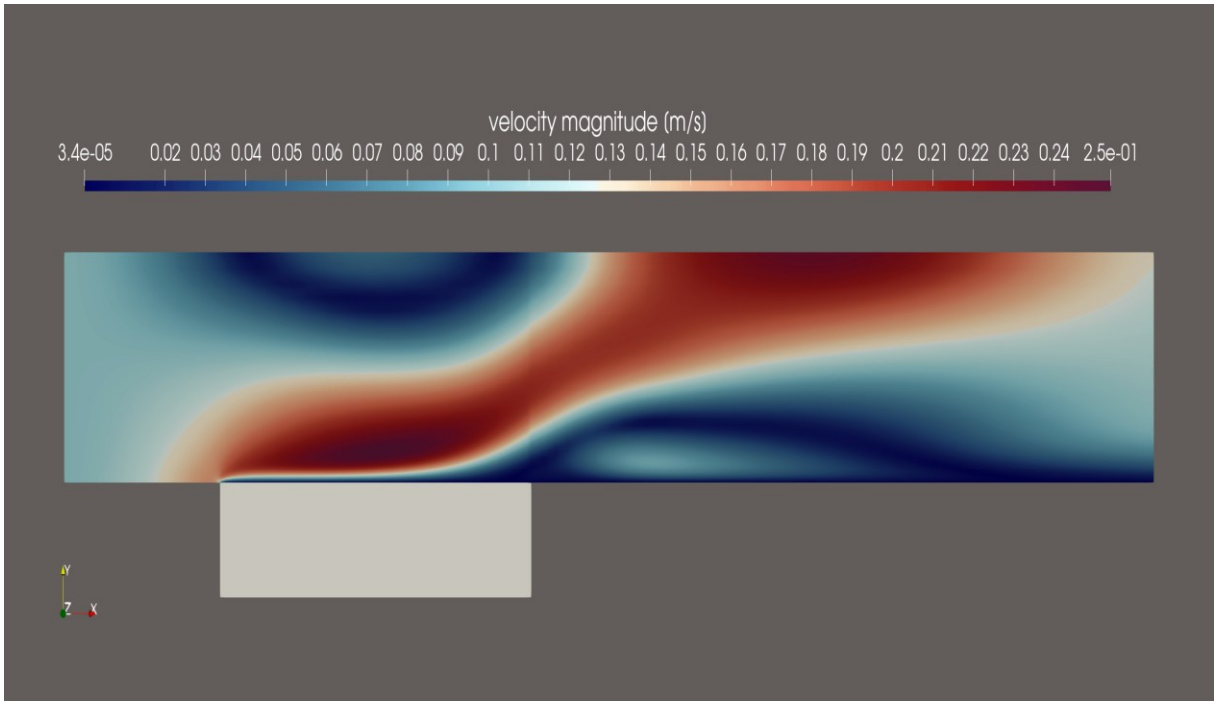


Figure 6. Velocity magnitude in the flow-over-heated-plate demonstrator using kub-feel++ and OpenFOAM coupled through preCICE

Common KUB coupling architecture. The pilot integration extends the same preCICE approach to Ktirio Urban Building (KUB) coupled with either OpenFOAM or RedSim for urban microclimate simulation. The building and CFD applications remain separate executables, each connected through its own adapter. Matching the exchanged quantities and interface conventions is intended to allow either microclimate solver to participate without redesigning the KUB coupling interface.

Exchanged quantities. The agreed pilot interface supplies building-surface temperatures from KUB to the microclimate solver, with heat fluxes available where required by the selected boundary-condition formulation. In return, the CFD solver supplies local air temperature and the three components of wind velocity near the buildings. KUB uses these data as outdoor conditions for its thermal and convective heat-transfer models, aggregating fields to the quantities required by each building model. Pollutant concentrations are a possible additional input for indoor air-quality studies; they are not part of a demonstrated coupled result at this stage.

Geometry and mapping. The two applications must describe the same physical building surfaces and preserve their identifiers, even when their numerical meshes differ. Face-centred data exchange is the starting point discussed for the adapters. Sampling must distinguish outdoor air values from wall values. Mapping choices, geometry simplifications and the treatment of ground temperature and radiative forcing require joint checks.

Parallel execution and synchronisation. The intended deployment uses separate Apptainer containers for KUB and the selected CFD solver, communicating through preCICE. Each application retains its own parallel execution. Coupling windows coordinate the exchange while allowing different internal time steps; a 30-minute exchange interval was discussed as an initial configuration, subject to testing. Joint container execution, MPI compatibility and synchronisation with RedSim/OpenFOAM are outside the StationReplay demonstration.

