Phân Tích Yêu Cầu Hệ Thống Phân Quyền: RBAC và Data Access Control

1. Tên Hệ Thống

Hệ thống Quản lý Nguồn Lực Doanh Nghiệp (ERP)

2. Mô Tả Chung

Hệ thống ERP cần triển khai cơ chế phân quyền toàn diện, kết hợp giữa **phân quyền theo tính năng (Feature-based Access Control)** và **phân quyền theo dữ liệu (Data Access Control)**. Mục tiêu là đảm bảo người dùng chỉ truy cập được các giao diện, chức năng và dữ liệu mà họ được phép, dựa trên vai trò và phạm vi được gán, từ đó tăng cường bảo mật, tính linh hoạt và khả năng quản lý.

3. Các Khái Niệm Chính

- **Người Dùng (User)**: Thực thể tương tác với hệ thống, được gán một hoặc nhiều vai trò.
- **Vai Trò (Role)**: Tập hợp quyền hạn, ví dụ: "Admin", "Quản lý Nhân sự", "Nhân viên".
- **Tính Năng/Giao Diện (Feature/UIId)**: Định danh duy nhất cho mỗi giao diện hoặc module, ví dụ: `UI_XEM_BAO_CAO_LUONG`, `UI_NHAP_NHAN_VIEN`.
- **Hành Động (Action)**: Thao tác người dùng có thể thực hiện, ví dụ: "Xem", "Thêm", "Sửa", "Xóa", "Duyệt".

4. Yêu Cầu Phân Quyền

Hệ thống yêu cầu hai loại phân quyền hoạt động song song:

A. Phân Quyền Theo Tính Năng (Feature-based Access Control)

- **Mô tả**: Kiểm soát quyền truy cập vào giao diện/chức năng cụ thể dựa trên **RBAC**.
- **Cơ chế**: Dựa vào vai trò của người dùng và `UIId` để kiểm tra quyền tại tầng API.
- **Ví dụ**: "Nhân viên" được truy cập `UI_XEM_THONG_TIN_CA_NHAN` nhưng không được truy cập `UI_NHAP_DANH_SACH_NHAN_VIEN`.

- **Điểm kiểm tra quyền**: Tại middleware API, sử dụng `UIId` từ Frontend để kiểm tra quyền chức năng.

B. Phân Quyền Theo Dữ Liệu (Data Access Control)

B1. Phân Quyền Theo Cấu Trúc Cây (Hierarchical Data Access Control)

- **Mô tả**: Kiểm soát truy cập dữ liệu theo cấu trúc cây, ví dụ: cây cơ cấu tổ chức hoặc cây thư mục tài liệu.
- **Cơ chế**: Sử dụng thuộc tính `LevelPath` (ví dụ: "1.2", "1.2.4") để xác định phạm vi truy cập.
- **Ví dụ**: "Trưởng phòng A" chỉ xem nhân viên có `LevelPath` bắt đầu bằng "2.%".
- **Điểm kiểm tra quyền**: Tại tầng dữ liệu, thêm điều kiện `WHERE LevelPath LIKE 'user_org_path.%'` vào truy vấn SQL.

B2. Phân Quyền Theo Phạm Vi Đối Tượng (Data Access Domain/ScopeData)

- **Mô tả**: Kiểm soát quyền trên từng đối tượng dữ liệu cụ thể, ví dụ: hồ sơ, hóa đơn, đơn hàng.
- **Cơ chế**: Lưu mối quan hệ giữa người dùng/vai trò và đối tượng (Owner, Editor, Approver) với `DepId` hoặc `ScopeData`.
- **Ví dụ**: Người tạo hồ sơ có quyền sửa, người duyệt chỉ có quyền xem/duyệt.
- **Điểm kiểm tra quyền**: Tại tầng logic nghiệp vụ, kiểm tra hành động và mối quan hệ với đối tượng.

5. Luồng Kiểm Tra Quyền

1. **Xác thực**: Xác định danh tính người dùng và vai trò.
2. **Kiểm tra quyền tính năng**: Middleware kiểm tra `UIId` và vai trò, trả về lỗi 403 nếu không được phép.
3. **Kiểm tra quyền dữ liệu (cây)**: Thêm điều kiện `LevelPath` vào truy vấn dữ liệu.
4. **Kiểm tra quyền dữ liệu (đối tượng)**: Kiểm tra hành động và mối quan hệ với đối tượng cụ thể.
5. **Thực hiện thao tác**: Nếu tất cả kiểm tra thành công, thực hiện thao tác hoặc trả dữ liệu.

6. Ưu Điểm

- **Bảo mật cao**: Kiểm soát chặt chẽ cả chức năng và dữ liệu.
- **Linh hoạt**: Hỗ trợ nhiều kịch bản phân quyền.

- **Rõ ràng**: Phân tách quyền tính năng và dữ liệu, dễ quản lý.
- **Tái sử dụng**: Cơ chế kiểm tra quyền áp dụng cho nhiều API và dữ liệu.

7. Hạn Chế/Thách Thức

- **Phức tạp**: Triển khai yêu cầu thiết kế và tối ưu cẩn thận.
- **Hiệu suất**: Điều kiện `LevelPath` và kiểm tra đối tượng có thể ảnh hưởng hiệu suất.
- **Quản lý quyền**: Cần giao diện quản trị mạnh mẽ để quản lý quy tắc phức tạp.
- **Đồng bộ dữ liệu**: Đảm bảo nhất quán giữa dữ liệu phân quyền và cơ sở dữ liệu.

8. Các Bước Tiếp Theo

- **Thiết kế Schema Database**: Tạo bảng lưu trữ `Roles`, `Features/UIIds`, `Role-Feature-Action`, `OrgUnits` (`LevelPath`), và `DocumentAccessControl` (Owner, Editor, Viewer).
- **Công cụ/Thư viện**:
- Sử dụng **Casbin** cho RBAC và ABAC