Opensource software at TotalEnergies

Research and Technology usage of opensource.

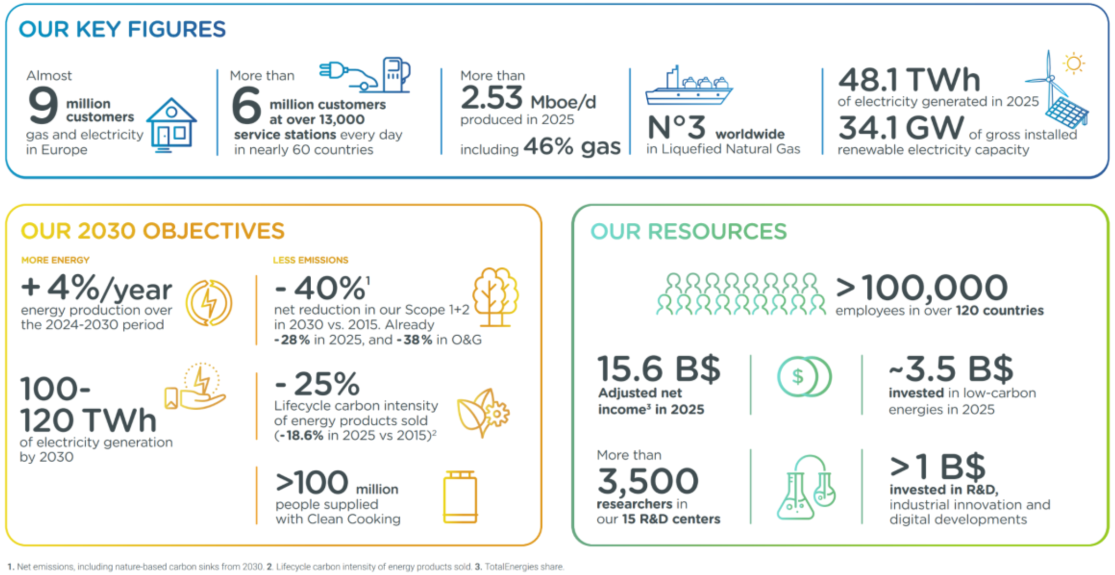
M. Pallud

04 Aug. 2026, DOE IESO Open-Source Software Summit

TotalEnergies

TotalEnergies

- TotalEnergies → multi-energies:
  - Legacy: oil & gas
  - Growth: solar, onshore/offshore wind, storage, …
  - Emerging: power & grids, …
- Onetech branch:
  - Central tech & R&D (3700p+): simulation, HPC, digital twins, AI platforms
- My role at TotalEnergies:
  - Scientific computing specialist / HPC background
  - Former experience: startups, large industry, software editor



