

About CASS

Consortium for the Advancement of Scientific Software

cass.community



“Advancing scientific discovery through software stewardship.”

CASS stewards and advances the freely available scientific software libraries and tools that power DOE high-performance computing — sponsored by the Department of Energy (ASCR, NGSST, SciDAC) and the Exascale Computing Project. Participation is open; DOE sponsorship isn't required. Most of the work happens in Working Groups created as needed — this panel is one of them.

CORE MEMBER PROJECTS

FASTMath

Scalable mathematical libraries

LEADS

Learning-accelerated domain science
(scientific ML)

PESO

Fostering & delivering the scientific software
ecosystem

RAPIDS2

SciDAC institute for computer science, data &
AI

S4PST

Stewardship for programming systems &
tools

STEP

Software Tools Ecosystem Project

OFFICERS

Chair: David Bernholdt (ORNL) · **Vice-Chair:** Jim Willenbring (SNL) · **Secretary:** Elaine Raybourn (UCF, SNL-Ret.)

Software Ecosystem Working Group



Sustaining collaboration across CASS members to build a curated, AI-ready scientific software ecosystem.

MEETING CADENCE

Webinars

1st Monday, even months
12–1 pm ET

Working Meetings

1st Monday, odd months
12–1 pm ET

Teatime

3rd Thursday, monthly
4–4:30 pm ET

STAY CONNECTED



#wg-ecosystem on the CASS Slack workspace

GET INVOLVED

Join the Working Group

SIGN UP AT

groups.io/g/cass-sci-soft-ecosystem-wg

OR EMAIL

cass-sci-soft-ecosystem-wg+subscribe@groups.io

Join Teatime Sessions

SIGN UP AT

groups.io/g/cass-scisoft-genai-teatime

OR EMAIL

cass-scisoft-genai-teatime+subscribe@groups.io



SOFTWARE ECOSYSTEM WORKING GROUP

Panel on Agent Skills

in Scientific Software Ecosystems



Monday, Oct 5, 2026 · 9–10 am PT / 12–1 pm ET



Informal discussion — brief intros, then open Q&A

Relaunching the CASS working group with a new rhythm: webinars in even months, informal discussions in odd months. First up — where agent skills fit in our software ecosystems.



Moderator: Mike Heroux

PANELISTS

AA

Ann Almgren

AMReX · LBNL

CB

Cody Balos

SUNDIALS · LLNL

SB

Satish Balay

PETSc · Argonne

RB

Ross Bartlett

Trilinos · Sandia

BG

Berk Geveci

VTK / ParaView · Kitware

Panel Discussion Guide

Five threads for today's conversation — jump in wherever, no need to go in order, no need to address them all.

1



What have you tried?

Agent skills or AI-assisted tooling explored so far — in development, testing, documentation, or user support.

2



Where are you headed?

The direction each project is currently thinking about — what's next, even if it's not yet a committed plan.

3



Biggest opportunity?

Where the most promising near-term payoff for agent skills shows up in each ecosystem.

4



Biggest open question?

Risks, concerns, or open questions about bringing agent-based tools into long-lived scientific software.

5



What would help?

What each project would like from this group — or the broader CASS/PESO community — going forward.



SOFTWARE ECOSYSTEM WORKING GROUP

Panel on Agent Skills

in Scientific Software Ecosystems



Monday, Oct 5, 2026 · 9–10 am PT / 12–1 pm ET



Informal discussion — brief intros, then open Q&A

Relaunching the CASS working group with a new rhythm: webinars in even months, informal discussions in odd months. First up — where agent skills fit in our software ecosystems.

