OASA-KGQA: Ontology-Aware Semantic Agent for KGQA in Smart Cities

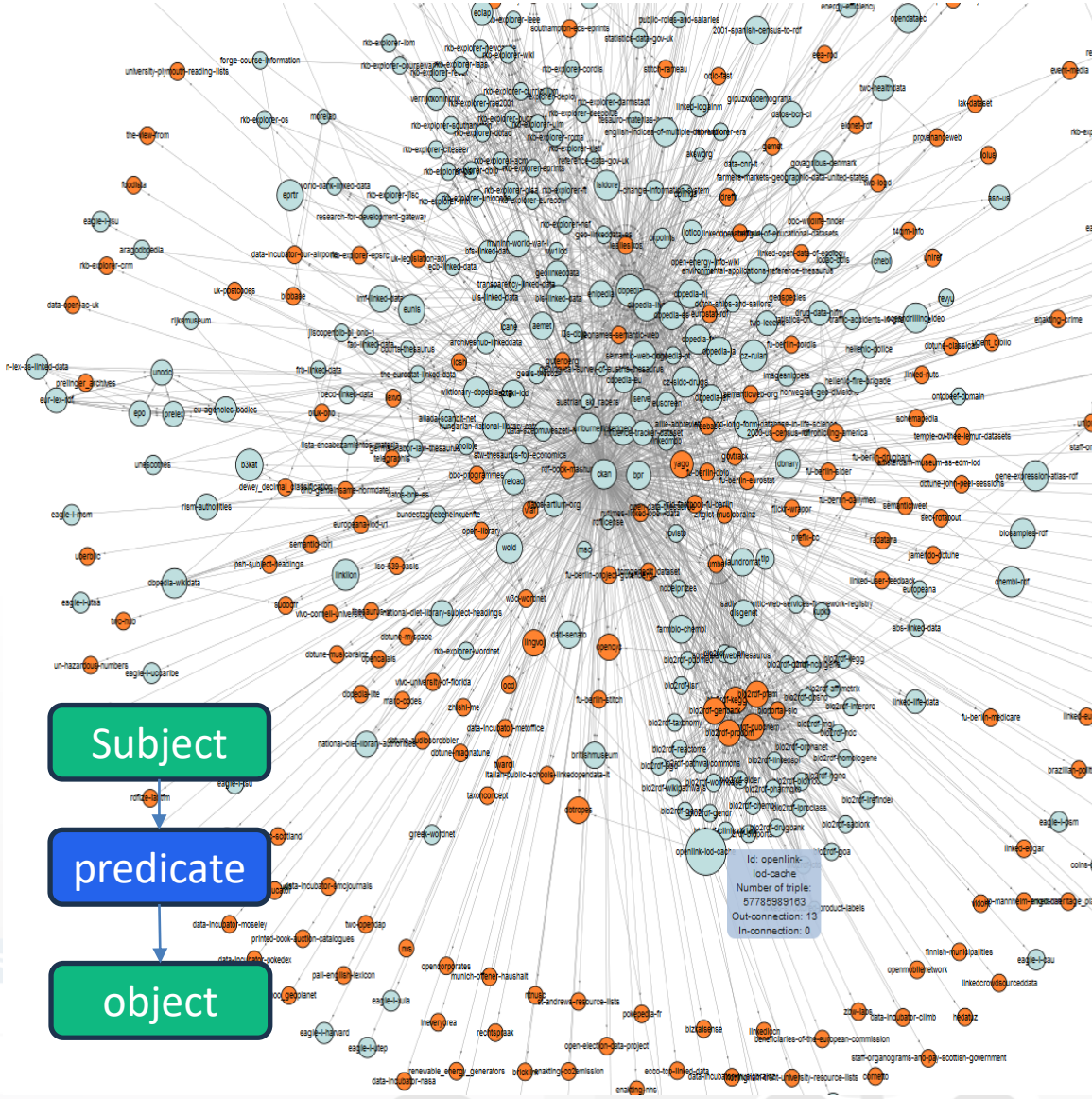
Zahra Fereidooni[1], Marco Fanfani[1],
Paolo Nesi[1], Gianni Pantaleo[1]

