# AI Knowledge Hub Security and Permission Design

A practical framework for controlling access, preventing information leakage, validating permissions and governing AI-assisted knowledge retrieval.

**Purpose of this white paper**

This guide is written for executives, IT leaders, compliance teams, operations leaders and implementation partners. It explains how to design security around an AI Knowledge Hub so that the system answers from approved organizational information without exposing restricted material or creating a second, disconnected security model.

| Design Area | Required Outcome |
|---|---|
| Core principle | The AI layer should respect the same identity, group membership and content permissions that govern the source systems. |
| Primary risk | A useful answer can still be a security failure if it reveals information the user was not authorized to access. |
| Operating model | Security must be designed, tested and monitored as an ongoing control system rather than a one-time configuration task. |
| Implementation outcome | A read-only, permission-aware knowledge experience with documented ownership, escalation and review procedures. |

Prepared by Pixeldust

# Executive Summary

An AI Knowledge Hub can make organizational information easier to find, but convenience changes the security surface. Employees no longer need to know where a document is stored or which folder contains an answer. The system retrieves, summarizes and presents information in response to natural-language questions. That improves productivity, but it also means access controls must work correctly at retrieval time, not merely at the folder level.

The strongest design does not create a new standalone permission structure for the AI system. It anchors access to the organization's existing identity platform, source-system permissions, approved groups and content governance. The AI agent should operate as a controlled retrieval layer rather than a universal repository with unrestricted access.

This paper presents a practical security model built around six controls: identity, authorization, source governance, retrieval boundaries, validation and monitoring. It also includes a permission matrix, threat model, test scripts, incident workflow and implementation checklist.

**Security design objective**

A user should receive the most useful answer that can be constructed entirely from information that user is authorized to access - and nothing else.

## What this framework prevents

- Unauthorized answers assembled from restricted documents.
- Permission leakage caused by broad service accounts or inherited access.
- Confidential content entering an index or knowledge source without approval.
- Users receiving outdated, draft or unverified instructions as authoritative answers.
- Cross-department exposure caused by poorly designed groups or duplicate repositories.
- Untraceable changes to sources, permissions or agent configuration.

# Contents

# 1. The Security Model

The security model separates six questions that are often incorrectly treated as one. A system may identify the user correctly but still retrieve from an over-permissioned source. It may retrieve only authorized content but present an outdated draft. It may produce an accurate answer without preserving enough evidence to audit what happened.

| Control Layer | Question | Expected Control |
|---|---|---|
| Identity | Who is asking? | Verified organizational identity, authentication status and tenant context. |
| Authorization | What may this user access? | Role, group membership, direct permissions and conditional restrictions. |
| Source governance | Which content is eligible? | Approved repositories, authoritative status, ownership and review state. |
| Retrieval boundary | What may the agent search? | Configured knowledge sources, connectors, indexes and exclusions. |
| Answer control | What may the system present? | Citations, confidence rules, restricted-topic behavior and escalation. |
| Evidence and monitoring | Can the event be reconstructed? | Logs, test records, change history, incidents and periodic review. |

## The least-privilege rule

The AI system should have no broader effective access than required to answer approved business questions. Broad access may appear easier during implementation, but it turns every indexing, configuration or prompt defect into a potential disclosure event. Least privilege should apply to users, administrators, knowledge sources, service identities and deployment environments.

> **Design test**
>
> If a temporary employee, contractor or volunteer asks the same question as an executive, should the system produce the same answer? If not, the answer path must be explicitly tested for both identities.

# 2. Roles and Responsibility

Security fails when everyone assumes someone else owns it. A small organization does not need a large governance committee, but it does need named accountability for access, source approval, testing and incident response.

| Role | Primary Responsibility | Typical Cadence |
|---|---|---|
| Executive sponsor | Approves risk posture, funding and escalation decisions. | Quarterly |
| System owner | Owns the Knowledge Hub, configuration and operating procedures. | Ongoing |
| Security or IT owner | Owns identity, groups, technical access and incident coordination. | Ongoing |
| Knowledge owner | Approves content, classification, authority and review schedule. | Monthly or quarterly |
| Content steward | Maintains files, metadata, versions and archiving. | Weekly or monthly |
| Tester or reviewer | Runs permission and answer-quality tests using realistic personas. | Before release and after changes |
| End user | Uses the system within policy and reports incorrect or suspicious answers. | Ongoing |

## Minimum separation of duties

- The person who configures access should not be the only person who validates it.
- Knowledge owners approve content authority; system administrators should not make business-policy decisions by default.
- Production changes should be documented and reviewable, even when a single consultant performs the implementation.
- High-risk permission changes should require explicit approval from both the system owner and the responsible knowledge owner.

