

Agentic RAG - Multisource

[Introduction](#)

[Structure \(v1\)](#)

[Implementation using Step Function & Lambdas \(v1\)](#)

[Planner](#)

[Retriever Knowledge Base](#)

[Retriever Vitic](#)

[Analyzer KB](#)




[Analyzer Vitic](#)

[Reflector](#)

[Finalizer](#)

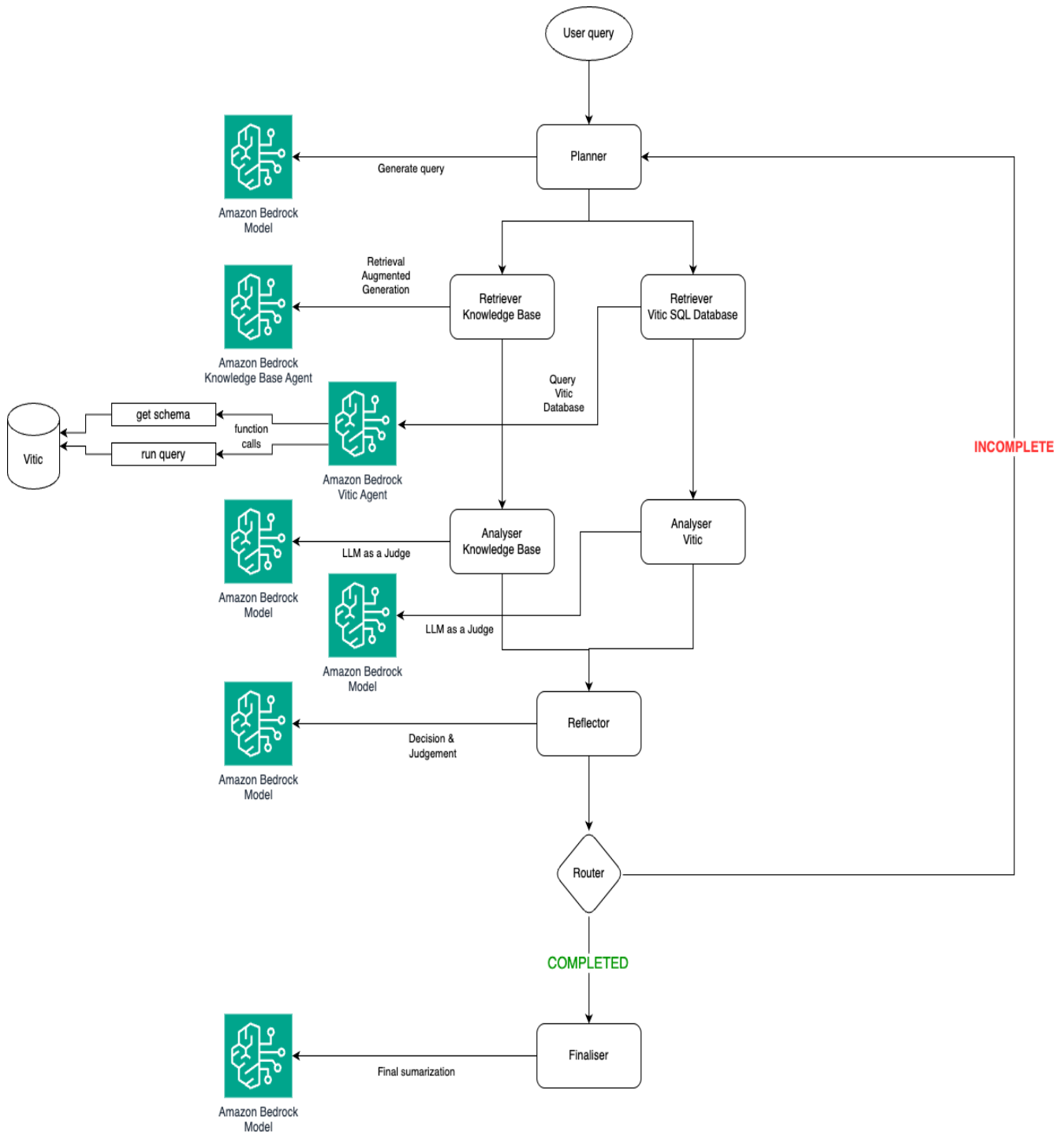
[Structure \(v2\)](#)

Introduction

Agentic RAG - Multisource extends the functionality of [Agentic RAG](#) to include integration of  [Vitic Agent V2](#) retriever. Custom  [Vitic Agent V2](#) retriever. Custom analysers for Knowledge Base and Vitic are included. The Reflection step is processing both analysis, from KB and from Vitic and output Verdict and Justification based on both criteria. The last step Agent has now also a summarisation role, like in the  [Combined RAG Fusion V2](#) .