[1]DISIT Lab, DINFO,
University of Florence, Florence, Italy
https://www.disit.org, https://www.snap4city.org

BY NC ND
SOME RIGHTS RESERVED

Consideration

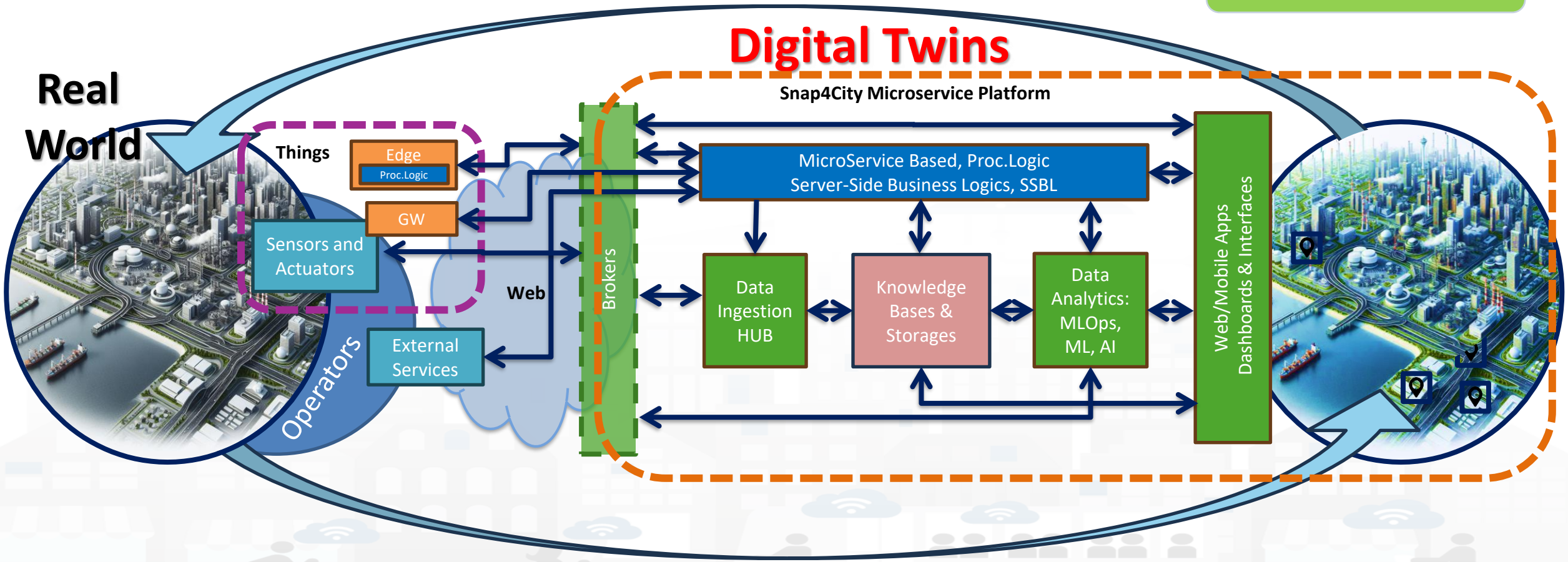
- A large **body of knowledge** has been **formalized** in Knowledge Graphs, Linked data, RDF triple stores, etc.
- **They are**
  - grounded on ontologies, i.e., vocabularies, …covering any domain….
  - **Public accessible >150 billion** of Linked Data triples worldwide in thousands of large RDF stores
  - modeled to reciprocal exploit definitions
  - Including real time data



Digital Twin Development Platform

Km4City KG scale

> 1B triples



Km4City Ontology elements 1.6.8

- **Km4C**: Km4City 1.6.8
- Using
  - **DCTERMS**: for metadata Dublin Core Metadata Initiative
  - **FOAF**: friends of a friends
  - **Good Relation**: entities relationships
  - **iot-lite**: IOT Vocabuary
  - **OTN**: Ontology of Transportation Networks
  - **OWL-Time**: time reasoning
  - **SAREF** Smart Appliances REFerence extension for building devices available at https://saref.etsi.org/saref4bldg/
  - **Schema.org** for people and organizations
  - **SSN**: Semantic Sensor Network Ontology (see https://www.w3.org/TR/vocab-ssn/
  - **WGS84** Datum of Geo-Objects
  - **GTFS**, General Transit Feed Specification, and **Transmodel**, for public transport infrastructures: lines/rides time schedules, real-time records, paths, etc.;
  - **BOT**: Building Topology Ontology. https://w3c-lbd-cg.github.io/bot/
  - **S4CITY**: SAREF extension for Smart City. https://saref.etsi.org/saref4city/v1.1.2/

