

Modelator 2

Judgment Day for Manual Modeling

Stéphane Bégaudeau - Software Architect

Goals

- Integrate AI in Sirius Web based modeling tools
- Compatible with every AI model
- Provide generic modeling capabilities for AI agents
- Easy to add to **Sirius Web** apps thanks to **Obeo Enterprise for Sirius**

Principles

- Rapidly produce demonstrable use cases
- Build knowledge, best practices and reusable technologies
- Leverage multiple AI agents, each focused on a specific purpose
- Combine various agents in powerful workflows

Two Approaches

- Internal AI agent
 - Sirius Web ecosystem acts as the LLM orchestrator
- MCP API for external AI agents
 - Sirius Web ecosystem as a tool and data provider

Internal Agent

AI

assistant Apr 22, 2026, 1:55:20 PM
Hello

stephane.begaudeau@obeo.fr Apr 22, 2026, 1:56:01 PM
What can you tell me about the current project?

assistant Apr 22, 2026, 1:56:02 PM
Summary of the current project model (read-only)
High-level view
The diagram shows a collection of NamedElement model elements (mostly Classes and Interfaces) that implement application-level project operations (create, delete, duplicate, rename, upload) and GraphQL data fetchers that call those services. Key Interfaces and their implementing Classes are present and connected via implementation references.
Primary Interfaces and Operations
Interface IProjectCreationApplicationService
Operation: createProject(input: ICreateProjectInput): IPayload
Implemented by: Class ProjectCreationApplicationService
Interface IProjectDeletionApplicationService
Operation: deleteProject(input: DeleteProjectInput): IPayload
Implemented by: Class ProjectDeletionApplicationService (edge present in diagram)
Interface IProjectDuplicationApplicationService
Operation: duplicateProject(input: DuplicateProjectInput): IPayload
Implemented by: Class ProjectDuplicationApplicationService
Key Classes (selected details)
Class ProjectCreationApplicationService
Attributes: projectCreationService: IProjectCreationService, projectTemplateProviders: List<IProjectTemplateProvider>, projectMapper: IProjectMapper, messageService: IMessageService, logger: Object
Operation: createProject(input: ICreateProjectInput): IPayload
Implements: IProjectCreationApplicationService
Class ProjectDeletionApplicationService
Attributes: projectDeletionService: IProjectDeletionService, logger: Object
Operation: deleteProject(input: DeleteProjectInput): IPayload
Class ProjectDuplicationApplicationService
Attributes: exportService: IProjectExportService, projectZipContentProvider: ProjectZipContentProvider, projectCreationService: IProjectCreationService, projectMapper: IProjectMapper, projectSearchService: IProjectSearchService, projectEditingContextService: IProjectEditingContextService, messageService: IMessageService, logger: Object
Operations: duplicateProject(input: DuplicateProjectInput): IPayload, getNatures(object: Object): List<String>
GraphQL data fetchers (Classes implementing DataFetcher behavior)
MutationCreateProjectDataFetcher
Attributes include: projectCreationApplicationService: IProjectCreationApplicationService, objectMapper, capabilityEvaluator, messageService

