

# Functional Neighborhoods

## User Authentication

- As a user, I want to sign up for an account so that I can access the application.
- As a user, I want to reset my password so that I can regain access to my account.
- As a user, I want to verify my email address so that I can secure my account.
- As a user, I want to log in to my account so that I can use the application.
- As a user, I want to log out of my account so that I can secure my session.
- As a user, I want to use multi-factor authentication so that I can enhance the security of my account.
- As a user, I want to use a QR code to log in so that I can easily access my account.

## User Profile Management

- As a user, I want to update my profile information so that it reflects my current details.

## Server Management

- As a user, I want to create a new server so that I can have a space for my community.
- As a user, I want to join an exPropose an architecture diagramisting server so that I can participate in a community.
- As a user, I want to leave a server so that I can manage my participation.

## Channel Management

- As a server admin, I want to create channels so that users can have organized discussions.
- As a server admin, I want to delete channels so that I can manage the server's structure.
- As a user, I want to join channels so that I can participate in discussions.

## Roles and Permissions

- As a server admin, I want to assign roles to users so that I can manage their permissions.
- As a server admin, I want to create custom roles so that I can define specific permissions.
- As a server admin, I want to manage permissions at the channel level so that I can control access to discussions.

## Friends

- As a user, I want to add friends so that I can easily communicate with them.

- As a user, I want to remove friends so that I can manage my contacts.
- As a user, I want to accept friend requests so that I can connect with others.

## Messaging

- As a user, I want to send messages in a channel so that I can communicate with others.
- As a user, I want to receive messages in a channel so that I can read others' communications.
- As a user, I want to send direct messages to another user so that I can have private conversations.

## Notifications

- As a user, I want to receive notifications for mentions so that I don't miss important messages.
- As a user, I want to receive notifications for new messages in channels I follow.

## Voice and Video Communication

- As a user, I want to join voice channels so that I can communicate with others in real-time.
- As a user, I want to start video calls so that I can have face-to-face conversations.
- As a user, I want to share my screen during video calls so that I can collaborate with others.
- As a user, I want to leave voice channels and video calls so that I can end my participation.

## Text Channel Integration

- As a user, I want to be able to create a webhook so that I can integrate text channels with external services.
- As a user, I want to send embeds in text channels so that I can share rich content.

## Internationalization

- As a user, I want to use the application in multiple languages so that I can understand the content better.

-----

Here's a structured way to categorize them into functional neighborhoods:

## User Management

## **User Authentication:**

- Sign up for an account
- Reset password
- Verify email address
- Log in to account
- Log out of account
- Use multi-factor authentication
- Use QR code to log in

## **User Profile Management**

- Update profile information

## **Friends Management**

- Add friends
- Remove friends
- Accept friend requests

## **Community and Server Management**

### **Server Management**

- Create a new server
- Join an existing server
- Leave a server

### **Channel Management**

- Create channels (admin)
- Delete channels (admin)
- Join channels (user)

## **Communication**

### **Messaging**

- Send messages in a channel

- Receive messages in a channel
- Send direct messages to another user

## **Voice and Video Communication**

- Join voice channels
- Start video calls
- Share screen during video calls
- Leave voice channels and video calls

## **Notifications**

### **Notification Management**

- Receive notifications for mentions
- Receive notifications for new messages in followed channels

## **Roles and Permissions**

### **Roles and Permissions Management**

- Assign roles to users (admin)
- Create custom roles (admin)
- Manage permissions at the channel level (admin)

## **Integration and Customization**

### **Text Channel Integration**

- Create a webhook
- Send embeds in text channels

### **Internationalization**

- Use the application in multiple languages

# Functional Neighborhoods and Microservices

## User Management

### User Authentication Service

- Sign up for an account
- Reset password
- Verify email address
- Log in to account
- Log out of account
- Use multi-factor authentication
- Use QR code to log in

### User Profile Service

- Update profile information

### Friends Management Service

- Add friends
- Remove friends
- Accept friend requests

## Community and Server Management

### Server Management Service

- Create a new server
- Join an existing server
- Leave a server

### Channel Management Service

- Create channels (admin)
- Delete channels (admin)

- Join channels (user)

# Communication

## Messaging Service

- Send messages in a channel
- Receive messages in a channel
- Send direct messages to another user

## Voice and Video Service

- Join voice channels
- Start video calls
- Share screen during video calls
- Leave voice channels and video calls

