OpenAI

Recommendations

Accelerator

July 2025

How to use this Accelerator

This accelerator offers a robust starting point for anyone looking to quickly ramp up to an intermediate (201-level) understanding of the tools, methodologies, and outcomes involved. You'll gain practical insights from proven reference architectures. To help you take the next step, we've curated a selection of additional resources tailored to deepen your expertise.

- Showcasing the options you have to build this solution
- Giving you a high-level approach to know where to start
- Showing you how to approach next steps

Overview

Recommendation systems analyze user behavior, preferences, and/or input to suggest relevant items, services, or content. They provide personalized recommendations to enhance user experience and engagement.

Example use cases

- Content recommendation
- Personalized shopping experience
- Travel planner

What we'll cover

- Value Proposition
- Building:
  - Process flow
  - High-level architecture
  - Production examples
  - Features & limitations
  - Key challenges & pitfalls
  - Solution Deep Dive
- Success & Takeaways

Value Proposition

Value Proposition

Increased Engagement

By surfacing content or products tailored to individual preferences, recommendation systems keep users actively involved and returning more frequently. Personalized experiences create a sense of relevance, which drives longer session times and higher interaction rates.

Revenue Growth

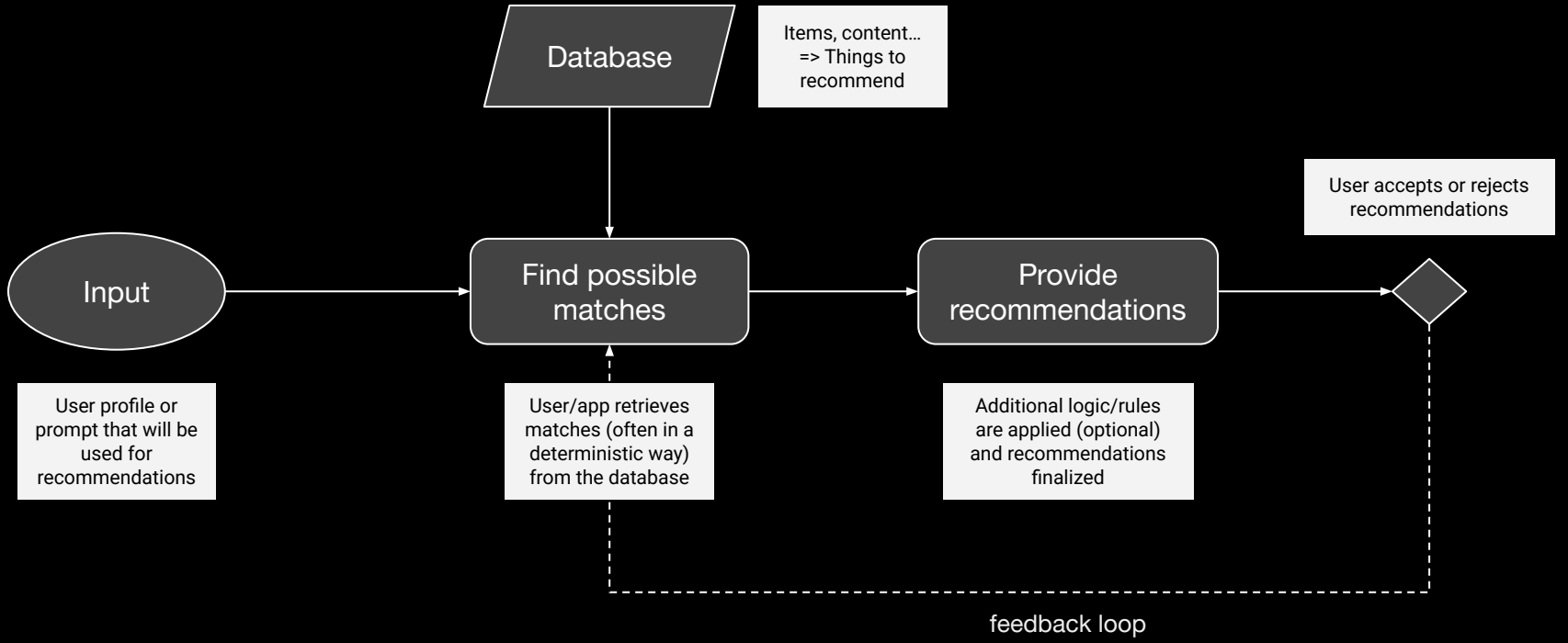
Recommendation engines can significantly boost conversion rates by presenting users with items they are more likely to purchase or explore. Whether it’s upselling, cross-selling, or just surfacing overlooked options, smart suggestions directly impact the bottom line.

Customer Satisfaction

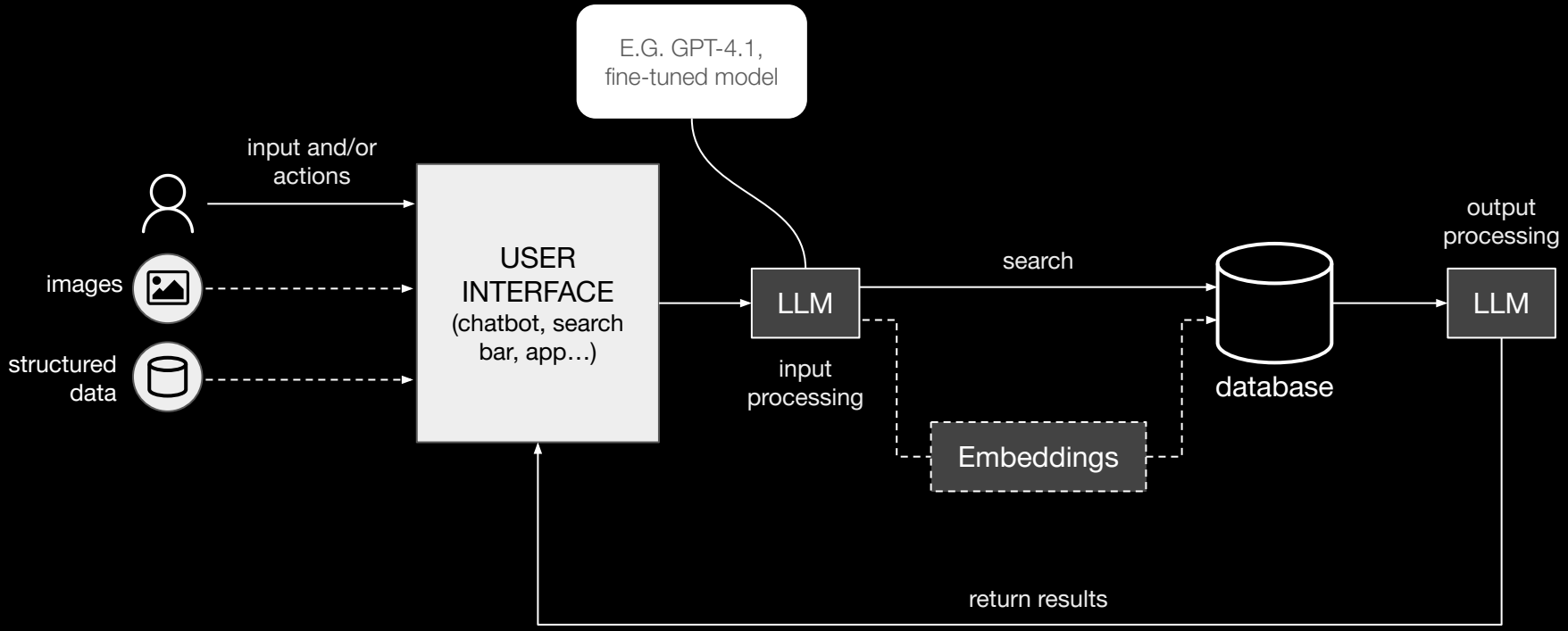
Offering personalized suggestions enhances user satisfaction by reducing friction in discovery. When users feel understood and supported in their journey—whether browsing content, shopping, or planning travel—they’re more likely to trust and stick with the platform.

