D4.7 Visualizations for Global Challenges

HIDALGO2
CENTRE OF EXCELLENCE

Date: May 20, 2025

EuroHPC
Joint Undertaking

Document Identification			
Status	Final	Due Date	30/04/2025
Version	1.0	Submission Date	20/05/2025

Related WP	WP4	Document Reference	D4.7
Related Deliverable(s)	D2.1, D2.2, D4.6, D4.8	Dissemination Level (*)	PU
Lead Participant	UNISTRA	Lead Author	Christophe Prud'homme (UNISTRA)
Contributors	SZE, PSNC, USTUTT, UNISTRA, FAU	Reviewers	Bartosz Bosak (PSNC), Dennis Hoppe (USTUTT)

Keywords:
visualization, visual data analysis, CFDR, VR

Document information

List of Contributors	
Name	**Partner**
David Caballero	MTG
Luis Torres	MTG
Uwe Wössner	USTUTT
Mátyás Constans	SZE
László Környei	SZE
Christophe Prud'homme	UNISTRA
Vincent Chabannes	UNISTRA
Gwenolle Chappron	UNISTRA
Ravi Kiran Ayyala	FAU
Michał Kulczewski	PSNC

Document History			
Version	**Date**	**Change editors**	**Changes**
0.1	12/03/2025	Christophe Prud'homme(UNISTRA)	First draft of the document
0.15	12/03.2025	Davide Padeletti (USTUTT), Christophe Prud'homme(UNISTRA)	Minor fixes in sections 3.2.2 and 3.2.3, updated ToC
0.2	13/03/2025	Marcin Lawenda (PSNC) Harald Köstler (FAU)	ToC, timeline and responsibilities approved
0.3	17/04/2025	All Contributors	Deliverable content
0.4	17/04/2025	Christophe Prud'homme (UNISTRA)	Version ready for internal review
0.5	30/04/2025	Christophe Prud'homme(UNISTRA) David Caballero(MTG) Mátyás Constans(SZE)	Apply changes requested by reviewers

0.6	7/05/2025	Christophe Prud'homme(UNISTRA)	Version ready for quality check. Still fixes and check to be done by the partners
0.9	13/05/2025	Christophe Prud'homme(UNISTRA) László Környei (SZE) Matyas Constans	Added corrections and fixes for SZE contributions.
0.95	13/05/2025	Rahil Doshi	Quality assurance check
1.0	20/05/2025	Marcin Lawenda	Final check and improvements

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Christophe Prud'homme (UNISTRA)	13/05/2025
Quality manager	Rahil Doshi (FAU)	13/05/2025
Project Coordinator	Marcin Lawenda (PSNC)	20/05/2025

Table of contents

Document information ... 3
Table of contents ... 5
List of tables ... 8
List of figures ... 8
List of acronyms ... 9
Executive summary ... 11
1 Introduction ... 13
1.1 Purpose of the document ... 13
1.2 Relation to other project work ... 13
1.3 Structure of the document ... 14
2 Updated visualization requirements ... 15
2.1 Urban Air Project (SZE) ... 15
2.1.1 Previous requirements ... 15
2.1.2 Updated requirements ... 16
2.1.3 Rationale ... 16
2.2 Urban Building Model (UNISTRA) ... 16
2.2.1 Previous requirements ... 16
2.2.2 Updated requirements ... 17
2.2.3 Rationale ... 17
2.3 Renewable Energy Sources (PSNC) ... 17
2.3.1 Previous requirements ... 17
2.3.2 Updated requirements ... 17
2.3.3 Rationale ... 18
2.4 Wildfires (MTG) ... 18
2.4.1 Previous requirements ... 18
2.4.2 Updated requirements ... 18
2.4.3 Rationale ... 18
2.5 Material in Water (FAU) ... 19
2.5.1 Previous requirements ... 19

2.5.2 Updated requirements 19
2.5.3. Rationale 19
2.6 Summary of requirements 19
3 Progress in visual analysis tool development 23
3.1 Computational Fluid Dynamics Rendering (SZE) 23
3.1.1 Performance optimizations 23
3.1.2 Real-time streaming enhancements 24
3.1.3 Feature updates 25
3.2 Unreal Engine Advanced Visualization (MTG) 26
3.2.1 Feature updates 26
3.2.2 Development of enhanced smoke and flames volumetric materials 27
3.2.3 Integration with CESIUM service 28
3.2.4 Gaussian splatting integration 28
3.2.5 Visualization of vegetation structures 29
3.3 VISTLE-COVISE (USTUTT) 30
3.3.1 Performance optimizations 30
3.3.2 Real-time streaming enhancements 31
3.3.3 Feature updates 31
3.4 KTIRIO-GUI (UNISTRA) 31
3.4.1 Performance optimizations 31
3.4.2 Real-time streaming enhancements 32
3.4.3 Feature updates 32
3.4.4 User interface updates 32
4 Pilot visualization application 33
4.1 Urban Air Project (SZE) 33
4.1.1 Pilot objectives 33
4.1.2 Pilot-specific adaptations 33
4.1.3 Expected benefits 34
4.1.4 Case studies and use examples 34
4.1.5 Preliminary results 36
4.1.6 Lessons learned 37

4.2 Urban Building Model (UNISTRA) 38
4.2.1 Pilot objectives 38
4.2.2 Pilot-specific adaptations 38
4.2.3 Expected benefits 39
4.2.4 Case studies and use examples 40
4.2.5 Preliminary results 43
4.2.6 Lessons learned 44
4.3 Renewable Energy Sources (PSNC) 44
4.3.1 Pilot objectives 44
4.3.2 Pilot-specific adaptations 45
4.3.3 Expected benefits 45
4.3.4 Case studies and use examples 45
4.3.5 Preliminary results 48
4.3.6 Lessons learned 48
4.4 Wildfires (MTG) 48
4.4.1 Pilot objectives 48
4.4.2 Pilot-specific adaptations 48
4.4.3 Expected benefits 49
4.4.4 Case studies and use examples 49
4.5 Material Transport in Water (FAU) 51
4.5.1 Pilot objectives 51
4.5.2 Expected benefits 51
4.5.3 Case studies and use examples 51
4.5.4 Preliminary results 52
4.5.5 Lessons learned 53
5 Future directions 54
5.1 Computational Fluid Dynamics Rendering (SZE) 54
5.2 Unreal engine advanced visualization (MTG) 54
5.3 VISTLE-COVISE (USTUTT) 56
5.4 KTIRIO-GUI (UNISTRA) 57
6 Conclusions 59

References 60

List of tables

Table 1. New visualization requirements *21*
Table 2. R Values for different building components *40*
Table 3. Occupancy patterns *40*

List of figures

Figure 1. CFDR client – Realtime Slice Data streaming. Visualization on a RES simulation data provided in NetCDF format by the PSNC team for the city of Poznan. *26*
Figure 2. Effect of smoke shadow projection on the terrain, in this case provided by the CESIUM service. The shadow density is proportional to the attenuation of light as it passes through the volume defined by the smoke column. The effects of Rayleigh scattering. *27*
Figure 3. Example of the integration of a flame front and smoke column simulation onto the geometry provided by CESIUM from the Google Tile Service. *28*
Figure 4. Example of the integration of a Gaussian Splatting Training of a property with a house in a WUI area, showing the local combustion of a hedge along one side of the fence. The advantage of GS is the precision and realism in scene capture. *29*
Figure 5. Integration of procedurally generated multi-layered vegetation structure in Unreal Engine with a heterogeneous volume simulating flames and smoke, imported as a VDB file. *30*
Figure 6. Stockholm LAWSON Wind Comfort *36*
Figure 7. Poznań Wind Field (data provided by PSNC) *37*
Figure 8. ParaView screenshot of a simulation of Paris with a radius of 5km (102506 buildings) displaying power output for 01/01/2023. .... *42*
Figure 9. Screenshot of the first section of ktirio.cases website with screenshots of ktirioGUI application. and table of content on the top right of the screen *43*
Figure 10. Screenshot of the averaged power output across the city automatically generated on ktirio.cases *43*
Figure 11. RES temperature forecast *46*
Figure 12. RES wind speed forecast *47*
Figure 13. RES prediction of damages to the overhead network *47*
Figure 14. Example of integrating WRF-SFIRE simulations into Unreal Engine 5 using Google Tile Service geometry via CESIUM for the 2019 Cadalso de los Vidrios wildfire. *50*
Figure 15. Figure 16. The same simulation, but viewed closer to the ground. The orange tint caused by Mie scattering due to the presence of particles in the atmosphere is clearly visible. *51*
Figure 16. Converged steady state visualization of 2D differentially heated square cavity *52*
Figure 17. Converged steady state visualization of 2D Rayleigh Bernard Convection *53*

List of acronyms

Abbreviation / acronym	Description
ASCII	American Standard Code for Information Interchange
APK	Android Application Package
AR	Augmented Reality
BP	Blueprint
EC	European Commission
CAVE	Cave-Assisted Virtual Environment
CEEC	Center of Excellence for Exascale CFD
CFD	Computer Fluid Dynamics
CFDR	Computer Fluid Dynamics Rendering
COVISE	Collaborative Visualization and Simulation Environment
CPU	Central Processing Unit
DTM	Digital Terrain Model
Dx.y	Deliverable number y belonging to WP x
ESRI	Environmental Systems Research Institute
FBX	FilmBoX
FGA	Fluid Grid ASCII
FOAM	Field Operation And Manipulation
fps	Frames per second (Hz)
GIS	Geographical Information System
GPU	Graphics Processing Unit
GS	Gaussian Splatting
GUI	Graphical User Interface
HLRS	High-Performance Computing Center Stuttgart (**H**öchst**L**eistungs**R**echenzentrum **S**tuttgart)
HPC	High Performance Computing
IO	Input-Output
LUMI	Large Unified Modern Infrastructure

MTW	Material Transport in Water
NeRF	Neural Radiance Field
RES	Renewable Energy Sources
SHP	Shapefile
SVT	Sparse Volumetric Texture
UAP	Urban Air Project
UBM	Urban Building Model
UE	Unreal Engine
UEAV	Unreal Engine Advanced Visualization
VDB	Volume Dynamic B+Tree
VR	Virtual Reality
VTK	Visualization Toolkit
WF	Wildfires
WP	Work Package
WRF	Weather Research and Forecasting
WUI	Wildland-Urban Interface
XR	Extended Reality

Executive summary

This document presents the latest developments in visualization strategies tailored for addressing global environmental challenges using high-performance computing. Building upon the foundational requirements outlined in D2.1, D2.2 and the initial work completed in D4.6, the current deliverable, D4.7, offers an in-depth overview of how our visualization tools and methodologies are evolving to meet the increasingly complex needs of our pilot studies.

Key highlights of this document include:

- **Updated Visualization Requirements:** The document begins with a comprehensive review of the original visualization requirements and details the new or modified requirements that have emerged based on pilot feedback and evolving project needs. Each pilot—ranging from urban air quality and building performance to renewable energy forecasting, wildfire simulation, and material transport in water—has prompted tailored adjustments. These updates ensure that our visualization strategies remain relevant and effective in supporting detailed analysis and decision-making.
- **Progress in Visual Analysis Tool Development:** Significant improvements have been made to our core visualization tools, including CFDR (Computer Fluid Dynamics Rendering), VISTLE-COVISE, UEAV (Unreal Engine Advanced Visualization), and KTIRIO-GUI. Enhancements in data compression, real-time streaming, interactive controls, and user interface design have contributed to a more robust and scalable visualization environment. These advances enable the efficient processing of large simulation datasets and provide users with more intuitive and immersive experiences.
- **Application to Pilot Studies:** The document details how advanced visualization strategies have been implemented across various pilot studies. It describes the specific adaptations and customizations made for each pilot, outlining the complete workflow from the extraction and processing of simulation data to its interactive representation. These case studies demonstrate practical benefits such as improved accuracy in urban air quality monitoring, enhanced analysis of building performance, more comprehensive visualization of weather phenomena affecting renewable energy outputs, and more realistic representations of wildfire dynamics and material transport in water.
- **Future Directions and Work Plan:** A forward-looking roadmap is provided that outlines planned developments, research opportunities, and integration strategies for further enhancing the visualization tools. Emphasis is placed on continuous improvement through collaborative feedback, exploration of emerging technologies, and a cohesive approach to integrating our visualization

platforms. This roadmap defines key milestones and collaborative efforts that will guide future work and ensure the framework remains adaptive and scalable.