# 3. Permission Architecture

Permission design should begin with user populations and information domains, not individual files. The most maintainable approach is group-based access aligned to real organizational roles. Direct, one-off user permissions create exceptions that are difficult to test and easy to forget.

| Population | Typical Knowledge | Recommended Access Pattern |
|---|---|---|
| All employees | General policies, approved procedures, directory information, standard FAQs | Broad organizational group |
| Department staff | Department SOPs, schedules, internal references and working guidance | Department security group |
| Managers | Performance processes, management procedures and escalation guidance | Manager group or role group |
| HR | Personnel procedures, restricted templates and employee relations guidance | Restricted HR group |
| Finance | Budgets, banking procedures, payroll processes and financial reporting guidance | Restricted finance group |
| Executives | Board materials, strategic planning and confidential leadership documents | Executive group |
| Contractors or volunteers | Only approved operational information required for assigned work | Dedicated limited-access group |

## Permission inheritance

Inheritance reduces administrative burden, but it can also spread mistakes. A secure design uses inheritance intentionally at the site, library or domain level and breaks inheritance only when a clear business requirement exists. Every broken inheritance point should have an owner and review date.

**Avoid the universal service account**

A service identity with access to every repository may cause the retrieval layer to see more than the asking user. Where delegated user access or permission-aware retrieval is available, prefer it. Where it is not, isolate the knowledge source and ensure it contains only material appropriate for every intended user.

# 4. Content Classification and Source Approval

Not every document that an employee can open should become a knowledge source. AI retrieval increases the reach of content, so source approval must consider sensitivity, authority, currency and intended audience.

| Classification | Meaning | AI Retrieval Rule |
|---|---|---|
| Public | Approved for external release | May be used in internal or external agents if current. |
| Internal | Routine operational information for employees | Suitable for broad internal retrieval. |
| Restricted | Departmental, financial, HR or management information | Use only with permission-aware retrieval and explicit ownership. |
| Highly restricted | Legal privilege, investigations, credentials, protected personal data or similarly sensitive records | Exclude by default; include only with a documented requirement and enhanced controls. |
| Draft or unapproved | Work in progress, proposed policy or informal notes | Exclude until approved or clearly labeled for a limited workflow. |
| Expired or superseded | No longer authoritative | Archive and exclude from active retrieval. |

## Source approval checklist

- Named business owner and content steward.
- Classification assigned and intended audiences documented.
- Authoritative status confirmed; duplicates and superseded versions removed.
- Review date and retention expectation recorded.
- Permissions tested using representative user accounts.
- Content format supports reliable extraction and citation.
- Sensitive fields, secrets and unnecessary personal data removed or isolated.
- Approval recorded before the source is connected to production retrieval.

# 5. Retrieval Boundaries and Answer Controls

Permissions determine what the system may retrieve. Answer controls determine how it should respond. Both are required. A properly permissioned system can still mislead users if it merges conflicting documents, omits warnings or presents draft guidance as final policy.

| Control | Expected Behavior | Security Value |
|---|---|---|
| Citations | Show the source title or link used for the answer. | Allows verification and exposes stale or incorrect sources. |
| Authority ranking | Prefer approved policies and canonical procedures over informal notes. | Reduces conflicting answers. |
| Recency rule | Prefer current approved versions and exclude superseded documents. | Prevents obsolete guidance. |
| Insufficient evidence behavior | State that the answer is not available and route the user to an owner. | Safer than guessing. |
| Restricted-topic behavior | Decline or redirect questions involving inaccessible information. | Prevents inference from restricted sources. |
| Read-only posture | Do not alter source systems unless a separate, approved workflow exists. | Limits operational and security impact. |
| Human escalation | Provide the responsible role or process for high-risk decisions. | Keeps professional judgment with authorized people. |

## Inference risk

An agent may reveal sensitive information without quoting a restricted document. For example, a user might infer an upcoming layoff, acquisition or disciplinary action from summarized fragments. Testers should evaluate not only direct document access but also whether combinations of authorized and unauthorized questions allow the system to disclose restricted conclusions.

