WHITE PAPER 03 / ROLES AND POLICY

# Designing Roles and Segregation of Duties Around Real Work

Use role mining to propose access patterns, then validate business purpose and conflicting permissions before assignment.

AIdentX Editorial | September 2026

## Executive summary

Roles make access easier to manage when they represent work that people genuinely perform. They become a problem when they preserve historical exceptions or combine responsibilities that should remain separate. This paper explains how to develop a role model from observed access, test it against business needs and govern segregation-of-duties conflicts without treating every exception in the same way.

### What you will take away

- Treat existing access as evidence to investigate, not a design to copy.
- Validate role contents against business tasks and policy.
- Give conflicts an accountable decision and a time-bounded exception path.

Written for business process owners, role designers and policy teams.

The worked scenarios are fictional. The checklists and examples are practical editorial guidance, not customer results or audited implementations.

# 1. Describe the business task first

A role is a manageable bundle of access associated with a job or responsibility. An entitlement is a specific permission or membership. A useful role model lets a business owner explain why its permissions belong together. A job title alone may be too broad, especially across countries, applications or project assignments.

> **Illustrative workplace case**
>
> Pine Harbour Foods is a fictional distributor. Procurement staff create purchase orders, while a separate finance group approves payments. Several employees have accumulated permissions from both teams during temporary assignments. The company wants simpler access roles without preserving those historical overlaps.

Begin with tasks such as maintaining supplier records, raising purchase orders and releasing payments. Ask process owners which tasks belong together and which require separation. Then map those tasks to actual application entitlements. This makes the role design reviewable by people who understand the business process.

Name an owner for each proposed role. That owner should be able to judge membership criteria, approve changes and explain the consequences of its permissions. A role without an owner can become a convenient container for access that nobody wants to review.

# 2. Use role mining to generate candidates

Role mining looks for patterns in existing access. It can identify permissions commonly held by a peer group and suggest candidate bundles. The pattern is a starting point, not proof of what the organisation should authorise.

Clean the input population before interpreting the result. Separate inactive accounts, temporary project permissions and known exceptions where possible. Check whether direct assignments, nested groups and application-local rights are included. Missing access paths can make a role appear simpler than it really is.

| Candidate pattern | Question for the owner |
| --- | --- |
| Common permission across a team | Is it required for the job or simply inherited? |
| Rare permission on one person | Is it a specialist responsibility or an unnecessary exception? |
| Large overlap between two roles | Are the jobs equivalent, or does a sensitive difference matter? |
| One role spanning several teams | Does the bundle have a coherent owner and business purpose? |

At Pine Harbour, a shared payment-release permission might reflect old transfers rather than a legitimate procurement responsibility. Removing that permission from the candidate role can be more important than maximising similarity with the existing population.

## 3. Translate conflicts into precise policy

Segregation of duties, or SoD, separates responsibilities where their combination could allow an inappropriate action without independent oversight. A rule must identify the conflicting capabilities and the scope in which the conflict matters. Labels such as “finance conflict” are too vague for consistent enforcement.

NIST SP 800-53 includes separation of duties in AC-5 and least privilege in AC-6. These concepts support defining responsibilities and limiting permissions according to the work that needs to be performed. NIST SP 800-53 Rev. 5: Access Control family.

For the fictional distributor, one policy concern might involve the ability to change supplier bank details and release a related payment. Confirm the exact application permissions and whether the conflict depends on company code, transaction scope or another boundary. The control design should reflect the real business exposure.

Evaluate effective access, not just the contents of one proposed role. Two individually acceptable roles may create a conflict when assigned together. Existing direct permissions can also combine with a new role. Document how those paths are included in the policy check.

## 4. Make exceptions deliberate and reviewable

Small teams and emergency situations sometimes require an arrangement that differs from the preferred separation. An exception needs an explicit business reason, appropriate approval and a description of how the remaining exposure is handled. Calling it an exception does not make it safe.

| Exception field | Purpose |
| --- | --- |
| Affected person and permissions | Makes the scope specific. |
| Business need and duration | Explains why the deviation is requested and when it ends. |
| Compensating review | Describes who checks the affected work and how. |
| Approver and owner | Assigns authority and ongoing responsibility. |
| Expiry or review condition | Prevents indefinite retention without reconsideration. |

At Pine Harbour, an authorised short absence-cover arrangement might include an independent review of affected transactions. The business owner must decide whether that measure is adequate in the actual process. A generic tick box stating “compensating control present” is insufficient without an operating description.

Keep exceptions out of the standard role unless the role’s intended purpose truly changes. Otherwise, a temporary need can become permanent access for everyone who receives the role later.

## 5. Test before changing the access model

Compare proposed memberships with current effective access. Identify who would gain or lose permissions and ask owners to review consequential differences. Include people with multiple assignments, temporary cover and application-specific exceptions in the test population.

IGAX describes AI-supported role mining, peer-based suggestions and a policy and SoD engine with request-time validation and change-impact analysis. Use these capabilities within an owner-approved role design and policy process, with integration behaviour tested for the connected applications. IGAX platform capabilities.

Test both permission grants and removals. A role change can withdraw access that someone needs for a legitimate task, while a failed removal can leave the intended policy unenforced. Plan a correction path and record the result of the first production changes.

- Role names and descriptions have clear business meaning.
- Membership rules distinguish normal work from exceptions.
- Sensitive combinations are checked across effective access.
- Owners have reviewed the impact of changed assignments.
- The removal path and exception path have been tested.

## 6. Keep the model useful over time

Review roles after a process redesign, acquisition or significant application change. A role that matched last year's organisation may no longer have a clear purpose. Retire unused candidates and consolidate only where the business meaning is genuinely the same.

Use certification decisions and access-request patterns as feedback. Repeated removal of one permission may indicate a role-design problem. Repeated requests for the same legitimate responsibility may justify a new role or an update to an existing one. Investigate the reason before changing the model.

> **Measure fitness, not just role count**
>
> A smaller role catalogue is not automatically better. Track whether roles have owners, understandable purposes, appropriate memberships and manageable exceptions. A slightly larger catalogue may be necessary to preserve meaningful business boundaries.

The result should be a model that owners can explain and operations teams can maintain. Similarity algorithms can reduce the search effort, but accountability for who receives what access remains a business and governance decision.

## References and further reading

Primary guidance used for the points cited in this paper. Examples, templates and recommended working practices are AIdentX editorial guidance. Sources reviewed September 2026.

NIST SP 800-53 Rev. 5: Access Control family

https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

IGAX platform capabilities

https://www.aidentx.com/igax/platform

### Put the guidance into practice

Read related articles and explore IGAX platform capabilities at www.aidentx.com/igax/resources.