Product: Intercom Fin AI Agent	Owner: Aman Dixit
Development POC: Aman Dixit	Design POC: Aman Dixit
Marketing POC: Aman Dixit	Last Version Edited: April 06, 2026

Technical PRD v1.0: Intercom Fin AI – The RAG-Powered Resolution Engine

1. Product Overview & Problem Statement

AI support agent + ops console that turns fragmented SaaS support knowledge into safe, permission-aware resolutions.

Specific User Persona

- **Primary Owner:** AI Operations Manager / CX Technical Lead at Series B–D SaaS (200–1,000 employees).
- **Primary "human in the loop":** Support agents and leads working in Intercom Inbox.
- **Org Context:** Sits between Support and Product, understands flows, data, and permissions but is not a full-time engineer.

Current Workflow vs. Pain Points (Before State)

- Tickets arrive in Intercom (Messenger / Email), agents manually read and interpret vague queries.
- They context-switch across Intercom Articles, Notion/Confluence, Slack, internal admin tools, and macros to construct answers.
- They manually choose between overlapping macros, often unsure which applies to this user's plan, version, or region.
- They synthesize responses by copy-paste, trying not to leak internal links, and ping Slack (#support-eng) for verification.
- They tag and escalate tickets by hand, and errors creep in because documentation is outdated or fragmented.
- Result: slow responses, inconsistent quality, leaked edge cases, and growing "knowledge debt."

Why RAG?

- Pure prompting cannot fit hundreds of articles and policies in-context and suffers from "lost in the middle" issues in long prompts.
- Pure fine-tuning bakes knowledge into weights, making per-tenant isolation, GDPR "right to be forgotten," and rapid policy updates impractical.
- RAG separates reasoning (LLM) from knowledge (indexed docs + live APIs):
    - Allows citations and answer vulnerability

- Supports instant updates via re-indexing specific docs.
- Enforces permission-aware retrieval per tenant and user.
- Enables hybrid static knowledge (articles) + dynamic context (billing, feature flags) via tool calls.

2. User Stories & Use Cases

User Stories

- **As an AI Operations Manager,** I want to configure which knowledge sources Fin can use, so that answers stay accurate and on brand.
- **As a Support Agent,** I want Fin to draft answers with citations in my Inbox, so that I can resolve tickets faster with confidence.
- **As an admin-end user,** I want Fin to answer billing and feature-availability questions based on my account, so that I don't need to wait for support.
- **As a support-lead,** I want Fin to avoid using low-trust sources like Slack directly, so that customers never see half-baked or speculative guidance.
- **As a security-compliance owner,** I want Fin to respect per-tenant, role, and plan permissions at retrieval time, so that private information is never exposed.
- **As a CX-leader,** I want to track automated resolution rate and CSAT, so that I can justify Fin's ROI.
- **As a support engineer,** I want internal-only SOPs surfaced as internal notes, so that juniors can handle complex cases safely.
- **As a billing owner,** I want to cap LLM costs per resolution, so that gross margins stay above target.

Primary Use Cases

1. **Self-service FAQ resolution (end user):**
    - **Happy Path:** End-user asks, "How do I reset my password?" → RAG retrieves public article → Fin answers with 1–2 steps + citation → user confirms and conversation ends.
    - **Edge Case:** No matching article → Fin explicitly says it cannot find up-to-date instructions and routes to a human with a summarized internal context snapshot.

2. **Complex Policy + context resolution (end user):**
    - **Happy Path:** User asks, "My trial ended yesterday; can I extend it to test Series?" → RAG fetches trial policy + tool call gets trial_end_date → Fin responds with policy-compliant, personalized extension/no-extension answer, with clear rationale and citation.
    - **Edge Case:** Policy document ambiguous or missing → Fin explains limitation, sets expectation, and routes to human with recommended macros and internal doc references.
3. **Agent Co-pilot Drafting (internal agent)**
    - **Happy path:** Agent opens complex conversation about a Shopify + VAT issue → Fin pulls integration + tax docs, drafts an internal-note answer with references, and suggests the final customer-facing reply.
    - **Edge case:** Conflicting docs (old version vs new) → Fin marks answer low confidence, highlights conflicting sources, and asks agent to choose or escalate.

Negative Use Cases (NON GOALS)

- Fin **will not** autonomously execute irreversible financial or account changes (refunds, deletions, plan downgrades) without explicit agent confirmation or pre-configured limits.
- Fin **will not** use unvetted sources (Slack, RFCs, old tickets) directly in customer-facing answers; these are reserved for routing and internal inspiration only.