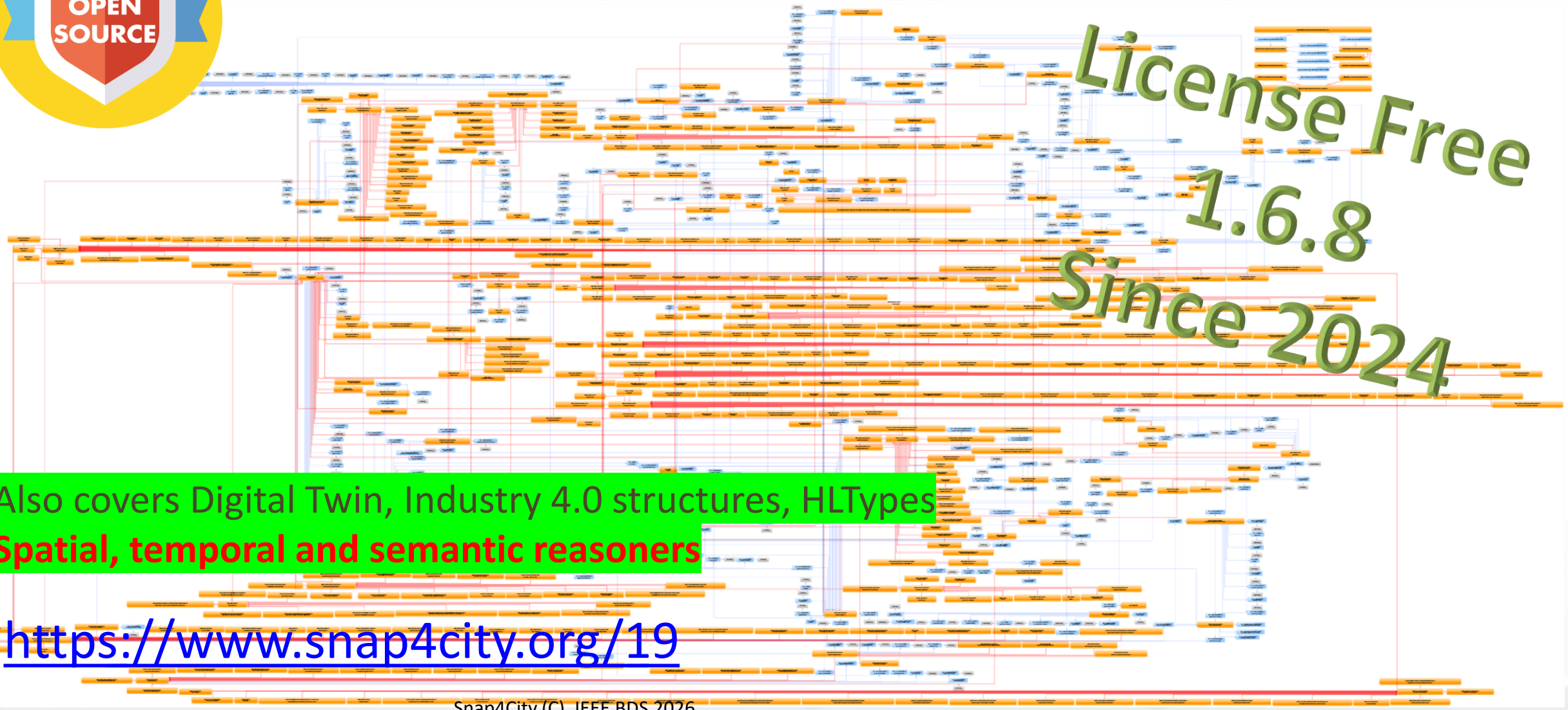
Smart-city Ontology km4city

100% OPEN SOURCE

License Free

1.6.8