- **Conclusions:** In summary, D4.7 represents a significant step forward in advancing our visualization strategies. By integrating enhanced data handling capabilities, improved interactivity, and robust cross-tool integration, the project is well-positioned to deliver impactful insights and decision-support tools for various environmental challenges. The progress documented here not only reflects our current achievements but also sets a clear path for future enhancements, ensuring that our visualization framework remains at the forefront of technological innovation in environmental simulation analysis.

The insights and advancements presented in this deliverable are expected to have a lasting impact on our ability to analyse complex simulation data, ultimately contributing to more informed and effective decision-making processes in addressing global challenges.

1 Introduction

This document presents the state-of-the-art in visualization strategies for global challenges, with an emphasis on enhancing visual data analysis in high-performance computing environments. The Introduction sets the context for the deliverable by explaining its objectives, its relation to other project work, and the overall structure of the document.

1.1 Purpose of the document

The purpose of this document is to detail the latest advances in visualization tools and strategies developed to support the analysis of simulation data for environmental challenges. It aims to:

- Explain how the initial requirements defined in D2.1 and D2.2 have evolved and how these updates have been integrated into the current visualization framework.
- Present the progress made since D4.6 in enhancing existing visualization tools, including improvements in performance, interactivity, and scalability.
- Demonstrate the application of these advanced visualization strategies across various pilot studies.
- Outline future directions and a work plan to ensure that the visualization framework remains robust and adaptable to emerging challenges.

This document is intended to serve as both a technical reference and a roadmap for future developments, ensuring that our visualization approaches continue to evolve in line with the project's objectives and stakeholder needs.

1.2 Relation to other project work

Deliverable D4.7 builds directly on the requirements and proof-of-concept implementations defined in D2.1, D2.2 and D4.6, refining and extending those foundations to address pilot-specific needs. In WP2 we established the generic data-exchange formats and end-user use cases; those have been re-visited here in light of the first pilot feedback loops and evolving simulation outputs from WRF, OpenFOAM, EULAG and other solvers.

Within WP4, D4.7 serves as the capstone to our tool-development track, integrating the CFDR, VISTLE-COVISE, UEAV and KTIRIO-GUI platforms into a coherent framework. It also lays the groundwork for the cross-pilot dashboard work in WP5 (see D5.3, D5.7) by defining how live and archived visualizations will feed into web portals. Finally, the workflows and formats specified here will support the exploitation and dissemination activities of WP6, ensuring that the visualization assets produced can be readily packaged for training, outreach, and future digital-twin deployments.

1.3 Structure of the document

This document is organized into several key sections, each designed to provide a comprehensive view of the project's visualization strategies and progress. The structure is as follows:

Chapter 1, Introduction, explains the overall purpose of the document, outlines the main goals and intended impact of the deliverable, and describes how the document relates to other project work. It concludes with a detailed explanation of how the document is organized to guide the reader through the content.

Chapter 2, Updated Visualization Requirements, reviews the original visualization requirements defined in D2.2, explains how these requirements were addressed in D4.6, and then presents the new or modified requirements that have emerged based on pilot feedback and evolving project needs.

Chapter 3, Progress in Visual Analysis Tool Development, provides an overview of the current visualization tools, describes recent enhancements and innovations since D4.6, and discusses the challenges encountered during development along with the solutions that have been implemented.

Chapter 4, Application to Pilot Studies, details how the visualization strategies have been tailored to meet the specific needs of each pilot. It includes descriptions of pilot-specific adaptations, customizations of the visualization tools, and practical case studies that illustrate the complete workflow from simulation data to interactive visualization.

Chapter 5, Future Directions and Work Plan, outlines the planned developments, research opportunities, and integration strategies for further enhancing the visualization tools. It presents a roadmap with key milestones and describes collaborative efforts that will support future work.

Chapter 6, Conclusions. summarizes the main outcomes of the current work, discusses the expected impact on the pilot applications, and outlines the next steps for ongoing development.

2 Updated visualization requirements

This section revisits the visualization requirements initially outlined in D2.1 and D2.2 and further addressed in D4.6, then details the new or modified requirements that have emerged. In particular, each pilot's evolving needs have prompted tailored adjustments to the visualization strategies. The following subsections explain the updated requirements for each pilot.

This section summarizes the visualization requirements as defined in D2.1 and refined in D2.2 and reviews how they were addressed in D4.6. The focus was on providing web-based 3D visualization of CFD data, interactive time-series handling, geo-temporal data representation, and support for various exchange file formats (e.g., NetCDF, CSV, Ensight Gold). D4.6 implemented these requirements by demonstrating proof-of-concept solutions such as the CFDR for CFD simulation data packaging and web rendering, VISTLE-COVISE for immersive visualization, UEAV for photorealistic VR experiences, and KTIRIO-GUI for urban building visualizations. This baseline work laid the foundation for improved performance, scalability, and the integration of additional requirements.

Based on pilot feedback and evolving project needs, new requirements and modifications have emerged. The following subsections outline these changes for each pilot, detailing what was required in D2.1, D2.2, D4.6 and what updates are now needed.

2.1 Urban Air Project (SZE)

2.1.1 Previous requirements

- Supported 1D, 2D, and 3D CFD data of urban air pollution with standard exchange formats.
- Visualization focused on scalar fields (e.g., pollutant concentrations) and vector fields (e.g., wind velocities) through simple interactive visualizations and web-based tools.
- Slice visualization corresponding to a given plane in 3D space with a normal (nx, ny, nz) and an offset.
- Alpha-blending-based volumetric visualization.

2.1.2 Updated requirements

- Fast, optimized slice visualization with interpolation on an arbitrary surface (terrain surface, with different elevation offsets, or other custom 2D geometries and their translations) mesh from a CFD mesh. Custom implementation in C/C++.
- Volumetric visualization based on raytracing. Streamline rendering. Custom implementation in C/C++.
- Full support for EnSight Gold format. EnSight Gold is vital in HPC because its "case + part" structure lets hundreds or thousands of MPI ranks write and read binary part files in parallel, avoiding serial I/O bottlenecks; its compact binary storage and timestep streaming minimize file size, I/O time, and memory footprint; and its rich metadata support and widespread tool compatibility ensure efficient, scalable post-processing and visualization of multi-terabyte, multi-variable simulation data.

2.1.3 Rationale

- Arbitrary surfaces for slices are very important in the context of urban air pollution, since the terrain is almost never flat, and has a lot of height variation (i.e., in an application to Stockholm).
- Alpha-blending volumetric visualization (additive alpha blending) is detached from reality, and while it may be suitable for basic visualizations, ray tracing gives much better results, both visually and numerically.
- Full support for EnSight Gold, both for 2D and 3D elements, was necessary to visualize data from UNISTRA, SZE, and other future projects. EnSight Gold was chosen as our core data format because it combines advanced visualization features with high-performance I/O—and it plugs straight into ParaView

2.2 Urban Building Model (UNISTRA)

2.2.1 Previous requirements

- Focused on generating scientific and information visualizations of building meshes, energy performance metrics, and environmental factors (e.g., solar shading, ambient temperature).

- Utilized static data displays integrated with 3D urban models using KTIRIO-GUI and ParaView.

2.2.2 Updated requirements

- **Advanced data overlay:** Enable overlay of statistical data (e.g., comfort indices, CO_2 levels) directly onto the 3D urban model with interactive filtering options.
- **Enhanced export formats:** Update data exchange protocols to support higher resolution datasets and improved interoperability with BIM and GIS systems.

2.2.3 Rationale

These updates aim to provide a more comprehensive tool for building performance analysis by combining real-time data processing with advanced visualization techniques that support both technical and managerial decision-making.

2.3 Renewable Energy Sources (PSNC)

2.3.1 Previous requirements

- Required visualization of unsteady surface and volumetric weather fields (wind, temperature, pressure) from forecasting tools like WRF and EULAG.
- Focused on static image generation and basic animations from cross-sectional data.

2.3.2 Updated requirements

- Include visualization in the workflow and integrate it with the portal/dashboard, possibly generating a link which can be shared with users and stakeholders.
- Visual comparison of forecasts between ensembles.
- Enable overlay of data, such as energy production forecasts, onto 3D models and geospatial data.

2.3.3 Rationale

The new requirements respond to the demand for using visualization for more in-depth data analysis, and to attract users and stakeholders.

2.4 Wildfires (MTG)

2.4.1 Previous requirements

- Aimed at creating realistic representations of fire fronts and smoke dispersion, using volumetric and vector field visualizations.
- Relied on basic CFD simulation outputs and standard exchange formats, introducing some immersive elements.

2.4.2 Updated requirements

- Integration of vegetation structures across multiple strata over large areas, enabling both a realistic 3D representation and the extraction of an estimate of the three-dimensional distribution of biomass and porosity.
- Development of a simple tool for vegetation description using procedural vegetation spawning. Alternatively, connection with external detailed models to simulate vegetation dispersal, establishment, and evolution.

2.4.3 Rationale

These requirements are considered relevant both for the implementation of the solution addressing wildfires in the WUI and for future developments, as well as for coupling with other pilots in future digital twins.

MeteoGrid is currently using the Procedural Vegetation Spawning tools integrated in Unreal Engine 5.0 and has already completed a workflow for extracting point clouds and volumetric distributions of biomass and porosity to be used in OpenFOAM solvers. In the coming months, a simple tool will be developed to enable these functionalities within the wildfire simulation workflow, with the potential to be integrated into the dashboard. MeteoGrid has also already carried out integrations via exchange files (XML) with the SORTIE ND application for simulating the evolution of vegetation populations, and the integration of this type of model into the wildfire simulation workflow is also planned.

2.5 Material in Water (FAU)

2.5.1 Previous requirements

- Emphasized visualization of CFD data, including scalar fields (e.g., temperature, concentration) and vector fields (e.g., fluid velocity), along with particle tracking.
- Utilized basic visualization pipelines in CFDR and ParaView for analysis.

2.5.2 Updated requirements

We haven't used anything from CFDR yet. FAU has always used primarily ParaView for our visualization purposes. Our research group has previously collaborated with VISTLE-COVISE in a cave and HLRS in another project, and for the current use case in HiDALGO2, we plan to do the same.

2.5.3. Rationale

The updated requirements for MTW are motivated by the VISTLE-COVISE is prioritized as one of our visualization tools because of the already existing collaboration our research group has with HLRS, Stuttgart.

2.6 Summary of requirements

The table below summarizes the new requirements presented in the previous sections by UNISTRA, PSNC, and MTG. The table follows the same nomenclature for requirements as in D2.2. Here are a few explanations about some columns

 # ID

A unique requirement identifier in the form

`REQ-<CategoryCode>-<SequentialNumber>`

“REQ” denotes “requirement,” followed by a category code (e.g. POR) and its sequence number.

Type

Specifies whether the requirement is: **F** = Functional (a behaviour or feature the system must provide) and **NF** = Non-functional (performance, usability, security, etc.).

Prio

The priority level of the requirement: **H** = High; **M** = Medium; **L** = Low.

Risk

An estimate of implementation difficulty or uncertainty: **H** = High risk; **M** = Medium risk; **L** = Low risk.

