


SKOPOS' SCANNING SERVICE

3:49 AM
8/13/2025 

Arranged by: Nathan Jovial Hartono - 13523032
- I just want to sleep

TABLE OF CONTENTS

[Business Description](#)

[Main Service](#)

[Business Model](#)

[Image 1. Business Model Canvas](#)

[Business Strategy](#)

[Business Capability Model](#)

[Image 2. Business Capability Model](#)

[Software Requirements](#)

[Functional requirements](#)

[Table 1. Functional Requirement](#)

[Use Case Identification](#)

[Table 2. Use Case Identification](#)

[Use Case Scenario](#)

[UC01 Scenario 1](#)

[Table 3. UC01 Scenario 1](#)

[UC01 Scenario 2](#)

[Table 4. UC01 Scenario 2](#)

[UC02 Scenario 1](#)

[Table 5. UC02 Scenario 1](#)

[UC02 Scenario 2](#)

[Table 6. UC02 Scenario 2](#)

[UC03 Scenario 1](#)

[Table 7. UC03 Scenario 1](#)

[UC03 Scenario 2](#)

[Table 8. UC03 Scenario 2](#)

[UC04 Scenario 1](#)

[Table 9. UC04 Scenario 1](#)

[UC04 Scenario 2](#)

[Table 10. UC04 Scenario 2](#)

[UC05 Scenario 1](#)

[Table 11. UC05 Scenario 1](#)

[UC05 Scenario 2](#)

[Table 12. UC05 Scenario 2](#)

[UC05 Scenario 3](#)

[Table 13. UC05 Scenario 3](#)

[UC06 Scenario 1](#)

[Table 14. UC06 Scenario 1](#)

[UC07 Scenario 1](#)

[Table 15. UC07 Scenario 1](#)

[UC07 Scenario 2](#)

[Table 16. UC07 Scenario 2](#)

[Sequence Diagram](#)

[UC01 Diagrams](#)

[Image 3. UC01 Scenario 1 Sequence Diagram](#)

[Image 4. UC01 Scenario 2 Sequence Diagram](#)

[UC02 Diagrams](#)

[Image 5. UC02 Scenario 1 Sequence Diagram](#)

[Image 6. UC02 Scenario 2 Sequence Diagram](#)

[UC03 Diagrams](#)

[Image 7. UC03 Scenario 1 Sequence Diagram](#)

[Image 8. UC03 Scenario 2 Sequence Diagram](#)

[UC04 Diagrams](#)

[Image 9. UC04 Scenario 1 Sequence Diagram](#)

[Image 10. UC04 Scenario 2 Sequence Diagram](#)

[UC05 Diagrams](#)

[Image 11. UC05 Scenario 1 Sequence Diagram](#)

[Image 12. UC05 Scenario 2 Sequence Diagram](#)

[Image 13. UC05 Scenario 3 Sequence Diagram](#)

[UC06 Diagrams](#)

[Image 14. UC06 Scenario 1 Sequence Diagram](#)

[UC07 Diagrams](#)

[Image 15. UC07 Scenario 1 Sequence Diagram](#)

[Image 16. UC07 Scenario 2 Sequence Diagram](#)

[Entity Relationship Diagram](#)

[Image 17. ER Diagram between Client and Service](#)

[Application Architecture Diagram](#)

[Image 18. Application Architecture Diagram](#)

Business Description

Main Service

The business provides API services where clients are able to perform text extraction from a PDF to perform a keyword search, which covers so many aspects in the process of hiring or just searching for a word in the client's text. Our service provides a user experience with high performance search functionality and guarantees no data is kept when a scanning occurs. Other than user experience, this service aims to provide a seamless experience for integrating the API to its respective client. The goal is to provide for the community, but if a client under the category of corporates or small businesses would like to integrate, then the service is charged for improving better cloud infrastructure to support the wide-scale usage of the services, while individuals are limited by its text count or file size.

Business Model

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
<ul style="list-style-type: none">• Cloud Service Provider: Microsoft• Collaboration with other software developers to integrate APIs seamlessly	<ul style="list-style-type: none">• App development• API maintenance• Cloud maintenance	<ul style="list-style-type: none">• A service involving CV uploads and keyword detection with secure API and seamless integration with other APIs	<ul style="list-style-type: none">• Self service resources and dedicated API maintenance	<ul style="list-style-type: none">• Hiring team of a company• committee hiring of an event• Individual research
	Key Resources		Channels	
	<ul style="list-style-type: none">• Cloud service• Software Developers• Cloud Maintenance Team		<ul style="list-style-type: none">• Web application• API documentation	
Cost Structure			Revenue Streams	
<ul style="list-style-type: none">• Software development• Cloud Infrastructure• API maintenance			<ul style="list-style-type: none">• Subscription based pricing for companies and enterprises looking to integrate the API	

Image 1. Business Model Canvas

The business idea simply revolves around providing a service that's free and reliable for certain groups of individuals. While the business does provide a simple web application, it is emphasized on its API implementation, and provides simple yet secure APIs for developers to implement. This is why it is emphasized on maintenance and Cloud's or VPS' infrastructure. The customers range from individuals looking to utilize the web application to large companies integrating with this API for a long period of time.

Business Strategy

Our strategy is driven by the freemium services we offer, which results in multiple competitive advantages against our competitors. Our advantages include:

- Convenient DX: We offer seamless, well-documented, and secure API services for developers. Developers can easily implement these services and build our reputation better.
- Enterprise-level exclusive: We offer exclusive features and more advanced and robust securities to companies or enterprises that seek to use our services as their own features, separating the users with freemium features and exclusive enterprises, which will pay for our future plans of improving scalability of the cloud service and attract more enterprises to use the same service with good reputation.
- Freemium with promising features: We offer freemium features that build a promising relationship between the client and our business, which might intrigue our clients into purchasing the more exclusive tier of this business.

We provide the freemium services to our clients in the hopes of obtaining revenue from the exclusive tier. A few key points our business might need to achieve are as follows:

- Obtain a large amount of clients to later evolve into a community and build respect to our business idea, whilst also creating potential exclusive tier customers within the community.
- Secure a number of exclusive tier customers within a time range of years to maintain revenue and improve the cloud infrastructure of our business to provide availability for more customers in need.
- Achieve profit every year to improve the quality and security of our service as our reputation climbs up to the most prominent entities of the scene.

