RHEINSIGHTS

# A Brief Guide to Secure Agentic Knowledge Retrieval

April 17, 2026

# Meet the Speaker

# An Enterprise Search Veteran

Serves TOP 500 organizations worldwide for their knowledge search needs since more than 15 years

Co-Founder of RheinInsights, a Germany-based specialist for reliable and secure large-scale RAG (and search) integrations. RheinInsights was founded in 2024.

Image: RheinInsights

# Agentic Use Cases

# Agentic and RAG Use Cases are Manifold

**Project Work**

Meeting preparation
Document summaries
Document generation
Project summaries and overviews
Knowledge search
…

**CRM**

Account and opportunities
Communication summaries
Contract and agreement search
…

**Onboarding, HR and Identity**

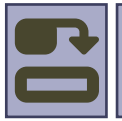
News
Corporate Identity
Expert search
PMO use cases
…

**Support**

Case deflection
Case preparation
Code search
Supporting the support
…

**Regulations and Processes**

Process retrieval
Regulation retrieval
Compliance
Checks
…



**Agents and Automation**

Agentic Retrieval
Data analysis
Data tagging
Workflow integrations
Offline research
…

**Legal Search**

Contract Analysis
Clause retrieval
Risk Management
…

# Should Every Agent Run in Sudo Mode?

**Imagine**

- A support agent with full access to the full HR database.
- A coding agent which can also access secret projects.

Broad agentic and bot use cases makes it

- necessary to **think of the boundaries** these should run within

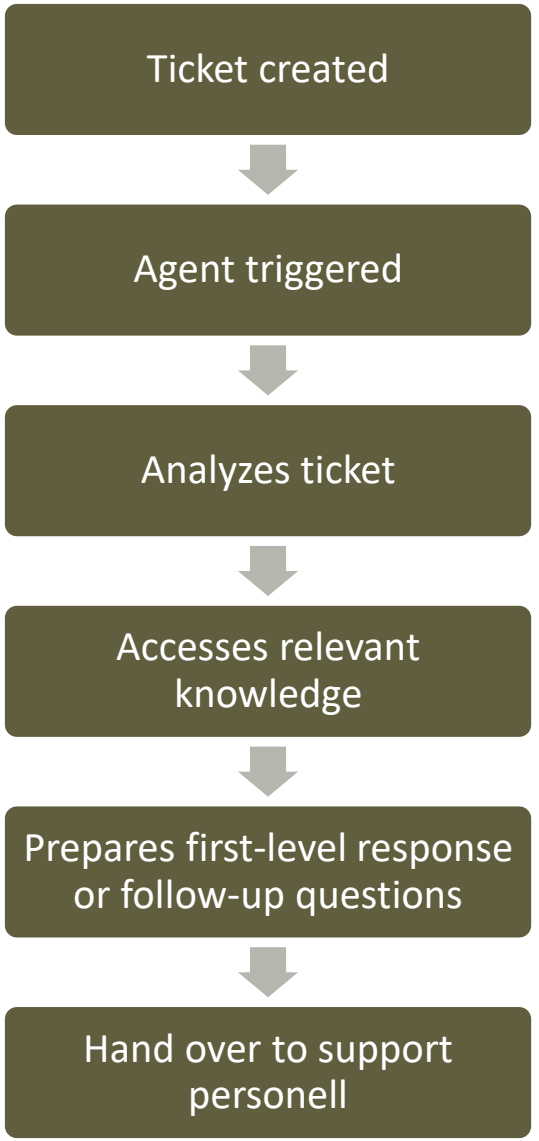
But, access to **relevant** knowledge is important.

For broader use cases more knowledge is needed


Image: AI generated with Copilot

# Deep Dive Example: Autonomous Support Agent

- Triggered whenever tickets come in
- Access to all knowledge bases and tickets (Jira, Confluence, MS Dynamics)
- Generates a first-level response based on the gathered knowledge (visible to actual support personell)
- May propose scripts
- Therefore access to knowledge bases and tickets which are accessible to first-level support personell



# Permission Based RAG

# Admin Connectors vs. User Connectors

## User Connectors

- Users individually connect their knowledge sources to an AI solution
- Claude, OpenClaw, …
- Caveat: no governance, inefficiencies, risk for data leak
- But: agent solution leverages the user‘s access rights – permission concept in place.

## Admin Connectors

- Centrally operated integrations
- Index *entire* content sources in OpenSearch
- Permissions needed

