Designing Generative AI solutions

When approaching a Generative AI solution, it is essential to establish some foundational concepts and considerations that impact how we design and implement a solution with Fabriq. These considerations, particularly around advanced information retrieval, highlight why certain technical choices are not only beneficial but essential for optimising search accuracy, relevance, and flexibility.

Selecting the most effective search methodology is highly dependent on the specific types and structures of data involved. Traditional keyword searches, while widely understood and useful for direct, exact-term matching, are limited when nuanced or contextual understanding is needed. They may fall short if the terms do not precisely match the language of the data source, especially in unstructured data environments where the information lacks detailed indexing. In these cases, more advanced approaches—such as semantic search and natural language processing (NLP) techniques—become invaluable, allowing for richer interpretations and a broader range of relevant results.

Specialised language within different industries or domains often carries unique meanings. For instance, in oil and gas, the term “fault” can denote a geological structure significant to resource exploration, whereas in manufacturing, it indicates a flaw or error. Adapting the search methodology to recognise and interpret these domain-specific terms ensures higher accuracy and relevance. Using distinct AI models trained for specific domains can also enhance semantic understanding, especially when domain-specific embeddings are incorporated to maintain high interpretive accuracy.

Traditional, Keyword based search

In implementing a keyword-based search, it is essential to consider the various technical models and algorithms that enhance search relevance and retrieval precision available in the Fabriq platform. Traditional keyword search can leverage several ranking and retrieval algorithms, each designed to interpret and prioritise keyword matches in different ways. The following approaches highlight the strengths and potential applications of each technique.

Search	Description	Pros	Cons	Use Cases
TF-IDF	Weighs terms by frequency in a document and inversely by frequency across all documents.	Simple and fast to implement; effective for keyword matching.	Does not handle synonym or document semantics well; sensitive to term frequency bias.	Good for basic keyword search, smaller datasets, or simple information retrieval tasks.

BM25	Builds on TF-IDF by adding document length normalization and term saturation.	Provides better ranking for longer documents; more effective than TF-IDF in most IR tasks.	Requires parameter tuning (e.g., `k1`, `b`); not well-suited for capturing semantic meaning.	Web search engines, information retrieval in large text corpora, where document length varies significantly.
Semantic Search	Uses embeddings / sentence transformers to capture contextual meaning of queries and documents.	Handles synonyms and contextual meanings; provides deep semantic matching.	Computationally intensive; requires a large dataset and often substantial hardware resources.	Ideal for natural language search, customer support systems, and when matching by intent rather than exact keywords is critical.
Hybrid Search	Combines traditional methods (e.g., TF-IDF/BM25) with semantic embeddings.	Balances exact keyword matching with contextual understanding; flexible and highly effective.	Higher computational requirements than traditional methods; can be complex to implement.	Required for domain specific implementations of conversational LLM search

Traditional keyword-based approaches, supported by advanced models like BM25 and TF-IDF, offer robust methods for relevance-ranking and data retrieval in diverse datasets. These techniques can be selectively combined with hybrid search methods to leverage both exact and contextually relevant retrieval as needed. Together, these approaches ensure that the solution supports fast, accurate, and context-sensitive search experiences across its vast and varied data sources.

Advanced implementation considerations, keyword search

In addition to the foundational methodologies outlined above, there are further implementation considerations and optimisations that can enhance the efficacy of keyword-based search. Below, we discuss several additional components and refinements that can contribute to building a responsive, scalable, and highly relevant search infrastructure. Implementing these more advanced methods adds complexity and should be considered only if the Input Data requires it.

Query Expansion and Synonym Recognition

Description: Query expansion is a technique used to improve search recall by adding related terms or synonyms to the original search query. This can involve expanding abbreviations, identifying synonyms, and using domain-specific terminology libraries. By recognising and including alternative expressions, query expansion helps bridge language gaps, particularly when users may not use exact terms found in the database.

Application: Query expansion can be applied to incorporate industry-specific synonyms and commonly used technical terms, enabling users to search effectively even if they are unfamiliar with precise jargon. For example, a query for "geo-energy" could also retrieve documents related to "geothermal energy," "georesources," and related phrases, ensuring broader and more inclusive results. This technique is often combined with Hybrid Search (see above).