# 6. Threat Model and Failure Modes

| Failure Mode | Description | Risk | Primary Mitigation |
|---|---|---|---|
| Over-permissioned source | Repository or service identity has broader access than intended. | High | Restrict source, use groups, test personas. |
| Stale group membership | Departed or transferred user remains in a privileged group. | High | Automated lifecycle process and periodic access review. |
| Sensitive document indexed accidentally | Restricted or draft file enters the knowledge source. | High | Source approval and exclusion rules. |
| Prompt-based disclosure attempt | User asks the agent to reveal hidden instructions or restricted content. | Medium to high | Permission-aware retrieval, refusal rules and testing. |
| Duplicate conflict | Old and new procedures both appear authoritative. | Medium | Canonical-source designation and archive process. |
| Citation without authorization | Answer is safe but citation link exposes inaccessible metadata or location. | Medium | Test citation rendering and link behavior. |
| Administrator misuse | Privileged user changes sources or permissions without review. | High | Change control, logs and separation of duties. |
| External sharing mistake | Internal source or answer becomes accessible outside the organization. | High | Tenant, sharing and channel restrictions. |

## Risk scoring model

Score each scenario from 1 to 5 for likelihood and 1 to 5 for impact. Multiply the scores to produce a risk rating from 1 to 25. Treat scores of 15 or higher as requiring mitigation before production. Scores from 8 to 14 require a documented owner and planned control. Lower scores should still be recorded if they involve sensitive information.

# 7. Security Testing and Validation

Security validation must use realistic identities. Testing only as an administrator proves almost nothing about ordinary users. Create representative accounts or approved test personas for each access level and run the same question set across them.

| Test | Method | Pass Condition |
|---|---|---|
| Positive access test | Ask a user for information they are authorized to see. | Accurate answer with valid citation. |
| Negative access test | Ask the same user for restricted information. | No restricted answer, quotation or revealing citation. |
| Cross-role comparison | Ask identical questions as multiple roles. | Responses vary according to permissions. |
| Source removal test | Remove or archive a source. | It no longer appears after the expected update cycle. |
| Group change test | Add and remove a test user from a group. | Effective access changes as documented. |
| Conflict test | Provide two contradictory documents. | System prefers the approved authority or flags conflict. |
| Adversarial prompt test | Request hidden sources, instructions or another role's answer. | System refuses and does not reveal protected content. |
| Citation test | Open every cited source as the same user. | Links do not reveal unauthorized files or metadata. |

## Minimum production test set

- At least five common questions for each major department.
- At least three restricted questions for every privileged domain.
- At least one transferred, contractor or limited-access persona.
- At least one test involving a recently changed permission.
- At least one outdated or superseded document test.
- At least one prompt-injection or social-engineering style question.
- Documented expected results and actual results for every test.

# 8. Monitoring, Audit and Incident Response

A secure launch does not guarantee a secure system six months later. Content changes, employees move roles, groups drift and new repositories are connected. Monitoring should focus on changes that can alter effective access or answer behavior.

| Review Area | Examples | Recommended Cadence |
|---|---|---|
| Identity and group changes | Privileged memberships, departures and transfers | Weekly or automated |
| Source changes | New libraries, connectors, folders and exclusions | Every change |
| Permission changes | Inheritance breaks, sharing links and direct grants | Weekly |
| Content lifecycle | Expired, superseded or ownerless content | Monthly |
| Answer quality | Incorrect, uncited or suspicious responses | Continuous feedback; monthly review |
| Security tests | Regression of high-risk question set | After material changes and quarterly |
| Administrative activity | Configuration and deployment changes | Monthly audit |