Achieving and maintaining a community of our business idea can be done by simply providing more features and fulfilling that promise, gaining trust with better customer service, and improving transparency of business operations and plans on what to do next through social media or news journalism. Securing a number of prestigious users can be done by providing more special services and royalty to customers within the prestigious circle, ensuring exclusivity and reliability of our business. Other than that, our business might be able to sign a binding contract with other businesses to support each other's business idea with future benefits and collaborations. While achieving profit may be ambitious, but by building reputation and being given the spotlight by other successful businesses, we may be able to create an ecosystem of returning customers and a huge demand from other businesses interacting with our business, which then stabilizes the monetary value of our business.

Business Capability Model

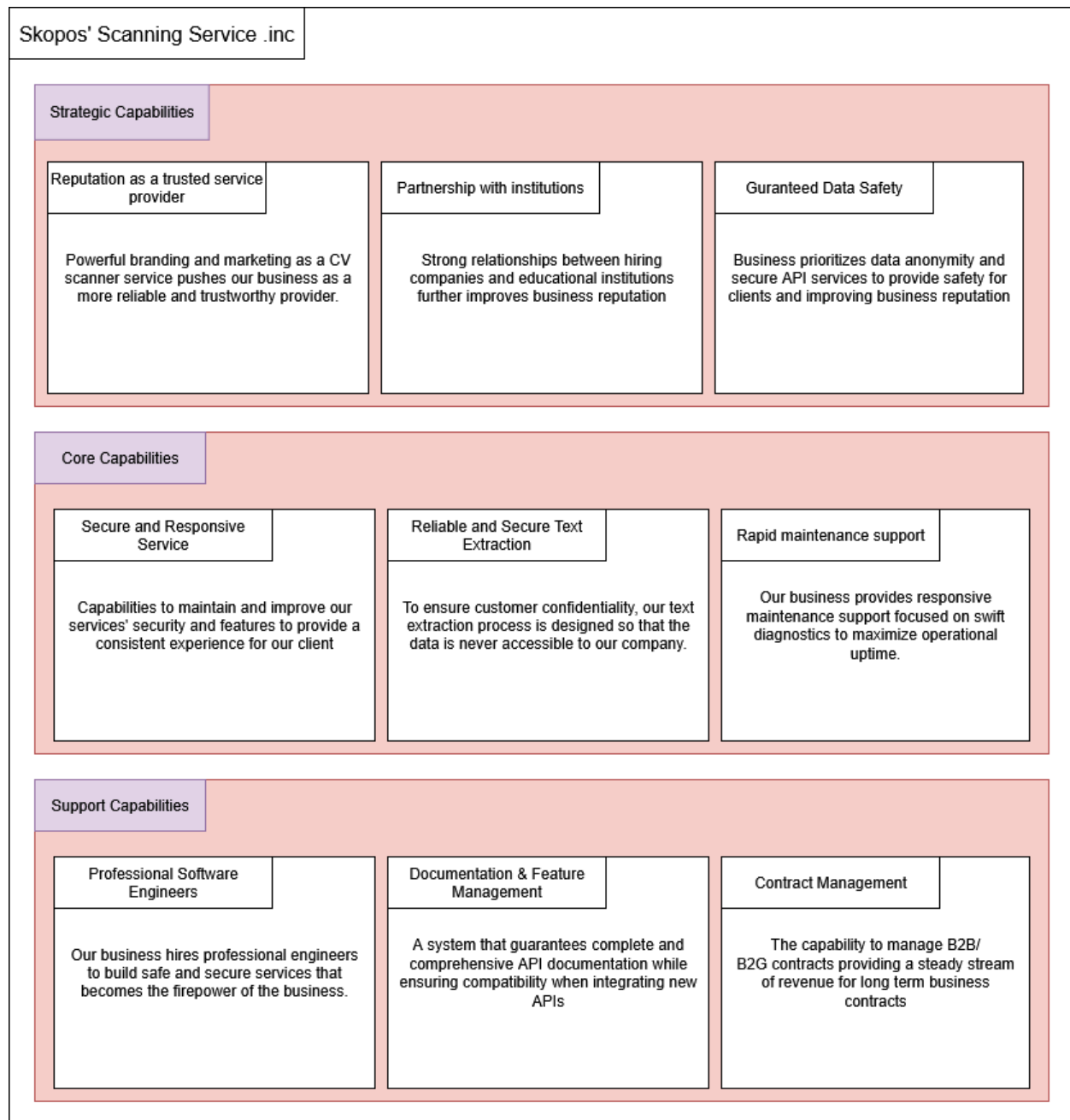


Image 2. Business Capability Model

Our business model provides a service API for clients, which utilizes Oauth2 authentication for safety and restriction use of our service. The service is a freemium, it divides clients into free tier users and premium tier users, where the premium tier provides more advanced features. The free tier only extends to text extraction from a PDF file, but the premium tier involves OCR technology scanning text from images in the PDF. Accessing the premium tier requires a client to pay a subscription fee to access the premium service. Aside from text extraction, the service supports keyword search using a fast algorithm “Aho-Corasik” and supports multiple keyword search at once, this feature is free for all clients.

The API service is run in a cloud infrastructure of Azure, provided by Microsoft, with a containerized environment with Docker to ensure consistency between development and production while also opening more ports for different applications in a cloud computing environment.

For the service itself, we utilize the programming language python using a backend framework, FastAPI, with documentation provided by SWAGGER API Documentation (access the [host]/docs endpoint). The framework is paired with database & data modelling libraries, i.e. sqlalchemy as the ORM and Pydantic to model the data for client request. The service is connected to a Postgresql database to store client information, like client id, client email, and its URI. For authentication and security, the service utilizes python-jose's cryptography library (JWT) for access token, and Bcrypt for generating client secrets when registering. The endpoint is protected with FastAPI's built-in OAuth2PasswordBearer class so unauthorized access isn't allowed. For text extraction and keyword search, we utilize libraries such as PyMuPDF (Fitz), Pytesseract, and Pyahocorasick.