Building

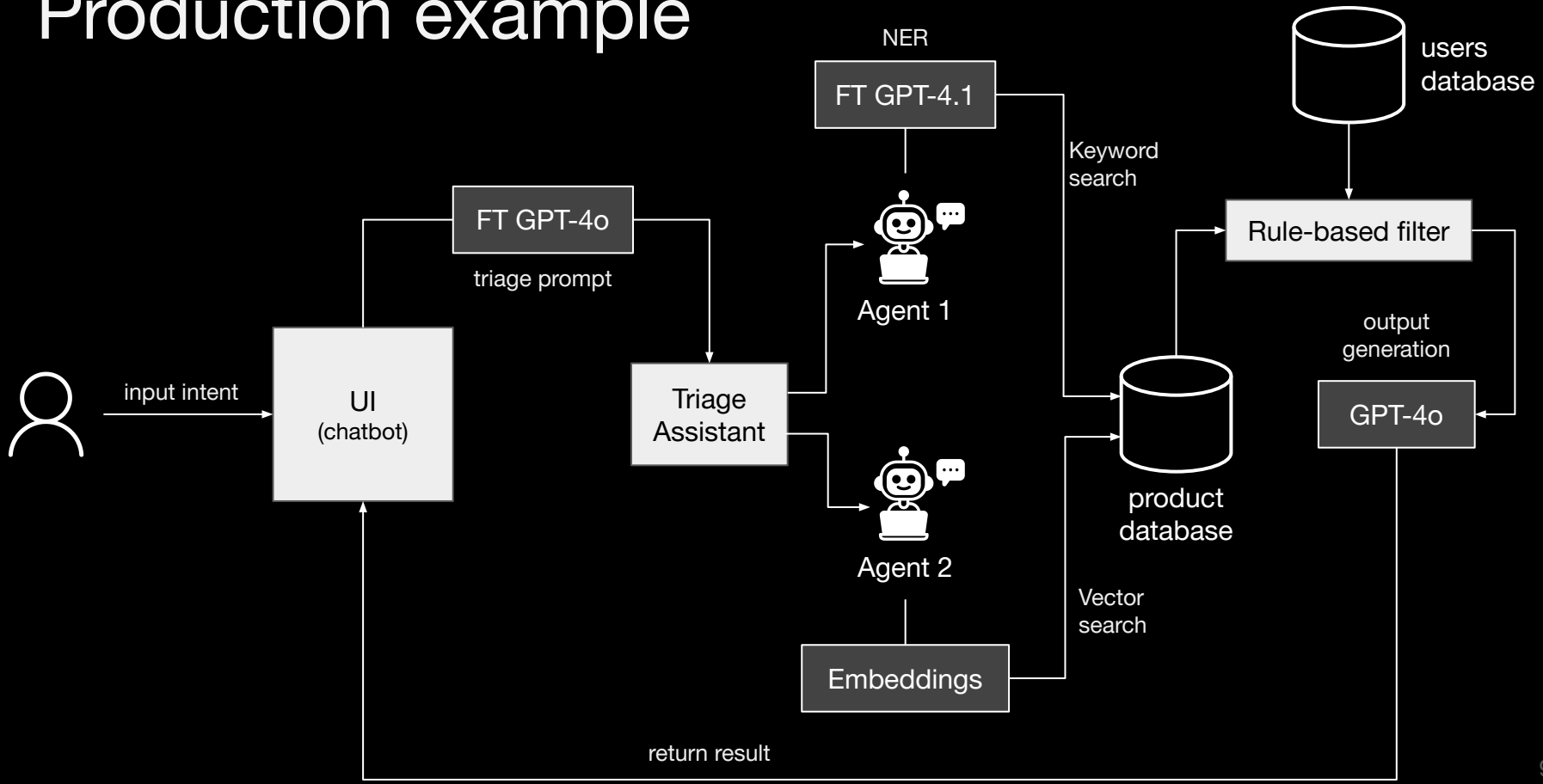
Process flow



High-level architecture



Production example



Features and limitations

Good for ✅

- **Providing relevant content suggestions based on user input**
- **Combining natural language, visual input and/or metadata**
- **Deterministic as well as semantic searches**

Not good for ❌

- **Providing general truth responses:**
  - If you need universally true info, this may not be the right path.
- **Generating truly novel responses:**
  - Responses will reflect user history, often yielding similar content.

Key challenges and pitfalls

CHALLENGES

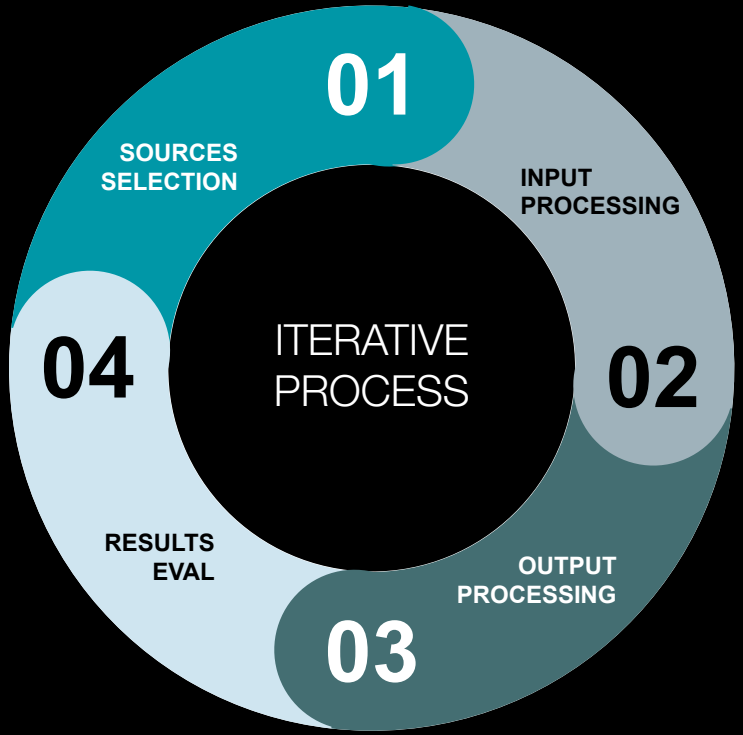
- Figuring out how to process and blend multiple input data sources
- Evaluating performance
- Depending on use case, figuring out how to overcome the “cold start” problem

PITFALLS & MITIGATIONS

- **Relying too much on LLMs**
  - Validate if a sub-system could use deterministic code.
- **“Content Bubble” Risk**
  - Too much reliance on memory may generate the same results. Keep your outputs fresh.

Solution Deep Dive

Process stages



01 SOURCES SELECTION

Relevant data points, signals or user input should be selected as reliable sources to provide recommendations

02 INPUT PROCESSING

User or app input can be transformed or entities can be extracted from it

03 OUTPUT PROCESSING

Knowledge base can be searched via multiple approaches and matching results can be used to generate a final answer

04 RESULTS EVAL

Performance should be evaluated against golden answers or ground truth

Accelerator Steps

01

Foundation

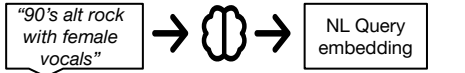
Enrich product data for better search and recommendations

ITEM TAGS **LLM** **ITEM DESCRIPTIONS** **LLM EMB. MODEL**

02

Integration

Combine search and RAG to find the right product



NL QUERY **LANGUAGE EMBEDDING MODEL** **NL QUERY EMBEDDING**

Evaluation

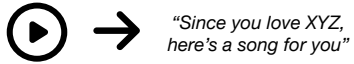
Continuously evaluate matches and incorporate user feedback

