OGC-AI: A Retrieval-Augmented Large Language Model Interface for Open Geospatial Consortium Web Services

Thunyathep Santhanavanich[1,2], Volker Coors[1]

[1]Center for Geodesy and Geoinformatics, Stuttgart University of Applied Sciences (HFT Stuttgart), Stuttgart, Germany
Schellingstrasse 24, 70174 Stuttgart, Germany
(thunyathep.santhanavanich, volker.coors)@hft-stuttgart.de
[2]Faculty of Environmental Sciences Technical University Dresden, 01062 Dresden, Germany

Keywords: OGC Standards, Large Language Models, Geospatial Data, Geo-AI Integration, Natural Language Interface

Abstract

In this research, OGC-AI is presented as a retrieval-augmented large language model (LLM) interface that enables plain-language access to Open Geospatial Consortium (OGC) web services while keeping organization-internal endpoints private. Standards documents and service metadata are automatically harvested and indexed; at inference time, relevant snippets are retrieved to compose syntactically correct, standards-compliant requests, execute them via a secure proxy, and return grounded answers with source links. As of 30 April 2025, the corpus comprises 397 documents across 92 OGC standards, spanning both legacy and modern APIs commonly used in Spatial Data Infrastructures. The two use cases are including (i) the use of OGC-AI with complex SensorThings API request, and (ii) generating a working CesiumJS example that consumes geospatial data from OGC API services. A retrieval-augmented strategy is favored over cache-augmented alternatives to accommodate a large, evolving standards landscape. Current limitations (e.g., multi-step analytics, semantic disambiguation, dependence on upstream document structures) and a roadmap toward interactive mapping, task decomposition, and quantitative evaluation are outlined. By lowering the skill barrier to OGC-compliant data access, OGC-AI advances the FAIR principles—especially Accessibility and Reusability within established SDIs.

1. Introduction

The rapid growth of geospatial data presents challenges in terms of data management, integration, and analysis. Generated from a diverse array of sources including remote sensing platforms, in-situ sensor networks, volunteered geographic information (VGI), and Internet of Things (IoT) devices, this data holds immense potential for scientific discovery, environmental monitoring, urban planning, disaster management, and countless other domains. However, realizing this potential is critically dependent on effective data management, sharing, and utilization strategies. Interoperability - the ability of different systems and organizations to access, exchange, and cooperatively use data—remains a cornerstone challenge. The Open Geospatial Consortium (OGC) has been instrumental in addressing this challenge for decades, developing a comprehensive suite of standards that define interfaces and encodings for publishing, discovering, and accessing geospatial data and processing services over the web (Sondheim et al., 1999, Castronova et al., 2013). Adherence to OGC standards is fundamental to implementing the FAIR data principles—ensuring data is Findable, Accessible, Interoperable, and Reusable—within the geospatial domain. These standards provide the syntactic and, to some extent, semantic foundation necessary for machines and humans to interact reliably with distributed geospatial resources (Ivánová et al., 2019). While these standards successfully encode interoperability "on paper," their real-world deployment remains uneven. Configuring, chaining, and querying heterogeneous OGC endpoints typically demands specialist knowledge of XML/JSON encodings, CRS transformations, pagination, and query parameters—tasks that are non-trivial even for experienced GIS analysts and virtually prohibitive for domain experts in environmental science, public health, or urban planning whose primary expertise lies elsewhere.

Recent breakthroughs in instruction-tuned LLMs demonstrate near-human performance at tasks involving code synthesis, semantic search, and multi-modal reasoning. Their proven ability to translate natural language questions into structured database queries suggests a promising avenue for lowering the entry barrier to standards-compliant geospatial infrastructures. However, generic, publicly hosted models still fall short when faced with domain-specific ontologies or standard-specific parameterizations (e.g., bbox, limit, crs). Moreover, the privacy constraints that sequester many government and corporate geodata holdings preclude the option of fine-tuning public models with proprietary examples. Most organizations expose data catalogues only on their intranets for security or licensing reasons; in such settings, public Large Language Models (LLMs) have no prior knowledge about the services' endpoint URLs, data schemas, or access tokens, and therefore cannot provide the "chat-with-your-data" functionality that users increasingly expect.