District demonstrators. The UAP-UBM demonstrators use two districts in the Strasbourg area. Wolfisheim provides a smaller neighbourhood case, while the Strasbourg 1 km domain represents a denser urban setting. Their different building layouts allow the coupling approach to be explored at neighbourhood and urban scales.

StationReplay demonstrator. The implemented KUB interface has been exercised with StationReplay, which supplies prescribed weather-station air temperature and wind and receives building-surface temperatures, solar shading and convective heat flux. This demonstrates the interface exchanges before connecting a UAP microclimate solver.

The Strasbourg visualisation in Figure 7 illustrates the building-surface temperatures exchanged through this interface. StationReplay uses prescribed weather; it does not simulate the response of the urban air to heat released by buildings. [24]

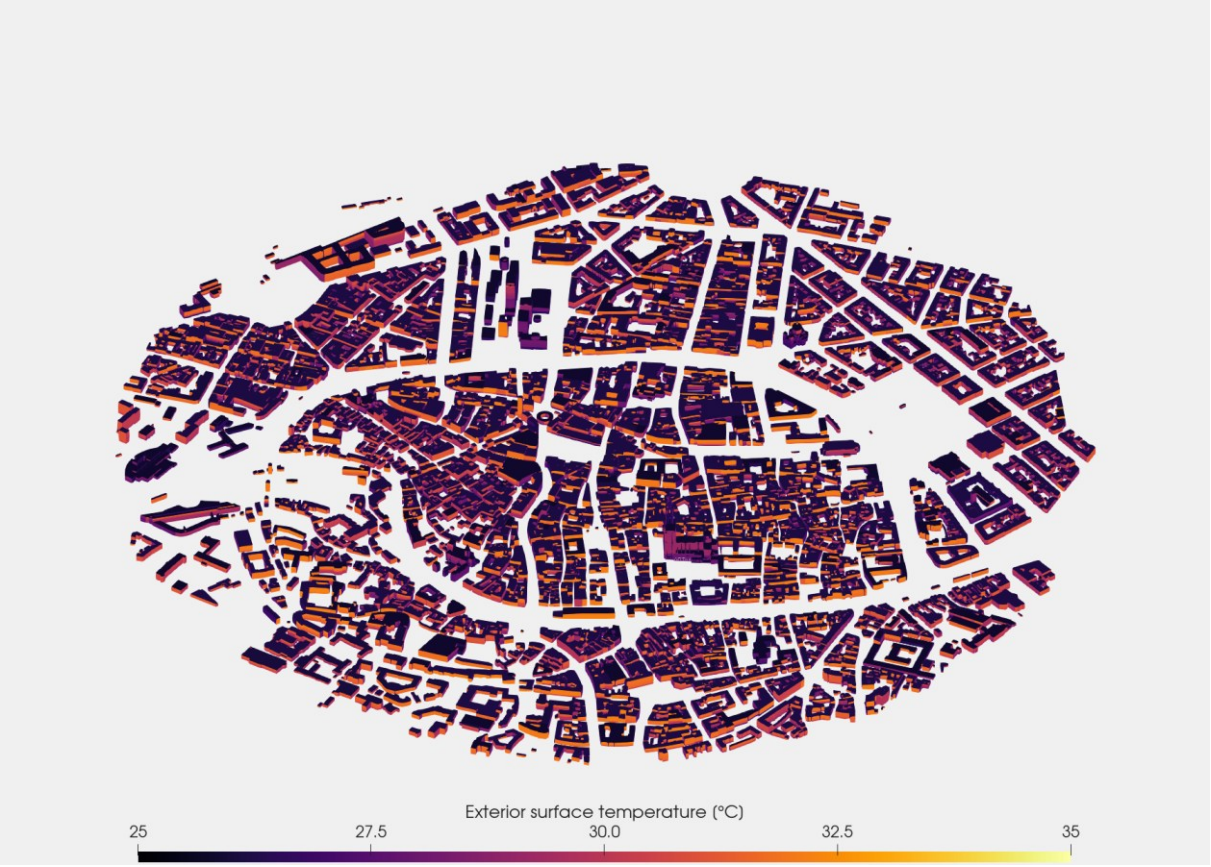


Figure 7. Building-surface temperatures in Strasbourg exchanged through StationReplay.

Status at M46. The KUB-StationReplay interface demonstrator is implemented and tested. Complete two-way integration with the RedSim/OpenFOAM UAP solver remains partial and is not demonstrated in this report. Container images are available to support this integration. D5.2 will report the coupled results and application synergies.