Table 1. New visualization requirements

#ID	Title	Description	Type	Prio	Risk	Required by	Responsible	Dependencies	
REQ-POR-073	Advanced Data Overlay	Using interactive filtering options, enable the overlay of statistical data (e.g., comfort indices, CO_2 levels) directly onto the 3D urban model.	F	L	M	UNISTRA	UNISTRA		Planned R03.2026
REQ-POR-074	Enhanced Export Formats	Update data exchange protocols to support higher resolution datasets and improved interoperability with BIM and GIS systems.	F	L	M	UNISTRA	UNISTRA		Planned R03.2026
REQ-POR-075	Visualization as a part of the workflow	Include visualization of RES results in the portal/dashboard	F	M	L	PSNC	PSNC, SZE		Planned R12.2025
REQ-POR-076	Visual comparison of ensembles	Enable comparison of ensemble results as a visualization	F	L	M	PSNC	PSNC		Planned R06.2026

#ID	Title	Description	Type	Prio	Risk	Required by	Responsible	Dependencies	
REQ-POR-077	Energy forecast data overlay	Enable overlay of forecast data, e.g., energy production, onto 3D model and geospatial data	F	M	M	PSNC	PSNC		Planned R03.2026
REQ-POR-078	Estimate biomass and porosity	Integration of vegetation structures across multiple strata over large areas, enabling both realistic 3D representation and the extraction of an estimate of the three-dimensional distribution of biomass and porosity.	F	H	M	MTG	MTG		Planned Q04 2025
REQ-POR-079	Simulate vegetation evolution	Development of a simple tool for vegetation description using procedural vegetation spawning. Alternatively, connection with external detailed models to simulate vegetation dispersal, establishment, and evolution.	F	M	H	MTG	MTG		Planned Q04 2025

3 Progress in visual analysis tool development

This section presents an overview of the current state and recent improvements in the visualization tools developed for the project. It highlights the key functionalities, innovations, and challenges that have been addressed since D4.6, setting the stage for the integration of advanced visualization strategies across pilot applications.

This section provides a summary of the main visualization tools currently in use:

- **CFDR (Computer Fluid Dynamics Rendering):**
  A tool that packages and compresses CFD simulation data on HPC or any computational platforms for efficient web-based visualization. It supports various exchange formats and enables real-time interactive data exploration. Data can be streamed from any EuroHPC machine (LUMI, Karolina, ...) via SSH. CFDR allows for real-time visualization of existing simulation data, in the form of slices, volumetric rendering, and streamlines. It supports most industrial formats: EnSight Gold, NetCDF, VTK, and VTM.
- **VISTLE-COVISE:**
  An immersive visualization platform designed for collective data analysis in CAVE environments, it enables collaborative exploration of large datasets and facilitates interactive adjustments in the visualization pipeline.
- **UEAV (Unreal Engine Advanced Visualization):**
  A platform that compiles photorealistic VR experiences integrating simulation data with geographic and contextual elements. It leverages Unreal Engine's capabilities for high-fidelity visual effects and immersive environments.
- **KTIRIO-GUI:**
  A graphical user interface developed in Qt (C++) that streamlines the preparation of urban building datasets, enabling data partitioning, visualization, and integration with HPC simulation workflows.

3.1 Computational Fluid Dynamics Rendering (SZE)

3.1.1 Performance optimizations

In the reporting period, two major areas of CFDR have been optimised: the EnSight Gold Case loader system and the custom slice generator, which were written to replace ParaView components. This work has resulted in the current version of CFDR not containing any components from ParaView or other visualization software.

EnSight Gold File Loader

To optimize the performance of the **EnSight Gold Case** mesh reader, the following strategies were applied:

- mmap for File I/O: Directly maps the mesh file into memory, reducing system call overhead.
- Custom String Library: Provides a view on the mapped memory (using a char* and length) to minimize allocations and streamline string handling.
- Designed with modern C practices and supports custom memory allocators for flexible, efficient memory management.

SIMD-Accelerated Slicer

The slicer hinges on the `get_values_at_mesh_encas(...)` function, that computes interpolated values at query points (from an .obj surface file) based on the EnSight Case mesh data. The following optimizations were implemented:

- Spatial Hash Bucketing: Spatial partitioning of mesh elements drastically reduces the number of distance checks per query point.
- OpenMP Multithreading: Each query point is evaluated in parallel using dynamic scheduling for load balancing.
- SIMD Abstraction Layer: A cross-platform abstraction was developed to exploit maximum vector width available on target hardware:
    - x86: SSE4.1 (128-bit), AVX2 (256-bit), AVX-512 (512-bit),
    - ARM: NEON (128-bit).

Primary benchmarks measurements were conducted, resulting in similar performance between ParaView and CFDR. Full slicing postprocessing was analysed for a 2.1M mesh and several time steps in EnSight Gold format using one full node of LUMI. While ParaView completed the task in 63 seconds, CFDR completed the same task within 74 seconds. These results, however, need to be further analysed as it is not yet clear how each tool utilizes the 128 available threads. This analysis is scheduled for the next two months.

3.1.2 Real-time streaming enhancements

CFDR currently has two modes: an **uncompressed** streaming mode, in which all slice data is transmitted as 32-bit floating points, and a **compressed mode**.

For the compressed mode, we first find the minimum and maximum of our dataset and send them to the client as metadata. For the compression of the actual data, we first normalize our dataset between 0 and 1 (apply x-> ((x – min) / (max – min)), then convert the data to 16-bit floating points (half-floats) (IEEE 754-2008 standard).

An array of 16-bit floats will be reinterpreted as 16-bit integers and compressed with the LZ4 algorithm. The LZ4 algorithm is heavily parallelized, thanks to the AVX512 extensions on newer chips, which enables compression in a time frame significantly less than the other parts of the workflow, making it unnoticeable.

3.1.3 Feature updates

CFDR is entirely independent from ParaView; every feature has been implemented in C/C++, including simulation file loads and CFD mesh slices.

- Full **EnSight Gold Case** and NetCDF support were major feature updates to CFDR. Supporting these two formats allows us now to cover most of the Pilot requirements, and most CFD application data as well. Full EnSight Gold feature support was challenging, since the format is quite huge and can describe a lot of different cell topologies (2D triangles, 2D n-gons, 3D tetrahedra, 3D n-gons, ...).
- Several performance optimizations were applied to the code of the reader, detailed in the section Performance Optimizations, providing read times that are roughly < 1 second for even 500k cell meshes.
- A “slicer” functionality in the CFDR pre-processing tool, that generates the data to be visualized from the input CFD mesh (EnSight Gold, NetCDF, VTK, ...) in a very fast way (see Performance Optimizations for benchmarks), given an input OBJ or STL triangle surface mesh.

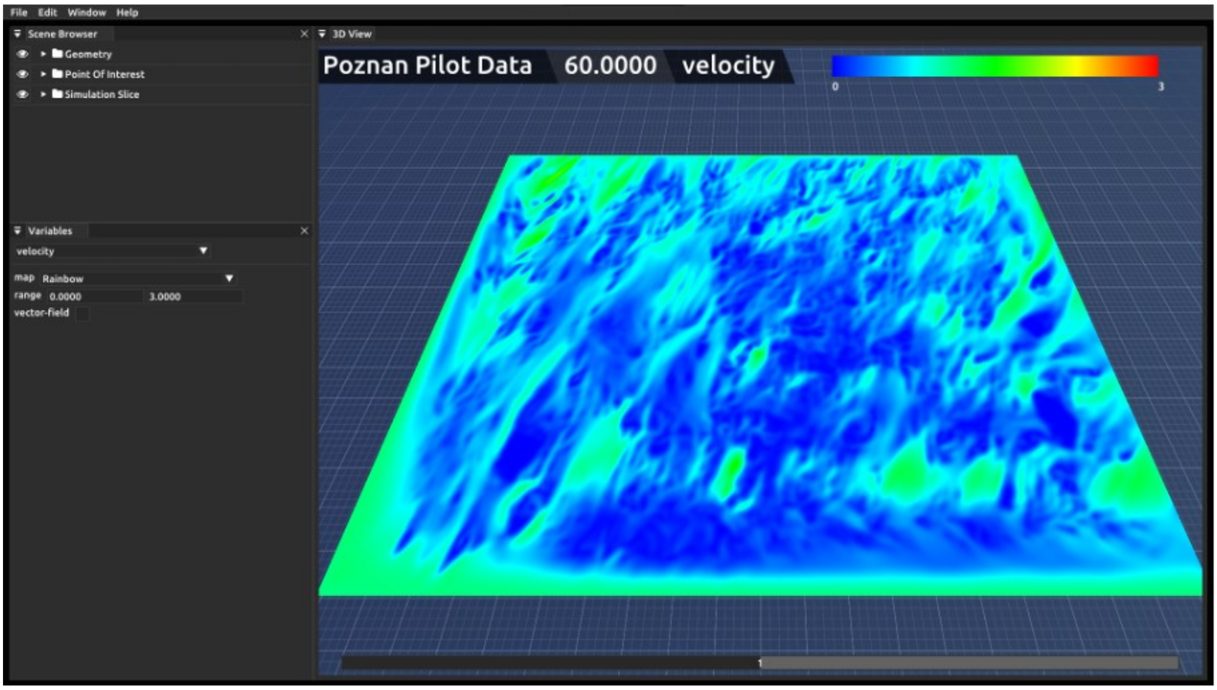

Figure 1. CFDR client – Realtime Slice Data streaming. Visualization on a RES simulation data provided in NetCDF format by the PSNC team for the city of Poznan.

A custom streamline and point cloud pre-processor/rendering engine is being worked on.

3.2 Unreal Engine Advanced Visualization (MTG)

3.2.1 Feature updates

The MeteoGrid team has refined a two-scale visualization workflow-landscape (≈10 000 km²) and wildland-urban interface (≈1 km²) - to display several hours of flame-front propagation and smoke in broad terrain, as well as detailed fire impacts on individual buildings. At the landscape level, WRF-SFIRE produces gridded NETCDF output on HPC systems, while at the WUI scale specialized OpenFOAM solvers generate reactive-porous-media results in ASCII or binary form. Both data streams are ingested into Unreal Engine 5.4, which was selected for its new support of heterogeneous volumetric (HV) rendering in VR, enabling realistic, layered smoke and flame effects that respond dynamically to terrain and structures. Deliverable D4.6 defined the integration guidelines, and we've developed lightweight parsers to translate both NETCDF and OpenFOAM outputs into UE's native volume formats without bloating the scene with empty cells. By leveraging UE5's GPU-accelerated volume shaders and custom shadow-projection routines, our demonstrators produce immersive, high-fidelity visualizations that firefighters have deemed remarkably lifelike during evaluation. This dual-scale approach ensures that strategic, large-area wildfire dynamics and localized WUI hazards can both be explored interactively in real time.

3.2.2 Development of enhanced smoke and flames volumetric materials

MeteoGrid converts WRF-SFIRE's 3D NETCDF output—where only the flame and smoke-filled cells carry density and temperature—into sparse VDB volumes via a lightweight C++ parser using OpenVDB. By interpolating WRF's sigma levels into terrain-relative heights, the workflow ensures accurate vertical placement in Unreal Engine, which now also renders dynamic smoke shadows. Firefighter feedback confirms these sparse-data VDB visualizations deliver highly realistic flame and smoke behaviour without filling the entire domain.

Figure 2. Effect of smoke shadow projection on the terrain, in this case provided by the CESIUM service. The shadow density is proportional to the attenuation of light as it passes through the volume defined by the smoke column. The effects of Rayleigh scattering.