To bridge this gap we present the OGC-AI, an AI middleware tool that couples organization-internal OGC endpoints with a private, retrieval-augmented LLM to deliver conversational access, visualization, and lightweight spatial analytics without exposing sensitive data or infrastructure details to external services. OGC-AI automatically harvests service metadata (capabilities documents, landing pages, OpenAPI descriptions, and sensor metadata) and indexes them in a vector database. At inference time, the system performs semantic similarity search on the incoming user prompt, injects the most relevant snippets into the model's context window, and instructs the model to (1) formulate correct OGC-compliant queries, (2) execute those queries via a secure proxy, and (3) translate the machine response into human-readable answers, maps, or charts, supporting the mixed legacy-and-modern service landscape typically found in long-standing Spatial Data Infrastructures. By simplifying data query, visualization, and basic analysis, OGC-AI promises to democratize access to valuable geospatial information, thereby fostering wider data utilization and advancing the goals of the FAIR data principles within the geoinformatics community.

OGC-AI aims to significantly lower the barrier to entry for utilizing OGC-standardized geospatial data. By abstracting away the complexities of specific OGC request syntax and service protocols, it empowers a broader range of users, including domain scientists, policy analysts, students, and citizens, to directly query and visualize geospatial information. For experienced GIS professionals, it can serve as a rapid assessment tool for exploring new services or performing quick data checks. Furthermore, it directly enhances the Accessibility and Usability aspects of the FAIR principles for OGC-based data, complementing the Findability and Interoperability inherently promoted by the standards themselves. By enabling interaction with internal organizational data resources, OGC-AI provides a valuable capability currently unmet by public LLM platforms. This research contributes to the burgeoning field of Geospatial Artificial Intelligence (GeoAI) by demonstrating a practical application of LLMs to improve human-computer interaction within established geospatial data infrastructures. Future work will involve expanding the range of supported OGC standards, refining the natural language understanding for more complex geospatial queries, enhancing the analytical capabilities, and exploring robust methods for handling authentication and authorization for secured services.

2. Background

2.1 Open Geospatial Consortium (OGC) Standards

The Open Geospatial Consortium (OGC) standards play a crucial role in the geospatial domain by promoting interoperability, accessibility, and efficient data sharing across various applications and sectors. These standards are essential for integrating geospatial data into modern web applications, enhancing decision-making processes, and supporting a wide range of social, economic, and environmental challenges. The relevance of OGC standards is evident in their application across diverse fields such as flood protection, photogrammetry, remote sensing, and urban planning. The following sections delve into the specific aspects of OGC standards and their significance.

While OGC standards have significantly advanced geospatial interoperability, challenges remain in their widespread adoption and implementation. For instance, some standards like LandInfra have limited software support, which hinders their practical application (Kumar et al., 2019). Additionally, the complexity of integrating diverse data formats and models poses ongoing challenges, despite the progress made with standards like the OGC CDB (Saeedi et al., 2017). Nonetheless, the continuous development and refinement of these standards are essential for addressing the evolving needs of the geospatial community.

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a framework that enhances language models by integrating external knowledge from retrieved documents, improving their knowledge capabilities and reducing hallucination. RAG systems typically utilize web search engines to fetch relevant information, which is then processed to augment the model's output. This approach is particularly beneficial for up-to-date organizational documents, as it allows the model to access and incorporate the latest information, ensuring that responses are accurate and relevant to current contexts (Lewis et al., 2020, Klesel and Wittmann, 2025). RAG combines retrieval and generation processes, where documents are first chunked, and relevant chunks are retrieved based on a user query. These chunks are then used as context for the LLM to generate a response (Lewis et al., 2020, Klesel and Wittmann, 2025).

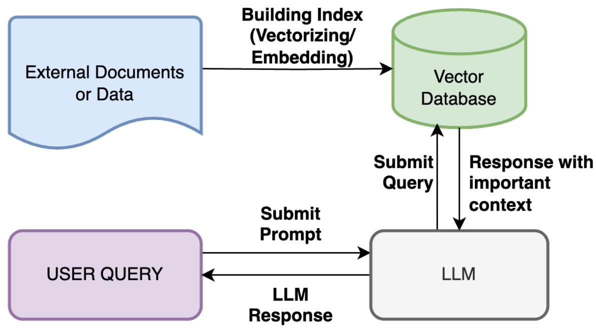

Figure 1. Retrieval-Augmented Generation pipeline.