Since 2024



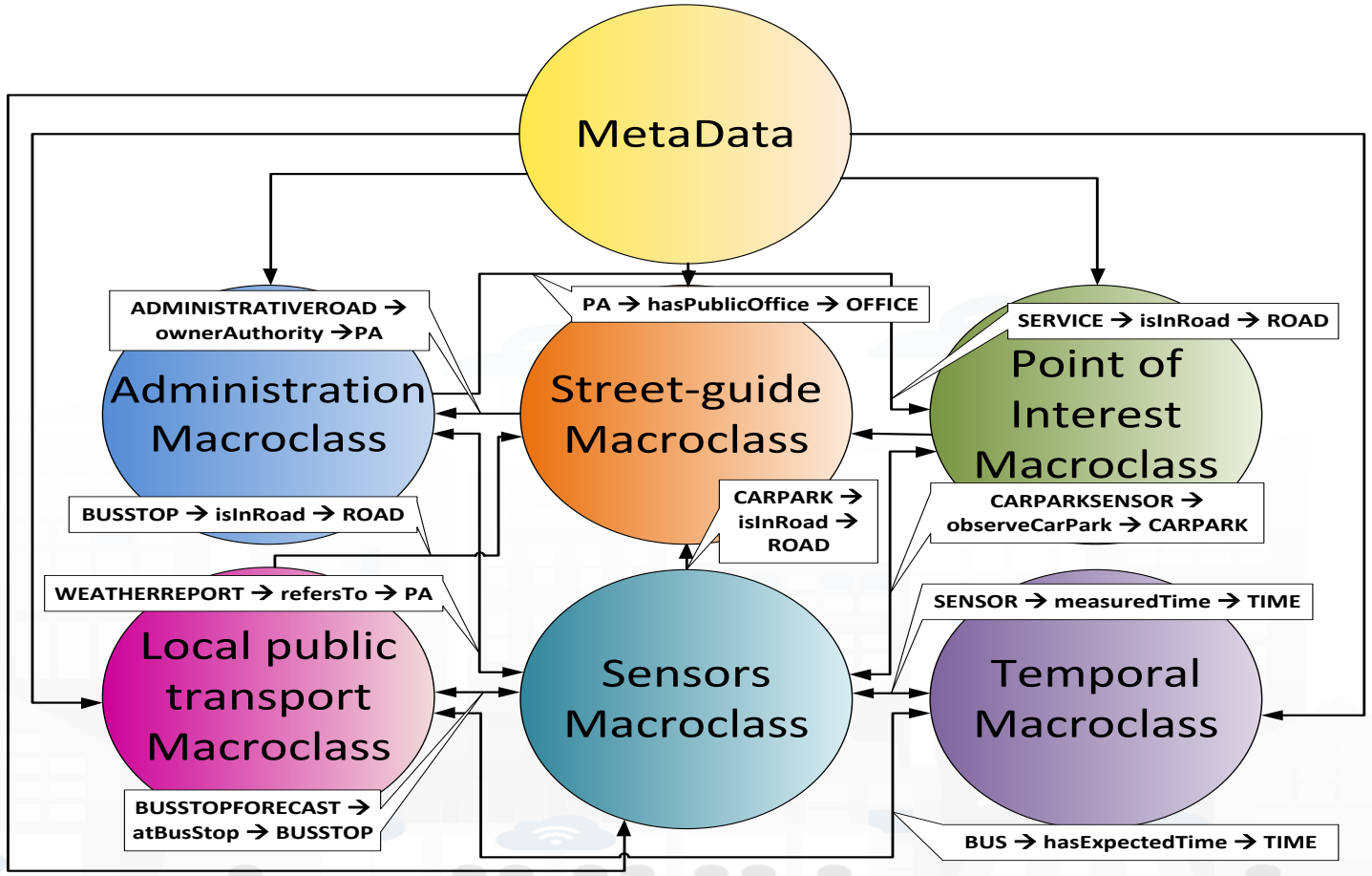
Also covers Digital Twin, Industry 4.0 structures, HLTypes

