

MOOSEngger — a Domain-Specific AI Agent for MOOSE Ecosystem

Mengnan Li, Zachary Prince, Alexander Lindsay, Cody Permann



U.S. DEPARTMENT
of **ENERGY**

Office of
Nuclear Energy

Motivation

Pain points

- Steep learning curve for Multiphysics workflows
- Docs/Support lag: guidance fragmented across sources
- Input authoring & debugging is slow and error-prone

What MOOSEngler changes

- Natural language → runnable MOOSE input file
- Schema validation, parameter suggestions, convergence hints
- Troubleshooting guidance + reusable knowledge base

Goal: Interactive intelligence assistant for real time model configuration and troubleshooting



How MOOSEngger Solve the Problem

User experience

- Terminal chat user interface(UI), streaming response
- Persistent session memory
- GitLab repository, documentation & unit testing

Retrieval & Parsing

- Hybrid Retrieval Argument Generation(RAG) (vector + keyword, structured database)
- Dedicated input parser interprets MOOSE .i input files preserves hierarchical [Block] structures

Flexibility

- Pluggable large language model (LLM) backends (OpenAI, HuggingFace, or Ollama)
- Built-in RAG & agent evaluations
- Extendable framework design

Agent core & tools

- LLM orchestrates multiple tools
- MOOSE-focused domain-specific toolset
- Local & remote MOOSE automatic execution

```
(moosenger-framework) [11m2@rn434323 agent_framework]$
```

Demo Case Brief

• Input File Generation

User query: Can you write a 2D heat transfer problem in MOOSE?

MOOSEngger: (2D heat transfer input file with short description...)

• Automated MOOSE Execution

User query: Can you run this input file and report me the results?

MOOSEngger: (Execute MOOSE binary with the generated input file, save outputs to workspace...)

• Refine mesh and Re-run

User query: Can you refine the mesh by a factor of 2 and re-run it?

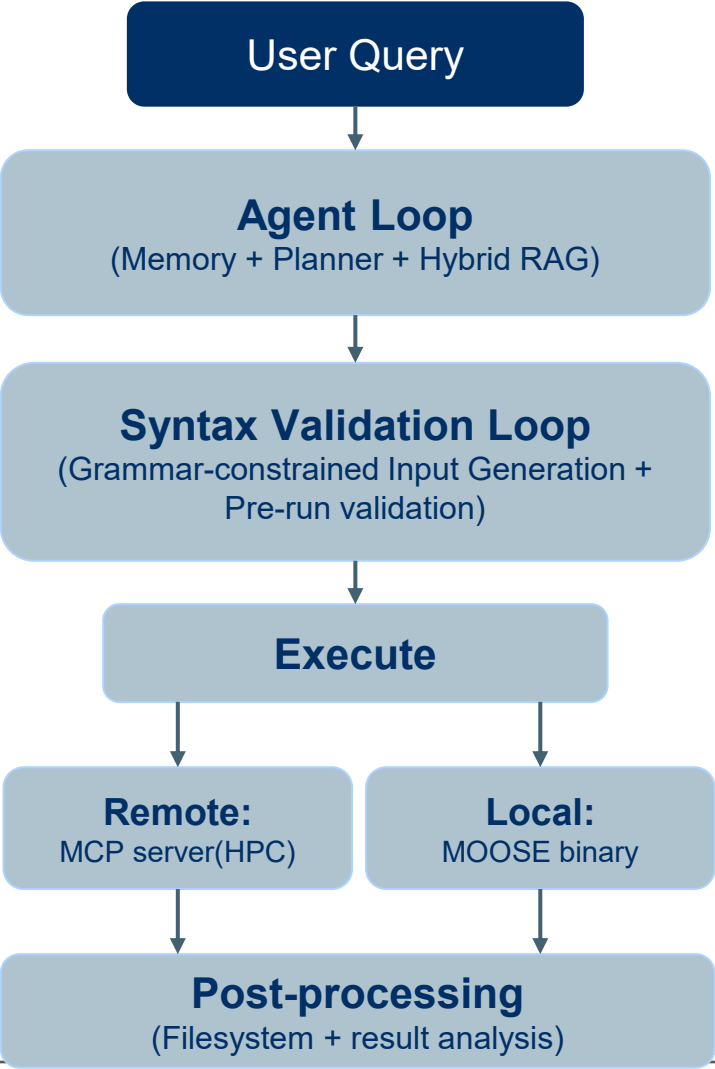
MOOSEngger: (Modify the input file and run MOOSE simulation again, report results...)