RAG is particularly effective for up-to-date organizational documents as it allows LLMs to access and utilize current information, ensuring that responses reflect the latest knowledge. This capability is crucial in environments with rapidly changing data, making RAG a superior choice for maintaining accuracy and relevance in generated outputs (Church et al., 2024). In domains characterized by frequent updates such as the evolving suite of OGC standards, RAG ensures that model outputs remain both accurate and relevant. By indexing each version of a standard within the OGC-AI platform, maintainers can seamlessly retire superseded documents and retain only the latest releases, thereby safeguarding against outdated references and streamlining the maintenance of the knowledge base.

As illustrated in Figure 1), the RAG pipeline begins by splitting external documents (e.g., the OGC standards documents) into manageable chunks and encoding each chunk into a fixed-length embedding vector; these vectors are then indexed in a vector database to enable efficient similarity search. When a user submits a query, the query is likewise encoded into the same embedding space and used to retrieve the top-k most semantically similar chunks from the database. Those retrieved chunks are concatenated with the original query to form an enriched prompt, which is passed to the LLM. Finally, the LLM generates a response conditioned on both the user's query and the retrieved context, ensuring that its output is grounded in the source documents while keeping the effective prompt size within practical limits.

2.3 Cache-Augmented Generation

Cache-Augmented Generation (CAG) is a method proposed as an alternative to Retrieval-Augmented Generation (RAG) for enhancing language models by integrating external knowledge sources. Unlike RAG, which involves real-time retrieval of information, CAG preloads relevant resources into the language model's extended context, thereby eliminating retrieval latency and minimizing errors associated with document selection as illustrated in figure 2. This approach is particularly effective when the knowledge base is limited and manageable, allowing for streamlined and efficient processing of queries without additional retrieval steps. The method has been shown to perform comparably or even better than traditional RAG pipelines in certain scenarios, especially when dealing with constrained knowledge bases (Chan et al., 2025).

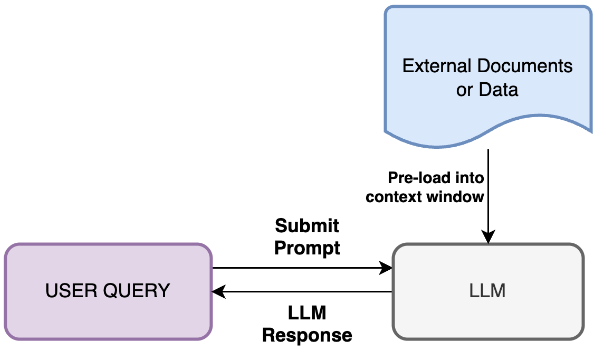

Figure 2. Cache-Augmented Generation pipeline.

However, our target domain involves hundreds of OGC standard documents and continuously evolving. Preloading such a vast and ever growing data would therefore explode the prompt size and negate the very efficiencies CAG is meant to deliver. For these reasons, while CAG remains an attractive approach for narrowly scoped or static knowledge bases, it is not directly applicable to our OGC-AI system. Instead, we rely exclusively on a Retrieval-Augmented Generation (RAG) strategy, which dynamically selects only those document chunks most relevant to each user query and thus keeps the LLM's context window well within practical bounds.

3. Methodology

3.1 Gathering OGC standard Documents

To retrieve the latest OGC standard documents and minimize the risk of using outdated standards, a programmatic approach leveraging web scraping techniques was employed. The core strategy involved identifying the specific web pages on the OGC website that list the standards and their associated documents. By directly accessing these pages via Python, the HTML content of each relevant page was retrieved and then parsed to locate the section dedicated to latest update documents for each standard and avoid the past or old-date documents. On each document, all hyperlink elements were extracted. These hyperlinks typically point to the actual document files or pages containing links to the documents (HTML and PDF). By collecting these URLs, a comprehensive list of potentially the most current standard documents available directly from the OGC website was compiled. This method ensures that the retrieved links reflect the current structure and content of the official OGC publications pages, retrieving the latest versions of the standards as published by the OGC.

