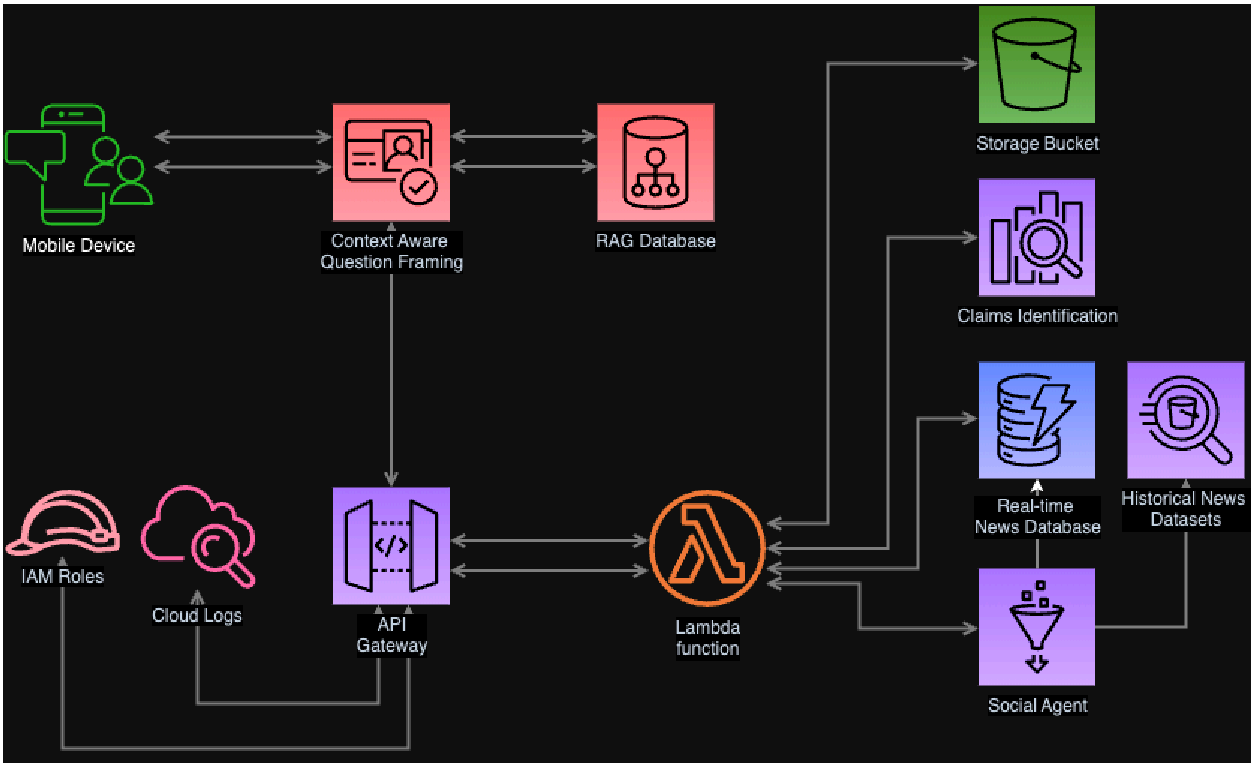

The system we intend to use is following the above architecture generalized across cloud providers[1]. We have investigated a minimalistic version of this architecture (minus logging and online information access) in order to develop a maternal health system and a menstrual health system called Sakhi, incubated at MIT in their flagship social innovation accelerator. This gives us confidence to say that such a model can be made to scale to a number of requests at a very reasonable cost, less than $0.2 per conversation for up to 5000 users with 15 conversations a month. We have not demonstrated the caching and engineering optimizations possible beyond using a serverless model, in order to avoid complexities with this architecture but we are very mindful of–and experienced with–optimizations possible to querying.

The purpose of each component is described below. Broadly, data comes in from the individual's device, gets answered through an offline context-specific RAG system and accesses an online social agent if needed to answer complex queries and have a dialogue[2]. We define a serverless lambda (or cloud) function that stores questions in a bucket for future optimizations, then identifies the claims in a query, passes them on to the socially intelligent agent. This agent answers questions informed by both a real time news database updated with the most recent events, as well as internet access to draw on additional sources. It has been trained to semantically match the publication tone and style as well as align with the goals of the publisher. Historical news datasets

[1] despite the use of the 'Lambda' terminology inspired by AWS (which are the equivalent of Cloud functions on GCP, for instance)

[2] The default functionality can be to always have a dialogue.

pictured in the diagram have been used to train the system (represented as another component but being clear about their relevance here).

1. **Mobile Device:** Acts as the user interface, allowing individuals to interact with the system through their devices.
2. **Context Aware Question Framing:** Processes user queries by framing them within the appropriate context for accurate information retrieval.
3. **RAG Database:** A vector database that stores relevant news data and knowledge graphs to support context-aware question framing.
4. **Storage Bucket:** Provides scalable storage for data, including queries, and additionally any relevant user interactions.
5. **Claims Identification:** Analyzes data to identify what constitutes claims in order to pass it to the subsequent component.
6. **Real-time News Database:** Stores and updates news articles in real-time as provided by the publisher in order to provide current information for analysis.
7. **Historical News Datasets:** Contains archived news data for reference and for fine tuning the models. Also useful for comparison in information verification.
8. **Social (Intelligent) Agent:** GenAI model such as a large language model that interacts with users to facilitate conversations grounded in a real-time database of breaking news grounding online information.
9. **Lambda Function:** Executes backend processes and logic to handle data processing and integration tasks efficiently.
10. **API Gateway:** Manages and routes requests between the mobile device and backend services, ensuring secure communication.
11. **IAM Roles:** Defines permissions and access controls for secure interaction with cloud resources.
12. **Cloud Logs:** Collects and stores logs for monitoring and auditing system activities.

User Interaction

Users interact with the system through a private messaging application. Here's how the interaction is facilitated:

1. Query Submission: Users send queries or messages through the application interface.
2. Processing and Retrieval: The system processes the query, retrieves relevant information from the vector database, and augments the query with this data.
3. Response Generation: The LLM generates a response based on the augmented query, ensuring it is contextually appropriate and informative.
4. Response Delivery: The generated response is sent back to the user through the messaging application, completing the interaction loop.

Overall, the proposed solution leverages cutting-edge technologies to create an intelligent, responsive, and scalable messaging system that meets user needs effectively.