Advantages:

- **Improved Recall**: Increases the likelihood of retrieving relevant documents by broadening the scope of the query. This is particularly useful when the domain specific term is not represented in the vector space of the embeddings model.
- **User-Friendly**: Reduces dependency on exact keywords, benefiting both technical and general users.

Limitations: Unchecked query expansion can lead to too many irrelevant results, especially in extensive datasets, making precise tuning and domain-specific filtering necessary. It is nevertheless a powerful tool when used in the right context.

Fuzzy Matching

Description: Fuzzy matching allows the search system to handle minor typographical errors, spelling variations, or slight differences in phrasing. This technique can be particularly beneficial when dealing with user-entered search terms, as it helps match queries that would otherwise miss relevant content due to minor errors or alternate spellings.

Application: Fuzzy matching can enhance user experience by returning results even when users accidentally mistype or use regional spelling variations (e.g., "geothermic" instead of "geothermal"). The fuzzy matching tolerance can be adjusted based on the nature of the search and dataset, allowing for flexibility between precision and inclusiveness.

Advantages:

- **Error Resilience**: Reduces friction for users by accommodating minor errors, improving search usability.

- **Flexible Matching**: Allows for broader capture of relevant results, especially useful in free-text search fields.

Limitations: Excessive tolerance can reduce precision by including too many loosely related results, so optimal configurations are essential for balance.

Filters and Faceted Search

Description: Faceted search allows users to filter search results by specific attributes (such as document type, publication date, author, or topic area) after an initial query is made. Contextual filters also enable the pre-selection of criteria to narrow down results before initiating a search, offering users greater control over their search parameters.

Application: Faceted search could support filtering by various metadata fields, such as geological regions, study types, or document categories. This would enable users to quickly refine search results based on specific interests, ensuring a streamlined and tailored search experience.

Advantages:

- **Enhanced Relevance**: Enables users to focus on specific aspects of the dataset, quickly finding content that aligns with their needs.
- **User Control**: Provides an intuitive, guided search process, especially valuable in large datasets.

Limitations: Faceted search depends heavily on comprehensive and consistent metadata tagging, which may require regular updates to stay relevant as the database grows. Faceted search will require features in the end user interface, and possibly a change to the ingested data (especially if the needed metadata must be generated or derived during ingest).

Proximity-Based Search

Description: Proximity-based search ranks results based on the closeness of keywords within documents. By prioritising documents where search terms appear close together, this method can enhance the relevance of search outcomes, particularly for queries where relationships between terms matter (e.g., “thermal conductivity” near the “15/9 B 24 well”).

Application: For example, a user searching for "fault line geothermal potential" would benefit from proximity-based search that prioritises documents discussing fault lines and geothermal potential in close relation, reducing unrelated content where terms appear separately.

Advantages:

- **Enhanced Precision**: Captures more relevant documents by focusing on the proximity of key terms within texts.
- **Contextual Retrieval**: Improves retrieval accuracy for compound queries, ideal for technical searches where terms are interrelated.

Limitations: Proximity search may require higher computational resources and is best suited to highly relevant contexts to avoid slowing down the search experience. Constructing the queries can take time for the user, typically not a feature that receives adoption outside of very narrow use cases. The expansion must also be communicated to the user visually, to explain the inclusion of results they did not explicitly request.

Summary of Traditional Keyword-Based Search Enhancements

Combining these advanced techniques will allow users to achieve precision and relevance across a broad array of data. From basic Boolean filtering to sophisticated hybrid matching, these methodologies ensure robust, user-friendly search experiences that cater to both general inquiries and specialised, technical research needs. With careful use of these technologies, the solution will remain responsive to the evolving demands of its users, supporting them in retrieving high-quality information in a complex domain with diverse data types. To ensure the solution remains fit for purpose now and into the future, it is important to have the option to support these advanced retrieval technologies.

