

From Intent to Data: An LLM-Agentic Framework for 6G Automated Experimentation

Sid Ali Mahmoudi*, Mahdi Sharara†, Mohamed Readh Fentazi*, and Adlen Ksentini*

*EURECOM, Sophia-Antipolis, France, †CTO Office, VIAVI Solutions, France

Emails: {Sid-Ali.Mahmoudi, Mohamed-Readh.Fentazi, Adlen.Ksentini}@eurecom.fr, Mahdi.Sharara@viavisolutions.com

Abstract—The increasing complexity of 6G experimental infrastructures presents significant challenges for researchers seeking to discover and generate datasets on heterogeneous testbeds and digital twins. Traditional approaches require a deep familiarity with platform-specific APIs, configuration schemas, and measurement pipelines, creating substantial barriers to adoption. This paper introduces a novel LLM-based agentic framework that enables intent-driven data search and automated experiment generation across distributed 6G experimental environments. The proposed system employs a cooperative ensemble of specialized agents, including: Chatbot, Search, Experiment, Translation, and Executor agents, that progressively translate high-level natural language user intents into concrete, executable experiment workflows. The framework integrates semantic scenario matching using sentence transformer embeddings, Model Context Protocol (MCP) for standardized tool interoperability, and a testbed-adaptive translation layer that decouples intent interpretation from platform-specific execution mechanisms. The paper further defines a comprehensive evaluation methodology covering intent understanding, dataset retrieval, semantic scenario matching, end-to-end pipeline reliability, and guardrail effectiveness, providing a rigorous basis for future experimental validation. The proposed architecture addresses critical challenges in experiment portability, reproducibility, and accessibility, establishing a foundation for intent-driven experimentation in next-generation network research.

Index Terms—Intent-Based Networking, 6G Testbeds, 6G Digital Twin, Large Language Models (LLMs), Multi-AI-Agent Systems, Data Generation, Semantic Search, Model Context Protocol (MCP).

I. INTRODUCTION

The transition toward sixth-generation (6G) networks is driving an unprecedented expansion of experimental infrastructure. Research testbeds now span digital twins, software-defined radio platforms, and cloud-native O-RAN deployments, each exposing distinct APIs, experiment configuration schemas, and data collection pipelines [1]. Generating or discovering a dataset for a specific 6G scenario—for example, CSI measurements at 28 GHz under a dense urban traffic steering policy—presently requires a researcher to identify the right testbed, learn its SDK, compose a valid experiment configuration, execute the workflow, and retrieve the outputs. This technical prerequisite creates a steep entry barrier that slows experimentation, hinders reproducibility, and limits participation to domain [2].

Intent-Based Networking (IBN) has established the principle that users should express *goals* rather than *configurations* [3]. Standards bodies including 3GPP, ETSI, and the TM Forum have formalized this concept for network management [4]–[6], with the TM Forum’s Autonomous Networks framework (IG1230, IG1253) and ETSI ZSM [7] defining the Intent Lifecycle from decomposition through translation, activation, and

assurance. Recent work has leveraged Large Language Models (LLMs) to translate natural-language intents into network orchestration commands [8], [9], and conversational LLM-based assistants have been deployed to ease service ordering across operational domains [10]. However, these efforts focus on network service configuration rather than on the research data lifecycle: the discovery, generation, and publication of experimental datasets.

The challenge is compounded by the distributed nature of 6G research data. Datasets produced on one testbed must be discoverable and reusable across the community, yet most data catalogues require structured queries that presuppose knowledge of dataset schemas and vocabulary. Conversational dataset discovery systems exist for general scientific repositories [11], but they lack the domain specificity, testbed integration, and automated generation capability required in 6G research contexts. Meanwhile, recent IBN work has demonstrated that LLMs alone are insufficient for reliable automation in critical infrastructure: they may produce syntactically invalid outputs or introduce hallucinated configuration values, causing service failures [12].

We present an agentic framework for intent-based data search and generation that addresses these challenges through three principal contributions:

- 1) A cooperative multi-agent architecture implemented with LangGraph that orchestrates the complete data lifecycle—from natural-language intent clarification through federated dataset search to automated testbed experiment execution—as a coherent, stateful pipeline.
- 2) A semantic scenario-matching algorithm that uses dense sentence-transformer embeddings and cosine similarity to map high-level user intents onto the closest predefined experiment scenarios in a testbed catalogue, avoiding the brittleness of direct LLM configuration generation.
- 3) A multi-layer guardrail stack comprising safety (input), hallucination, and data-leakage (output) guardrails, alongside a fine-tuned NER module for 6G technical entity extraction.