Structure (v1)

The architecture diagram for the Agentic RAG - Multisource, that integrates Knowledge Base and Vitic data retrieval, is shown in the next figure.

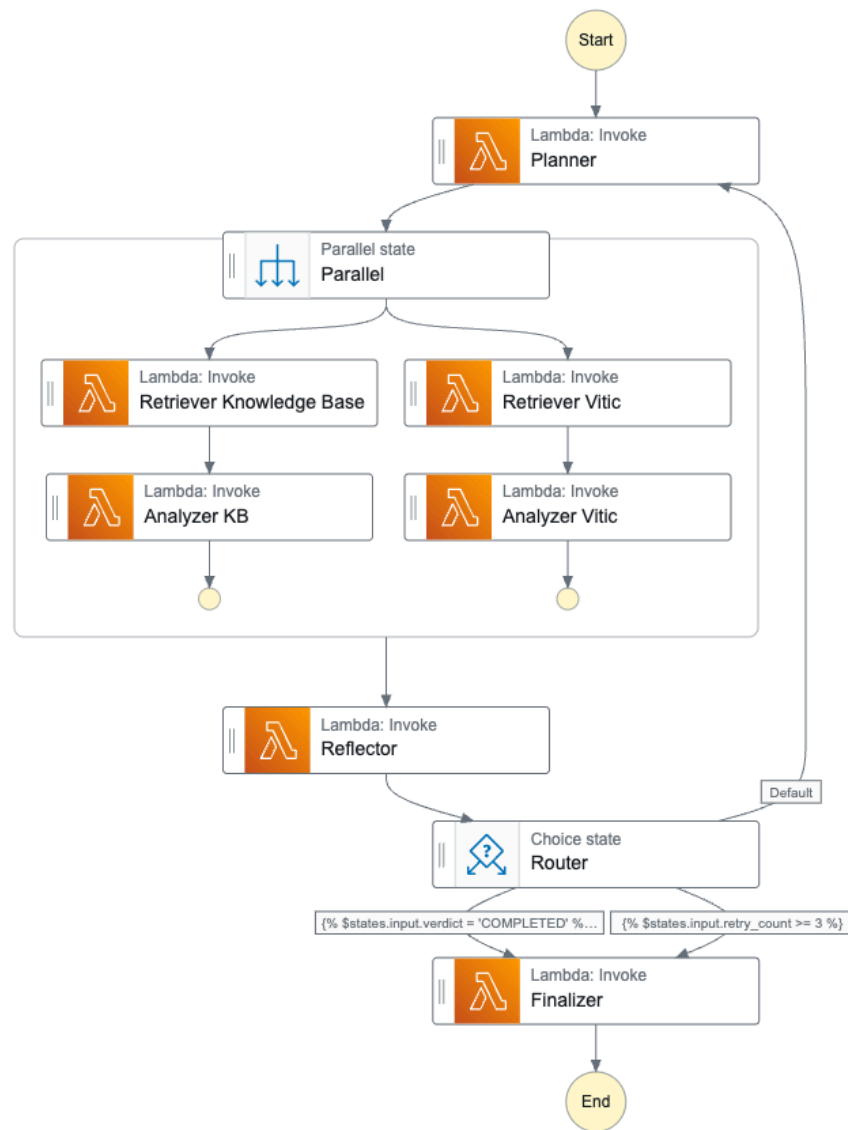


The Planner, Retriever (Knowledge Base/Vitic), Analyzer (KB/Vitic), Reflector, Finalizer steps are implemented as following:

- **Planner** - invoke a **Bedrock Model** to generate current query; add the query to query history list; if route count > 0, will use for generating current query also the retrieval history
- **Retriever Knowledge Base** - invoke a **Bedrock Agent (Knowledge Base)** to retrieve the current content (summary + citations) based on current query; add the retrieved data to the retrieval history
- **Retriever Vitic** - invoke a **Bedrock Agent (Vitic)** to retrieve the result of querying Vitic database with a SQL query generated based on current question and on the database schema. To retrieve the database schema, **Vitic Bedrock Agent** is using a function call, `get_schema`. After the SQL query is generated, the **Vitic Bedrock Agent** executes the SQL query using a second function call, `run_query`. Both function calls are available through one Action Group defined for the **Vitic Bedrock Agent**.
- **Analyzer KB** - use the current query, and the retrieval information from **Retriever Knowledge Base** and invoke a **Bedrock Model** to evaluate the quality of retrieval relative to the current query. Add a synthesis (`synthesis_kb`) (`synthesis_kb`).

