Carthy.ai

High level Architecture

MVP initial customer abilities

Admin interface:(admin & user)

- Identify and enable org data sources;
- Identify and enable private data sources- all kind files; text ; audio(?) video(?); folders ; folder monitoring (triggering)
- Identify and enable retrieval api (internal & external)
  - We will need to create predefined retrieval api data sources per vertical
- Create public and private libraries by content - based on user files & data and admin preferences

User capabilities

- Create email/ chat request. - Create request endpoint
    - Create endpoint integration
      - E-mail
      - Zapier
      - Upload files for context or more data \text
- refer to public private library
- Refer to data source

Carthy response to user :

Send back to the user a formatted answer based on customer request

Identity

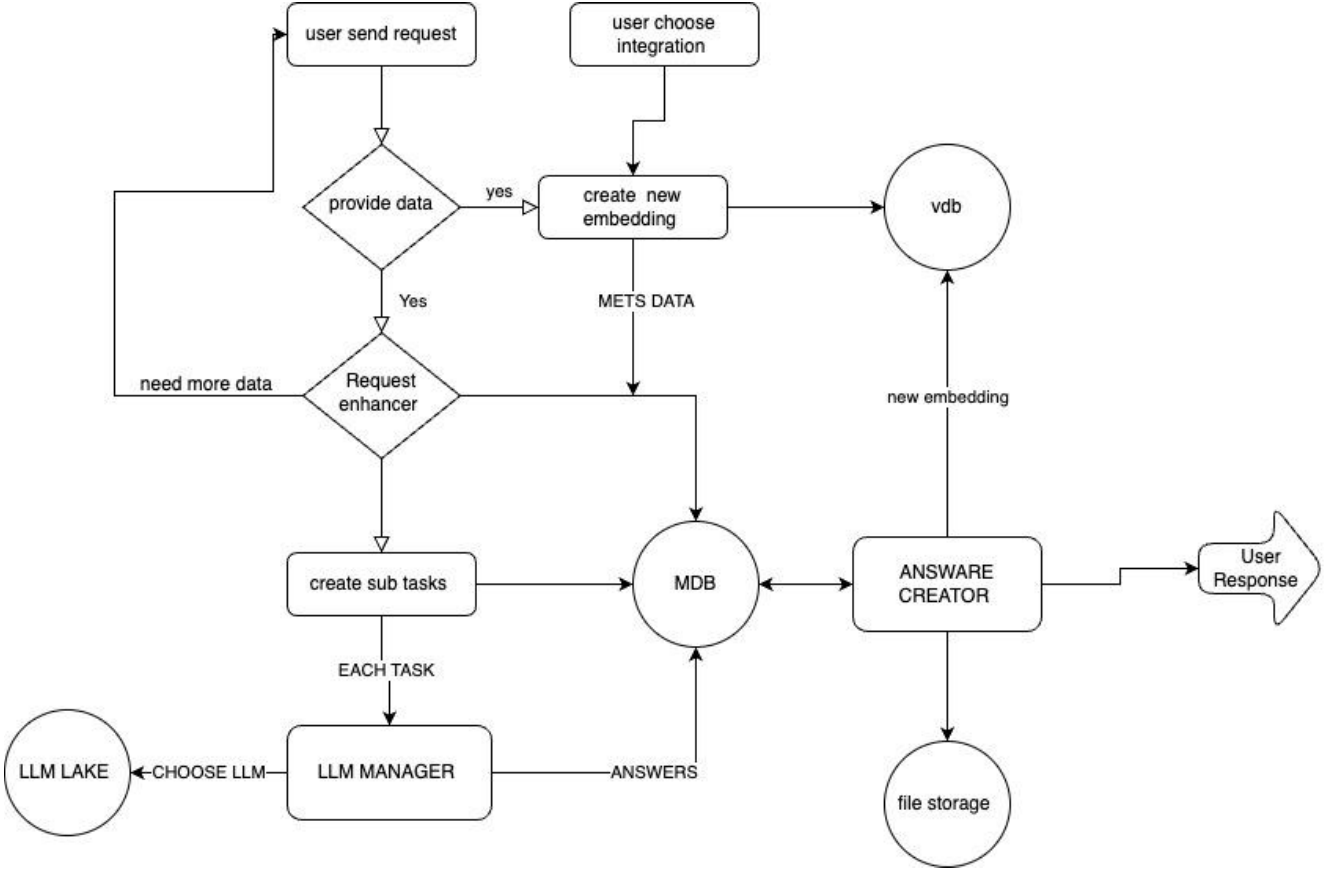
1. User (one time) Authentication
2. User unique identity - email address for email request or chat request both related to user unique entity on Carthy.