PANELISTS

AA

Ann Almgren

AMReX · LBNL

CB

Cody Balos

SUNDIALS · LLNL

SB

Satish Balay

PETSc · Argonne

RB

Ross Bartlett

Trilinos · Sandia

BG

Berk Geveci

VTK / ParaView · Kitware

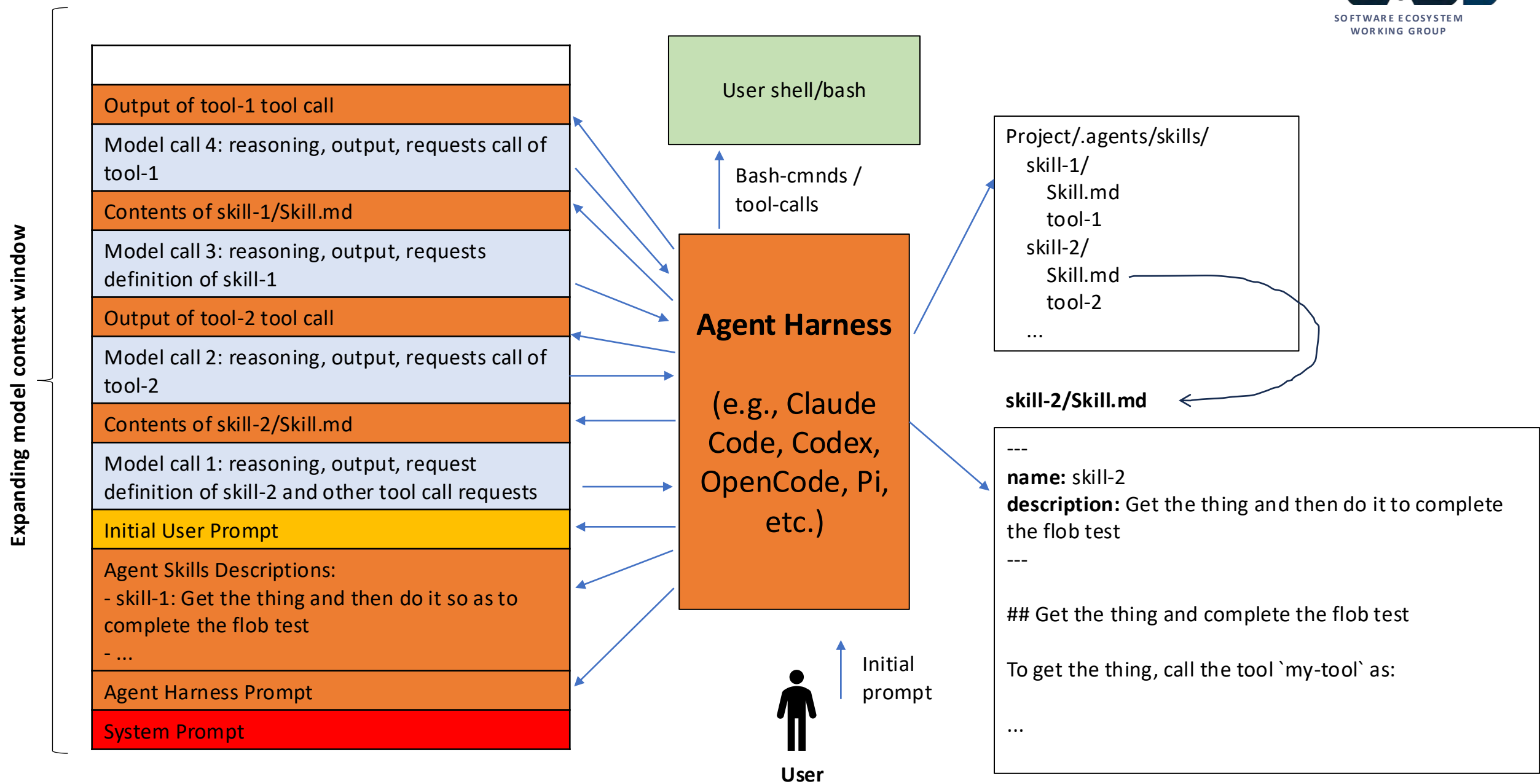


Moderator: Mike Heroux

What are Agent Skills?

- Means to inform the LLM about a set of additional context and tool definitions for different tasks without injecting all of it into the context window all at once.
- Allows the LLM to progressively ask for more details about defined skills and tools as they are needed.
- Additional “skills”(context and tool descriptions) can be progressively disclosed as needed during the agent harness loop.
- Allows for very efficient management of the model context window for maximum speed and intelligence and avoiding more frequent compaction of the context window.

Agent + Model Context Management and Agent Skills



Issues with Agent Skills?

- **How to you get the LLM to use your Agent Skills?**
 - Model may not call a skill when it should
 - May need to explicitly tell the agent to use 'my-skill'
- **How do you test your Agent Skills?**
 - Custom eval suites?
 - <https://developers.openai.com/blog/eval-skills>
 - <https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents>
- **How do you distribute Agent Skills or get Agent Skills?**
 - How to avoid prompt injection attacks as corrupted agent skills?
 - <https://snyk.io/blog/toxicskills-malicious-ai-agent-skills-clawhub>
 - Careful version control, source verification, etc.?
 - Treat Agent Skills like high-consequence software?



