ATC AI Intranet: High-Level PRD Summary

9 EPICs Overview(Use Cases —> Use cases Examples for Agentic Workflows)

EPIC 1: Enterprise Search (Indexed Search using Elasticsearch)

Description: Fast, scalable, secure search foundation leveraging Elasticsearch for indexing diverse sources (documents, KBs, frameworks, SaaS, EX features). Supports semantic search via vector database integration (RAG), real-time and batch indexing, with sub-second query response times.

Key Features:

- Real-time and batch indexing of all content sources
- Semantic enhancement via vector database (hybrid keyword + semantic search)
- Persistent search bar with autocomplete and advanced filters
- Admin configuration dashboard for indexing management
- Inline preview and deep linking to sources

EPIC 2: AI Assistant (Conversational Intelligence)

Description: Context-aware conversational AI for answering questions and providing insights, grounded in indexed data (search, KBs, frameworks, EX) to minimize hallucinations. Multi-LLM support with citations and confidence scoring.

Key Features:

- Multi-LLM support (GPT-4, Claude, custom models) with RAG grounding
- Contextual synthesis with inline citations and confidence scores
- Threaded chat interface with transparency pane (show sources/reasoning)
- Multi-language NLP support (100+ languages)
- Personalization based on org chart and user profile

EPIC 3: Knowledge Base (Team/Department Categorization)

Description: Structured KB repository organized by team and department (aligned with org chart), fully indexed for search and AI access. Includes version control, approval workflows, and role-based access.

Key Features:

- Hierarchical categorization by org structure
- Rich text editor with version history and attachments
- Approval workflows for quality curation
- Search within KB with advanced filtering
- Role-based access control and audit logs
- Auto-population from frameworks and SaaS integrations

EPIC 4: Framework/Accelerator Integration (Single Source of Truth Hub)

Description: Unified hub connecting frameworks, accelerators, SaaS products, and KBs behind a secure firewall. Core differentiator for services firms building client deliverables—centralizes all assets into one AI-powered platform.

Key Features:

- Centralized registry for frameworks, accelerators, custom SaaS
- Unified indexing across code repos, docs, SaaS APIs, file shares
- Cross-framework search and AI-powered comparison
- Integration dashboard with sync status and conflict resolution
- Client isolation and role-based access for multi-client scenarios
- Admin monitoring and sync management

EPIC 5: Role-Based Access Control (Security & Permissions)

Description: Granular security controls tied to org chart structures for automatic, dynamic access management. Zero-trust architecture with real-time enforcement across all platform components.

Key Features:

- Permissions matrix linked to org chart (hierarchical inheritance)
- Dynamic content filtering in search, AI, KB, and dashboards
- Access request and approval workflows
- Compliance-ready audit logs (SOC 2, ISO 27001, GDPR)

- Visual indicators for restricted content (locks, blur effects)

EPIC 6: Custom Agentic Workflows (Automation & Orchestration)

Description: Build multi-step agentic workflows that automate complex tasks by orchestrating AI, search, integrations, and human approvals. Grounded in indexed data with multi-LLM reasoning support.

Key Features:

- Visual drag-drop workflow designer (triggers, actions, conditions, loops)
- Multi-LLM support with grounding in search/KB/frameworks
- Template library with pre-built workflows
- Branching logic and error handling with retry/fallback
- Structured outputs (tables, summaries, dashboards, notifications)
- No-code builder + code mode (YAML/JSON) for advanced users

EPIC 7: Central Dashboard (Admin vs Employee Views)

Description: Role-differentiated dashboards—Admin for system monitoring and analytics, Employee for personalized daily summaries and quick access to relevant content.

Key Features:

Admin Dashboard:

- System health monitoring (indexing, search, AI performance)
- Usage analytics and insights (user engagement, content analytics)
- Quick admin actions (re-index, review requests, error logs)

Employee Dashboard:

- Personalized widgets (tasks, news feeds, org updates, quick links)
- Auto-population based on role and org context
- Customizable layout with drag-drop widgets
- Real-time updates and notifications

EPIC 8: Productivity Assistant (Personal Employee AI)

Description: Dedicated AI agent for each employee that proactively curates daily tasks, summarizes relevant content, and streamlines workflows. Distinct from general AI Assistant—focuses on personal work context management.

Key Features:

- Auto-curated task list from workflows, approvals, calendar, channels
- Conversational task management (natural language + voice input)
- AI-generated daily brief (highlights, dept news, org updates)
- Query-based content curation (summarize channels, track framework updates)
- Integration with calendar and workflows
- Customizable dashboard widgets for personalized curation

EPIC 9: Employee Experience (EX) Features (NewsFeed, Channels, Org Chart, Collaboration)

Description: Comprehensive employee experience platform integrated with search, AI, and KB. Includes social/collaboration features (feeds, channels), organizational tools (org chart, directory), and event management.

Key Features:

News Feeds:

- Real-time company/dept news posts with rich media
- Follow/subscribe model with personalized timelines
- Reactions, comments, sharing

Channels:

- Threaded discussions by topic/project/team
- Direct messaging and group chats
- File sharing and collaborative editing
- Integration with KB (link entries to channels)

Org Chart & Directory:

- Interactive hierarchical org visualization
- Employee profiles with contact info, reporting structure
- Search by name, dept, role, skills
- Integration with permissions (dept-based access)

Events & Calendar:

- Company events, team meetings, personal calendars
- RSVP tracking and reminders
- Integration with Productivity Assistant (auto-create tasks)

Collaboration Tools:

- Polls, surveys, Q&A forums
- Document co-editing with version control
- @mentions and notifications across feeds/channels

Cross-EPIC Integration Points

All 9 EPICs are deeply integrated to create a unified platform:

- **Search (EPIC 1) indexes content from KBs (3), Frameworks (4), EX (9)**
- **AI Assistant (EPIC 2) grounds responses in Search (1), KBs (3), Frameworks (4), EX (9)**
- **Workflows (EPIC 6) orchestrate actions across Search (1), AI (2), KBs (3), Integrations (4)**
- **Dashboards (EPIC 7) display data from all EPICs, customized by Permissions (5)**
- **Productivity Assistant (EPIC 8) curates from Workflows (6), EX (9), Calendar, driven by AI (2)**
- **Permissions (EPIC 5) enforce access control across all EPICs in real-time**

Deployment Model

- **Customer-hosted (on-prem, private cloud, hybrid, air-gapped)**
- **One-time customization fees for setup/integration**
- **Annual IP licensing for proprietary components**

Target Market

- **Services firms (consulting, system integrators) building frameworks/accelerators for clients**
- **Enterprises seeking unified AI-powered knowledge access behind secure firewall**

Competitive Positioning

Direct competitor to Glean, but designed specifically for services firms needing to unify client deliverables into a cohesive, AI-powered single source of truth.

ATC AI Intranet: Complete Product Requirements Document (PRD)

Executive Summary

ATC's AI Intranet is a secure, customer-hosted enterprise platform designed to serve as the **single source of truth** for modern organizations. Unlike traditional intranets, this framework unifies enterprise search (powered by Elasticsearch), AI assistance (multi-LLM), knowledge management, custom frameworks/accelerators, SaaS products, and employee experience features behind a secure firewall.

Deployment Model: On-premises, private cloud, hybrid, or air-gapped environments
Monetization: One-time customization fees for setup/integration + annual IP licensing for proprietary components (custom agents, semantic modules, advanced features)
Target Market: Services firms (consulting, system integrators) building accelerators/frameworks for enterprise clients, and enterprises seeking unified AI-powered knowledge access

Competitive Positioning: Direct competitor to Glean, but designed for **services firms** who need to unify client deliverables (frameworks, accelerators, SaaS, KBs) into a cohesive, AI-powered employee experience.