The users will authenticate to the system using oauth - Preferably using Auth0 as IDP

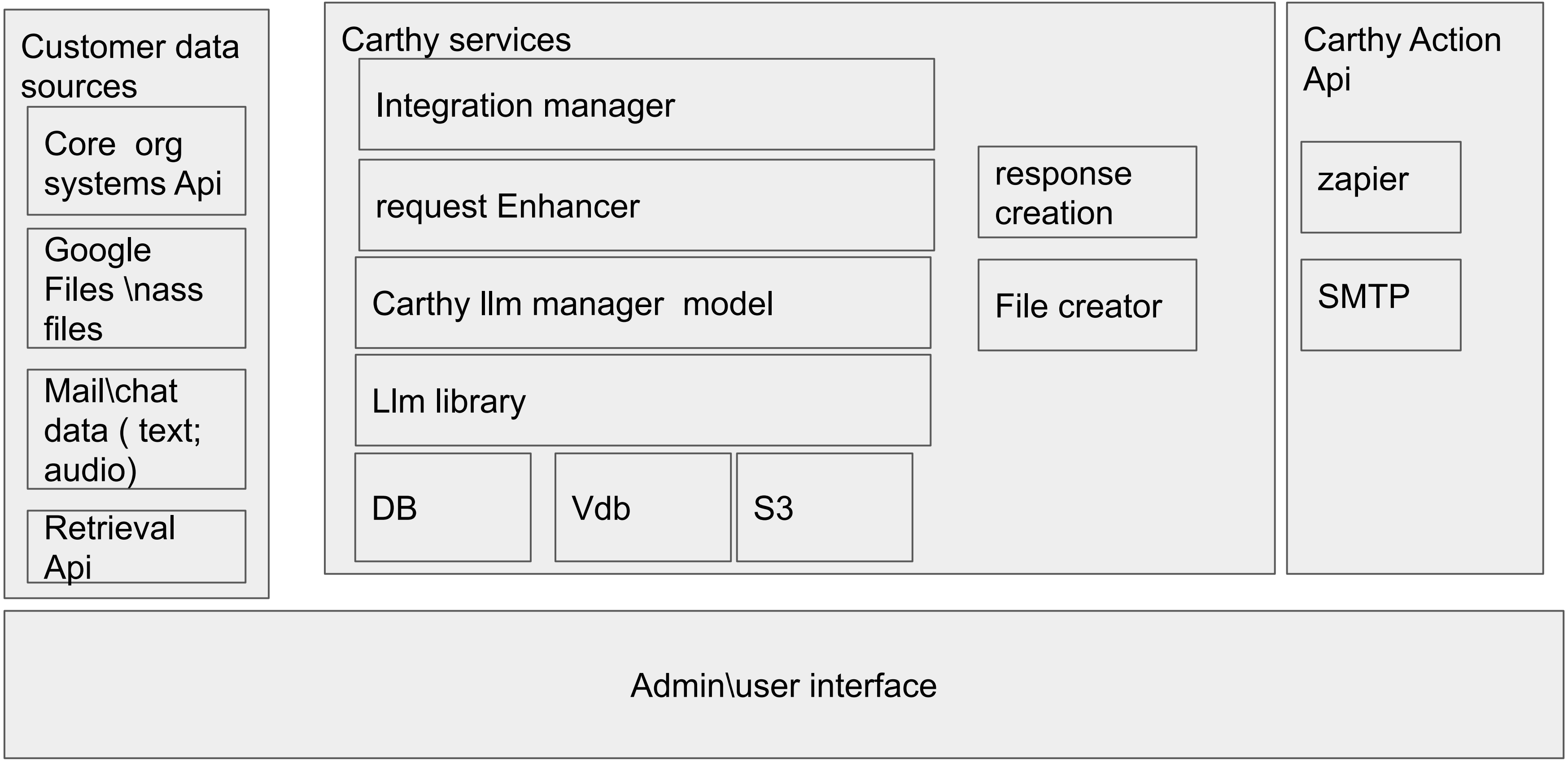
This will allow the admins to connect to their own SSO (GApps, Okta, MS AD)