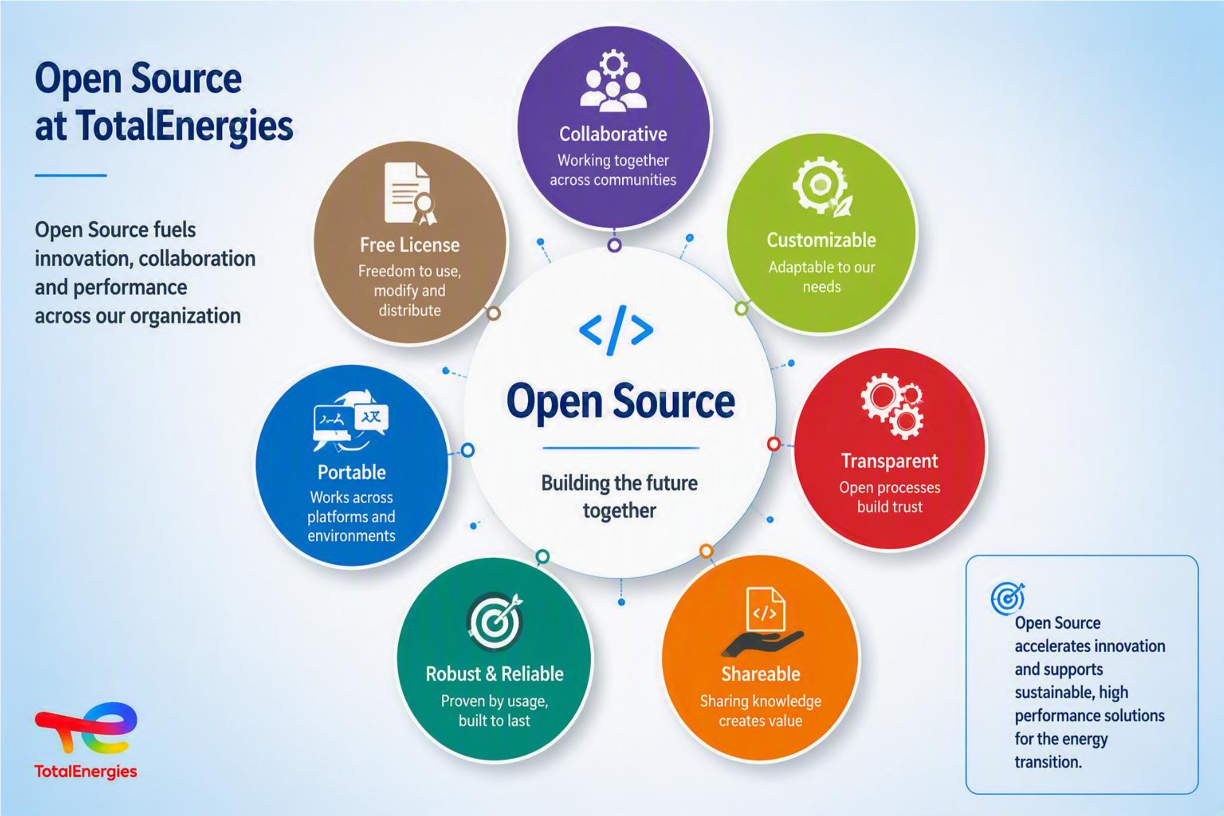
TotalEnergies climate report 2026

TotalEnergies's Pangea-4 HPC cluster (2024)

Why Open Source Matters for Energy industry

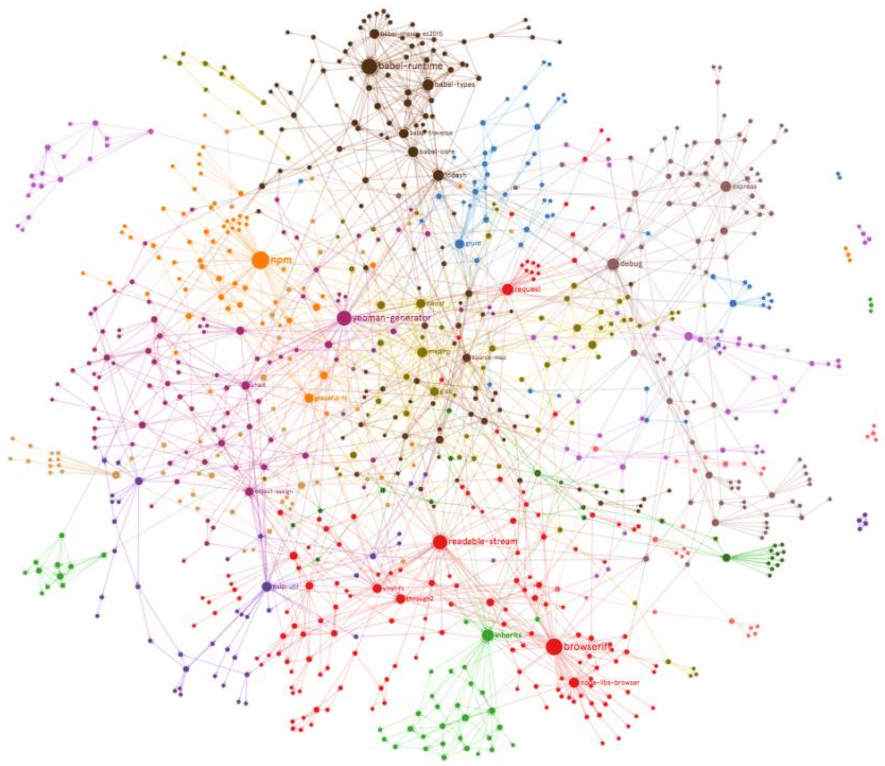
TotalEnergies

- Technical:
  - Multi-physics, multi-scale: atmosphere → turbine → farm → grid
  - Mix of HPC, reduced-order models, and data-driven methods
- Organizational:
  - Reproducibility for regulators and partners
  - Faster innovation and tech transfer
  - Collaboration with labs, academia, startups
- Open source as: shared methods, tools, datasets



