Fangorn: A Conversational Platform for Geospatial Intelligence Analytics [Demo]

Duncan Botti, Newel Hirst, Akshay Jain, Michael Fore, Jeffrey McIntyre, Justin Downes, Rohan Pradhan
Amazon Web Services
Seattle, WA, USA
{dunbotti,nhirst,jainaaks,mikefore,jeffrmci,jusdow,awsrohan}@amazon.com

Abstract

Geospatial analysis traditionally requires specialized GIS expertise, complex software interfaces, and significant manual effort to orchestrate data from heterogeneous sources. We present Fangorn, an agentic platform that enables analysts to perform sophisticated geospatial intelligence operations through natural language conversation. Fangorn combines a modular tool ecosystem based on the Model Context Protocol (MCP) with persistent workspace datasets and reusable analytical workflows. Users explore data through conversational interaction; the system orchestrates queries across external services (ESRI ArcGIS, OGC WFS, GDELT, Amazon Location), persists results as workspace datasets, and can serialize successful workflows into scheduled plans. Our demonstration showcases the complete lifecycle from ad hoc exploration to production-grade automated geospatial analysis, highlighting the platform's plugin-driven multi-viewport interface with synchronized maps, data tables, and real-time execution feedback.

CCS Concepts

• **Computing methodologies** → **Intelligent agents**; **Spatial and physical reasoning**; • **Information systems** → *Geographic information systems*.

Keywords

geospatial AI, LLM agents, natural language interfaces, MCP, agentic workflows, GIS

ACM Reference Format:
Duncan Botti, Newel Hirst, Akshay Jain, Michael Fore, Jeffrey McIntyre, Justin Downes, Rohan Pradhan. 2026. Fangorn: A Conversational Platform for Geospatial Intelligence Analytics [Demo]. In *Proceedings of ACM SIGSPATIAL (SIGSPATIAL '26).* ACM, New York, NY, USA, 4 pages.

1 Introduction

Geospatial analysis is a critical capability across domains including environmental monitoring, urban planning, logistics, and intelligence analysis. Yet the technical barriers to performing sophisticated geospatial operations remain substantial. Analysts must navigate complex GIS software interfaces, understand spatial data formats and coordinate systems, write specialized queries against heterogeneous data services, and manually orchestrate multi-step analytical workflows. These barriers limit geospatial analysis to specialists and create bottlenecks in decision-making processes that depend on spatial insights.

SIGSPATIAL '26, Riverside, CA, USA
2026.

Recent advances in large language models (LLMs) have demonstrated significant potential for geospatial analysis, from automated GIS workflow generation to agentic multi-step spatial reasoning [3, 10]. However, most existing systems focus on code generation or single-pass tool selection rather than supporting the full lifecycle of geospatial analysis: exploratory querying, iterative refinement, dataset curation, and operationalization of successful workflows.

We present **Fangorn**, an agentic platform that enables analysts to perform sophisticated geospatial intelligence operations through natural language conversation. Fangorn makes three key contributions:

(1) **Conversational geospatial analysis**: Analysts interact through natural language; the system handles complex tool orchestration, coordinate transformations, and multi-source data integration transparently.
(2) **Modular tool ecosystem**: A Model Context Protocol (MCP) [1] gateway provides access to diverse geospatial services (ESRI ArcGIS, OGC WFS/WMS, GDELT, Amazon Location) through a unified interface, enabling extensibility without modifying the core platform.
(3) **Agentic workflows with dataset persistence**: Ad hoc queries create persistent workspace datasets; successful analytical workflows can be serialized into reusable plans with scheduling and parameterization, supporting the transition from exploration to production.

Our demonstration showcases the complete lifecycle from exploratory conversation to automated geospatial monitoring, highlighting Fangorn's plugin-driven multi-viewport interface with synchronized maps, data tables, and real-time execution feedback.

2 System Architecture

Fangorn follows a distributed compute model inspired by Apache Spark, adapted for agentic workflows. The architecture separates query planning, coordination, and execution into distinct components, enabling adaptive planning, scalable compute, and extensibility.