8.2 UBM-RES coupling

Contributors: Christophe Prud'homme and Javier Cladellas (UNISTRA), Michał Kulczewski and Krzysztof Kotecki (PSNC)

D5.7 described complementary UBM outputs to feed more data into RES-energy, an AI-based model for forecasting electrical energy production from photovoltaics. Subsequent work added solar-panel meshing in Ktirio-geom, see Figure 8, and reported the UBM-to-RES interface at v1.0. The UBM for PSNC's PV infrastructure is ready as a Docker container, which provides detailed solar-shading data for each photovoltaic panel. Solar zenith and azimuth angles are outside the current outputs; these quantities would further detail the radiation forecast used by RES-energy to predict PV production. RES-PV and UBM are prepared and data to be exchanged are generated during each separate run. Combined consumption of RES-PV and UBM data by RES-energy and end-to-end execution through the QCG Portal are not demonstrated by the reported results.

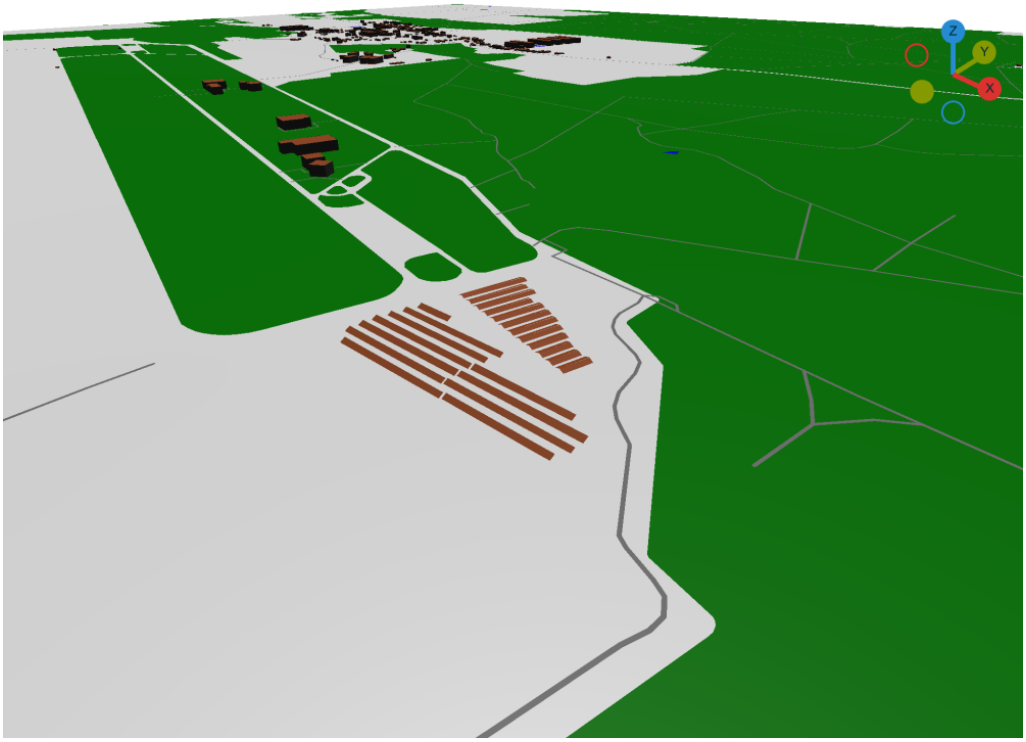


Figure 8. Detailed photovoltaic-panel mesh for the UBM-RES interface

8.3 UAP-WF coupling

Contributors: Zoltán Horváth (SZE) and Luis Torres (MTG)

Several WRF-Sfire simulations have been generated, including smoke dispersion with various pollutants taken into account, such as fine particles and toxic gases. These outputs provide the data for a WF-to-UAP workflow; end-to-end ingestion and input-format compatibility are not demonstrated by the reported results. Simulated concentrations have not been validated against observations, which are scarce beyond the affected areas.

8.4 UBM-WRF coupling

Contributors: Christophe Prud'homme and Philippe Pinçon (UNISTRA), Luis Torres (MTG)

Since D5.7 [2], one-way WRF-to-UBM coupling and visualisation of the meteorological results have been implemented.

The WRF simulations feeding KUB are produced by an automated workflow. The user first defines the simulation from the KUB GUI (study area, time period and main options), from which a script generates the WPS and WRF namelists, including the nested domains down to 1 km resolution. The static geographical data provided by UCAR (terrain, land use, soil types) are downloaded once and stored in a shared location within HiDALGO2, while the global ECMWF meteorological data used as initial and boundary conditions are retrieved for the requested period through the HiDALGO2 DMS (WP4). The simulation then runs either on a machine where WRF is already installed, or on a EuroHPC system, where the MATHSO orchestrator deploys a WRF container image, uploads the input data and runs WPS and WRF. The resulting outputs are stored in the HiDALGO2 DMS, from which they are retrieved for preprocessing. End-to-end execution and deployment on the target system are not demonstrated in this report. [18]

Deployment scope. WRF preprocessing and the KUB weather interface are implemented. End-to-end execution through MATHSO on the target system is outside the demonstrated scope. D5.2 will provide the fuller account of operational use and synergies.