MATCHES **RECS** **USER FEEDBACK**

03

Presentation

Use the LLM to generate a personalized, multi-modal output



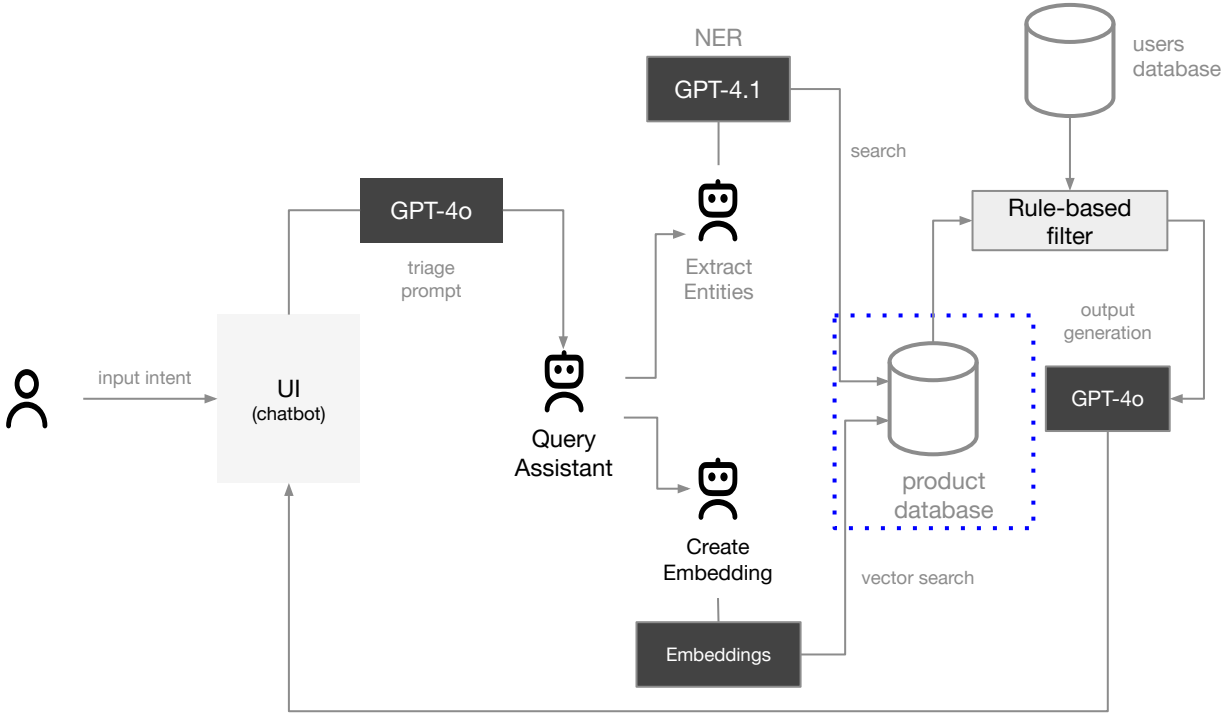
MUSIC ITEMS

Foundations

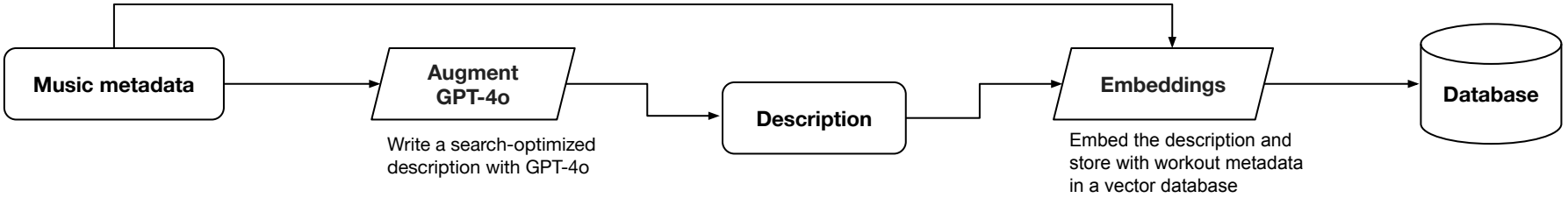
01

Foundation

Enrich product data for better search and recommendations



Example: LLM-powered Music Search



Lyrics: "*Annie, jump a little higher…*"

Artist: *Alex Rivers*

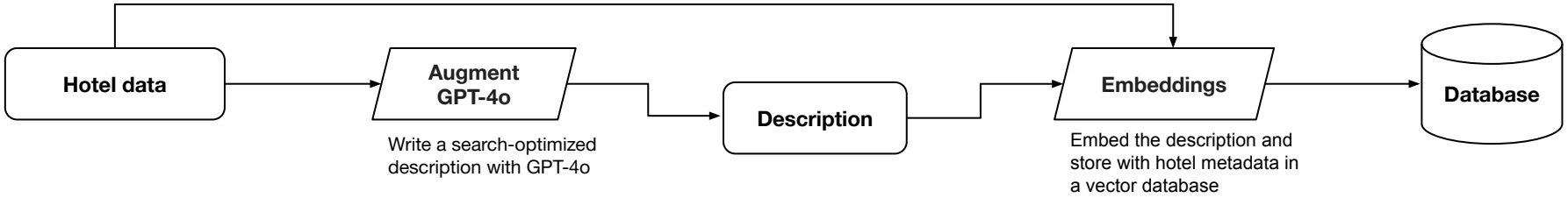
Tags: *Rock, Energetic, Upbeat, Classic Rock, 1970s, Anthem, Guitar Solos, Saxophone, Storytelling, Joyful, Rebellion, Coming-of-age, New Jersey Sound, Live Performance Favorite*

Annie (Come Out Tonight) by Alex Rivers is a quintessential high-energy classic rock anthem from the 1970s. Known for its infectious upbeat tempo, the song features dynamic guitar solos and powerful saxophone sections that highlight Rivers's vibrant coastal sound. The track tells a compelling coming-of-age story filled with themes of youthful rebellion and exuberance, showcasing Rivers's unparalleled storytelling prowess.

This song captures the joy and exuberance of youth, making it a timeless anthem of rock history. It is a fan favorite in live performances, renowned for its electrifying and joyous atmosphere. Annie (Come Out Tonight) stands out for its spirited and anthemic qualities, resonating with generations of listeners and cementing its place as an enduring and iconic track in Alex Rivers's illustrious career.

With its blend of energetic rhythms and heartfelt lyrics, Annie (Come Out Tonight) remains a defining piece of the 1970s rock era, embodying the rebellious spirit and vibrant sound that characterized Rivers's music during this period.

Example: LLM-powered Travel Search



BrewDog DogHouse Edinburgh is a boutique hotel just off the Royal Mile, with an on-site bar and rooftop terrace. Amenities include complimentary Wi-Fi, free bike hire, room service, and a 24-hour front desk.

Discover the eclectic charm of BrewDog DogHouse Edinburgh Hotel, nestled just a stone's throw from the historic Royal Mile. This boutique haven delights with its mix of modern amenities and Scottish flair, perfect for couples and explorers alike. Revel in the unique combination of cozy rooms, a vibrant restaurant, and terrace views, all while enjoying the convenience of free bikes and WiFi. Immerse yourself in the heart of Edinburgh's history with a smattering of craft beer culture thrown in – a perfect score for location by couples – and make your stay a part of your adventure.

Best Practices - Foundations

Centralize and Consolidate Your Data

Bring together all relevant data points; product info, user profiles, interaction logs, and text descriptions into a single, well-governed repository. A unified data source simplifies enrichment and retrieval.

Use LLMs to Enrich Your Content

Leverage a model like GPT-4.1 to auto-generate summaries, tags, and rich metadata for each item. This boosts semantic relevance, making downstream search and recommendations more accurate.

Design for Ongoing Updates

Your item catalog is dynamic; new products, shifting user preferences, and stale metadata are inevitable. Establish a regular refresh pipeline for embeddings, tags, and descriptions to maintain recommendation quality over time.

Validate Before You Ship

Run QA on enriched metadata before serving it in production. Ensure auto generated tags and summaries are accurate and add value, especially for high-impact or long-tail items

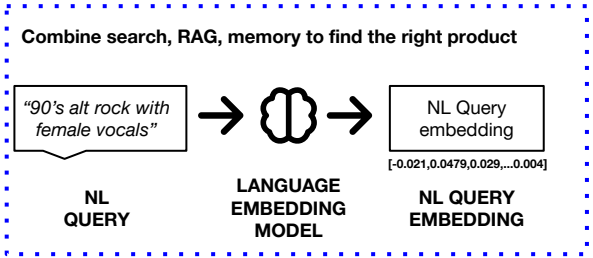
Capture Key Structured Signals

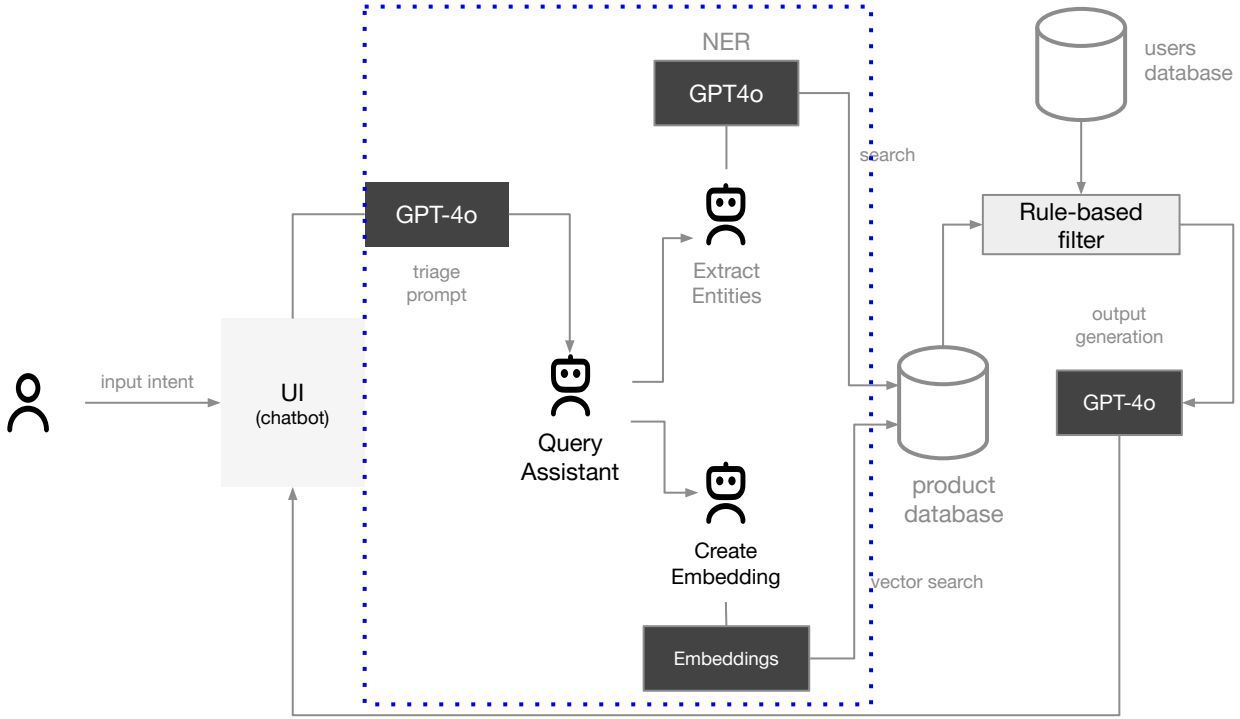
Retain structured attributes like category, price range, and rating. These fields are critical for hybrid search setups that blend semantic retrieval with filtering, ranking, or sorting.