This automated process allows for efficient collection of document links from numerous standard pages. While this approach relies on the OGC website's structure remaining relatively consistent, it provides a more direct and potentially more up-to-date method of discovering standard documents compared to manually browsing or relying on potentially static lists. The final compiled list of URLs serves as a starting point for accessing and further processing the standard documents. As of 30 April 2025, the tool successfully extract 397 documents from 92 OGC standards based on the OGC official site[1] as shown in the Appendix A.

3.2 Document Vectorization and Embedding

Each harvested document is first split into overlapping, fixed-length chunks (e.g., 500 tokens with 50 token overlaps) to preserve local context while bounding vector dimensionality. During splitting, we remove boilerplate elements (e.g., headers, footers, duplicated figure captions) and normalize text (lowercasing, Unicode normalization, and removal of non-semantic characters) to reduce noise in the resulting embeddings. Then pre-trained transformer-based embedding model from OpenAI embeddings API has been used to convert each text chunk into a dense, 1024-dimensional vector.

The embedding vectors are then ingested into a vector database configured for approximate nearest neighbor (ANN) search using an inverted file index with product quantization. For each vector, we store: Chunk ID (unique identifier), Source URL (the original OGC endpoint or document link), Byte offset (position within the document), Metadata tags (standard type: OGC API Features, OGC WMS, etc.) These metadata fields allow us to filter retrieval results by standard or service type at query time, supporting both unconstrained semantic search and targeted lookups.

3.3 Client Application

We implement a lightweight single-page web application as shown in figure 3. It communicates with the Assistant backend via the OpenAI Chat Completions API. In the frontend, built with vanilla JavaScript, users are presented with a query input box; basic form validation ensures well-formed input before submission. Upon entry, the application issues an HTTPS POST request to the Chat Completions endpoint, specifying the Assistant's ID and including the user's message as the prompt. All retrieval-augmented generation (RAG) operations described in Section 3.2 are performed entirely within the Assistant, eliminating the need for any external vector store or retrieval service on the client side. Once the Assistant generates the final OGC request (e.g., an XML GetMap request or URL), the response is returned to the client and rendered in the browser, with clickable citations linking back to the original standard documents.

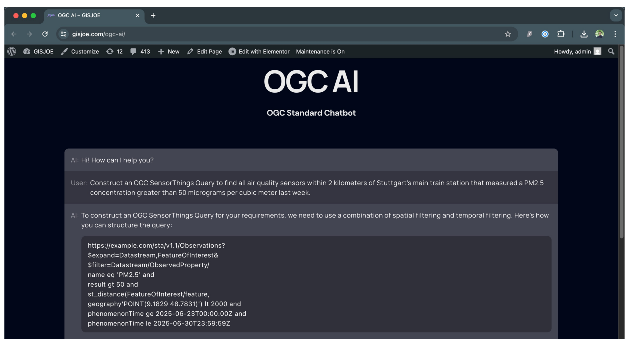

Figure 3. OGCAI web client application.

4. Use Cases

To demonstrate the practical utility of OGC-AI, this section presents two representative use cases. The first illustrates how

[1] https://www.ogc.org/standards/

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-W16-2025-157-2025 | © Author(s) 2025. CC BY 4.0 License.

the system can translate a complex, multi-conditional natural language query into a precise OGC SensorThings API request. The second showcases its capability to act as a development assistant, generating a functional code snippet for a 3D web application based on a user's prompt.

4.1 Querying OGC SensorThings API for Complex Sensor Data

A common challenge for environmental scientists or urban planners is retrieving specific time-series data from sensor networks. The OGC SensorThings API provides a standardized way to access this data, but constructing the correct URL with multiple filters can be complex and error-prone. OGC-AI simplifies this process significantly.