Conversational, LLM-based search

Conversational search powered by Large Language Models (LLMs) is a cutting-edge approach that enables users to interact with the database in a conversational manner, asking complex, natural language questions and receiving detailed responses. Unlike traditional keyword-based methods, conversational search leverages the advanced understanding and generative capabilities of LLMs to interpret user intent, handle ambiguous queries, and respond with relevant, contextually aware information. This approach is especially valuable for exploring unstructured data sources, where users may not know specific keywords but need insights based on general topics, trends, or inferred meanings.

Key Features of Conversational, LLM-Based Search

Natural Language Understanding
LLMs excel at interpreting the meaning and intent behind user queries, even when phrased in complex, colloquial, or domain-specific language. This ability enables the platform to accommodate questions that go beyond simple keyword matching, as the LLM can understand nuances, synonyms, and related concepts.

Context Retention and Follow-Up Questions
Conversational search enables users to engage in multi-step interactions, where the system retains the context of previous questions and answers. This continuity allows for a more intuitive

experience, as users can ask follow-up questions without repeating prior details, and the LLM understands references to earlier parts of the conversation.

Dynamic Query Expansion
Through its extensive language understanding, an LLM can dynamically expand on user queries, broadening or refining search terms to capture relevant information that may not directly match the initial keywords. This is particularly useful for diverse data sources, as the model can interpret general questions and seek relevant documents or sections across structured and unstructured content.

Technical Approaches and Methods

Embeddings for Semantic Understanding
The sentence transformers create tensors to represent words, phrases, and documents in a high-dimensional space where numerical representations of semantically similar items are positioned closely together. Semantically relevant records from trusted documents allows the vector store to interpret a user query in relation to other similar terms and concepts, enabling retrieval that captures the broader meaning or intent behind the search.

Application: Embedding-based search enables the platform to effectively interpret and locate related terms or phrases within unstructured datasets. For example, if a user asks, “What are the latest findings on geothermal energy?” the model can interpret this as a broad query, retrieving documents with relevant terms like “geothermal studies,” “renewable energy,” and specific research topics, even if “findings” is not explicitly mentioned.

Retrieval-Augmented Generation (RAG)
In retrieval-augmented generation, the system creates a grounded response from the large language model by combining trusted documents with prompts, examples and instructions for the LLM. The RAG approach retrieves a set of highly relevant documents from the database based on keywords or embeddings, then synthesises responses by summarising and integrating content from these documents.

Application: This approach is ideal for extensive academic and technical documents. The model can retrieve relevant sections and generate summaries, giving users concise yet informative answers. For instance, in response to “Explain fault detection methods in geothermal wells,” the system can generate a synthesised answer based on relevant technical documents, combining detailed sources into a clear explanation.

Conversational Context
Conversational LLM-based search relies on ranking mechanisms that consider relevance not just to the user’s immediate question but also to the ongoing conversational context. The scoring mechanisms can be configured to prioritise certain document types or sources based on situational relevance, improving the accuracy of responses over a sustained interaction.

Application: When users ask follow-up questions or clarify their original query, the system responds based on the cumulative context of the conversation. For example, if a user starts by asking about “environmental impacts of geothermal energy” and follows up with “What are common mitigation methods?” the system understands this as an extension of the initial query, retrieving documents focused on mitigation within geothermal projects.

Hybrid Search

Hybrid search extracts domain specific Named Entities from a user query and uses these entities as a keyword search, while the remainder of the query is a vector search.

Application: This approach enhances relevance by precisely identifying key entities, like wellbore IDs or chemical names, and matching them directly while interpreting the broader context of the query through vector search. This is particularly useful in industries like energy or medical, where queries often contain critical identifiers alongside broader informational needs, enabling efficient retrieval from both structured and unstructured data sources.

Advantages of Conversational, LLM-Based Search

- **Enhanced User Experience**: With its natural language capabilities, conversational search offers a more intuitive and user-friendly experience, especially for non-experts or users unfamiliar with specific keywords.
- **Greater Flexibility**: LLMs can adapt to a wide range of query types, from broad conceptual questions to highly specific technical inquiries, enabling flexible, exploratory interactions.
- **Better answers with citations**: By understanding the semantic relationships within user queries, conversational search achieves more relevant and comprehensive results, especially when exploring complex or nuanced topics.