Integration - Retrieval

02

Integration: RAG

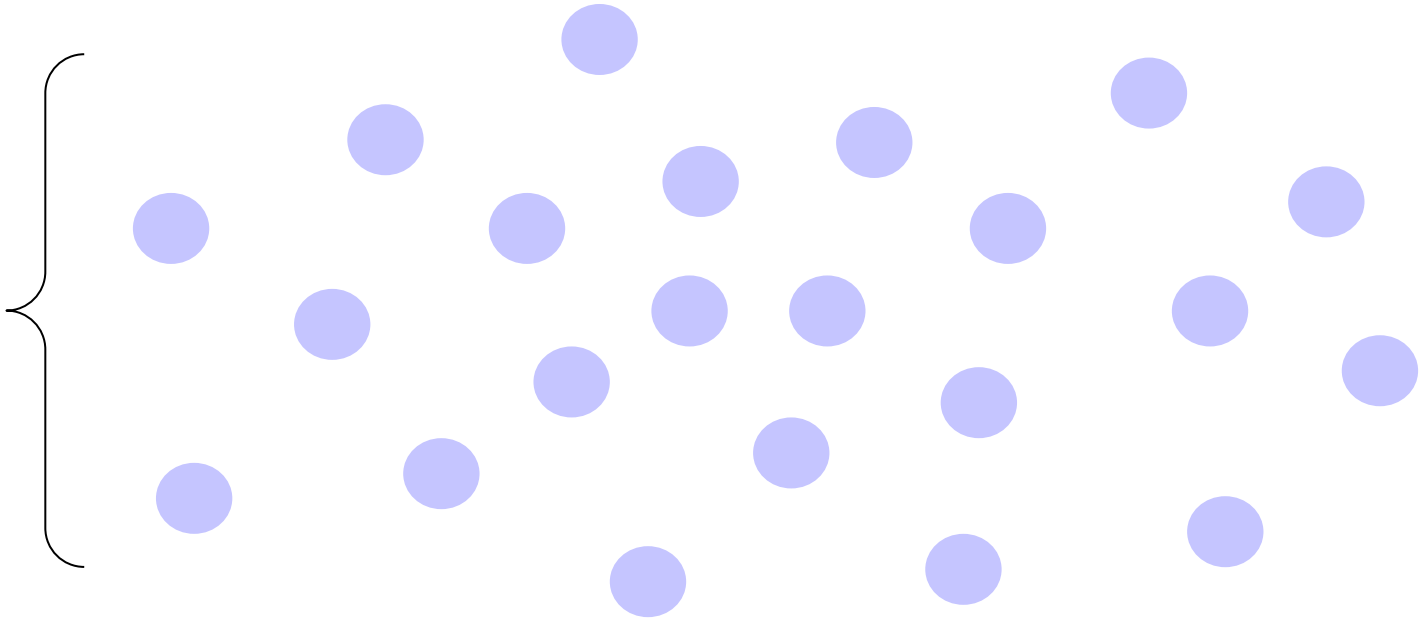




Traditional search

"Hey, I'm going to Edinburgh for five days and want somewhere nice."

Hotels



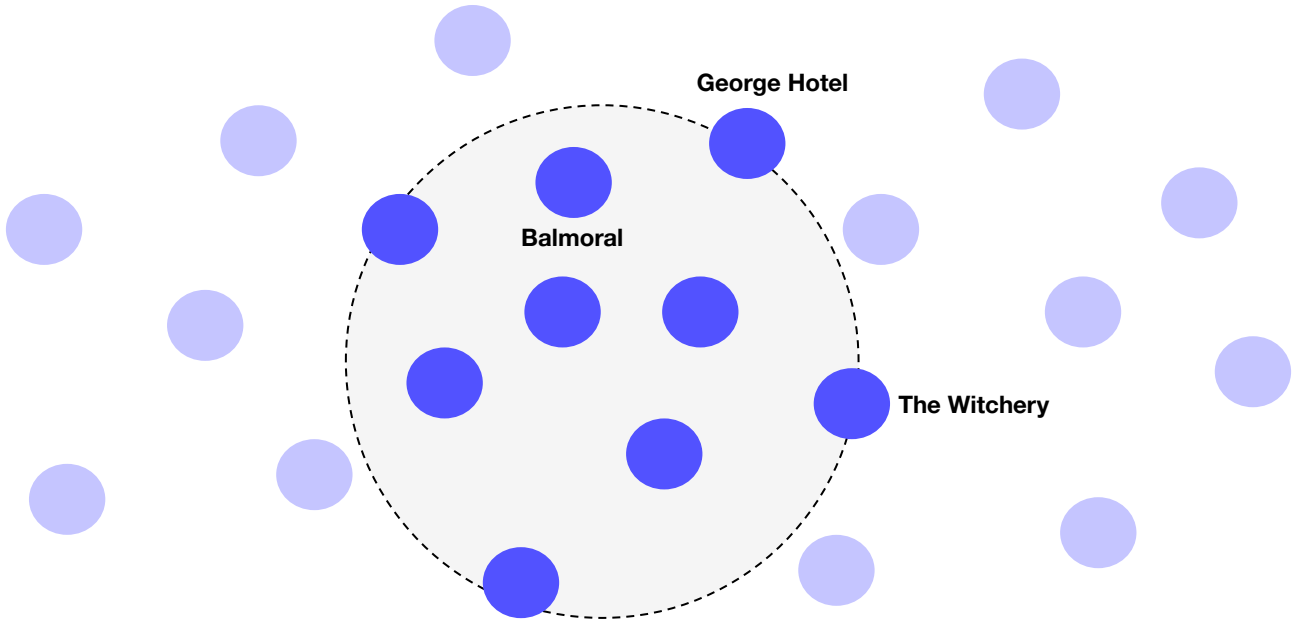
Available hotel

Recommended hotel

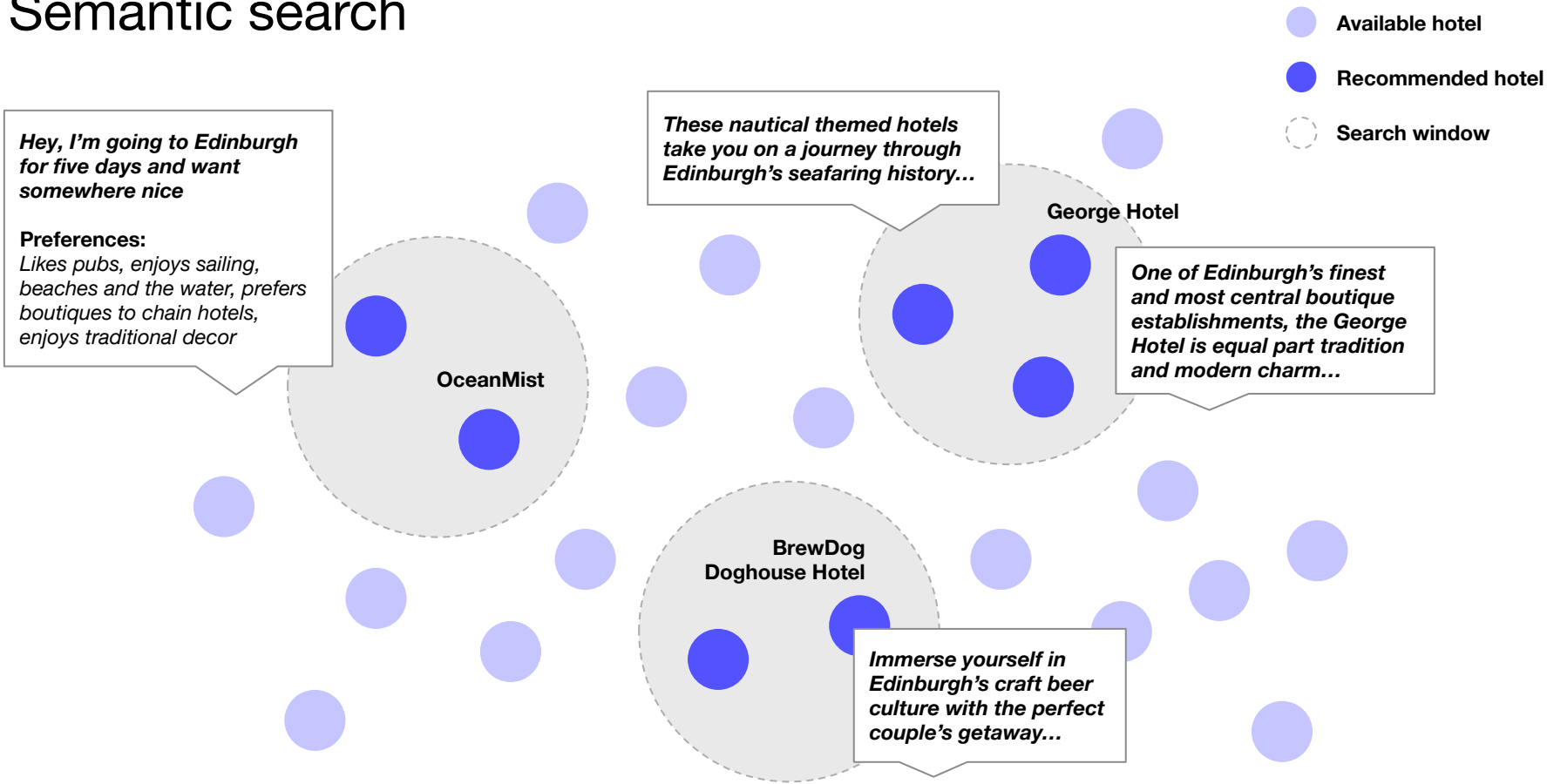
Search window

Traditional search

"Hey, I'm going to Edinburgh for five days and want somewhere nice."