MOOSEngger Demo



U.S. DEPARTMENT
of **ENERGY** | Office of
Nuclear Energy

Technical Approach



Built-in Agent Evaluation Module

Test problem	MOOSEngerr		Baseline (ChatGPT 5.2 API)	
	Pass rate	Duration (s)	Pass rate	Duration (s)
Diffusion	23/25 (0.920)	814.6	9/25 (0.360)	265.3
Transient heat conduction	17/25 (0.680)	5484.7	0/25 (0.000)	473.2
Solid mechanics	17/25 (0.680)	4748.7	0/25 (0.000)	571.5
Porous flow	20/25 (0.800)	4713.0	1/25 (0.040)	442.5
Navier–Stokes	18/25 (0.720)	2977.1	0/25 (0.000)	706.4
Overall	95/125 (0.760)	18738.1	10/125 (0.080)	2458.9

Prompt example:

- "Create a 2D Couette flow: top wall moving, bottom stationary, periodic left/right, incompressible solve. Run and compare with linear velocity profile."
- "Build a 2D steady Stokes flow test (very low Re): drive with a constant body force in x, no-slip top/bottom, periodic in x. Run and output u(y)."



What It Enables

- This approach lowers the barrier to entry for new users of MOOSE (and similar tools), allowing those with less expertise to set up and run complex simulations more easily
- It also enables faster iteration for domain experts – tedious setup and debugging steps are automated, so engineers can spend more time on analysis and exploration instead of manual input preparation

Future Vision:

- Integrate with graphical interfaces or integrated development environments (IDEs) (e.g. NEAMS workbench), tie into high-performance computing(HPC) job schedulers to further streamline usage and include safety-aware features (e.g. human-in-the-loop checks for critical simulations)
- Serve as the framework to accelerate MOOSE-based application agent development and all together enable multi-agent workflows for complex multi-physics problems in reactor design.