Key Considerations for Implementation

1. **Data Privacy and Open Access**
    a. Conversational search must respect data privacy protocols while allowing for a flexible user experience. To ensure privacy, user conversation histories can be handled by requiring registered accounts for conversational access, allowing users to control their own data and delete histories as needed.
2. **Moderation and Safe Use**
    a. Conversational interactions introduce the potential for misuse, and the Fabriq platform includes moderation capabilities to detect and block inappropriate or unacceptable content. These measures will help maintain a professional environment for all users accessing the conversational interface.

Summary of Conversational Search

Conversational, LLM-based search represents a powerful enhancement to any Generative AI solution, and offer dynamic, natural language capabilities that cater to both general and technical users. By supporting conversational interactions, context retention, and retrieval augmented generation, this approach ensures that users can engage with the database in a highly interactive, flexible, and insightful manner. Special attention should be paid to Data Privacy and Moderation, and best practise would be that this capability is made available to registered users only. There are a long range of methodologies to examine when implementing conversational, LLM based search ranging from the mundane to the hyper parameter. One key component to being successful in this area is to employ the correct methodologies at the correct place based on the data and the user requirements.

Conversational search, Agentic Workflows

Domain specific Retrieval Augmented Generation, as described above, still has some limitations. Consider a relatively simple question such as “what are the headlines of the 5 most recent articles from the Norwegian Offshore Directorate’s news feed?”. Retrieval Augmented Generation will *not* automatically generate a filter for Faceted Search (see above), nor will it sort the results by one of the properties of these facets.

Conversational Search with Agentic Query Expansion

Similar to synonym generation for keyword search, this technique can be applied to RAG. The Named Entities in the user query can be looked up in domain specific databases, then these expanded terms can be used for retrieval.

Advantages: Inject otherwise hidden domain knowledge into the query that even an expert user may not be aware of.

Limitations: Not all users want all queries expanded with all entities at all times, but offering the choice increases user friction.

Application: This approach is useful for domains where finding hidden information has high business value. Consider this example:

- Before: “Did any wellbores near 34/6 2 S find HC in the same formation? Tell me about them”
- After: “Did any wellbores near 34/6 2 S find HC in the same formation? Tell me about them (NB: 34/6-2 S found hydrocarbons in the Cook Formation, and nearby wellbores that also found hydrocarbons in this formation are 34/4-5, 34/5-1 S, 34/6-2 A, 34/6-3 A, 34/6-3 S, 34/6-5 S)”

FABRIQ

Did any nearby wellbores find HC in the same formation? Tell me about them (Hint: 34/6-2 S found hydrocarbdons in the Cook Formation, and nearby wellbores that also found hydrocarbons in this formation are 34/4-5, 34/5-1 S, 34/6-2 A, 34/6-3 A, 34/6-3 S, 34/6-5 S)

20

Certainly, several nearby wellbores also encountered hydrocarbons in the Cook Formation. Here's a summary of each:

1. **34/4-5**: This well encountered hydrocarbons in the Cook Formation. The Cook Formation was found to be oil-bearing with a 20-meter oil column. The reservoir properties were good, with an average porosity of 22% and permeability up to 500 millidarcies. The well was classified as an oil discovery.
2. **34/5-1 S**: The primary well bore 34/5-1 S proved a hydrocarbon-bearing column within the Cook Formation in 64 meters TVD of interbedded sandstones and siltstones. The Cook Formation was the target for the sidetrack as well. However, the appraisal sidetrack 34/5-1 A encountered the Cook Formation with worse reservoir properties than the main bore and did not show hydrocarbons.
3. **34/6-2 A**: This well was a sidetrack from 34/6-2 S, drilled to further delineate the Cook Formation. It confirmed the presence of hydrocarbons and helped to better understand the extent and characteristics of the reservoir.
4. **34/6-3 A**: This well encountered a significant oil column in the Cook Formation with good reservoir quality. The well was tested and flowed oil at commercial rates, confirming the productivity of the reservoir.
5. **34/6-3 S**: This well also found hydrocarbons in the Cook Formation. The reservoir quality was good, and the well contributed to the delineation of the hydrocarbon accumulation discovered by the 34/6-2 S well.
6. **34/6-5 S**: This well encountered hydrocarbons in the Cook Formation with good reservoir properties. The well was tested and confirmed the presence of movable hydrocarbons.