Semantic search



Best Practices - Retrieval

Use Hybrid Retrieval

Relying on only one search method (keyword or semantic) can miss key matches.

- Use keyword search for exact matches and fast lookups
- Use semantic search for deeper, contextual relevance

Combining both gives you the best of precision and recall.

Keep Only the Most Relevant Matches

Merge the top-N keyword results with the top-N semantic results, then apply a re-ranking step to sort by relevance.

This creates a more accurate, final retrieval set that balances exactness and meaning.

Apply Smart Pre- and Post-Filters

Use keyword filters (e.g., category: shoes) to narrow large datasets before semantic ranking.

Post-filter if needed to enforce business constraints.

Test RAG Components in Isolation

Evaluate NER, keyword search, and semantic search independently. This allows you to iterate, benchmark performance, and troubleshoot issues without cross-contamination.

Create a Feedback Loop

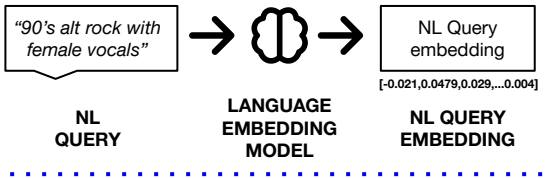
Track how users interact—clicks, likes, purchases—to refine keyword weighting and update embeddings. User signals help the system learn and continuously improve.

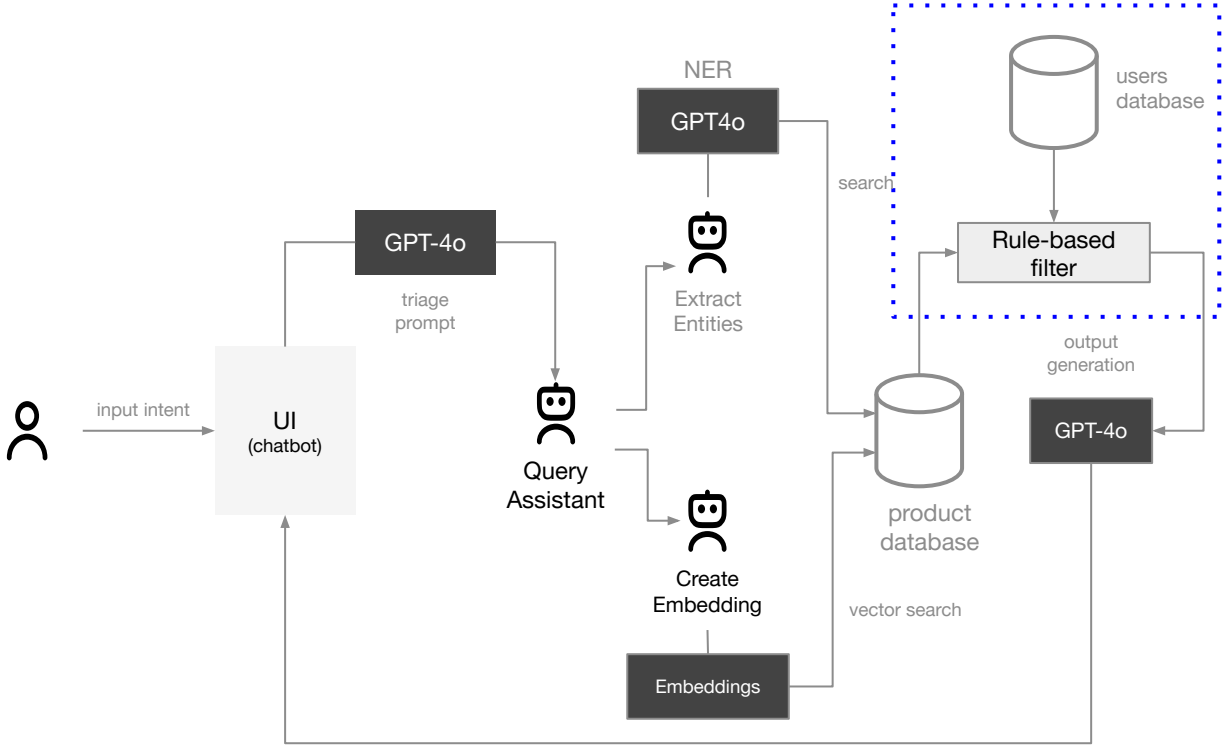
Integration - Memory

02

Integration: Memory

Combine search, RAG, memory to find the right product





Memory management

Short term memory

Manage conversation state and history for contextual answers

Long term memory

Store user data to provide personalised experiences

Personalized Recommendations

- ⚡ Keep context from ongoing conversations enables more relevant, in-the-moment suggestions.
- ⚡ Storing user preferences, behaviors, and past interactions helps tailor responses over time.
- ⚡ Combining both leads to dynamic personalized recommendations—responding accurately in the moment while adapting to evolving user needs

Managing Memory

Short term memory

```
system_prompt =  """You are a helpful
assistant.
Summarize the following conversation
between a user and an assistant,
highlighting the key points and any
important information or preferences
expressed by the user."""
```

Summarization

Periodically condense conversations using LLM-generated summaries.

```
for message in conversation_history:
  msg_embed = get_embed(msg['content'])
  similarity = cos_sim(input, msg_embed)
    message['similarity'] = similarity
```

Selective Inclusion

Keep only the most relevant past interactions, identified by semantic similarity.

```
for e in doc.ents:
  if e.label_=="PERSON": state["name"] =
state["name"] or e.text
  if e.label_=="AGE": state["age"] =
state["age"] or e.text
  if e.label_=="MONEY": state["budget"] =
state["budget"] or e.text
```

Conversation State Tracking

Extract structured information (intent, preferences) through NLP to maintain interaction context.

Managing Memory

Long term memory

Stores conversation summaries and user preferences for future reference—enabling personalized, engaging interactions (like a barista remembering your usual order).

User Profiles

Store structured data (preferences, history) for consistent, personalized experiences.

Personalization

Continuously update profiles based on user input and feedback.

Best Practices - Memory Implementation

Memory Structure Design

Use short-term memory for recent, relevant messages; long-term memory stores structured user data and embeddings for fast retrieval.

Adaptive Memory via Feedback

Let users fix outdated info live; learn from past interaction success/failure patterns to improve future agent behavior.

Performance Optimization

Use vector embeddings for fast semantic search; cache high-frequency items like preferences in memory or prompts.

Managing State and Agent Communication

Keep essential user data in a shared space with concise action logs to reduce noise and streamline agent coordination.

Robust Governance and Security Controls

Allow users to access/edit their data and only retrieve what's needed to enhance privacy and data protection.

Adaptive Memory via Feedback and Reflection

Let users fix outdated info live; learn from past interaction success/failure patterns to improve future agent behavior.