Document Structure

This PRD is organized into **9 EPICs** representing major functional areas:

1. **Enterprise Search** (Indexed Search using Elasticsearch)
2. **AI Assistant** (Conversational Intelligence)
3. **Knowledge Base** (Team/Department Categorization)
4. **Framework/Accelerator Integration** (Single Source of Truth Hub)
5. **Role-Based Access Control** (Security & Permissions)
6. **Custom Agentic Workflows** (Automation & Orchestration)
7. **Central Dashboard** (Admin vs Employee Views)
8. **Productivity Assistant** (Personal Employee AI)
9. **Employee Experience Features** (NewsFeed, Channels, Org Chart, Collaboration)

Each EPIC includes:

- **Description**: Strategic goals and objectives
- **Features**: Key capabilities with rationale

- **Sub-Features**: Granular implementation details (UI, interactions, flows, edge cases, prototype guidance)

EPIC 1: Enterprise Search (Indexed Search using Elasticsearch)

Description

Enable fast, scalable, and secure search as the foundation for a single source of truth. Leverage **Elasticsearch** for indexing to handle large-scale data efficiently, integrating with KBs, frameworks/accelerators, SaaS products, and Employee Experience (EX) features for unified access.

Objectives:

- Reduce search time to sub-second response (<500ms for 90th percentile)
- Support semantic queries via vector database integration (RAG)
- Ensure relevance through ML-powered ranking algorithms
- Prioritize real-time indexing for dynamic content (EX feeds, channels, org updates)
- Handle billions of documents with distributed architecture

Feature 1.1: Indexed Search Using Elasticsearch

Sub-Feature 1.1.1: Real-Time and Batch Indexing

Implementation Details:

- **Data Sources**: Documents (PDF, DOCX, PPT), emails, KBs, SaaS APIs, frameworks/accelerators, EX feeds/channels/org data, collaboration tools
- **Architecture**: Elasticsearch distributed cluster with configurable shards/replicas for high availability
- **Indexing Modes**:
    - *Real-time*: Webhook-triggered indexing for EX feeds, channels, org changes (<5 second latency)
    - *Batch*: Scheduled crawling for large repositories (nightly/hourly configurable)
- **Content Processing**: OCR for scanned documents, metadata extraction (author, date, dept, tags), entity recognition (people, projects, locations)

Prototype Details:

- Admin panel with indexing queue visualization (pending, processing, completed)
- Progress indicators for batch jobs (e.g., "Indexing 1,247 of 5,000 documents - 24% complete")
- Logs with error details (e.g., "Failed to index 3 documents: Permission denied")

Edge Cases:

- Partial index failures (retry logic, admin alerts)
- Large file handling (>100MB files chunked for processing)
- Duplicate detection (hash-based deduplication)
- API rate limiting for SaaS sources (exponential backoff)

Sub-Feature 1.1.2: Semantic Enhancement via Vector Database

Implementation Details:

- **Vector DB Integration**: Pinecone, Weaviate, or Qdrant for embedding storage
- **Embedding Models**: Support for OpenAI Ada-002, Cohere, or custom-trained embeddings
- **Hybrid Search**: Combine keyword (Elasticsearch BM25) + semantic (vector similarity) with configurable weighting
- **RAG Pipeline**: Query → Vector search for context → LLM grounding → Response with citations

Search Capabilities:

- Full-text search with BM25 ranking
- Fuzzy matching (Levenshtein distance) for typo tolerance
- NLP-powered query understanding (intent detection, entity extraction)
- Wildcard and regex support for advanced users

Prototype Details:

- Toggle between "Keyword" vs "Semantic" vs "Hybrid" search modes
- Relevance score visualization (0-100 scale with explanation tooltips)
- "Similar Documents" suggestions based on vector similarity

Edge Cases:

- Queries with no vector embeddings available (fallback to keyword)
- Multilingual queries (auto-detect language, use appropriate embeddings)
- Very long queries (truncate to embedding model limits, show warning)

Sub-Feature 1.1.3: Search UI - Top Persistent Search Bar

UI Components:

- **Search Bar**: Omnipresent across all pages (header position)
    - Autocomplete dropdown with suggestions (recent searches, popular queries, indexed entities)
    - Voice input button (speech-to-text)
    - Advanced filters toggle (collapsed by default)
- **Results Display**:
    - Responsive card layout (grid on desktop, list on mobile)
    - Each card contains:
        - **Thumbnail**: Document preview, app icon, or generic file icon
        - **Title**: Highlighted query matches (bold)
        - **Snippet**: 2-3 lines of contextual excerpt with query highlights
        - **Metadata**: Source badge (e.g., "EX Feed", "KB - Engineering", "Framework X"), last modified date, author/dept
        - **Relevance Score**: Visual indicator (e.g., 92/100 with progress bar)
        - **Quick Actions**: "Open", "Preview", "Share", "Add to KB"

Prototype Details:

- Mock search bar with animated autocomplete (type "project" → suggest "Project Alpha", "Project Reports", "Project Management KB")
- Results grid with 8-12 sample cards across different sources
- Hover states for cards (elevate shadow, show additional metadata)
- Mobile mockup with simplified card layout

Sub-Feature 1.1.4: Interactions and Flows

Primary Flow:

1. User enters query in search bar → Instant results appear as they type (debounced after 300ms)
2. Query highlights appear in yellow across titles/snippets
3. User refines via advanced filters (source type, date range, department from org chart, author, tags)
4. Pagination with infinite scroll (load 20 results at a time)
5. Click result → Inline preview modal (75% screen width) OR deep link to source application
6. User can "Open in Channel" for EX content or "View in KB" for knowledge articles

Advanced Filters:

- **Source Type**: Checkboxes for "Documents", "EX Feeds", "Channels", "KBs", "Frameworks", "SaaS Apps"
- **Date Range**: Preset options (Today, This Week, This Month, This Year) + custom date picker
- **Department**: Dropdown populated from org chart hierarchy
- **Author**: Autocomplete search for people
- **File Type**: PDF, DOCX, XLSX, PPT, etc.
- **Tags**: Multi-select from indexed tags

Error Handling:

- **No Results**: Display "No results found for '[query]'" with suggestions:
    - "Try different keywords"
    - Related queries based on query expansion (e.g., "Did you mean: 'project timeline'?")
    - "Browse [Department] KB" or "Ask AI Assistant"
- **Query Timeout**: "Search is taking longer than expected. Results may be incomplete." with retry button
- **Index Unavailable**: "Search temporarily unavailable. Please try again." with status page link

Prototype Details:

- Interactive filter panel (slide in from left on desktop, bottom sheet on mobile)
- Mock "no results" state with suggested actions
- Preview modal with document viewer (PDF renderer, syntax highlighting for code)
- Error state screens for timeouts and failures

Sub-Feature 1.1.5: Admin Configuration Dashboard

Admin Features:

- **Indexing Configuration**:
    - Schedule management (cron expressions for batch jobs)
    - Custom field mappings (e.g., prioritize "dept" field in scoring, boost recent documents)
    - Synonym management (e.g., "PM" → "Project Manager", "Product Manager")
    - Stopword configuration (language-specific)
- **Index Health Monitoring**:
    - Cluster status (green/yellow/red) with shard allocation details
    - Index size, document count, query latency (p50, p90, p99)
    - Re-index triggers for schema changes
    - Disk usage alerts (threshold warnings)
- **Logs and Troubleshooting**:

    - Indexing logs (filterable by source, date, status)
    - Query logs for analytics (most searched terms, zero-result queries)
    - Error logs with stack traces for failed indexing jobs

Prototype Details:

- Dashboard with status cards (green/yellow/red indicators)
- Indexing schedule calendar view (show upcoming and past jobs)
- Interactive log viewer with filters and search
- Mock shard allocation diagram (visualize distributed architecture)

Edge Cases:

- Schema conflicts during re-indexing (show affected documents, provide manual resolution UI)
- Disk space warnings (auto-cleanup of old indices, admin approval required)
- Multiple admins editing same config (optimistic locking with conflict resolution)

Screenshot References *(Extract from Glean video, adapt for ATC)*

Show Image *Reference: Persistent search bar with autocomplete*

Show Image *Reference: Search results card layout with metadata*

Show Image *Reference: Advanced filters panel*

EPIC 2: AI Assistant (Conversational Intelligence)

Description

Deliver intelligent, context-aware conversational AI for answering questions, providing insights, and assisting with tasks. The AI Assistant is **grounded in indexed data** (search, KBs, frameworks, EX) to minimize hallucinations and provide accurate, cited responses.

