Functional Requirements

- Admin can scrap the webpage and subpages of a given url and store them as document files (html, pdf, doc, etc)
- User can ask questions and get a response based on information extracted from the files
- User can provide feedback to the response in the form of thumbs up or down. If thumbs down, user can provide additional text on why and what the correct answer should be.
- Questions, responses, and feedback are logged and Admin can query the data for debugging and analysis

Admin API

```
POST /scrap_url
input {
    REQ input_url STRING or URL
    OPT recursion_limit INT = 1
}
response {
    run_id ID
}
```

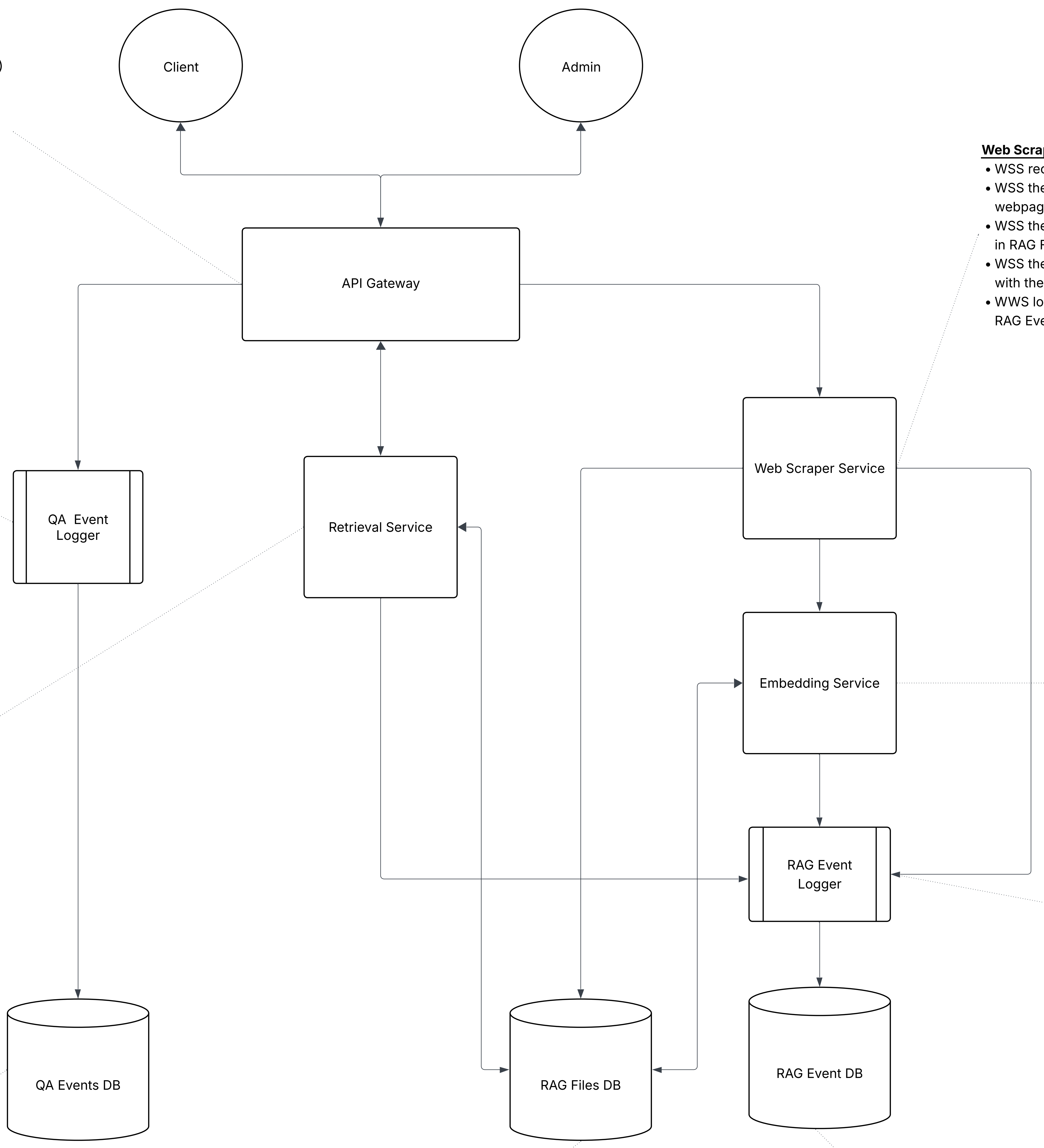
Client API

```
asnyc GET /ask
input {
    REQ question STRING
}
response {
    answer_id ID
    answer STRING
    sources[] ARRAY[JSON_STRING]
}

async POST /feedback
input {
    REQ answer_id
}
response {
    feedback_id ID
}
```

