



A Gateway for Network Knowledge Graph Management

draft-nmop-cui-nkg-gateway-00

Yong Cui, Jialin Niu, Mingzhe Xing, Lei Zhang

July 20, 2026

The Maturing NKG Ecosystem in NMOP

➤ Standardization Discussion:

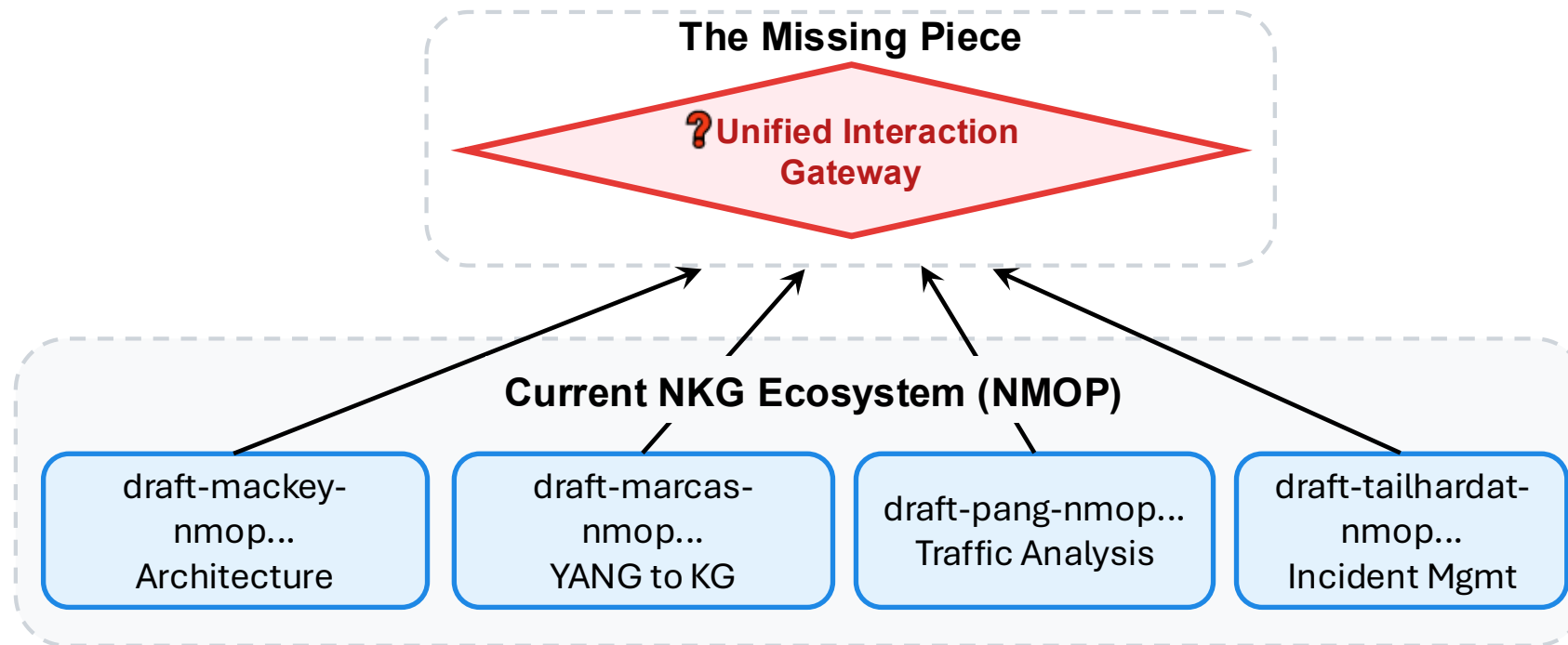
- Moving from concepts to operational domains.

➤ Maturing Foundations:

- Architecture & Data Unification (YANG to KG).
- Domain Apps (Traffic & Incident Management).

➤ The Critical Gap:

- Pipelines are ready, but Interaction is hard.
- How do users/apps interact without KG operation expertise?