Software Requirements

Functional requirements

ID	Requirement	Explanation
F01	<i>Client Account Management</i>	<ul style="list-style-type: none">• The system must provide an endpoint for new clients to register for API service access• The system must store client profile information, their id, client name, client email, URI, and their subscription tier• The system must allow the client to update its email and information
F02	<i>Authentication & Authorization Service</i>	<ul style="list-style-type: none">• The system must authenticate clients using the OAuth2 client credentials grant flow and issue a time-limited access token upon successful authentication• The system must validate a client's access token before accessing any endpoints• The system must authorize endpoints based off on the client's subscription tier
F03	<i>Text Extraction Service</i>	<ul style="list-style-type: none">• The system must provide an endpoint that accepts a PDF as file input and extract its text content to be returned via an HTTP request.• For premium tier, the system must provide an endpoint that utilizes Optical Character Recognition for text extraction from images found within the PDF file• For premium tier, the system must return an error message for unauthorized clients (free tier clients)
F04	<i>Keyword Search Service</i>	<ul style="list-style-type: none">• The system must provide an endpoint that accepts a body of text and a list of keyword input• The system must utilize an Aho-Corasik algorithm to perform keyword search from input text• The system must return a response indicating which keywords were found in the text
F05	<i>Subscription Management</i>	<ul style="list-style-type: none">• The system must provide an endpoint for free tier clients to upgrade their account to premium tier• The system must provide an endpoint for clients to manage their subscription, i.e. cancellation, extension, and more.

Table 1. Functional Requirement

Use Case Identification

No	Use Case	Explanation
UC01	Registering as a client	Sistem provides an endpoint for new clients to register
UC02	Authentication for token	System provides an authentication endpoint for clients to obtain their bearer token
UC03	Endpoint authorization	System must authenticate a token before a user is able to access the endpoint
UC04	Extracting Text (Free Tier)	System provides a text extraction service from a PDF with text
UC05	Extracting Text from Images (Premium Tier)	System provides a text extraction service from images within a PDF using OCR
UC06	Keyword Search	System provides service to search a keyword within a body text
UC07	Upgrading Tier	System provides an endpoint for user to obtain the premium subscription tier via payment

Table 2. Use Case Identification

Use Case Scenario

UC01 Scenario 1

Name: Registering as a client

Scenario: new client registration

<i>New client registration</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access the endpoint “/register-client” with its required request body	
	2. System returns a response of its client id and client secret to be used for authorization

Table 3. UC01 Scenario 1

UC01 Scenario 2

Name: Registering as a client

Scenario: User tries to register with existing client

<i>User tries to register with existing client</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access the endpoint “/register-client” with its required request body	
	2. System returns a response error

Table 4. UC01 Scenario 2

UC02 Scenario 1

Name: Authentication for token

Scenario: Requesting authentication token with provided client credentials

<i>Requesting authentication token with provided client credentials</i>	
<i>Actor action</i>	<i>System Reaction</i>

<i>Requesting authentication token with provided client credentials</i>	
1. Client access the endpoint “/oauth/token” with its required request body	
	2. System returns a response of access_token and token_type

Table 5. UC02 Scenario 1

UC02 Scenario 2

Name: Registering as a client

Scenario: Requesting authentication token with ineligible client credentials

<i>Requesting authentication token with ineligible client credentials</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access the endpoint “/oauth/token” with its required request body	
	2. System returns a response error

Table 6. UC02 Scenario 2

UC03 Scenario 1

Name: Endpoint authorization

Scenario: Authorization with valid token before endpoint access

<i>Authorization with valid token before endpoint access</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access any endpoint services that requires a token access with its valid token	
	2. System allows access to the endpoint service

Table 7. UC03 Scenario 1

UC03 Scenario 2

Name: Endpoint authorization

Scenario: Authorization with invalid token before endpoint access

<i>Authorization with invalid token before endpoint access</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access any endpoint services that requires a token access with its invalid token	
	2. System rejects access to the endpoint service

Table 8. UC03 Scenario 2

UC04 Scenario 1

Name: Extracting Text (Free Tier)

Scenario: Text extraction for PDF file <= 5 MB

<i>Text extraction for PDF file <= 5 MB</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access the “/extract-text” endpoint with its token and request body	
	2. System processes the text in the PDF file and returns a response of text content

Table 9. UC04 Scenario 1

UC04 Scenario 2

Name: Extracting Text (Free Tier)

Scenario: Text extraction for PDF file > 5 MB

<i>Text extraction for PDF file > 5 MB</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access the “/extract-text” endpoint with its token and request body	
	2. System returns a response error

Table 10. UC04 Scenario 2

UC05 Scenario 1

Name: Extracting Text from Images (Premium Tier)

Scenario: Text extraction for PDF file <= 50 MB

<i>Text extraction for PDF file <= 50 MB</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Premium Client access the “/extract-text-ocr” endpoint with its token and request body	
	2. System processes the text in the PDF file and returns a response of text content

Table 11. UC05 Scenario 1

UC05 Scenario 2

Name: Extracting Text (Free Tier)

Scenario: Text extraction for PDF file > 50 MB

<i>Text extraction for PDF file > 50 MB</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Premium Client access the “/extract-text-ocr” endpoint with its token and request body	
	2. System returns a response error

Table 12. UC05 Scenario 2

UC05 Scenario 3

Name: Extracting Text (Free Tier)

Scenario: Free tier client accessing the endpoint

<i>Free tier client accessing the endpoint</i>	
<i>Actor action</i>	<i>System Reaction</i>

<i>Free tier client accessing the endpoint</i>	
1. Free Tier Client access the “/extract-text-ocr” endpoint with its token and request body	
	2. System returns a response error

Table 13. UC05 Scenario 3

UC06 Scenario 1

Name: Keyword Search

Scenario: Search multiple keywords in text body

<i>Search multiple keywords in text body</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Client access the “/search-text” endpoint with its token and request body	
	2. System starts a search with the keyword inputs within the text body input

Table 14. UC06 Scenario 1

UC07 Scenario 1

Name: Upgrading Tier

Scenario: Free tier client tries to apply for premium tier

<i>Free tier client tries to apply for premium tier</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Free Tier Client access the “/upgrade-to-exclusive” endpoint with its request body	
	2. System processes the client credential and updates the tier to premium

Table 15. UC07 Scenario 1

UC07 Scenario 2

Name: Upgrading Tier

Scenario: Existing premium tier client tries to apply for premium tier

<i>Premium tier client tries to apply for premium tier</i>	
<i>Actor action</i>	<i>System Reaction</i>
1. Premium Client access the “/upgrade-to-exclusive” endpoint with its token and request body	
	2. System returns a response error

