# Optimizing User, Group, and Role Management with Access Control and Workflows in service now

## SOLUTION ARCHITECTURE :

| | |
|---|---|
| Date | 02:11:2025 |
| Team ID | NM2025TMID07577 |
| Project name | Optimizing User, Group, and Role Management with Access Control and Workflows in service now |
| Maximum marks | 4 marks |

The Solution Architecture phase translates the proposed solutions into a high-level, technical design blueprint, specifying the ServiceNow components, integrations, and data flows necessary to optimize User, Group, and Role Management (UGRM).

## □ Solution Architecture Blueprint for Optimized UGRM

The architecture focuses on three integrated layers: Data Source (Integration), Processing (Core Platform), and Delivery (Service Portal/Flows).

## 1. Data Source and Synchronization Layer (Integration)

This layer is responsible for ensuring a single, authoritative source of user data in ServiceNow.

| Component | Purpose & Configuration | Architectural Impact |
|---|---|---|
| Identity Provider (IdP) | The single source of truth for User Records (names, status, department, manager). | Automates user creation and deactivation. Uses Scheduled Data Imports via LDAP/SAML/SCIM. |
| HR System (HRIS) | The authoritative source for User Lifecycle Status (Hire, Transfer, Termination). | Triggers the provisioning/deprovisioning workflows. Integration via Integration Hub (Spoke) or direct SOAP/REST API. |
| Custom Transform Maps | Custom scripting on the `sys_import_set` to enforce data cleanup (e.g., standardizing department names) before writing to `sys_user`. | Ensures data integrity and prevents manual data fixes. |

## 2. Processing and Governance Layer (Core Platform)

This is the control center where group, role, and access logic are enforced.

| Component | Purpose & Configuration | Architectural Impact |
|---|---|---|
| User & Group Tables (`sys_user`, `sys_user_group`) | Group Classification Field: Add a mandatory field (e.g., `u_group_type`) to clearly designate the purpose (Security, Assignment, Reporting). | Enforces Group Standardization and aids in reporting/auditing. |
| Role Hierarchy | Strict definition of role inheritance, minimizing direct role assignments. Roles assigned only to groups. | Enforces Least Privilege Principle and simplifies role maintenance. |

| | | |
|---|---|---|
| Access Controls (ACLs) | Refactor ACLs to utilize a Role-based Security Model where possible. Complex ACLs should be isolated and heavily documented. | Improves Performance (faster ACL evaluation) and Security Clarity. |
| Access Certification | Scheduled jobs to run quarterly attestation campaigns for high-risk groups/roles. | Automates Compliance and reduces Access Sprawl. |

## 3. Delivery and Automation Layer (Workflows)

This layer manages the end-user experience and automated fulfillment of access requests.

- Centralized Service Catalog Item (SCI):
  - One SCI is built on the Request Item (`sc_req_item`) table for all access requests.
  - It presents Catalog Variables that map directly to the standardized roles/groups (or abstract Service Offerings).
- Access Provisioning Workflow (Flow Designer/Workflow):
  - Logic: The flow dynamically determines the required Approval chain (Manager, Group Owner, Security) based on the risk level of the requested access.
  - Automation: Upon final approval, the flow executes an Action/Script to update the `sys_user_grmember` table (or `sys_user_has_role` for exceptions) and closes the request.
- Virtual Agent/Service Portal Integration:
  - Provides a self-service front-end for request submission and real-time status tracking, eliminating reliance on email/chat follow-ups.

This architecture centralizes governance, leverages platform automation capabilities, and establishes clear separation between the authoritative data sources and the processing logic.