Open Source in Practice: Tools & Fragmentation

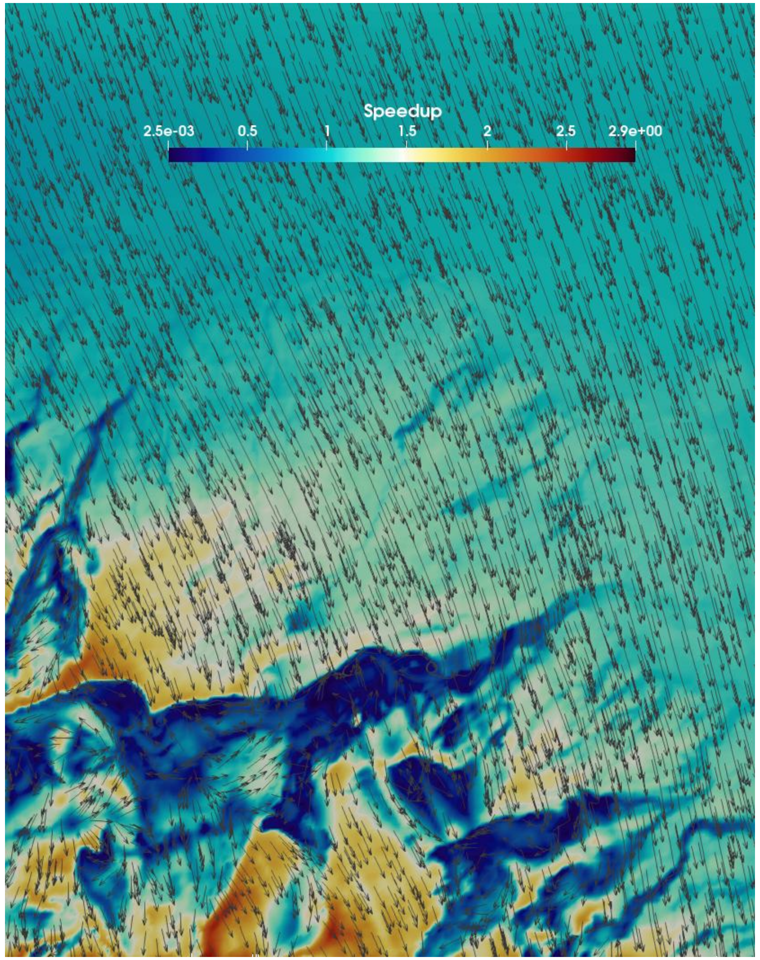
- Examples:
  - **Kynema-SGF / amr-wind** – atmospheric & wind farm CFD
  - **PVLib** – solar system performance
  - **Seahowl** – floating offshore wind dynamics (on Project Chrono)
  - Python stack – NumPy, SciPy, Pandas, xarray, Jupyter, Marimo, …
- Academic pattern:
  - Many overlapping codes, often per PhD/project
  - Limited long-term maintenance
  - Identify clear “flagship” codes with strong governance
  - Explosion of number of dependencies (python)
    - package managers have an increasingly difficult job



COMFLOW: NLR–TotalEnergies Collaboration for Complex Terrain Wind CFD

TotalEnergies

- **COMFLOW** – high-fidelity CFD workflow for wind resource over complex terrain
- Collaboration:
  - **Three-year research partnership** between **TotalEnergies** and **NLR** (formerly NREL)
  - Joint development, validation, and industrial deployment
- Scientific objectives:
  - Wind speed, turbulence intensity, and temperature maps ≥ 100 m resolution over **50×50 km** domain
  - Multiple atmospheric stability regimes
  - Turnaround < 1 day for standard studies
- Technical foundation:
  - Built on **AMR-Wind** (AMReX based solver, multi architecture)
  - New **complex terrain wall flux models** and **ABL parameterizations**
  - **Private** integrated workflow