Table 16. UC07 Scenario 2

Sequence Diagram

UC01 Diagrams

UC01 Scenario 1: New Client Registration

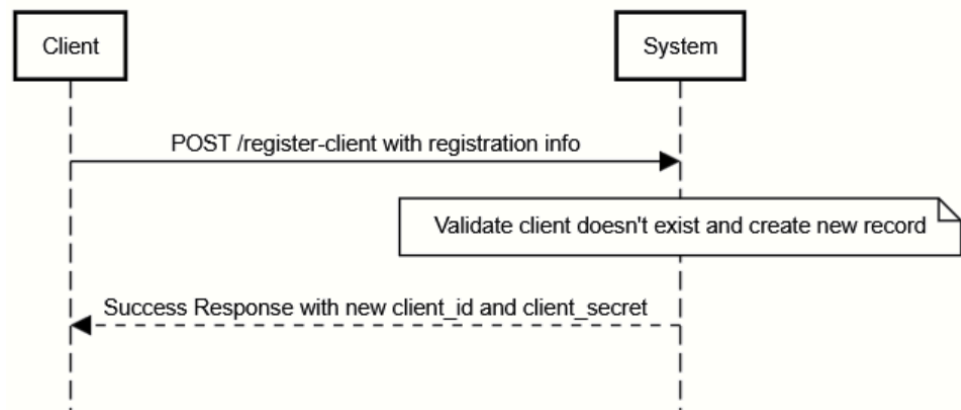


Image 3. UC01 Scenario 1 Sequence Diagram

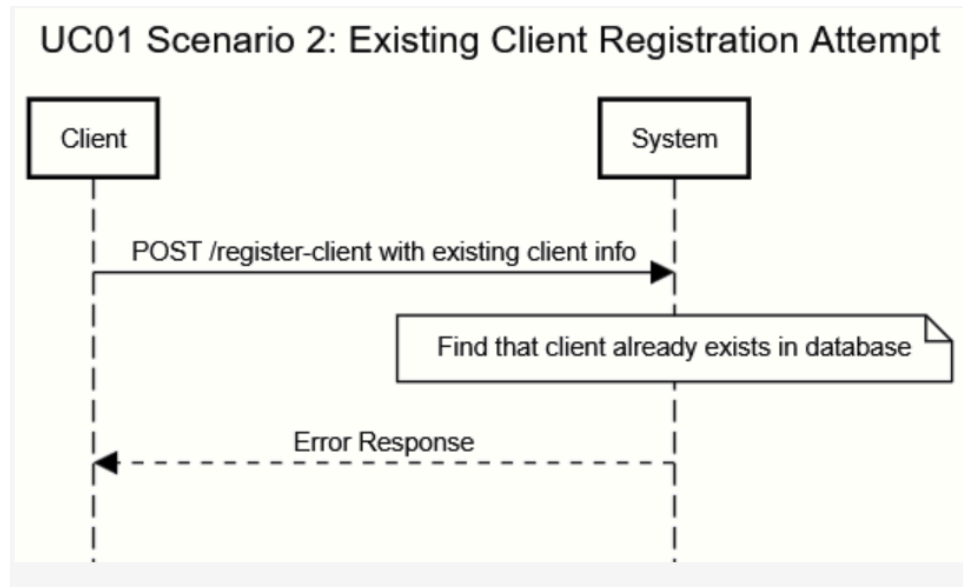


Image 4. UC01 Scenario 2 Sequence Diagram

UC02 Diagrams

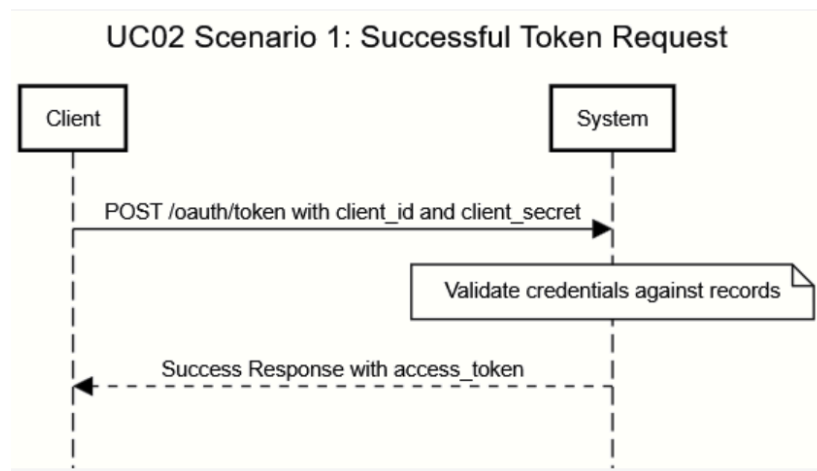


Image 5. UC02 Scenario 1 Sequence Diagram

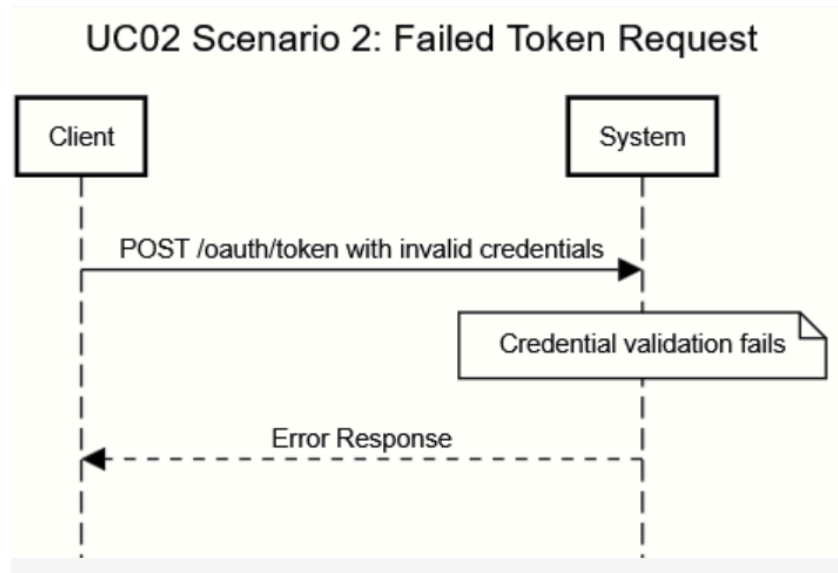


Image 6. UC02 Scenario 2 Sequence Diagram

UC03 Diagrams

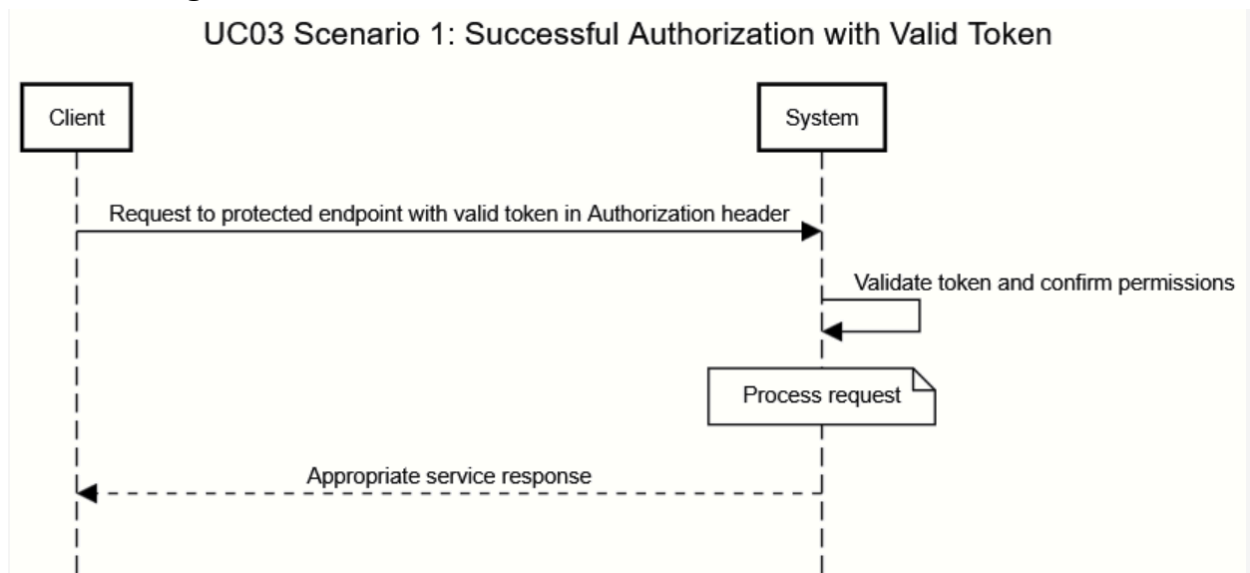


Image 7. UC03 Scenario 1 Sequence Diagram

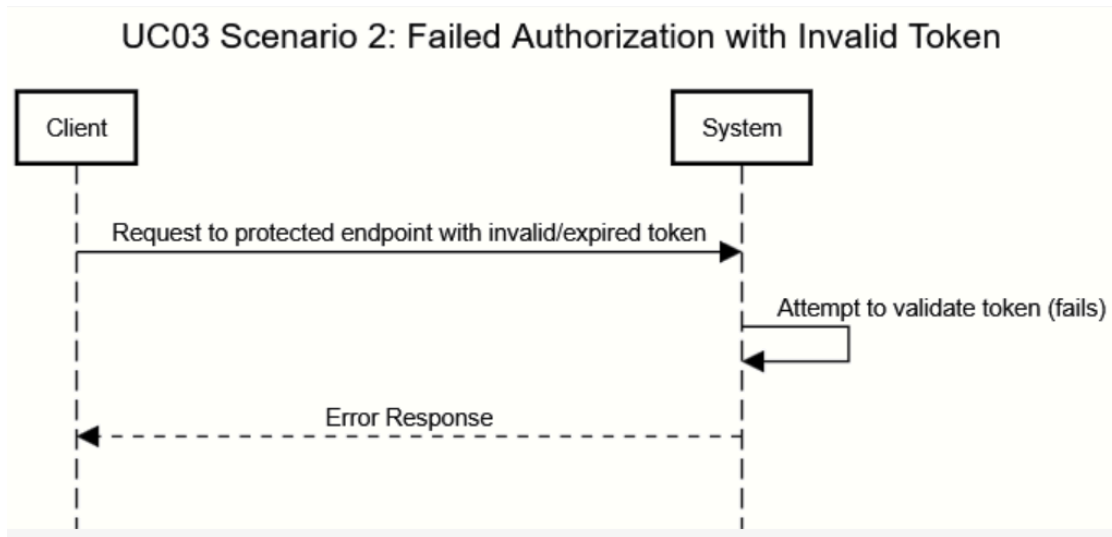


Image 8. UC03 Scenario 2 Sequence Diagram

UC04 Diagrams