SOFTWARE ECOSYSTEM WORKING GROUP

Panel on Agent Skills

in Scientific Software Ecosystems



Monday, Oct 5, 2026 · 9–10 am PT / 12–1 pm ET



Informal discussion — brief intros, then open Q&A

Relaunching the CASS working group with a new rhythm: webinars in even months, informal discussions in odd months. First up — where agent skills fit in our software ecosystems.



Moderator: Mike Heroux

PANELISTS

AA

Ann Almgren

AMReX · LBNL

CB

Cody Balos

SUNDIALS · LLNL

SB

Satish Balay

PETSc · Argonne

RB

Ross Bartlett

Trilinos · Sandia

BG

Berk Geveci

VTK / ParaView · Kitware

AMReX (and AMReX-based applications): What have we tried?

The most “hands-on” types of activities have been

- Designing new code
- Writing new code
- Debugging – finding and fixing
- Developing tests
- Reviewing PRs
- Identifying new (github) issues (to be resolved in PRs later)
- Documentation generation and fact checking

AMReX (and AMReX-based applications): What have we tried?

The less “hands-on” types of activities have been

- Discussion of new features – potentially starting from very conceptual – working through issues, edge cases, implementation details
- Exploration of “literature” – what is done in other codes or papers?
- Exploration of ideas – e.g. why would a particular formulation be more unstable under certain circumstances?

AMReX (and AMReX-based applications): Where are we headed?

In the short term we are still paying off technical debt

- finding bugs that have been there for years,
- implementing features we've been "meaning to but didn't have the time"
- extending code coverage, testing ...

AMReX (and AMReX-based applications): Biggest opportunity?

What used to take weeks to months can now be done, literally, in hours to days.

We can start to ask ourselves (as developers)

- What would I do if implementation was easy/free/painless?
- But also – does anyone still need my software if it is now so easy to build their own?

We can start to ask ourselves (as users / researchers)

- What ideas would I pursue that I don't pursue now because it would take too long / would be too tedious to explore?"

AMReX (and AMReX-based applications): What would help?



An acknowledgement that this is NOT “business as usual”

AMReX (and AMReX-based applications): Biggest open questions?

- First: What happens when the limitless becomes limited again?
- Then: What happens when we're no longer in the loop?



SOFTWARE ECOSYSTEM WORKING GROUP

Panel on Agent Skills

in Scientific Software Ecosystems



Monday, Oct 5, 2026 · 9–10 am PT / 12–1 pm ET



Informal discussion — brief intros, then open Q&A

Relaunching the CASS working group with a new rhythm: webinars in even months, informal discussions in odd months. First up — where agent skills fit in our software ecosystems.



Moderator: Mike Heroux

PANELISTS

AA

Ann Almgren

AMReX · LBNL

CB

Cody Balos

SUNDIALS · LLNL

SB

Satish Balay

PETSc · Argonne

RB

Ross Bartlett

Trilinos · Sandia

BG

Berk Geveci

VTK / ParaView · Kitware



SOFTWARE ECOSYSTEM WORKING GROUP

Panel on Agent Skills

in Scientific Software Ecosystems



Monday, Oct 5, 2026 · 9–10 am PT / 12–1 pm ET



Informal discussion — brief intros, then open Q&A

Relaunching the CASS working group with a new rhythm: webinars in even months, informal discussions in odd months. First up — where agent skills fit in our software ecosystems.



Moderator: Mike Heroux

PANELISTS

AA

Ann Almgren

AMReX · LBNL

CB

Cody Balos

SUNDIALS · LLNL

SB

Satish Balay

PETSc · Argonne

RB

Ross Bartlett

Trilinos · Sandia

BG

Berk Geveci

VTK / ParaView · Kitware

PETSc — what we have tried

Three review skills, four MCP servers, and a six-skill simulation pipeline.

Three review skills

- **/review-branch** — reviews a local Git branch against **main** or **release** branches. Also usable via **make branch-review**. Prints the review report to the terminal. Useful for authors to review and fix their changes before a merge request is created.
- **/review-mr** — reviews the merge request in GitLab CI. Publishes **ai-review.html**, which authors and reviewers can check, then iterate on the issues it found.
- **/review-mr-post** — the same review, but it posts inline comments with apply-able suggested code fixes directly on the MR page. Only findings with a concrete fix are posted.