Objectives:

- Achieve high resolution rate (target: 82%+ queries answered without human escalation)
- Provide contextual, role-aware responses (leverage org chart and user profile)
- Support multi-turn conversations with memory
- Enable multi-LLM flexibility (customer choice: GPT-4, Claude, custom models)

- Integrate seamlessly with productivity workflows (email drafting, summarization, analysis)

Feature 2.1: Core AI Assistant

Sub-Feature 2.1.1: Multi-LLM Support with Grounding

Implementation Details:

- **LLM Options**: GPT-4 Turbo, Claude 3 Opus/Sonnet, custom fine-tuned models on customer data
- **Model Selection**:
  - Admin-configured default per deployment
  - User-selectable via dropdown (if enabled by admin)
  - Auto-routing based on query type (e.g., code → specialized model, general → Claude)
- **Grounding Mechanisms**:
  - **RAG (Retrieval-Augmented Generation)**: Query → Elasticsearch/Vector DB retrieval → Context injection → LLM generation
  - **Citation Linking**: Every factual statement includes inline source links (e.g., "[1] Engineering KB - API Guidelines")
  - **Confidence Scoring**: Display confidence level (High/Medium/Low) based on source quality and retrieval score
- **Custom Training**:
  - Option to fine-tune models on customer-specific data (e.g., 1000+ hours of support tickets, internal docs)
  - Feedback loop: User ratings (thumbs up/down) → Retraining pipeline

Prototype Details:

- LLM selector dropdown in chat interface (show model name, version, description)
- Response with inline citations (clickable [1], [2] markers linking to sources)
- Confidence indicator badge ("High confidence - 3 sources" tooltip)

Edge Cases:

- LLM API unavailable (fallback to cached responses or queue for retry)
- Query outside knowledge base (respond "I don't have information on this. Would you like me to search the web?" with admin-controlled web search toggle)
- Conflicting information from sources (present both perspectives with citations)

Sub-Feature 2.1.2: Contextual Synthesis and Multi-Language Support

Capabilities:

- **Summarization**: "Summarize the latest news feed posts in my department"
- **Comparative Analysis**: "Compare Framework A vs Framework B for cloud migration"
- **Explanations**: "Explain our company's parental leave policy"
- **Step-by-Step Guidance**: "How do I submit an expense report?"
- **Response Styles** (user-selectable):
  - **Factual**: Concise, bullet-point format
  - **Balanced**: Narrative with pros/cons
  - **Creative**: Conversational, engaging tone
- **Multi-Language NLP**:
  - Support 100+ languages via Google Translate or DeepL integration
  - Auto-detect query language, respond in same language
  - Translate sources if needed (with "Translated from [language]" disclaimer)

Prototype Details:

- Response style toggle (radio buttons: Factual / Balanced / Creative)
- Language selector dropdown (auto-detect with override option)
- Mock responses in different styles for same query (e.g., "What is our vacation policy?")

Sub-Feature 2.1.3: Chat UI - Threaded Conversation Interface

UI Components:

- **Chat Window**:
  - User bubbles (right-aligned, blue background)
  - AI bubbles (left-aligned, gray background)
  - Timestamps (collapsed by default, show on hover)
  - Typing indicator (animated dots) during generation
- **Input Bar**:
  - Text input field (multi-line, auto-expand up to 5 lines)
  - Voice input button (press-and-hold or toggle)
  - Attachment button (upload documents for context)
  - Send button (enter key or click)
- **Side Panel** (collapsible):
  - **LLM Selector**: Dropdown to change model mid-conversation
  - **Response History**: List of past conversations (searchable, resumable)
  - **Transparency Pane**: Show reasoning steps, sources used, confidence scores for current response

Prototype Details:

- Chat mockup with 5-7 message exchanges (varied query types)

- Side panel in open and collapsed states
- Transparency pane showing "Sources used: [KB Article], [EX Channel Post], [Framework Doc]" with confidence breakdown

---

Sub-Feature 2.1.4: Interactions and Conversation Flows

Primary Flow:

1. User types query → AI generates response with inline citations
2. User clicks citation → Inline preview modal or navigate to source
3. User asks follow-up → AI maintains context from conversation history
4. User unsatisfied → "Regenerate" button (try alternative phrasing or different LLM)
5. Complex query → AI asks clarifying questions ("Did you mean [Option A] or [Option B]?")
6. Escalation → "I need more help" button → Routes to human support or creates task in Productivity Assistant

Advanced Interactions:

- **Thread Branching**: User can "branch" conversation at any point (explore alternative responses without losing main thread)
- **Save Conversation**: Bookmark important exchanges, share with colleagues
- **Export**: Download conversation as PDF or markdown

Error Handling:

- **Low Confidence Response**: Display warning banner "I'm not certain about this answer. Here's what I found, but please verify." with sources
- **No Grounding Sources Found**: "I don't have enough information to answer this accurately. Would you like me to [Search Web] or [Ask in #general-channel]?"
- **LLM Timeout**: "Response generation is taking longer than expected. Please try again or simplify your query."

Prototype Details:

- Interactive regenerate button with spinner animation
- Branching UI (show split icon, navigate between branches)
- Low-confidence warning banner with yellow alert icon
- Mock clarifying question flow (e.g., "Did you mean vacation policy for US employees or international?")

Sub-Feature 2.1.5: Personalization via Org Context

Context Sources:

- **User Profile**: Role, department, location, tenure, reporting structure from org chart
- **Activity History**: Recent searches, KB articles viewed, channels joined, frameworks accessed
- **Department-Specific Content**: Auto-filter responses to prioritize user's dept KBs, channels, and frameworks

Personalization Examples:

- Query: "What's our expense policy?" → Response prioritized for user's country/region
- Query: "Summarize recent updates" → Pulls from user's subscribed channels and dept news feeds
- Query: "Find project documentation" → Weights results from user's dept and project tags

Prototype Details:

- Mock responses showing personalization (e.g., "Based on your role as [Engineering Manager], here's...")
- Admin config panel for personalization settings (enable/disable context types)

Screenshot References *(Extract from Glean video, adapt for ATC)*

Show Image *Reference: AI Assistant chat interface*

Show Image Show Image *Reference: Inline citations and source transparency*

Show Image *Reference: AI Assistant embedded in search results*

Show Image *Reference: Multi-turn conversation with context*

EPIC 3: Knowledge Base (Team/Department Categorization)

Description

Provide a structured, categorized Knowledge Base (KB) as a central repository for organizational knowledge. KBs are organized by **team and department** (aligned with org chart), fully indexed for search, and integrated with AI Assistant for intelligent retrieval.

Objectives:

- Enable team/dept-specific organization (avoid monolithic, ungoverned wiki sprawl)
- Support easy curation with approval workflows (maintain quality)
- Seamless integration with search, AI Assistant, and EX features
- Version control and audit trails for compliance

Feature 3.1: Knowledge Base Structure and Management

Sub-Feature 3.1.1: Hierarchical Categorization by Team/Department

Implementation Details:

- **Organizational Alignment**: KB categories mirror org chart structure
  - Level 1: Company-wide (e.g., "HR Policies", "IT Guidelines")
  - Level 2: Department (e.g., "Engineering", "Sales", "Marketing")
  - Level 3: Team (e.g., "Backend Team", "Enterprise Sales", "Content Marketing")
  - Level 4+: Sub-categories (e.g., "API Documentation", "Onboarding Guides")
- **Elasticsearch Indexing**: Each KB entry is indexed with:
  - Category path (e.g., "Engineering > Backend > API Docs")
  - Tags (e.g., "python", "rest-api", "authentication")
  - Author, creation/modified dates, version number
  - Full-text content (with semantic embeddings)
- **Dynamic Categories**: Admins can create/modify category structures; changes propagate to search index

Prototype Details:

- Expandable tree view (left sidebar) with folder icons and counts (e.g., "Engineering (47)")
- Breadcrumb navigation (e.g., "Home > Engineering > Backend > API Docs")
- Drag-and-drop category management for admins

Edge Cases:

- Entry belongs to multiple categories (support tags instead of forcing single category)
- Department restructure (bulk re-categorization tool with preview)
- Orphaned entries after category deletion (move to "Uncategorized" with admin alert)

