1. Role Service (**role.service.js**):
   * CRUD operations for roles
   * Permission management
   * Role hierarchy validation
   * Caching with Redis
   * Event publishing for role changes
   * Role-user relationship management
2. User Service (**user.service.js**):
   * Role assignment/removal for users
   * Role-based queries
   * User-role relationship management
   * Role validation
   * Event handling for role changes
3. Models:
   * **Role** model with permissions and hierarchy
   * **UserRole** junction model for many-to-many relationship
   * Proper associations between User and Role
4. Features:
   * Role-based access control
   * Permission management
   * Role hierarchy
   * Caching
   * Event-driven updates
   * Data validation
   * Error handling

The role-management-service appears to be redundant since all role management functionality is already well-implemented in the user service. It would be safe to delete the role-management-service to:

1. Avoid duplication of functionality
2. Prevent potential data inconsistencies
3. Simplify the architecture
4. Reduce maintenance overhead
5. Maintain a single source of truth for role management