Four MCP servers

- **PETSc compile-run** (:8080) — builds and runs PETSc programs on the user's behalf.
- **PETSc documentation** (:8081) — serves the PETSc documentation into the connected tool.
- **PETSc RAG** (:8082) — retrieval-augmented context for both users and developers.
- **PETSc Claude code generator** (:8083) — creates and tests PETSc example programs.
- **Hosted at mcp.petsc-ai.org**, or self-host from **[petsc/petsc_mcp_servers](#)**. Experimental; configured today for Claude, VS Code, Gemini CLI and Warp.

Six pde-sim skills

- **orchestration** — drives the pipeline, coordinating four specialist subagents from modeling to analysis.
- **pde-formulation** — well-posed models from a physical description: classification, boundary and initial conditions, scaling.
- **numerical-methods** — FD/FVM/FEM, time integration and stability, convergence theory, manufactured solutions.
- **petsc-solvers** — composing KSP, PC (GAMG, fieldsplit, ASM), SNES, TS and the DM layer.
- **petsc-codegen** — DM setup, residual and Jacobian assembly, the options database, MPI and GPU execution.
- **pde-visualization** — VTK and ParaView: isosurfaces, streamlines, colormaps, parallel formats.
- **Early and experimental** — opt-in; nothing loads until **/pde-sim** runs.

Merge requests, and what has improved

How the review sits in the merge-request cycle, and the classes of defect it has turned up.

Merge request workflow

- **Several models review the same merge request** — Opus and Sonnet through Claude Code, Nemotron and Muse (spark) through opencode.
- **LLMs help clean up the commit series before merge** — rewritten messages and a tidied history, which makes the human review easier as well.
- **The skills live in the PETSc repository**, so an author can run them while the merge request is still in progress.
- **One skill set, several tools.** `claude`, `opencode` and `codex` are the ones in regular use.

Major improvements

- **Documentation inconsistencies the models flagged are being fixed** — including gaps that surfaced only because a model got the answer wrong.
- **Missing Fortran interfaces flagged by the models are being added** — the PCASM and PCMG stubs were filled in through September.
- **Guidelines that no script checks.** Much of PETSc's style is enforced by `checkbadSource`, `petsclinter` and `clang-format`; the models flag what is left, and those are getting fixed too.

Where to find it

- **`petsc-ai.org`** — vision, progress and papers for the PETSc AI effort as a whole; `mcp.petsc-ai.org` is one piece of it.
- **PETScAgent-Bench** — a framework for authoring PETSc and TAO benchmarks and scoring LLM-generated code. Third in the AgentX-AgentBeats code-generation track, February 2026.
github.com/petsc/petscagent-bench
- **Merge requests** — the review jobs run on these, and each attaches its `ai-review.html` report.
gitlab.com/petsc/petsc/-/merge_requests



SOFTWARE ECOSYSTEM WORKING GROUP

Panel on Agent Skills

in Scientific Software Ecosystems



Monday, Oct 5, 2026 · 9–10 am PT / 12–1 pm ET



Informal discussion — brief intros, then open Q&A

Relaunching the CASS working group with a new rhythm: webinars in even months, informal discussions in odd months. First up — where agent skills fit in our software ecosystems.

PANELISTS

AA

Ann Almgren

AMReX · LBNL

CB

Cody Balos

SUNDIALS · LLNL

SB

Satish Balay

PETSc · Argonne

RB

Ross Bartlett

Trilinos · Sandia

BG

Berk Geveci

VTK / ParaView · Kitware



Moderator: Mike Heroux

VTK/ParaView : What have we tried?

- Agent assisted code development – we want the whole team to do this
- Developing tests
- CI/dashboard management
- Issue tracking
- Documentation generation
- Web site generation

VTK/ParaView : What have we tried?

About This Leaderboard

Evaluation Setup: All agents are evaluated using Claude Opus 4.6 or GPT-5.2 as LLM judges for most task suites. Topology Visualization uses rule-based evaluation and does not require an LLM judge. Scores and completion rates are reported as mean $\pm$ standard deviation across three repeated trials.

Metrics: Overall Score (quality of outputs), Completion Rate (% of tasks completed), pass@k (success in at least one of first k trials), pass^k (success in all k trials).