Consider a user who needs to investigate air quality in a specific area of Stuttgart during a particular week. They could pose the following query to OGC-AI:

> *"Construct an OGC SensorThings Query to find all air quality sensors within 2 kilometers of Stuttgart's main train station that measured a PM2.5 concentration greater than 50 micrograms per cubic meter last week."*

This query is complex as it involves three distinct conditions: a spatial filter (proximity to a landmark), a temporal filter (a specific time frame), and a value-based filter on a specific observed property. OGC-AI would deconstruct this request and, by referencing its knowledge of the SensorThings API standard, generate the corresponding syntactically correct API call:

```
https://example.com/sta/v1.1/Observations?
$expand=Datastream,FeatureOfInterest&
$filter=Datastream/ObservedProperty/
name eq 'PM2.5' and
result gt 50 and
st_distance(FeatureOfInterest/feature,
geography'POINT(9.1829 48.7831)') lt 2000 and
phenomenonTime ge 2025-06-23T00:00:00Z and
phenomenonTime le 2025-06-30T23:59:59Z
```

In this scenario, OGC-AI abstracts away the need for the user to know the specific syntax for '$filter', '$expand', the 'st_distance' geospatial function, or the correct ISO 8601 date format. The system handles the lookup of the main train station's coordinates and translates "last week" into a concrete date range, thereby making sophisticated data retrieval accessible to non-technical experts.

4.2 Generating Code for Utilizing OGC Services

Beyond simple data retrieval, OGC-AI can act as a co-pilot for application development, translating a user's goal into functional code. This is particularly useful for leveraging modern OGC API standards that provide data streams for direct integration into web applications.

In this use case, a user wants to visualize 3D building data for Stuttgart. They provide the following prompt:

> *"Based on this OGC API - 3D GeoVolumes Web Service collection `https://ogcapi.hft-stuttgart.de/ogc\_api\_geovolumes/collections?f=json`, help me write a simple 3D web application using CesiumJS loading 3D building models and 3D terrain with a start viewpoint at the city of Stuttgart, using OpenStreetMap as the basemap."*

OGC-AI recognizes the key components of the request: the data source (OGC API - 3D GeoVolumes), the technology stack (CesiumJS), and the specific visualization requirements (viewpoint, basemap). Then, it generates a complete HTML/JavaScript code snippet, explaining how the different parts work together. The result is shown in the figure 4.

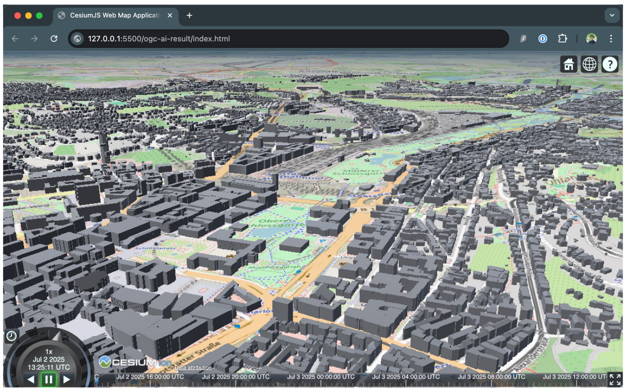

Figure 4. Resulting 3D web application developed by the OGC-AI.

5. Discussion

The development of OGC-AI demonstrates a promising approach to bridging the gap between the complex, structured world of OGC standards and the intuitive, flexible nature of human language. By leveraging a Retrieval-Augmented Generation (RAG) architecture, our tool effectively lowers the barrier to entry for accessing and utilizing valuable geospatial data. This section discusses the current implementation strategy, key architectural considerations, the inherent limitations of the system, and promising directions for future research.

5.1 Deployment Strategy and User Engagement

In its current incarnation, OGC-AI is deployed as an internal tool, accessible only to authenticated users. This controlled rollout serves a dual purpose. Firstly, it ensures compliance with any potential licensing or security constraints associated with the underlying data services. Secondly, and more importantly, it creates a focused environment for gathering high-quality user feedback. By monitoring usage patterns and engaging directly with domain experts, we can iteratively refine the system's performance, usability, and alignment with real-world practitioner needs before considering a wider public release. Future iterations may explore a tiered access model, potentially offering a public-facing endpoint with limited functionality to promote broader adoption while reserving advanced features for registered users.