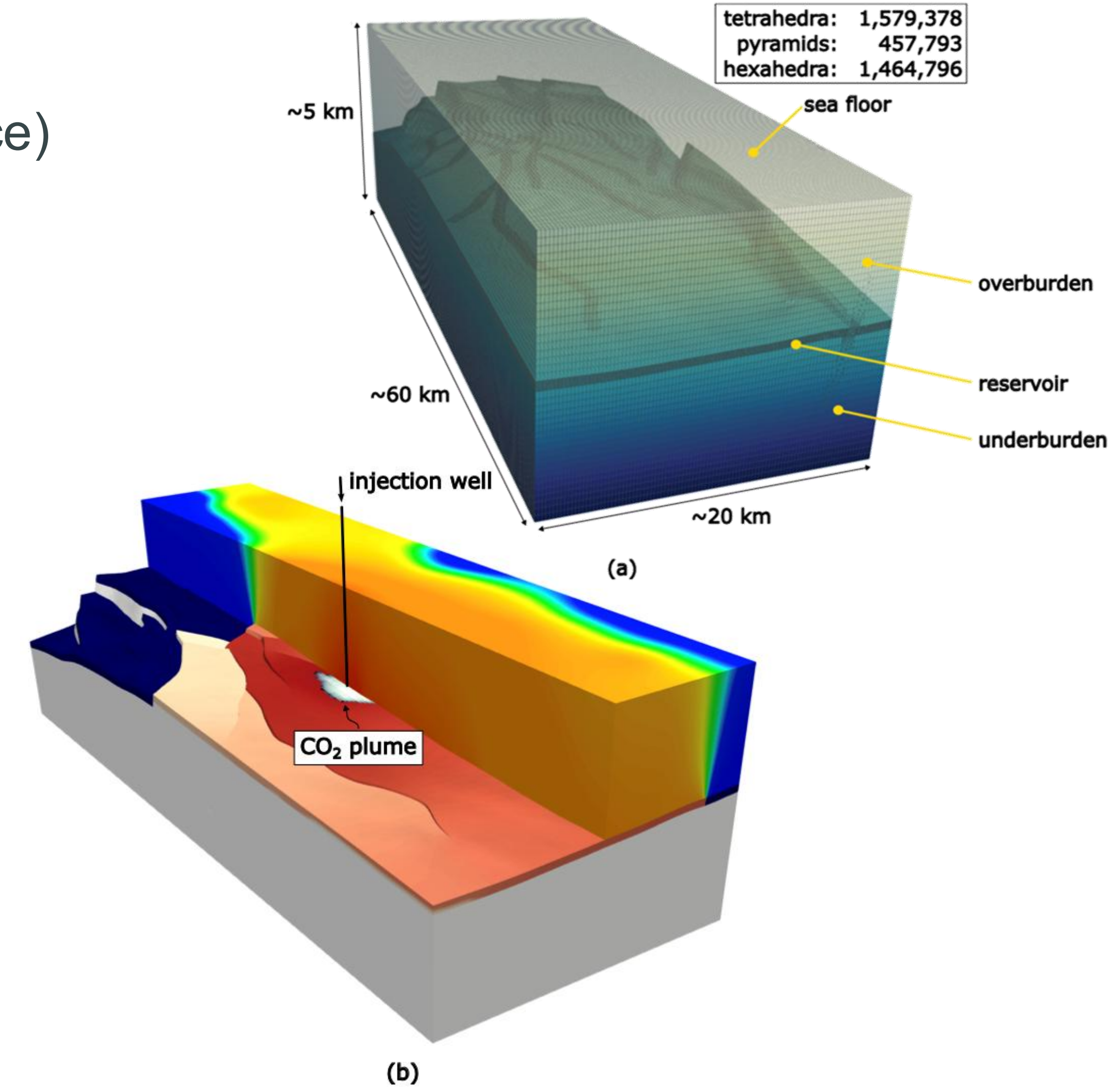


Wind speed-up map, La Reunion for sector 350deg wind at 10m.s-1 and 80m height above elevation.