Sub-Feature 3.1.2: Entry Creation and Rich Editing

UI Components:

- **Rich Text Editor**:
  - WYSIWYG interface (bold, italic, lists, tables, code blocks)

    - Markdown support (side-by-side preview)
    - Syntax highlighting for code snippets (100+ languages)
    - Embedded media (images, videos, diagrams)
    - Internal linking (autocomplete to other KB entries, docs, frameworks)
- **Metadata Panel**:
    - Category selector (dropdown with tree view)
    - Tags (multi-select autocomplete from existing tags)
    - Attachments (upload PDFs, spreadsheets, etc.)
    - Related links (suggest similar KB entries via vector search)
    - Visibility (Public, Department-only, Team-only)
- **Version History**:
    - Auto-save drafts every 30 seconds
    - Manual version creation (e.g., "Version 1.2 - Updated API endpoints")
    - Side-by-side diff viewer (compare versions)
    - Rollback to previous version (with admin approval)

Prototype Details:

- Editor mockup with toolbar and metadata panel
- Version history drawer (list versions with dates, authors, change summaries)
- Diff viewer with highlighted changes (green additions, red deletions)

Sub-Feature 3.1.3: Approval Workflows for Quality Curation

Workflow Options (admin-configurable per category):

1. **Direct Publish**: No approval needed (for team-level categories)
2. **Single Approver**: Requires approval from category owner or dept admin
3. **Multi-Stage**: Draft → Peer Review → Manager Approval → Publish

Approval UI:

- **Submitter View**: "Submit for Review" button → Shows approval status (Pending, Approved, Rejected with feedback)
- **Approver View**: Queue of pending entries with preview, approve/reject buttons, comment field
- **Notifications**: Email/Slack alerts for approval requests and decisions

Prototype Details:

- Mock approval queue for manager (5-7 pending entries with metadata)
- Entry status badges (Draft, Pending Review, Approved, Published)
- Rejection feedback modal (required comment field)

Edge Cases:

- Approver unavailable (auto-escalate to backup approver after 3 days)
- Conflicting edits during approval process (lock entry during review)
- Urgent publish (emergency override button for admins with audit log)

Sub-Feature 3.1.4: Search Within KB and Filtering

KB-Specific Search:

- Search bar scoped to "Knowledge Base" (or specific category)
- Filters: Category, tags, author, date modified, version
- Sort options: Relevance, Most Recent, Most Viewed, Alphabetical

Integration with Global Search:

- KB entries appear in global search results with "KB" badge
- Quick action: "Browse this category" (navigate to KB tree view)

Prototype Details:

- KB search bar with category filter dropdown
- Results list with KB-specific metadata (category path, tags, version)

Sub-Feature 3.1.5: Role-Based Access Control

Permission Levels:

- **View**: Read-only access
- **Contribute**: Create/edit entries in assigned categories (subject to approval workflow)
- **Curate**: Approve entries, manage category (add/remove contributors)
- **Admin**: Full control (create categories, configure workflows, audit logs)

Access Rules:

- Default: Dept members have "View" on dept KB, "Contribute" on team KB
- Sensitive categories (e.g., "Executive Memos"): Explicit access list only
- Audit logs: Track all access (view, edit, approve, delete) with user, timestamp, action

Prototype Details:

- Permission matrix editor (rows = users/groups, columns = categories, cells = permission level)

- Locked entry icon (hover shows "Restricted - Request access")
- Audit log viewer (filterable by user, category, action, date)

Sub-Feature 3.1.6: Integration with Frameworks and Auto-Population

Auto-Population Scenarios:

1. **Framework Documentation**: When a framework is deployed, auto-create KB category with framework README, API docs, runbooks
2. **SaaS Product Manuals**: Sync product docs from integrated SaaS tools (e.g., Salesforce knowledge base → ATC KB category)
3. **Historical Data Migration**: Import existing wiki/Confluence/SharePoint content with category mapping

UI Flow:

- Admin selects source (Framework X, SaaS Y) → Preview KB structure (tree view) → Configure mapping (source categories → ATC categories) → Import with progress bar
- Imported entries tagged with "Imported from [Source]" for traceability

Prototype Details:

- Import wizard with source selector and mapping interface
- Progress screen (e.g., "Importing 127 of 500 entries - 25% complete")
- Preview pane showing source content → ATC KB structure

Edge Cases:

- Duplicate titles (append source name, e.g., "API Guide (Framework A)")
- Import failures (log errors, allow manual retry or skip)
- Source updates (optional continuous sync with conflict resolution)

Sub-Feature 3.1.7: Integration with EX Channels for Discussion

Channel Linking:

- Each KB entry can link to a dedicated discussion channel (auto-created or manually linked)
- Entry footer shows "Discuss in #api-docs-discussion" link
- Channel posts can reference KB entries with rich preview (title, snippet, link)

Use Cases:

- Users ask questions about KB entry → Discussion happens in channel → Top questions/answers fed back to KB as FAQs
- KB author announces updates in linked channel

Prototype Details:

- KB entry mockup with "Discuss" button (opens channel in side panel)
- Channel post with KB entry preview card (thumbnail, title, "View in KB" button)

EPIC 4: Framework/Accelerator Integration (Single Source of Truth Hub)

Description

Create a unified hub connecting **frameworks, accelerators, SaaS products, and KBs** behind a secure firewall. This EPIC addresses the core differentiator for **services firms** (consulting companies, system integrators) who build custom solutions for clients and need a centralized, AI-powered platform for employees to access all deliverables.

Key Value Proposition:

> "Services firms building accelerators, frameworks, and SaaS products for their clients need a secure home that connects all these assets (FW + KB + SaaS) into one AI intranet for employees. This is the **single source of truth** within a secure firewall, designed with AI in mind."

Objectives:

- Unify disparate client deliverables (frameworks, accelerators, custom SaaS) into one searchable, AI-accessible repository
- Ensure secure, role-based access (e.g., Client A employees can't see Client B frameworks)
- Enable AI Assistant to reason across all integrated assets (search, summarize, compare frameworks)
- Support hybrid integrations (on-prem frameworks + cloud SaaS + air-gapped KBs)
- Reduce "tool sprawl" for consulting firm employees (one login, one search, one AI)

Feature 4.1: Framework and Accelerator Hub

Sub-Feature 4.1.1: Unified Hub for Custom Deliverables

What Are Frameworks/Accelerators?

- **Frameworks**: Reusable architectures, templates, or methodologies built by services firms (e.g., "Cloud Migration Framework", "Data Governance Framework")
- **Accelerators**: Pre-built code, automation scripts, or tools that speed up implementation (e.g., "Terraform Accelerator for AWS", "ETL Pipeline Accelerator")
- **Custom SaaS**: Bespoke applications built for clients (e.g., "Client Portal", "Analytics Dashboard")

Hub Functionality:

- **Centralized Registry**: Catalog of all frameworks, accelerators, and SaaS products with metadata:
    - Name, version, description, owner (team/dept), tech stack, deployment status
    - Links to: Documentation (KB), code repos (GitHub), demo environments, support channels
- **Unified Indexing**: All framework docs, code comments, runbooks, and SaaS manuals are Elasticsearch-indexed for global search
- **AI Accessibility**: AI Assistant can query across frameworks (e.g., "Compare security features of Framework A vs B")

Prototype Details:

- Hub dashboard with framework cards (name, icon, version, "View Docs", "Access Code" buttons)
- Each card shows: Tech stack badges (e.g., Python, AWS, Docker), last updated date, user rating (5-star)
- Filter/sort options: By tech stack, by client, by team, by recency

Edge Cases:

- Framework with no documentation (show warning, prompt team to add KB entries)
- Deprecated frameworks (archive with "Deprecated" badge, hide from default view)
- Framework name conflicts (enforce unique naming or require namespace, e.g., "ClientA.CloudMigration")

Sub-Feature 4.1.2: Connection Dashboard with Integration Status

Integration Types:

1. **Code Repositories**: GitHub, GitLab, Bitbucket (index README, code comments, issues)
2. **Documentation Sites**: Confluence, Notion, custom wikis (sync to KB)
3. **SaaS APIs**: Custom client portals, analytics dashboards (index metadata, user guides)