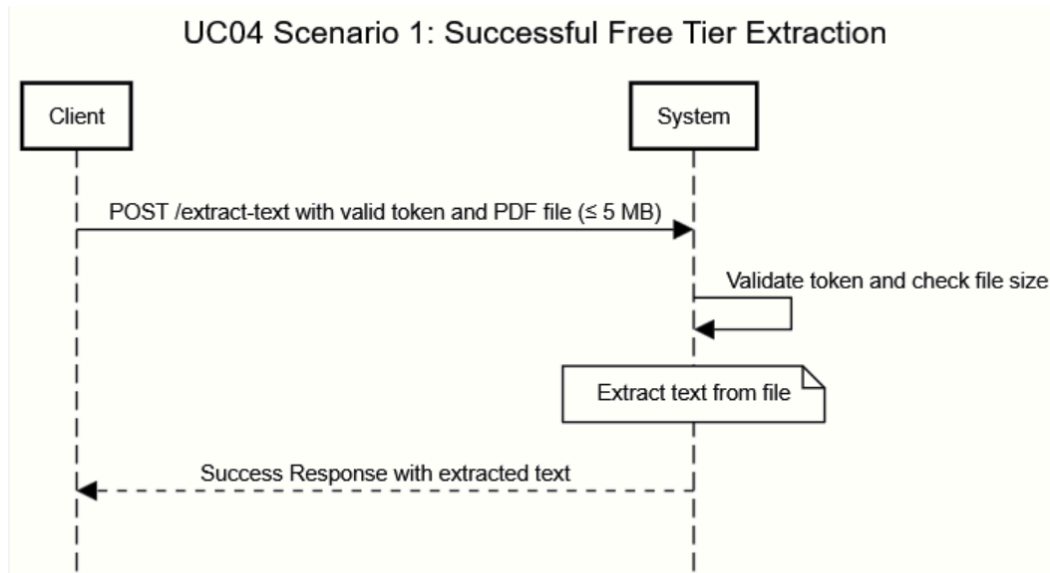


Image 9. UC04 Scenario 1 Sequence Diagram

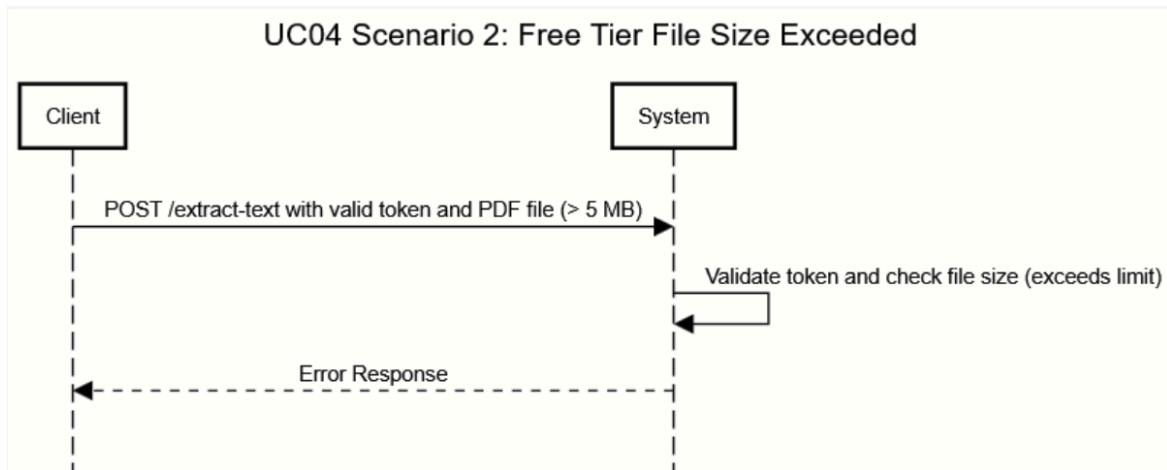


Image 10. UC04 Scenario 2 Sequence Diagram

UC05 Diagrams

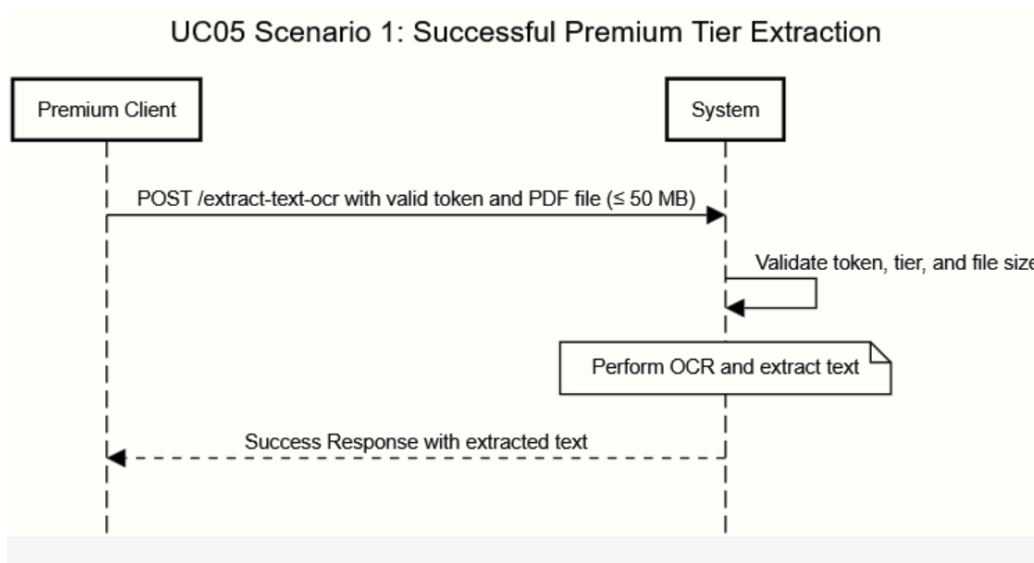


Image 11. UC05 Scenario 1 Sequence Diagram

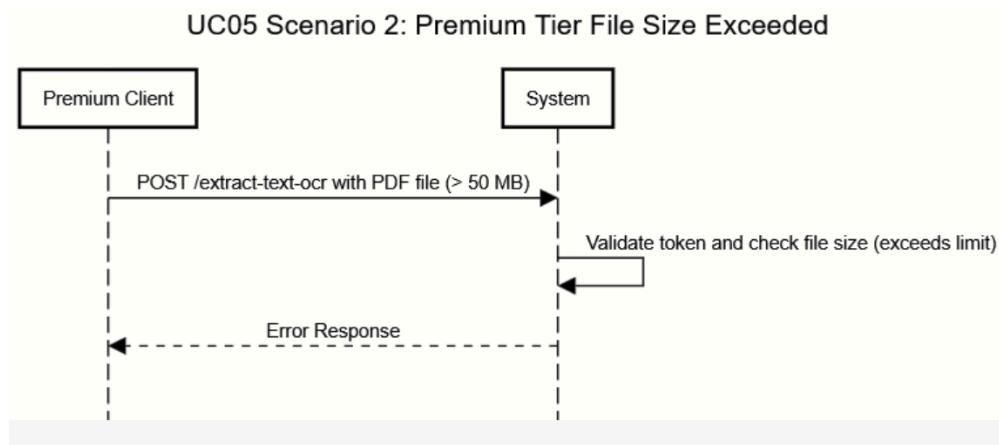


Image 12. UC05 Scenario 2 Sequence Diagram

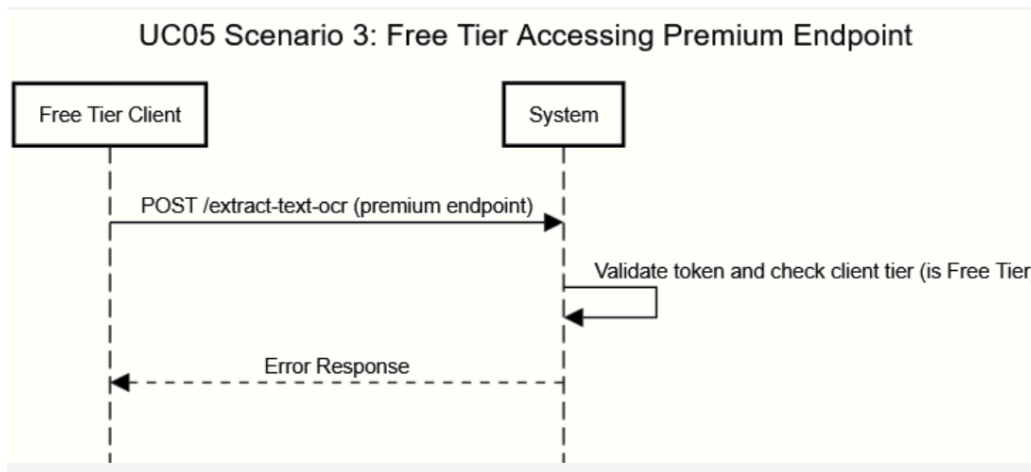


Image 13. UC05 Scenario 3 Sequence Diagram

UC06 Diagrams

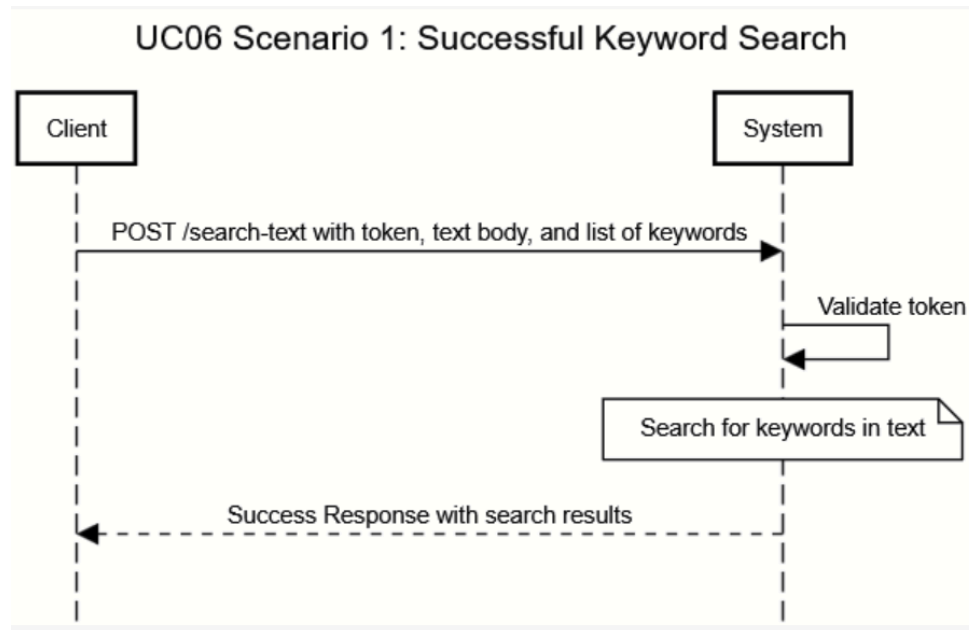


Image 14. UC06 Scenario 1 Sequence Diagram

UC07 Diagrams

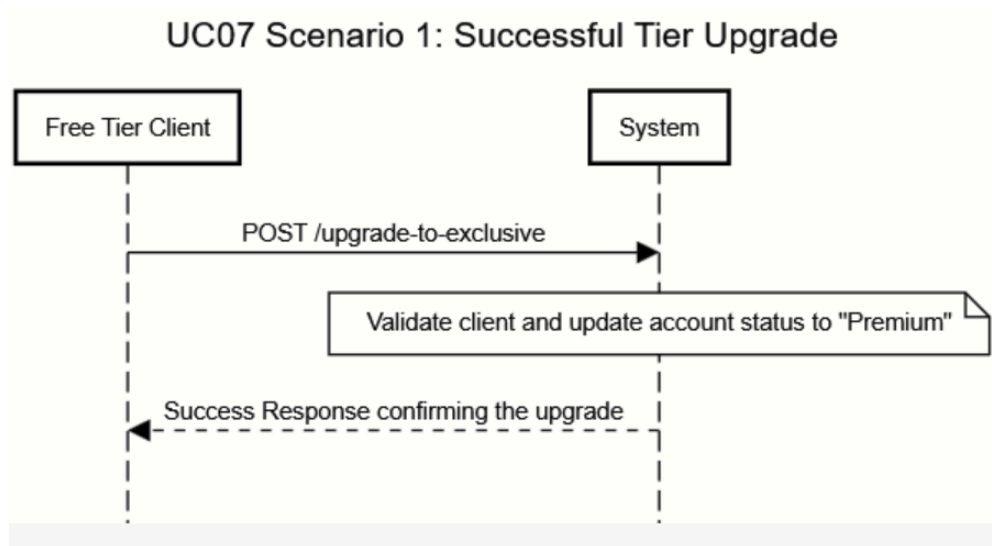


Image 15. UC07 Scenario 1 Sequence Diagram

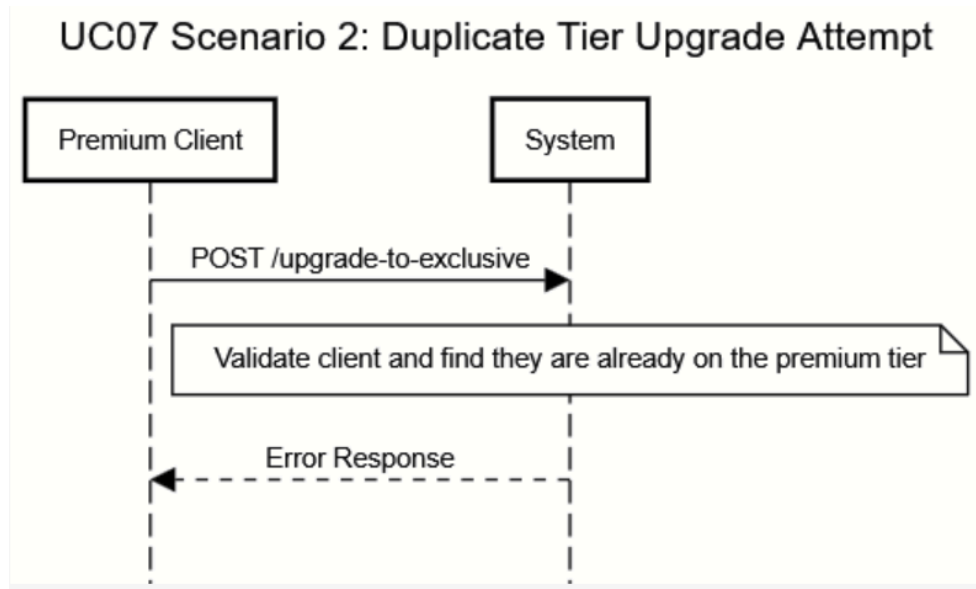