Motivation: The Semantic Gap

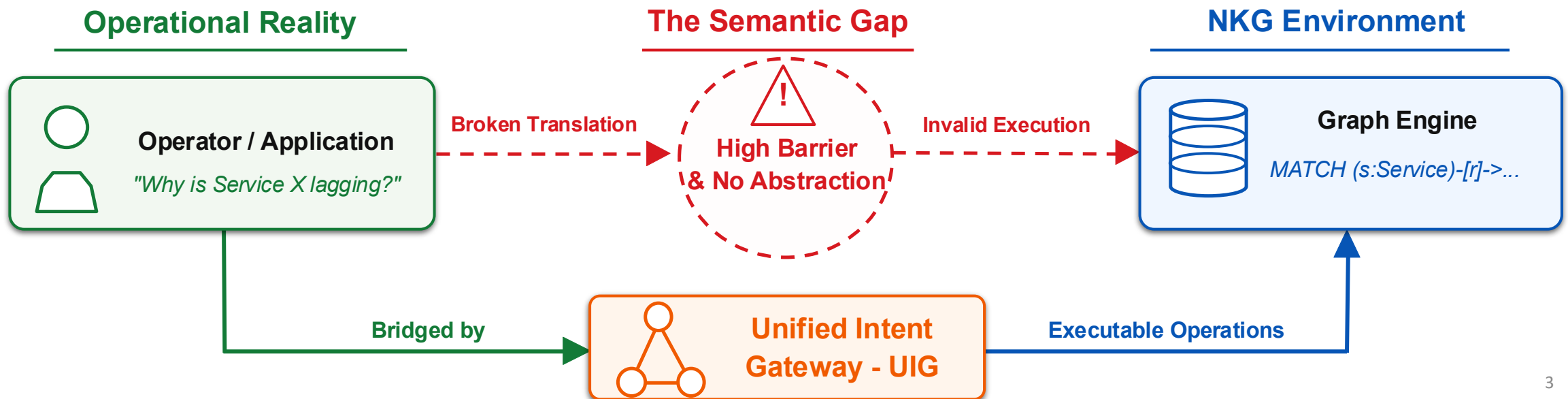
➤ High Usage Barrier:

- Requires deep knowledge of underlying graph schemas, entity types, and query syntax (e.g., Cypher, SPARQL).

➤ Lack of Intent Abstraction:

- Mismatch between human User Expectations and strict Graph Operations.

➤ Our Proposal: A intent-aware gateway to bridge this gap.