Step 3: Presentation

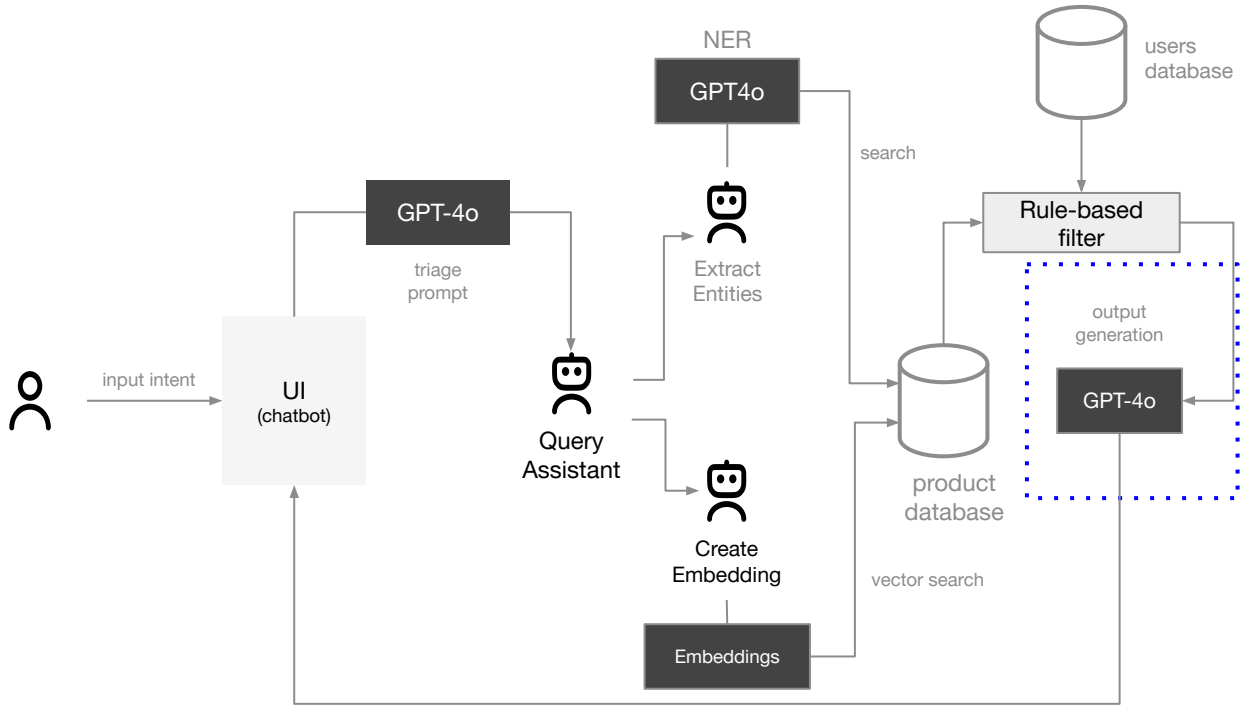
03

Presentation

Use the LLM to generate a personalized, multi-modal output

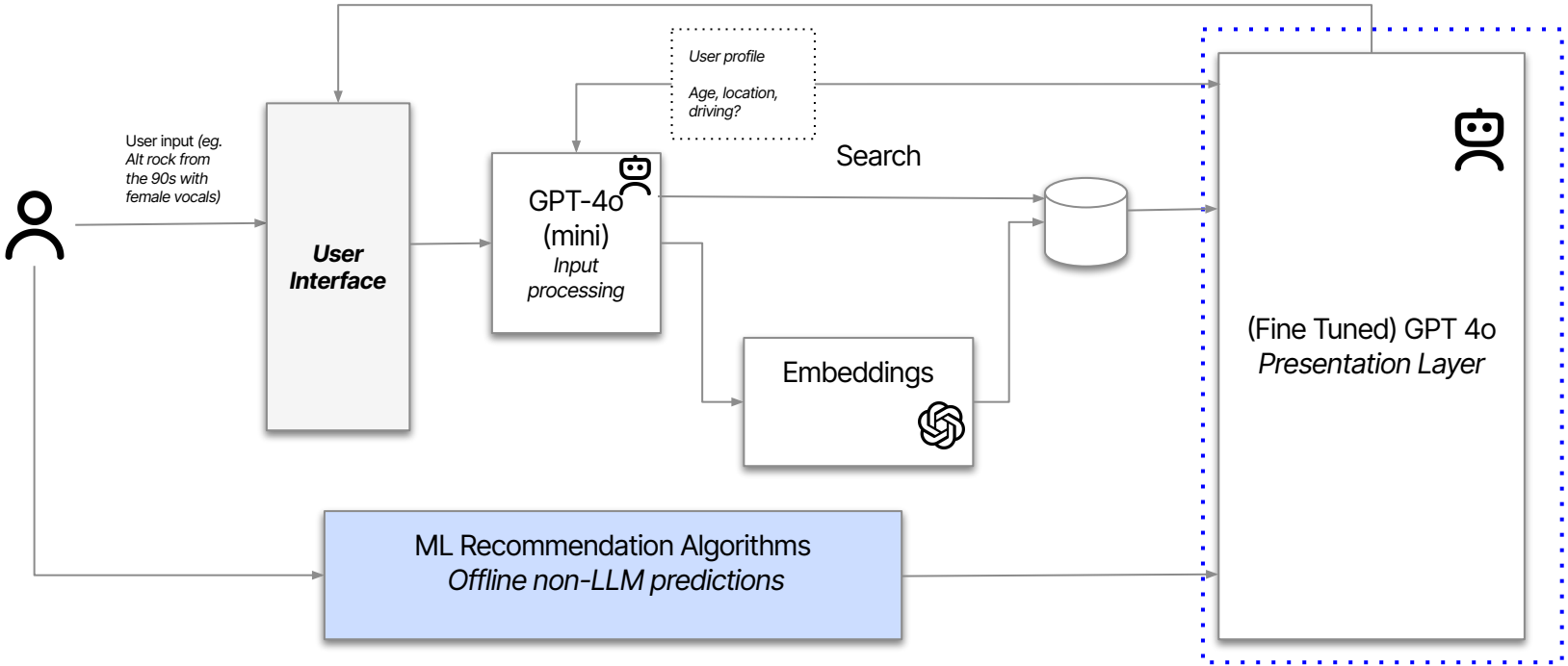
"Since you love XYZ, here's a song for you"

MUSIC ITEMS



Sample Architecture

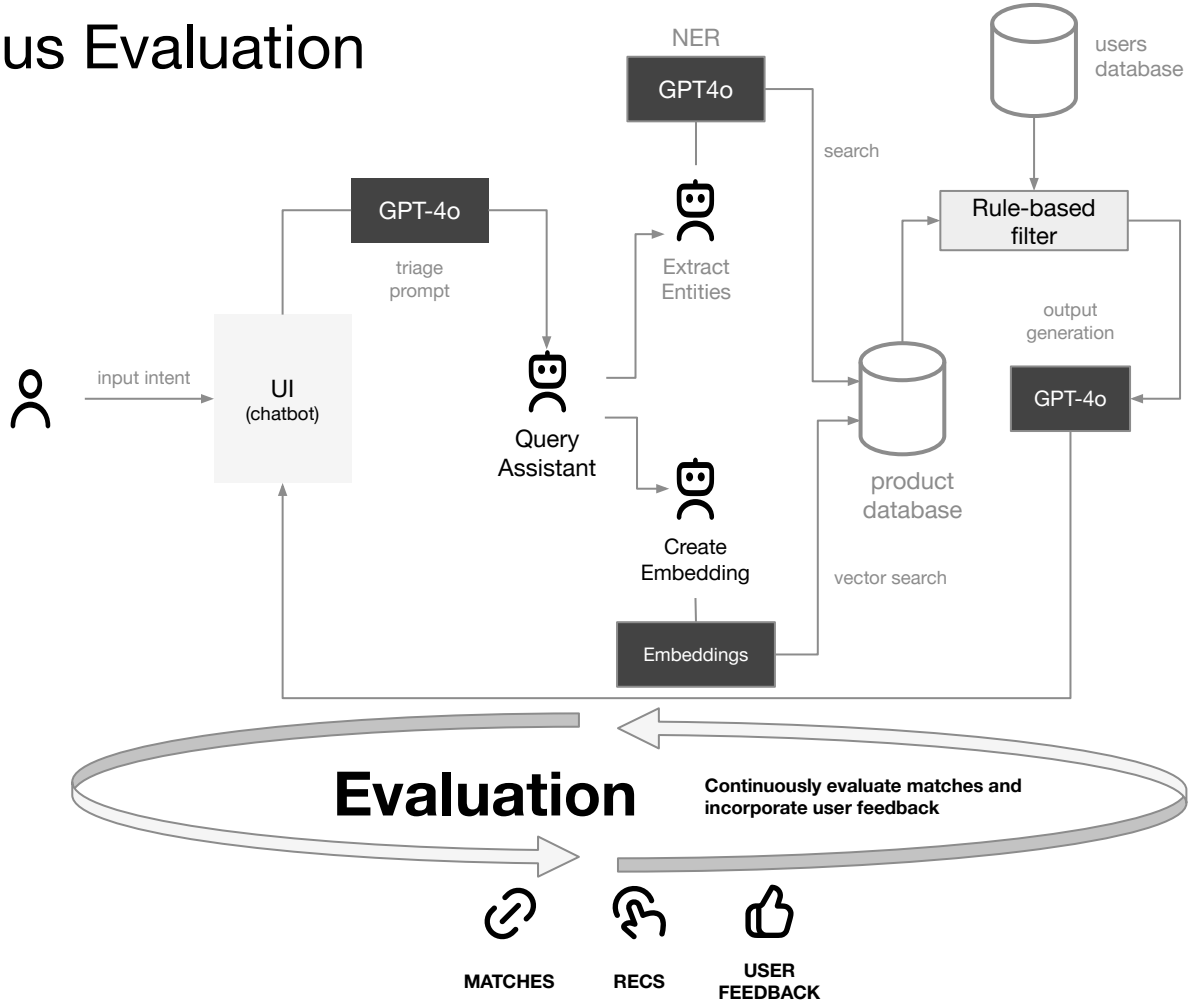
Hey there, road warrior! "Seether" by Veruca Salt is a perfect track for your next drive. It's got that raw, edgy sound you crave, with lyrics that dive into inner conflicts and empowerment. Imagine cruising along the coastline or through winding mountain roads, with Nina Gordon and Louise Post's harmonies fueling your sense of adventure. This track will make you feel unstoppable, just like the open road ahead of you. Enjoy the ride!