The remainder of this paper is structured as follows. Section II reviews related work. Section III presents the architecture, workflow state machine, and key algorithms. Section IV describes the implementation. Section V proposes a comprehensive evaluation methodology with supporting mathematical formalizations. Section VI concludes.

II. BACKGROUND AND RELATED WORK

A. Intent-Based Networking and LLM Integration

IBN systems abstract the complexity of low-level network configurations behind high-level goal expressions called intents [3], [4]. Early IBN platforms relied on structured intent formats (JSON, YAML), which, while more user-friendly than raw configuration files, still require domain expertise. The

LLM-centric IBN architecture proposed in [8] advances this paradigm by enabling natural-language intent expression in multi-domain 5G deployments, using Code Llama with few-shot in-context learning to decompose intents into domain-specific Infrastructure-Level Intents (ILIs) for Cloud/Edge and RAN domains. While closely related to our intent translation framework capabilities, the focus of [8] is network service deployment rather than research data generation.

The TelcoAgent-Bench framework [13] introduces the first multilingual benchmarking suite for telecom LLM agents, defining metrics including Intent Recognition Accuracy (IRA) as embedding-based cosine similarity, which directly informs our intent evaluation design. Its finding that models struggle with consistent tool-execution ordering motivates the guardrail and state-machine design choices.

The LLM-enabled service configuration system in [9] further demonstrates the value of a knowledge base of historical intent-NSD pairs with few-shot retrieval over sentence-transformer embeddings (MPNet-v2) and a closed-loop validation agent. Their results show that user satisfaction and NSD cosine similarity improve monotonically over successive interactions, confirming that embedding-based retrieval combined with human feedback is a viable strategy for intent-to-configuration translation. We adopts the same retrieval paradigm but generalizes it to the research data lifecycle rather than service deployment.

B. Conversational Assistants for Intent-Driven Management

An LLM-based Conversational Assistant (Chatbot) for Intent-driven Management within the 6G-INTENSE project [10] is built on the ETSI ZSM Intent Management paradigm and compliant with TM Forum Intent Lifecycle standards (IG1230, IG1253, TMF921 API). The architecture translates vertical users' natural-language service descriptions into structured Business Intents, which are propagated across Autonomous Domains (T-DMO, DMO, NCF) until triggering low-level network actions. The system handles deployment, migration, and abortion of services on a Kubernetes-based cloud-edge infrastructure, incorporating real-time telemetry feedback to detect and resolve resource conflicts through interactive chatbot dialogue.

C. Hybrid AI-Symbolic Validation for Reliable Automation

The reliability problem in LLM-based network automation is addressed in [12] through a hybrid AI-symbolic pipeline that generates TM Forum-compliant service and resource intents (in Turtle/TTL format) and validates them against deterministic guardrails for syntactic, semantic, and policy compliance. The context-aware RAG [14] pipeline uses category-aware few-shot selection and Chain-of-Thought (CoT) prompting [15] for complex structured-to-structured resource intent generation. Crucially, the deterministic Constraint Parser Validator-not the LLM-enforces policy compliance, yielding Guardrail Success Rates of 31.2–56.2% across models and preventing misconfiguration in cases where the LLM output was semantically plausible but policy-violating.

D. RAG-Based Intent Reasoning for Application-Network Interaction

An Intent-RAG framework combining RAG and few-shot learning to translate application-domain intents into structured network intents is proposed in [16]. A key contribution is the *intent refinement* stage, which reduces retrieval ambiguity via domain-guided query instructions, and a coherent reranker that outperforms keyword-based and embedding-based search alone.

The Search Agent implements an analogous refinement step via the NER module and `'is_complete_intent()'` assessment: the Chatbot Agent iteratively enriches the user intent before forwarding structured keywords to the dataset search and scenario-matching pipelines. The RAG paradigm is also directly applied in the Translation Agent's ChromaDB-backed scenario retrieval engine.

E. Specification Mining and Intent Translation

AutoSpec [17] demonstrates the use of LLM-based synthesis-and-repair cycles to translate natural-language RFC specifications into executable I/O grammars, highlighting the value of structured preprocessing, formal validation, and feedback-driven refinement for reliable intent-to-formal-spec translation. The iterative self-repair principle in AutoSpec parallels the clarification and validation loops in the Chatbot Agent.

Recent work [18] shows that LLMs can translate informal natural-language intent into formally checkable postconditions, demonstrating the feasibility of embedding-based semantic evaluation rather than strict text matching. This motivates the use of cosine similarity for both scenario retrieval and intent quality assessment in our case.