5.2 Architectural Considerations and Maintainability

Our choice of a RAG architecture was primarily driven by the dynamic and extensive nature of the OGC standards. The cost-benefit analysis between RAG and CAG is a key consideration. While the declining cost of model inference may eventually make CAG—preloading large portions of the knowledge base into the context window—economically viable, RAG remains superior for our use case. The OGC landscape is not static; standards evolve, and new ones are introduced. RAG allows for dynamic, query-relevant context injection, which is more scalable and adaptable than the static preloading of CAG.

The long-term reliability of OGC-AI hinges on the freshness and accuracy of its knowledge base. To this end, we have planned for our future work an automated maintenance pipeline. A daily harvesting script checks the official OGC portal for new or updated standard documents. Upon detection, new documents are chunked, vectorized, and upserted into the vector database, while embeddings for obsolete versions are deprecated. This automated re-indexing is crucial for preventing ”knowledge base decay.” To further ensure system integrity, we plan to implement periodic sanity checks, such as monitoring embedding distributions and sampling nearest-neighbor clusters to verify semantic coherence, guarding against model drift or silent failures in the ingestion pipeline.

5.3 Limitations and Current Challenges

While OGC-AI shows significant promise, it is essential to acknowledge its current limitations.

- **Query Complexity:** The system currently excels at handling direct, single-intent queries that map to a single OGC service request. It is less equipped to handle complex, multi-step analytical questions (e.g., ”Find all hospitals within 1km of areas in Stuttgart that experienced flooding last year”) which would require chaining multiple service calls (OGC API - Features and OGC API - Processes) and performing intermediate spatial analysis.

- **Semantic Disambiguation:** Natural language is inherently ambiguous. A query like ”show me data for Berlin” could refer to administrative boundaries, sensor data located within the city, or documents about the city. While the LLM can often infer user intent from context, robust disambiguation remains a significant challenge that may require interactive clarification dialogues with the user.

- **Dependency on Upstream Data Sources:** The document harvesting mechanism is tightly coupled to the HTML structure of the OGC website. Any significant changes to the website's layout could break the data ingestion pipeline, requiring manual intervention and code updates.

- **Lack of Formal Evaluation:** The system's performance has, to date, been assessed qualitatively through internal testing. A rigorous, quantitative evaluation is a necessary next step. Without metrics comparing the generated OGC requests against expert-crafted baselines for syntactic and semantic correctness, the true accuracy and reliability of the system remain unquantified.

5.4 Future Research and Development Directions

Building on the current foundation, we have identified several key areas for future work that will evolve OGC-AI from a query interface into a more comprehensive interactive geospatial assistant.

- **Integration of Interactive Mapping:** The most critical next step is to move beyond text-only responses. We plan to integrate a lightweight client-side mapping widget (e.g., Leaflet, OpenLayers). This will enable the system to render GeoJSON or GML responses directly as map layers, providing immediate visual feedback. Furthermore, it opens the door to map-based queries, allowing users to draw a bounding box or polygon and ask questions about that specific area.

- **Enhanced Analytical Capabilities:** To address the limitation of handling complex queries, we will explore methods for task decomposition. This involves teaching the LLM to break down a complex user request into a sequence of logical sub-tasks, each corresponding to a specific OGC service call. The results from one call could then be used as input for the next, creating a dynamic service chain orchestrated by the LLM.

- **Robust Evaluation Framework:** We will develop a structured evaluation protocol to quantitatively assess performance. This framework will include metrics such as: (1) syntactic validity of generated requests, (2) semantic accuracy (i.e., does the retrieved data correctly answer the user's question?), (3) end-to-end task success rate, and (4) comparison of response time and quality against a human expert baseline.

- **Provenance and Trustworthiness:** To increase user trust, we will enhance the system's explainability. Instead of just citing the source document, the interface will highlight the specific text chunks retrieved by the RAG process that were used to formulate the OGC request. This transparency allows users to verify the reasoning of the AI, making the system more of a ”glass box” than a ”black box.”