This will also allow users to create a “simple” account on Auth0

User WorkFlow



Carthy services-HLD



Customer data sources

Core org systems Api

Google Files \nass files

Mail\chat data (text; audio)

Retrieval Api

Carthy services

Integration manager

request Enhancer

Carthy llm manager model

Llm library

DB

Vdb

S3

response creation

File creator

Carthy Action Api

zapier

SMTP

Admin\user interface

Carthy platform main components

User Interface: This component represents the interface through which the user interacts with the carthy solution. It allows users to send requests and receive answers in the defined format.

Data Sources: This component represents various data sources such as files, chat, documents, databases, APIs, email etc from which the software solution integrates and retrieves data.

Email Service: This component handles the email communication between the customer and carthy solution. It receives requests sent to the email address and refers them to relevant data or requests for additional data.

Integration manager- This component responsible to manage all data sources api integrations

Data Uploader: This component is responsible for uploading the data from the sources. It transforms the data into embeddings and stores them in the vector storage.

Vector Storage: This component stores the transformed data in vector form for efficient retrieval and processing.

Carthy platform main components

Request Enhancer: This engine receives customer requests and understands the context. It creates a sequence of sub tasks based on the request and enhances it for further processing.

Entity Database: This component stores all the sub tasks, files metadata, requests, and answers. It acts as a central repository for managing and organizing the data.

LLM Chain Engine: This engine analyzes each sub task and determines the most relevant LLM - Language Model manager. It provide the most relevant LLM for each task . It decides on the correct sequence of LLMs to be used for processing the sub tasks.

Answer Processor: This component processes the sequence of sub tasks and generates the final answer. It formats the answer according to the predefined format and sends it back to the user

File storage;This component stores the files that was created as the answer for future requests as context data

User Interface: This component represents the interface through which the user interacts with the software solution. It allows users to send requests and receive answers in the defined format.

Data Sources: This component represents various data sources such as files, chat, documents, databases, APIs, email etc from which the software solution integrates and retrieves data.

Data Uploader: This component is responsible for uploading the data from the sources. It transforms the data into embeddings and stores them in the vector storage.

Email Service: This component handles the email communication between the customer and the software solution. It receives requests sent to the email address and refers them to relevant data or requests for additional data.

Vector Storage: This component stores the transformed data in vector form for efficient retrieval and processing.

Request Enhancer: This engine receives customer requests and understands the context. It creates a sequence of sub tasks based on the request and enhances it for further processing.

