High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components of **PropSecure - Real Estate Property Security Platform**. PropSecure is a secure, scalable backend system for managing real estate property listings, user roles, and access logs, with a modern Angular frontend for agents and admins. The platform emphasizes security, auditability, and efficient background processing.

1. System Architecture Overview

Architecture Description:
PropSecure follows a modular, layered architecture with a RESTful Spring Boot backend, Angular frontend, MySQL database, and Dockerized deployment. Security is enforced via Spring Security with JWT. AOP is used for logging, and background checks are processed asynchronously.

Module/Component	Role/Responsibility
Angular Frontend	UI for agents/admins; interacts with backend via REST
API Gateway (Spring)	Exposes REST endpoints; handles authentication/authorization
Business Logic Layer	Manages property, user, and access log operations
Security Layer	JWT-based authentication, role-based access control
AOP Logging	Logs access and actions for auditability
Async Processing	Handles background checks asynchronously
Persistence Layer	Manages data storage/retrieval via MySQL
Docker Environment	Containerizes all services for deployment

2. Component Interactions

Interaction Step	Description
1. User Login	Frontend sends credentials; backend issues JWT on success
2. Authenticated Requests	Frontend includes JWT; backend validates and authorizes
3. Property Management	CRUD operations via REST; business logic applies rules
4. Access Logging (AOP)	All sensitive actions logged via AOP
5. Background Checks	Triggered by property/user events; processed asynchronously
6. Data Persistence	All state changes stored/retrieved from MySQL

3. Data Flow Overview

Data Flow	Source	Destination	Notes
User Auth (JWT)	Frontend	Backend	JWT issued, validated on each request
Property/User CRUD	Frontend	Backend/MySQL	REST API, business logic, persistence
Access Logs	Backend (AOP)	MySQL	Logged automatically
Background Check Requests/Status	Backend	Async Worker	Results stored in MySQL

4. Technology Stack

Layer/Function	Technology/Framework
Frontend	Angular
Backend/API	Java, Spring Boot
Security	Spring Security, JWT
Logging	Spring AOP
Async Processing	Spring Async
Database	MySQL
Build/Dependency	Maven
Containerization	Docker

5. Scalability, Reliability & Security

- **Scalability:**
  - Stateless backend enables horizontal scaling via Docker containers.
  - Async processing decouples long-running tasks.
- **Reliability:**
  - Centralized logging and audit trails via AOP.
  - Persistent storage in MySQL with backup strategies.
- **Security:**
  - JWT-based authentication and role-based access control.
  - All sensitive actions logged for traceability.
  - Open-source stack with regular updates.

End of Document