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

Data Flow Diagrams and User Stories:

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

This combined phase translates the Solution Architecture into visual data movement and user-centric requirements.

Data Flow Diagram (DFD) for Optimized UGRM

The DFD illustrates how data related to users, groups, and access flows through the ServiceNow platform, ensuring synchronization and automated governance.

External Systems and Data Flow

1. HRIS (Source of Truth for Status): Sends User Lifecycle Status changes (Hire, Termination, Transfer) to ServiceNow.
2. IdP (Source of Truth for Attributes): Sends core User Attributes (Name, Email, Manager, Department) to ServiceNow.

ServiceNow Data Processing Flow

Step	Component/Table	Data Flow Action	Architectural Note
1. User Sync	$\rightarrow$ Import Set Table $\rightarrow$ Transform Map $\rightarrow$ `sys_user`	Creates or updates User Records based on HRIS/IdP data.	Automation: Scheduled Jobs handle recurring syncs.
2. Access Request	Service Catalog $\rightarrow$ `sc_req_item` $\rightarrow$ Flow Designer	User requests a Role or Group membership.	Centralization: Single entry point for all access requests.
3. Approval Routing	Flow Designer $\rightarrow$ Approvals (`sysapproval_approver`)	Dynamic logic routes the request based on the Risk Level of the requested entitlement.	Governance: Enforces required approvals by Manager/Owner/Security.
4. Fulfillment	Flow Designer $\rightarrow$ Script/Action $\rightarrow$ `sys_user_grmember` & `sys_user_has_role`	If approved, the system automatically writes the new membership/role.	Automation: Eliminates manual fulfillment by the UGRM Admin.
5. Access Review	Access Certification $\rightarrow$ Group Owners	System initiates a periodic review of high-risk group memberships.	Compliance: Reduces Access Sprawl by enforcing attestation.

6. ACL Evaluation	`sys_user`, `sys_user_group`, `sys_user_role` $\rightarrow$ ACL Engine	Every record access triggers a check against the user's entitlements.	Performance: Optimized ACLs ensure quick lookup times.

User Stories Phase

User stories capture the required functionality from the perspective of the different stakeholders, focusing on the desired outcome of the optimization.

Persona 1: End-User/Requester

As a...	I need to...	So that I can...
New Employee	request all necessary application access through one centralized catalog item	start working on my assigned tasks immediately after my start date.
Requester	view the real-time approval status and current approver of my access request	avoid constantly emailing the UGRM team for updates.
Transferring Employee	have my old, unused roles automatically revoked and my new baseline roles automatically assigned upon transfer	ensure my access follows the principle of least privilege in my new department.

Persona 2: UGRM Administrator

As a...	I need to...	So that I can...
UGRM Admin	automate user deactivation and role revocation when the HRIS status is set to "Terminated"	eliminate manual offboarding steps and reduce security risk immediately.
UGRM Admin	enforce a Group Classification standard (Security, Assignment, Reporting) during new group creation	simplify governance and ensure roles are only assigned to security-classified groups.
UGRM Admin	utilize a clear, documented Role Hierarchy	easily manage role relationships and avoid creating redundant custom roles.

Persona 3: Security/System Owner

As a…	I need to…	So that I can…
Security Officer	receive an automated notification and log an incident whenever a user is directly assigned a high-risk role (e.g., `admin`)	immediately investigate and document the exception to the governance policy.
System Owner	trigger quarterly access certification campaigns for all groups that contain production data access roles	satisfy audit requirements and prevent stale access.
System Owner	generate a report of all ACLs that use complex scripting conditions	simplify future maintenance and identify potential performance bottlenecks.