2.1 Core Components

The platform comprises five primary components organized around a forest metaphor:

Heartwood provides the foundation layer: configuration management, structured logging, domain models, and the MCP server framework shared across all services.

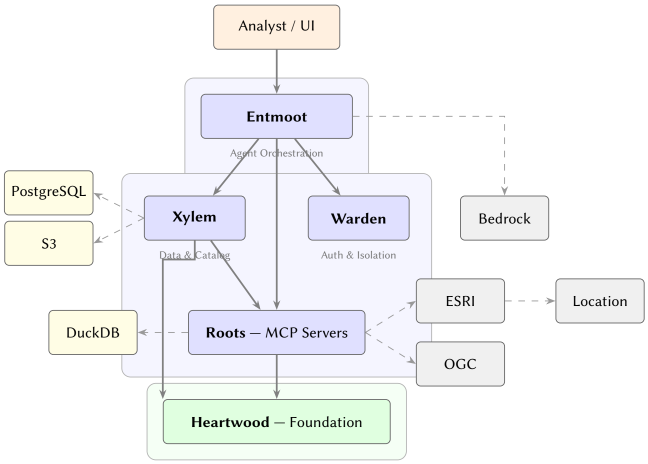

Figure 1: Fangorn system architecture. Analysts interact through Entmoot (agent orchestration), which coordinates Xylem (data catalog/gateway), Warden (authorization), and Roots (MCP servers connecting to external services). Heartwood provides the shared foundation layer.

Xylem serves as the data transport and coordination layer. It maintains a metadata catalog tracking datasets, locations, schemas, and lineage; provides the MCP gateway that routes tool calls to appropriate workers; and offers a DuckDB-based query engine for SQL-based spatial analysis.

Roots implements compute workers that access external systems: ESRI ArcGIS feature services, OGC WFS endpoints, GDELT global event data, Amazon Location geocoding, and specialized analytics workers for spatial operations (e.g., H3-based correlation, buffering, intersection).

Entmoot orchestrates agent execution. It receives natural language requests, discovers relevant tools, decomposes queries into tool sequences, adapts plans based on intermediate results, and streams execution events to clients.

Warden handles authorization and multi-tenant isolation, implementing a fail-closed security model with workspace-scoped access policies. Datasets, plans, and execution history are strictly partitioned—agents cannot access resources outside their assigned workspace boundary.

2.2 MCP Gateway Architecture

The Model Context Protocol (MCP) [1] provides Fangorn's tool integration standard. Each external data source or analytical capability is implemented as an MCP server exposing typed tools with JSON Schema parameters and structured outputs.

The Xylem gateway maintains connections to registered MCP servers, provides unified tool discovery via semantic search, routes invocations with workspace context headers, and aggregates results. This architecture enables:

- **Extensibility**: New data sources require only implementing an MCP server; no changes to the core platform.
- **Isolation**: Each MCP server runs independently with its own dependencies and failure domain.
- **Discovery**: Tools are discoverable by natural language description, enabling the agent to find relevant capabilities without explicit configuration.

Current integrations include ESRI ArcGIS (feature services, geocoding), OGC WFS/WMS, GDELT (global events), Amazon Location (geocoding, routing), and specialized workers for DuckDB SQL, H3 spatial indexing, and geospatial correlation.

3 Agentic Workflows and Dataset Persistence

A key design principle in Fangorn is that conversational exploration naturally produces persistent artifacts. This enables a workflow lifecycle from ad hoc exploration to production automation.

3.1 Workspace Data Model

Each workspace in Fangorn functions as a "data pond"—an accumulation point where agents deposit datasets from various sources. When a user asks a question that requires querying external data, the agent:

(1) Invokes the appropriate MCP tools (e.g., query ESRI feature service)
(2) Receives results as GeoJSON or tabular data
(3) Persists results to S3 as GeoParquet with full schema
(4) Registers the dataset in the workspace catalog with provenance metadata

Datasets are first-class objects with dimensions capturing their modality (geospatial, tabular, text), schema, source lineage, and creation context. Users can inspect, query, visualize, and export datasets independently of the conversation that created them.

