From metadata to embeddings: enabling agentic AI for subsurface intelligence

B. Lasscock[1]*, D. Arunabha[2], L. Chen[2], M. Gajula[1], K. Gonzalez[1], C. Liu[2], B. Michell[1], S. Namasivayam[1], V.S. Ravipati[2], A. Sansal[1], M. Sujitha[2], G. Suren[2] and A. Valenciano[1] present a practical framework for AI-assisted subsurface data access based on explicit data representations, agent-based workflows, and efficient information retrieval.

Abstract

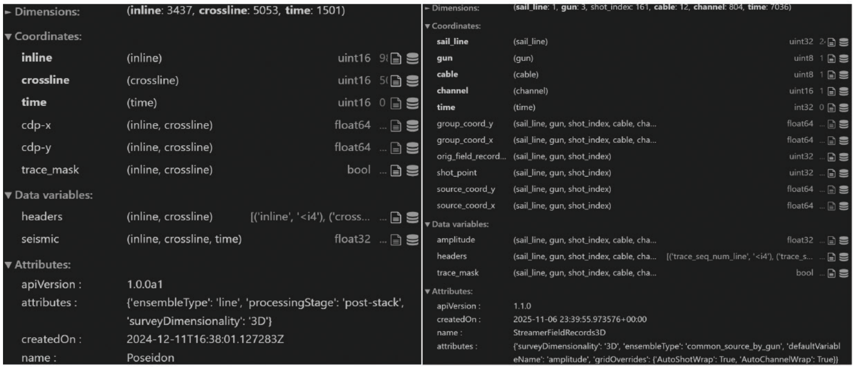
This article presents a practical framework for AI-assisted subsurface data access based on explicit data representations, agent-based workflows, and efficient information retrieval. We demonstrate large-scale conversion of SEG-Y archives into self-describing MDIO v1 datasets and present a case study on agent-driven reconstruction of seismic metadata from legacy text headers. A second case study evaluates embedding-based retrieval across acquisition and processing reports, showing that vector quantisation and graph-based indexing enable low-latency, relevance-driven search. These capabilities are integrated into an interactive, multi-agent system that supports natural-language analysis and coordinated access to structured and unstructured subsurface information.

Introduction

Energy industry organisations and data providers hold petabytes of seismic data, well data, and technical reports, yet much of this information remains difficult to locate, integrate, and use operationally. The challenge is rarely a lack of data; it is the absence of a consistent, machine-readable structure across legacy formats and fragmented metadata sources. When concepts such as geometry, sampling, units, and provenance are implicit, or scattered across SEG-Y headers, PDFs, and spreadsheets, automation becomes fragile, and digital workflows are obstructed.

To address the issue, we have created a digital platform that make subsurface assets self-describing and accessible to modern AI. We introduce a practical digitalization stack: (1) a self-describing seismic representation using MDIO v1 (Sansal 2023a, 2023b; Michell 2025), (2) schema standards based on templates that unify how seismic datasets are represented and used, (3) agent-driven workflows that reconstruct missing or inconsistent metadata in legacy SEG-Y files at scale with verification, and (4) embedding-based retrieval that enables fast, relevance-focused discovery across acquisition and processing documentation. Together, these components close the gap between 'data in files' and data that can be searched, validated, and utilised by downstream applications and AI systems.

The resulting ecosystem supports natural-language interaction across technical, commercial, and operational data by expressing each modality through explicit, machine-readable metadata. Seismic volumes expose geometry, coordinate reference systems, and processing context in a consistent form, while well-log data is integrated through columnar representations suitable for analytical and machine-learning workflows (Gonzalez 2024, 2025). Unstructured documents, such as acquisition and processing reports, are mapped to standardised metadata fields using automated template recognition and hybrid retrieval techniques, including dense retrieval methods (Karpukhin et al., 2020). Structured enterprise systems, including orders, entitlements, contracts, and financial records, are incorporated through normalisation pipelines. Collectively, this approach transforms subsurface data from static archives into an active, queryable knowledge layer that supports AI-assisted analysis, valida-



```
► Dimensions:    (inline: 3437, crossline: 5053, time: 1501)
▼ Coordinates:
  inline         (inline)                                   uint16  9(
  crossline      (crossline)                                uint16  5(
  time           (time)                                     uint16  0
  cdp-x          (inline, crossline)                        float64 ...
  cdp-y          (inline, crossline)                        float64 ...
  trace_mask     (inline, crossline)                        bool    ...
▼ Data variables:
  headers        (inline, crossline)   [('inline', '<i4'), ('cross...  ...
  seismic        (inline, crossline, time)                  float32 ...
▼ Attributes:
  apiVersion :   1.0.0a1
  attributes :   {'ensembleType': 'line', 'processingStage': 'post-stack',
                 'surveyDimensionality': '3D'}
  createdOn :    2024-12-11T16:38:01.127283Z
  name :         Poseidon
```

