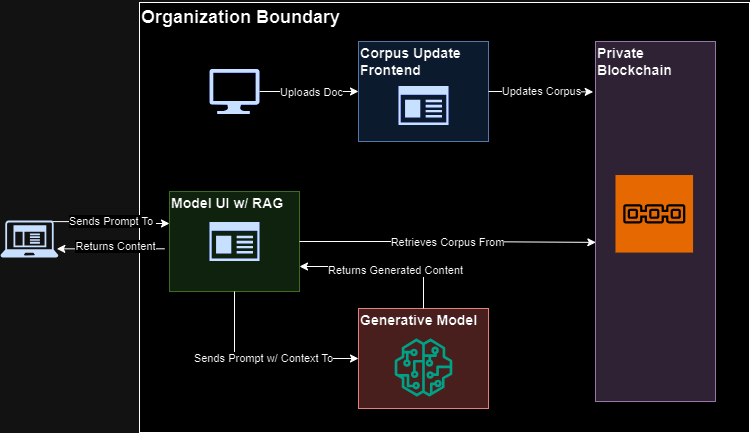
Decentralized RAG: Combining Retrieval Augmented Generation with Blockchains

Reed White (12367576)

# Project Architecture

## High-Level Application Flow

The following architecture is expected to be implemented in this project:

FIGURE I. High-Level Architecture

With the following components:

* Model Frontend w/ RAG: the model interaction GUI webapp and the RAG logic will run in a single process. The generative model should be loaded onto an end-user-facing server and exposed to the Internet or desired intranet. This only reads from the Corpus smart contract.
* Corpus Administrator Frontend: the corpus GUI webapp should be loaded onto an internal server and only exposed to administrative users. This reads and writes to the Corpus smart contract.
* Private Blockchain: the blockchain should be deployed and maintained internally to the organization’s/company’s logical boundaries. It should only be exposed to systems that connect to it, such as the model or the administrative frontend. It contains the Corpus smart contract that stores the corpus.
* Administrative Users: these are internal company users that maintain the corpus. They should connect to the corpus administrator frontend via browsers within the organization’s boundaries. They will require MetaMask to perform more operations.
* End Users: these are external company users/customers that interact with the model frontend. They connect to the model via browsers over the Internet or desired intranet. They have no additional requirements.

## Application Flows

Users for this system are split into two distinct categories: administrators and End users. Administrators represent internal users within the organizational boundaries of the hosting company that submit documents to, remove documents from, and maintain the corpus of their company. End users represent external users outside of the organizational boundaries that interact with the chatbot/RAG. The high-level application diagram can be broken down to represent each flow:

### Administrators

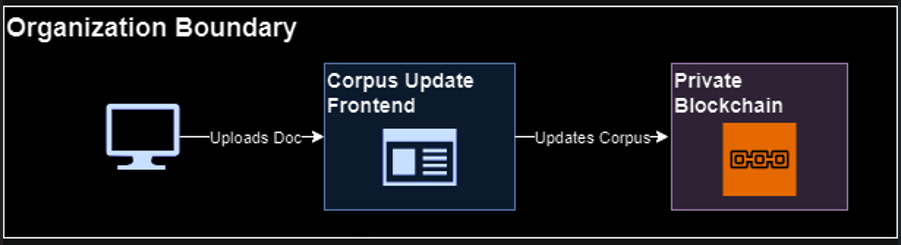


FIGURE II. Administrator Flow

Administrator users modify the corpus stored in the blockchain via the Node.js Corpus Administrator Frontend application. MetaMask is required in order to submit or modify (write) documents via the frontend, but not to retrieve (read) documents.

### End Users

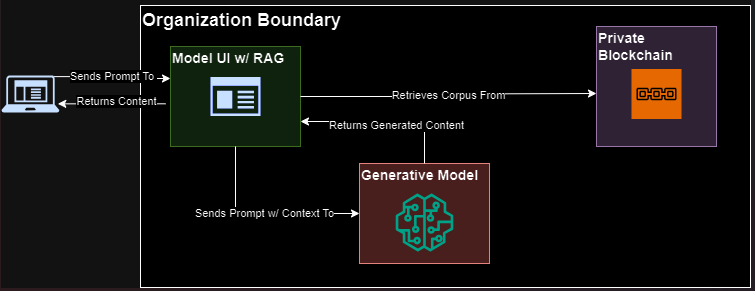


FIGURE III. End User Flow

End users interact solely with the chatbot/Model Frontend. The current design does not require users to authenticate nor have MetaMask, as it is intended to be a public-facing chatbot. Because the RAG system only reads from the blockchain, it also does not have any MetaMask, wallet, or private keys.