3.2 From Exploration to Plans

Successful analytical workflows can be captured as **Aspen Plans**—reusable, parameterized DAGs of execution steps stored as workspace objects. Plans define what tools to call, in what order, with what parameters, and can include LLM reasoning steps for decisions requiring intelligence.

Plan creation follows naturally from conversation:

(1) User explores data through conversational queries
(2) System tracks the sequence of tool calls and their dependencies
(3) User requests plan creation: "Save this workflow as a daily monitoring plan"
(4) System serializes the workflow DAG with parameterized inputs
(5) Plan can be scheduled (cron), triggered on demand, or used as a template

Parameters support three resolution types: **STATIC** (fixed values), **COMPUTED** (evaluated at execution time, e.g., "now - 7d"), and **DERIVED** (flow from upstream node outputs). This enables plans that adapt to current conditions while maintaining reproducible structure.

4 Interactive Multi-Viewport Interface

Fangorn's web interface provides a synchronized multi-viewport workspace where analysts can observe agent execution, refine analyses through conversation, and examine results across multiple modalities simultaneously (Figure 2).

4.1 Execution Transparency and Conversational Refinement

Fangorn provides full visibility into the agent's reasoning: as the system works, analysts observe real-time streaming of tool invocations with parameters, status, and intermediate results. This enables analysts to verify data source selection, intervene when needed, and build understanding of how queries decompose into tool sequences. The system preserves conversational context across turns, so follow-up queries like "filter that to only high-severity alerts" operate on previous results without restating context—enabling iterative refinement from broad exploration to precise analysis through natural dialogue.

4.2 Multi-Modal Workspace Views

Analysts examine results through synchronized viewports: geospatial maps (MapLibre GL JS with Deck.gl for GPU-accelerated rendering), filterable data tables, Markdown-rendered analytical reports, and retrieved artifacts. All viewports are coordinated—selecting a map feature highlights the corresponding table row, and filtering updates layer visibility—allowing analysts to fluidly move between spatial, tabular, and textual representations.

5 Demonstration Scenarios

Our demonstration showcases two scenarios that illustrate the full lifecycle from exploratory geospatial analysis to operational monitoring. Each scenario demonstrates multi-source data fusion, conversational refinement, and the transition from ad hoc investigation to automated workflows.

5.1 Scenario 1: Maritime Vessel Anomaly Detection

An intelligence analyst investigating potential illicit ship-to-ship transfers begins with a natural language hypothesis: "Find vessels that stopped transmitting AIS for extended periods in the Gulf of Aden over the last 72 hours." The system decomposes this into a temporal pattern query, reasoning that "extended periods" implies transmission gaps exceeding six hours. It queries the AIS dataset via DuckDB SQL, identifies vessels with significant dark periods, and renders their tracks on the map with gaps highlighted in red.

The analyst refines the investigation through follow-up dialogue:

(1) "Cross-reference those vessels with any that had close encounters with other ships" — the system performs a spatiotemporal join, identifying rendezvous patterns near the gap locations
(2) "Show me the flag states and vessel types for the matches" — the system enriches results with vessel registry metadata
(3) "Create a daily monitor that flags new dark vessels in this region and alerts me" — the system serializes the complete analytical workflow into an Aspen Plan with a rolling 72-hour window, scheduled for daily execution

This scenario demonstrates the complete explore-refine-operationalize lifecycle: from an initial hypothesis through iterative multi-turn analysis to a production monitoring plan, all through natural language conversation.

5.2 Scenario 2: Humanitarian Crisis Response

A logistics analyst receives reports of civil unrest and asks: "There are reports of civil unrest in Port Sudan. What can you tell me about the situation and nearby logistics infrastructure?" The system orchestrates queries across multiple sources: GDELT for recent protest and conflict events geolocated near Port Sudan, Amazon Location SearchNearby for airports, ports, and hospitals within 100km, and route calculation for potential evacuation corridors.