Spatial, temporal and semantic reasoners

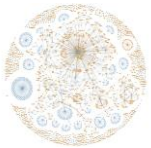
https://www.snap4city.org/19

Smart-city Ontology: 1.6.8

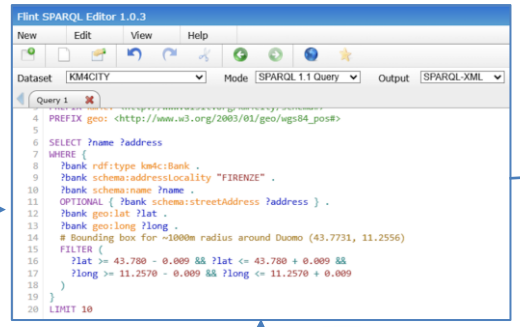
- covers different aspects:
  - Administration
  - Street-guide
  - Points of interest
  - Local public transport
  - Sensors
  - Temporal aspects
  - Metadata on the data
  - Industry 4.0 structures
  - Digital Twin models
  - High Level Types



Manual vs Automated



(A) Write SPARQL query



Results in triples

(B) Generate SPARQL

a KGSA tool

Ask queries in natural language

Get results in natural language

Query Results

#	name	address
1	CASSA DI RISPARMIO DI PARMA E PIACENZA	VIA PIETRAPIANA
2	CASSA DI RISPARMIO DI FIRENZE	VIA DEGLI SPEZIALI
3	CREDEM	VIA LORENZO IL MAGNIFICO
4	MONTE DEI PASCHI DI SIENA	VIA DELLO STATUTO
5	CASSA DI RISPARMIO DI FIRENZE	VIA DELLA CERNAIA
6	CASSA DI RISPARMIO DI PISTOIA E PESCIA	VIA SANTA CATERINA D'ALESSANDRIA
7	CASSA DI RISPARMIO DI FIRENZE	VIA SANTA CATERINA D'ALESSANDRIA

LLM Alone Is Not Enough!

Hallucination	May generate answers or SPARQL queries that look correct but are false.
Factual inaccuracy	Can misinterpret facts stored in the knowledge graph.
Domain knowledge	May not fully understand specific concepts or ontology terms.
Difficulty with unseen KGs	Struggles when the graph structure or schema was not seen before.
Weak schema understanding	May confuse classes, properties, and relationships in complex ontologies.
Incorrect SPARQL generation	Can produce invalid, incomplete, or non-executable queries.
Limited reliability	Outputs are not always consistent for the same or similar questions.
Low interpretability	It is difficult to understand why the model produced a specific answer.

From LLM Limitations to Agentic KGQA

Neuro-Symbolic AI combines neural models with symbolic reasoning to improve understanding, reliability, and structured decision-making.

Agentic Neuro-Symbolic

Plans multi-step actions, uses tools, checks outputs, and reflects before producing the final KGQA result.

Neural: LLM, predictions, ..

Symbolic: KG, GIS, Ontology, reasoners

LangGraph

A framework for designing agent workflows with steps, tools, loops, memory state, and conditional decisions.

Model Context Protocol +Tools

MCP connects the LLM to external tools and data through a standardized interface, enabling reliable tool use in agentic workflows.