4. **File Shares**: On-prem network drives, S3 buckets (index documents, runbooks)

Dashboard UI:

- **Connection Cards** (one per integration):
  - Integration name (e.g., "GitHub - Cloud Framework Repo")
  - Status indicator (Connected ✅, Syncing 🔄, Error ❌)
  - Sync logs (last sync time, items synced, errors)
  - Actions: "Re-sync Now", "Configure", "Disconnect"
- **Aggregated Metrics**:
  - Total integrations: 12 connected, 2 pending, 1 error
  - Items indexed: 15,234 documents, 837 code files, 124 SaaS endpoints
  - Search coverage: "98% of frameworks indexed"

Prototype Details:

- Dashboard mockup with 5-7 connection cards (varied statuses)
- Sync log modal (table with timestamp, items synced, status, error messages)
- "Add Integration" button → Wizard (select type, enter credentials, test connection)

Edge Cases:

- API rate limits hit during sync (exponential backoff, resume from checkpoint)
- Credential expiration (alert admin, pause syncing until re-auth)
- Large repo sync timeout (chunk into smaller batches, show progress)

Sub-Feature 4.1.3: Query Across Unified Content

Cross-Framework Search:

- Search query (e.g., "API authentication best practices") returns results from:
  - Framework A's authentication module documentation (KB)
  - Accelerator B's OAuth implementation code (GitHub)
  - SaaS Product C's auth API reference (auto-generated docs)
  - Company-wide security KB entry on auth standards

AI Assistant Capabilities:

- **Comparison**: "Compare authentication approaches in Framework A vs Accelerator B"
- **Recommendation**: "Which framework should I use for a HIPAA-compliant data pipeline?"
- **Code Examples**: "Show me code snippets for API rate limiting from our accelerators"

Prototype Details:

- Mock search results showing mixed sources (framework docs, code, SaaS manuals) with distinct badges
- AI Assistant chat showing cross-framework comparison in table format
- Code snippet result with syntax highlighting and "Copy" button

---

Sub-Feature 4.1.4: Conflict Resolution for Overlapping Content

Conflict Scenarios:

- Multiple frameworks have documentation with same title (e.g., "Getting Started Guide")
- Code files with identical names across repos (e.g., "config.yaml")
- Duplicate SaaS endpoints from different integrations

Resolution Strategies:

1. **Namespace Prefixing**: Automatically prepend source name (e.g., "Framework A: Getting Started")
2. **Disambiguation UI**: Show conflicts to admin, allow manual merge or keep separate
3. **Priority Rules**: Admin sets priority (e.g., "Framework A docs override Accelerator B docs for topic X")

Prototype Details:

- Conflict resolution modal (side-by-side comparison, select "Keep Both" or "Merge")
- Admin panel for priority rules (drag-drop list of sources to set precedence)

Sub-Feature 4.1.5: Security and Role-Based Access

Access Control for Multi-Client Scenarios:

- **Client Isolation**: Employees assigned to "Client A" can only search/access Client A frameworks
- **Cross-Client Teams**: Admin can grant specific users access to multiple client frameworks (e.g., architects, leadership)
- **Granular Permissions**: Per-framework access (e.g., "Engineering" team has code access, "Sales" has docs-only)

Implementation:

- Framework metadata includes "Access Group" field (e.g., "Client A Engineers")
- Search/AI results auto-filtered by user's access groups
- Audit logs for cross-client access (compliance requirement for some firms)

Prototype Details:

- Framework registry with "Access" column (shows groups with access)
- Locked framework card (hover shows "Request access - Contact [owner]")
- Audit log showing cross-client access events (user, framework, action, timestamp)

Sub-Feature 4.1.6: Admin Monitoring and Sync Management

Admin Tasks:

- **Monitor Integrations**: Health checks, sync frequency (hourly, daily, weekly), error alerts
- **Trigger Manual Syncs**: Force re-index (e.g., after major framework update)
- **Configure Sync Scope**: Include/exclude specific repos, folders, or file types (e.g., exclude test files)
- **Bandwidth Management**: Schedule syncs during off-peak hours for large datasets

UI Components:

- Integration health dashboard (traffic light indicators: green/yellow/red)
- Sync schedule calendar (visualize when each integration syncs)
- Error alert center (filterable by integration, severity, date)

Prototype Details:

- Admin dashboard with real-time integration status
- Manual sync button with confirmation dialog ("This will re-index 12,000 documents. Continue?")
- Sync log with retry button for failed items

Edge Cases:

- Sync during peak usage (throttle indexing, show "Background sync in progress" banner)
- Integration source goes offline (pause sync, alert admin, auto-resume when back online)
- Circular dependencies in sync (detect and prevent infinite loops)

EPIC 5: Role-Based Access Control (Security & Permissions)

Description

Implement robust, granular security controls to ensure users only access content appropriate for their role, department, and clearance level. Tie permissions to **org chart structures** for automatic, dynamic access management.

Objectives:

- Zero-trust architecture: Default deny, explicit grant
- Minimize admin overhead via org-chart-based inheritance (e.g., dept manager → access to all team KBs)
- Compliance-ready audit trails (SOC 2, ISO 27001, GDPR)
- Real-time enforcement across search, AI, KB, frameworks, EX

Feature 5.1: Hierarchical Role-Based Permissions

Sub-Feature 5.1.1: Permissions Matrix Linked to Org Chart

Permission Model:

- **Roles**: User, Contributor, Curator, Manager, Admin
- **Resources**: KB categories, frameworks, channels, dashboards, workflows
- **Actions**: View, Edit, Approve, Delete, Administer

Org Chart Integration:

- Permissions inherit down org hierarchy (e.g., Engineering VP → all Engineering KBs)
- Team-specific access (e.g., Backend Team → Backend framework code)
- Cross-functional access (e.g., Project Alpha team → KBs from Eng + Marketing)

Admin Tools:

- Visual permission matrix (rows = users/groups, columns = resources, cells = role)
- Bulk actions (e.g., "Grant all Engineering users 'View' on Framework X")
- Permission templates (e.g., "New Engineer" template auto-assigns standard access)

Prototype Details:

- Interactive matrix editor with drag-drop role assignment
- Org chart visualization (click dept → see aggregated permissions)
- Template library (5-7 pre-defined templates with descriptions)

Edge Cases:

- User in multiple depts (union of permissions, or allow admin to set precedence)
- Dept restructure (trigger permission review workflow)

- Contractor vs employee (flag contractors, require explicit access grants)

Sub-Feature 5.1.2: Dynamic Content Filtering

Real-Time Enforcement:

- **Search Results**: Auto-filter by user's permissions (no manual hiding)
- **AI Assistant**: Only grounds on accessible sources (shows "Some sources excluded due to permissions" if relevant)
- **KB Browse**: Tree view only shows categories user can access
- **Dashboard**: Widgets auto-populate with permitted data

Visual Indicators:

- **Locked Icon**: On restricted content in browse views
- **Badge**: "Restricted - Request Access" on search results user can see but not open (tease existence for discoverability)
- **Blur Effect**: Thumbnails/snippets blurred for no-access items (privacy)

Prototype Details:

- Search results with mixed access levels (3-4 locked items with "Request Access" button)
- AI Assistant response showing "[2 restricted sources not included]" disclaimer
- KB tree with locked folders (grayed out, lock icon, tooltip on hover)

Edge Cases:

- Permission change mid-session (refresh access token, warn user "Your permissions have changed")
- Search for restricted terms (show "No results" vs "X restricted results - Request access?")
- Shared links to restricted content (show login prompt or access denied page)

Sub-Feature 5.1.3: Access Request and Approval Workflow

User Flow:

1. User encounters restricted content → Click "Request Access" → Modal opens
2. Select reason (dropdown: "Need for project", "Joining team", "Temporary access") + free-text justification
3. System auto-routes to resource owner (e.g., KB category curator, framework team lead)
4. Owner approves/rejects → User notified via email/Slack
5. If approved → Access granted immediately (real-time token refresh)

Admin View:

- Queue of pending access requests (sortable by resource, requester, date)
- Batch approve (checkbox multiple requests)
- Auto-approve rules (e.g., "Auto-approve KB access within same dept")

Prototype Details:

- Request modal (reason dropdown, justification text area, "Submit" button)
- Owner notification mockup (email/Slack with "Approve" / "Reject" buttons)
- Admin queue (table with 5-7 pending requests, batch action toolbar)

Edge Cases:

- Request for already-accessible content (show "You already have access")
- Owner unresponsive (auto-escalate to manager after 48 hours)
- Temporary access expiration (auto-revoke after X days, send reminder before expiry)

Sub-Feature 5.1.4: Audit Logs for Compliance

Log Events:

- User login/logout, role changes, permission grants/revokes
- Content access (view, download, edit), search queries, AI Assistant queries
- Admin actions (permission changes, integration configs, index operations)

Audit UI:

- Searchable log viewer (filter by user, resource, action, date range)
- Export to CSV/JSON for external SIEM tools
- Pre-built compliance reports (e.g., "All access to PII data last quarter")

Prototype Details:

- Audit log table (columns: timestamp, user, action, resource, IP address, outcome)
- Advanced filters panel (multi-select for action types, date picker)
- Export dialog (select date range, columns to include, format)

Edge Cases:

- Log retention limits (auto-archive logs >90 days to cold storage)
- PII in logs (redact sensitive search queries, hash user IDs)
- Tamper detection (cryptographic log signing, alert on integrity violations)

Sub-Feature 5.1.5: Admin Assignment via Org Chart

Simplified Admin Workflow:

- Admin drags user from org chart to resource (e.g., drag "Jane Smith" to "Framework X" → Grant "Contributor" role)
- Dept-level assignments (e.g., drag entire "Backend Team" to framework → All members get access)
- Visual feedback (highlighted resources show who has access on org chart)

Prototype Details:

- Split-screen view (org chart on left, resources on right)
- Drag animation (ghost image of user/dept icon following cursor)
- Post-assignment confirmation toast ("Access granted to 12 users")

Edge Cases:

- Drag user to resource they already have (show "Already has access - Update role?" dialog)
- Dept with 100+ members (show warning "This will affect 127 users - Confirm?")
- Drag to incompatible resource (e.g., contractor to confidential KB → Block with error message)

EPIC 6: Custom Agentic Workflows (Automation & Orchestration)

Description

Enable users to build **custom, multi-step agentic workflows** that automate complex, repetitive tasks by orchestrating AI, search, integrations, and human approvals. Workflows are **grounded in indexed data** (KBs, frameworks, EX) and support multi-LLM reasoning.

Objectives:

- Automate business processes (e.g., "Weekly report generation from EX feeds and frameworks")
- Support agentic AI (autonomous multi-step reasoning, tool use, decision-making)
- Integrate with all platform components (search, AI, KB, frameworks, EX, SaaS)
- Provide no-code builder for non-technical users, plus code mode for developers

Use Cases:

- "Generate monthly dept summary from news feeds + KB updates + framework changes"
- "Route support questions to correct team based on org chart + KB content"
- "Auto-create KB entries from resolved support tickets (AI summarization)"
- "Trigger alerts when framework docs are out of sync with code repos"

Feature 6.1: Agentic Workflow Builder

Sub-Feature 6.1.1: Multi-Step Workflow Designer

Workflow Components:

1. **Triggers**: Schedule (cron), event (new KB entry, channel post, search query), manual (button click)
2. **Actions**:
   - **Search**: Query Elasticsearch/vector DB, return top N results
   - **AI Reasoning**: Send context to LLM, get structured output (JSON, summary, classification)
   - **Data Transform**: Parse, filter, aggregate data (e.g., extract dates, group by dept)
   - **Integration Call**: Fetch/post to SaaS APIs (e.g., Slack message, Salesforce update)
   - **Human Approval**: Pause workflow for manual review (notification + approval form)
   - **Output**: Save to KB, send email, create dashboard widget, post to channel
3. **Conditions**: If/else branching (e.g., if confidence < 80% → escalate to human)
4. **Loops**: Iterate over lists (e.g., for each search result, run AI summarization)

Workflow Designer UI:

- **Canvas**: Drag-drop flow diagram (nodes + connectors)
- **Node Types**: Trigger (green star), action (blue rectangle), condition (yellow diamond), output (red circle)
- **Properties Panel**: Configure each node (e.g., search query, LLM prompt, condition logic)
- **Test Mode**: Run workflow with sample data, visualize execution path

Prototype Details:

- Canvas with 5-7 node example workflow (trigger → search → AI summarization → approval → post to channel)
- Node property panel (show search query input, LLM model selector, prompt template)
- Execution visualization (animated flow showing active node, completed nodes in green, current node pulsing)

Edge Cases:

- Infinite loop detection (max iterations limit, warn user "Loop detected at node X")
- Node execution timeout (configurable per node, fail gracefully with error log)
- Missing required fields (prevent workflow save, highlight incomplete nodes in red)

Sub-Feature 6.1.2: Grounding in Indexed Data and Multi-LLM Support

Grounding Mechanisms:

- **Search Node**: Returns structured results (title, snippet, URL, relevance score) that feed into AI nodes
- **Context Injection**: AI nodes automatically include search results, KB entries, framework docs in prompts
- **Citation Tracking**: Workflow outputs include citations (e.g., "Based on [KB Article], [Framework X Docs]")

Multi-LLM Routing:

- Each AI node can specify LLM model (GPT-4, Claude, custom)
- Auto-routing based on task (e.g., code generation → specialized model, summarization → fast model)
- Cost optimization (use cheaper models for simple tasks, expensive models for complex reasoning)

Prototype Details:

- Search node showing mock results (3-4 results with metadata)
- AI node with prompt template editor (include variables like `{search_results}`, `{user_dept}`)
- LLM selector dropdown in AI node properties (show model name, cost per 1K tokens, latency estimate)

Sub-Feature 6.1.3: Template Library and Pre-Built Workflows

Template Categories:

1. **Reporting**: Weekly dept summary, monthly framework usage stats, KB contribution leaderboard
2. **Support Automation**: Route questions to teams, auto-suggest KB articles, escalate unresolved queries
3. **Content Management**: Auto-tag new KB entries, detect outdated docs, suggest merges
4. **Compliance**: Flag sensitive data in channels, audit framework access, generate access reports

Template UI:

- Gallery view (cards with template name, description, preview diagram, "Use Template" button)
- Each template includes pre-configured nodes and sample data
- User can clone, modify, and save as custom workflow

Prototype Details:

- Template gallery with 6-8 templates (varied use cases)
- Preview modal showing full workflow diagram (read-only)
- "Use Template" → Opens in workflow designer (editable)

Edge Cases:

- Template depends on unavailable integration (show warning "Requires Slack integration - Enable in settings")
- Template with deprecated nodes (auto-migrate to new node types, notify user)

Sub-Feature 6.1.4: Execution, Branching, and Error Handling

Execution Flow:

1. Trigger fires (schedule, event, manual) → Workflow starts
2. Each node executes sequentially (or in parallel for independent branches)
3. Condition nodes evaluate (true/false) → Branch to corresponding paths
4. Human approval nodes pause execution → Send notification → Resume on approval
5. Workflow completes → Generate execution log and outputs

Branching Logic:

- **If/Else**: Based on node output (e.g., if AI confidence > 80% → auto-post, else → human review)
- **Switch/Case**: Multiple branches based on classification (e.g., route query to Dept A/B/C based on topic)

Error Handling:

- **Retry Logic**: Auto-retry failed nodes (configurable: 3 retries with exponential backoff)
- **Fallback Paths**: Define alternative actions on failure (e.g., if AI fails → use rule-based logic)
- **Alerts**: Email/Slack notification for workflow failures (include error details)

Prototype Details:

- Execution log viewer (table with node name, status, duration, output preview, error message if failed)
- Branching diagram showing active path highlighted (e.g., "Condition = True → Took left branch")
- Error node with retry button and "View Error Details" link

Edge Cases:

- Partial workflow failure (some nodes succeed, others fail → Mark workflow as "Partially Completed")
- Approval timeout (if no response in X hours → Auto-reject or escalate)
- Concurrent executions (if same workflow triggered twice → Queue or run in parallel based on config)