- **Analyzer Vitic** - use the current query, and the retrieval information from **Retriever Vitic** and invoke a **Bedrock Model** to evaluate the quality of retrieval relative to the current query. Add a synthesis (`synthesis_sql`).
- **Reflector** - uses the two two synthesis (for KB & Vitic) and both both the retrieval histories (for KB & Vitic) and present them through invoking a **Bedrock Model** to provide a verdict and a justification.
- **Router** - decide based on routing number and decision if the next step is **Finalizer** or reroute to Planner. Condition is:
 - If verdict is COMPLETED or route_count >=3 will route to **Finalizer**
 - If verdict is INCOMPLETE and route_count < 3 will route to **Planner**
- **Finalizer** - Use a **Bedrock Model** to formulate the summary from the Knowledge Base summary & Vitic summary and reformat the input to prepare the output in a standardised form.

Implementation using Step Function & Lambdas (v1) [🔗](#)



Agentic RAG - Multisource implementation using Step Function & Lambdas

The **Planner**, **Retriever (Knowledge Base/Vitic)**, **Analyzer (KB/Vitic)**, **Reflector**, **Finalizer** steps are implemented as lambda functions

The **Router** is implemented as a choice state with the following condition combination:

```
1 {% $states.input.verdict = 'COMPLETED' %}
```

OR

```
1 {% $states.input.retry_count >= 3 %}
```

In the following we will detail the implementation of the agents included.

Planner [🔗](#)

The instructions for the planner agent are shown in the next code snippet:

```
1 prompt = f"""
2     You are a scientific research planning agent with expertise in chemistry, toxicology, pharmacology,
3     and carcinogenicity.
4     You operate as the planning component in an Agentic RAG system designed to break complex biomedical
5     questions into evidence-focused subqueries.
6     You are tasked with generating the next best subquery based on the research goal and prior
7     retrieval history.
8     Your target is a high-quality scientific knowledge base containing peer-reviewed studies, reviews, and
9     regulatory guidance documents.
10    Main question: {original_query}
11    Previous attempts (Knowledge Base):
12    {history_block_kb or 'None yet'}
13    Previous attempts (SQL):
14    {history_block_sql or 'None yet'}
15
16    Your subquery should aim to:
17    - Retrieve specific, verifiable evidence (e.g., dose-response data, potency thresholds, specific
18    assay outcomes, strain/species results, molecular targets).
19    - Support responses that are structurable with distinct sections (e.g., In Vitro, In Vivo,
20    Regulatory Assessment).
21    - Align closely with the intent of the main question – whether it asks for numerical counts,
22    mechanisms, assay outcomes, or regulatory interpretations.
23    - Surface regulatory-relevant insights if applicable (e.g., ICH S2(R1), FDA labeling, thresholds
24    of toxicological concern).
25    - Fill known gaps from previous attempts – avoid repeating or rephrasing earlier subqueries.
26
27    Avoid:
28    - Broad or vague rewordings of the main question.
29    - Subqueries that invite speculative synthesis without grounding in concrete evidence.
30
31    Only return the next best subquery as a concise, single line. No commentary or formatting.
32    """
```

The Planner is instructed to generate next subquery based on the initial query and the history of previous attempts (questions and answers) from both Knowledge Base and for Vitic (SQL) parallel branches. Additional instructions include domain-specific instructions, to help adapt the next subqueries to the actual toxicology domain. Frequent errors that should be avoided are also specified.

The model used is: `MODEL_ID = "anthropic.claude-3-sonnet-20240229-v1:0"`

Retriever Knowledge Base [🔗](#)

This agent implementation is similar (but separate, to allow customisation of input and output according to the needs of this Agent collaboration workflow) to the [Knowledge Base Agent](#) described in the dedicated section of [Knowledge Base RAG Fusion Agent](#).

Retriever Vitic [🔗](#)

The agent implementation has two possible options:

- Using a Lambda function call to the Lambda implementing [Vitic Agent V2](#).
- Using an Bedrock Agent invoking for the [Vitic Agent with Action Group](#).

Default functionality will use the up-to-date, function call to [Vitic Agent V2](#).