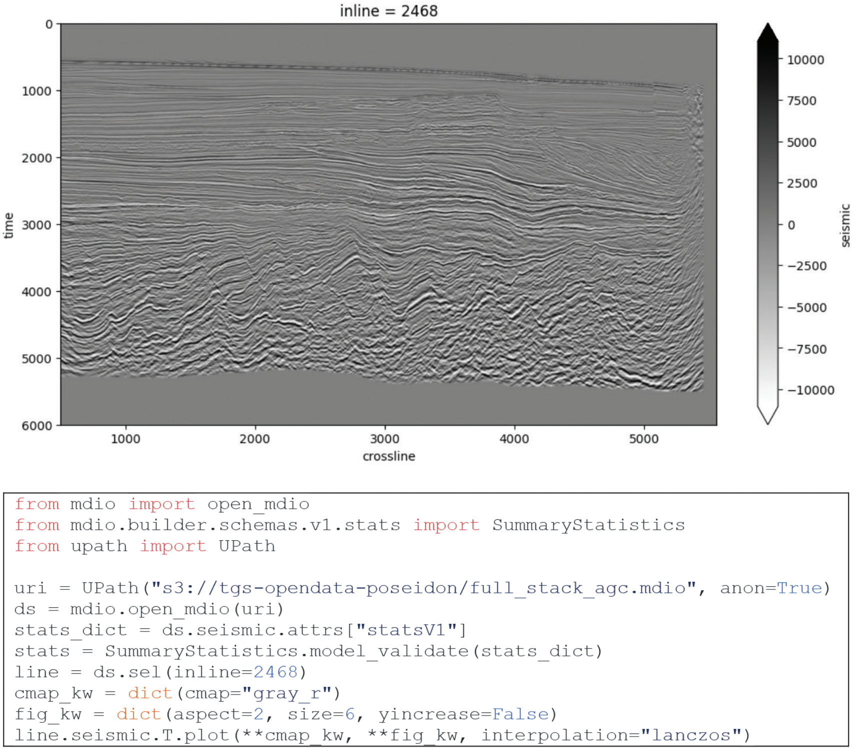
```
► Dimensions:          (sail_line: 1, gun: 3, shot_index: 161, cable: 12, channel: 804, time: 7036)
▼ Coordinates:
  sail_line            (sail_line)                                    uint32  2
  gun                  (gun)                                          uint8   1
  cable                (cable)                                        uint8   1
  channel              (channel)                                      uint16  1
  time                 (time)                                         int32   0
  group_coord_y        (sail_line, gun, shot_index, cable, cha...     float64 ...
  group_coord_x        (sail_line, gun, shot_index, cable, cha...     float64 ...
  orig_field_record... (sail_line, gun, shot_index)                   uint32  ...
  shot_point           (sail_line, gun, shot_index)                   uint32  ...
  source_coord_y       (sail_line, gun, shot_index)                   float64 ...
  source_coord_x       (sail_line, gun, shot_index)                   float64 ...
▼ Data variables:
  amplitude            (sail_line, gun, shot_index, cable, cha...     float32 ...
  headers              (sail_line, gun, shot_index, cable, cha...  [('trace_seq_num_line', '<i4'), ('trace_s...  ...
  trace_mask           (sail_line, gun, shot_index, cable, cha...     bool    ...
▼ Attributes:
  apiVersion :         1.1.0
  createdOn :          2025-11-06 23:39:55.973576+00:00
  name :               StreamerFieldRecords3D
  attributes :         {'surveyDimensionality': '3D', 'ensembleType': 'common_source_by_gun', 'defaultVariabl
                       eName': 'amplitude', 'gridOverrides': {'AutoShotWrap': True, 'AutoChannelWrap': True}}
```

Figure 1 A high-level view of two MDIO v1 datasets, viewed using Xarray (left) a 3D post-stack dataset; (right) a streamer field dataset.

[1] TGS ASA | [2] AWS Generative AI Innovation Center

[*] Corresponding author, E-mail: Ben.Lasscock@tgs.com

DOI: 10.3997/1365-2397.fb2026015



```
from mdio import open_mdio
from mdio.builder.schemas.v1.stats import SummaryStatistics
from upath import UPath

uri = UPath("s3://tgs-opendata-poseidon/full_stack_agc.mdio", anon=True)
ds = mdio.open_mdio(uri)
stats_dict = ds.seismic.attrs["statsV1"]
stats = SummaryStatistics.model_validate(stats_dict)
line = ds.sel(inline=2468)
cmap_kw = dict(cmap="gray_r")
fig_kw = dict(aspect=2, size=6, yincrease=False)
line.seismic.T.plot(**cmap_kw, **fig_kw, interpolation="lanczos")
```

Figure 2 An inline slice sampled from the Poseidon dataset using the code on the left.