Sub-Feature 6.1.5: Structured Outputs and Role-Based Execution

Output Formats:

- **Tables**: Structured data (CSV, Excel) for reports
- **Summaries**: Rich text documents (PDF, markdown) for KB entries or emails
- **Dashboards**: JSON data to populate dashboard widgets
- **Notifications**: Formatted messages to Slack, email, EX channels

Role-Based Execution:

- Workflows can be configured to run with specific dept/team context (e.g., "Run as Engineering dept" → Only access Engineering KBs and frameworks)
- User-triggered workflows inherit user's permissions (security enforcement)
- Scheduled workflows run as admin (or designated service account) with explicit permissions

Prototype Details:

- Output node with format selector (dropdown: Table, Summary, Dashboard Widget, Notification)
- Preview pane showing sample output (e.g., mock table with 5 rows, summary with 3 paragraphs)
- Execution context selector (admin-only: "Run as [User/Dept/Admin]")

Edge Cases:

- Output too large (e.g., 10,000-row table → Paginate or split into multiple files)
- Insufficient permissions for output target (e.g., can't post to restricted channel → Fail with error)

- Output schema mismatch (e.g., dashboard widget expects specific fields → Validate before execution)

Sub-Feature 6.1.6: No-Code Builder + Code Mode

No-Code Builder (default):

- Visual drag-drop interface (as described above)
- Form-based node configuration (dropdowns, text inputs, toggles)
- Suitable for business users, non-developers

Code Mode (advanced):

- Switch to YAML or JSON workflow definition
- Full control over logic (e.g., custom Python/JavaScript snippets in action nodes)
- Version control integration (commit workflows to Git)

Prototype Details:

- Toggle button "Visual Mode / Code Mode" in workflow designer
- Code editor with syntax highlighting (YAML/JSON)
- Live preview: Edit code → Visual diagram updates in real-time

Edge Cases:

- Code syntax error (highlight line, show error message)
- Code mode creates invalid workflow (validation on switch back to visual mode)
- Complex workflows not fully representable in visual mode (show warning "Some advanced features hidden in visual view")

Screenshot References *(Extract from Glean video, adapt for ATC)*

Show Image *Reference: Agent dashboard (adapt as workflow list)*

Show Image *Reference: Agent creation UI (adapt as workflow designer canvas)*

Show Image *Reference: Agent execution (adapt as workflow execution log)*

EPIC 7: Central Dashboard (Admin vs Employee Views)

Description

Provide **role-differentiated dashboards** that serve distinct needs: **Admin dashboards** for system monitoring, usage analytics, and configuration; **Employee dashboards** for personalized daily summaries, tasks, and quick access to relevant content.

Objectives:

- Admin: Real-time platform health, usage insights, actionable alerts
- Employee: Personalized, context-aware landing page that aggregates EX feeds, tasks, KBs, frameworks
- Both: Customizable widgets, responsive design, real-time data

---

Feature 7.1: Admin Dashboard

Sub-Feature 7.1.1: System Health and Monitoring

Metrics Displayed:

- **Indexing Health**:
  - Elasticsearch cluster status (green/yellow/red)
  - Total documents indexed, indexing rate (docs/sec), index size (GB)
  - Failed indexing jobs (count, most recent error)
- **Search Performance**:
  - Query latency (p50, p90, p99 over last 24 hours)
  - Search volume (queries/hour, top queries)
  - Zero-result queries (count, list for analysis)
- **AI Assistant Usage**:
  - Total queries, resolution rate (% answered without escalation)
  - Average confidence score, LLM model usage breakdown
  - Escalations to human (count, reasons)
- **User Activity**:
  - Active users (daily, weekly, monthly)
  - Most accessed KBs, frameworks, channels
  - Login failures, permission denied events

UI Components:

- **Status Cards**: Large tiles with traffic light indicators (green/yellow/red) and primary metric (e.g., "Search Latency: 124ms ✅")
- **Charts**: Line graphs (usage over time), bar charts (top queries), pie charts (LLM model distribution)

- **Alerts Panel**: Red banner for critical issues (e.g., "Elasticsearch cluster unstable - 2 shards unassigned")

Prototype Details:

- Dashboard mockup with 6-8 status cards (varied metrics)
- Interactive chart (hover shows exact values, click drills down to details)
- Alerts panel with "Dismiss" and "View Details" buttons

Edge Cases:

- Metric collection failure (show stale data warning "Data as of [timestamp] - Refresh failed")
- Too many alerts (collapse into "5 more alerts - Expand to view")
- Admin on mobile (simplified view, critical metrics only)

Sub-Feature 7.1.2: Usage Analytics and Insights

Analytics Views:

1. **User Engagement**:
    - Heatmap: Active hours (day of week × hour of day)
    - Adoption funnel: Users registered → Active last 7 days → Power users (daily active)
    - Feature usage: % using AI Assistant, workflows, KB, search
2. **Content Analytics**:
    - Most viewed KB entries, frameworks, channels
    - Contribution rates: KB entries created, edited, approved
    - Stale content alerts: Documents not updated in 6+ months
3. **AI Performance**:
    - Resolution rate trend (line chart over 3 months)
    - Common failure modes: Low-confidence responses, missing sources, timeouts
    - LLM cost analysis: Tokens consumed per model, cost per query

Export and Reports:

- Export any chart as PNG/CSV
- Schedule automated reports (weekly email with key metrics)
- Custom date range selection

Prototype Details:

- Tabbed interface (User Engagement, Content Analytics, AI Performance)
- Drill-down capability (click bar in chart → See individual users or documents)

- Export button with format selector (PNG, CSV, PDF)

Edge Cases:

- No data for date range (show empty state "No data available for selected range")
- Report generation timeout (run in background, email when ready)
- Large exports (split into multiple files if >10,000 rows)

Sub-Feature 7.1.3: Quick Admin Actions

Action Shortcuts (buttons on dashboard):

- "Trigger Full Re-index" (with confirmation dialog)
- "Restart AI Assistant" (for config changes)
- "Review Access Requests" (badge shows pending count)
- "View Recent Errors" (opens log viewer)
- "Configure Integrations" (shortcut to integration dashboard)

Prototype Details:

- Action button toolbar (horizontal row at top of dashboard)
- Confirmation dialog for destructive actions (e.g., re-index warns about downtime)
- Badge notifications on buttons (e.g., "3" on "Access Requests")

Feature 7.2: Employee Dashboard

Sub-Feature 7.2.1: Personalized Landing Page

Dashboard Widgets (customizable layout):

1. **My Tasks**: To-do list from Productivity Assistant (upcoming deadlines, pending approvals, assigned workflows)
2. **News Feed Summary**: Latest posts from subscribed EX channels, filtered by dept/interests
3. **Org Updates**: Announcements, events, policy changes relevant to user's dept
4. **Recent Activity**: User's recent searches, KB entries viewed, frameworks accessed
5. **Quick Links**: Bookmarked KBs, frameworks, channels (user-curated)
6. **AI Assistant Shortcut**: Chat input box for quick queries
7. **Trending Content**: Most viewed KBs/frameworks in user's dept this week

Layout Options:

- Grid layout (2-3 columns, draggable widgets)
- Preset layouts (select from "Task-Focused", "News-Heavy", "Minimal")
- Responsive (auto-adjust on mobile: single column, collapsed widgets)

Prototype Details:

- Dashboard mockup with 5-6 widgets (varied data)
- Drag handle on each widget (move to reorder)
- Settings icon (click → Configure widget: Show/hide, adjust size, set filters)

Edge Cases:

- No data for widget (show empty state "No tasks for today - Great job!")
- Widget data loading slowly (show skeleton loader, timeout after 5 seconds with error)
- Too many widgets (limit to 10, hide extras in "More" menu)

Sub-Feature 7.2.2: Auto-Population Based on Role and Org Context

Personalization Logic:

- **Role**: Managers see team metrics widget (team activity, pending approvals)
- **Dept**: Engineering sees framework updates, Sales sees CRM integration news
- **Projects**: User assigned to "Project X" → Widget showing project-specific channels, KBs, frameworks
- **Activity**: Frequent KB viewer → Prioritize "Recent KB Updates" widget

Admin Configuration:

- Default widget templates per role (e.g., "New Engineer" gets onboarding tasks + training KBs)
- Dept admins can set recommended widgets for their teams

Prototype Details:

- Role-specific dashboard mockups (3 personas: Employee, Manager, Exec)
- Admin panel for default widget configuration (drag widgets to template, set filters)

Edge Cases:

- User with multiple roles (show union of recommended widgets, let user prune)
- Role change (auto-update dashboard, notify user "Your dashboard has been updated")
- Conflicting recommendations (prioritize based on primary role)

Sub-Feature 7.2.3: Widget Customization and Curation

User Controls:

- **Add Widget**: Button opens widget library (browse by category: Tasks, News, Analytics, Quick Access)
- **Configure Widget**: Click settings icon → Set filters (e.g., "News Feed: Show only #engineering-channel")
- **Resize**: Drag widget edges to adjust size (small, medium, large)
- **Remove**: X button (confirm "Remove widget?")