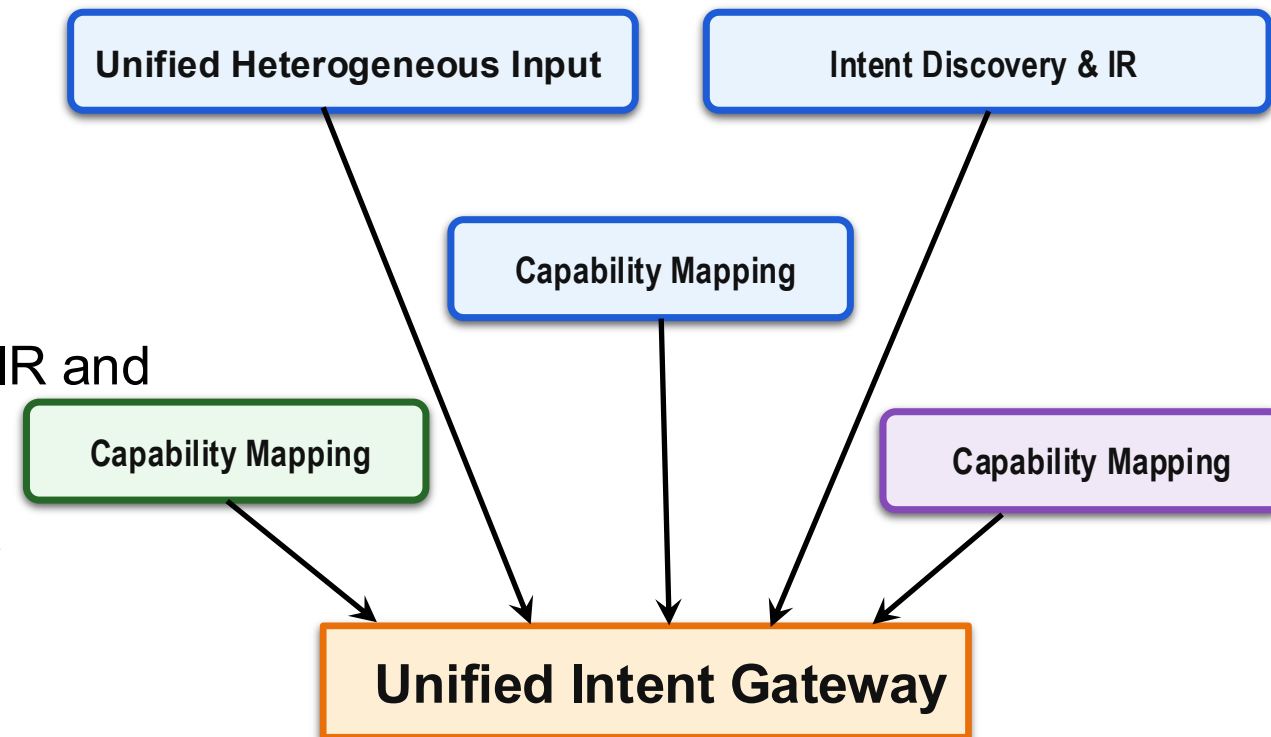
Problem Statement & Key Requirements

➤ The Core Challenge:

- Translating high-level requests into validated graph operations while preserving security and traceability.

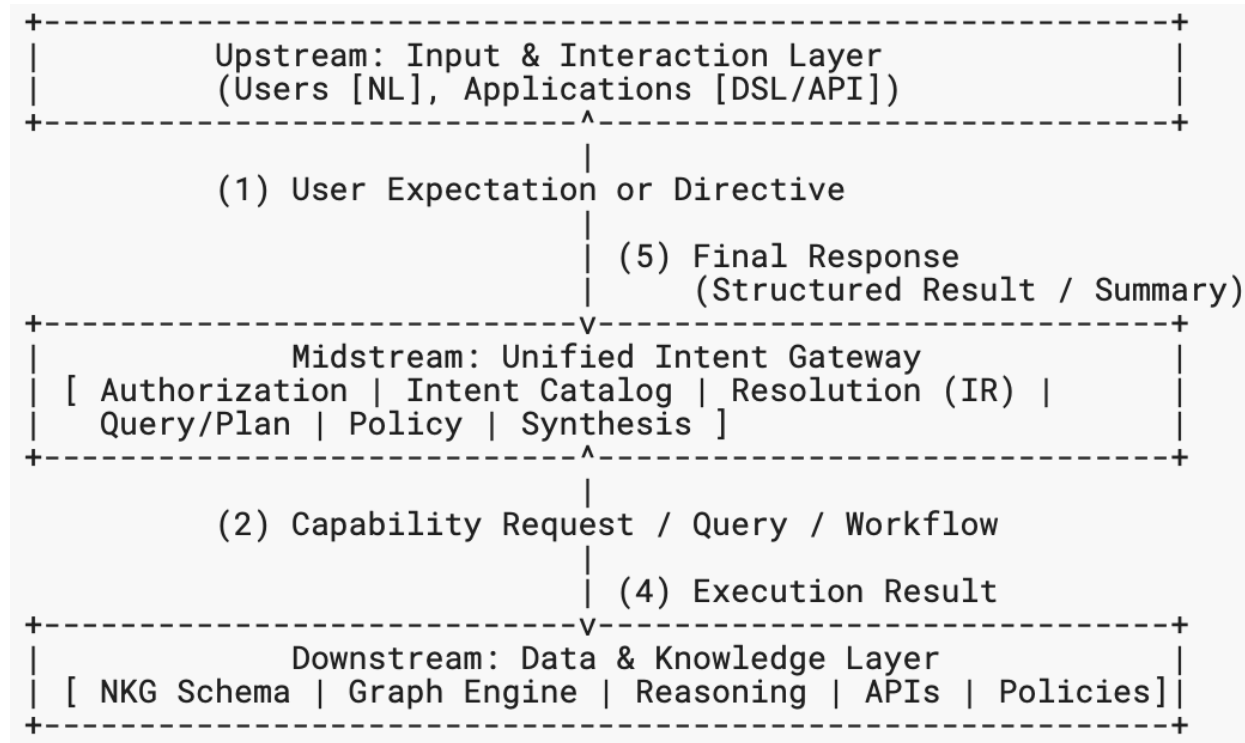
➤ Key Gateway Requirements:

- Unified Input & Discovery: Handle heterogeneous inputs and infer intents.
- Representation & Mapping: Normalize to IR and map to schema-aware capabilities.
- Security & Synthesis: Enforce strict policy validation and synthesize responses.



Gateway Architecture Overview

- **Three-Layer Architecture:** (1) Upstream (Input & Interaction), (2) Midstream (Unified Intent Gateway), and (3) Downstream (Data & Knowledge).



- **Central Mediator:** The UIG connects user/app requests to underlying graph capabilities.