↑ Higher is better ↓ Lower is better ● Best score in category

Claude Opus 4.6 Judge

GPT-5.2 Judge

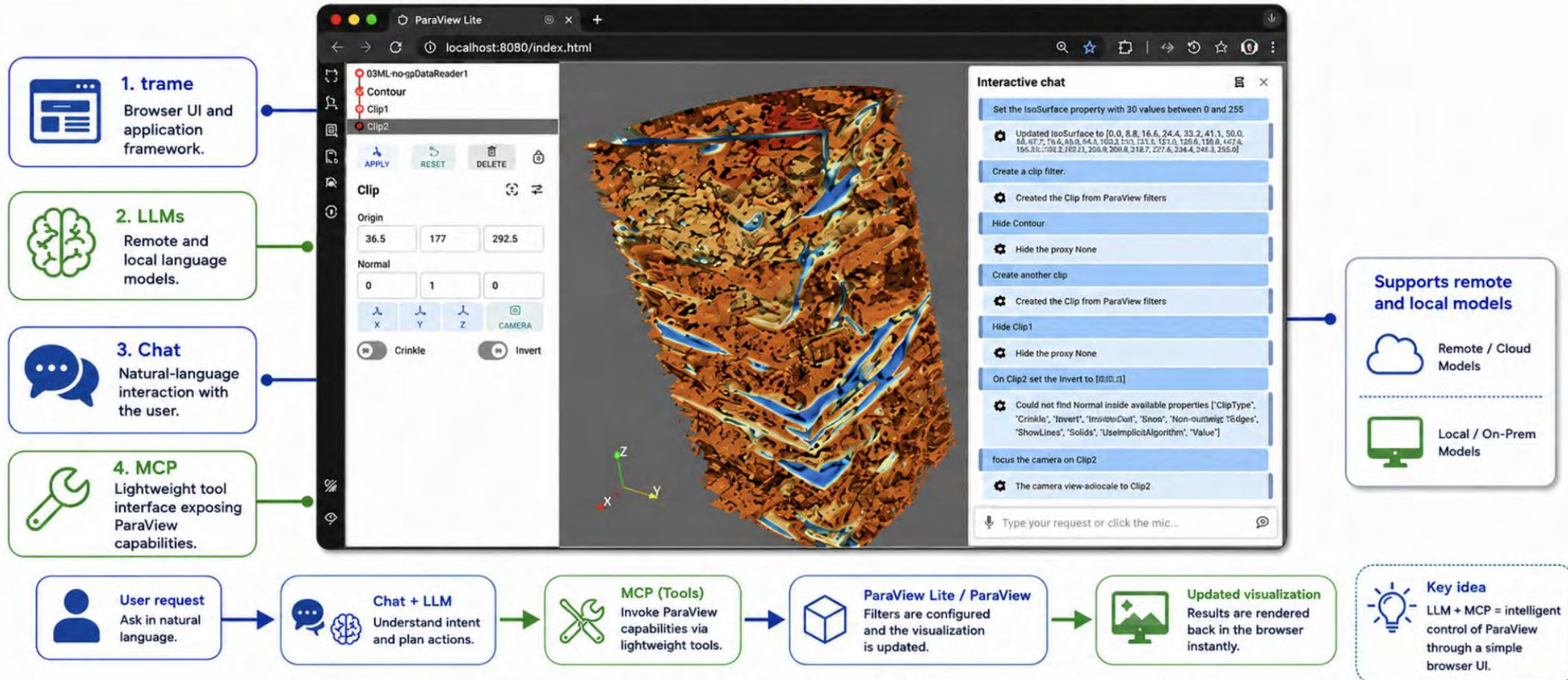
ParaView Visualization

Rank	Agent + Model	Overall Score ↑	Completion Rate ↑	pass@1 ↑	pass@3 ↑	pass^3 ↑
1	Claude Code Claude-Sonnet-4.5	62.57 $\pm$ 0.51	97.92 $\pm$ 2.08	0.736	0.917	0.500
2	Codex GPT-5.2	60.17 $\pm$ 1.43	95.14 $\pm$ 2.41	0.771	0.938	0.521
3	ChatVis Claude-Sonnet-4.5	37.37 $\pm$ 3.02	54.17 $\pm$ 3.61	0.458	0.562	0.312
4	ChatVis GPT-5.2	30.77 $\pm$ 1.10	44.44 $\pm$ 1.20	0.382	0.604	0.167
5	ParaView-MCP Claude-Sonnet-4.5	26.43 $\pm$ 8.80	53.47 $\pm$ 15.07	0.257	0.417	0.146
6	ParaView-MCP GPT-5.2	24.63 $\pm$ 3.72	46.53 $\pm$ 6.36	0.236	0.292	0.188

VTK/ParaView : What have we tried?

trame-llm

Browser-based AI control of ParaView through trame, chat, MCP, and local or remote LLMs



A lightweight trame application where an LLM uses chat and MCP tools to manipulate ParaView; with just a few lines of MCP code, nearly all of ParaView can be exposed to the UI.

