

AI4Europe T4.4 Meeting - Updates

Agenda

- Unified protobuf
- Federation API
- Playground-App
- Open discussion

Overview: LLM Containers in AI-Builder

Design-Studio

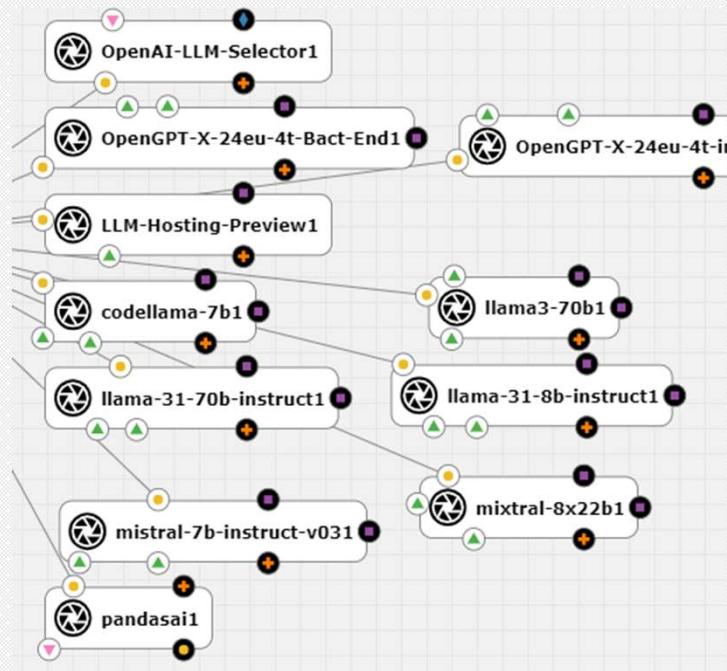
Home / Design Studio / Design-Studio

Marketplace

Solutions Models

LLM

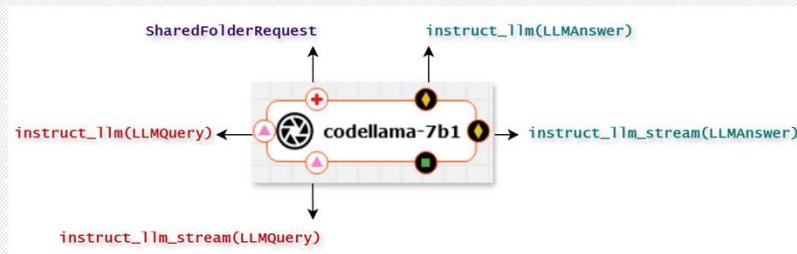
- codellama-7b (1.0.0)
- llama-31-70b-instruct (1.0.0)
- llama-31-8b-instruct (1.0.0)
- llama3-70b (1.0.0)
- llama3-8b (1.0.0)
- LLM-Hosting-Preview (1.0.0)
- mistral-7b-instruct-v031 (1.0.0)



[Link to Graphene Tutorial: LLMS in AI-Builder](#)

The Role of the Unified Protobuf Interface

- Flexible and standardized way of defining and representing connectors for different language models in the Eclipse Graphene platform
- Offers model switching ability
- Develop language, model agnostic framework



```
syntax = "proto3";

message Empty {
}

message LLMConfig {
    double temp = 1; // Can be expanded - To be discussed.
}

message PromptInput {
    string system = 1;
    string user = 2;
    string context = 3;
    string prompt = 4; // The application-specific prompt to the LLM for processing.
}

// Encapsulates a single question string, representing a user query.
message UserQuestion {
    string question = 1;
}

// Contains a q_id field, which uniquely identifies a conversation or query. This ensures continuity and state tracking across interactions
message ConvоВID {
    string q_id = 1;
}

// Bundles together the LLM config, prompt input, user question, and conversation ID into a single message.
message LLMQuery {
    LLMConfig config = 1;
    PromptInput input = 2;
    UserQuestion qa = 3;
    ConvоВID id = 4;
}

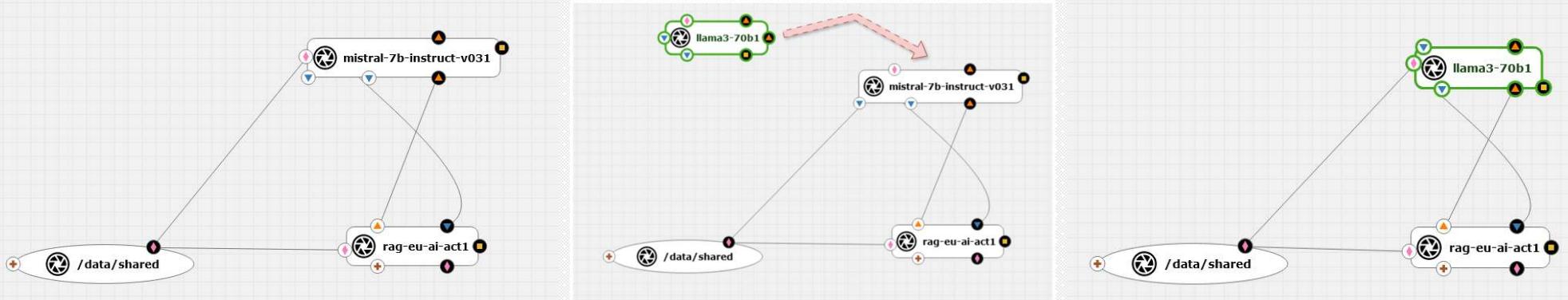
// Response from the LLM
message LLMAnswer {
    string text = 1;
    ConvоВID id = 2;
}

message Status {
    string message = 1;
}

service LLMService {
    rpc instruct_llm( LLMQuery ) returns( LLMAnswer );
    rpc instruct_llm_stream( stream LLMQuery ) returns( stream LLMAnswer );
}
```