3. Technical Architecture (PM-Level)

Pipeline: High Level Flow

1. **Data Ingestion Layer:**
    - Connectors for: Intercom Articles (HTML/Markdown), public PDFs, Notion/Confluence, CRM/admin DB (via APIs), past conversations.
    - For each document: normalize, parse, optional OCR (for complex PDFs), generate semantic metadata (topic, product, audience, visibility, tenant).
2. **Chunking & Embedding Layer:**
    - Chunk docs using a semantic header-aware approach (recursive character splitting around headings, bullets, and sections).
    - Generate embeddings per chunk using a cost-efficient embedding model; attach metadata including tenant_id, visibility, roles, plan, and source_type.

- Store chunks in a multi-tenant vectorDB (e.g., Pinecone, pgvector, Chroma) with strict namespace/metadata filtering.

3. **Query Processing & Retrieval Layer**
    - On each user message:
        - Safety + intent classification (small "router" model).
        - Query refinement (rephrase, clarify target entity, detect locale).
        - Build a vector query with pre-filter: tenant_id + visibility + role/plan filters.
        - Retrieve top N (e.g., 40) fast; rerank to top K (5–10) via a stronger reranked model.
4. **Dynamic Context & Tool Layer**
    - If query requires fresh data (billing, feature flags, usage), call registered tools: billing API, CRM, admin DB, Intercom CDAs.
    - Inject structured tool outputs alongside retrieved chunks into the prompt.
5. **LLM Generation Layer**
    - Use "Needle" model for most answers; if low confidence or high complexity, escalate to "Sword."
    - System prompt enforces use only provided context, include citations, say "I don't know" when unsupported, and respect visibility rules.
6. **Response post-processing**
    - Validate answer against constraints (hallucination heuristics, policy keywords, missing citations).
    - For end-users: render chat answer with citation UI.
    - For agents: optionally show source snippets + suggested macros and tags.
7. **Observability & Logging**
    - Structured Logs per step: query, refined query, retrieved chunks, tools used, model version, latency, cost, confidence.
    - Integrate with LangSmith/Arise/Datadog – style traces and an internal eval dashboard.

Chunking Strategy

- **Default: semantic-header based recursive chunking (e.g., 400-800 tokens)** to keep sections coherent (e.g., whole step-by-step procedures).
- **Justification:**
    - **Pure f**ixed-size chunks risk cutting steps mid-procedure; header-aware improves multi-step instructions.

    - **Overly** large chunks hurt retrieval precision and inflate tokens.

- **Special Rules:**
    - **For** policy docs, align chunks with policy clauses and tables.
    - **For** internal SOPs, preserve "Notes/Warnings" within the same chunk as the procedure they qualify.

Model Selection (Sword Vs Needle)

- **Needle (default): G**PT-4o mini / Claude Haiku / Gemini Flash (configurable).
    - **U**sed for: intent detection, initial retrieval queries, FAQ answers, short responses.
- **Sword (on demand): GPT-40 / Claude Sonnet / Gemini Pro**
    - Triggered by low confidence (<0.7), multi-doc synthesis, dynamic + policy questions, VIP User Segments.
- **Routing** logic is explicit and tunable per workspace (AI Ops Manager can bias toward cost or quality).

Observability

- **Every query emits:**
    - **User/tenant** anonymized IDs, intent, model path (Needle vs Sword), steps taken, latency, token usage.
    - **Retrieval diagnostics:** number of docs, sources, coverage score.
    - **Evaluation hooks:** automatic offline eval against golden dataset for a random sample.
- **Integration Targets:**
    - **LLM Trace Tool** (e.g., LangSmith, Arize) for pipeline-level traces and evals.
    - Datadog / internal metrics for latency, error, cost, and volume.

Product: Intercom Fin AI Agent
Development POC: Aman Dixit
Marketing POC: Aman Dixit