These wells collectively contributed to the understanding of the hydrocarbon potential in the Cook Formation in the area, helping to assess the viability of development and production.

To expand the query several steps were required:

1. Identify tokens in question that could be wellbores.
2. Lookup database of wellbores to see if we get a match.
3. If yes, use the latitude and longitude of the matching wellbore as the centre point of a polygon with the desired radius.
4. Lookup database of wellbores to see if this one has HC shows in a formation.
5. Lookup database to see if any of the nearby wellbores have HC shows in the same formation.

Conversational search with Agentic Faceted Filtering

This approach provides a filter to the semantic search element of Retrieval Augmented Generation. It relies on an extra LLM agent to extract the Named Entities, and another to create a filter that is appropriate for the Structured Data available on the Unstructured Data in the vector store. This structured data is created on ingest and is typically date, genre, tag, topic, author etc. These are all examples of structured metadata that could be used to narrow the conversational search.

Application: Resolve queries such as “show me the recently published articles”

Advantages: Keeps the user interface clean, as traditionally this type of filter was done with Faceted Search (see above) and reduces user friction by allowing a natural language version of the query

Limitations: It can be time consuming designing agentic filtering workflows, and this approach is less reliable than the buttons and sliders that they seek to replace.

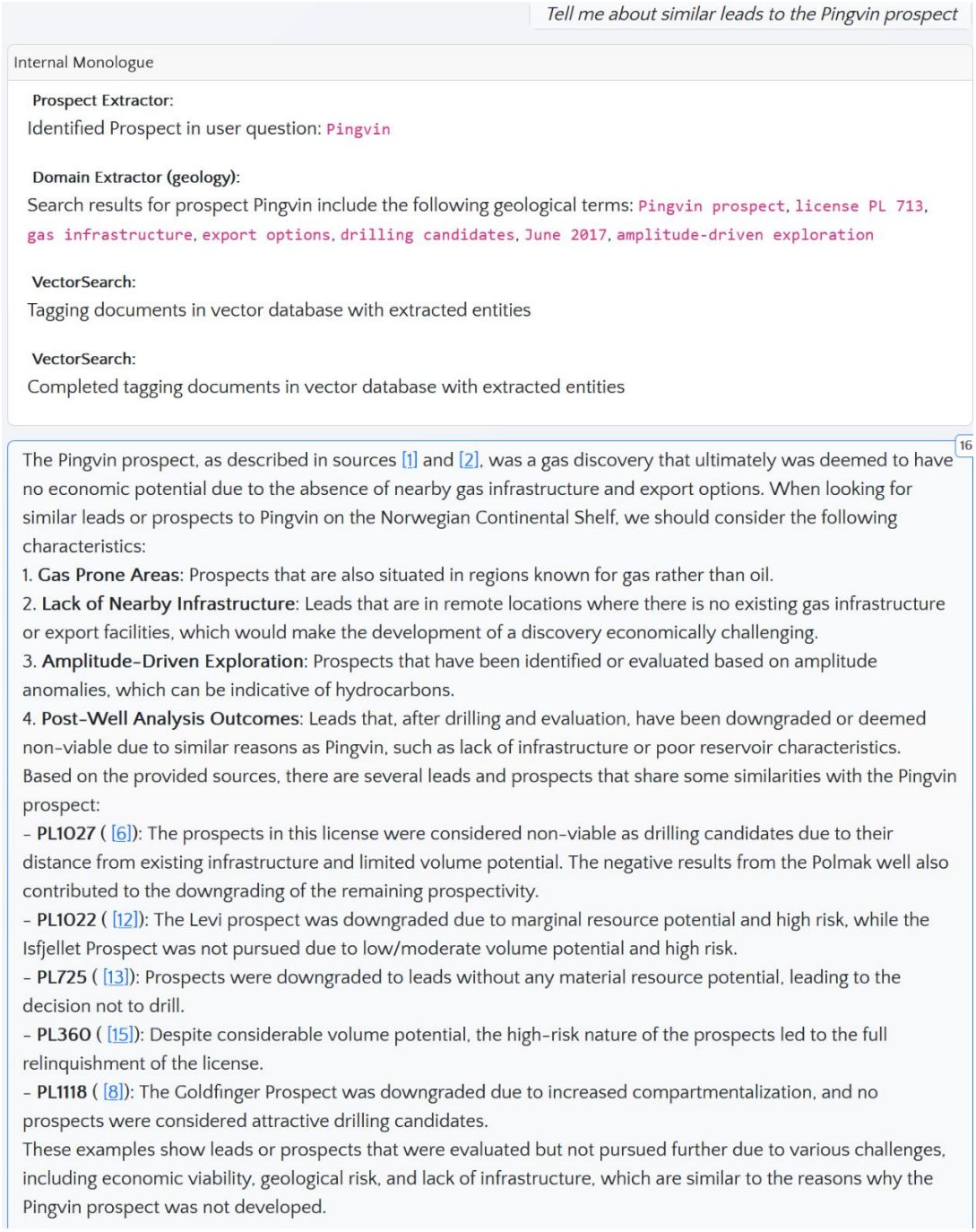
Conversation search with domain specific entity comparison