The detailed meteorological fields from the WRF model are preprocessed to extract only the meteorological data needed for a KUB simulation into an HDF5 file. The following meteorological data are retrieved from the native WRF curvilinear grid, along with the corresponding timestamps: air temperature, relative humidity, wind speed and direction, solar direct and diffuse radiation, surface pressure and cloud cover. Figure 9

shows the Strasbourg metropolitan area in relation to the WRF temperature field and the simulated domain.

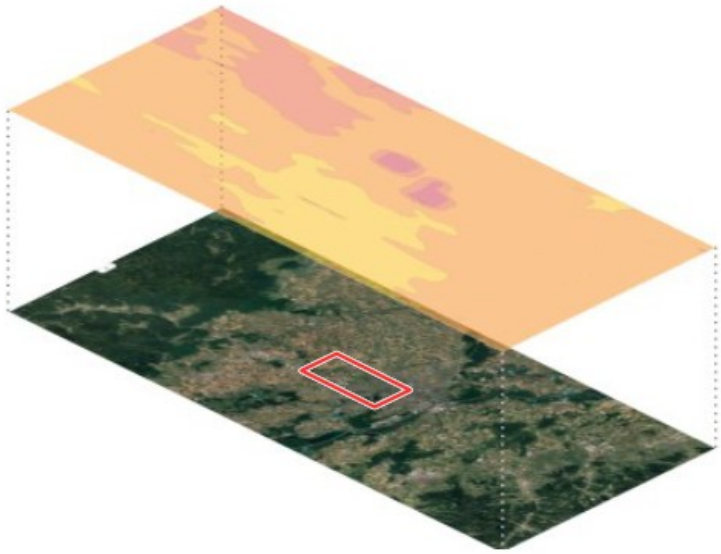


Figure 9. The Strasbourg metropolitan area overlaid on the temperature map generated by the WRF model. The simulated area is outlined in red.

The C++ weather module of KUB reads this aggregated HDF5 file and performs spatio-temporal interpolation at runtime within the Feel++ framework. A mesh is built directly on the native WRF grid and replicated on every MPI process. At the kilometre-scale WRF resolutions, the area covered by a city simulation amounts to at most a few thousand nodes, so the duplicated storage is negligible compared with the urban mesh it feeds. The meteorological fields are discretised with first-order Lagrange Q1 finite elements. Inputs and outputs are centralised: only the master rank keeps the HDF5 file open and reads the data for the current simulation time, which are then broadcast to all processes. Since the mesh is replicated, every rank requires the full slice, so a single read followed by a broadcast is preferable to having all ranks issue the same request to the HDF5 file. Each process then interpolates linearly in time and evaluates the resulting fields at the coordinates of the buildings registered on it.

This provides each building with its own spatially and temporally varying urban climate conditions rather than a single source of weather data taken from the nearest weather station. Figure 10 compares the outdoor temperature fields obtained using WRF inputs and weather-station data for downtown Strasbourg.