Template	Grid Dimensions	Chunk Sizes	Coordinates
CdpAngleGathers2D	cdp, angle	16×64×1024	cdp x/y
CdpAngleGathers3D	inline, crossline, angle	8×8×32×512	cdp x/y
CdpOffsetGathers2D	cdp, offset	16×64×1024	cdp x/y
CdpOffsetGathers3D	inline, crossline, offset	8×8×32×512	cdp x/y
CocaGathers3D	inline, crossline, offset, azimuth	8×8×32×1×1024	cdp x/y
PostStack2D	cdp	1024×1024	cdp x/y
PostStack3D	inline, crossline	128×128×128	cdp x/y
StreamerFieldRecords3D	sail_line, gun, shot_index, cable, channel	1×1×16×1×32×1024	source x/y, group x/y, shot_point, ffid
StreamerShotGathers2D	shot_point, channel	16×32×2048	source x/y, group x/y
StreamerShotGathers3D	shot_point, cable, channel	8×1×128×2048	source x/y, group x/y, gun

Table 1 Summary of MDIO v1 seismic product templates currently used in data management. Each template defines the core dataset dimensions and coordinate variables that structure CDP, offset, angle, shot, streamer, common-offset/common-angle (CoCa), and post-stack data in 2D and 3D, with depth/time variants.

tion, and decision-making across technical and commercial workflows.

MDIO v1: From files to self-describing datasets

Most seismic digitalization challenges occur in the same area: geometry and semantics are implicit. In SEG-Y, key concepts such as dimensionality (2D vs. 3D, post-stack vs. gathers), coordinate scalars, and navigation are derived from trace-order and header conventions that vary by project and vendor. This makes automation fragile and requires every downstream process, analytics, visualisation, and ML to repeat the same interpretation logic.

An MDIO v1 dataset offers a self-describing representation of seismic data, with explicit structural metadata rather than inferred. Each dataset specifies its main dimensions (e.g., cdp, angle, inline, crossline), along with associated coordinate variables (e.g., cdp_x, cdp_y, coordinate scalars). The MDIO dataset framework is defined as a JSON schema, which details coordinates, dimensions, masks, and key survey metadata as separate arrays. This design allows clear interpretation of survey geometry and navigation information where applicable. To promote open and reproducible use, MDIO is released as open-source software under the Apache 2.0 licence (Sansal 2025a), along with cloud-compatible SEG-Y parsing tools (Sansal 2025b). Each MDIO v1 dataset is naturally compatible with the popular Xarray Python library (Hoyer 2017), enabling access to seismic variables and coordinates through a well-established third-party tool.

The following example demonstrates interactive access to a post-stack 3D dataset using the Python MDIO v1 library. Here, seismic amplitudes are indexed by inline and crossline coordinates and visualised without additional geometry reconstruction. The dataset provides sel and isel commands to access data according to coordinate values and logical indexing, respectively. The result is shown in Figure 2 as a seismic inline sampled from the Poseidon dataset.

Seismic template definitions

With MDIO v1, specific conventions for JSON-Schema or templates are designed to support a wide variety of seismic product types across acquisition, processing, and migration stages. Each template establishes the dataset's dimensional structure and required coordinate variables, offering a standard representation for common seismic products. These templates form the structural foundation for both data ingestion and downstream use.

At the time of writing, we have ingested petabytes of seismic data from more than 100,000 individual SEG-Y files into MDIO v1, covering a wide variety of field data, pre-stack, and post-stack seismic product types. Due to the large volume of SEG-Y data, providing a detailed schematisation of the seismic data was impractical, so that task has been deferred to generative AI agents discussed in the next section. The current set of extendable seismic templates used in operational data management is summarised in Table 1. Importantly, because MDIO v1 separates headers, coordinates, and other data from the traces, these templates can be refined by editing the dataset without reingestion.

Cloud-scale ingestion benchmarking

Large-scale ingestion of legacy SEG-Y data into MDIO v1 was benchmarked using source SEG-Y files stored in Amazon S3 (standard storage class). The ingestion workflow is designed to operate directly on object storage, without modifying or relocating the source data. Benchmark ingestion and conversion workflows were executed on c7g.8xlarge instances (32-core AWS Graviton processors), providing a reproducible and well-defined compute environment for parallel SEG-Y parsing. The ingestion process first scans the SEG-Y headers to discover and validate dataset dimensions, coordinate variables, and associated metadata required by the schema. This phase extracts geometry information without materialising trace data. Second, a write phase constructs the chunked MDIO v1 dataset and writes it directly to object storage in the MDIO format. Table 2 shows the timing data for ingesting a collection of post-stack SEG-Y files into MDIO v1. We find that the end-to-end throughput, including the

File No	Total Time (s)	Scan Time (s)	Ingest Time (s)	SEG-Y Size (GB)	MDIO Size (GB)
1	3188.81	1269	1886	1962.0	1473.2
2	2467.06	997	1440	1524.6	1045.7
3	1696.93	649	1022	1020.6	690.4
4	1157.45	440	698	667.7	436.4
5	811.30	337	462	499.7	305.0

Table 2 Benchmark results for ingestion of post-stack SEG-Y datasets into MDIO v1.

Survey	Domain	Gather / Seismic type	Migration Stage	Regularization
2D	Post-stack	Seismic section	—	—
2D	Pre-stack	CDP gathers (offset)	Pre-migration	—
2D	Pre-stack	Shot gathers	—	—
3D	Post-stack	Seismic volume	—	—
3D	Pre-stack	CDP gathers	Post-migration	Regularised
3D	Pre-stack	Shot gathers	—	—
3D	Pre-stack	CDP gathers	Pre-migration	Maybe Regularised
OBN	Pre-stack	Receiver gathers	—	Non-regularised
OBN	Pre-stack	Receiver gathers	—	Partly regular – Pressure
OBN	Pre-stack	Receiver gathers	—	Non-regular – 4C
Land	Pre-stack	Shot gathers	—	—
Land	Pre-stack	OVT gathers	Post-migration	—
Land	Pre-stack	CDP gathers	Post-migration	Regularised

Table 3 Canonical seismic product taxonomy used in MDIO v1. The table summarises representative 2D, 3D, marine, land, and OBN products and classifies each by survey type, processing domain, gather type, migration stage, and regularisation status.