MeteoGrid has also carried out the study and implementation of light attenuation and occlusion in surrounding smoke, based on the concentration levels obtained from the simulations and the presence of soot (black carbon). Since Unreal Engine uses its own attenuation functions, an assessment was made comparing them with the Bouguer-Beer-Lambert law, and adjustments were introduced to enhance realism in contextual perception scenarios at ground level, particularly at the urban scale.

3.2.3 Integration with CESIUM service

MeteoGrid has also studied and developed a workflow in Unreal Engine for integrating simulations with the CESIUM service, specifically the Google Tiling Service, which provides a high-resolution photogrammetric model of the world. To accurately position the simulations (heterogeneous volumes, HV) of smoke and fire within CESIUM's 3D geometry, it was necessary to define the CFD simulation domain dimensions (in cm), set the local origin position with altitude referenced to the geoid (Zo = Z + OG, where OG is the Geoid Undulation at the local origin), correct the displacement applied to the emitters, position the centre of the HV relative to its lower-left back corner (LLBC), and finally align the HV with the local coordinate system origin.

Figure 3. Example of the integration of a flame front and smoke column simulation onto the geometry provided by CESIUM from the Google Tile Service.

3.2.4 Gaussian splatting integration

MeteoGrid has worked in detail on the integration of Gaussian Splatting Trainings into Unreal Engine, generated from extensive sets of drone photographs. These Gaussian Splatting (GS) training files are stored in the standard PLY exchange format and are read in Unreal Engine through dedicated plug-ins. These GS environments capture all the details of a complex scene, providing the opportunity to present reality as it is and to overlay fire simulations onto it. The use of GS is particularly valuable for immersive experiences with property owners in WUI areas, allowing them to visualize and understand how a wildfire would unfold around their home and which factors are most critical to its survival.

Figure 4. Example of the integration of a Gaussian Splatting Training of a property with a house in a WUI area, showing the local combustion of a hedge along one side of the fence. The advantage of GS is the precision and realism in scene capture.

3.2.5 Visualization of vegetation structures

MeteoGrid’s workflow leverages Unreal Engine’s Procedural Vegetation Spawning to generate realistic 3D vegetation layers—extracting biomass distributions and porosity factors as point clouds for use in OpenFOAM solvers—enabling coherent fire–vegetation interaction simulations. A built-in Unreal evolution engine models individual plant growth, seed dispersal, germination, and light/space competition (with plans to adopt SORTIE-ND or iLandscape on HPC), all culminating in immersive urban-scale wildfire demonstrators that integrate detailed buildings, smoke, and flames.

Figure 5. Integration of procedurally generated multi-layered vegetation structure in Unreal Engine with a heterogeneous volume simulating flames and smoke, imported as a VDB file.

3.3 VISTLE-COVISE (USTUTT)

3.3.1 Performance optimizations

Since the last report, GPGPU support has been improved by tightly integrating VTK-m into VISTLE and by implementing some of the visualization modules on this foundation. The performance of interacting with the particle tracer has been improved by (i) creating fewer data objects and (ii) assigning all of the output data for a single timestep to a single MPI rank instead of scattering it across all nodes. Additionally, the pixel-accurate rendering of spheres in a shader, e.g., for showing the particles resulting from such a tracing operation, improved rendering performance compared to a polygonal approximation.

3.3.2 Real-time streaming enhancements

Setting up the connection between a user-facing local renderer to a remote renderer on a remote compute facility has been simplified by routing the data through an existing connection. This avoids the need to take firewalls and network address translation into account. Also, it is now possible to stream data between instances of the visualization system compiled with different configurations: now it is possible to communicate between systems disagreeing on the size of indices (32 bit vs. 64 bit) or the precision of scalars (float vs double).

3.3.3 Feature updates

Creation of a digital twin of the whole of Germany, Switzerland, parts of Austria and Estonia. Tiled databases were created for the majority of the federal states of Germany to enable high-performance rendering of environments of wind turbines. In addition to the terrain data and LOD2 buildings, high-voltage power lines and wind turbines were also converted to tiled databases where available. Switzerland was completely implemented, in Austria, so far only Vienna and the city of Salzburg are available, while the entire province of Salzburg has been requested and will be prepared soon. Dynamic loading of tiled databases. Until now, environment data for our digital twins has been created manually. Now a new plugin has been developed that makes it possible to use the same basic data for different projects. Sections for simulation data are cut out dynamically during the loading process. This avoids so-called Z-fighting and reduces manual modelling work. The plugin also displays 360° sky spheres depending on the location. Also, additional file format compatibility has been improved, enabling more versatile workflows. EnSight Gold data (both ASCII and binary) can now be imported. Also, urban planning scenarios involving flood simulations have been enabled by integrating with an OpenFOAM based workflow employing shallow water models.

3.4 KTIRIO-GUI (UNISTRA)

3.4.1 Performance optimizations

Data generation in the user interface has been improved by implementing multithreading.

3.4.2 Real-time streaming enhancements

One enhancement to KTIRIO-GUI was 3D rendering improvements from Qt 6.8.

3.4.3 Feature updates

The task manager is a key component of the GUI implementation, responsible for handling all processes with several enhancements: (i) it ensures that the user interface remains responsive by detaching threads from the main thread, (ii) manages complex workflows, including parallel processing, and (iii) monitors the progress of tasks. Additionally, the GUI now incorporates weather data management from OpenMeteo, a feature that has also been integrated into the KUB simulator. This component automatically fetches weather data into the input dataset, utilizing the time study and geographic position for accurate retrieval.

3.4.4 User interface updates

Several updates have been made to the user interface: (i) the geographical setup has been fully reviewed and is now carried out from the stack view, (ii) users must review all the parameters to avoid missing any information in the workflow, (iii) geographic data projects can be reopened by importing the setup from a JSON file, (iv) datasets can be exported from Ktirio Urban Simulator UI, (v) a weather UI has been added to allow fetching weather data, and (vi) many parameters of OpenMeteo API have been integrated.

4 Pilot visualization application

This section details how the enhanced visualization strategies are tailored for each pilot. For every pilot subsection, the content integrates both pilot-specific adaptations and practical case study examples. The structure for each pilot includes:

- **Pilot objectives:** A brief overview of the goals for the pilot.
- **Pilot-specific adaptations:** Details on how the visualization tools have been customized to meet the pilot's unique requirements, including adjustments in data processing, integration of specific data sources, and interface enhancements.
- **Implementation details:** An outline of the technical workflow, highlighting challenges encountered and solutions implemented.
- **Expected benefits:** A summary of the improvements in data interpretation, user interaction, and overall decision support.
- **Case studies and use examples:** Practical scenarios illustrating the complete data flow—from simulation data to interactive visualization—along with examples of user interaction features and preliminary results.

4.1 Urban Air Project (SZE)

4.1.1 Pilot objectives

Focus on monitoring and analysing urban air pollution through high-resolution CFD data, and LAWSON wind comfort zones for urban planning. The aim is to provide a detailed visualization of pollutant dispersion, wind dynamics thanks to different visualization techniques, such as streamlines, slices and volumetric point cloud rendering.

4.1.2 Pilot-specific adaptations

Customization of Visualization Tools: In order to properly visualize urban data, we had to break down each visualization into two distinct phases; a post-processing phase, done offline, and a visualization phase accessible in real-time from the web. The reason for the separation of these two phases is the size of the data, being up to hundreds of terabytes, and the necessary computation power for slices/streamlines/volumetric data needing an HPC machine when we're dealing with meshes with over hundreds of millions of cells.

Implementation details: The post-processing tool of CFDR was implemented using. MPI + C/C++, while the web-based client version of CFDR was implemented using WASM (from C/C++ source), alongside WebGL/OpenGL ES 2.0-3.0 for 3D graphics.

4.1.3 Expected benefits

Since we have written our post-processing tool with MPI-C/C++, we expect our post-processing tool to be fast enough to work with very large scale data and scale well. We also expect our web-based visualization client to be real-time thanks to the acceleration provided by the WASM/Emscripten framework.

4.1.4 Case studies and use examples

Scenario description:

Our main case study for CFDR is for UAP (Urban Air Pollution) and wind comfort estimation for urban planning. In the sections below, we present various results and methodologies for visualization of urban wind data, thanks to CFDR.

The simulations were done on Gyor, Hungary; Stockholm, Sweden in collaboration with them and Poznan, thanks to data provided by PSNC. Further case studies will be conducted, thanks to the other Pilot data as well in the future.

Data flow and visualization process:

In order to create a visualization with CFDR, the following steps are necessary:

1. Run a simulation, exporting the results into one of the formats supported by CFDR (Ensight Gold, VTK, VTM, NetCDF)

2. Write a CFDR visualization script in **LUA**, to specify the type of visualization the end-user would like to do. We provide an example for such a script below:

```
visualize.lua+
-- First, we declare a workspace. This is where the data will be packaged.
cfdr_workspace("./WWW_dev/gyor_m1", "Gyor-M1")

-- Declare a table containing all the 3D models to use.
city = {
  { id = "buildings",
    visible = true,    path = "./buildings.obj",   r = 0.9, g = 0.8, b = 0.9, a = 1.0 },
}

-- Push the geometry data to the visualizatio stack
cfdr_push(city, "GEO")

-- Declare a table containing all points of interests.
poi = {
    { id = "Sensor", visible = false, x = -285.910110, y = 455.263173, z = 5.0, r = 1.0 },
}

-- Add points of interests.
cfdr_push(poi, "POI")

slice = {
    { id = "z_5m", preload_series = false, visible = false, path = "./slice/z_5m" },
    { id = "x",    preload_series = false, visible = false, path = "./slice/x" },
    { id = "y",    preload_series = false, visible = false, path = "./slice/y" },
}

-- Push the slice data to the visualization stack
cfdr_push(slice, "SLI")


```

In this sample, we create a simple UAP slice visualization for Gyor, we add some geometries for the buildings.

3. Run the CFDR post-processing tool, on the lua script. This is done with a simple command, assuming the script is saved as the file ‘`visualize.lua`’
4. Finally, once the data has been generated by cfdr_compile, the data can be visualized by placing the output folder somewhere accessible by the website (i.e. next to the index.html file of the client, or some other folder accessible), and going to the CFDR client’s website URL, with a special tag such as: `#?location=”path/to/folder/to/visualize”`

Example: hpehpc.mathso.sze.hu:8090/#?location=test_5

User interaction and feedback:
CFDR is currently used internally at SZE, in order to visualize data from different solvers from RedSIM: OpenFOAM, XYST, ...
As mentioned in the Data Flow section, CFDR has a lua scripting API used to create visualization projects; our team at SZE uses this API in order to create visualizations for case studies.

Data from other pilots is currently under integration, and we have visualized data from the PSNC team already (see Visualization on structured meshes).

4.1.5 Preliminary results

We present below our current results for urban window flow on multiple geometries, post-processed in CFDR completely (CFDR code only, no external codes/libraries such as ParaView), and visualized in the CFDR web-browser application.

Visualization on unstructured tetrahedral meshes

When visualizing a slice, a vector-field can be put on top of the slice (tiny black arrows), in order to visualize wind direction.

