projectEUREKA!

SOLUTION BRIEF

by Omnibond

TEACHING · RESEARCH · COMMERCIAL R&D · ENTERPRISE

One workspace for innovation.

Turn complex workflows into repeatable results.

projectEUREKA! keeps people, apps, data, and saved flows in one browser project, next to the files. Teams spend less time on setup and handoffs, and more time finishing the work. Deploy on AWS, Google Cloud, or on-prem Kubernetes.

TEACHING

More lesson time, less laptop setup

Run a course as a project. Students open Jupyter, VS Code, RStudio, Octave, or a Linux desktop in the browser, so you do not install the course tools on every laptop. Course bots or agents get Owner, Editor, or Reader on that project only.

Result: *fewer setup tickets for instructors and TAs; the same workspace from campus, home, or travel.*

RESEARCH

From prep to refinement, then run it again

Link data prep, CPU or GPU jobs, quantum circuit calls, and language-model steps into one saved flow you can run again. Mix compute types in sequence and refine until your checks pass, instead of moving files by hand between systems.

Result: *the lab keeps people, data, and flows in one project; long jobs keep running after a laptop sleeps.*

COMMERCIAL R&D

Build the analysis once, reuse it

Save a multi-stage analysis, then run it for overnight work and shift handoffs. Stages share files in project storage; CPU and GPU machines shut down when their stage ends.

Result: *less repeated setup across model work, evaluation, and simulation support; the next person picks up from the project, not from someone's laptop.*

ENTERPRISE

Project roles, not a shared login

Give each person and each bot or agent Owner, Editor, or Reader on the projects they need, instead of one broad shared account. Platform teams put AI and simulation workspaces on infrastructure they operate.

Result: *clearer who can change apps, flows, and storage; mistakes stay limited to that project's members, data, and apps.*

FROM A TASK TO SHARED OUTPUTS



Laptops stay the client. Heavy AI and simulation work runs next to project data in the environment you operate.

What backs those outcomes

- **Owner, Editor, Reader** on each project set who can change apps, flows, and storage.
- **35 ready-made apps** (19 launch + 16 flow), or any container you add.
- **Storage on the project:** home folders, shared project folders, S3 / GCS buckets, NFS or OrangeFS mounts.
- **Where it runs:** AWS, Google Cloud, or on-prem Kubernetes; machines start when needed and shut down when work ends.

Bots and agents with least privilege

Give each bot or agent only the role it needs (Owner, Editor, or Reader) on one project, with only that project's data and apps. For a course lab or commercial R&D workflow, mistakes stay limited to that project instead of a broad shared login.

TEACHING & RESEARCH

Free for non-commercially funded teaching and research. Cloud, cluster, and GPU costs stay with your institution or lab.

ENTERPRISE & COMMERCIAL

Separate licensed path for commercial engagement and production governance. Cloud and cluster costs always remain yours.

Free for non-commercially funded teaching and research

Cloud and cluster costs stay with your institution or lab. Enterprise is a separate licensed path.

projecteureka.ai
info@omnibond.com

omnibond

projectEUREKA!

SOLUTION BRIEF

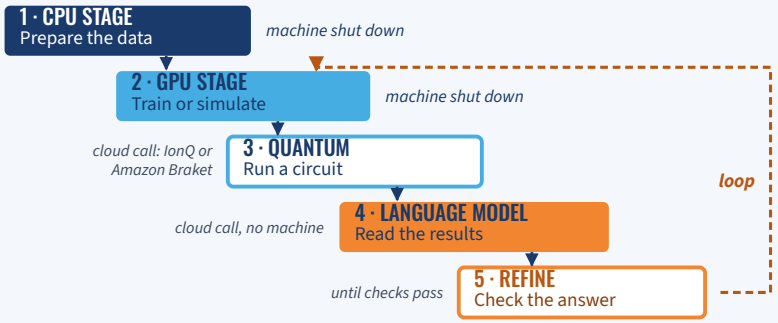
by Omnibond

HOW THE WORK GETS DONE

Connect prep, compute, and refinement in one run

A **saved flow** is a sequence of stages on a project that you can run again. Build it once, then start a run. Stages pass files forward; CPU or GPU machines shut down when their stage ends. Quantum and language-model stages are cloud calls from the control node (no dedicated machine for those steps).

ONE FLOW, MIXED STAGES



One run executes the stages in order. Each CPU or GPU machine shuts down when its stage ends.

Why teams use flows

- **Save once, run again.** Import a sample or build your own. Each run keeps the flow and prior run files.
- **Mix stage types** in one sequence: CPU, GPU, quantum, language model, file copy, and refine.
- **Refine until checks pass.** A refine stage can repeat a model call until your checks succeed.
- **Roles on the project.** Readers can run a saved flow. Editors and Owners create and edit. Owners can stop any run.
- **Monitor from the browser.** See each step waiting, running, or done without watching a laptop.

AFTER YOU START A RUN: WHAT STAYS AND WHAT GOES



Quantum and language-model stages are cloud calls (no session machine). Files you keep stay in project storage for the next run.

Stage types you can mix

- **CPU / GPU:** scripts on a sized machine; optional containers (OpenFOAM, GROMACS, BLAST, PyTorch, WRF, and more).
- **Quantum:** a circuit as a cloud call (IonQ or Amazon Braket).
- **Language model:** a cloud call that can read files from earlier stages.
- **File copy:** move files among home, project storage, buckets, or a machine.

Machines only when needed

- CPU/GPU stages start a machine, then **shut it down when the stage ends**.
- Quantum and language-model stages need **no session machine**.
- Launchers show cores, memory, GPUs, and an **estimated price**.
- Interactive apps: start, stop, or delete from the project card.

Catalog and samples

- **35 ready-made apps** at containers.projecteureka.ai.
- Download once; every project on the environment reuses it.
- Import **sample flows** (BLAST, GROMACS, OpenFOAM, quantum checks, PyTorch), then run them.
- Admins can add any SIF container your team needs.

Education and research

- **Faculty and instructors:** shared course tools without debugging every laptop.
- **Research staff and graduate students:** shared projects, data, and overnight jobs.
- **Campus labs and centers:** browser workstations next to shared storage (NFS or OrangeFS mounts).

Commercial and enterprise

- **Platform and infrastructure:** a common pattern for AI and simulation workspaces under your policies.
- **Product, ML, simulation/CAE:** shared projects, browser desktops, and data next to the codes.
- **Security-minded IT:** Owner / Editor / Reader on infrastructure you operate, including on-prem Kubernetes when policy requires it.

Use case: give bots and agents only the access they need

Limit what an automated helper can reach. Put each bot or agent on a project with only the Owner, Editor, or Reader role, plus that project's data and apps. Education teams keep a course helper inside one semester project; commercial teams keep an R&D or ops agent inside one initiative. Access stays on that project's members, data, and apps instead of a broad shared account. Architecture, catalog, and paths: **projecteureka.ai · info@omnibond.com**

Free for non-commercially funded teaching and research

Cloud and cluster costs stay with your institution or lab. Enterprise is a separate licensed path.

projecteureka.ai
info@omnibond.com

omnibond