5.5 Broader Implications for the Geospatial Community

The OGC-AI project contributes to the broader field of Geospatial Artificial Intelligence (GeoAI) by proposing a new paradigm for human-computer interaction within Spatial Data Infrastructures (SDIs). By abstracting the technical complexities of service protocols, it significantly enhances the *Accessibility* and *Reusability* of data, directly advancing the FAIR principles for the geospatial domain. This democratization of data access empowers domain scientists, policymakers, and even the general public to engage with geospatial information without requiring specialized GIS training. In doing so, OGC-AI not only serves as a practical tool but also as a model for how LLMs can be responsibly integrated into established data ecosystems to unlock their full potential.

6. Conclusion

This paper has introduced OGC-AI, a retrieval-augmented large language model interface designed to bridge the significant gap between the complexity of OGC standards and the growing need for accessible geospatial data. We have demonstrated that by grounding a powerful LLM with a comprehensive knowledge base of OGC standard documents, it is possible to create an intuitive conversational interface that effectively translates complex, human-language queries into precise, machine-readable web service requests. The presented use cases highlight the system's potential to significantly lower the technical barrier to entry for a diverse range of users to utilize the OGC standards.

Ultimately, OGC-AI represents a critical step towards the true democratization of geospatial information. By abstracting the syntactic and structural complexities of OGC services, our approach enhances the Accessibility and Usability of data, thereby strengthening the implementation of the FAIR data principles within established Spatial Data Infrastructures. While challenges in handling multi-step analytical reasoning and ensuring robust evaluation remain, this work serves as a compelling proof-of-concept for a new paradigm of human-computer interaction in the geospatial domain. We believe that conversational AI systems like OGC-AI will play a pivotal role in unlocking the full potential of global geospatial data resources, empowering scientists, policymakers, and the public alike to address pressing environmental and societal challenges more effectively.

ACKNOWLEDGEMENTS

This research is financially supported by the iCity Management project, founded by the German Federal Ministry of Research, Technology and Space (BMBF) under grant agreement No 13FH9M2IA.

References

Castronova, A. M., Goodall, J. L., Elag, M. M., 2013. Models as web services using the open geospatial consortium (ogc) web processing service (wps) standard. *Environmental Modelling & Software*, 41, 72–83.

Chan, B. J., Chen, C.-T., Cheng, J.-H., Huang, H.-H., 2025. Don't do rag: When cache-augmented generation is all you need for knowledge tasks. *Companion Proceedings of the ACM on Web Conference 2025*, 893–897.

Church, K. W., Sun, J., Yue, R., Vickers, P., Saba, W., Chandrasekar, R., 2024. Emerging trends: a gentle introduction to RAG. *Natural Language Engineering*, 30(4), 870–881.

Ivánová, I., Brown, N., Fraser, R., Tengku, N., Rubinov, E., 2019. Fair and standard access to spatial data as the means for achieving sustainable development goals. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 33–39.

Klesel, M., Wittmann, H. F., 2025. Retrieval-Augmented Generation (RAG). *Business & Information Systems Engineering*, 1–11.

Kumar, K., Labetski, A., Ohori, K. A., Ledoux, H., Stoter, J., 2019. Harmonising the OGC standards for the built environment: A CityGML extension for Landinfra. *ISPRS International Journal of Geo-Information*, 8(6), 246.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T. et al., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33, 9459–9474.

Saeedi, S., Liang, S., Graham, D., Lokuta, M. F., Mostafavi, M. A., 2017. Overview of the ogc cdb standard for 3d synthetic environment modeling and simulation. *ISPRS International Journal of Geo-Information*, 6(10), 306.

Sondheim, M., Gardels, K., Buehler, K., 1999. GIS interoperability. *Geographical information systems*, 1, 347–358.

APPENDIX

Appendix A: List of all OGC Standard used as resources in the OGC AI platform.

Standards

OGC Standards are internationally recognized specifications that let different systems exchange information seamlessly. They are organized by functional area (e.g., data discovery, organization). The functional capabilities of the Web Service Standards are now available as more modern web APIs. Implementers are encouraged to use these newer OGC API Standards.

OGC APIs

Standards that build upon the legacy of the OGC Web Service Standards but define resource-centric APIs that take advantage of modern web development practices.