Current Usage

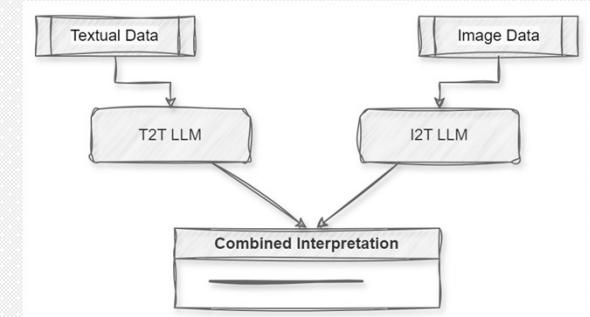
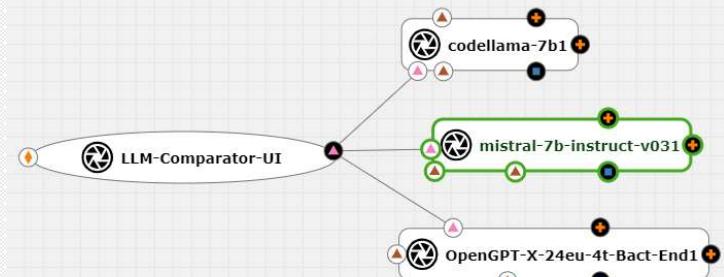
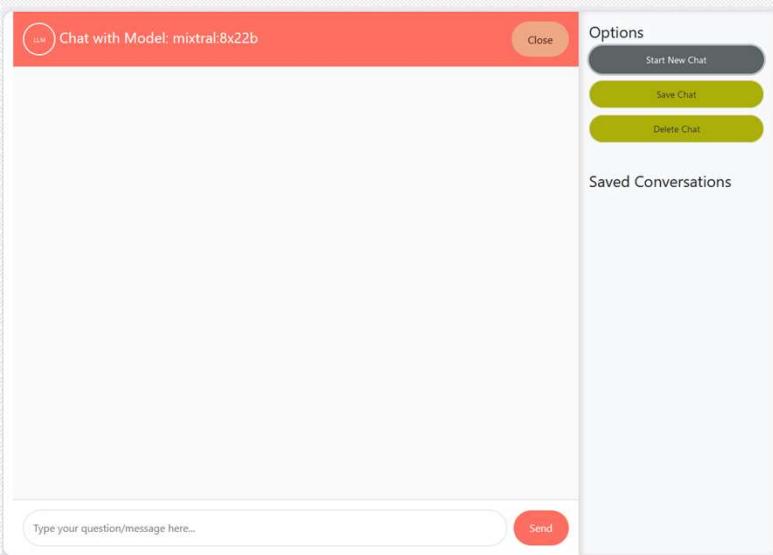
- Ability to switch between multiple LLMs with minimal efforts – Unified Protobuf Interface
- Easy to test and compare different models for various tasks.
- If a model fails or underperforms, we can alter the pipeline to a different LLM.