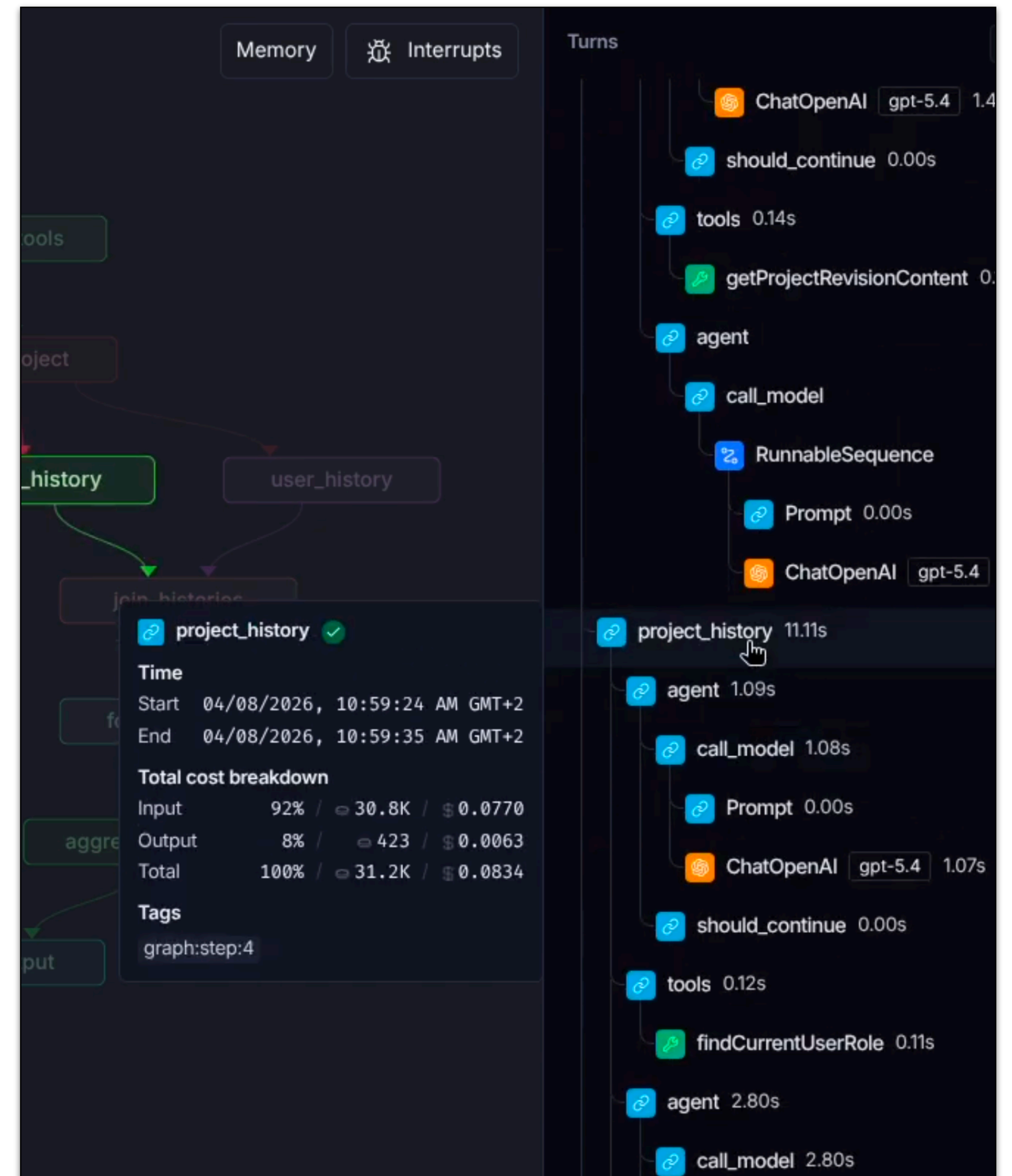
Record RenameProjectSuccessPayload
Record Component id

ctDataFetcher
or
ectUpdateApplicationService
ment): IPayload
e
Payload
e
ervice
Payload
e
Payload
e
teProvider>

Internal Agent

- Interact with the agent from your workbench
- Provide directly the chat interface
- Control prompts
- Expose specific internal tools to the LLM

Proxy Agent



Proxy Agent

- The internal agent can act as a proxy for remote agents
 - Foundational models API support
 - The remote agents are the brain (generic tools & data)
 - Sirius based application is the hand (domain based tools & data)

Context

The internal AI agent can leverage information from your workbench

- Project nature (UML, SysML, Arcadia, etc)
- Available data
- Opened representation
- Semantic selection