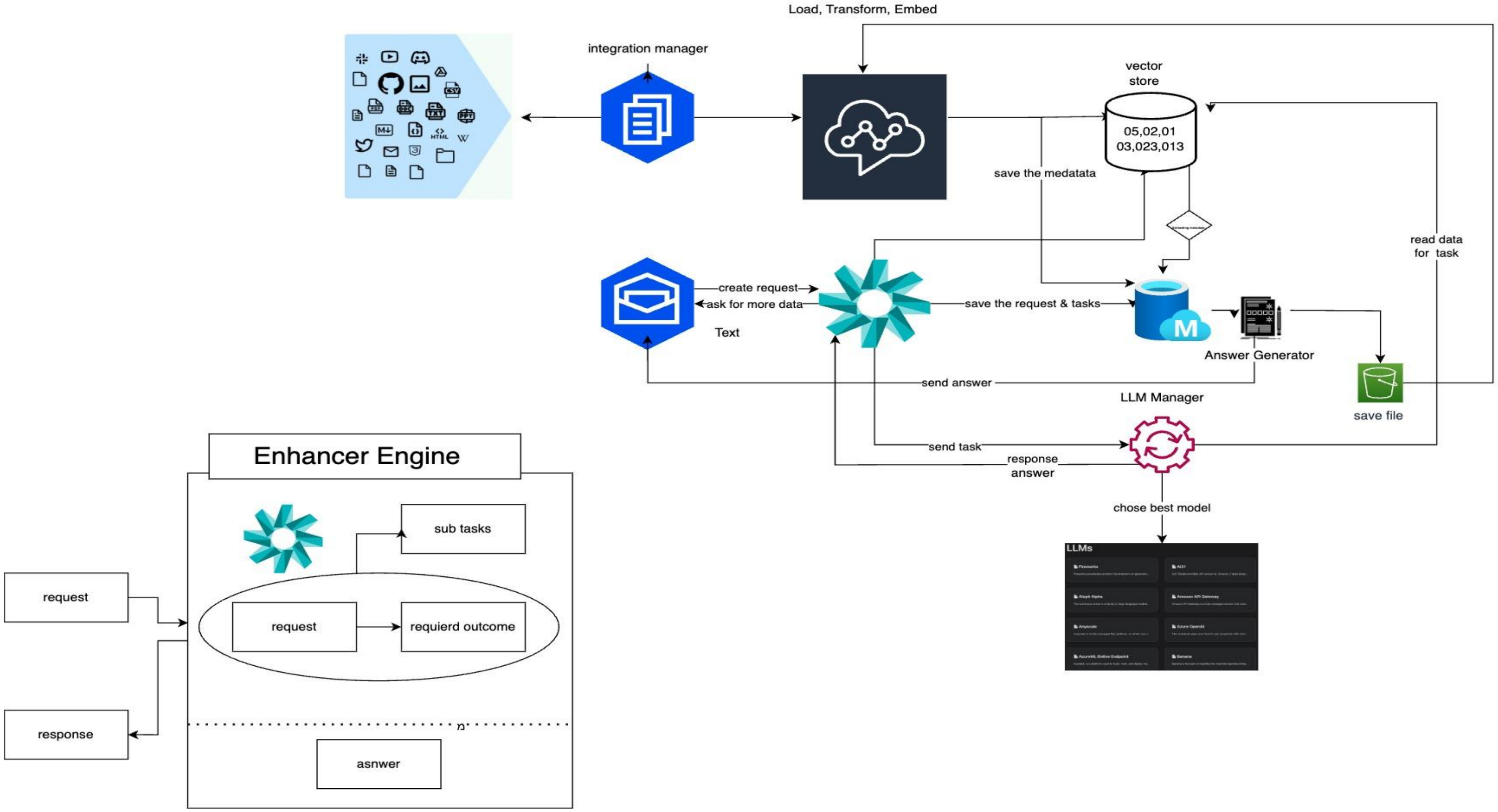
Entity Database: This component stores all the sub tasks, files metadata, requests, and answers. It acts as a central repository for managing and organizing the data.

LLM Chain Engine: This engine analyzes each sub task and determines the most relevant LLM - Language Model manager. It provide the most relevant LLM for each task . It decides on the correct sequence of LLMs to be used for processing the sub tasks.

Answer Processor: This component processes the sequence of sub tasks and generates the final answer. It formats the answer according to the predefined format and sends it back to the user

File storage;This component stores the files that was created as the answer for future requests as context data

Platform HLD Architecture



Open Discussion

- How to create embed for the files (inhouse/external service)
- What is the enhancer how can it understand the request
- Is there any preferred vector storage (or
- What are the option regarding the implementation llm manager (using langchain or is there other option)
- let s defined the first data source integration