VTK/ParaView : What have we tried?

vtk-prompt

Prompt-driven VTK code generation across command-line and web interfaces

1

 Describe the visualization

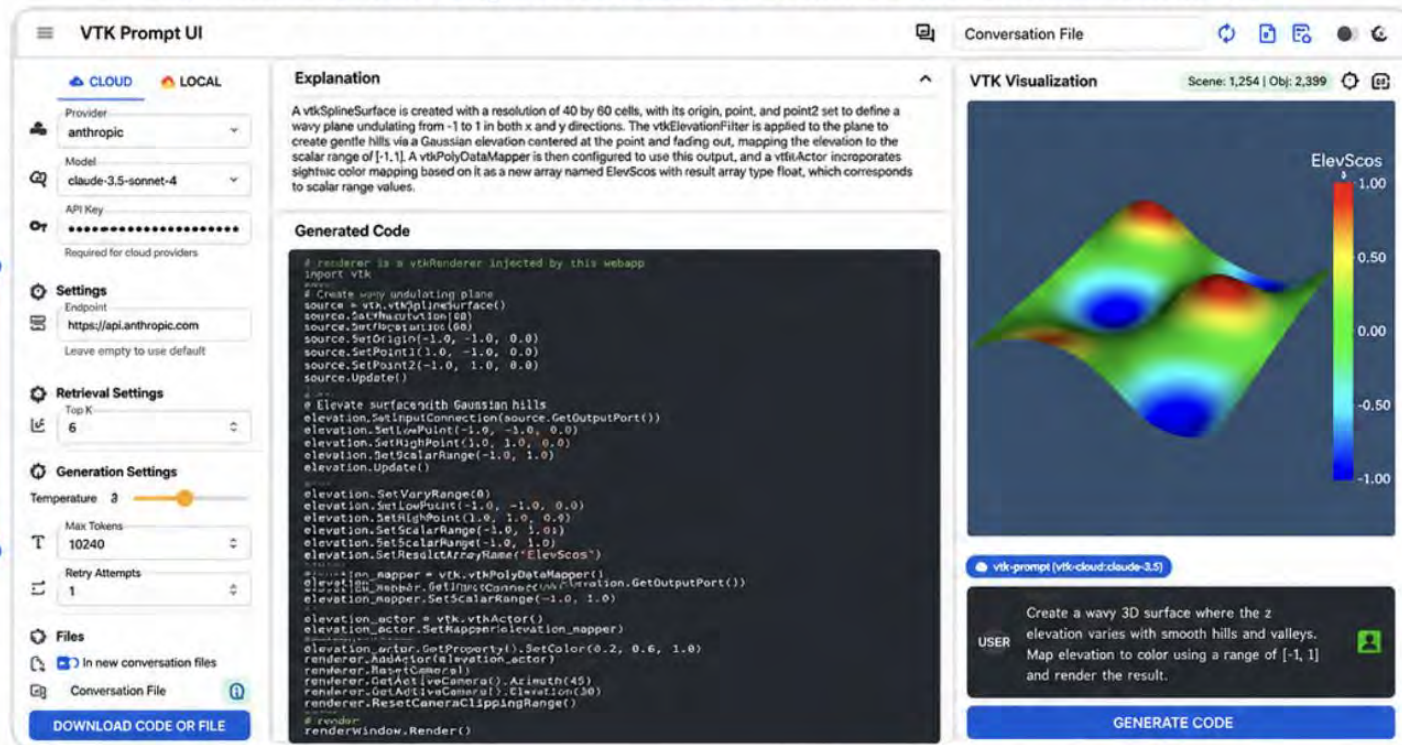
Provide your intent in natural language.

2



Generate explanation and VTK code

LLMs analyze your intent, explain the steps, and generate VTK Python code you can use.



3

 Render the result

Visualize instantly in the built-in VTK viewport.



VTK Ontology
Model VTK domain concepts



VTK DSL
Describe visualizations in a compact VTK DSL



VTK MCP
Expose VTK tools and data via MCP servers



VTK RAG
Retrieve VTK knowledge for better generation



LLMs



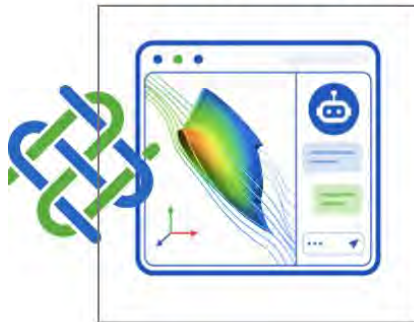
Remote LLMs
Cloud-hosted models



Local LLMs
Run models locally

VTK/ParaView : What have we tried?