Conversational search is an effective tool for semantic retrieval then summarisation of these retrieved results. Some scenarios call for the retrieval of a particular domain entity, then comparison to another. Consider how one might identify case law that has similarity to a specific case, or how one might identify contracts that have a similar risk profile to a specific contract. Domain specific entity comparison can be a powerful tool in such situations.

Application: Geological analogues, legal analogues, contractual risk analogues. See the image below for an example of finding geologically similar Prospects. First the prospect in the question was identified and an answer is returned (this is standard conversational search). Then, the domain specific terms from the answer are extracted and used as a filter in a subsequent hybrid search (see above).

Advantages: Can reveal hidden information within a large corpus and use domain specific business logic for this retrieval. Useful when there is high business value in such retrieval.

Limitations: Takes some time to create the agentic workflows, and a workflow for each entity type is required



Summary of Conversational, Agentic workflows

Agentic workflows are a powerful tool that can allow industry to extract value from their structured and unstructured data. These techniques are designed to overcome specific

limitations with vector search, generalised embeddings models and generalised language models. We expect that the state-of-the-art models will remain industry agnostic, so there will be an ongoing need to apply domain specific retrieval technology. Most projects do not require such advanced retrieval methods, but readers should be aware of them in case the right use case can be found within their organisation.

Technical considerations

An effective platform for advanced keyword- and conversational search with NLP and LLM capabilities suitable for enterprise production environments must support robust server-side capabilities on enterprise grade technology. This platform will form the backbone for these capabilities in the Generative AI solution, enabling efficient, secure, and high-performance management of large-scale text processing, language understanding, and domain-specific search. Additionally, the platform must meet enterprise requirements for user management, data privacy, cloud integration, and other operational needs. This space in technology is very new, constantly evolving, and saturated with hype and buzz words. The choice between building a custom GenAI platform or leveraging Fabriq involves several critical trade-offs:

Here are the core considerations we have made for this RFP response;