SEG-Y header scanning and writing, realises high throughput on the test machine, with linear scaling in wall time with the file size. We also observe that MDIO provides a consistent lossless data compression between 25%-39% despite explicitly storing coordinates and other information in the dataset.

Automated metadata reconstruction for large-scale SEG-Y ingestion

We aim to ingest a library of more than 1 million SEG-Y files into the MDIO v1 format. This is addressed through three coupled metadata reconstruction tasks:
(i) seismic product type identification,
(ii) header field extraction, and
(iii) schema mapping to standardised MDIO v1 fields.

Together, these steps define the minimum requirements for constructing MDIO v1 datasets with explicit dimensions, coordinates, and consistent metadata. The reconstruction process begins with the identification of the canonical seismic product type, which determines the dataset's high-level geophysical structure (e.g., 2D vs. 3D, pre-stack vs. post-stack, gather organisation). MDIO v1 formalises this classification using a controlled taxonomy spanning marine, land, and ocean-bottom node (OBN) surveys. A representative subset of this taxonomy is summarised in Table 3.

SEG-Y text headers contain critical metadata describing survey geometry, acquisition environment, processing history, and product semantics. However, these headers are free-form, inconsistently structured, and weakly standardised, often encoding essential information using non-standardised language and project-specific conventions. Accurate interpretation, therefore, typically requires subject-matter expertise (SME). As an example, in Table 4, we give an example of information defined in the text header, and how they inform the MDIO v1 schema.

Moreover, SEG-Y text headers frequently declare custom header overrides, such as non-standard byte locations for inline and crossline coordinates, which must be extracted and applied to correctly interpret trace headers. In addition, the same geophysical concept may be referred to in the text header using multiple different aliases over time (e.g., CDP, CMP, CMP NO.), requiring normalisation to derive a consistent metadata schema across the library. At the scale of a library with millions of SEG-Y files, manual interpretation of text headers is not feasible; an automated solution guided by subject matter expertise is required to fully populate.

SEG-Y Header Cue	MDIO Interpretation
cdp, angle, inline, crossline	MDIO dimensions
cdp_x, cdp_y, coordinate_scalar	MDIO coordinate fields
LINE = single value	MDIO: 2D survey
INLINE or XLINE present	MDIO: 3D survey
SOURCE = VIB/DYNAMITE	MDIO: land acquisition
"PSDM", "DEPTH MIGRATION"	MDIO: post-migration
"CMP gathers"	MDIO: pre-migration
"regularised", "binned", "resampled"	MDIO: regularisation flag
PRODUCT keywords	MDIO: product_type classification

Table 4 Mapping of key SEG-Y header cues — primarily from the text and binary headers — to their corresponding MDIO v1 interpretations.

To address these challenges, we constructed a labelled training and evaluation dataset derived from 57 manually interpreted SEG-Y files, each containing full text headers, binary header information, and SME-validated ground-truth annotations. Each session included canonical seismic product type labels, standardised header overrides, and unit definitions, providing authoritative reference data for product classification, field extraction, and schema mapping. Although modest in size, this dataset was intentionally curated to span heterogeneous acquisition types, processing workflows, and header conventions, encompassing approximately 800 distinct metadata labels representative of the broader seismic data library.

We then developed a multi-agent AI system to perform end-to-end metadata reconstruction suitable for MDIO v1 ingestion. The system leverages large language models within a structured pipeline that combines free-text reasoning with explicit domain constraints derived from SME guidance. All agents were implemented using the Anthropic Claude Sonnet 4 LLM. No model fine-tuning was performed; performance gains were achieved through agent decomposition, prompt refinement, rule encoding, and SME-in-the-loop iteration. The automated pipeline consists of three primary agents:

1. Template (product type) classification agent – this agent determines the canonical seismic product type consistent with the MDIO v1 taxonomy. It evaluates structural cues (e.g., presence of LINE versus INLINE/XLINE), acquisition indicators (e.g., SOURCE descriptors), and processing descriptions (e.g., migration and regularisation statements). Outputs are validated using rule-based consistency checks. This step also includes an additional verification agent to verify the predicted template class from the classification agent.
2. Field extraction agent – given the selected template, it identifies relevant metadata fields from SEG-Y text and binary headers, including field names, byte locations, data types, and semantic intent. Extracted fields are normalised into an intermediate representation independent of the original SEG-Y syntax.
3. Schema mapping agent – this agent maps extracted fields to standardised MDIO v1 metadata namespaces, resolving SEG-Y aliases and assigning fields to their appropriate roles (dimensions, coordinate variables, or auxiliary metadata). Fields that cannot be mapped unambiguously are explicitly flagged. This step also includes an additional verification agent to verify if the mapped alias from the mapping agent indeed matches the same concept as the extracted field in SEG-Y data.