Widget Library:

- Categories: Tasks, News & Feeds, Content Discovery, AI & Search, Analytics, Custom (user-created)
- Preview: Click widget in library → See demo with sample data
- One-click add: Click "Add to Dashboard" → Widget appears in default position

Prototype Details:

- Widget library modal (grid of 10-12 available widgets with descriptions)
- Configuration modal (show filter options, preview updated widget)
- Dashboard in edit mode (dashed borders, drag handles visible)

Edge Cases:

- Widget conflict (e.g., two "AI Assistant" widgets → Warn "You already have this widget")
- Invalid widget config (e.g., filter for non-existent channel → Show error on widget)
- Dashboard reset (button to restore default layout with confirmation)

Sub-Feature 7.2.4: Integration with Productivity Assistant

Seamless Handoff:

- Query in AI Assistant widget → If complex, prompts "Would you like to curate a full dashboard for this?"
- Productivity Assistant analyzes user activity → Suggests new widgets (e.g., "You view Security KB often - Add 'Security Updates' widget?")

Daily Curation:

- On login, Productivity Assistant auto-updates "My Tasks" widget (prioritize based on deadlines, importance)
- "Daily Brief" widget: AI-generated summary of personalized content (e.g., "3 new KBs in your dept, 1 framework update, 5 feed posts")

Prototype Details:

- AI Assistant widget with curation prompt (button "Curate My Dashboard")
- Daily Brief widget showing AI-generated summary (3-5 bullet points with expand/collapse)
- Suggestion toast notification (bottom-right corner: "Suggested widget: Security Updates - Add?")

Sub-Feature 7.2.5: Real-Time Updates and Notifications

Live Data:

- Widgets auto-refresh at configurable intervals (default: 5 minutes)
- Real-time updates for critical data (e.g., new task assigned → "My Tasks" widget updates immediately)
- WebSocket-based for low-latency (fallback to polling if WebSocket unavailable)

Notifications:

- In-widget badges (e.g., "3 new feed posts" on News Feed widget)
- Browser notifications for high-priority (e.g., "Approval required for workflow X")
- Notification center (bell icon in header, dropdown list of recent alerts)

Prototype Details:

- Widget with badge indicator (red circle with count: "5")
- Notification center dropdown (5-7 recent notifications with timestamps, "Mark all as read" button)
- Real-time update animation (widget content fades out, new content fades in)

Edge Cases:

- Too many notifications (batch into summary: "10 new updates - View all")
- Notification for deleted content (e.g., channel post deleted → Remove notification)
- User on slow connection (disable real-time updates, show "Refresh to update" message)

EPIC 8: Productivity Assistant (Personal Employee AI)

Description

The **Productivity Assistant** is a dedicated AI agent for each employee, designed to curate daily tasks, summarize relevant content, and streamline workflows. Unlike the general AI

Assistant (EPIC 2), which answers ad-hoc queries, the Productivity Assistant **proactively** manages the employee's work context.

Objectives:

- Reduce information overload (auto-curate from EX feeds, channels, KBs, frameworks)
- Surface high-priority tasks and deadlines (from workflows, approvals, calendar)
- Provide personalized daily briefs and summaries
- Enable conversational task management (e.g., "Add [task] to my list", "Remind me about [X] tomorrow")

Feature 8.1: Personalized Task Management

Sub-Feature 8.1.1: Auto-Curated Task List

Task Sources:

- **Workflows**: Tasks assigned from custom agentic workflows (e.g., "Review KB entry", "Approve framework deployment")
- **Approvals**: Pending approvals for KB entries, access requests, workflow outputs
- **Calendar**: Events from integrated EX calendars (meetings, deadlines, reminders)
- **Channel Mentions**: Posts in EX channels where user is @mentioned
- **AI-Detected**: Productivity Assistant scans user activity and suggests tasks (e.g., "You viewed 'Project X KB' 5 times this week - Create summary?")

Task Prioritization:

- **Algorithm**: Deadline proximity + importance (user-set or AI-inferred) + dept urgency
- **Visual**: Tasks color-coded (red = overdue, yellow = due today, green = upcoming)
- **Sort Options**: By deadline, priority, category (workflows, approvals, calendar)

Prototype Details:

- Task list widget (5-7 tasks with checkboxes, title, due date, priority icon)
- Task detail modal (click task → Show description, source, related links, "Mark Complete" button)
- Empty state: "No tasks for now - You're all caught up! 🎉"

Edge Cases:

- Task with no deadline (default to low priority, sort to bottom)
- Overdue tasks (persist until manually dismissed, show "Overdue by X days")
- Completed tasks (move to archive, accessible via "View Completed" toggle)

Sub-Feature 8.1.2: Conversational Task Management

Natural Language Commands:

- "Add 'Review Q3 report' to my tasks" → Creates task with today's date
- "Move 'Submit proposal' to Friday" → Updates task deadline
- "Remind me about the engineering sync tomorrow" → Creates reminder task
- "What's my top priority today?" → AI Assistant answers based on task list

Voice Input:

- Click microphone icon in task widget → Speak command → AI processes and confirms action
- Example: "Add task: Call vendor about contract renewal" → AI creates task and reads back confirmation

Prototype Details:

- Task widget with chat input bar (text or voice)
- Confirmation toast (e.g., "Task added: Review Q3 report - Due: Today")
- Failed command handling (e.g., "I didn't understand. Did you mean: [Suggestion]?")

Edge Cases:

- Ambiguous date (e.g., "Friday" when today is Thursday vs next Thursday → Ask for clarification)
- Voice recognition error (show text preview, allow edit before confirming)
- Duplicate task detection (warn "Similar task exists: [Title] - Add anyway?")

Sub-Feature 8.1.3: Integration with Calendar and Workflows

Calendar Sync:

- Import events from EX calendar (or external: Google Calendar, Outlook)
- Auto-create tasks for meetings with action items in description (AI extracts: "Meeting notes mention 'Follow up with client' → Create task")
- Reminder notifications (15 min before meeting, customizable)

Workflow Integration:

- When workflow assigns task → Appears in Productivity Assistant with context (link to workflow, related docs)

- Task completion triggers workflow continuation (e.g., user marks "Review doc" complete → Workflow proceeds to next step)

Prototype Details:

- Calendar event card in task list (show time, attendees, location)
- Workflow task with badge "From Workflow: Weekly Report Generation"
- Task completion confirmation (checkbox animation, confetti for major milestones)

Edge Cases:

- Calendar sync conflict (event deleted in calendar but task remains → Prompt "Event deleted - Remove task?")
- Workflow task can't be completed (missing prerequisite → Show blocker with instructions)
- Recurring calendar events (option to create recurring task vs one-time)

Feature 8.2: Daily Brief and Content Curation

Sub-Feature 8.2.1: AI-Generated Daily Summary

Daily Brief Content:

- **Today's Highlights**: Top 3-5 priority tasks, upcoming meetings, overdue items
- **Dept News**: Summary of latest EX feed posts in user's dept (AI-generated bullet points)
- **Org Updates**: Company-wide announcements, policy changes, exec messages
- **Framework & KB Updates