Upstream: Unified Input & Interaction

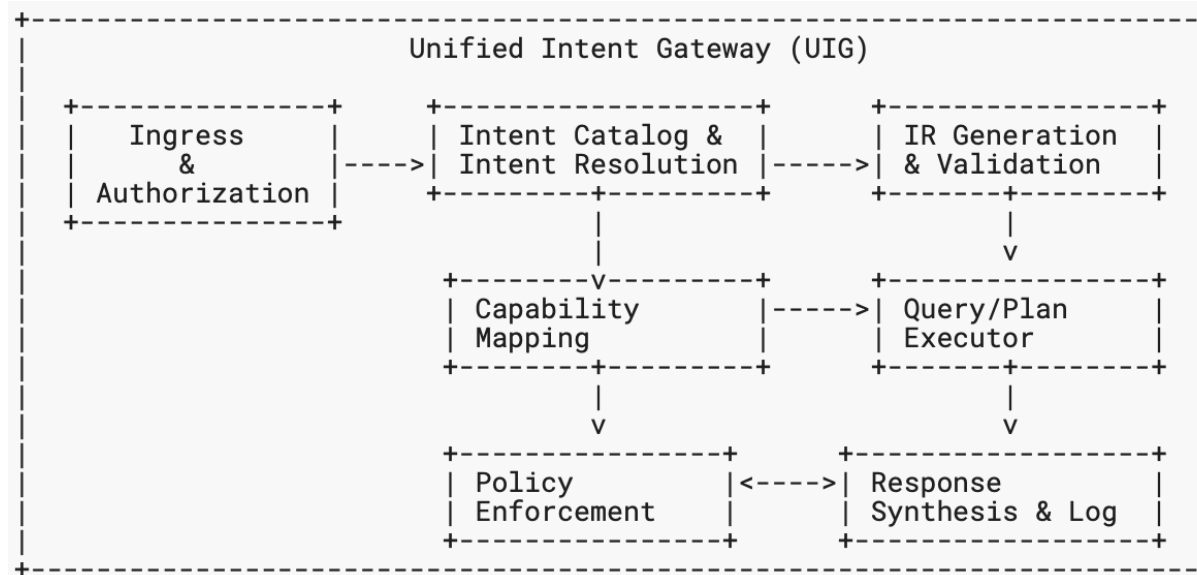


- **Heterogeneous Inputs:** Accepts requests from both human operators and systems.
 - User Input (NL): Natural-language expectations.
 - Application Input (DSL/API): Structured machine directives.
 - Contextual Metadata: Transmits essential context (e.g., requester identity, tenant, operational scope) along with the request.

Midstream: Core Workflow of the UIG



- **Parsing & Discovery:** Matches NL/DSL inputs against the Intent Catalog to infer supported intents. Agents or parsers extract parameters to build the Intermediate Representation (IR).
- **Validation before Execution:** Strictly checks the IR for schema constraints, required fields, and semantic consistency.



- **Mapping & Planning:** Translates validated IR into specific Cypher queries for graph inferences.
- **Policy & Synthesis:** Rejects unauthorized requests via Policy Enforcement and synthesizes responses.

Downstream: Execution & Knowledge Layer



➤ **The Execution Engine:**

- Houses the NKG storage, graph databases (e.g., Neo4J), and reasoning functions.

➤ **Graph Operations:**

- Executes efficient retrieval, causal path extraction, graph traversals, and dynamic topology updates.

➤ HTTP Traffic Investigation

- **Scenario:** A service reports abnormal response latency, and an operator investigates HTTP traffic.
- **Input (NL):** "Check whether there is an HTTP traffic attack affecting service_X."
- **Expected UI Behavior:** Authenticate and authorize the requester; Infer an incident_investigation intent with suspected_attack set to HTTP_Flood; Clarify or assign a time window if the request does not provide one; Map the IR to NKG capabilities for traffic fact retrieval and anomaly checking; Synthesize results into structured output and generate an operator-facing explanation.
- **Example Query (Cypher) & Output:**

```
MATCH (src:IP)-[r:SENDS_HTTP]->(dst:Server)
WHERE r.request_rate > $threshold AND dst.name = "service_X" AND r.window = "5m"
RETURN src, r.request_rate, dst
ORDER BY r.request_rate DESC
```
- Structured JSON containing suspected sources, request rates, affected services, evidence paths;
- **Operator Summary:** "High-rate HTTP requests from suspicious source nodes to service_X are observed within the last five minutes; the evidence path is consistent with an HTTP Flood."



Thank You

July 20, 2026