

E3SM AI Platform Concept Notes

Conceptual Strategy and Architecture

Version	0.1
Status	Draft for team review
Owner	Tom Vo
Audience	E3SM stakeholders, technical leads, application owners, and implementation team
Related PRD	SimBoard AI Assistant Prototype PRD

1. Purpose

This is a working notes document for discussion and alignment. It captures possible directions for a shared E3SM AI Platform, including reusable retrieval, citations, agent orchestration, controlled tool access, security boundaries, observability, and open decisions.

It may also help frame future feasibility studies and narrowly scoped prototypes, but it should not be read as a project plan, staffing commitment, production design, or committed roadmap.

2. Platform Vision

- Provide a **common AI foundation for E3SM applications** instead of building separate assistant stacks for each app.
- Use **app-specific agents** so each application can keep its own workflow, permissions, and output format.
- Use a **shared RAG knowledge service** for cited answers from E3SM documentation, source code, diagnostics, metadata, examples, and related knowledge sources.
- Use **MCP tool services** for controlled access to live external systems, such as repositories, schedulers, diagnostics services, issue trackers, and file systems.
- **Start with a narrow SimBoard AI Assistant prototype**, then identify reusable patterns that could support broader E3SM-ASSIST-style Q&A and future RRM configuration workflows.

3. Assumptions, Constraints, and Risks

Assumptions

- Exploratory concept, not a production commitment.
- Initial users: SimBoard AI Assistant and E3SM-ASSIST.
- Prototype capabilities are read-only.
- Initial sources include selected E3SM docs, code, diagnostics, metadata, and examples.
- Full platform work requires explicit FTE, ownership, review, and maintenance planning.

Constraints

- Source-based answers should include citations.
- Access must follow user roles and permissions.
- Prototype tools should be reviewed, allowlisted, and read-only.
- Shared ingestion/retrieval should avoid duplicate app-specific pipelines.

- Indexes must stay fresh enough to avoid stale citations.

Risks

- Scope may grow before staffing, ownership, security, and support are defined.
- Even prototype work may require more development capacity and specialized AI/RAG, backend, evaluation, security, and operations expertise than currently available.
- Scientific summaries or explanations may be incorrect.
- Sources may be stale, incomplete, non-authoritative, or misinterpreted.
- Permission gaps could expose restricted information.
- User-supplied LLM API keys could create security, audit, reproducibility, support, and data-governance issues.
- Applications may drift into separate custom implementations.

Mitigation: Keep prototypes read-only and narrowly scoped, use curated authoritative sources, enforce permissions before retrieval or tool use, avoid user-supplied LLM keys for shared workflows, require citations, log activity, and require explicit review before expanding toward production use.