Each stage is followed by an automated verification step enforcing internal consistency, domain constraints, and schema compatibility. Ambiguous cases and previously unseen patterns were routed

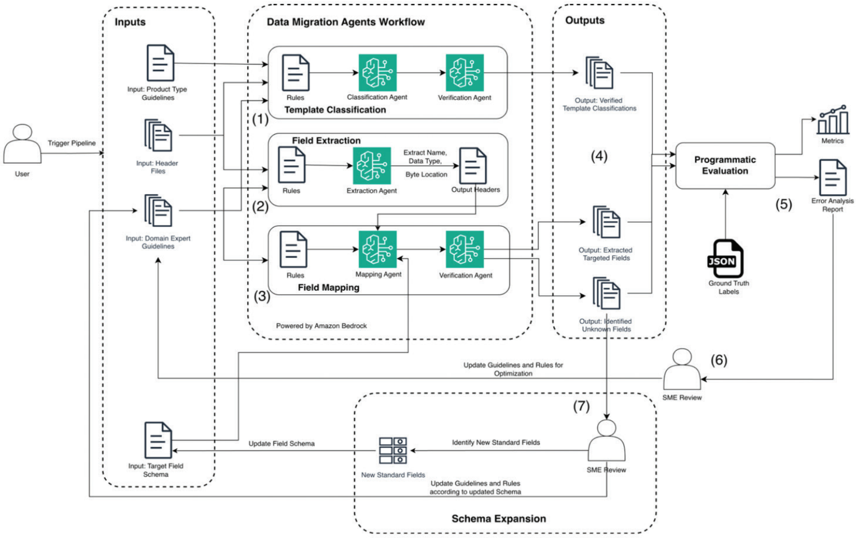

Inputs
Data Migration Agents Workflow
Outputs
User
Trigger Pipeline
Input: Product Type Guidelines
Input: Header Files
Input: Domain Expert Guidelines
Input: Target Field Schema
Rules
Classification Agent
Verification Agent
Template Classification
(1)
Field Extraction
Rules
Extraction Agent
Extract Name, Data Type, Byte Location
Output Headers
(2)
Rules
Mapping Agent
Verification Agent
Field Mapping
(3)
Powered by Amazon Bedrock
Output: Verified Template Classifications
(4)
Output: Extracted Targeted Fields
Output: Identified Unknown Fields
Programmatic Evaluation
Metrics
Error Analysis Report
(5)
Ground Truth Labels
JSON
Update Guidelines and Rules for Optimization
SME Review
(6)
(7)
Update Field Schema
New Standard Fields
Identify New Standard Fields
SME Review
Update Guidelines and Rules according to updated Schema
Schema Expansion

Figure 3 Conceptual multi-agent workflow for end-to-end SEG-Y metadata reconstruction.

Task	Description	Accuracy	Precision	Recall
Template classification	Identification of canonical seismic product type	95.00%	N/A	N/A
Field extraction	Extraction of header fields and attributes	99.17%	99.33%	99.50%
Field schema mapping	Mapping to MDIO v1 canonical fields	98.16%	98.16%	98.16%
End-to-end extraction and mapping	Correct extraction and correct schema mapping	96.83%	96.99%	97.16%

Table 5 Performance of agentic SEG-Y metadata reconstruction tasks.

through an SME review loop, enabling controlled refinement of prompts, rules, and schema definitions. A conceptual overview of this workflow is shown in Figure 3.

The system was evaluated end-to-end on the labelled dataset across three primary tasks: template classification, field extraction, and field schema mapping. In Table 5, we show that the performance was assessed using exact-match accuracy against SME-validated ground truth. In addition, we report an end-to-end field extraction and mapping metric, in which a field is considered correct if and only if it is both successfully extracted and correctly mapped to the MDIO v1 schema.

These results demonstrate that an agent-based AI system can interpret heterogeneous SEG-Y headers and reconstruct standardised metadata at or above human-level accuracy, while dramatically reducing manual effort. The system achieved low operational latency and low cost, making it viable for large-scale migration of seismic archives exceeding one million datasets. By combining SME knowledge with structured agent reasoning, the approach enables full end-to-end metadata reconstruction, transforming implicit SEG-Y information into explicit MDIO v1 datasets with first-class dimensions, coordinates, and standardised metadata.

Interactive analysis framework

Having established standardised, self-describing seismic datasets through MDIO and agent-driven metadata reconstruction, the remaining challenge is how this information is accessed and combined in practice. In subsurface analysis, seismic data is important, but we also depend on a broader context of well logs, technical reports, and commercial and operational records. We address this by introducing a chat-based analytical interface that enables natural-language queries to coordinate structured operations across seismic and related geoscientific and commercial data sources, without replacing existing systems.