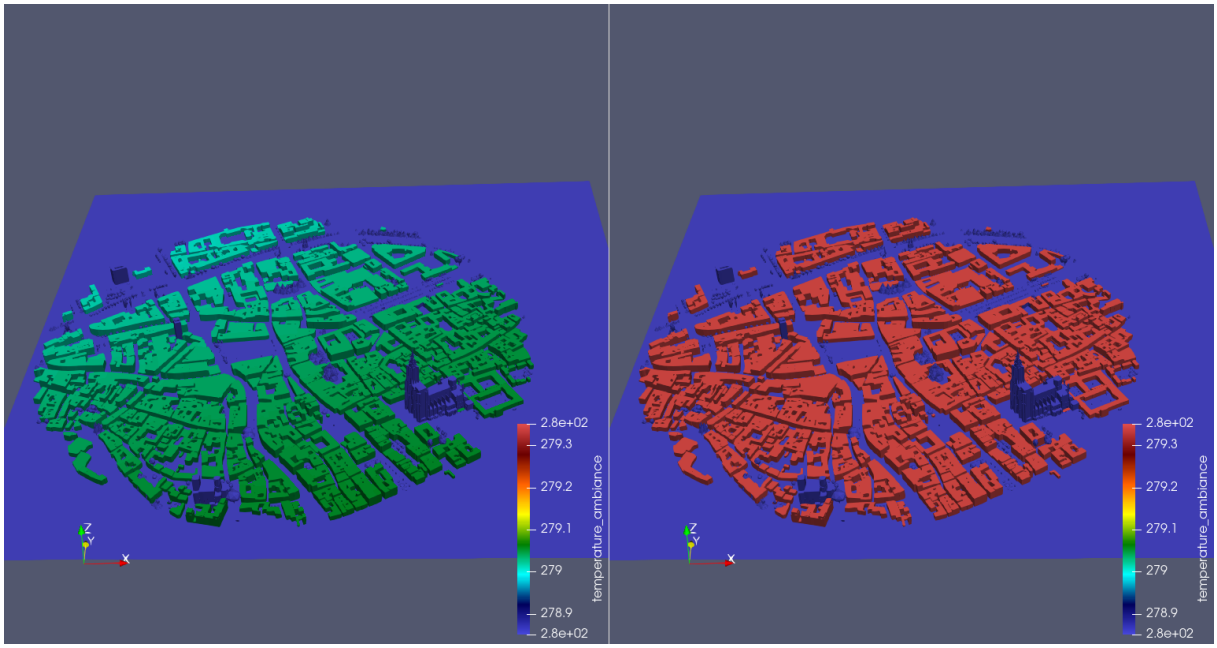


Figure 10. Comparison of outdoor temperatures in downtown Strasbourg (WRF on the left, weather station on the right)

8.5 RES-WRF coupling

Contributors: Michał Kulczewski, Wojciech Szeliga (PSNC) and Luis Torres (MTG)

The objective of this coupling is to provide more detailed data to the EULAG model, by using a more efficient and more detailed setup of the WRF-MTG. The coupling is done by exchanging data via files - the output of WRF-MTG is the input for EULAG. This loose coupling allows WRF-MTG and EULAG to run in a single workflow, where the execution of EULAG is postponed until WRF-MTG successfully finishes. Another possibility is to consume the data already produced by WRF-MTG by transferring them from the CKAN repository to the HPC site where EULAG will run. In either case, the workflow is managed by the QCG Portal, see Figure 11. The internal RES-WRF coupling is tested and finalised. End-to-end integration of the MTG-specific WRF configuration is outside the demonstrated implementation scope.