The analyst iterates through conversational refinement:

(1) The system presents a map with event clusters (colored by severity) overlaid with infrastructure points of interest and calculated routes to the nearest airport
(2) "What about overland routes to Egypt?" — the system calculates alternative land routes via CalculateRoute, comparing transit times and road conditions
(3) "Summarize the situation and options" — the agent generates a Markdown report synthesizing event intensity, infrastructure availability, and route options with estimated travel times

This scenario showcases multi-source data fusion (GDELT events, Location service infrastructure, routing), the synchronized multi-viewport interface (map with routes and events, tabular infrastructure data, generated report), and how persistent datasets enable collaborative handoff to other analysts.

6 Related Work

Geospatial LLM Benchmarks: Recent work has characterized LLM capabilities for geospatial tasks. GeoAnalystBench [10] evaluates code generation for GIS workflows; GeoBenchX [3] benchmarks agentic tool use across complexity tiers; CityBench [2] tests LLMs as urban agents. These benchmarks focus on evaluation rather than providing an end-to-end platform for operational use.

Agent Frameworks: The ReAct pattern [9] combines reasoning and acting in LLM agents. AgentBench [6] demonstrates that agentic evaluation requires distinct infrastructure from static benchmarks. Fangorn builds on these patterns while adding geospatial-specific capabilities and workflow persistence.

Conversational GIS Interfaces: Natural language interfaces to geospatial systems have been explored through NL-to-SQL translation for spatial databases [5] and LLM-driven GIS assistants such as Autonomous GIS [4]. These systems focus on translating individual queries to spatial operations. Fangorn extends this paradigm by providing an agentic architecture that orchestrates multiple heterogeneous services, maintains persistent datasets across sessions, and supports workflow operationalization beyond single-query interactions.

Geospatial Knowledge in LLMs: GeoLLM [7] probes whether LLMs encode useful geographic priors; STARK [8] benchmarks spatiotemporal reasoning. These works inform our understanding