Dense passage retrieval using bi-encoder architectures such as `'all-MiniLM-L6-v2'` [19] has become the standard approach for semantic search tasks. ChromaDB [20] provides a lightweight, embeddable vector database suitable for real-time similarity search over experiment scenario catalogues. The sentence-transformer paradigm enables query-document similarity computation without requiring re-ranking with large cross-encoder models, which is critical for low-latency interactive applications. The MPNet-v2 model demonstrated strong performance in the IBN context [9], while reranking on top of embedding retrieval further improves context precision [16].

F. Model Context Protocol

The Model Context Protocol (MCP) [21] standardizes the interface between LLM agents and external tools/services, enabling agents to invoke APIs through a unified JSON-RPC-based protocol. MCP has been adopted in several agentic frameworks as a decoupling layer between agent intelligence and service implementation, eliminating hard-coded API dependencies. We exploits MCP to achieve testbed-agnostic execution by routing experiment management calls through standardized MCP servers that adapt to platform-specific APIs.

G. Positioning

Table I positions our framework relative to representative prior work across the dimensions most relevant to our use case. Unlike IBN systems focused on network service configuration [8]–[10], it targets the research data lifecycle. Unlike hybrid validation pipelines [12] focused on service/resource intent generation, it adds federated dataset discovery and testbed experiment execution. Unlike application-network RAG frameworks [16], it provides testbed-native semantic scenario matching. Unlike benchmarking frameworks [13], 6G-DALI is designed to support heterogeneous experimental platforms through a unified MCP-based abstraction layer, with adapters developed for four representative 6G testbeds including one digital twin testbed.

III. PROPOSED ARCHITECTURE

A. System Overview

Our framework follows an *intent-first* design: the system accepts a free-form natural-language description of a researcher's

TABLE I
COMPARISON OF REPRESENTATIVE RELATED SYSTEMS

System	Intent Input	Dataset Discovery	Experiment Gen.	Multi-Testbed	Guardrails	Standards
IBN-LLM [8]	NL	×	×	Partial (5G)	Partial	×
IBN-Config [9]	NL	×	×	×	Validation	ETSI NSD
Chatbot-IDM [10]	NL	×	×	✓ (cloud-edge)	×	ETSI ZSM/TMF
Hybrid-RAG [12]	NL	×	×	×	✓ (3 layers)	TM Forum TTL
Intent-RAG [16]	NL	Partial	×	×	×	3GPP/TMF
TelcoAgent-Bench [13]	NL	×	×	×	×	Multilingual
Our framework	NL	✓	✓	✓ Adapter-ready	✓ (3 layers)	MCP

data need and autonomously executes the necessary steps to satisfy it. The system identifies two primary fulfillment pathways:

- **Dataset Discovery Pathway:** the requested data already exists in the federated data space and can be retrieved with a structured catalogue query.
- **Dataset Generation Pathway:** no suitable dataset is indexed; the system translates the intent into an experiment configuration, executes it on a suitable testbed, and publishes the resulting data back to the data space.

The routing decision between pathways is made dynamically based on dataset search results presented to the user, who confirms whether the results satisfy their requirements.

The overall architecture, illustrated in Fig. 1, is structured as a **LangGraph StateGraph** comprising five specialized agent nodes, a conditional router, a multi-layer guardrail stack, an NER-based keyword extraction module, a ChromaDB vector store, and two MCP servers. A FastAPI web server exposes the framework as an HTTP service with Server-Sent Events (SSE) for real-time streaming of agent responses to the frontend.

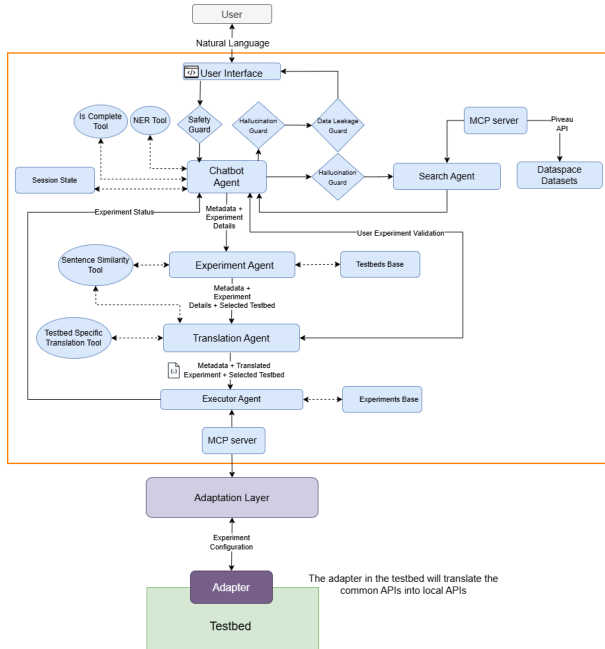


Fig. 1. Intent-based data search and generation LLM-agents system architecture.