- **Fit for Purpose**: The middleware solution should align closely with the specific functional and business needs of the organisation. This involves ensuring the platform is not only capable of handling generic NLP and LLM tasks but is also adaptable to the unique requirements of the organisation's domain, use case(s), and user base.
- **Futureproofing and Adaptability**: The middleware should be adaptable to emerging technologies and industry trends. With the rapid evolution of NLP and LLM capabilities, choosing a solution that allows for easy updates and integrations ensures long-term viability.
- **API Accessibility and Integration**: The middleware should offer a robust API framework to facilitate seamless integration with other enterprise systems. Scalable, well-documented APIs allow the platform to serve various applications while ensuring extensibility.
- **Scalability and Performance**: Middleware must handle high-concurrency environments with options for horizontal scaling. It should also be compatible with cloud environments but ideally without being dependent on a single cloud provider.
- **Comprehensive Monitoring and Analytics**: Built-in monitoring tools and analytics are essential for tracking performance, user engagement, and system health. These insights enable proactive maintenance and continuous improvement of the middleware's capabilities.
- **Cost and Licensing**: Cost of licensing should be evaluated up against cost of building, including operating cost over time. Ideally, a third-party provider should have a predictable price model that is easy to understand and budget for.

- **User Management:** Middleware should support comprehensive user management, allowing seamless integration with single sign-on (SSO) systems for unified user authentication and identity management. Role-based access controls and permission settings are essential to protect sensitive data and ensure that only authorised personnel can access certain functionalities or data segments. Ideally user authentication and authorisation should align with industry standards such as OAuth and OpenID Connect and cater for federation across identity providers.
- **Security and Data Privacy**: Built-in protections against common security threats, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), are essential for safeguarding user data. Additionally, enterprise-grade middleware should offer data encryption.
- **Cloud Integration and Deployment Options**: The middleware must support flexible deployment in public, private, or hybrid cloud environments. Compatibility with major cloud providers (AWS, Azure, Google Cloud) ensures scalability, and high availability while enabling organisations to leverage cloud-native security and scaling capabilities.

By carefully evaluating these factors, one can make informed decisions about the middleware solution that best align with strategic goals and operational requirements. Selecting the appropriate approach for your organisation not only enhances operational integrity and ensures compliance but also positions the organisation to adapt to future technological advancements.

Database Technology considerations

Ensuring use of the most suitable database technologies for any GenAI platform is of pivotal consideration. An effective solution for website, keyword- and conversational search both requires support for multiple databases technologies to handle both structured and unstructured data efficiently within the different components the solution will be made up of.

Considerations taken when looking at suitable database technologies;

- **Relational Databases for Structured Data**: Structured data is best stored in traditional relational databases, which support schema-based organisation, indexing, and SQL-based querying. These databases are effective for managing structured data from logs, user records, or predefined data tables, where consistency and normalisation are important.
- **Vector Databases for Semantic Search**: Vector databases are particularly effective for managing unstructured data in applications that require semantic search capabilities. By representing documents, text, and multimedia as high-dimensional embeddings, vector databases enable similarity-based retrieval, which allows for more intuitive and context-aware search results beyond traditional keyword matching. This approach is ideal for applications needing advanced text processing and content recommendations, such as conversational AI, research tools, and document discovery.

- **NoSQL Databases for Flexibility and Scalability**: NoSQL databases are well-suited for storing unstructured and semi-structured data due to their flexible schema designs. Unlike relational databases, NoSQL databases can handle evolving data formats without requiring extensive schema updates, making them ideal for dynamic and complex datasets. Their scalability also supports high-performance retrieval and storage, particularly useful for applications with large volumes of diverse data types, such as text documents, multimedia, or log data.
- **Data Integration Strategy**: In cases involving diverse data sources, the middleware should support both schema-based and schema-less data integration strategies. Schema-based integration offers consistency, while schema-less integration provides flexibility, allowing for dynamic adaptation as new data sources or formats are introduced.
- **Security and Compliance in Data Handling**: As with other middleware components, database technology should offer encryption, access control, and compliance with data privacy standards. Role-based access, logging, and monitoring features help maintain compliance and ensure data integrity.

By combining the strengths of relational, non-relational and vector databases, one can support seamless and accurate data retrieval across both structured and unstructured data sources as well as the rigid data requirements of a Web Platform. This ensures the platform's continued suitability for advanced search and NLP capabilities.

Given the scarcity of mature frameworks for conversational search and agentic workflows, and the scarcity of the expertise to create and support them, many organisations choose to externalise this risk into a managed SaaS middleware component. As the vendor of such a component, we recommend this approach and are confident that Fabriq will be the right choice for your organisation.