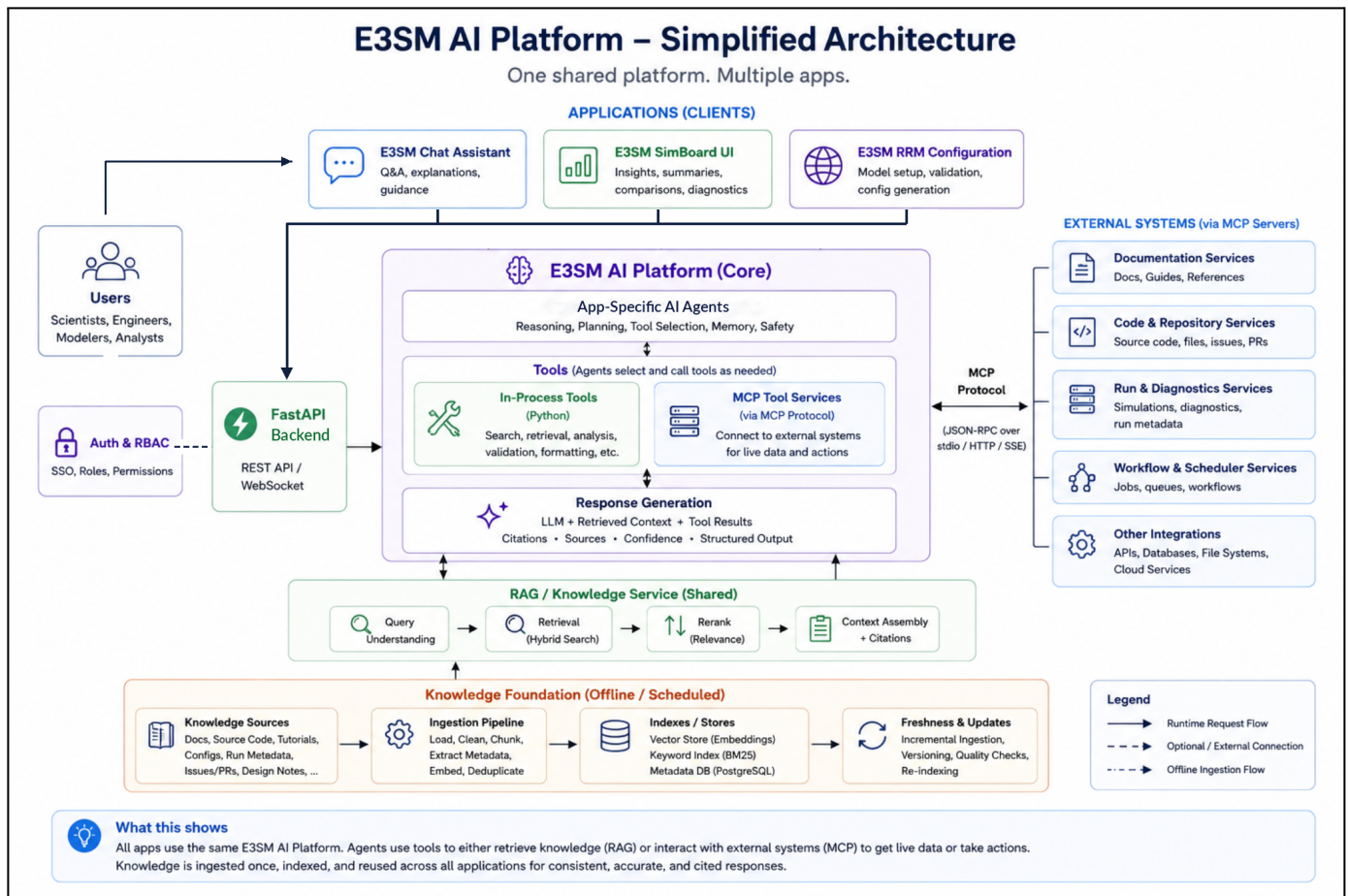
3. LLM Integration Considerations

LLM access remains an open implementation decision. A shared SimBoard or E3SM AI assistant **should not rely on individual users supplying personal API keys** as the default model. User-supplied keys could create security, audit, reproducibility, support, cost-tracking, and data-governance issues.

A production or shared prototype would need **an approved and centrally managed LLM access model**, such as an institution-approved service endpoint, funded service account, or approved hosted/local model. Until that is defined, prototype work should focus on retrieval, citations, structured metadata summaries, evaluation, and limited developer-only experimentation.

4. Conceptual Architecture

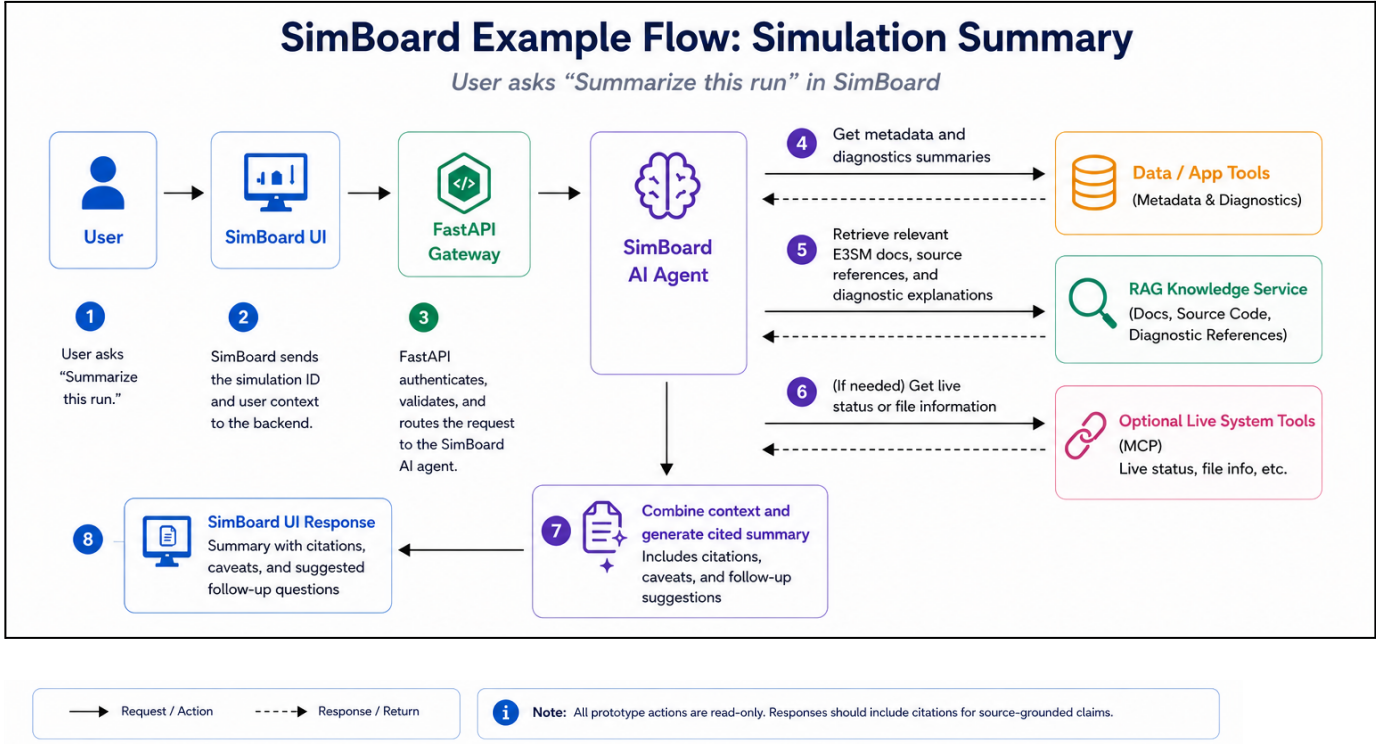
The architecture diagram illustrates a conceptual direction for how multiple E3SM applications could share common AI platform capabilities. Each application may use an app-specific agent to preserve its workflow, permissions, and response format, while common platform services could provide reusable retrieval, citation support, controlled tool access, security boundaries, and observability. Agents may use RAG to retrieve indexed E3SM knowledge and, where appropriate, MCP-style tool interfaces to access approved live external systems.