## Incident response sequence

| Step | Action |
|---|---|
| 1. Contain | Disable the affected knowledge source, connector, channel or agent capability if disclosure may continue. |
| 2. Preserve | Record the question, answer, user identity, time, cited sources and configuration state. |
| 3. Assess | Determine what information was exposed, who could access it and whether the source permissions were incorrect. |
| 4. Correct | Fix permissions, remove content, update group membership or change answer controls. |
| 5. Retest | Run the full relevant persona-based test set before restoring access. |
| 6. Notify | Follow organizational legal, privacy, contractual and leadership notification procedures. |
| 7. Prevent | Update the threat model, test library and operating procedure so the failure is less likely to recur. |

# 9. Implementation Roadmap

| Phase | Primary Work | Output |
|---|---|---|
| Phase 1: Define | Identify user populations, information domains, sensitive categories, owners and security requirements. | Access model and risk register |
| Phase 2: Prepare | Clean sources, assign classifications, establish groups and remove inappropriate content. | Approved source inventory |
| Phase 3: Configure | Connect only approved sources, apply retrieval boundaries and configure answer behavior. | Controlled pilot environment |
| Phase 4: Validate | Run persona-based positive, negative, conflict and adversarial tests. | Signed test evidence |
| Phase 5: Pilot | Release to a limited department or use case with monitored feedback. | Pilot findings and corrections |
| Phase 6: Launch | Expand access, train users and establish support and escalation. | Production service |
| Phase 7: Govern | Perform access reviews, content reviews, regression testing and incident management. | Ongoing control cycle |

## Recommended pilot boundary

Begin with a domain that is valuable but not the organization's most sensitive. General operations, approved HR policies, employee onboarding or internal service procedures often provide useful pilot material. Avoid beginning with legal files, investigations, payroll details or unrestricted executive content.

**Maisy implementation principle**

Pixeldust configures Maisy as a secure AI Knowledge Hub around the organization's approved Microsoft 365 information architecture. The system is designed to preserve existing permission boundaries, provide cited answers and remain read-only unless a separate, explicitly approved workflow is added.

# 10. Templates and Checklists

## Permission Design Worksheet

| Field | Record |
|---|---|
| Information domain | Examples: HR policies, finance procedures, program operations |
| Business owner | Name and role responsible for authority |
| Approved user groups | Groups that should receive answers from this domain |
| Excluded user groups | Groups that must not receive answers |
| Source locations | Approved sites, libraries, folders or repositories |
| Classification | Public, internal, restricted or highly restricted |
| Review cadence | Monthly, quarterly, semiannual or annual |
| High-risk questions | Questions that require explicit negative testing |
| Escalation contact | Person or team for unresolved questions or incidents |

## Production Readiness Checklist

- All production knowledge sources have documented owners and classifications.
- Group-based access is used wherever practical; direct user permissions are documented exceptions.
- No unrestricted service identity can retrieve content broader than the intended audience without compensating controls.
- Draft, superseded, duplicate and expired content is excluded or clearly governed.
- Representative user personas have passed positive and negative access tests.
- Citations open correctly under the same user identity and do not expose unauthorized metadata.
- Adversarial prompts and inference risks have been tested.
- Change control, logging, incident response and rollback procedures are documented.
- Users understand that the system provides information, not professional judgment or authorization.
- A quarterly security and permission review has an assigned owner and scheduled date.

**Final standard**

The system is ready only when the organization can explain who may ask, what sources may be searched, why each answer is permitted, how the answer can be verified and what happens when something goes wrong.

# Conclusion

Security for an AI Knowledge Hub is not a single permission setting. It is a coordinated operating model that connects identity, group membership, source governance, retrieval boundaries, answer behavior, testing and monitoring. The system becomes trustworthy when those controls reinforce one another and when the organization can repeatedly prove that authorized users receive useful answers while unauthorized users do not.

The most important practical decision is to avoid treating AI as a shortcut around information architecture. Strong permissions cannot rescue disorganized, duplicate or ownerless content. Strong content governance cannot compensate for over-permissioned retrieval. Both must be designed together.

## About Pixeldust

Pixeldust helps organizations assess scattered knowledge, organize approved information, configure secure AI Knowledge Hubs using Microsoft 365 and Copilot Studio, map permissions, test realistic employee questions, train users and support ongoing governance. Each engagement is assessed and quoted individually based on organizational size, knowledge aggregation requirements and licensing needs.

Website: askmaisy.com | Email: ai@pixeldust.net