B. Agent Descriptions

Chatbot Agent: The primary user-facing component and the entry point for all interactions, analogous to the Tenant-Domain Orchestrator Chatbot in [10]. Responsibilities: (i) validate input with the safety guardrail; (ii) extract 6G/5G technical keywords via the NER tool; (iii) assess intent completeness with the ‘is_complete_intent()’ tool, which serves a role equivalent to the Intent Refinement stage in [16]; (iv) maintain session state across multi-turn dialogues; (v) generate LLM responses filtered by hallucination and data-leakage guardrails; (vi) advance the ‘WorkflowStage’ state machine when sufficient information has been gathered.

Search Agent: A stateless component that translates extracted keywords and constraints into a structured query for the DALI data space API. The agent invokes the ‘search_datasets_in_dataspace()’ tool. Retrieved results—including title, description, keywords, category, access URL, and relevance score—are synthesized into a human-readable response by the Chatbot Agent.

Experiment Agent: Responsible for testbed selection. The agent encodes the clarified experiment intent using the ‘all-MiniLM-L6-v2’ model and computes cosine similarities against the capability vectors stored in the ‘TestbedDB’ ChromaDB collection. The highest-scoring testbed profile (name, type, capabilities, endpoints) is stored in session state and forwarded to the Translation Agent.

Translation Agent: Performs semantic scenario matching (Section III-C). The agent maps the user intent to the top-k predefined scenarios in the selected testbed’s catalogue using embedding similarity, producing a ranked list of candidate configurations for user confirmation. This retrieval-first strategy, rather than generative configuration synthesis, avoids the structural errors observed in direct LLM configuration generation [12], [17].

Executor Agent: Manages the complete experiment life-cycle. The agent invokes ‘launch_experiment()’ via the MCP server’s adaptation layer, monitors execution status through an event-driven notification mechanism, and notifies the user upon completion, providing access to the generated dataset.

C. Semantic Scenario Matching Algorithm

The Translation Agent employs a retrieval-based approach rather than generative configuration synthesis, as direct LLM generation of testbed-specific configurations is error-prone and difficult to validate at scale. The algorithm is formalized as follows.

Let $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$ be the set of predefined experiment scenarios in the selected testbed’s catalogue. Each scenario s_i is associated with a natural-language description d_i encompassing scenario name, theme, topological parameters, and objective. The embedding model $f : \mathbb{R}^V \rightarrow \mathbb{R}^{384}$ is the

‘all-MiniLM-L6-v2’ bi-encoder, a 22M-parameter model producing 384-dimensional dense vectors optimized for semantic textual similarity.

Offline indexing: Each scenario description d_i is encoded as $e_i = f(d_i) \in \mathbb{R}^{384}$, with $\|e_i\|_2 = 1$, and stored in a ChromaDB collection indexed per testbed ID.

Online retrieval: Given the user’s clarified intent q , represented by the concatenation of all user utterances throughout the interaction, the intent embedding is computed as $\mathbf{q} = f(q) \in \mathbb{R}^{384}$. The similarity score for scenario i is:

$$\text{sim}(q, s_i) = \cos(\mathbf{q}, \mathbf{e}_i) = \frac{\mathbf{q} \cdot \mathbf{e}_i}{\|\mathbf{q}\|_2 \cdot \|\mathbf{e}_i\|_2} \quad (1)$$

Candidate filtering and ranking: Let the threshold $\tau = 0.1$; the candidate set is $\mathcal{C} = \{s_i \mid \text{sim}(q, s_i) \geq \tau\}$. The top- k results (currently $k = 3$) are returned by descending similarity:

$$\text{TopK}(q) = \underset{s_i \in \mathcal{C}}{\text{argmax}_k} \text{sim}(q, s_i) \quad (2)$$

Testbed-adaptive dispatch: The architecture decouples intent encoding from testbed-specific retrieval backends. For the VIAVI AI-RSG digital twin, matching is performed against a static JSON scenario catalogue loaded at startup. For other testbeds, the Translation Agent may additionally invoke dedicated testbed-specific translators or query a Testbed Knowledge Base through MCP tools. This adaptive dispatch pattern ensures that the same agent implementation supports heterogeneous platform-specific retrieval strategies without modification.