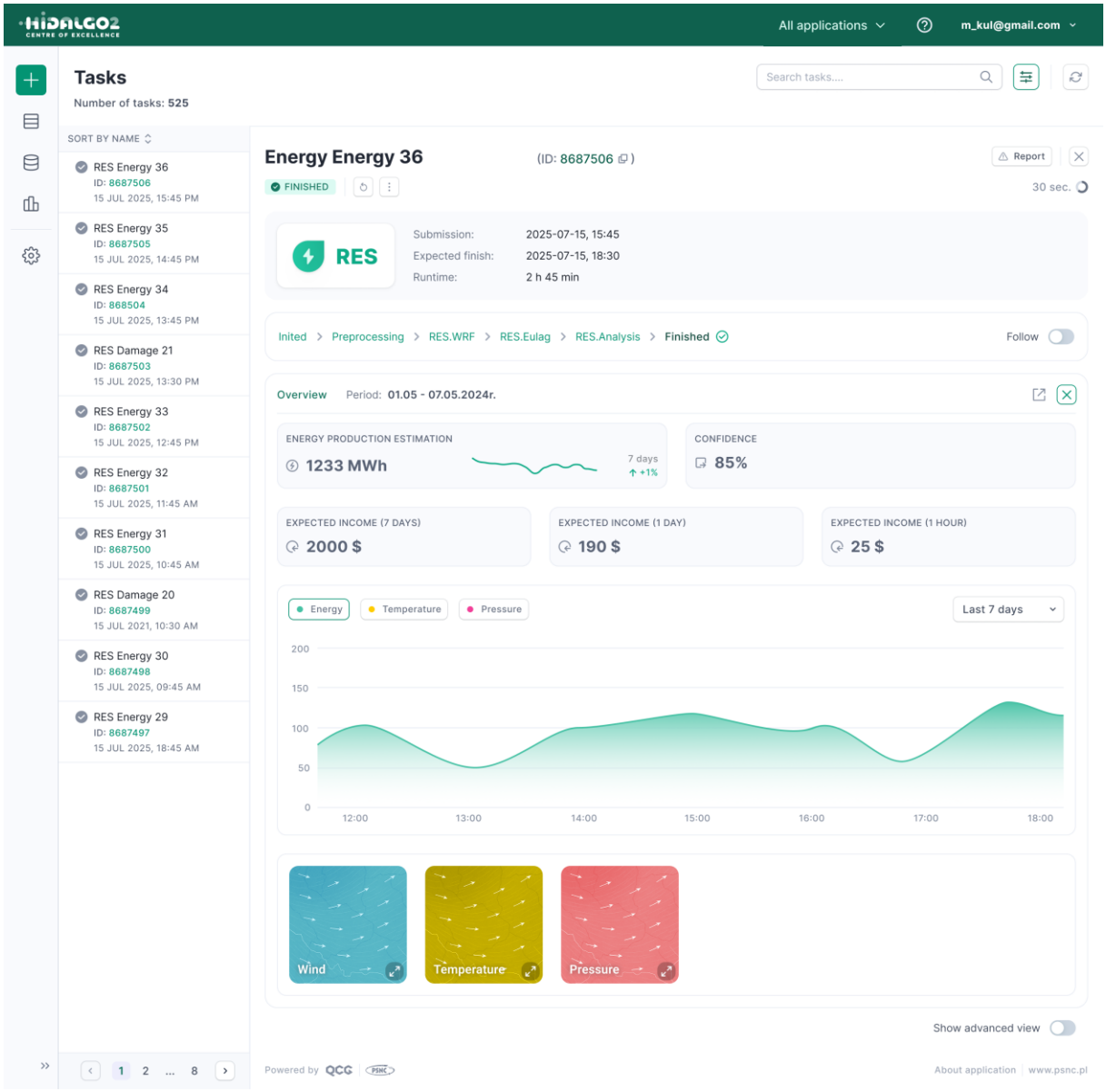


Figure 11. Coupled WRF-RES workflow in the QCG Portal

8.6 MTW-WF coupling

Contributors: Harald Köstler (FAU) and David Caballero (MTG)

The envisaged MTW-WF coupling concerned a visualisation of a flowing river in the vicinity of a wildfire. The physical coupling was not pursued because the simulation scales differ substantially. Water visualisation in Unreal Engine has been implemented, but no integrated MTW-WF demonstration is reported. Combined river and wildfire visualisation is a possible perspective for D5.2, not an implementation pending completion in D5.8.

8.7 Technical summary of coupling implementations

Table 11 consolidates the technical implementation evidence and remaining work reported in Sections 8.1-8.6.

Table 11. Technical summary of coupling implementations

Coupling	Exchanged quantities/data	Exchange mechanism/interface	Implemented components and demonstration evidence	Scope and limitations
UAP-UBM	Benchmark: temperature and heat flux. StationReplay to KUB: prescribed air temperature and wind. KUB to StationReplay: building-surface temperatures, solar shading and convective heat flux. Full UAP integration targets local CFD air temperature and wind; pollutants optional.	In-memory preCICE exchange exercised by StationReplay. Benchmark: consistent temperature mapping and conservative flux transfer. Integration of RedSim/OpenFOAM through the same KUB interface remains partial.	C++23 adapter; heated-plate conjugate heat-transfer proof of concept with kub-feel++ and OpenFOAM (Figure 5 and Figure 6). The KUB-StationReplay interface demonstrator is implemented and tested with prescribed weather and building-surface fields, illustrated for Strasbourg. Container images are available.	Interface demonstration completed; complete two-way integration with the RedSim/OpenFOAM UAP solver remains partial and is not demonstrated in this report. Further coupled results and application synergies are covered in D5.2.
UBM-RES	Panel geometry and solar-shading data; solar angles planned.	One-way UBM inputs for RES-energy alongside RES-PV outputs; combined consumption and end-to-end QCG execution are not demonstrated.	Interface v1.0 and Docker container for PSNC's PV infrastructure reported available; RES-PV/UBM then RES-energy sequence described.	Solar-angle outputs and end-to-end QCG execution are outside the implementation demonstrated here.