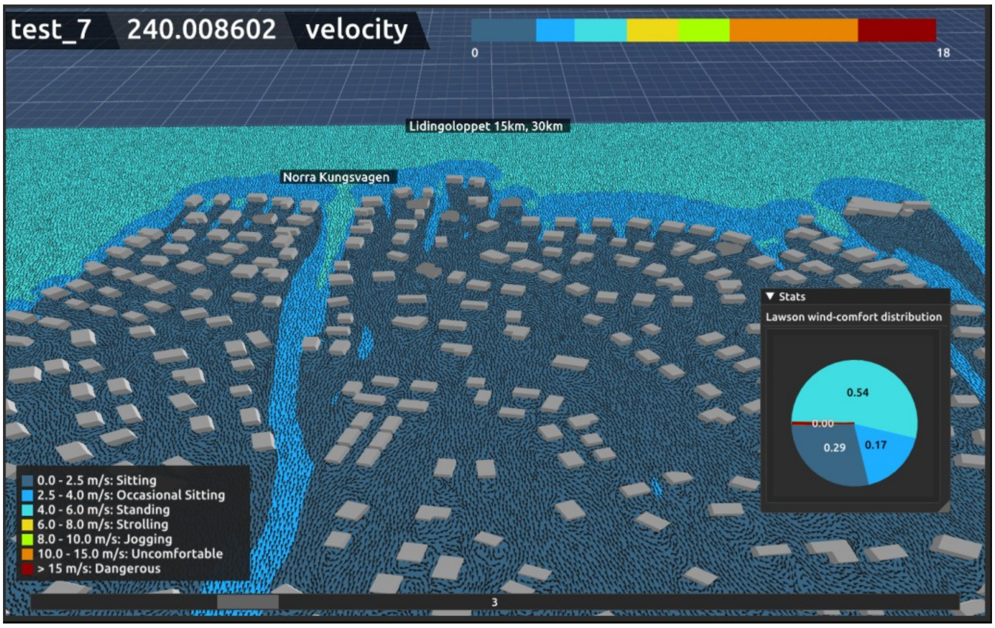

Figure 6. Stockholm LAWSON Wind Comfort

CFDR can compute statistics on the fly for wind-comfort, for example, the LAWSON wind comfort scale. The computation of these statistics only takes a few milliseconds, and thus is completely real-time. (> 60fps)

Visualization on structured NETCDF meshes

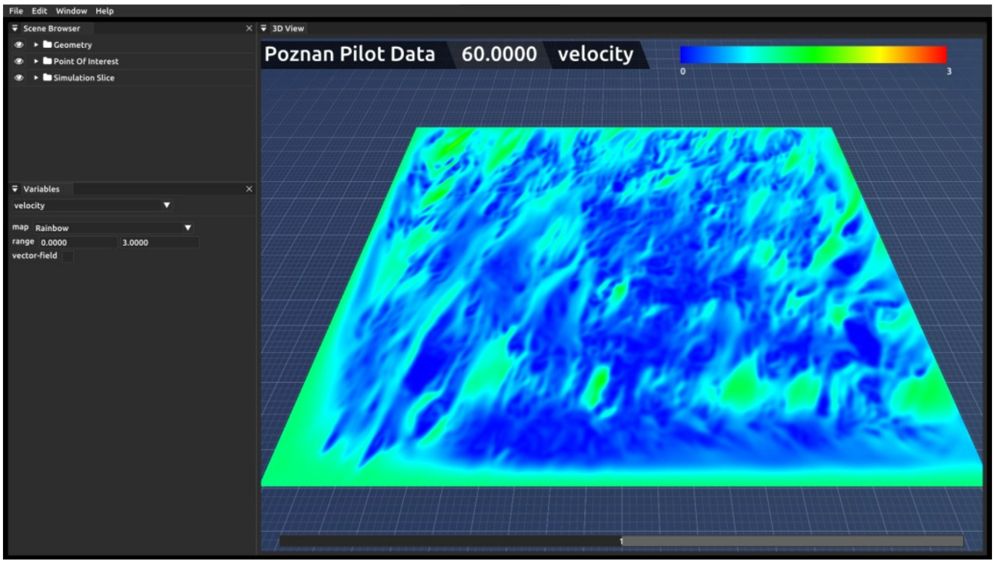

Figure 7. Poznań Wind Field (data provided by PSNC)

CFDR can visualize data from EnSight GOLD, VTK, VTM and NetCDF files currently; it supports visualization of structured (NetCDF) or unstructured grid (Ensight GOLD).
The data above was provided by the PSNC team.

4.1.6 Lessons learned

CFDR postprocessing

The goal of CFDR's preprocessing tool is to make huge datasets visualizable from the web, by creating visualization slices/streamlines/volumetric data

From a performance perspective, since we are working with **massive datasets** in terms of size on disk (multiple terabytes at times). Thus **file-read operations** and **well scaling parallel** processing is of key importance.

For file-read operations, we currently use **MPI for process** management**,** combined with **memory-mapped files**; however, we are currently working on an **io_uring** (Linux only) implementation combined with **MPI for process management**, in order to have truly asynchronous multi-process file-reads.

Aside from heavy IO work which is physically bound to disk read speeds, it was also very important to **accelerate our algorithms with a combination** of **SIMD + PTHREADS + MPI**. Our SIMD implementation is based on evaluating values at vertices for our slices, from polyhedral cells (each polyhedra is subdivided into multiple tetrahedra, and the algorithm handles tetrahedra only). Each SIMD entry processes one tetrahedra, thus we gain 4x (for SSE), 8x (AVX2) or 16x (AVX512) speeds depending on what the CPU ISA supports.

CFDR client-side visualization

On the client side, **data compression** and efficient data representations turn out to be key. As detailed in **3.2.1 - Real-Time Streaming Enhancements**, in order to have a seamless experience streaming visualization data, we had to first convert our data to 16-bit floating points, followed by an LZ4 compression in order to reduce packet sizes.

Another aspect that was crucial was the use of **WASM + OpenGLES 2.0 / WEBGL** in order to render 3D data efficiently. The entire client is written in C++, compiled with emscripten to WASM. Unfortunately, emscripten is very heavy so it is in our future plan to switch to LLVM directly (emscripten is built on top of LLVM).

Unfortunately, ES 2.0 / 3.0 is very far behind in terms of GPU API-s, but seems like the only option currently for cross-platform / cross-browser rendering, since WebGPU is still unsupported on a lot of browsers as of writing this (non-experimental support), such as Safari, Firefox.

In order to make some geometry computations faster on the CPU-side (vector field distribution, etc.), we also leveraged SIMD instructions in WASM; these appear to be properly supported in all major browsers these days.

4.2 Urban Building Model (UNISTRA)

4.2.1 Pilot objectives

Analyse city performance and energy efficiency by visualizing both static and dynamic data from urban building simulations. The focus is on integrating environmental factors with architectural data.

4.2.2 Pilot-specific adaptations

Customization of visualization tools:

KTIRIO-GUI has been enhanced to support the specific needs of urban building simulations. Enhancements include:

- Improvements in data generation and preparation via multithreading.
- Integration of advanced data processing capabilities to handle large volumes of static and dynamic simulation data.
- Updates to export protocols to support higher-resolution datasets and to improve interoperability with external tools such as BIM and GIS systems.
- Enhanced support for automated weather data integration using the Open Meteo API, which enriches simulation inputs with real-time meteorological information.

Implementation details

- **Data Integration**

The simulation data—comprising building geometry, occupancy schedules, and environmental parameters—is integrated into the KTIRIO-GUI system. The system associates imported building data with defined thermal zones.

- **API Adjustments**

Necessary API modifications have been implemented to ensure smooth and efficient data exchange between simulation outputs and the visualization platform, including handling of new data formats and real-time updates from the FMU (Functional Mock-up Unit).

- **Workflow Management**

The task manager within KTIRIO-GUI now effectively handles complex workflows. This is achieved by detaching processing threads from the main thread, ensuring a responsive user interface even during intensive data processing tasks.

4.2.3 Expected benefits

The improvements in the Urban Building Model pilot are expected to provide:

- More accurate and detailed visualization of multi-story buildings with clearly defined thermal zones.
- Enhanced integration of real-time meteorological data, resulting in more precise simulations of building performance.
- Improved interoperability with external analytical tools, facilitating seamless data exchange and further analysis.

- A more intuitive interface that caters both to technical experts and decision-makers by clearly representing key performance metrics.

4.2.4 Case studies and use examples

Scenario description:
A one-year simulation was conducted for the city of Strasbourg, covering the period from January 1, 2023, to December 30, 2023. Meteorological data was sourced from Open Meteo, with measurements taken at a specified location. The simulation domain is defined by a grid measuring 2 km by 1.5 km, cantered at the coordinates 48.581833, 7.750832. Level of Detail 0 (LOD0) geometry was employed for modelling buildings.

Buildings in the simulation are divided into up to 10 thermal zones, with each zone corresponding to a single floor of nominal height 3 meters. All buildings are assumed to be equipped with boiler-based heating systems and standard insulation, with thermal properties uniformly defined. The table summarizes the R values for different building component.

Table 2. R Values for different building components

Description	R value (W/m²/K)
Slab	2.24
Wall	3.88
Roof	0.62

Occupancy patterns are defined based on tailored scenarios for each building type, influencing heating temperatures with up to three setpoints (occupancy, inoccupancy, and vacation):

Table 3. Occupancy patterns

Building Type	Occupancy Range	Occupancy Temperature	Inoccupancy Temperature	Vacation Temperature
Building	Monday to Friday, 8h–18h; 2 weeks' vacation in August and 1 week in December	19	16	7

Residential	24/7, 7 days a week; 2 weeks' vacation in August and 1 week in December	19	-	7
Office	Monday to Friday, 8h–18h	19	16	-
School	Monday to Friday, 8h–18h; French school vacation (16 weeks)	19	16	7

- A one-year simulation was conducted for the city of Strasbourg, from January 1, 2023, to December 30, 2023. Meteorological data were sourced from Open Meteo, with measurements taken at x location. The simulation domain consists of a grid measuring 2 km by 1.5 km, cantered at the specified longitude and latitude coordinates: 48.581833 , 7.750832. Level of Detail 0 (LOD0) geometry was employed for modelling buildings.

- **Data flow and visualization process:**

  Static data:
  The simulation domain's geometric characteristics include a mean building height of 12.5 meters and a median height of 12.4 meters. The average building surface area is 483 m² (median: 194 m²), with 3,664 buildings modelled. The distribution comprises 73.3% buildings, 21.3% apartments, 0.77% schools, and 4.63% other structures.

  Dynamic data (preprocessing):
  Meteorological parameters—such as wind speed, exterior temperature, cloud cover, solar radiation (direct and diffuse), relative humidity, and atmospheric pressure—are recorded hourly using KTIRIO-GUI. Historical data from the past 30 years is used to calculate monthly averages for comparison.

  Dynamic data (postprocessing):
  Data from the FMU instance is processed to derive key thermal performance metrics:

  - **Interior temperature:** Aggregated hourly averages to assess thermal stability.

- **Heating power:** Averaged per hour to evaluate fluctuations in energy demand.
- **Comfort indicators:** Computed on an hourly basis to gauge deviations from optimal indoor conditions.