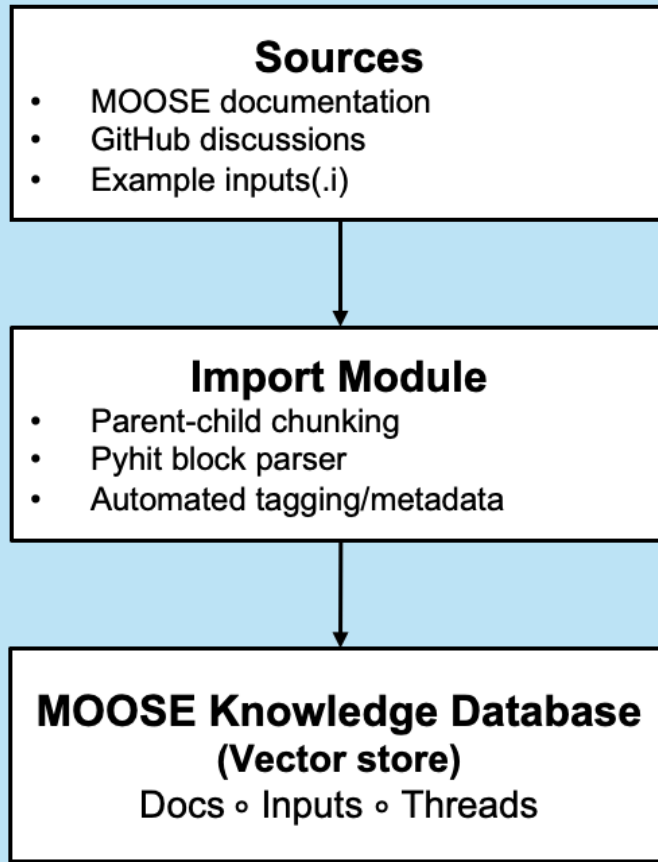


U.S. DEPARTMENT
of **ENERGY**

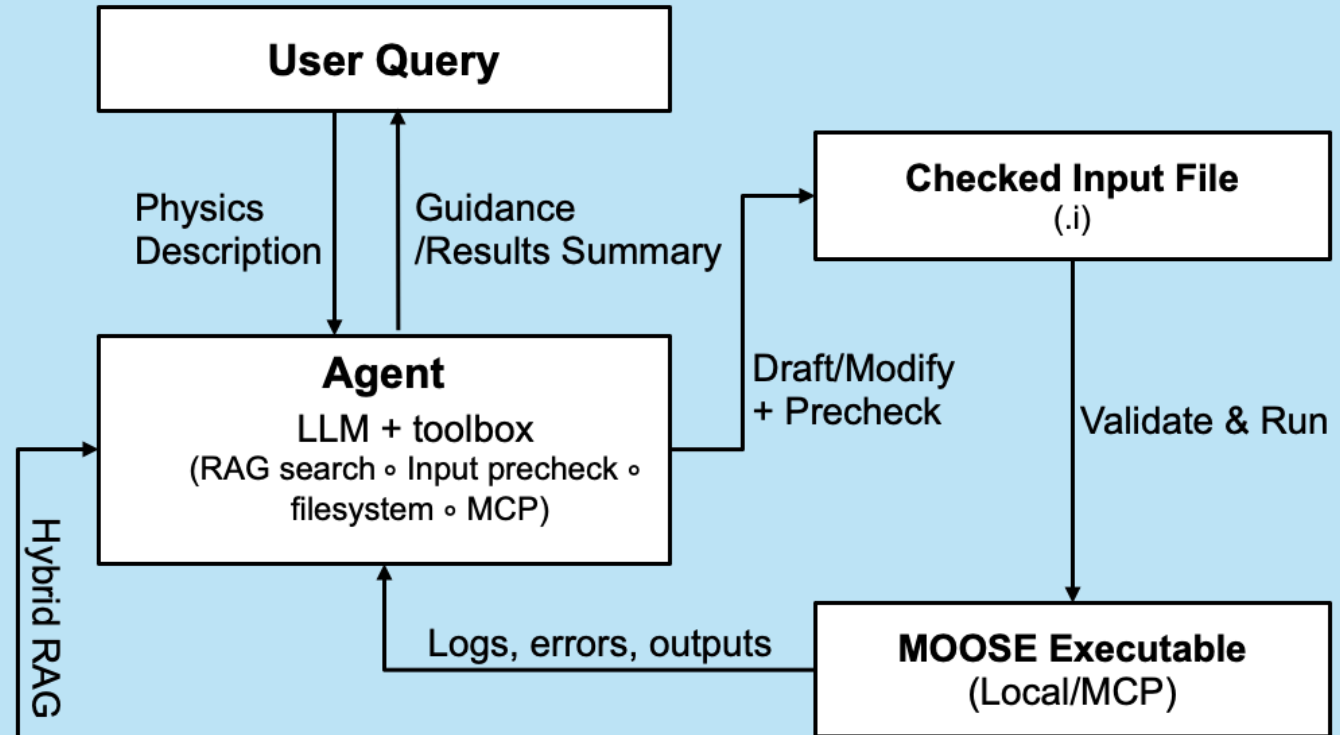
Office of
Nuclear Energy

MOOSEng Workflow

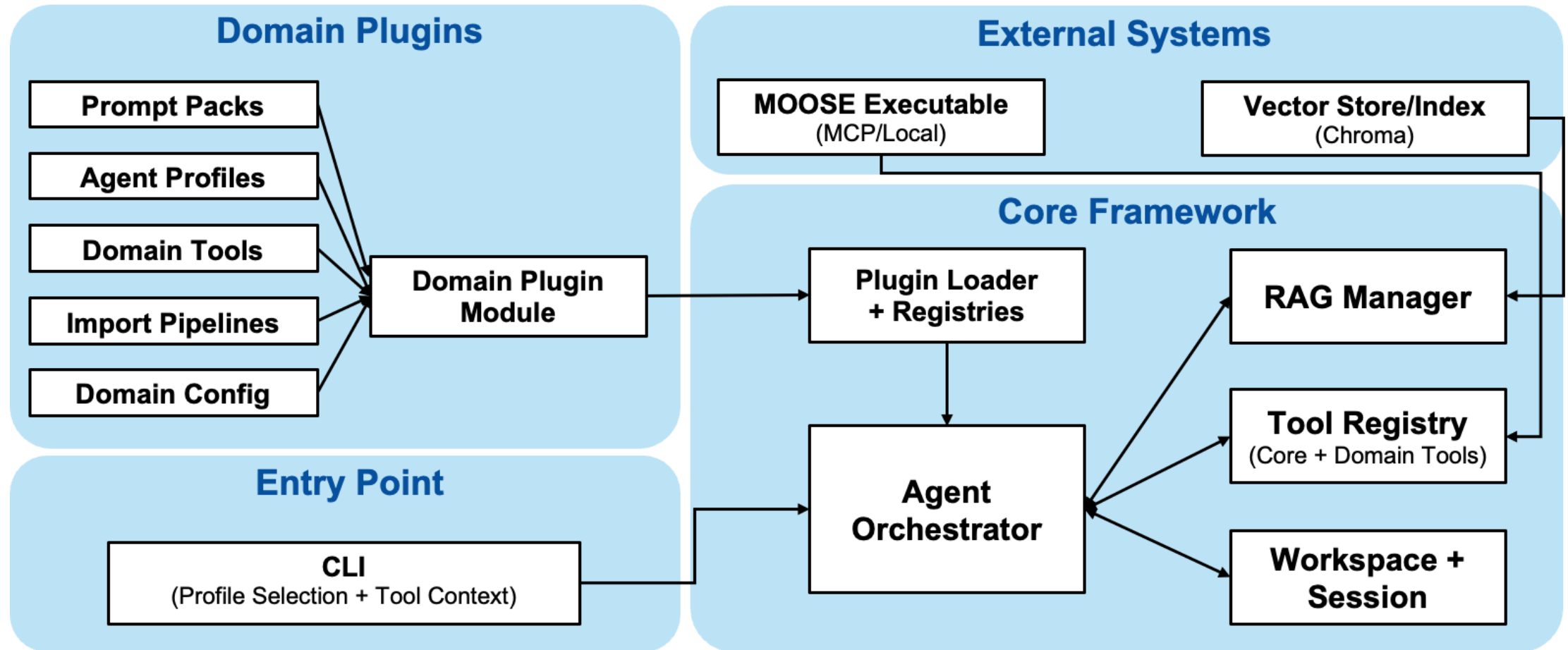
Data Ingestion



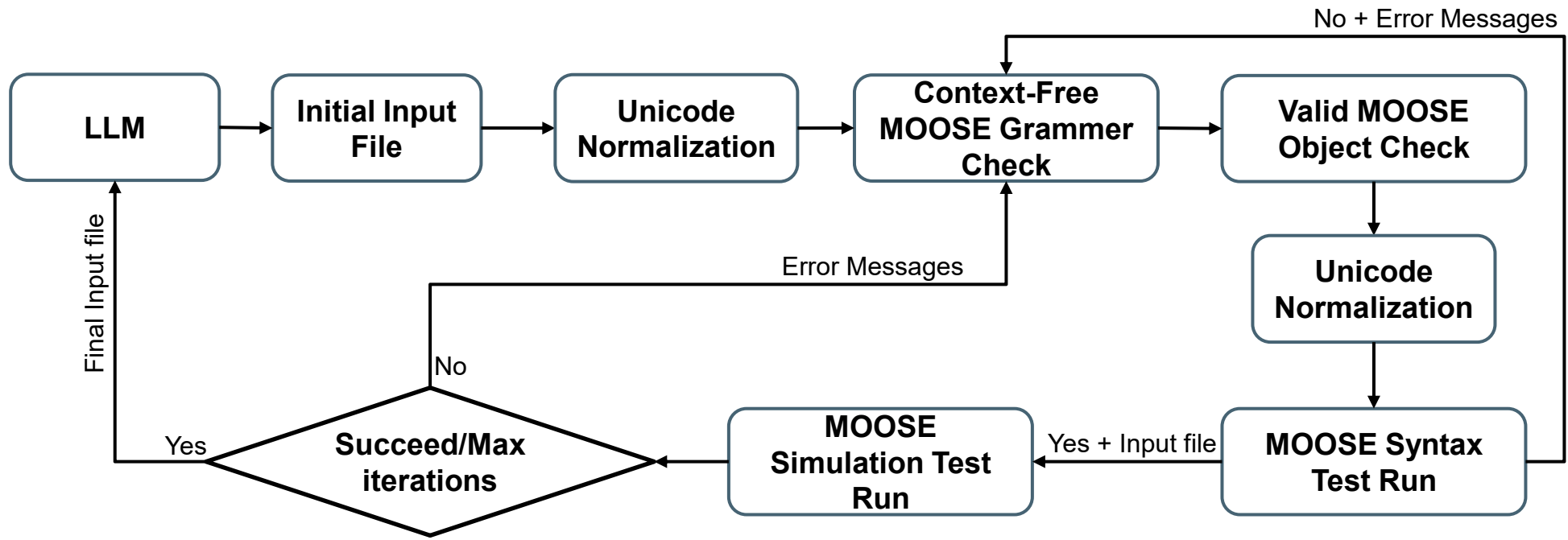
Real-Time Inference



MOOSEng System Design



MOOSE Input Precheck Tool(Composed Tool)



MOOSE Doman Specific Toolset

Unicode Normalization

Context-Free Grammer Check

Valid MOOSE Object Check

MOOSE Syntax Test Run

MOOSE Simulation Test Run

RAG Evaluation

- Integrate open-source LLM evaluation package Ragas into MOOSEng
- Support customized evaluation test by users
- moosenger eval --eval-dataset <jsonl>
- Automatically generate evaluation reports(.md or .json)

RAG Evaluation

Generated: 2025-10-22 22:45:31

Summary (means ± std)

Metric	Mean ± Std
Faithfulness	0.767 ± 0.205
Answer Relevancy	0.518 ± 0.377
Context Recall	1.000 ± 0.000

Table of Contents

- Q1. What is MOOSE and who develops it?
- Q2. Which libraries does MOOSE use for finite elements and solvers?
- Q3. Which component runs a time-dependent simulation in MOOSE?

Per-sample Results

Q1. What is MOOSE and who develops it?

• Scores

Metric	Score
Faithfulness	0.800
Answer Relevancy	0.886
Context Recall	1.000