D. Multi-Layer Hybrid AI-Symbolic Guardrail Architecture

A critical lesson from recent LLM-based network automation research is that generative models alone cannot be trusted for reliable infrastructure control: even state-of-the-art models produce syntactically invalid, semantically incomplete, or policy-violating outputs at non-trivial rates [12]. We worked to address this through a hybrid AI-symbolic guardrail architecture in which LLM reasoning is decoupled from correctness enforcement, following the principle established in [12] of separating probabilistic generation from deterministic validation.

The guardrail stack enforces reliability and security at both the input and output boundaries of the LLM reasoning pipeline, and includes a deterministic validation sub-layer for structured outputs.

Safety Guardrail (input): An LLM-based classifier applied to every user message before processing. The system prompt restricts the classification domain to 6G/5G telecommunications, O-RAN, dataset search, and experiment execution. Short follow-up replies are evaluated in conversation context to avoid false positives. The classifier outputs a binary ‘allowed’ / ‘not_allowed’ decision; blocked inputs trigger a rejection response without further pipeline execution. This design defends against prompt injection, off-topic requests, and adversarial inputs.

Structural Validation Sub-layer: Experiment configuration parameters extracted by the Translation Agent undergo deterministic rule-based validation before being forwarded to the Executor Agent. This sub-layer performs:

- *Syntax validation:* checks that scenario parameters conform to the testbed’s configuration schema (parameter names, types, required fields).
- *Semantic validation:* verifies that parameter values are within feasible ranges (e.g., UE count > 0 , frequency in supported band).
- *Correlation validation:* ensures that inter-parameter dependencies are respected (e.g., a beamforming scenario requires a minimum cell count).

If validation fails, a self-correction prompt is issued to the LLM with the detected errors, analogous to the closed-loop correction mechanism in [9], [12]. This validation loop can execute at most $R_{\max}$ correction rounds before escalating to the user for clarification.

Hallucination Guardrail (output): Verifies that the LLM-generated response is factually consistent with the information present in the current session context (retrieved datasets, matched scenarios, testbed capabilities). Responses asserting facts not grounded in the context—such as fabricated dataset URLs or citation hallucinations—are detected and either corrected or flagged.

Data-Leakage Guardrail (output): Scans LLM outputs for patterns indicative of confidential information disclosure, including internal IP addresses, API keys, credentials, and system-internal parameters. Responses containing sensitive artifacts are suppressed before being delivered to the user.

E. 6G Named Entity Recognition

The NER tool (‘kw_ner_extract’) combines rule-based pattern matching with a fine-tuned transformer-based sequence tagger to extract 6G/5G technical entities from user utterances. Entity categories include: measurement type (throughput, RSRP, SINR, CSI, BLER), technology (5G NR, LTE, O-RAN, Massive MIMO, beamforming), scenario (dense urban, rural, indoor, HetNet), frequency band (sub-6 GHz, mmWave, 28 GHz), topology (UE count, cell count, gNB count), and use case (energy saving, traffic steering, network slicing, QoD). Extracted keywords populate the ‘KwNERResult’ session object and serve as structured input to both the Search Agent’s data space query and the intent completeness assessment, providing the domain-guided query enrichment shown to improve retrieval quality in [16].

F. Session State Machine

The framework maintains a persistent ‘SessionState’ object—serialized to PostgreSQL via the LangGraph checkpoint mechanism—that tracks the full interaction history and workflow progress. The ‘WorkflowStage’ enumeration defines eight stages forming a directed state machine:

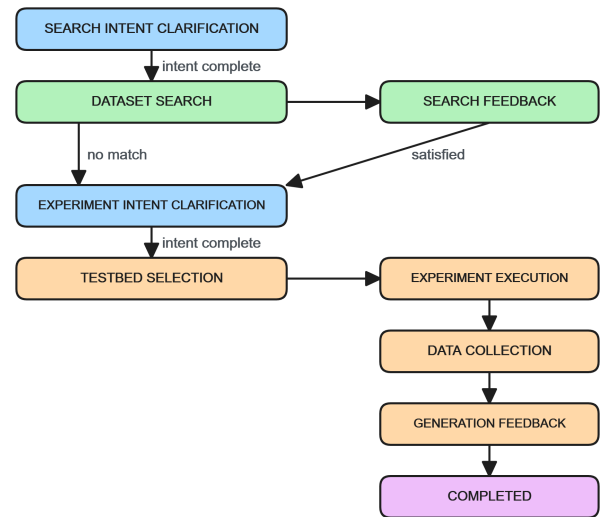


Fig. 2. System Workflow Stages.