# Notifications

## Notification Service

- Receive notifications for mentions
- Receive notifications for new messages in followed channels

# Roles and Permissions

## Roles and Permissions Service

- Assign roles to users (admin)
- Create custom roles (admin)
- Manage permissions at the channel level (admin)

# Integration and Customization

## Text Channel Integration Service

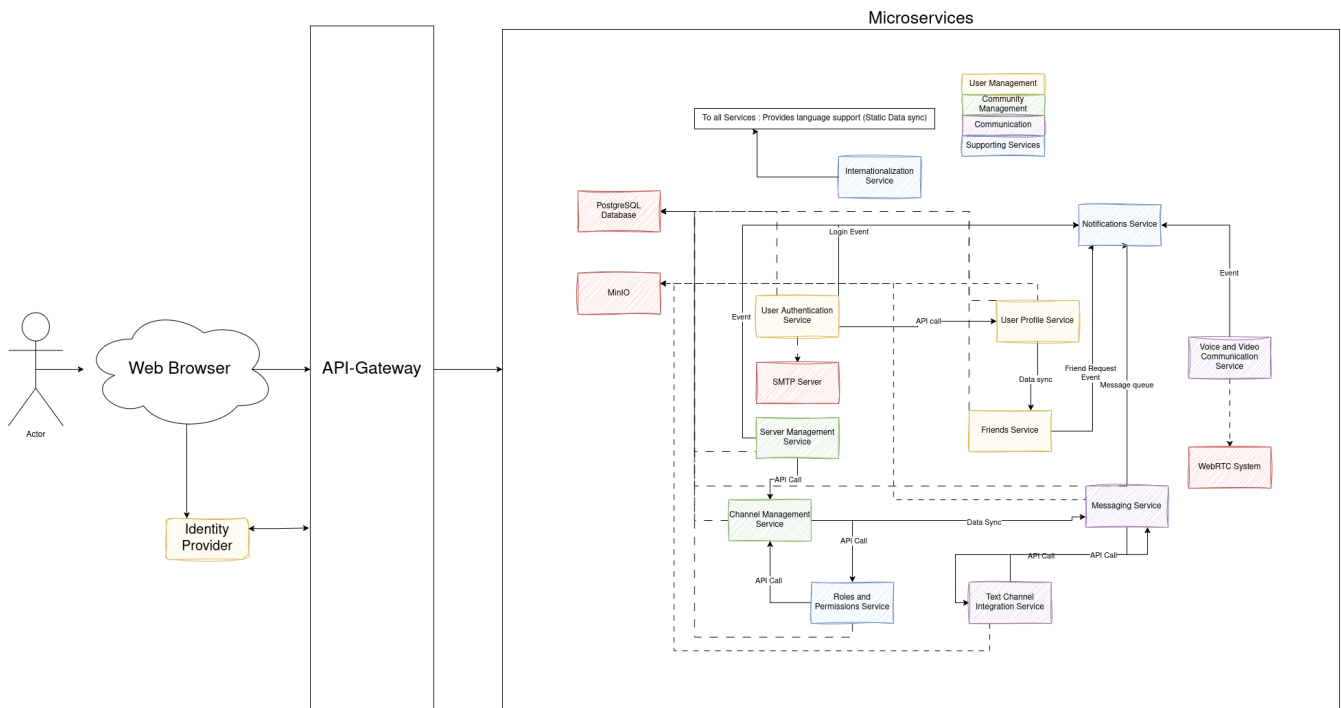
- Create a webhook
- Send embeds in text channels

# Internationalization Service

- Use the application in multiple languages

## Proposed Architecture Diagram

The architecture diagram represents these microservices as distinct components, each responsible for a specific set of functionalities. Each service communicates with others through well-defined APIs, ensuring loose coupling and high cohesion. This architecture promotes scalability, maintainability, and independent deployment of each service.



## Key Considerations

- **Single Responsibility Principle**: Each microservice should focus on a single responsibility, ensuring that it does one thing well.
- **Cohesion**: Related functionalities should be grouped together within the same microservice.
- **Loosely Coupled**: Minimize dependencies between microservices to reduce the impact of changes.
- **Scalability**: Ensure that each microservice can scale independently based on demand.
- **Resilience**: Design the system to handle failures in individual microservices without affecting the entire application.