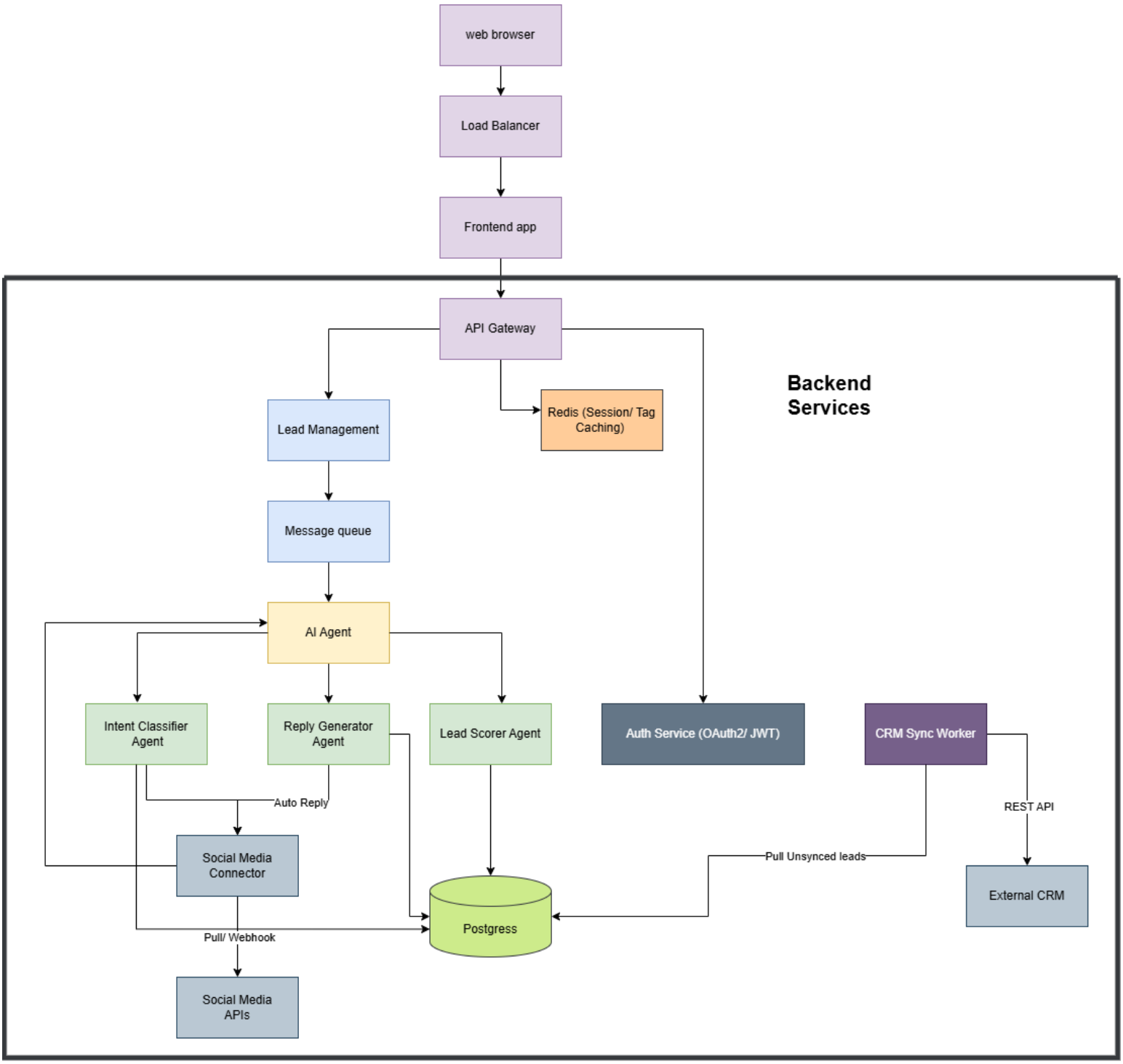
System Architecture for AI-Powered Social Media Lead Management SaaS

The platform enables companies to connect their social media pages, receive AI-generated replies to incoming messages, automatically tag leads based on intent/engagement, and sync all interactions with an external CRM

System Architecture:



Explanation:

This will be a multi-agent system. There will be a main agent that controls 3 agent such as intent classifier agent, reply generator agent and lead scorer agent. There will be webhook setup, to trigger request to AI agent, in case any of the social media doesn't provide webshook functionality a time based system will be set up to check recent social media activities.

