Ecco una possibile struttura per l'app **WhatsPup**, organizzata per livelli e componenti principali. Questa ossatura include una suddivisione tra frontend, backend e database, integrando funzionalità avanzate e tecnologie menzionate.

Frontend (Angular)

1. **Struttura Componenti**:
   - `AppComponent` : entry point dell'app.
   - `AuthComponent` : login e registrazione (OAuth).
   - `ChatComponent` : gestione delle chat individuali e di gruppo.
   - `GroupComponent` : creazione e gestione dei gruppi.
   - `ProfileComponent` : gestione del profilo utente (incluso premium).
   - `NotificationComponent` : notifiche in tempo reale.
2. **Moduli Angular**:
   - **RouterModule**: gestione delle rotte (es. `/login` , `/chat` , `/profile`).
   - **HttpClientModule**: comunicazione con il backend per API REST.
   - **ReactiveFormsModule**: gestione dei form (es. login e registrazione).
   - **Angular Material**: design moderno e reattivo.
3. **Librerie aggiuntive**:
   - **Socket.IO-client**: per il tempo reale.
   - **ngx-translate**: traduzione delle chat in tempo reale.
   - **Stripe.js**: gestione pagamenti.

Backend (Spring Boot)

1. **Struttura Controller**:
   - **AuthController**: autenticazione e registrazione (OAuth con JWT).
   - **ChatController**: API per invio/ricezione messaggi, gestione della cronologia.
   - **GroupController**: API per creare e gestire gruppi.
   - **NotificationController**: gestione notifiche push tramite Firebase.

- **UserController**: gestione profilo utente (incluse opzioni premium).

2. **Servizi**:
   - **AuthService**: autenticazione sicura con gestione token JWT.
   - **ChatService**: logica per memorizzare e recuperare messaggi dal database.
   - **NotificationService**: invio notifiche in tempo reale.
   - **PaymentService**: integrazione con Stripe per gli abbonamenti.
3. **Librerie aggiuntive**:
   - **Spring Security**: gestione sicura dell'autenticazione.
   - **Socket.IO**: comunicazione in tempo reale.
   - **Spring Validation**: validazione dati in ingresso.

Database (PostgreSQL)

1. **Tabelle principali**:
   - `users` : dati utente (id, nome, email, ruolo, stato premium, token).
   - `messages` : cronologia messaggi (id, mittente, destinatario, testo, timestamp).
   - `groups` : informazioni sui gruppi (id, nome, membri).
   - `subscriptions` : gestione degli abbonamenti premium (utente_id, tipo, data_inizio, data_fine).
   - `notifications` : gestione notifiche (id, utente_id, tipo, messaggio, stato).
2. **Relazioni**:
   - **Users-Groups**: relazione molti-a-molti.
   - **Messages-Users**: relazione uno-a-molti (mittente e destinatario).

Comunicazione in Tempo Reale

1. **Socket.IO (Backend)**:
   - Integrazione con Spring Boot per notifiche in tempo reale e aggiornamento chat.
2. **Socket.IO-client (Frontend)**:
   - Notifica nuovi messaggi o modifiche nei gruppi in tempo reale.

Funzionalità Aggiuntive

1. **Media Management**:
    - Libreria **Sharp** per compressione e ottimizzazione file multimediali.
2. **Traduzione Messaggi**:
    - Integrazione con **Google Cloud Translation API**.
3. **Tema Dinamico**:
    - Angular Material per supportare tema scuro/chiaro.

Questa struttura offre un solido punto di partenza per implementare **WhatsPup** in modo modulare ed estensibile. Fammi sapere se vuoi che sviluppi una parte in dettaglio!