PROMPT CoT / Few-shot prompting

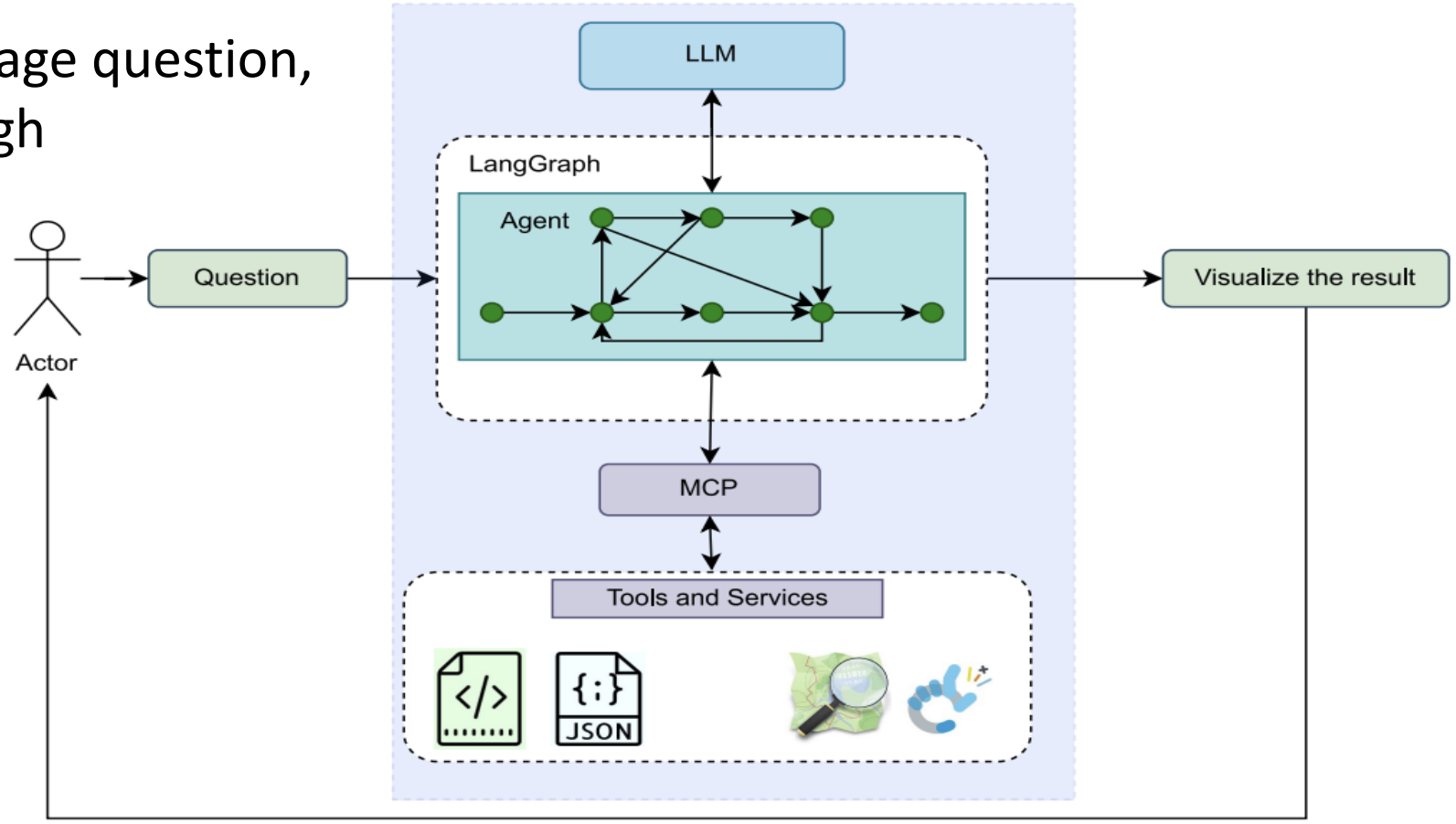
CoT: Chain-of-Thought prompting guides the LLM to reason step by step before producing the final answer or SPARQL query.

Few-shot prompting: Few-shot prompting gives the LLM a few examples of questions and correct outputs so it can follow the same pattern for new questions.

OASA-KGQA is a neuro-symbolic single-agent framework for KGQA, implemented with **LangGraph**.

The user submits a natural language question, and the agent processes it through a graph-structured workflow.