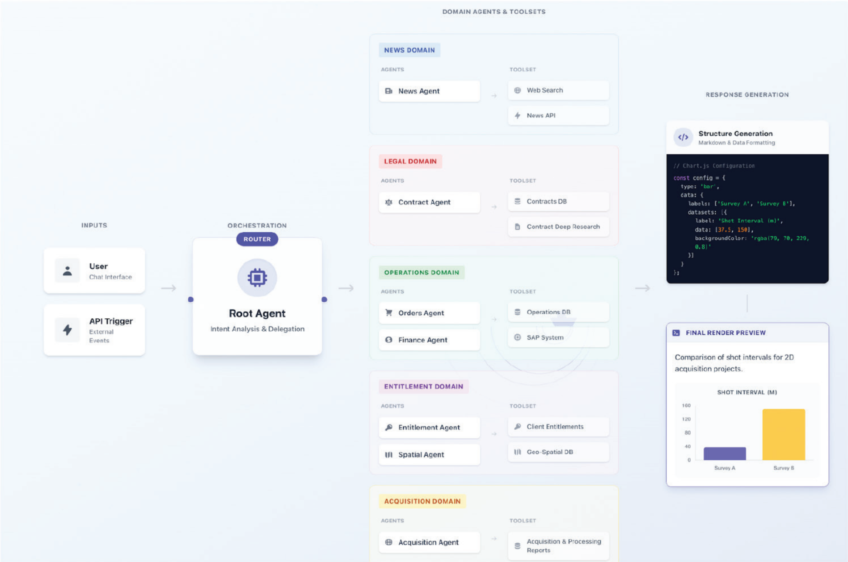
The system is implemented as a hierarchical multi-agent framework (Google 2025) shown in Figure 4 in which a root agent interprets user intent and decomposes queries into explicit, inspectable sub-tasks executed by domain-specific agents. Each agent encapsulates procedural domain knowledge and invokes constrained computational tools, such as database queries, spatial operations, or document retrieval, to ensure reproducible and consistent results. This architecture supports multi-step, cross-domain analysis while maintaining transparency, traceability, and alignment with subject-matter-expert workflows.

As an example, in Figure 5, a user asks: ‘Find 3D marine seismic surveys in the Gulf of Mexico’. The UI streams information about agents and tool calls for the user.

The interface responds conversationally by summarising the results of the query. For example, when asked about available data, it identifies several 3D marine seismic surveys in the Gulf of Mexico and presents a short, ranked list (e.g., Survey A, Survey B), while indicating that additional results are available and can be explored on request.

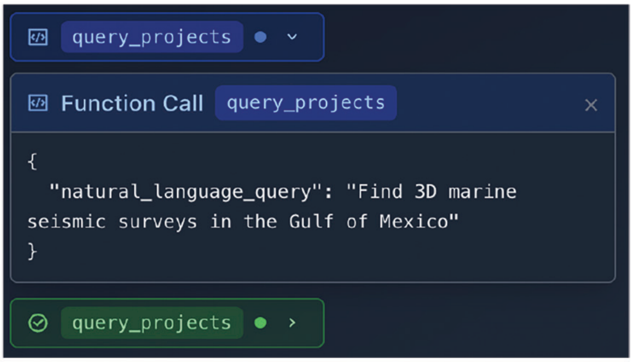
‘Of course, I can help with that. I found several 3D marine seismic surveys in the Gulf of Mexico. Here are a few of them:

- Survey A
- Survey B



```
// Chart.js Configuration
const config = {
  type: 'bar',
  data: {
    labels: ['Survey A', 'Survey B'],
    datasets: [{
      label: 'Shot Interval (m)',
      data: [37.5, 150],
      backgroundColor: 'rgba(79, 70, 229,
      0.8)'
    }]
  }
};
```

Figure 4 A hierarchical multi-agent system for intent decomposition and cross-domain subsurface analysis.



```
{
  "natural_language_query": "Find 3D marine
seismic surveys in the Gulf of Mexico"
}
```

Figure 5 The UI streams information that allows the user to understand the reasoning of the multi-agent system. In this case, the system is responding to a search request for 3D seismic surveys in the Gulf of Mexico.

There are many more. Please let me know if you would like to know more about any of these surveys.'

Interactive analysis is driven by a deterministic rendering protocol that allows agents to communicate through a visual medium. By emitting streamed markdown containing both narrative text and structured code blocks (e.g., Chart.js), the language model moves beyond text-only responses to 'express" via interactive visualidations (Figure 6). This architecture separates reasoning from execution: the model generates the visual specification, while the interface renders the final output, ensuring reproducible, interpretable, and scalable analysis across large geoscientific and commercial datasets.

High-performance retrieval on seismic acquisition reports

To enable efficient Retrieval-Augmented Generation (RAG) (Karpukhin et al., 2020) over a large corpus of seismic acquisition and processing reports, we implemented a semantic retrieval pipeline based on dense text embeddings using the Google Gemini embedding model (gemini-embedding-001).

Scaling retrieval-augmented generation (RAG) to large subsurface archives is constrained by the storage and indexing cost of high-dimensional embeddings, which can exceed the size of the original text. We use 3072-dimensional embeddings to preserve technical fidelity, exceeding the default dimensional limits of standard PostgreSQL vector indexing (PGVector 2025). This is addressed using half-precision vector quantisation (halfvec) with PGVector, enabling support for the full embedding dimension while reducing index size by approximately 50% and improving distance-computation performance.