Explorer

- Papaya Dashboard Diagram
- Sirius Components - Domain
- Apache
- Reactor
- Sirius Components - Core
- Sirius Components - Gantt
- Sirius Components - View
- GraphQL
- Sirius Components - EMF
- Sirius Components - Papaya
- Sirius Components - Charts
- Google
- Spring Projects
- Sirius Components - Forms
- Sirius Components - Validation
- Sirius Components - Trees
- Sirius Components - Form Description Editors
- Sirius Components - Portals
- Sirius Web
- Sirius Components - Selection
- Sirius Components - Web
- Eclipse EMF
- Reactive Streams
- Faster XML
- Java
- Sirius Components - Deck
- Sirius Components - Diagrams

sirius-web class diagram



AI

assistant

Apr 22, 2026, 1:55:20 PM

Hello

Learn more

Follow these links to view the code of this Sirius Web internal test project on Github.

- Backend code
- Frontend code



React Flow

Query ▶

Customizable

The behavior of the AI agent can be configured

- System prompt
- Execution modes (Query and Edit by default)
- Query and edition tools

System Prompt

Generic & Low Level Technical Details

Domain Specific Details

Methodology Vocabulary

System Prompt

Generic & Low Level Technical Details

Obeo Enterprise for Sirius

Domain Specific Details

Obeo Enterprise for SysON

Methodology Vocabulary

Arcadia Extension for SysON (POC)

System Prompt

Generic & Low Level Technical Details

EMF

Domain Specific Details

SysMLv2

Methodology Vocabulary

Arcadia

Tools

- Can be contributed by specifiers
- Can retrieve easily the current context
 - Project
 - Selection
 - Representation

Explorer

+ ↕ ≡ 🔍 ⚙

> Capella.sysml

🔍

Create a new Representation

- LAB
Logical Architecture Blank

Open an existing Representation

Select the representation to open

← →

Details

↻ ▶

No object selected



User Experience

- Batch modification for semantic and graphical update
- Lower the quantity of tokens consumed by tools
- Tools can now stream feedback to end users

MCP API

The screenshot displays a web interface for project management. At the top, there are four buttons for creating new projects: "+ Blank Studio", "+ Flow", "+ Papaya - Blank", and "+ Blank Project". Below these is a section titled "Existing Projects" which contains a table with the following data:

Name	Owner
Papaya - Example	
Studio	
Flow	

Below the table is a terminal window showing the output of the Gemini CLI. The user asks, "What are the projects accessible on the mcp server oe4sirius?". The terminal response is:

```
✓ search_projects (oe4sirius MCP Server) {"perPage":100}
[
  {
    "id": "d8f944ac-4c29-4060-be76-2d3d8926334c",
    "name": "Papaya - Example"
  },
  {
    "id": "7555213a-9307-43a1-9f80-a488bb3d5df5",
    "name": "Studio"
  },
  {
    "id": "bb0511ae-170b-49ca-b291-a41bc851c85c",
    "name": "Flow"
  }
]
```

The terminal also includes a summary: "The projects accessible on the oe4sirius server are:" followed by a list of project names and their IDs.

MCP API

- Can use any popular agent (Codex, Gemini cli, Claude code, etc)
- Can work on a wider context (multiple organizations or projects)
- MCP tools
 - Can be contributed by specifiers
 - Can expose similar tools as internal agent tools

Joint Laboratory



Joint Laboratory

- Combine Obeo's and CEA's strength
- Provide dedicated agents to improve the modeling experience
 - Project understanding
 - Onboarding
 - Modeling Assistance
- Add new capabilities in the Sirius Web ecosystem