Coupling	Exchanged quantities/data	Exchange mechanism/interface	Implemented components and demonstration evidence	Scope and limitations
UAP-WF	WRF-Sfire smoke/pollutant outputs, including fine particles and toxic gases.	One-way WF-to-UAP ingestion; mechanism and format under verification.	WRF-Sfire simulations and pollutant data generated; no validated end-to-end ingestion reported.	End-to-end ingestion and input-format compatibility are not demonstrated. Concentration validation is limited by the available observations.
UBM-WRF	Temperature, humidity, wind, direct/diffuse solar radiation, pressure, cloud cover and timestamps.	HDF5 preprocessing; master-rank read/MPI broadcast; Q1 spatial and linear time interpolation.	C++ weather module and WRF preprocessing implemented; Strasbourg visualisation and weather-station comparison (Figure 9 and Figure 10). WPS/WRF generation and MATHSO workflow documented; end-to-end target-system execution is not demonstrated.	Target-system deployment and operational validation are outside the demonstrated scope.
RES-WRF	WRF meteorological output for EULAG.	File-based workflow: WRF precedes EULAG; QCG orchestration; CKAN transfer alternative.	Internal RES-WRF coupling tested/finalised; MTG-specific end-to-end integration is not demonstrated. Workflow illustrated in Figure 11.	The MTG-specific WRF-to-EULAG workflow is outside the end-to-end demonstrations reported here.
MTW-WF	No physical data exchange implemented; MTW-WF coupling not pursued.	Physical coupling not pursued because of different simulation scales; no integrated exchange mechanism reported.	Water visualisation in Unreal Engine reported; no integrated MTW-WF demonstration reported.	Physical coupling was not pursued. Combined visualisation is a perspective for D5.2.

9 Conclusions

D5.8 concludes the WP5 implementation-report series by bringing together the capabilities developed across the five pilot applications and their couplings. Since D5.7, the pilots have extended both the physical processes they represent and the software workflows used to prepare, execute and analyse simulations on HPC resources.

The developments address complementary application needs. UAP extends the modelling of urban airflow and air pollution, while UBM connects richer building representations with energy and indoor-environment simulation. RES integrates weather information and prediction methods for renewable electricity production. WF develops the connection between environmental data, fire simulation and visualisation. MTW extends particle-laden flow modelling to thermal interactions. Across these domains, the contribution of WP5 is the implementation of these capabilities in application software and workflows.

A common result is the closer connection between application data and numerical models. Improved preprocessing, configurable execution and result analysis support the use of the models in different study settings. Build and testing workflows, packaging and deployment arrangements provide the supporting software infrastructure. Validation examples in the pilot sections establish the scope of the implemented capabilities and guide their further development.

The coupling work adds interfaces through which applications can use complementary information from other models. File-based exchanges support sequential workflows, while direct solver interfaces support exchanges during a coupled calculation. The implementations reported here connect meteorological, thermal and environmental quantities across application domains. Their individual demonstrations and remaining integration requirements are documented in Section 8, providing a technical basis for further coupled applications.

Further development can build on these components by enriching input data, extending validation and completing the application-specific integration of coupled workflows. The implementation advances recorded in D5.8 provide the basis for this continuation and for reuse of the pilot software. D5.2 will draw on these results to assess the final synergies and benefits of integrated models, software and hardware.

References

[1] HiDALGO2 consortium, "HiDALGO2 Description of Action," Grant Agreement 101093457, Annex 1, Task 5.2 and Urban Buildings use case, 2022.

[2] L. Torres et al., "D5.7 Implementation Report on Pilot Applications," version 1.0, 2025, Sections 3, 7.1, 7.2 and 7.4. DOI: https://doi.org/10.13140/RG.2.2.21832.02567.

[3] C. Prud'homme et al., "UB - WP5: Technical progress slides," UNISTRA, 2026, slides 4-8, 10-11 and 13-14.

[4] UNISTRA / Cemosis, "Ktirio Urban Building documentation: scenario capabilities, IAQ, ensemble execution and visualisation," 2026. [Online]. Available: https://kub.ktirio.fr/.

[5] UNISTRA / Cemosis, "Ktirio BESTEST corrected six-case matrix and numerical stability audit," 30-31 August 2026. Internal records: annual-acceptance.json, evidence.json and modeler-notes.adoc.

[6] UNISTRA / Cemosis, "Ktirio Urban Building documentation: installation, execution and deployment," 2026. [Online]. Available: https://kub.ktirio.fr/.

[7] ASHRAE, "ANSI/ASHRAE Standard 140-2023: Method of Test for Evaluating Building Performance Simulation Software," 2023.

[8] M. Bauer et al., "waLBerla: A block-structured high-performance framework for multiphysics simulations," Computers & Mathematics with Applications, vol. 81, pp. 478-501, 2021. DOI: https://doi.org/10.1016/j.camwa.2020.01.007.

[9] M. Bauer, H. Köstler and U. Rüde, "lbmpy: Automatic code generation for efficient parallel lattice Boltzmann methods," Journal of Computational Science, vol. 49, article 101269, 2021. DOI: https://doi.org/10.1016/j.jocs.2020.101269.

[10] R. K. Ayyala Somayajula, H. Köstler and A. Jonnalagadda, "Computational investigation of total-enthalpy based conjugate heat transfer in particle-laden flows using lattice Boltzmann method," International Journal of Heat and Mass Transfer, vol. 271, article 129308, 2026. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2026.129308.