AI-Native CFD Analysis Stack

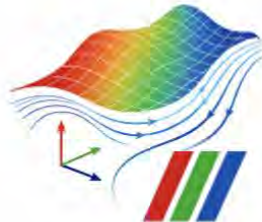


Viewer

Browser-based interaction brings visualization to the user while keeping data where it lives. Rendering and analysis execute next to the dataset; only pixels and lightweight interaction events move. Massive CFD results stay in place.

Agent

LLMs turn intent into analysis plans and orchestrate tool calls that execute next to the data. The agent maintains context, drives ParaView/simulation/inference workflows, and explains results through the lens of CFD physics.



Simulation

The CFD solver becomes part of the analysis loop. In-situ pipelines run directly on simulation memory to extract features, compute diagnostics, steer parameters, and compress data before it becomes an I/O problem. Analysis happens while the simulation is still alive.

Model

Learned models execute where the data lives. They act as surrogates for users, tools for agents, and closures for solvers—providing predictions, corrections, classifications, or reduced-order estimates every timestep without moving data off-node.

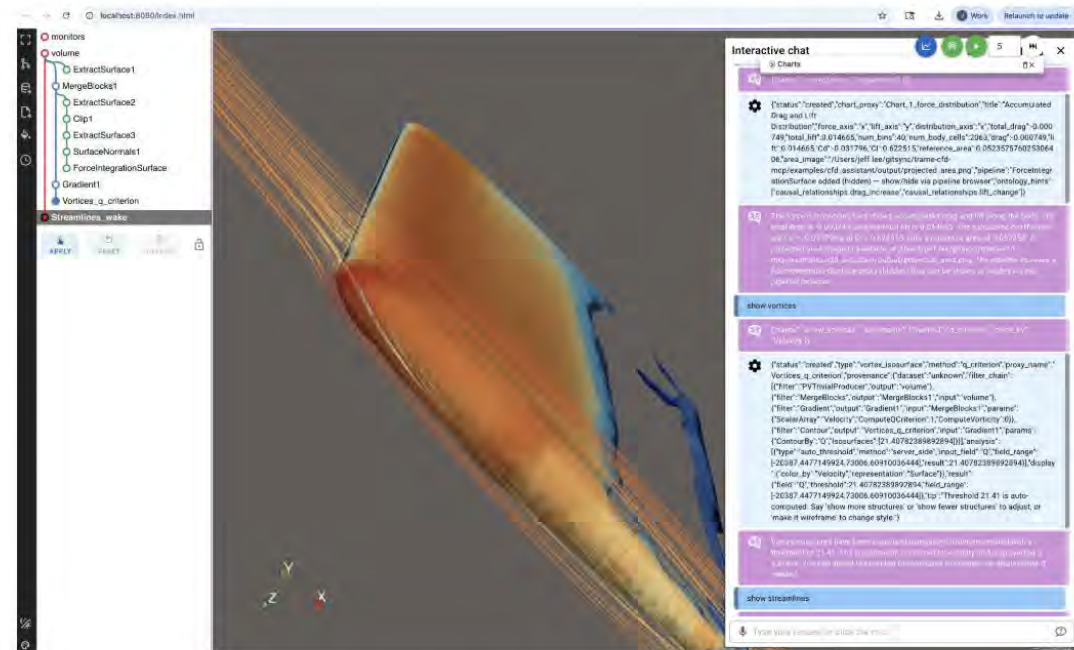


VTK/ParaView : What have we tried?

First View

Flow Assessment

- **Ask:** *What do the vorticity and velocity fields look like in critical regions?*
- The **LLM** agent turns the prompt into ParaView tool calls and pipeline operations.
- **ParaView** renders the CFD result: surface fields, streamlines, and flow features.
- **Catalyst** connects to live simulation data; frame delivers the browser-based interaction.
- **MCP** tools expose ParaView/Simulation/Model capabilities to the agent.