- **User interaction and feedback:**

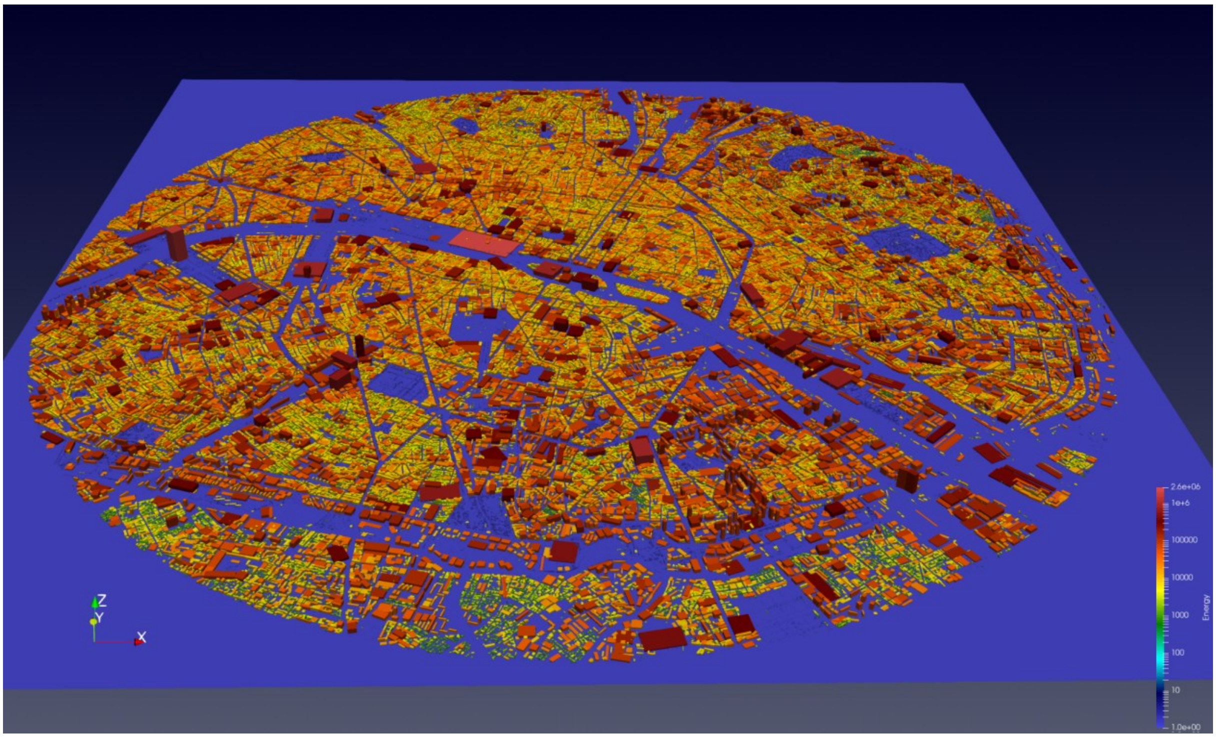

Figure 8. ParaView screenshot of a simulation of Paris with a radius of 5km (102506 buildings) displaying power output for 01/01/2023

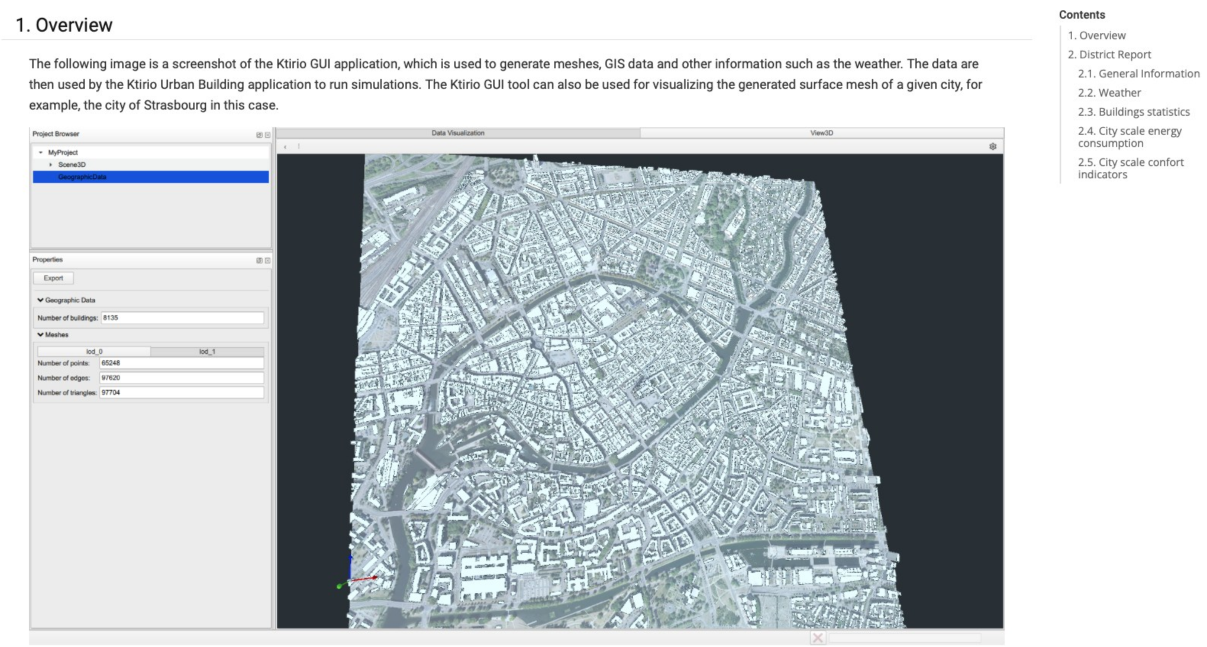

Figure 9. Screenshot of the first section of ktirio.cases website with screenshots of ktirioGUI application. and table of content on the top right of the screen

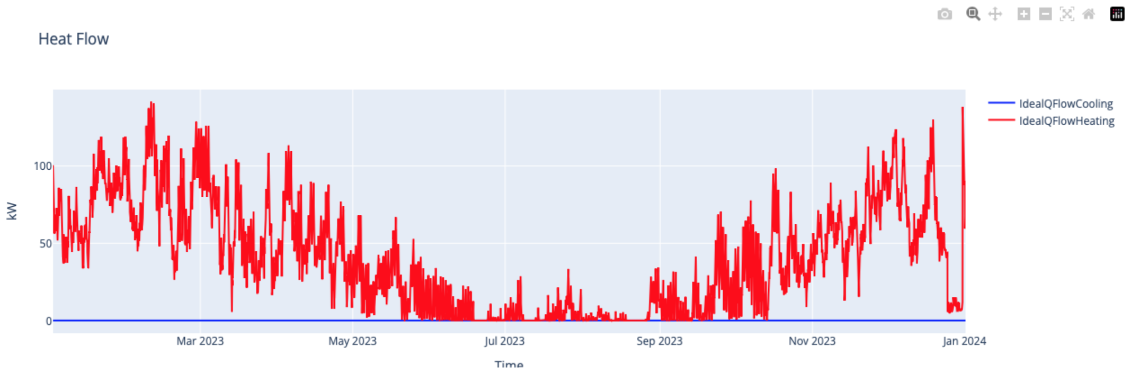

Figure 10. Screenshot of the averaged power output across the city automatically generated on ktirio.cases

4.2.5 Preliminary results

- As the pilot development progresses, simulations become increasingly detailed and precise. The newly generated data is collected and prepared for visualization using tools such as ParaView and ktirio.cases. Efforts have been made to present this information in a clear, concise, and visually engaging manner. As of now, more is on the way results obtained by the simulation will be presented in a future deliverable D 5.7
- The integration of solar shading mechanisms enabled the visualization of new graphs while enhancing the overall accuracy of the simulation.

- Advancements in meteorological data processing have contributed to a higher level of detail in the simulation, aligning with professional reference for the representation of meteorological information.
- The ParaView tool facilitates the visualization of data in an intuitive and accessible manner for users with limited expertise. Additionally, the spatial representation of data offers valuable insights for specialized audiences seeking specific points of interest.
- Ktirio.cases caters to a more knowledgeable audience by presenting averaged results across a range of parameters, offering a comprehensive and accessible overview of city's simulation result.

The simulation assessed heating power demand across various building types and occupancy patterns. Seasonal variations had a significant effect, with colder months requiring higher energy input while transitional periods experienced a decrease in heating needs. Vacation periods led to noticeable reductions in energy consumption.

4.2.6 Lessons learned

The integration of advanced simulation features—such as solar shading, multi-story modelling, and diverse heating systems—has increased computational demands. However, this trade-off is essential for achieving higher precision and more detailed representations of building performance. Future work will focus on optimizing the balance between data volume and computational efficiency to deliver precise yet performant visualization outputs.

4.3 Renewable Energy Sources (PSNC)

4.3.1 Pilot objectives

Enhance the understanding of weather phenomena affecting renewable energy outputs by visualizing unsteady weather fields in three dimensions. The goal is to improve energy production forecasting accuracy, prevent damages to the overhead electrical network, and support strategic energy planning.

4.3.2 Pilot-specific adaptations

Customization of visualization tools: As for now we are using ParaView for general scientific visualization, Python-based tools for processing and visualizing forecast data, and own tools to plot basic data onto geospatial information as reported in deliverables D4.6 and D5.3.

Implementation details: Currently, the visualization process is part of the workflow to some extent. Within a single workflow, after coupled solvers finalized computations, there is a post-processing stage at which static and dynamic visualization is created to plot some basic weather forecast data. More advanced scientific visualization is done offline using ParaView. The same applies to the projection onto geospatial data.

4.3.3 Expected benefits

Present energy production forecasts and potential damages to the infrastructure in a modern, eye-catching, yet valuable visualization for scientific purposes and stakeholder requirements.

4.3.4 Case studies and use examples

Scenario description: predictions of damages to the overhead network due to extreme weather conditions. The scenario is focused on the effects of excessive wind speed and wind gusts on the overhead infrastructure. In the workflow, three levels of domain nesting were used, where the outer domain of the scale of the whole country had a resolution of 3.6km while the final and most fine-grained domain was discretized with the spacing of just 100 meters.

Data flow and visualization process:

- **Static data:** The domain of buildings and streets is prepared from SHP files provided by municipal authorities. In the absence of this data, it is scraped from OpenStreetMap and converted to the format required by RES solvers. Outside the urban environment, DigitalElevationModel data required for scenarios outside the urban environment is obtained from Copernicus or SRTM repositories. Information on the overhead electrical network is provided by the Distributed System Operator.
- **Preprocessing (dynamic data)**: Meteorological forecast data is required to formulate initial/boundary conditions. The openly available data from the Global Forecasting System is used.

- **Postprocessing (dynamic data)**: basic forecasted data such as wind speed and direction, pressure or temperature is created as static and animated pictures. Manual postprocessing allows to plot this data onto geospatial information for the stakeholder analysis on potential damages.