The main workflows of the three agents is written below:

1. **Intent Classifier Agent**
    - **Input**: Raw message text from social media.
    - **Output**: Intent (Sales/Support/Spam) + confidence score.
    - **Model**: Fine-tuned LLM (e.g., GPT-3.5) or a smaller distilled model for speed/cost.
    - **Prompt strategy**: Zero-shot with few-shot examples embedded; output forced to JSON schema.
2. **Reply Generator Agent**
    - **Input**: Original message, detected intent, company tone guidelines.
    - **Output**: Natural language reply text.
    - **Model**: LLM with guardrails to avoid hallucinations. Uses **retrieval-augmented generation (RAG)** if FAQ/knowledge base is available.
3. **Lead Scorer Agent**
    - **Input**: Message metadata, user profile, interaction history.
    - **Output**: Lead score (hot/warm/cold) and reason.
    - **Model**: Rule-based + ML classifier (e.g., XGBoost) or LLM with chain-of-thought prompting.
    - **Tagging**: Automatically writes lead_tier and tags to the database.

All agents are stateless microservices orchestrated by the **AI Orchestration Service**, which manages retries, fallbacks, and logging.

Frontend-Backend-AI Data Flow:

1. User logs in via the Frontend → API Gateway → Auth Service → JWT issued.
2. Company admin connects social pages via Social Connector (OAuth2 flow). Credentials encrypted and stored.
3. Incoming messages are received via webhooks (if available) or polling. Each message is published to a Message Queue.
4. AI Orchestration Service consumes messages, invokes the three agents in parallel (where possible):
    - Classifier runs → stores intent.
    - ReplyGen runs → generates reply → sends back via Social Connector.
    - Scorer runs → updates lead score in DB.
5. Lead data (message, intent, reply, score) is persisted in PostgreSQL.
6. CRM Sync Worker periodically fetches unsynced leads and pushes them to the external CRM API. Success/failure logged.

Authentication and Authorization:

- **User Authentication**: OAuth2 / OpenID Connect. Platform uses Auth0 or a self-hosted Keycloak. JWT tokens passed in Authorization header.
- **Social Media Auth**: Each platform's OAuth2 flow is handled by the Social Connector; refresh tokens are rotated and encrypted at rest.
- **Role-Based Access Control (RBAC)**:
    - Admin: manage company settings, view all leads.
    - Agent: view assigned conversations, override auto-replies.
    - Viewer: read-only access.
- **API Gateway** validates JWT and enforces rate limits per tenant.

Data Security and Privacy:

- **Encryption at Rest**: AES-256 for database fields containing PII (names, emails, social handles).
- **Encryption in Transit**: TLS 1.3 for all internal and external communications.
- **Privacy Compliance**:
    - Anonymization of personal data before sending to LLM (e.g., replace names with placeholders).
    - Data retention policies: auto-delete raw messages older than 90 days (configurable).
    - Audit logs for all AI actions and CRM sync operations.
- **Isolation**: Multi-tenant database with row-level security; each tenant sees only their data.

Cost Optimization Strategies:

1. **Caching**: Use Redis to store recent LLM responses for identical queries (hash of message + tenant tone). Reduces API calls by ~30%.
2. **Model Selection**:
    - Classifier: use a smaller, cheaper model (e.g., gpt-3.5-turbo or fine-tuned DistilBERT).
    - ReplyGen: only invoke for non-spam, high-confidence intents; use streaming responses.
3. **Asynchronous Processing**: Message queue prevents idle waiting; workers scale based on queue length.
4. **Batched CRM Sync**: Push lead updates in batches (every 5 min) instead of real-time to reduce API costs.
5. **LLM Token Optimization**: Prompt engineering to use fewer tokens; dynamic truncation of long conversations.