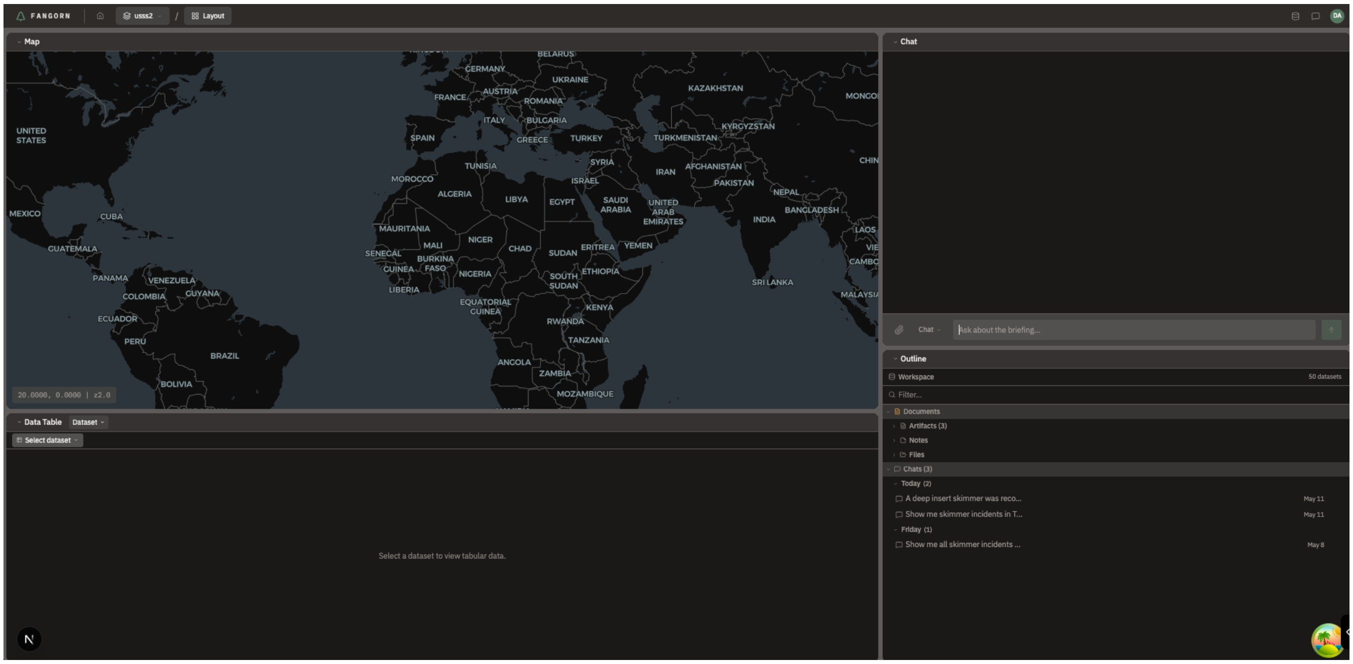

Figure 2: Fangorn's multi-viewport interface showing a weather alert analysis. The Map Viewport (left) displays facility locations with overlaid weather alert polygons. The Chat Viewport (right) shows the natural language query and streaming tool execution feedback. The Data Table (bottom) presents the resulting dataset with facility-alert associations. All viewports are synchronized—selecting a row highlights the corresponding map feature.

of LLM capabilities that Fangorn leverages for query understanding and result synthesis.

7 Conclusion

Fangorn demonstrates that sophisticated geospatial analysis can be made accessible through natural language conversation while maintaining the rigor required for operational use. By combining an extensible MCP-based tool ecosystem, persistent workspace datasets, and reusable analytical workflows, Fangorn supports the complete lifecycle from exploratory querying to production automation. Our demonstration showcases natural language queries against multiple geospatial data sources, real-time agent execution visibility, synchronized multi-viewport exploration, and workflow operationalization. Fangorn is under active development with plans for expanded data modalities, additional MCP integrations, and enhanced collaborative features.

References

[1] Anthropic. 2024. Model Context Protocol Specification. https://modelcontextprotocol.io/.

[2] Jie Feng, Jun Zhang, Junbo Yan, Xin Zhang, Tianjian Ouyang, Tianhui Liu, Yuwei Du, Siqi Guo, and Yong Li. 2024. CityBench: Evaluating the Capabilities of Large Language Model as World Model. *arXiv preprint arXiv:2406.13945* (2024).

[3] Varvara Krechetova and Denis Kochedykov. 2025. GeoBenchX: Benchmarking LLMs in Agent Solving Multistep Geospatial Tasks. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Generative and Agentic AI for Multi-Modality Space-Time Intelligence (GeoGenAgent '25)*. ACM, 27–35.

[4] Zhenlong Li and Huan Ning. 2023. Autonomous GIS: the next-generation AI-powered GIS. *International Journal of Digital Earth* 16, 2 (2023), 4668–4686. doi:10.1080/17538947.2023.2278895

[5] Mengyi Liu, Xieyang Wang, Jianqiu Xu, Hua Lu, and Yongxin Tong. 2025. NALSpatial: A Natural Language Interface for Spatial Databases. *IEEE Transactions on Knowledge and Data Engineering* 37, 4 (2025), 2056–2070. doi:10.1109/TKDE.2025.3525587

[6] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. AgentBench: Evaluating LLMs as Agents. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*. OpenReview.net.

[7] Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David B. Lobell, and Stefano Ermon. 2024. GeoLLM: Extracting Geospatial Knowledge from Large Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*. OpenReview.net.

[8] Pengrui Quan, Brian Wang, Kang Yang, Liying Han, and Mani Srivastava. 2025. Benchmarking Spatiotemporal Reasoning in LLMs and Reasoning Models: Capabilities and Challenges. In *Proceedings of the 39th Conference on Neural Information Processing Systems, Datasets and Benchmarks Track (NeurIPS 2025)*.

[9] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*. OpenReview.net.

[10] Qianheng Zhang, Song Gao, Chen Wei, Yibo Zhao, Ying Nie, Ziru Chen, Shijie Chen, Yu Su, and Huan Sun. 2025. GeoAnalystBench: A GeoAI Benchmark for Assessing Large Language Models for Spatial Analysis Workflow and Code Generation. *Transactions in GIS* 29, 7 (2025), e70135. doi:10.1111/tgis.70135