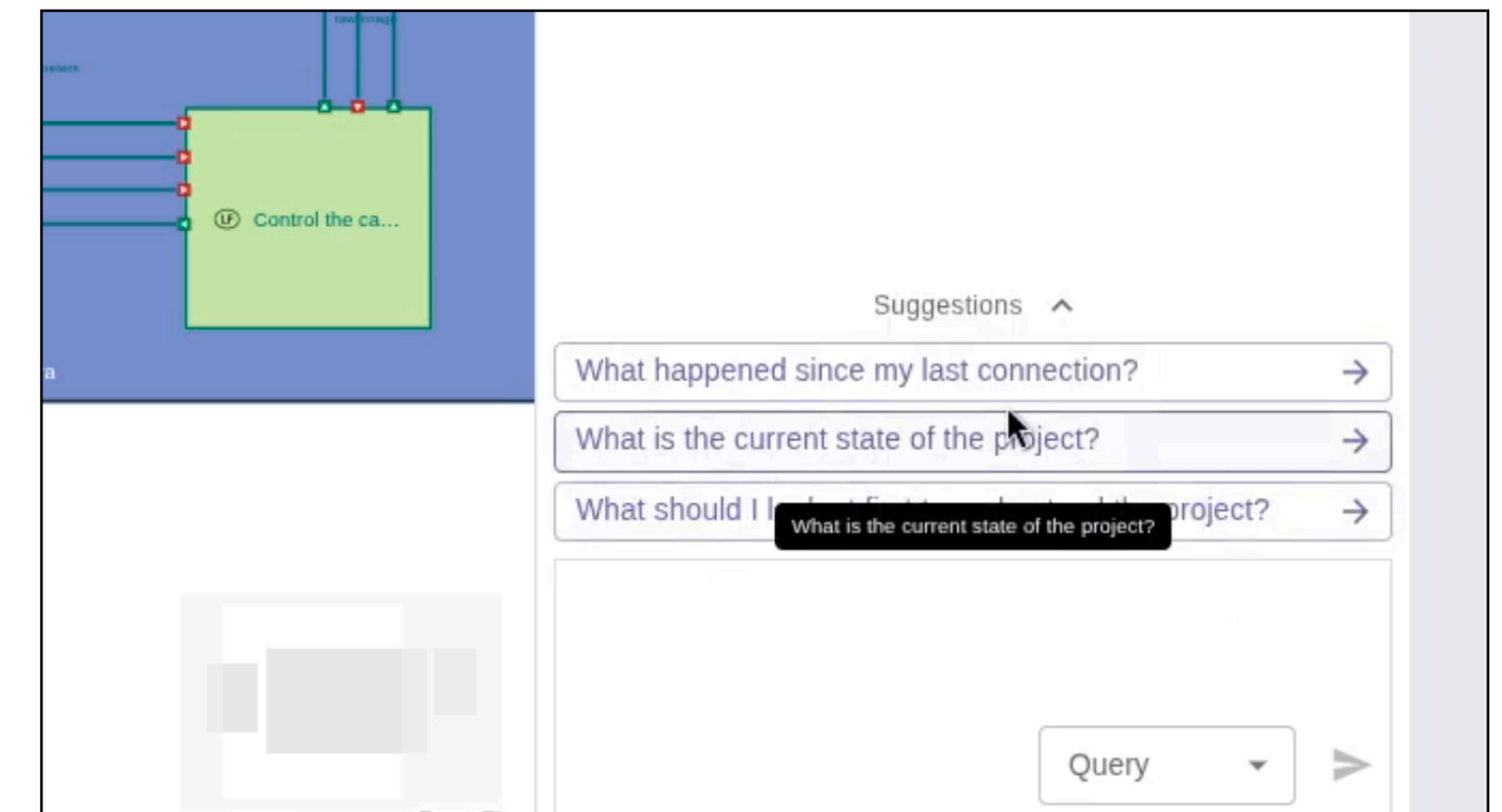
Coming Soon

User Experience

- Undo / Redo support
- Provide support for other representations
 - Form, tree, deck, gantt, table, portal

Internal Agent

- Add support for follow up questions
- Personalized conversation
 - History
 - Proficiency



Internal Agent

- Context pinning
 - Be able to view the context of the chat
 - Have a stable context while working with the workbench
- Graphical selection support

MCP API

- Better pagination and error reporting
- OAuth2 authentication support for the MCP API
- Support for MCP resources to pin a context for external agents

Thanks!

stephane.begaudeau@obeosoft.com