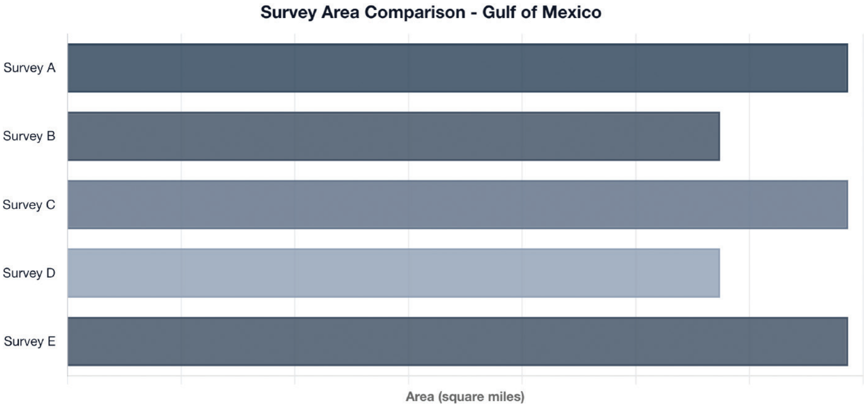


Figure 6 An auto-generated chart summarising the area of a collection of seismic surveys, prompt – 'visualise the survey area' with additional customisation available through the prompt – 'Implement a corporate-style blue-grey theme.'

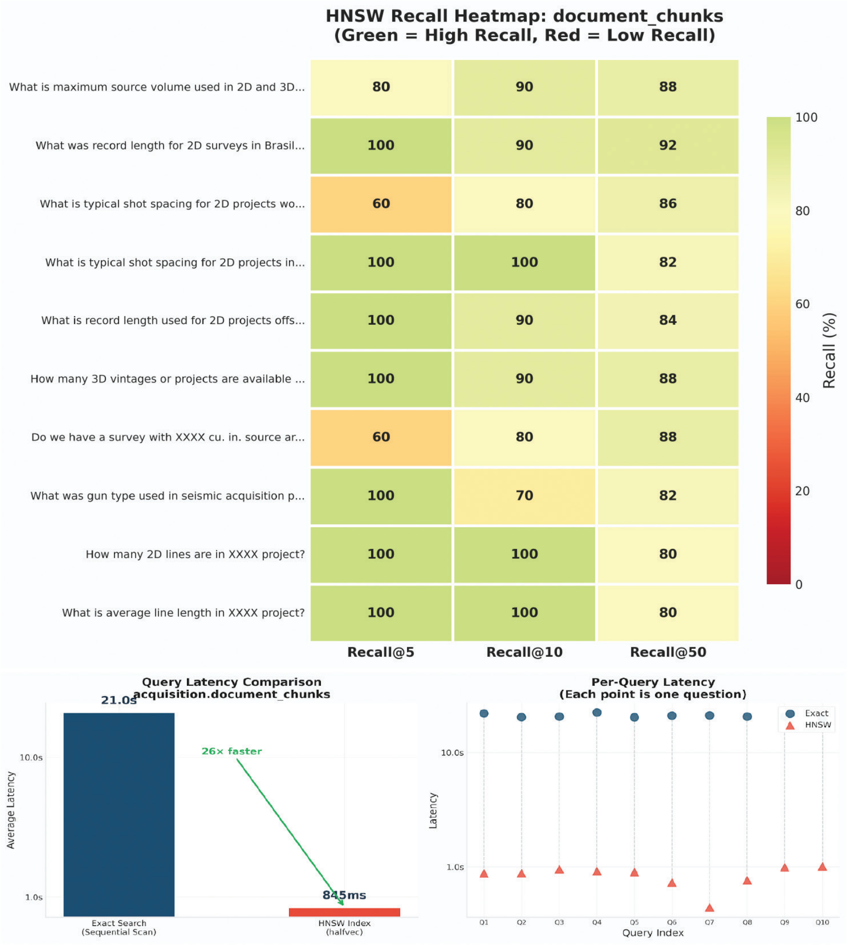


Figure 7 (top) Recall statistics for each of the questions in the acquisition evaluation set. (bottom) The performance impact of replacing exhaustive vector scanning with graph-based HNSW indexing at half precision.

Embeddings are indexed using the Hierarchical Navigable Small World (HNSW) graph algorithm, selected for its sub-second query latency and scalability to millions of vectors. Index parameters were tuned to favour high recall, ensuring accurate retrieval of domain-specific technical details. Although retrieval is executed within PostgreSQL, the system implements a semantic RAG pipeline rather than a Text-to-SQL approach, retrieving unstructured text fragments based on semantic similarity. This is well-suited to subsurface reports, where key technical information is often embedded in narrative text rather than normalised fields.

Evaluation was performed on a corpus of 1445 seismic acquisition and processing reports using a representative set of ten domain-specific queries. Retrieval performance was evaluated using Recall@K, defined as the fraction of top-K chunk results returned by a halfvec HNSW index that match the top-K results from an exact full-precision (FP32) exhaustive vector search. Documents were indexed at the chunk level, with 1000-character segments and 200-character overlap, yielding 242,312 embeddings (approximately 167 chunks per document) and preserving technical context across segments.