API Gateway

- Routes Admin API requests (scrap) to the Web Scraper service
- Routes Client API requests (ask, feedback) to the Retrieval Service
- Authenticate Client and Admin before they can make any API requests
- Log Client API requests using QA Event Logger



QA Event Logger + DB

- QA Event Logger is a logging service that logs all the events related to clients question, response, and feedback
- QA Event DB stores this information in a relational database

Retrieval Service

- RS receives input from /ask API
- RS then queries RAG Files DB to get the text chunks based on the question asked
    - RS logs retrieval events using RAG Event Logger
- RS then queries the ChatGPT API using a prompt based on the question text and text chunks retrieved
    - RS logs ChatGPT event using RAG Event Logger
- RS passes the response back to the Client via the API Gateway

Web Scraper Service

- WSS receives inputs from /scrap_url API
- WSS then recursively scraps the webpages and subpages from the url
- WSS then stores the scraped documents in RAG Files DB
- WSS then invokes the Embedding Service with the document filespaths
- WWS logs each webpage scraped using RAG Event Logger

Embedding Service

- ES receives the filepaths input for WSS
- ES extracts text chunks from documents from each filepath and store the text chunks in RAG Files DB
    - ES logs each document text extraction using RAG Event Logger
- ES converts text chunks into text embeddings and store text embeddings in RAG Files DB
    - ES logs each conversion using RAG Event Logger

RAG Event Logger + DB

- RAG Event Logger is a logger service that logs all events related to the RAG pipeline: getting the data, embedding the data, and retrieving the data.
- RAG DB stores all of this information in a relational database.
    - This should not be confused with the RAG Files DB which stores the intermediate and final output documents & text embeddings from the RAG pipeline

QA Events Data Model

User
- user_id ID (PK)
- answer_id[] ARRAY[ID]
- ...metadata

Answer
- answer_id ID (PK)
- answer_text STRING
- sources[] ARRAY[JSON_STRING]
- question_text STRING
- feedback_id NULLABLE[ID]
- timestamp TIMESTAMP

Feedback
- feedback_id ID (PK)
- answer_id ID (SK)
- liked NULLABLE[bool]
- reason NULLABLE[STRING]
- timestamp TIMESTAMP

RAG Files DB

- RAG Files DB is a filesystem that stores output documents from each step of the RAG system
- Web Scraper Service stores the scraped documents in this DB
- Embedding Service stores text chunks extracted from the scraped documents
- Embedding Service stores vector embeddings converted from the text chunks
- Retrieval Service queries the vector embeddings and convert them back to text chunks before constructing a response to user questions

RAG Events Data Model

WebScraperEvent
- run_id (PK)
- input_url URL
- output_path FILEPATH
- event_type STARTED/COMPLETED/FAILED
- reason STRING
- timestamp TIMESTAMP

EmbeddingEvent
- run_id (PK)
- input_path FILEPATH
- chunks_path FILEPATH
- event_type STARTED/COMPLETED/FAILED
- reason STRING
- timestamp TIMESTAMP

RetrievalEvent
- request_id ID (PK)
- question_text STRING
- text_chunks[] ARRAY[JSON_STRING]
- timestamp TIMESTAMP