Best Practices - Presentation

User Experience	**Relevance First** Use visual hierarchy (larger font, imagery, cards) to make the top suggestion stand out without overwhelming the user.	**Enable Quick Actions** Include buttons like “Save,” “Try,” “Skip,” or “Tell me why” to make recommendations actionable and interactive.	**Support Lightweight Feedback** Let users give feedback (e.g., thumbs up/down, “Not relevant”) to close the loop and improve future outputs.
LLM Usage	**Explain the “Why” Behind Each Recommendation** Use natural language to surface rationale: “Because you liked [X]…” or “Popular near you right now.”	**Adapt Tone to Context** Vary language style depending on the use case — playful for entertainment, neutral for utilities, formal for enterprise. Can leverage fine tuning for more tonality control.	**Fuse Multiple Signals Seamlessly** Combine user history, embeddings, metadata, and model scores into a cohesive narrative that feels human and intentional.

Continuous Evaluation



Evaluation

Find possible matches

- % surfaced matches vs expected
- % irrelevant matches (subjective evaluation)

Provide recommendations

- % recommended accepted vs rejected
- Similarity score of historic behaviour

User feedback

- User engagement and retention (e.g.: in-app clicks)
- Thumbs up/down feedback and CSAT scores

Lean on deterministic evals for simple cases, then work your way up

01 Deterministic

Multiple choice answers
Formatting checks
String contains
BLEU or ROUGE scores

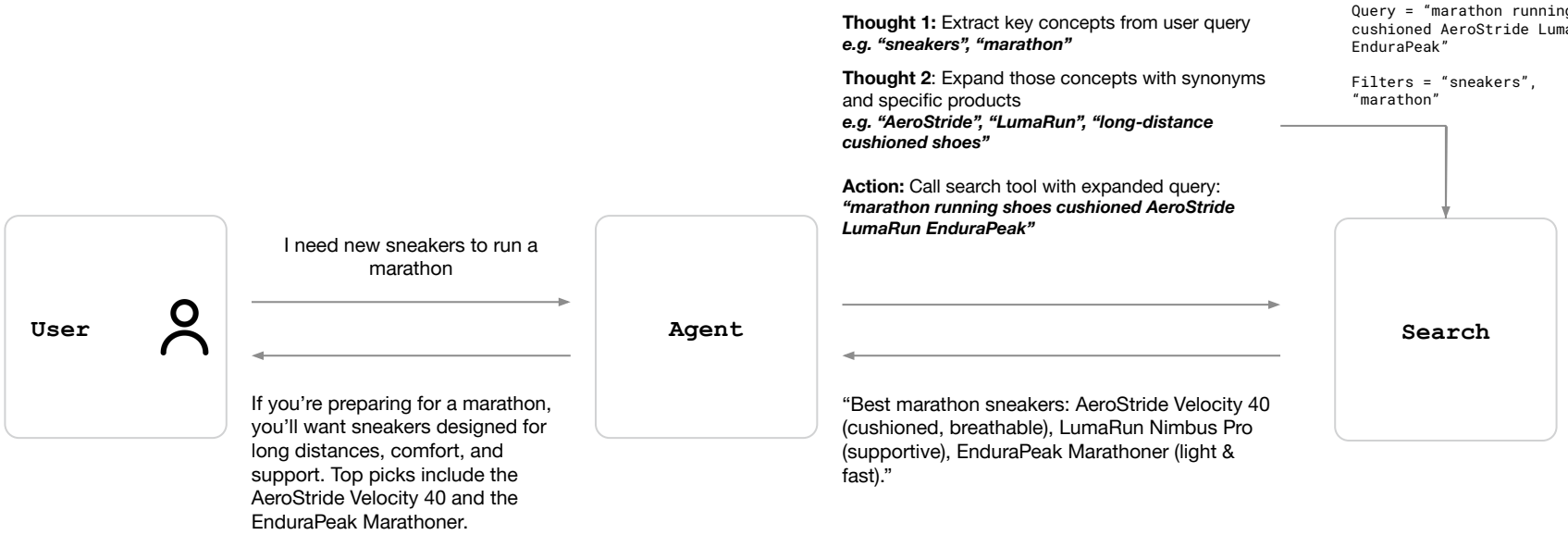
02 Model-Graded Evals

Faithfulness
Relevance
Factuality
Tone

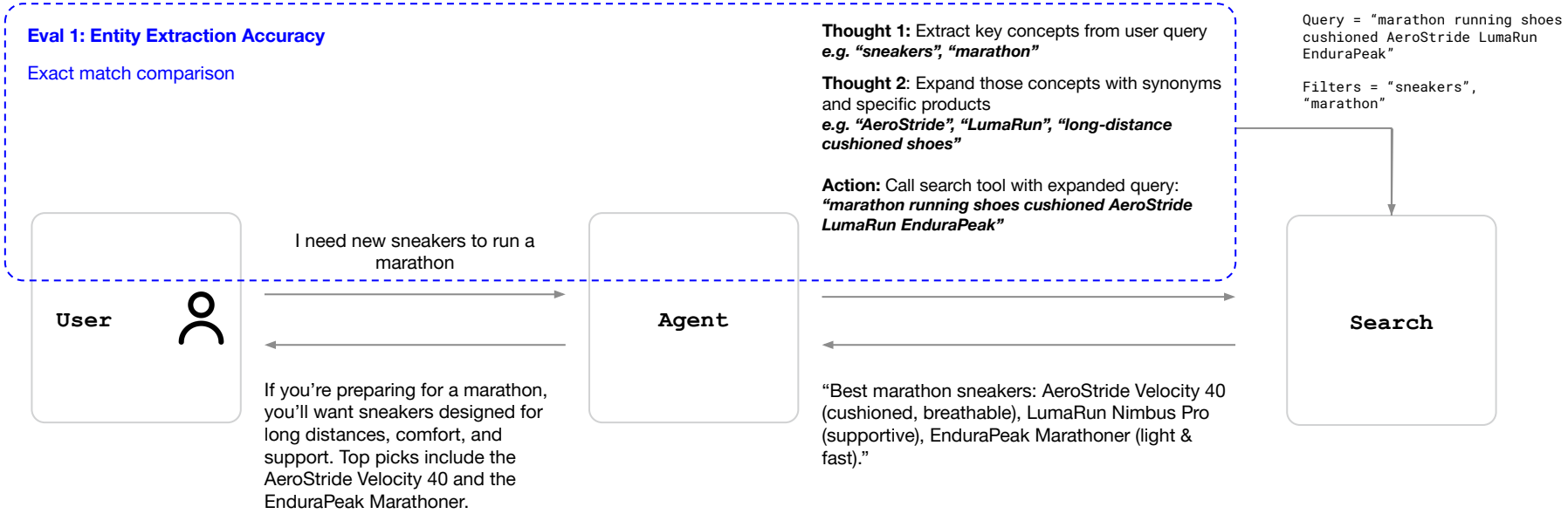
03 Human-Graded Evals

Side-by-side preference
Nuanced quality scores

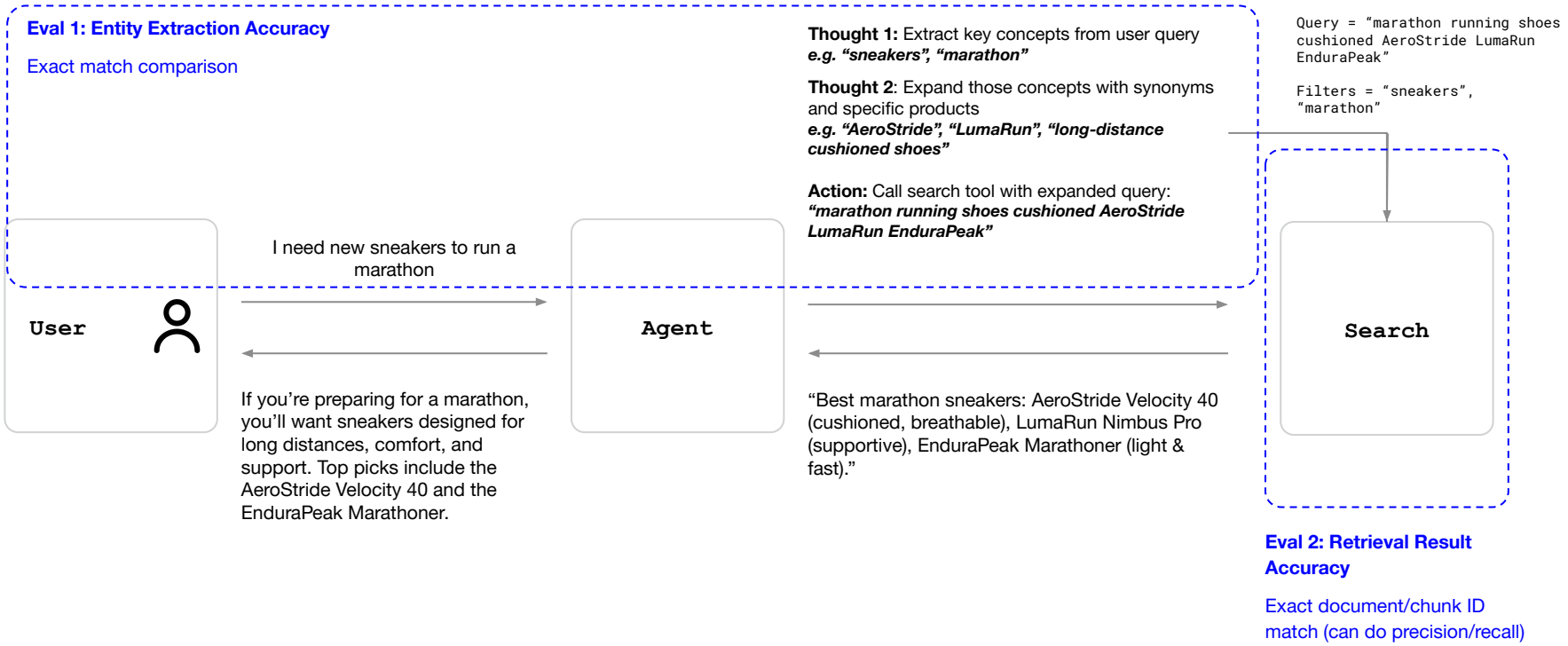
Q&A with RAG can use many deterministic evals