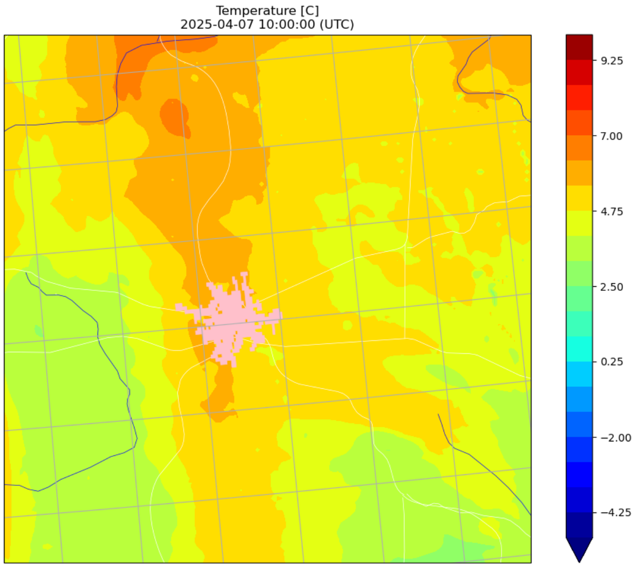

Figure 11. RES temperature forecast

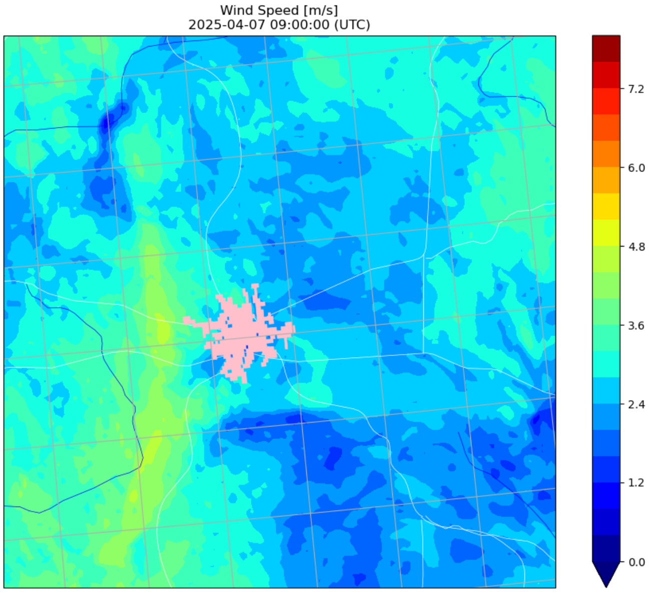

Figure 12. RES wind speed forecast

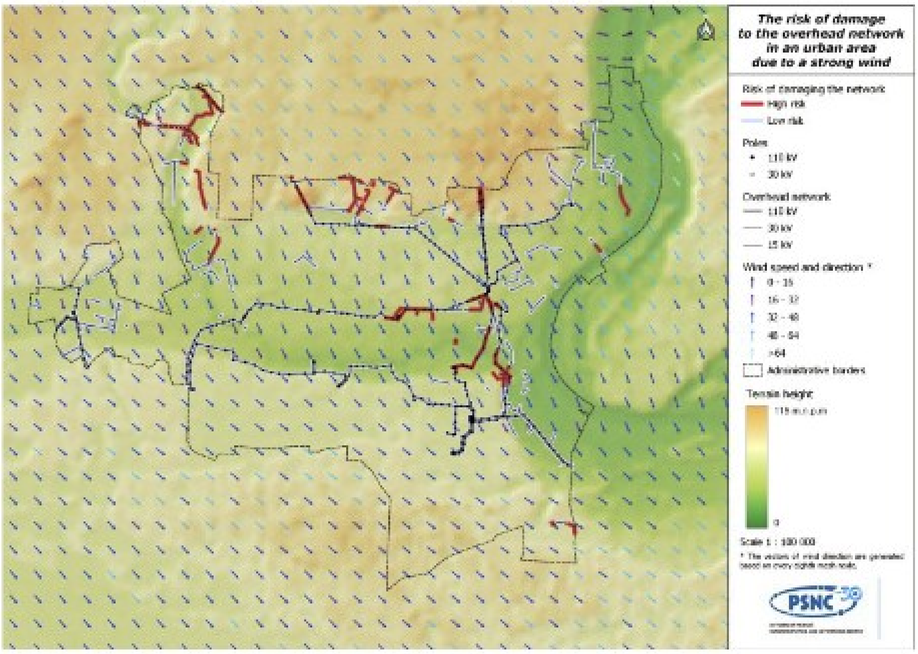

Figure 13. RES prediction of damages to the overhead network

4.3.5 Preliminary results

RES pilot is able to provide valuable insight into potential damages to the overhead network through visualization processing. It is of utmost importance for the stakeholders to understand the problem in clear way.

4.3.6 Lessons learned

There are important steps to be taken. First, an automatic plotting of forecast data onto geospatial information is required. Second, the forecast and visualization of energy production is to be provided. Automating the overlay of forecast data onto geospatial maps is essential to save hours of manual work, ensure consistent, reproducible outputs, and enable near-real-time updates for emerging extreme-weather alerts, while extending the workflow to include energy-production forecasting directly links meteorological conditions to expected generation—empowering operators and stakeholders with clear, combined weather-and-yield visualizations for more informed scheduling, risk assessment, and strategic planning.

4.4 Wildfires (MTG)

4.4.1 Pilot objectives

Develop realistic visualizations of wildfire dynamics, focusing on accurately representing fire fronts and smoke dispersion at two scales, the landscape and the urbanization. The aim is to support emergency response planning and training.

4.4.2 Pilot-specific adaptations

No adaptations have been made to the calculation software. Only the specific data for the analysis areas have been obtained: (i) Digital Elevation Model, (ii) Forest fuel map and (iii) Observational atmospheric data for the simulation days.

Additionally, optimized parameterizations and configurations have been carried out for this type of simulation. Four nested downscaling domains were used to achieve atmospheric grid resolutions between 100 and 200 meters, and fire simulation grid resolutions between 10 and 20 meters. The configuration of the innermost domain was done through subdomain partitioning for computation on HPC facilities.

Customization of visualization tools

For advanced wildfire visualization at the landscape scale, a bridge software has been developed to convert NetCDF files into VDB files of sparse volumetric data, enabling their integration into Unreal Engine.

Implementation details

The visualization of the simulations was carried out using the results obtained with WRF-SFIRE on several EuroHPC facilities. The outputs were written in NetCDF format. Only the passive smoke tracer variable (Trace17_1) was used. The simulation was implemented in Unreal Engine using VDB files, which were converted into volumetric textures and applied in volumetric materials through Ray Marching routines. This process was repeated for all frames, corresponding to one per minute of simulation time, totalling 6 hours (360 frames in total). The resulting heterogeneous volume visualization was integrated into Google Tile Service geometry via CESIUM. The final experience was compiled in Unreal Engine for Windows platforms and viewed using MetaQuest 3 virtual reality headsets.

4.4.3 Expected benefits

The visualization of fire and smoke simulations from wildfires in immersive environments helps users understand the details of their spread across the landscape in a much more intuitive way. The user can move around within the scene and also navigate along the time axis. Similarly, multiple simulations can be integrated into the same experience for comparison and analysis of landscape sensitivity.

4.4.4 Case studies and use examples

Several implementations of the visualization workflow have been carried out for some of the wildfire simulations run on HPC systems.

Cadalso de los Vidrios (Scaffold of Glass)

A series of tests was conducted for the Almorox-Cadalso fire (2019), Spain, which took place precisely in this area. The fire origin was set at coordinates Lat = 40.244977°N and Lon = -4.376590°E, with the start time set at 17:54 local time (15:54 UTC) on June 28th, 2019.

An area of 51,493 km² in the centre of the Iberian Peninsula was selected, spanning the Autonomous Regions of Madrid and Castilla-La Mancha in Spain. A central point within the study region was designated as the reference position, located at coordinates Lat = 40.550938°N and Lon = -4.0767517°E. The target resolution of the atmospheric grid is 200 meters, while the fire simulation grid has a resolution of 20 meters—ten times more detailed. Four nested domains (D01–D04) with increasing resolutions were established (Fig. 1), enabling the downscaling of atmospheric data from the ERA5 reanalysis for the dates and times of the fires. A total of 360 frames were generated, each corresponding to a one-minute simulation timestep, covering a total duration of 6 hours.

Figure 14. Example of integrating WRF-SFIRE simulations into Unreal Engine 5 using Google Tile Service geometry via CESIUM for the 2019 Cadalso de los Vidrios wildfire.

Figure 15. Figure 16. The same simulation, but viewed closer to the ground. The orange tint caused by Mie scattering due to the presence of particles in the atmosphere is clearly visible.

4.5 Material Transport in Water (FAU)

4.5.1 Pilot objectives

Visualize complex CFD data related to fluid dynamics and particle transport in water, with an emphasis on multi-scale data handling and dynamic particle tracking.

4.5.2 Expected benefits

The visualization using ParaView is for the ease of the user, as it is an open-source tool for visualization.

4.5.3 Case studies and use examples

Scenario description: We have developed a fluid and temperature two-way interaction solver that is capable of solving thermal convection problems. We have until now done visualizations of a natural convection problem involving a differentially heated cavity with a hot plate on one side and a cold plate on the other side. Another visualization is the classical Rayleigh-Bernard Convection problem with a hot plate at the bottom and a cold plate at the top of the domain.

Data flow and visualization process: The simulations are run for a certain number of time steps until the problem is converged. The user can specify the folder name for writing the vtk files and the frequency of writing the vtk files, i.e., write a vtk file for both temperature and fluid after every specified number of time steps. This feature is implemented to save disk space and computational time, as writing vtk files is computationally very expensive. After the simulation is complete, the vtk files can be loaded into ParaView and viewed with various available options, typically in surface visualization mode.

4.5.4 Preliminary results

For Brevity, 2D simulation visualizations have been presented in the current deliverable. However, the same visualizations can also be obtained for 3D simulations. There are 64 cells in the x and y directions and one cell in the z-direction. These simulations for visualizations are done on very small domain sizes and will later be extended to bigger domains as and when the model development progresses.

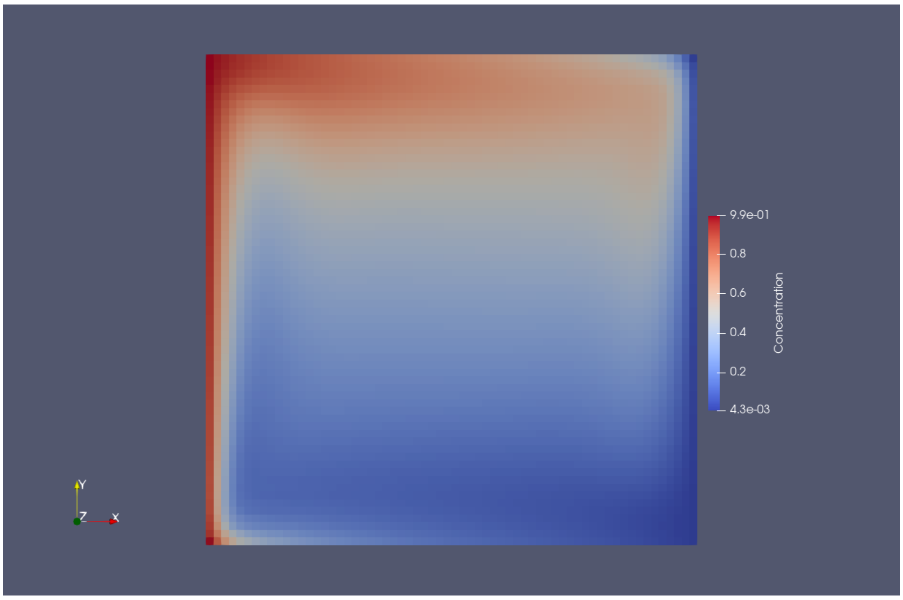

Figure 16. Converged steady state visualization of 2D differentially heated square cavity

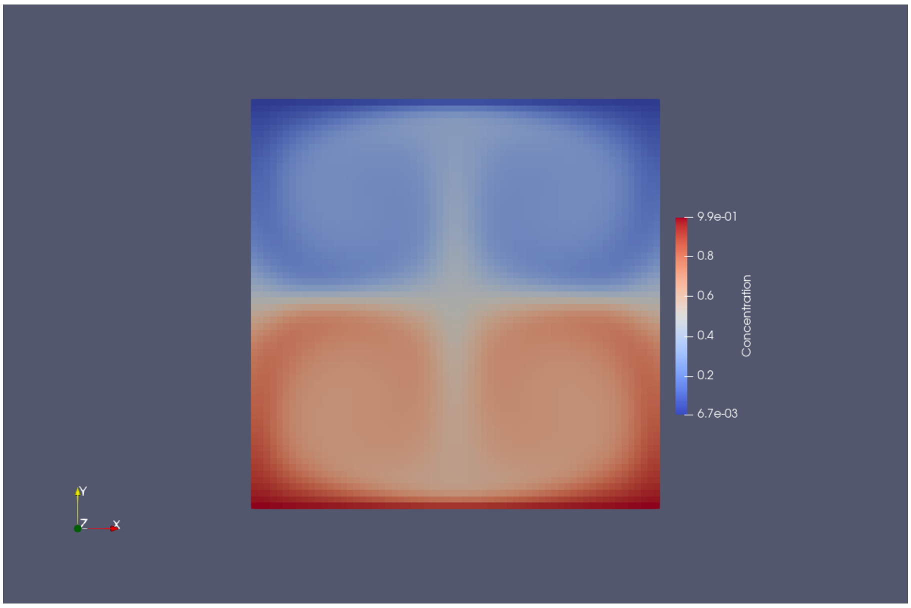

Figure 17. Converged steady state visualization of 2D Rayleigh Bernard Convection

4.5.5 Lessons learned

The writing of the VTK files is in general, a computationally expensive process. Therefore, care must be taken while setting VTK writing frequency because the higher the frequency of writing, the higher the overall runtime of the simulation. It is worth noting that for performance benchmarks, it is not a good idea to write VTK outputs, as a major chunk of performance can be lost due to it.