[11] Y. Hu, D. Li, X. Niu and S. Shu, "Fully resolved simulation of particulate flows with heat transfer by smoothed profile-lattice Boltzmann method," International Journal of Heat and Mass Transfer, vol. 126, pp. 1164-1167, 2018. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2018.05.137.

[12] S. Kemmler, C. Rettinger, U. Rüde, P. Cuéllar and H. Köstler, "Efficiency and scalability of fully-resolved fluid-particle simulations on heterogeneous CPU-GPU architectures," The International Journal of High Performance Computing Applications, vol. 39, no. 3, pp. 345-363, 2025. DOI: https://doi.org/10.1177/10943420241313385.

[13] R. K. Ayyala Somayajula, "Computational investigation of total-enthalpy based conjugate heat transfer in particle-laden flows using lattice Boltzmann method," dataset and source code, Zenodo, 2026. DOI: https://doi.org/10.5281/zenodo.18185501.

[14] S. Eibl and U. Rüde, "A Modular and Extensible Software Architecture for Particle Dynamics," Proceedings of the 8th International Conference on Discrete Element Methods; arXiv:1906.10963, 2019. DOI: https://doi.org/10.48550/arXiv.1906.10963.

[15] preCICE developers, "preCICE coupling library and OpenFOAM adapter documentation," 2026. [Online]. Available: https://precice.org/adapter-openfoam-overview.

[16] EuroCC / CASTIEL, "High-resolution urban wind comfort computation for the city of Stockholm (NCC/CoE Booklet 2025 success story)", 2025. [Online]. Available: https://hpc-portal.eu/materials/publications/high-resolution-urban-wind-comfort-computation-the-city-stockholm. Accessed: 15 October 2026.

[17] F. Martín et al., "Using dispersion models at microscale to assess long-term air pollution in urban hot spots: A FAIRMODE joint intercomparison exercise for a case study in Antwerp," Science of the Total Environment, vol. 925, article 171761, 2024. DOI: https://doi.org/10.1016/j.scitotenv.2024.171761.

[18] NSF NCAR / UCAR, "Weather Research and Forecasting Model: official model overview", n.d. [Online]. Available: https://www.mmm.ucar.edu/models/wrf. Accessed: 15 October 2026.

[19] OpenWFM developers, "WRF-SFIRE: coupled weather-fire model, source and documentation", n.d. [Online]. Available: https://github.com/openwfm/WRF-SFIRE. Accessed: 15 October 2026.

[20] NIST, "Fire Dynamics Simulator and Smokeview: official documentation", n.d. [Online]. Available: https://pages.nist.gov/fds-smv/. Accessed: 15 October 2026.

[21] Academy Software Foundation, "OpenVDB: sparse volumetric data library and documentation", n.d. [Online]. Available: https://www.openvdb.org/. Accessed: 15 October 2026.

[22] NCAR, "EULAG model homepage and documentation", n.d. [Online]. Available: https://www2.mmm.ucar.edu/eulag/. Accessed: 15 October 2026.

[23] Epic Games, "Sparse Volume Textures: Unreal Engine documentation", n.d. [Online]. Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/sparse-volume-textures-in-unreal-engine. Accessed: 15 October 2026.

[24] UNISTRA and SZE, "UAP-UBM coupling: Progress, status and roadmap," presentation, 2026. [Online]. Available: https://projekty.drive.man.poznan.pl/s/ZebnRk7GrbxT85i.

[25] C. Prud'homme et al., "D5.1 Achievements, Synergies, and Coupled Simulations," version 1.0, 2024. DOI: https://doi.org/10.13140/RG.2.2.16715.40480.

[26] L. Torres et al., "D5.3 Research Advancements for the Pilots," version 1.0, 2023. DOI: https://doi.org/10.13140/RG.2.2.19390.46400.

[27] H. Köstler et al., "D5.4 Research Advancements for the Pilots," version 1.0, 2024. DOI: https://doi.org/10.13140/RG.2.2.23080.79368.

[28] H. Köstler et al., "D5.5 Research Advancements of Pilots," version 1.0, 2025. DOI: https://doi.org/10.13140/RG.2.2.26784.85763.

[29] H. Köstler et al., "D5.6 Implementation Report on Pilot Applications," version 1.0, 2024. DOI: https://doi.org/10.13140/RG.2.2.12011.55849.

[30] L. Környei et al., "D2.9 Dashboard and Services," version 1.0, 2026. DOI: https://doi.org/10.13140/RG.2.2.12141.14561.

[31] N. Chalvantzis et al., "D4.5 Advances in HPDA and AI for Global Challenges," version 1.0, 2026. DOI: https://doi.org/10.13140/RG.2.2.31739.07204.