Q&A with RAG can use many deterministic evals

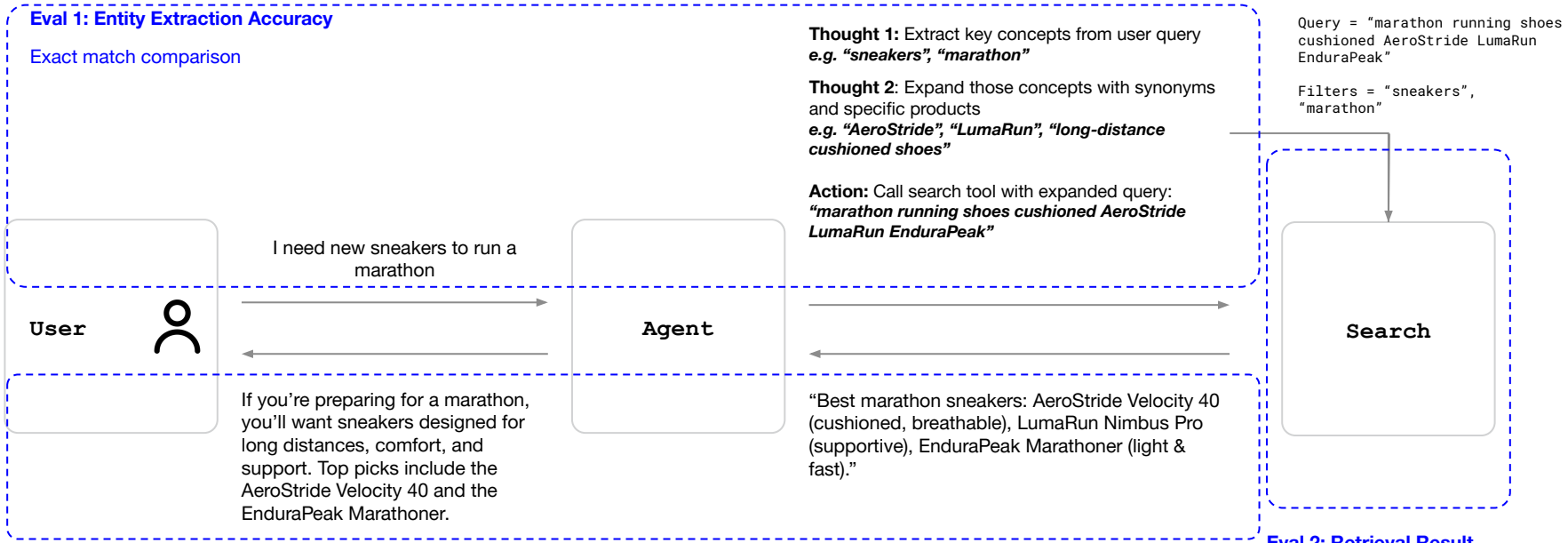


Q&A with RAG can use many deterministic evals



Q&A with RAG can use many deterministic evals

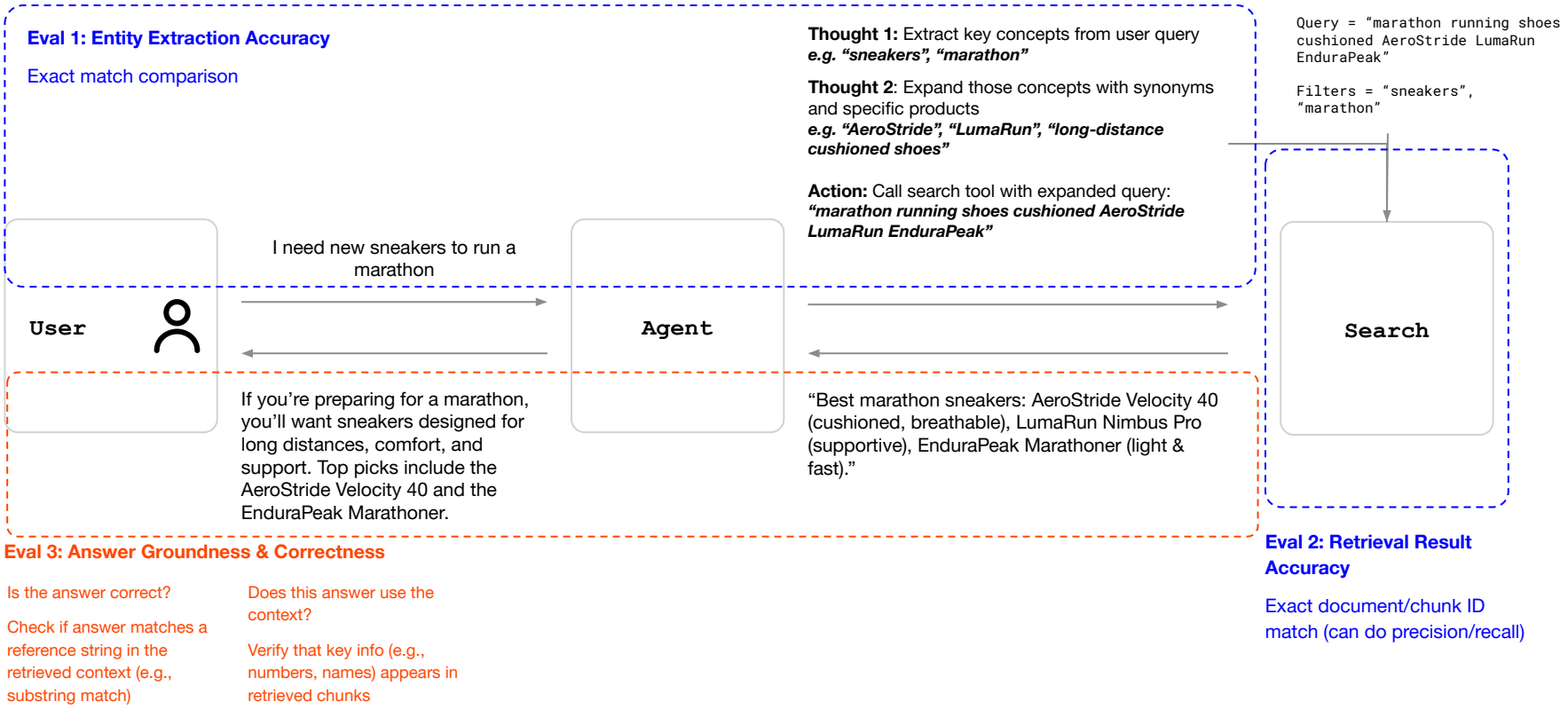
--- Deterministic evals



Q&A with RAG can use many deterministic evals

--- Deterministic evals

--- Model-graded evals



Success & Customer Stories

Success metrics

Process Step	Solution	Evaluation	Metric
Find possible matches	Automated	% of matches surfaced vs expected % of irrelevant matches Subjective evaluation of relevance	% of relevant matches seen by users vs baseline
Provide recommendations	Co-pilot (used by agent)	% suggestions used by agent	% improvement in user engagement vs baseline + hours saved by agents vs baseline
	Automated	% recommendations accepted vs rejected by test user	% improvement in user engagement vs baseline

Key Takeaways

- **Personalization Drives Value:** Recommendation systems enhance engagement and satisfaction by tailoring experiences to individual user preferences, leading to longer sessions, deeper interaction, and revenue growth.

- **Data Depth & Effective LLM Use:** Richer input data leads to better results. Pair LLMs with deterministic methods to boost precision and avoid repetitive “content bubbles.” Rather than relying on LLMs for memory, use them to clearly convey insights and guide users effectively.

- **Data Science Best Practices:** Ensure rigorous evaluation, incorporate relevant data sources, and consistently test and validate to build reliable, high-performing systems that overcome the “cold start” problem.