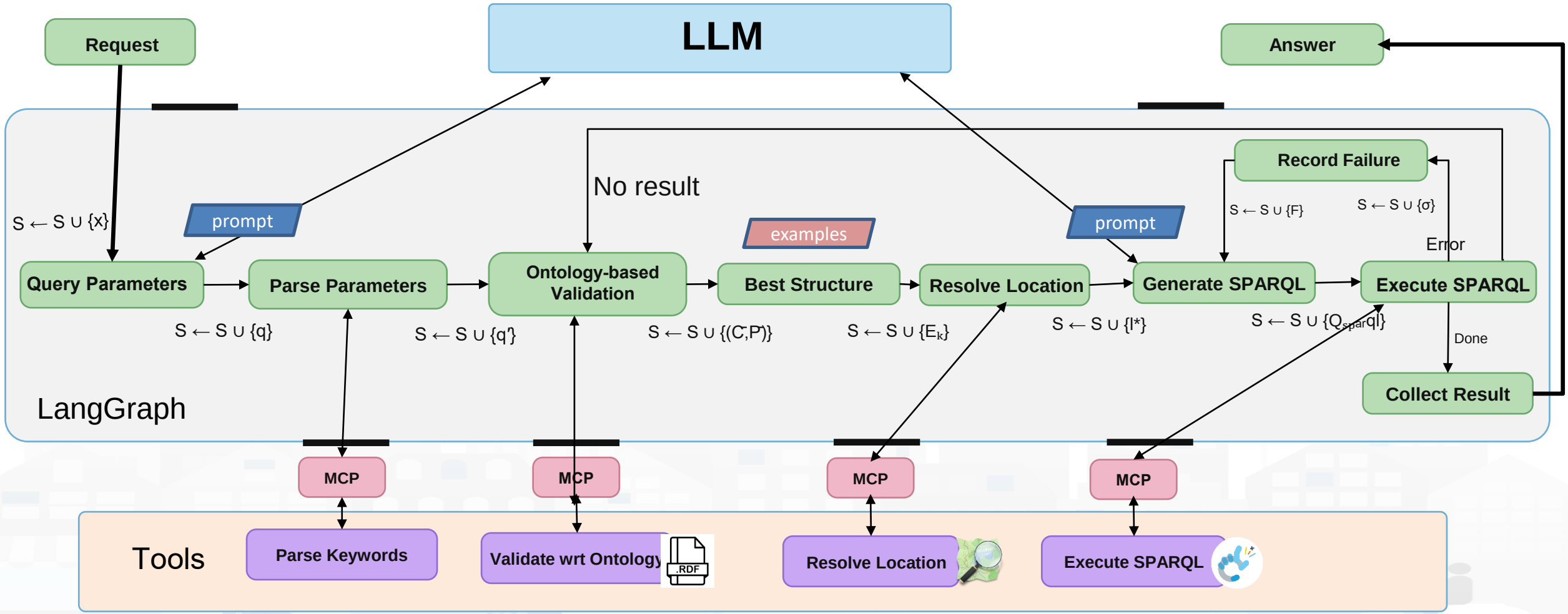
1. User asks a question.
2. LangGraph agent manages the workflow.
3. LLM handles semantic generation.
4. MCP tools provide symbolic capabilities and KG access.
5. Results are visualized for the user.



What workflow

- Request is analyzed to identify entities, they are formally mapped in the ontology terms.
- According to the best identified structure (on the basis of the parameters identified/recognized) → the SPARQL query is generated by LLM by placing all parameters and by using a specific prompt
- SPARQL exeeution may get success, empty or fail (error)
- Detected SPARQL syntax errors are tryed to be corrected for a a number of times (Error<X)

OASA-KGQA Agent Workflow



KG Data-Set: Km4City of Snap4City.org

Snap4City KG

>1B+ triples

semantic index for urban data

- Real-world KG implemented in **Snap4City** and grounded on the **Km4City ontology**.
- Km4City models urban domains such as **mobility**, **energy**, **environment**, **public services**, **sensors**, **roads**, **buildings**, and governance.
- Data are **semantically enriched**, indexed in the KG, and retrieved through **APIs** supporting **relational**, **spatial**, and **temporal** queries.
- **24 natural-language questions** were created. Each one has a manually validated SPARQL query used to produce the **ground-truth** answer.