The query set used in the evaluation was constructed to represent typical subsurface information retrieval tasks:

1. What is the maximum source volume used in 2D and 3D surveys offshore?
2. What was the record length for 2D surveys in Brazil from 2000 to the present?
3. What is typical shot spacing for 2D projects worldwide?
4. What is typical shot spacing for 2D projects in Brazil?
5. What is the record length used for 2D projects offshore Canada?
6. How many 3D vintages or projects are available in the Santos Basin?
7. Do we have a survey with XXXX cu. in. source array in our library?
8. What was the gun type used in seismic acquisition projects?
9. How many 2D lines are in XXXX project?
10. What is the average line length in XXXX project?

As shown in Figure 7, the HNSW half-precision index maintains high retrieval fidelity despite aggressive quantisation. For approximately half of the queries (e.g., Q2, Q4, Q5, Q6, Q9, Q10), 100% recall was achieved at Recall@5, indicating that the most relevant document chunks consistently surfaced among the top results. More abstract or globally scoped queries, such as worldwide shot spacing (Q3) or specific source-array configurations (Q7), exhibited lower initial recall (60% at K=5). However, recall increased substantially to 86-88% at Recall@50. This behaviour is well aligned with RAG workflows, where retrieving relevant context within the top 20-50 chunks is sufficient to fully populate a standard LLM context window, ensuring that downstream answer generation remains robust.

Using HNSW half-precision indexing yields a 26× improvement in throughput, reducing average query latency from 21 s to 845 ms. As shown in Figure 7 (bottom), HNSW maintains stable, sub-second latency across all queries, whereas exact sequential scans exhibit consistently high execution times. This demonstrates that approximate indexing combined with half-precision quantisation effectively removes the latency bottleneck of exhaustive vector scanning, enabling interactive semantic retrieval over domain-specific acquisition reports at scale. While vector quantisation and HNSW indexing are well established in general information retrieval, these results provide an evaluation of their performance on a subsurface reports corpus. The observed performance gains and limited impact on retrieval accuracy support real-time retrieval-augmented analysis in energy-sector applications.

Conclusions

This work demonstrates that subsurface digitalization can be advanced by coupling self-describing seismic representations with tool-augmented, agent-based AI workflows. Together, these components enable large-scale conversion of legacy SEG-Y archives into explicit, machine-readable datasets and support accurate reconstruction of standardised seismic metadata, providing a practical foundation for scalable analysis and AI-assisted workflows.

An interactive, agent-based interface brings together seismic data, acquisition and processing documents, contracts, operational reports, and related metadata through a deterministic, markdown-based protocol that separates language-model reasoning from data execution. Within this interface, embedding-based semantic retrieval enables efficient natural-language search over a corpus of subsurface-specific documents. Benchmarking shows that half-precision vector quantisation combined with HNSW indexing delivers substantial reductions in query latency with minimal impact on retrieval quality, enabling interactive use at scale. While these techniques are well established in general information retrieval, their application and performance characteristics for subsurface acquisition and processing reports are evaluated here.

References

Gonzalez, K., Sansal, A., Valenciano, A. and Lasscock, B. [2025]. Well log foundation model – Making promptable AI models for interpretation. IMAGE 2025, *Proceedings*.

Gonzalez, K., Sylvester, Z., Valenciano, A. and Lasscock, B. [2024]. From well logs to 3D models: A case study of automated stratigraphic correlation in the Midland Basin. IMAGE 2024, *Proceedings*, 10.1190/image2024-4101544.1.

Google. [2025]. Agent Development Kit (ADK) Documentation. Google AI. Retrieved from https://google.github.io/adk-docs/.

Hoyer, S. and Hamman, J. [2017]. Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, **5**(1), 10.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W.-T. [2020]. Dense Passage Retrieval for Open-Domain Question Answering. In 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), *Proceedings*, 6769–6781.

Michell, B., Sansal, A., Lasscock, B. and Roberts, M. [2025]. MDIO v1: Schematizing seismic data for AI and processing. IMAGE 2025, *Proceedings*.

PGVector. [2025]. Open-source vector similarity search for Postgres (Version 0.8.x) [Computer software]. https://github.com/pgvector/pgvector.

PGVector. [2025]. HNSW index support. GitHub documentation. https://github.com/pgvector/pgvector?tab=readme-ov-file#hnsw.

Sansal, A. [2025a]. mdio-python: Python library for the MDIO multidimensional energy data format (Version 1.1) GitHub https://github.com/TGSAI/mdio-python.

Sansal, A. [2025b]. *segy:* The Ultimate Python SEG-Y I/O with Cloud Support and Schemas (Version 0.4.1.post2). https://github.com/TGSAI/segy.

Sansal, A., Lasscock, B. and Valenciano, A. [2023a]. MDIO: Open-source format for multidimensional energy data. *The Leading Edge*, **42**(7), 465–473.

Sansal, A., Lasscock, B. and Valenciano, A. [2023b]. Integrating energy datasets: the MDIO format. *First Break*, **41**(10), 69-75.