Image 16. UC07 Scenario 2 Sequence Diagram

Entity Relationship Diagram

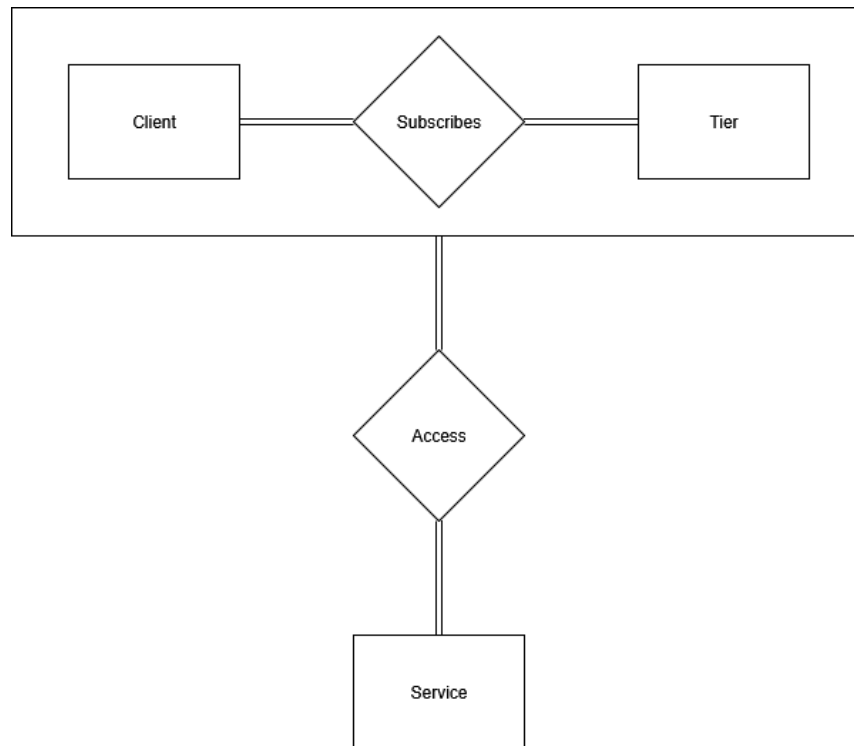


Image 17. ER Diagram between Client and Service

A new client registration subscribes the client with a free tier, and a service's access is determined by the client's credentials via the OAuth2 token, and the client's tier for exclusive endpoint access. Many clients subscribe to a tier and many tiers are subscribed to a client, which represents the many to many relationship. Client and Tier are aggregated to represent a single entity because a service requires the client's credentials and tier. Many "Client that subscribes to a Tier" accesses to a service and many services are accessed by a "Client that subscribes to a Tier".