Question
Retrieve up to 10 bank branches in Firenze within 1 km of the Duomo, returning name, street address.
Retrieve up to 40 sensors in Firenze, returning name, locality, street address.
Retrieve up to 20 libraries in Firenze within 300 meters of the Duomo, returning name, street address.
Retrieve up to 20 bus stops in Firenze within 300 meters of the Duomo, returning name.
Retrieve up to 20 car parks in Firenze within 500 meters of the Duomo, returning name, coordinates, and street address.
Retrieve up to 50 restaurants in Firenze within 1 km of the Duomo, returning name, coordinates, and street address.
Retrieve all weather sensors in Firenze ordered by distance from (11.2377, 43.7791), returning name, coordinates.
Retrieve up to 10 theatres in Firenze within 500 meters of the Duomo, returning name, coordinates, street address.
Retrieve all services in Firenze within 1 km of (43.8100, 11.2051) classified as accommodation, bar, restaurant, shop, tourism, or car rental, returning name, type, latitude, longitude, full address, and road name.
Retrieve up to 50 services in Firenze with address locality Firenze, returning URI, name, type label, latitude, longitude, full address, and road name.
Retrieve up to 50 services in Firenze with address locality Firenze, excluding parking or sensor labels, returning URI, name, type label, latitude, longitude, full address, and road name.
Retrieve up to 50 services in Firenze that have more than 10 RDF triples, returning URI, name, type label, and number of triples.
Retrieve all parking spots in Firenze with available space, returning status.
Retrieve all entities in Firenze with access to entry RT048001000082AC, returning identifier.
Retrieve all road elements in the Firenze part of Viale Francesco Redi, returning the identifier.
Retrieve all road elements in the Firenze part of Viale Francesco Redi, returning the identifier.
Retrieve all hotels in Scandicci, returning name, address.
Retrieve all services in Firenze on Via Riguccio Galluzzi, returning name and type.
Retrieve all services in Firenze named AUTONOLEGGIO AMODEO DI AMODEO BRUNO, returning road name.
Retrieve all weather sensors in Firenze, returning count.
Retrieve all banks in Firenze named CREDEM, returning the name.
Retrieve all sensors in Firenze named METRO762, returning longitude, latitude.
Retrieve all roads in Firenze named Via Reginaldo Giuliani, returning the identifier.
Retrieve all geometries in Firenze named FI-PO-PT-Test1_DISIT_AC, returning WKT.

Assessment

Compared Approaches

Baseline

Static pipeline using **template-based parsing, ontology grounding, lightweight retrieval, and prompt engineering**. More rigid and requires more manual tuning.

LLM + Ontology

Directly gives the LLM the **natural-language question** and **ontology structure**, then asks it to generate SPARQL without the agent workflow or specific prompt engineering.

OASA-KGQA

Agentic graph with **query parsing**, **ontology validation**, **location resolution**, **few-shot retrieval**, **SPARQL generation**, **execution**, and **error handling**.

Evaluation Metrics(HITS@1)

Error-free

The generated SPARQL query is syntactically correct and executes without runtime errors. Results may be diff. from expected

Exact-match

The query output is identical to the ground-truth response. This is a strict evaluation of both understanding and query generation.
Results are those expected.

COMPARISON RESULTS. RATES ARE EXPRESSED AS HITS@1 CONSIDERING 24 TOTAL QUESTIONS.

Method	*error-free (%)*	*exact-match (%)*
LLM+Ontology	20.83	00.00
baseline	70.83	50.00
OASA-KGQA	**95.83**	**70.83**

Why OASA-KGQA improves performance

The gain is not from a bigger model - it is from better orchestration.

Better term identification

NL terms are extracted into structured slots before generation.

Ontology validation

Classes and properties are checked against the target schema.

Spatial grounding

Landmarks and place names become coordinates.

Few-shot guidance

Similar validated examples guide SPARQL patterns.

Failure-aware loops

Empty results and syntax errors are handled differently.

State preservation

Useful context is kept across regeneration attempts.

In short: constrain the LLM before generation, then verify with the KG after generation.

Limitations and future directions

A practical system, with clear opportunities for improvement.

Current limitations

- Evaluation uses 24 questions.
- Exact match is strict but does not capture partial usefulness.
- Error recovery is bounded by counters and prompt rules.
- Domain transfer requires ontology and example setup.

Future work

- Larger and more diverse benchmarks.
- Better automatic example selection.
- Richer error diagnosis and repair rules.
- Evaluation across additional domain ontologies.
- User-facing result explanation.

Conference discussion point

How much symbolic structure is needed before an LLM becomes a dependable KG interface?