How to read the diagram:

1. Users start in the **E3SM applications** at the top.
2. Requests enter through the **FastAPI Backend**.
3. The **FastAPI Backend** routes each request to the **E3SM AI Platform Core**.
4. The platform invokes the right **app-specific AI agent**.
5. The agent selects tools based on the request and application context.
6. Tools retrieve knowledge through **RAG** or call live systems through **MCP**.
7. The agent returns a structured response with **sources**, **citations**, and confidence signals.
8. Separately, the **Knowledge Foundation** keeps indexes current through ingestion, embedding, indexing, and re-indexing.

Example AI workflow



5. Core Layers

Layer	Role	Candidate Software
Applications	User-facing apps such as SimBoard and E3SM-ASSIST	Existing E3SM app UIs
FastAPI Backend	Backend entry point for request routing, validation, auth context, and streaming or API responses.	FastAPI
App-Specific AI Agents	Orchestrators that decide which tools to call and how to structure responses for each app.	Pydantic AI
Tool Layer	Typed tools that expose retrieval, metadata lookup, diagnostics lookup, comparison, validation, and external integrations.	Pydantic AI tools
RAG / Knowledge Service	Retrieves relevant indexed E3SM knowledge and returns source-grounded context with citations.	LlamaIndex
Knowledge Foundation	Offline or scheduled ingestion pipeline that prepares docs, code, metadata, examples, and diagnostics for retrieval.	Python ingestion jobs
Indexes / Stores	Vector, keyword, and metadata stores used for fast, filtered, version-aware, permission-aware retrieval.	PostgreSQL + pgvector
MCP Tool Services	Standardized tool interfaces for live external systems and reusable integrations.	FastMCP, Atlassian Roovo MCP Server for Jira/Confluence, Slack MCP Server, GitHub MCP Server, or other approved MCP-compatible services.
External Systems	Repositories, schedulers, diagnostics services, file systems, issue trackers, databases, and APIs.	Existing E3SM systems
Cross-Cutting Services	Auth, role-based access control (RBAC), logging, tracing, evaluation, monitoring, caching, audit records, and cost tracking.	Existing E3SM/DOE HPC services + OpenTelemetry

6. RAG and MCP Responsibilities

Capability	Primary Use	Examples
RAG / Knowledge Service	Retrieve prepared knowledge from indexed sources.	Find E3SM docs, source snippets, diagnostics documentation, config examples, and metadata records.
MCP Tool Services	Call live external systems or reusable tools through a controlled interface.	Check job status, read current repository files, inspect diagnostics availability, query issue or PR status.
Agent Orchestration	Decide which tools to use and synthesize a response.	SimBoard agent summarizes a run by combining metadata, diagnostics, RAG citations, and caveats.

7. Centralized Knowledge Base Strategy

A centralized knowledge service is recommended, but it should not be one undifferentiated vector store. The platform should maintain logical indexes with shared access patterns and metadata filtering.

- **Docs index:** E3SM documentation, tutorials, design notes, and user guides.
- **Source-code index:** selected repositories, branches, commit metadata, file paths, and line references.
- **Diagnostics index:** diagnostics documentation, metric definitions, plot descriptions, and interpretation notes.
- **Simulation metadata index:** case IDs, run metadata, provenance, configuration summaries, and diagnostics summaries.
- **Configuration examples index:** examples and templates for future RRM and workflow guidance.