GEOS: Industry–Lab Flagship for Geoscience

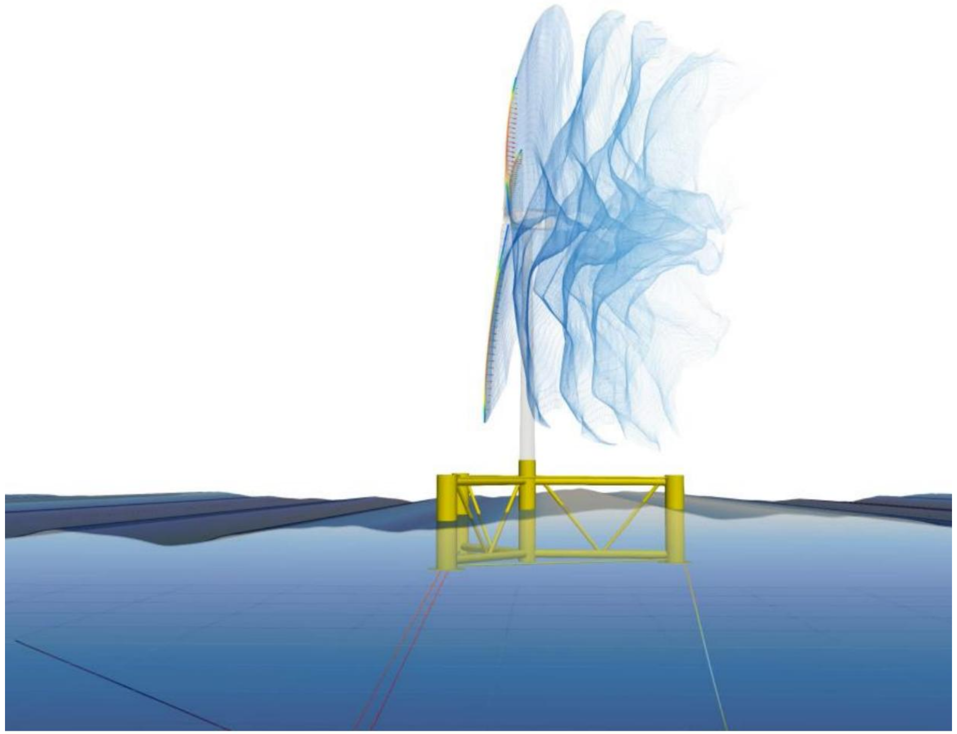
TotalEnergies

- **GEOS**** – https://www.geos.dev/
- Open geoscience platform (geomechanics, subsurface)
  - Co-developed with national labs, university and industry
- Key features:
  - Shared roadmap & validation benchmarks
  - Designed for long-term, production-quality use
- Future:
  - Governance. What model to choose?



Seahowl: Floating Offshore Wind Simulation

- **Seahowl** – https://github.com/Total-RD/seahowl
- Purpose:
  - Dynamics of floating offshore wind systems:
    - Turbine, platform, moorings, controls
- Built on:
  - **Project Chrono** multibody dynamics engine
- History:
  - Internal industrial tool → open-sourced June 2026
- Intent:
  - Reusable, robust platform; not a one-off research code



SEAHOWL simulation of Culzean pilot

From Internal Tool to Open Source: 18-Month Journey

- Internal process (~18 months):
  - IP & patent checks on models
  - Legal review of licenses (e.g. Project Chrono)
  - Security & export-control review
  - Business alignment (competitive impact)
- Stakeholders:
  - R&D, legal, cybersecurity, IT, business lines, comms
- Key issue:
  - No pre-defined process for open-sourcing scientific codes

After Opening Seahowl: Governance Questions

- Questions:
  - How to handle external contributions?
  - How to prioritize external requests vs internal roadmap?
  - Who owns docs, examples, support?
- Needed governance:
  - Maintainers, review rules, contribution guidelines
  - Release policy, versioning, deprecation rules
- Risk:
  - Becoming “just another code” in a crowded landscape
- Goal:
  - Seahowl as state of the art for floating offshore wind
  - bridging research (OpenFAST, Project Chrono) and industry engineering tools

Academia vs Industry: Different Goals for Open Source

- Academia:
  - Fast, exploratory development tied to projects & papers
  - Frequent API changes, experimental branches
  - Success = publications, novelty, citations
- Industry:
  - Stability, traceability, integration in workflows
  - Security, IP, long-term maintenance
  - Success = reliability, productivity, ROI
- Tension:
  - Same code base expected to serve conflicting objectives

Conclusion: Strategic Governance, Not Just Code

- Open source for large industrials:
  - **Benefits**: accelerated innovation, cost efficiency, ecosystem leadership
  - **Risks**: IP exposure, maintenance burden, security & compliance
- Governance levers:
  - Clear models (internal, consortium, foundation, dual licensing)
  - IP & license strategy; security and compliance processes
  - Roadmaps, Semantic versioning, SLAs*, and maintenance commitments
- Take-away:
  - Treat open source as **strategic infrastructure**, not a side project

*SLA: Service Level Agreement

AI, Models, and the Future of “Open Source” ?

- Open-sourcing AI models:
  - Physics-specific models; often trained for specific HW (e.g. NVIDIA-only)
  - Accessibility problem if weights are tied to a single platform
- Open data:
  - Need open datasets to retrain/extend models
  - New data licenses (e.g. OpenMDW, RAIL, OSAID, Creative Commons variants) add legal complexity
- Coding with LLMs:
  - GitHub Copilot / LLMs create “meta-code” (prompts, context, design notes)
  - This context should be versioned and reused like code in Git
- New license & IP questions:
  - Licenses for models, datasets,