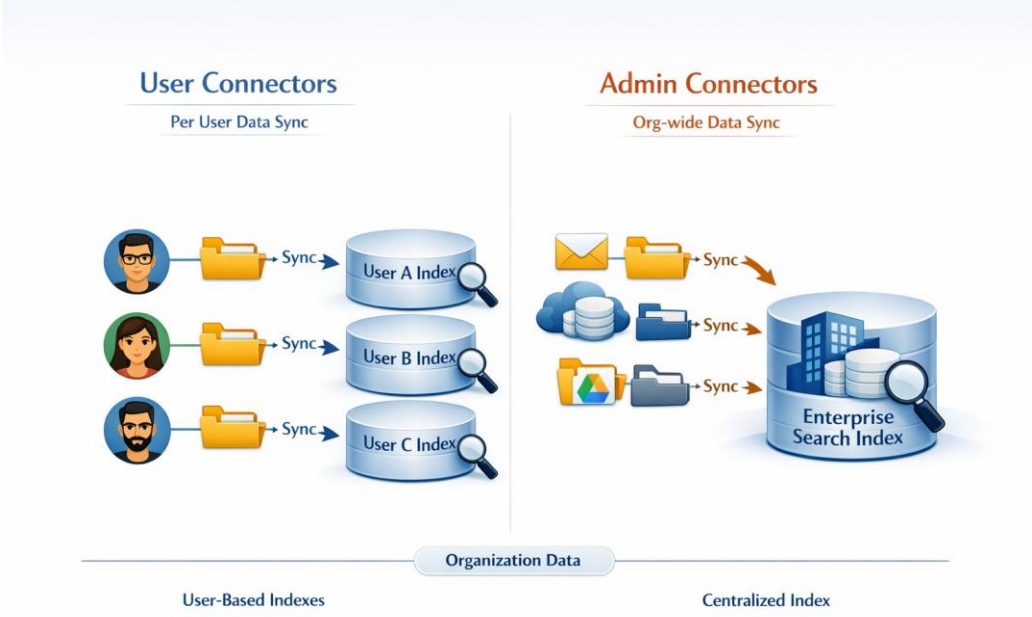



Image: AI generated with Copilot

# Example Systems

- SharePoint Online comes with site permissions, based on roles, direct sharing and for certain lists creator-owner permissions

- Confluence comes with space permissions, which can be narrowed down page by page (artificial ACLs in our case)

- Livelink/OpenText Content Server comes with group permissions and additional fields for security clearance levels and supplemental markings (three ACL fields by conjunction)
- …

# Permission-Based RAG – In a Nutshell

**Indexing Time**

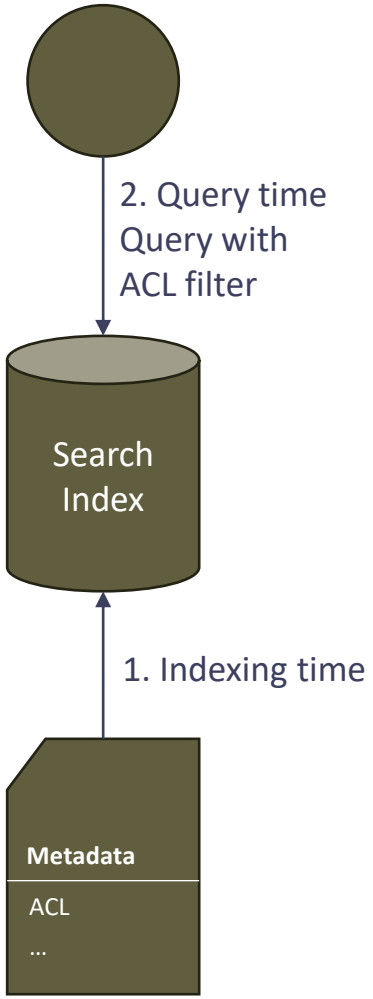
- At indexing time all documents are enriched with an ACL field
- A principal store is built up

**Query Time**

- At query time, the agent, bot or user authenticates against the RAG application
- Before OpenSearch is queried, the ACL tokens for the user are fetched from the principal store
- Then the hybrid / vector search is executed, including filters on the ACL fields

**Result**

- Only documents where the agent has access to are returned



# Permission-Based RAG

Admin Connectors need to reflect the content source permissions in search.

**Approach**

Each indexed document comes with

- The original document body
- A vector representation of the content
- Metadata
- Allow and Deny Access Control Lists as part of the metadata (keyword field)

```
{
    "settings": {
        "index": {
            "knn": true
        },
        "analysis": {
            "analyzer": {
                "en": {
                    "tokenizer": "standard",
                    "filter": [
                        "lowercase",
                        "stemmer"
                    ]
                }
            }
        }
    },
    "mappings": {
        "properties": {
            "allow_acl": {
                "type": "keyword"
            },
            "body": {
                "type": "text",
                "analyzer": "en"
            },
            "id": {
                "type": "keyword"
            },
            "vectorBody": {
                "type": "knn_vector",
                "dimension": 1536,
                "space_type": "l2",
                "mode": "on_disk",
                "method": {
                    "name": "hnsw"
                }
            }
        }
    }
}
```

# Allow and Deny Access Control Lists

The ACL field includes

- Users, roles, groups and potentially artificial tokens
- Symbolize who has access to this document and who might be fully excluded

# User-Group-Relationships

An additional principal index comprises the user-group relationships as follows.

```
{
    "mappings": {
        "properties": {
            "id": {
                "type": "keyword"
            },
            "parentPrincipals": {
                "type": "keyword"
            }
        }
    }
}
```

## Approach

- The principal index serves as key-value store.
- The connector needs to
  - Collect the user-group relationships from the content source. Stored as list of strings.
  - Harmonize each user name to e.g. the user principal name / mail address (UPN).
  - The UPN serves as key, the parent groups, roles or articifial tokens belonging to the user serve as value.

# Thank You!

RheinInsights GmbH
Alt-Pempelfort 2
40211 Düsseldorf

Germany

info@rheininsights.com

+49 211 971 71 345