IV. IMPLEMENTATION

A. Technology Stack

The LLM backend is configurable: the production deployment uses OpenAI-compatible endpoints (supporting GPT-4o, Qwen, Mistral, and local models), while the guardrail components can be served by a local lightweight model to minimize latency and avoid external API calls for safety-critical decisions. The choice of local model for guardrail evaluation also follows the findings in [12] that smaller models can be competitive on well-scoped classification tasks.

B. MCP Server Integration

Two MCP servers expose external services to the agents:

Dataset Search MCP Wraps the SparkWorks DALI data space search API. Exposes a ‘search_datasets’ tool accepting ‘(query, page, page_size)’ and returning paginated results with title, description, keywords, access URL, category, and relevance score.

Experiment MCP: Provides the adaptation layer to testbed-specific experiment management APIs, following the MCP decoupling principle of [21]. Exposes tools for ‘launch_experiment’, ‘get_experiment_status’, ‘list_experiments’, and ‘get_experiment_results’. The VIAVI ADK integration translates these standard tool calls into ADK-specific REST requests, shielding all upstream agents from platform-specific implementation details. This separation of concerns enables portability across heterogeneous experimental infrastructures by requiring only a platform-specific API translation adapter while preserving the agent orchestration logic.

C. Deployment

The complete system is packaged as a Docker Compose application comprising: the main FastAPI server, a PostgreSQL instance for session persistence, and the two MCP servers. This packaging facilitates reproducible deployment and integration into research infrastructures while remaining independent of any specific testbed implementation.

V. EVALUATION METHODOLOGY

A. Evaluation Dimensions

We propose a comprehensive evaluation framework across five dimensions, extending the four-dimension evaluation of intent-driven systems [8], [12] with data-lifecycle-specific metrics:

- 1) **Intent Processing Quality**-accuracy of intent understanding and keyword extraction.
- 2) **Dataset Retrieval Quality**-relevance of dataset search results.
- 3) **Scenario Matching Quality**-precision of semantic scenario retrieval.
- 4) **End-to-End Pipeline Reliability**-latency, structural correctness, and robustness.
- 5) **Guardrail Effectiveness**-safety and security enforcement accuracy.

B. Intent Recognition Accuracy (IRA)

Following TelcoAgent-Bench [13], we evaluate intent recognition using embedding-based semantic similarity rather than strict label matching. For a test set of M annotated intent-description pairs $\{(u^{(i)}, y^{(i)})\}_{i=1}^M$, the agent-extracted intent $\tilde{y}^{(i)}$ is compared to the gold-standard intent label $y^{(i)}$ using cosine similarity of sentence embeddings:

$$\text{IRA} = \frac{1}{M} \sum_{i=1}^M \frac{f(\tilde{y}^{(i)}) \cdot f(y^{(i)})}{\|f(\tilde{y}^{(i)})\| \cdot \|f(y^{(i)})\|} \quad (3)$$

We additionally define **Keyword Extraction F1 (KE-F1)** to evaluate the NER module separately. Let $K^{(i)}$ be the set of gold-standard technical keywords and $\hat{K}^{(i)}$ be the extracted keywords for sample i :

$$\text{P}_{\text{NER}} = \frac{1}{M} \sum_{i=1}^M \frac{|\hat{K}^{(i)} \cap K^{(i)}|}{|\hat{K}^{(i)}|}, \quad \text{R}_{\text{NER}} = \frac{1}{M} \sum_{i=1}^M \frac{|\hat{K}^{(i)} \cap K^{(i)}|}{|K^{(i)}|} \quad (4)$$

$$\text{KE-F1} = \frac{2 \cdot \text{P}_{\text{NER}} \cdot \text{R}_{\text{NER}}}{\text{P}_{\text{NER}} + \text{R}_{\text{NER}}} \quad (5)$$

C. Dataset Retrieval Quality

Dataset discovery is evaluated using standard information retrieval metrics. Given a query q and a ranked list of retrieved datasets $[d_1, d_2, \dots, d_K]$, with binary relevance judgments $r_i \in \{0, 1\}$:

$$\text{Precision@k} = \frac{1}{k} \sum_{i=1}^k r_i, \quad \text{Recall@k} = \frac{\sum_{i=1}^k r_i}{R} \quad (6)$$

where R is the total number of relevant datasets in the catalogue. **Normalized Discounted Cumulative Gain (NDCG@k)** incorporates rank position:

$$\text{DCG@k} = \sum_{i=1}^k \frac{2^{r_i} - 1}{\log_2(i + 1)}, \quad \text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}} \quad (7)$$

where IDCG@k is the ideal DCG. For multi-turn query refinement sessions, we define the **Mean Session NDCG** as the average NDCG over all turns, measuring how retrieval quality evolves through iterative clarification.