A Client entity contains attributes of client_id and client_secret. A Tier entity contains attributes of tier level, which are Free and Premium. A Service entity contains attributes of service_name.

Application Architecture Diagram

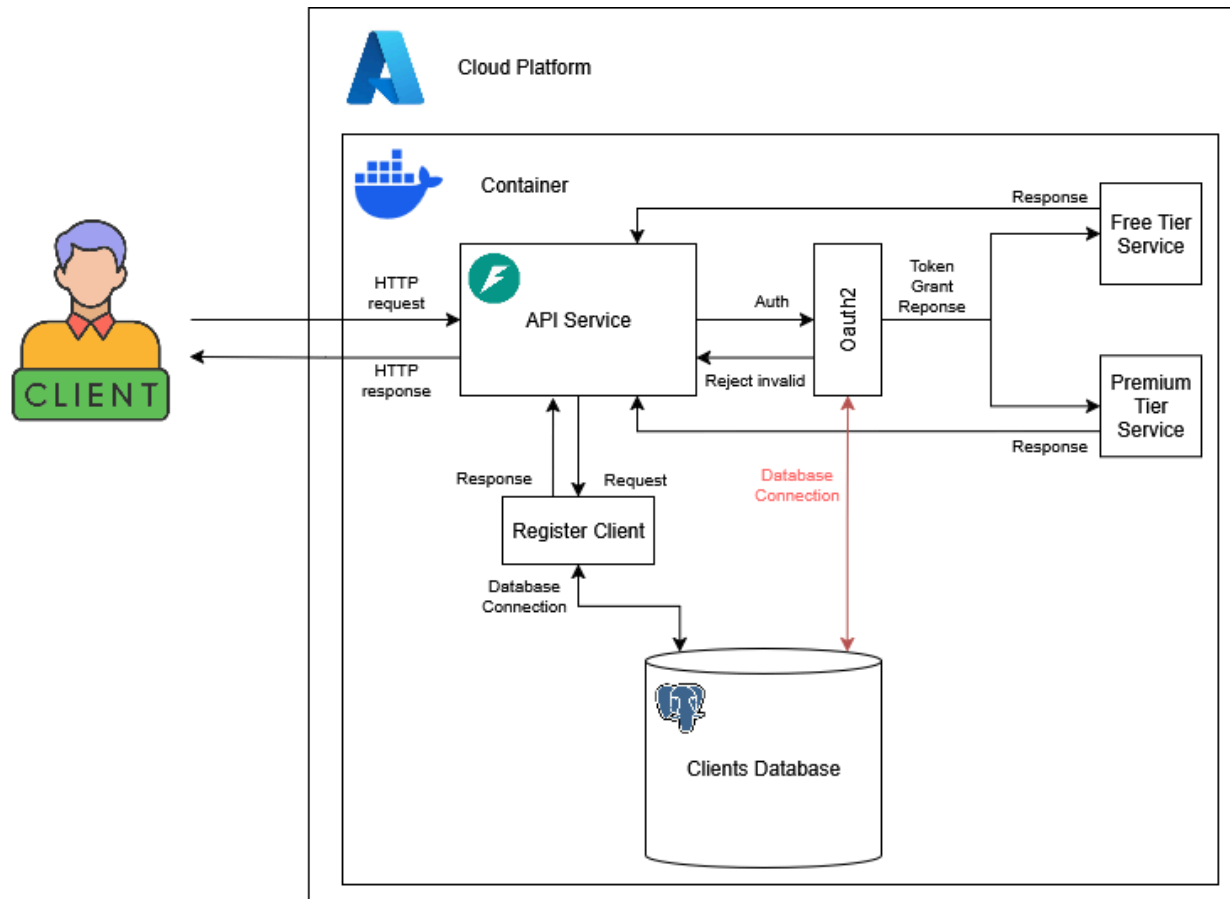


Image 18. Application Architecture Diagram

The client performs an HTTP request to the FastAPI service. The FastAPI service is run inside a cloud platform, Azure, and in a contained environment using docker, which redirects port 80 into port 8000 of the machine. The port 8000 is exposed to the public for developers to integrate with their application. The API service provides a register client service, which performs CRUD operations by connecting to the local postgresql database inside the container's volume. Other services can only be accessed with a token granted by the oauth2 scheme. The oauth2 token granting service is connected to the database to validate client id and its secret, ensuring security between each client. If it's invalid, it simply gets rejected and returns an HTTP error response. If valid, the access to the services is granted and returns responses according to its logic. For more information about the source code, check the following github repository: [link!](#)