All retrieval should be metadata-aware, version-aware, and permission-aware.

8. Initial Reuse Targets

Application	How It Uses the Platform
SimBoard UI	Contextual, read-only simulation summaries, diagnostics explanations, run comparisons, and cited help.
E3SM Chat Assistant (E3SM-ASSIST)	Broader Q&A over E3SM documentation, source code, diagnostics, and workflows.
RRM Configuration Assistant	Guided configuration support, examples, validation tools, caveats, and reproducible output in a later phase.

9. What Is Shared vs App-Specific

Shared Platform Capability	App-Specific Behavior
Knowledge ingestion and indexing	Which sources and filters each app uses
RAG retrieval and citation service	How retrieved context is presented to users
MCP integration pattern and tool registry	Which tools each agent is allowed to call
Auth, RBAC, logging, tracing, evaluation	App-specific UI, workflows, and success metrics
Operational monitoring and usage tracking	Response format, prompts, and user-facing controls

10. Security and Governance Principles

- Apply role-based access control (RBAC) before retrieved content or tool results are returned to the agent or user.
- Start with read-only tools for the prototype.
- Allowlist MCP tools and restrict credentials by app and user role.
- Require human approval before any future write action, job submission, or configuration-changing operation.
- Log source IDs, tool calls, latency, errors, and user feedback for auditability and evaluation.

11. Possible Validation Path (Feasibility Prototype)

This candidate validation path focuses on prototype delivery and validation, not full platform production. The goal is to test whether shared E3SM AI capabilities are useful across more than one application, starting with a narrow SimBoard AI Assistant prototype.

The initial work should be anchored in a concrete SimBoard workflow. Reusable patterns for retrieval, citations, knowledge indexing, logging, evaluation, and LLM access can then be identified and tested with a second prototype consumer such as E3SM-ASSIST. A separate shared platform should be considered only after prototype evidence shows that common capabilities are worth extracting.

Refer to the [SimBoard AI Assistant Prototype PRD](#) for more information.

Stage	Focus	Candidate Activities	Decision / Outcome
1. SimBoard vertical slice	Validate one concrete AI-assisted workflow inside SimBoard.	Build a read-only simulation summary flow using existing SimBoard metadata, diagnostics context where available, caveats, citations, basic logging, and feedback.	Evidence that AI-assisted summaries are useful and feasible in a real application context.
2. Reusable pattern extraction	Identify which pieces are worth sharing beyond SimBoard.	Review retrieval, citation, indexing, logging, evaluation, LLM access, and response-structure patterns from the SimBoard prototype.	Candidate shared capabilities are identified without prematurely committing to a full platform.
3. Second-workflow validation	Test whether the same patterns work for a broader assistant workflow.	Test whether the same patterns work for a broader assistant workflow. Apply selected shared patterns to E3SM-ASSIST-style Q&A over curated documentation and selected sources, either within SimBoard or as a separate application.	Evidence that shared retrieval, citation, logging, and evaluation patterns are useful beyond the initial SimBoard summary workflow.
4. Review and decision	Decide whether the shared platform direction is justified.	Review answer quality, citation usefulness, source coverage, retrieval behavior, staffing needs, ownership, LLM access, security, and maintenance requirements.	Decision to continue, narrow, expand, revise, or pause the shared platform approach.

Future Work

Workstream	Focus	Future Outcome
RRM Configuration Assistant	Add guided configuration support, examples, validation tools, expert review, and reproducible outputs.	Higher-confidence support for regionally refined model setup.
Write-Capable MCP Tools	Add controlled write actions only after approval workflows, stronger permissions, and audit logging are defined.	Agents can safely perform approved actions in external systems.

Production Rollout	Plan security review, deployment, monitoring, support ownership, scaling, and operational maintenance.	The platform is ready for broader operational use.
--------------------	--	--

12. Open Questions / Decisions Needed

- Should E3SM-ASSIST-style broader Q&A be developed as a separate application, embedded within SimBoard, or supported through shared backend capabilities that can serve both?
- Which E3SM repositories, branches, documentation, diagnostics, and metadata sources are authoritative for the first index?
- Which sources are public, internal, restricted, or role-specific?
- What citation and metadata schema is needed for source tracking, version awareness, permissions, and freshness?
- How often should indexes be refreshed, and who owns source freshness decisions?
- Which MCP tools, if any, should be supported in the prototype?
- What remains out of scope for the prototype, especially write actions, job submissions, and configuration changes?
- Who owns evaluation datasets, scientific correctness review, and answer-quality feedback?
- What quality bar is required before expanding beyond prototype use?
- Who owns ingestion, indexing, retrieval APIs, logging, and maintenance?