D. Scenario Matching Quality

Semantic scenario matching is evaluated with **Scenario Precision@k (SP@k)** and **Mean Reciprocal Rank (MRR)**. Given a test set of $(q^{(i)}, s^{*(i)})$ pairs where $s^{*(i)}$ is the ground-truth scenario:

$$\text{SP@k} = \frac{1}{M} \sum_{i=1}^M \mathbf{1}[s^{*(i)} \in \text{TopK}(q^{(i)})], \quad \text{MRR} = \frac{1}{M} \sum_{i=1}^M \frac{1}{\text{rank}(s^{*(i)})} \quad (8)$$

We further define the **Intent-Scenario Alignment Score (ISAS)** as the cosine similarity between the intent embedding and the selected scenario embedding, measuring semantic coherence of the match independently of ground truth:

$$\text{ISAS} = \frac{1}{M} \sum_{i=1}^M \text{sim}(q^{(i)}, s_{\text{selected}}^{(i)}) \quad (9)$$

E. End-to-End Pipeline Reliability

Motivated by the end-to-end system reliability analysis in [12], we adopt the following hierarchy of pipeline-level metrics:

Syntactic Correctness Rate (SCR): The percentage of scenario configurations produced by the Translation Agent that pass the structural validation sub-layer on the first attempt, without requiring self-correction:

$$\text{SCR} = \frac{\# \text{ configurations passing validation on first attempt}}{M} \quad (10)$$

Pipeline Success Rate (PSR): The percentage of user requests processed to a clear outcome (dataset delivered or experiment launched) without an unrecoverable agent error:

$$\text{PSR} = \frac{\# \text{ requests reaching COMPLETED or SEARCH_FEEDBACK}}{\# \text{ total requests}} \quad (11)$$

Semantic Score: A composite $[0, 1]$ score measuring how well the final response satisfies the user’s annotated requirements, computed as a weighted average of requirement identification F1, value match accuracy, and unit accuracy-following the scoring framework of [12].

End-to-End Latency: T_{e2e} decomposed as:

$$T_{e2e} = T_{\text{Tools}} + T_{\text{LLM}} + T_{\text{RAG_search}} + T_{\text{guardrails}} + T_{\text{MCP}} \quad (12)$$

F. Guardrail Effectiveness

Guardrail performance is assessed on an adversarial test suite comprising: (i) 100 prompt injection attempts, (ii) 50 off-topic requests, (iii) 50 hallucination-triggering prompts (e.g., requests for non-existent datasets), and (iv) 30 data-leakage-triggering prompts. Metrics are False Positive Rate (FPR, legitimate requests incorrectly blocked) and False Negative Rate (FNR, harmful requests incorrectly allowed):

$$\text{FPR} = \frac{FP}{FP + TN}, \quad \text{FNR} = \frac{FN}{FN + TP} \quad (13)$$

An effective guardrail should achieve $\text{FNR} < 5\%$ and $\text{FPR} < 10\%$. We report results per guardrail layer (safety, structural validator, hallucination, data-leakage) to identify which layer provides the dominant protection for each attack type—a breakdown shown to be essential for understanding hybrid architecture effectiveness in [12].

G. Proposed Experiments

TABLE II
PLANNED EVALUATION EXPERIMENTS

#	Experiment	Description	Metrics
1	LLM Backbone	GPT-4o, Qwen-3-8B, Llama-3-8B, Mistral-7B, Gemma-2-2B; LOO-CV	IRA, KE-F1, SP@3, PSR
2	Embedding Ablation	MiniLM-L6, MPNet-v2, multilingual-MiniLM, text-embedding-3-small	SP@k, MRR, ISAS, latency
3	Threshold Sensitivity	Sweep $\tau \in \{0.0, 0.05, 0.1, 0.2, 0.3\}$, $k \in \{1, 3, 5\}$	SP@k vs. recall
4	Clarification Efficiency	50 simulated sessions; NER+completeness vs. pure LLM	Turns to intent
5	Cross-Testbed	VIAVI DT, other TBs; single τ/k config	SP@k, MRR
6	Guardrail Adversarial	230 adversarial prompts across 4 attack types	FPR, FNR, GSR
7	Guardrail Ablation	PSR/SCR/hallucination rate with vs. without guardrail layers	PSR, SCR

VI. CONCLUSION

We have presented a multi-agent LLM framework that implements an intent-first paradigm for 6G research data acquisition. The proposed architecture spans the complete research data workflow, from natural-language intent clarification through federated dataset discovery to automated experiment specification and execution via standardized MCP interfaces. Its principal contributions include a LangGraph-based five-agent orchestration pipeline with an explicit session state machine, a semantic scenario-matching algorithm grounded in sentence-transformer embeddings, and a hybrid AI-symbolic guardrail architecture that combines LLM reasoning with deterministic validation for reliable experiment configuration.