The following code snippet shows this implementation:

```
1  if use_agent_retrieval:
2      # initialize Bedrock agent
3      bedrock_agent = boto3.client("bedrock-agent-runtime")
4      # use Vitic agent coordinator (with Action groups for tools)
5      response = bedrock_agent.invoke_agent(
6          agentId=agent_id,
7          agentAliasId=agent_alias_id,
8          sessionId=session_id, # keep unchanged to preserve memory
9          inputText=current_subquery,
10     )
11     response_time = round(time.time() - start_time - process_time, 4)
12
13     logger.info(pprint.pprint(response))
14
15     # Parse the response body - should be treated as an event stream
16     event_stream = response['completion']
17     final_answer = None
18     citations = [] # Initialize citations to avoid unbound variable error
19     try:
20         for event in event_stream:
21             if "chunk" in event:
22                 data = event["chunk"]["bytes"]
23                 final_answer = data.decode("utf8")
24             elif "trace" in event:
25                 logger.info(json.dumps(event["trace"], indent=2))
26             else:
27                 raise Exception("unexpected event.", event)
28     except Exception as e:
29         raise Exception("unexpected event.", e)
30
31     metadata = []
32 else:
33     # invoke latest implementation of Vitic retrieval
34     lambda_client = boto3.client('lambda')
35     response = lambda_client.invoke(
36         FunctionName='test_function_26May2025',
37         InvocationType='RequestResponse',
38         Payload=json.dumps({'question': current_subquery}).encode('utf-8')
39     )
40     response_time = round(time.time() - start_time - process_time, 4)
41
42     response_payload = json.loads(response['Payload'].read())
```

```

43     result = response_payload['response']
44     metadata = response_payload['metadata']
45     final_answer = result

```

Analyzer KB [🔗](#)

The code for model invoke is show in the following code snippet:

```

1     combined = f"\n\nInitial Question: {question}"
2     combined += "\n\n".join(f"Query: {r['query']}\nResult: {r['result']}" for r in retrieval_history_kb)
3
4     prompt = f"""
5     You are an expert scientific analysis agent supporting a biomedical research workflow focused on
6     toxicology, pharmacology, carcinogenicity, and genotoxicity.
7
8     Your task is to analyze the evidence retrieved so far and synthesize key insights related to the original
9     research question.
10
11     Initial Question:
12     {question}
13
14     Evidence History:
15     {combined}
16
17     Your analysis should:
18     - Summarize findings by categories (e.g., In Vitro results, In Vivo findings, Mechanistic data, Regulatory
19     context).
20     - Clearly highlight which parts of the question have been answered and which remain open.
21     - Use scientific specificity where possible (mention doses, assay names, species, study types).
22     - Do NOT speculate – only use what's present in the evidence.
23     - Avoid repeating full retrievals; focus on synthesizing and organizing insights.
24
25     Conclude your analysis by stating whether the current evidence is:
26     - Nearly complete (but still needs confirmation or depth), OR
27     - Clearly incomplete in key dimensions (e.g., missing quantitative data, regulatory framing, etc.).
28     """
29
30     response = bedrock_runtime.invoke_model(
31         modelId=MODEL_ID,
32         contentType="application/json",
33         accept="application/json",
34         body=json.dumps(
35             {
36                 "anthropic_version": "bedrock-2023-05-31",
37                 "messages": [{"role": "user", "content": prompt}],
38                 "max_tokens": 2048,
39                 "temperature": 0.2,

```

The Analyzer is responsible for delivering a summarizing conclusion that integrates the initial question, the compiled responses from successive iterations, and the relevant references. The summary must:

- Be structured by category;
- Clearly identify which aspects of the original question have been addressed and which remain unanswered;

- Avoid speculation;
- Exclude citations or reference mentions in the synthesis;
- Be curated so that only references explicitly addressing the compound or subject of the query are included - any content derived from non-specific references must be excluded from the final summary.

The model used is: `MODEL_ID = "anthropic.claude-3-sonnet-20240229-v1:0"`.

Analyzer Vitic [🔗](#)

This Analyzer has a similar function to the Analyzer for KB, the only difference is that it will use the history of the Vitic agent query and answers. The code is shown in the following code snippet:

```

1 combined = f"\n\nInitial Question: {question}"
2 combined += "\n\n".join(f"Query: {r['query']}\nResult: {r['result']}" for r in retrieval_history_sql)
3
4 prompt = f"""
5 You are an expert scientific analysis agent supporting a biomedical research workflow focused on
6 toxicology, pharmacology, carcinogenicity, and genotoxicity.
7
8 Your task is to analyze the evidence retrieved so far and synthesize key insights related to the original
9 research question.
10
11 Initial Question:
12 {question}
13
14 Evidence History:
15 {combined}
16
17 Your analysis should:
18 - Summarize findings by categories (e.g., In Vitro results, In Vivo findings, Mechanistic data, Regulatory
19 context).
20 - Clearly highlight which parts of the question have been answered and which remain open.
21 - Use scientific specificity where possible (mention doses, assay names, species, study types).
22 - Do NOT speculate – only use what's present in the evidence.
23 - Avoid repeating full retrievals; focus on synthesizing and organizing insights.
24
25 Conclude your analysis by stating whether the current evidence is:
26 - Nearly complete (but still needs confirmation or depth), OR
27 - Clearly incomplete in key dimensions (e.g., missing quantitative data, regulatory framing, etc.).
28 """