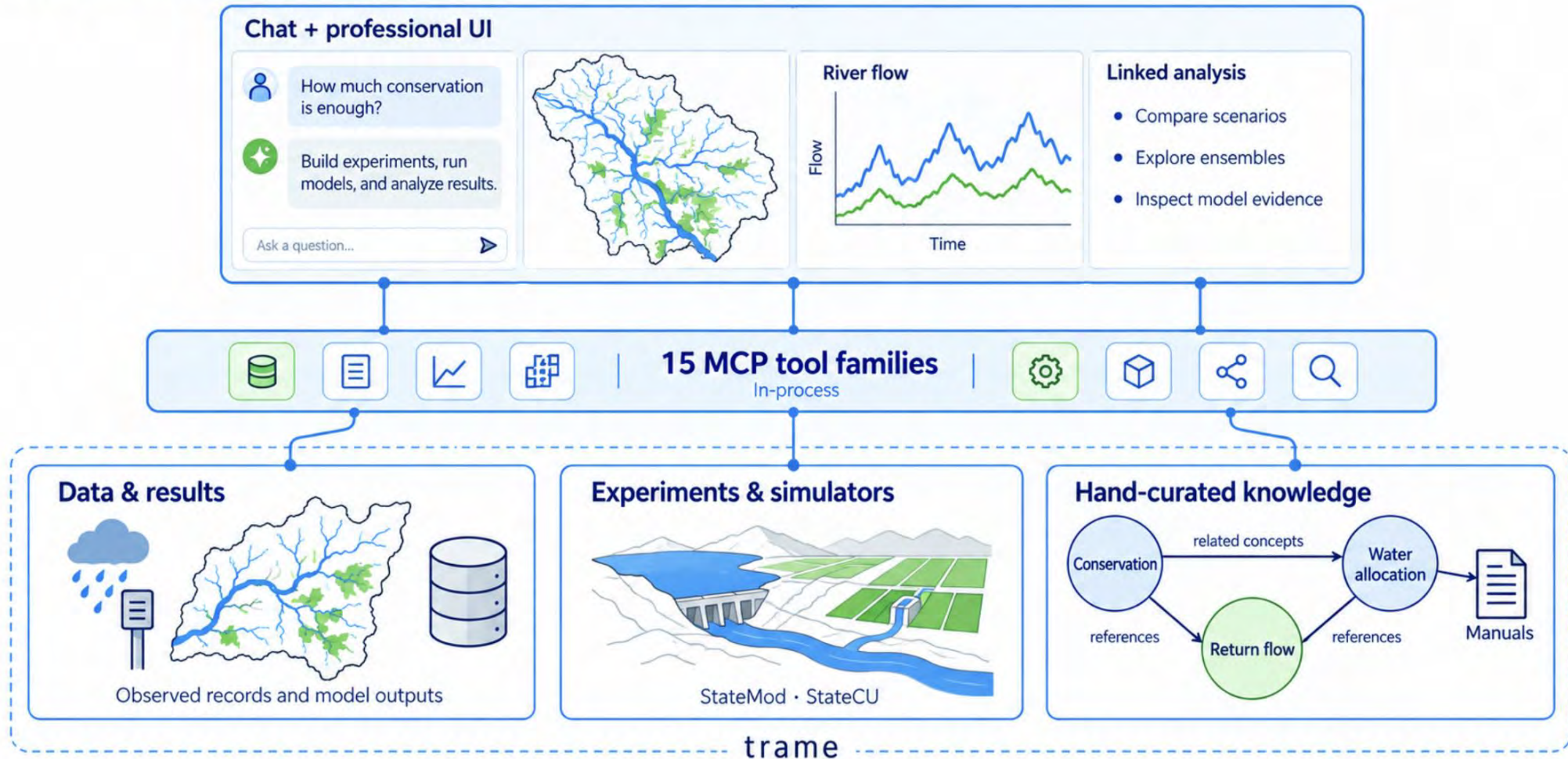


Prompt-driven CFD analysis in ParaView, where the agent builds the pipeline, renders flow features, and explains the result.

VTK/ParaView : What have we tried?

AI connected to the science

A frame application with tools and domain knowledge



VTK/ParaView : Where are you headed?

- Knowledge bases
- AI accessible documentation
- Improved APIs that are human and AI accessible

Problem solving: Domain specific vertical workflows and applications

VTK/ParaView : Biggest opportunity?

Focus on Problem Solving

AI is the big connector: Developers understand and include science better. Scientists understand computing better. Augmented teams working more closely together.

Faster iteration: Significantly decrease development cycle enables focus on the problem rather than tooling.

VTK/ParaView : Biggest risk?

Loss of expertise

VTK/ParaView : What can CASS do?

- Make tools more accessible to AI assisted use
 - Better documentation
 - Accessible source code
 - Better user guides
 - Guidance to build knowledge bases
 - Guidance for correctness
- Encourage community contributions – we are receiving more MRs, detailed issue reports thanks to AI.