The implementation demonstrates how heterogeneous experimental platforms can be integrated through platform-specific MCP adapters while preserving a unified interaction model. The evaluation methodology presented in this paper establishes a rigorous framework for future quantitative assessment of the proposed system across multiple 6G testbeds.

ACKNOWLEDGMENT

This work is supported by the European Union’s Horizon Europe programme under the 6G-DALI project (Grant Agreement No. 101192750).

REFERENCES

- [1] Z. Tao, W. Xu, Y. Huang, X. Wang, and X. You, “Wireless network digital twin for 6g: Generative ai as a key enabler,” *IEEE Wireless Communications*, vol. 31, no. 4, p. 24–31, Aug 2024. [Online]. Available: <http://dx.doi.org/10.1109/mwc.002.2300564>
- [2] F. Minucci, R. M. N. Oishi, H. Xiong, D. Verbruggen, C. Thys, R. Hersyandika, R. Beerten, A. Colpaert, V. Ranjbar, and S. Pollin, “Building a real-time physical layer labeled data logging facility for 6g research,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.01298>
- [3] E. Zeydan and Y. Turk, “Recent advances in intent-based networking: A survey,” in *Proc. IEEE Vehicular Technology Conference (VTC)*, 2020.
- [4] 3GPP, “Intent driven management services for mobile networks,” 3GPP, Tech. Rep. TS 28.312, 2023.
- [5] ETSI, “Zero-touch network and service management (ZSM); intent-driven autonomous networks; generic aspects,” ETSI, Tech. Rep. GS ZSM 011, V2.1.1, 2024.
- [6] TM Forum, “Intent management API TMF921A,” TM Forum, Tech. Rep. v1.1.0, 2022.
- [7] —, “IG1230 autonomous networks technical architecture,” TM Forum, Tech. Rep., 2023.
- [8] A. Mekrache, A. Ksentini, and C. Verikoukis, “Intent-based management of next-generation networks: An LLM-centric approach,” *IEEE Network*, vol. 38, pp. 29–36, 2024.
- [9] A. Mekrache and A. Ksentini, “LLM-enabled intent-driven service configuration for next generation networks,” *arXiv preprint*, 2024.
- [10] G. Adhane, J. P. Fonseca, K. Togias, S. Roy, G. Famitafreshi, K. Ramantas, and C. Verikoukis, “LLM-based conversational assistant for intent-driven management,” in *Proc. IEEE NFV-SDN*, 2025, pp. 1–6.
- [11] G. Agrawal, T. Kumarage, Z. Alghamdi, and H. Liu, “Mindful-RAG: A study of points of failure in retrieval augmented generation,” *arXiv preprint arXiv:2407.12216*, 2024.
- [12] N. Raptis, J. P. Fonseca, G. Adhane, K. Ramantas, and C. Verikoukis, “Reliable LLM-enabled intent-driven management: A validation pipeline for network automation,” in *Proc. IEEE NFV-SDN*, 2025.
- [13] L. Bariah, B. Mefgouda, F. Tavakkoli, E. Molero, L. Powell, and M. Debbah, “Telcoagent-bench: A multilingual benchmark for telecom AI agents,” *arXiv preprint*, 2025.
- [14] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, “A survey on rag with llms,” *Procedia computer science*, vol. 246, pp. 3781–3790, 2024.
- [15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [16] S. Mostafa, M. K. Abdel-Aziz, M. S. Elbamby, and M. Bennis, “RAG-enabled intent reasoning for application-network interaction,” *arXiv preprint arXiv:2505.09339*, 2026.
- [17] K. Liu, D. Chakraborty, A. Liggesmeyer, and A. Zeller, “Synthesizing precise protocol specs from natural language for effective test generation,” *arXiv preprint arXiv:2511.17977*, 2025.
- [18] M. Endres, S. Fakhoury, S. Chakraborty, and S. K. Lahiri, “Can large language models transform natural language intent into formal method postconditions?” in *Proc. ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (FSE)*, 2024.
- [19] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in *Proc. EMNLP*, 2019.
- [20] ChromaDB, “Chroma—the AI-native open-source embedding database,” 2024, <https://www.trychroma.com/>.
- [21] Anthropic, “Model context protocol specification, version 2025-06-18,” 2025, <https://modelcontextprotocol.io/>.