```

The instructions can be summarised as following:

- Summarise findings by category (In Vitro, In Vivo, Mechanistic, Regulatory).
- Highlight which parts of the question are answered and which remain open.
- Use specific scientific details (doses, assays, species, study types).
- Do not speculate — only include what's in the evidence.
- Synthesize insights without repeating full evidence text.
- Conclude whether the evidence is nearly complete or clearly incomplete.

The model used is: `MODEL_ID = "anthropic.claude-3-sonnet-20240229-v1:0"`.

Reflector [🔗](#)

The Reflector agent will use summarisation from both KB and Vitic Analysers to generate the Verdict and Justification. The code for its instructions is shown here:

```

1  prompt = f"""
2      You are a biomedical domain expert responsible for evaluating whether the current synthesized answer
sufficiently addresses
3      the original question.
4
5      Domain: Toxicology, Genotoxicity, Pharmacology, Carcinogenicity
6      Knowledge Base: Peer-reviewed literature and regulatory-grade studies
7      SQL: Vitic database
8      Context: All insights must be grounded in the provided data – no assumptions.
9
10     Question:
11     {original_question}
12
13     Synthesized Answer from Knowledge Base:
14     {synthesized_kb}
15
16     Synthesized Answer from Vitic SQL Database:
17     {synthesized_sql}
18
19     Instructions:
20     - Evaluate if the synthesis answers the scientific scope of the question (e.g., potency,
mechanisms, assay results, regulatory implications).
21     - Determine whether the synthesis shows specific, verifiable evidence (e.g., numerical data, assay
names, dose levels, species, ICH/FDA context).
22     - Apply an information relevance threshold of {threshold_value} – does the current synthesis meet
or exceed this threshold?
23
24     Respond ONLY with:
25     Verdict: COMPLETED or INCOMPLETE
26     Justification: (Brief explanation of what is sufficient or what is missing – be specific, e.g., “lacks
assay dose data” or
27     “no regulatory conclusion presented”)
28     """

```

The function of the Reflector is:

- Assess if the synthesis addresses the scientific scope (potency, mechanisms, assays, regulatory context).
- Check for specific, verifiable evidence (e.g., doses, assay names, species, regulatory references).
- Judge if it meets the set relevance threshold.

The model used is: `MODEL_ID = "anthropic.claude-3-sonnet-20240229-v1:0"`.

Finalizer [🔗](#)

The following code snippets shows the instructions for this agent:

```

1  prompt = f"""
2      You are an expert scientific analysis agent supporting a biomedical research workflow focused on
toxicology, pharmacology, carcinogenicity,
3      and genotoxicity.
4
5      Your task is to take the two synthesis generated from two systems (Knowledge base and Vitic) and generate
a unique summary.
6
7      Initial Question:
8      {question}
9
10     Synthesized Knowledge Base:

```



```

11     {synthesized_kb}
12
13     Synthesized Vitic:
14     {synthesized_sql}
15
16     Your summarisation should:
17     - Keep all details.
18     - Remove the references to the way that the information was retrieved, such as reference to tables, or
    knowledge base.
19     - Use scientific specificity where possible (mention doses, assay names, species, study types).
20     - Do NOT speculate – only use what's present in the evidence.
21     - Avoid repeating full retrievals; focus on synthesizing and organizing insights.
22
23     """

```

The instructions can be summarised as following:

- Combine the two syntheses (from Knowledge Base and Vitic) into one unified summary.
- Preserve all scientific details.
- Remove any mention of data sources or retrieval methods.
- Use specific scientific terms (e.g., doses, assays, species, study types).
- Do not speculate — base the summary strictly on the provided evidence.
- Avoid repeating raw data — synthesize and organise the insights clearly.

The model used is: `MODEL_ID = "anthropic.claude-3-sonnet-20240229-v1:0"`.

Structure (v2) [↗](#)

In this version, we replace the Bedrock Agent call from Vitic Retriever [Vitic Agent V2](#) with a lambda call for the [Vitic Agent with Action Group](#).