- GeoAPI Implementation Specification
- OGC API – Common
- OGC API – Environmental Data Retrieval
- OGC API – Features
- OGC API – Maps
- OGC API – Moving Features
- OGC API – Processes
- OGC API – Records
- OGC API – Tiles
- OGC SensorThings API

Services

Standards that implement XML Remote Procedure Calls using the HyperText Transfer Protocol (HTTP).

- 3D Portrayal Service
- Coordinate Transformation Service
- Location Service (OpenLS)
- Table Joining Service
- Web Coverage Processing Service (WCPS)
- Web Coverage Service
- Web Feature Service
- Web Map Context
- Web Map Service
- Web Map Tile Service
- Web Processing Service
- Web Service Common
- Web Services Security

Data Models and Encodings — General

Standards that provide general rules to organize geospatial information, typically sent by a service provider or produced by an application.

- 3D Tiles
- Common Query Language (CQL2)
- CoverageJSON
- EO Dataset Metadata GeoJSON(-LD)
- Filter Encoding
- Geodetic data Grid eXchange Format (GGXF)
- Geography Markup Language (GML)
- GeoPose
- Geospatial User Feedback (GUF)
- GeoXACML
- GML in JPEG 2000 for Geographic Imagery
- Indexed 3D Scene Layers (I3S)
- KML
- Observations, Measurements, and Samples
- OGC Cloud Optimized GeoTIFF
- OGC GeoTIFF
- OGC Moving Features
- OGC Open Modelling Interface (OpenMI)
- OGC Two Dimensional Tile Matrix Set
- Open GeoSMS – Core
- Simple Feature Access – Part 1: Common Architecture
- Simple Feature Access – Part 2: SQL Option
- Simple Features for CORBA
- Simple Features for OLE/COM
- Styled Layer Descriptor
- Symbology Conceptual Core Model
- Symbology Encoding
- Time Ontology in OWL
- Training Data Markup Language for Artificial Intelligence
- Well-known text representation of coordinate reference systems

Data Models and Encodings — Domain Specific

Standards that provide domain specific rules to organize geospatial information.

- CityGML
- CityJSON
- Indoor Mapping Data Format (IMDF)
- LAS Specification
- MUDDI
- OGC Augmented Reality Markup Language 2.0 (ARML 2.0)
- OGC Geoscience Markup Language (GeoSciML)
- OGC IndoorGML
- OGC LandInfra / InfraGML
- OGC WaterML
- OGC WaterML 2: Part 4 – GroundWaterML 2 (GWML2)
- PipelineML Conceptual and Encoding Model
- Points of Interest (POI)
- TimeseriesML (TSML)

Publish-Subscribe, Syndication & Context

Standards for subscription management and other syndication purposes.

- GeoRSS
- OGC API – EDR – Part 2: Publish-Subscribe Workflow
- OGC Web Services Context Document (OWS Context)
- Publish/Subscribe

Sensors

Standards to accessing sensors that are connected to the Web or the Internet of Things (IoT).

- OGC PUCK Protocol
- OGC SensorThings API
- Semantic Sensor Network Ontology
- Sensor Model Language (SensorML)
- Sensor Observation Service
- Sensor Planning Service (SPS)
- STAplus: Sensor Things API Extension
- SWE Common Data Model Encoding
- SWE Service Model Implementation

Discovery

Standards for searching and finding geospatial resources.

- Catalogue Service
- Catalogue Services Standard 2.0 Extension Package for ebRIM Application Profile: Earth Observation Products
- GeoSPARQL – A Geographic Query Language for RDF Data
- OpenSearch
- Ordering Services Framework for Earth Observation Products

Containers

Standards that specify rules for storing and retrieving geospatial information efficiently.

- CDB
- GeoPackage
- OGC Hierarchical Data Format Version 5 (HDF5®)
- OGC network Common Data Form (netCDF) standards suite
- Zarr Storage Specification

Abstract Specification

A reference model used in the development of OGC Standards.

- Abstract Specification

Figure 5. The OGC Standards used as resources in the OGC AI platform (https://www.ogc.org/standards/).