5 Future directions

This section outlines the planned developments, upcoming features, and potential research opportunities for enhancing the visualization tools.

5.1 Computational Fluid Dynamics Rendering (SZE)

New features

As outlined in section 2.2.1 in the updated requirements, our goal is first and foremost to develop volumetric visualization methods, and streamline visualizations in order to support visualization of pilot data that requires it (i.e. MTG wildfires volumetric smoke data).

Performance optimization

Similar to our SIMD-based slicing algorithm, we plan on accelerating other parts of the preprocessing step for CFDR, such as volumetric data processing and streamline generation (multiple streamlines generated at once thanks to SIMD).

We also plan on implementing a more aggressive data compression algorithm, based on a more lossy compression scheme (our current compression scheme is technically lossy, since we converts 32 bit floating points to 16 bit floating points, but the compression of the 16 bit floating point values themselves are non-lossy LZ4).

Documentation

We plan on documenting in really fine detail all the API-s for the CFDR preprocessing tool, alongside the usage of the client web-based GUI tool. The documentation will be on a website online available to all users, alongside a list of comprehensive examples.

5.2 Unreal engine advanced visualization (MTG)

Enhancements and improvements:

The future development of visualization in the wildfire pilot will focus heavily on the portability of demonstrators to Android platforms, particularly for use with MetaQuest3 virtual reality headsets. As demonstrated throughout the project, rendering smoke and flames, vegetation, and other elements of the forest landscape requires significant graphical power to ensure 60 frames per second in stereoscopic view. Currently, MeteoGrid is using high-performance workstations equipped with NVIDIA GeForce RTX 4080 graphics cards connected via USB cable to virtual reality devices like the

MetaQuest3. The advantage of this setup is that it provides the graphical power necessary to render complex scenes. However, the main drawback is the need for a wired connection, which limits user mobility.

The alternative currently under development involves organizing scenes using hybrid techniques with immersive 360-degree stereoscopic videos. This approach encapsulates the rendering of complex environments into these videos, which are then used as the background for the main scene. In this main scene, additional 3D objects with which the user can interact are placed. This hybrid strategy allows the demonstrators to run on devices with less memory and, more importantly, significantly lower graphical capabilities. Even so, the Android portability strategy also includes optimization of polycount (3D model complexity, especially vegetation), the use of Level of Detail (LoD), and selection of lower-resolution textures.

On another front, development will continue on the vegetation definition tool, which will allow users to describe 3D models of plants based on a catalogue by species, age, and health status. As explained earlier, this workflow is aimed not only at providing greater visual realism in the simulations but also at enabling a more accurate representation of biomass distribution and porosity.

In the demonstrators designed for the wildland-urban interface (WUI) scale, the integration of ad-hoc photogrammetric models of both buildings and vegetation will be proposed. Additionally, the integration of Gaussian Splatting training models for the same selected areas will be explored. Both types of models will be integrated with the smoke and flame representations from fluid dynamics simulations.

For the development of these experiences, a residential area in the city of Gyor (Hungary) near a forested zone has been selected. This decision is based on the fact that Gyor is the study area for other pilots within the project (UAP), allowing for data and model exchange between them.

Integration and testing:

Key milestones for integration and testing of new features:

- Development of 3D wildfire scenes, with vegetation and volumetric flames and smoke imported from HPC simulations.
- Selection of the workflow and tools for the rendering of 360 stereoscopic immersive videos. Rendering of wildfire scenes and storage in 360 stereoscopic immersive videos (360 stereoscopic cinematics).
- Development of the visualization strategy for 360 stereoscopic immersive videos in the VR scenes of the demonstrators in Unreal Engine: stereoscopic projection of 360 videos.

- Integration of other scene components, 3D models. Assembly of the final hybrid scene.
- Development of interaction and navigation mechanics.

Collaboration and feedback:

There will be two lines of collaboration: on the one hand, interaction with other pilots within the project, especially UAP, since they share the same geographical region and plan to carry out coupled WF-UAP simulations. On the other hand, interaction and collaboration with firefighting services and property owners in the urbanized area, with the goal of presenting the simulation results for awareness and training purposes. This interaction will take place through presentations using the use case demonstrators, during which stakeholders' feedback and recommendations will be gathered.

Roadmap and future work:

The proposed activities will be carried out throughout 2025, and new case studies will be implemented during 2026. Additionally, at the end of 2025 and the beginning of 2026, a joint demonstrator on wildfires (WF) and air pollution (UAP) will be developed.

5.3 VISTLE-COVISE (USTUTT)

Enhancements and improvements

Outline opportunities to improve overall performance, scalability, and functionality. Identify areas for future research and technology exploration. Within the framework of this project, supporting more workflows will be a major focus. Thus, importing data from additional sources and more varied processing methods will be enabled on a case-by-case basis.

Integration and testing

Define key milestones for integrating new features with existing systems. Propose a generic testing strategy to validate enhancements and ensure quality. Together with other projects that also use the frameworks COVISE and VISTLE, continuous integration workflows will be improved. A major outcome here will be a product ready for deployment to all relevant releases of Debian and RedHat-based Linux distributions. Also, a framework for testing for functional regressions will be devised. Initial ideas rely on fuzzy matching of known good results, e.g., renderings, against results from test runs. However, as these efforts are too huge to be undertaken within the framework of this project, no timelines for this are defined.

Collaboration and feedback

Emphasize the need for ongoing collaboration with pilot teams and stakeholders. Include plans to collect and incorporate user feedback for iterative refinements. Communication with decision-makers and other users of the system is important to identify deficiencies and areas for improvements. This feedback will enable focusing the efforts on useful enhancements.

Roadmap and future work

Develop a roadmap that outlines both short-term and long-term objectives. Ensure that the future work aligns with overall project goals and evolving requirements. A major effort will be to enhance the performance of data streaming between visualization clusters. Especially the bottleneck of funnelling all the data through a single rank both on the sending and receiving side will be addressed: the system will be enhanced to allow for direct connections between individual ranks of all participating clusters, if network topology permits. This will allow to better saturate available network bandwidth by using several network links in parallel.

5.4 KTIRIO-GUI (UNISTRA)

Enhancements and improvements:

The plan is to improve overall performance, scalability, and functionality. In this context, developers should identify (i) areas for future research and technology exploration, and (ii) opportunities to offload intensive computations (such as geometric tools) by creating HPC or cloud services. Additional improvements include replacing the current Qt Charts module with the new Qt Graphs module (available in Qt ≥6.8), which will enable the visualization of data in both 2D and 3D graphs. There should also be a focus on WASM improvements, such as enabling export of data files into the local filesystem and addressing missing Qt modules like Qt Location and Qt Position. These modules, which support dynamic linking of shared libraries, are currently under development by the Qt Team; alternatively, developing a custom solution should be considered, with careful estimation of the required effort. Finally, support for varying levels of detail when handling large 3D datasets should be improved to enhance the fluidity of 3D manipulation.

Integration and testing:

Developers should define key milestones for integrating new features with existing systems and propose a generic testing strategy that validates these enhancements while ensuring quality. This includes setting up integration tests that simulate realistic usage scenarios to verify that performance and functionality meet the set requirements.

Collaboration and feedback:

It is important to emphasize the need for ongoing collaboration with pilot teams and other stakeholders. Plans include systematic approaches for collecting and incorporating user feedback into iterative refinements, ensuring that the tool evolves in line with user needs and operational requirements.

Roadmap and future work:

Develop a comprehensive roadmap that outlines both short-term and long-term objectives. In the short term (within three to six months), the focus should be on: (i) deploying simulation capabilities on HPC resources via the UI, (ii) visualizing simulation reports, (iii) rendering simulation results on 3D meshes, (iv) developing a specific WASM application for deployment into the Hidalgo2 portals (which involves removing unused components and reviewing the UI in the Hidalgo2 context), and (v) advancing the development of geometric features and algorithms, including plans to mesh additional city entities (such as roads, rivers, and parks) to allow the GUI to apply more refined colouring and textures—potentially replacing the current reliance on satellite images.

In the long term, the focus will shift to: (i) enabling advanced data overlay so that statistical data (such as comfort indices and CO_2 levels) can be directly overlaid on 3D urban models with interactive filtering options, (ii) enhancing export formats by updating data exchange protocols to support higher-resolution datasets and improved interoperability with BIM and GIS systems, (iii) further improving 3D visualization to achieve a more immersive experience with new camera perspectives (for example, first-person view) and enhanced interactivity (such as allowing users to click on a building to retrieve metadata).

The roadmap should ensure that future work aligns with overall project goals and evolving requirements, while also planning for continued maintenance and support rather than extensive new development beyond these planned enhancements.

6 Conclusions

Deliverable **D4.7** marks a turning-point for the HiDALGO2 visual-analytics stack. During the last reporting cycle we moved from proof-of-concept demonstrators to **production-grade, cross-pilot workflows** that can already be embedded in pre-operational digital-twin services.

- **Technical maturity.** All four core tools (CFDR, VISTLE-COVISE, UEAV and KTIRIO-GUI) now run end-to-end without third-party visualisation software, scale to EuroHPC machines, and stream compressed assets to web or VR clients in (near-)real time.
- **Requirement closure.** The updated requirement set (REQ-POR-073…077) has either been met or is scheduled with committed resources and clear ownership. Full EnSight Gold support across CFDR and VISTLE-COVISE closes a long-standing interoperability gap.
- **Pilot impact.** Early adopters report tangible benefits: city-planners in Strasbourg can inspect comfort indices in minutes rather than days; network operators in Poland receive storm-damage maps fast enough to redirect repair crews; wildfire-fighters confirm that UEAV smoke dynamics foster better tactical awareness than 2-D maps.
- **Synergy across pilots.** Shared data formats and a common streaming layer allow, for the first time, **cross-coupled visualisations** such as wildfire smoke feeding city-wide air-quality dashboards. This validates the architectural decisions taken in WP2 and WP4.
- **Readiness for WP5 & WP6.** D4.7 delivers stable APIs, container images and documentation that can be integrated by the dashboard (WP5) and dissemination (WP6) teams without refactoring.

Looking ahead, the future-work plan in Section 5 focuses on three levers that will keep the platform at the cutting edge:

1. **Performance headroom** (via SIMD-enabled volumetrics, direct rank-to-rank streaming, and GPU path tracing);
2. **Portability** (lightweight WebAssembly and Android builds to reach low-cost devices);
3. **Usability & uptake** (online docs, Lua & Python bindings, and automated deployment recipes).

With those elements in place, the Hidalgo² visualisation framework is on track to support fully digital-twin operations and to remain sustainable beyond the project lifetime.

References

1. Caballero, D. (2024) A method for importing and drawing SHP files in Unreal Engine. Technical Notes Series (preprint). DOI: 10.13140/RG.2.2.26576.48648/1.
2. Museth, K. (2013) VDB: High-resolution sparse volumes with dynamic topology. ACM Trans. Graph. 32, 3, Article 27 (June 2013) 22 pages. DOI: http://dx.doi.org/10.1145/2487228.2487235
3. Caballero, D., Torres, L., Türk, K., Horváth, Z., Constans, M., Prud'homme, C., Koestler, H., Szeliga, W., & Lawenda, M. (2024). HiDALGO2 D4.6 Visualizations for Global Challenges. ResearchGate. https://doi.org/10.13140/RG.2.2.30758.95045/1