Building Blocks in a RAG EU_AI_Act Pipeline

Future Directions: Applications and Possibilities

- Serving as LLM Chatbots
- Creating custom workflows/pipelines with different modalities
- LLM-Comparator node



[Link to Graphene Tutorial: LLMS in AI-Builder](#)

Federation API

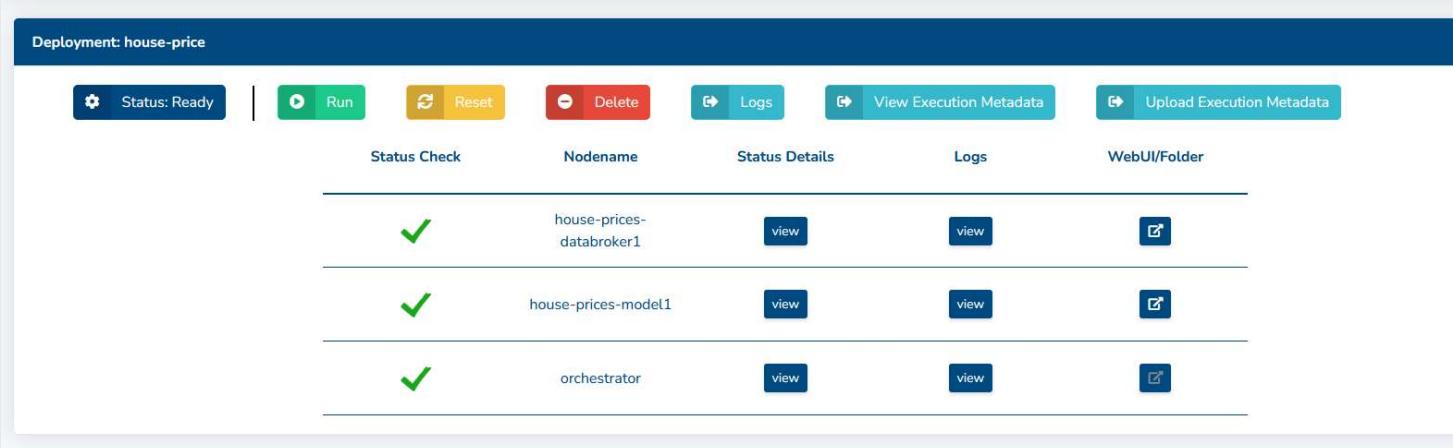
The screenshot shows the Graphene Federation Data Exchange API documentation. At the top, there's a navigation bar with the Swagger logo and the URL `/federation/v3/api-docs`. Below the header, the title "Graphene Federation Data Exchange" is displayed with an "OAS 3.0" badge. The main content area contains several sections: "Introduction" (describing the API's purpose), "Protobuf Types" (listing proto types like `ExecutionRunData`), "Solution Types" (listing toolkit types like `CP`), and "Binary Content". At the bottom, there are links to the Eclipse Graphene Federation Project website and Apache 2.0 license information.

The screenshot shows the Graphene Federation API documentation. At the top, there's a "Servers" dropdown set to `https://aiexp-dev.ai4europe.eu/federation`. The main content area is titled "Graphene Federation API" with a subtitle "Get catalogs, solutions and solution revisions with their respective artifacts and documents". Below the title, a list of API endpoints is provided:

Method	Endpoint	Description
POST	<code>/set_execution_run_data</code>	Set execution run data with origin
GET	<code>/get_solution_icon</code>	Get the solution icon as bytes
GET	<code>/get_solution_artifacts</code>	Get the list of artifacts of the specified solution
GET	<code>/get_solution</code>	Get solution
GET	<code>/get_catalog_solutions</code>	Get the list of solutions of the specified catalog as shallow solution references
GET	<code>/get_catalog_list</code>	Get the list of catalogs
GET	<code>/get_catalog</code>	Get catalog
GET	<code>/get_artifact_content</code>	Get the file content of the specified artifact as bytes

<https://aiexp-dev.ai4europe.eu/federation/swagger-ui/index.html#/>

Updates to Playground-App



The screenshot shows the playground-app interface for a deployment named "house-price". The top navigation bar includes links for "Status: Ready", "Run", "Reset", "Delete", "Logs", "View Execution Metadata", and "Upload Execution Metadata". The main content area displays a table with columns: Status Check, Nodename, Status Details, Logs, and WebUI/Folder. The table lists three nodes: "house-prices-databroker1", "house-prices-model1", and "orchestrator", all of which are marked as "Status: Ready" with green checkmarks. Each node has "view" buttons for Status Details, Logs, and WebUI/Folder.

Status Check	Nodename	Status Details	Logs	WebUI/Folder
✓	house-prices-databroker1	view	view	🔗
✓	house-prices-model1	view	view	🔗
✓	orchestrator	view	view	🔗

The playground-app has been extended to support uploading of execution-run.json to the origin graphene system of the pipeline.

Thank you