Owner: Aman Dixit
Design POC: Aman Dixit
Last Version Edited: April 06, 2026

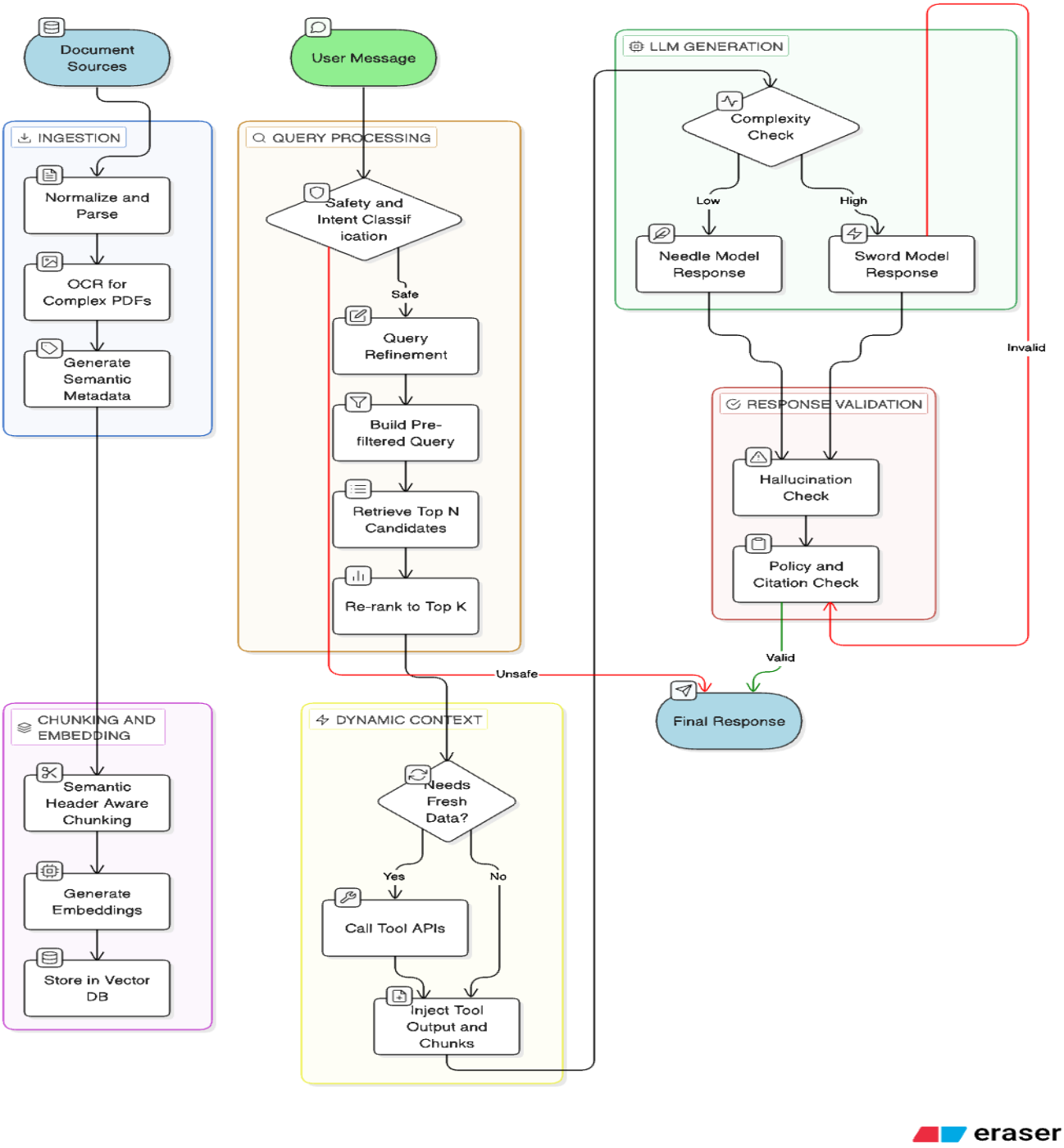

Figure: Architectural Diagram

Product: Intercom Fin AI Agent
Development POC: Aman Dixit
Marketing POC: Aman Dixit

Owner: Aman Dixit
Design POC: Aman Dixit
Last Version Edited: April 06, 2026

4. RAG Failure Points & Mitigation

For each of the 9 points: what breaks, how we detect, what we fix as PM/spec.

1. **Data Quality/OCR**
    - **Breaks**: Mis parsed PDFs (missing tables, broken headings), OCR errors create garbage chunks, "ghost" outdated policies remain indexed.
    - **Detection**:
        - Ingestion-time validation (sample pages vs source).
        - Heuristics: unusually low embedding norms, high OCR error rate, or high user "bad answer" feedback clustered by source.
    - **Fix:**
        - Require visual/structural tests for new parsers.
        - Mark certain PDFs as "human-reviewed only" or exclude them from v1.
        - Introduce "source quality" score and expose to AI Ops Manager.
2. **Chunking Strategy**
    - **Breaks:** Procedure split across chunks; answer uses only half of necessary instructions; missing deprecation notes.
    - **Detection:**
        - Golden eval failures on multi-step and negative-constraint questions.
        - LLM-Judge scoring low on completeness.
    - **Fix:**
        - Tune chunk sizes and overlap strategy; require that any "Warning/Note" near a step stays in same chunk.
        - Add rule-based sections for common doc patterns ("Limitations," "Deprecated").
3. **Embedding Quality**
    - **Breaks:** Semantically similar questions not retrieved; synonyms ("round-robin" vs "load balancing") missed.
    - **Detection:**
        - Low recall on golden questions where correct chunk is known.
        - Analyze nearest neighbors for canonical queries, spot semantic misses.
    - **Fix:**
        - Upgrade embeddings model or domain-adapt via contrastive training.
        - Add synonym/alias dictionaries and explicit keyword fields to metadata.

4. **Search/Retrieval**
    - **Breaks:** Relevant docs not in top N; over-retrieval of generic FAQ when specialized doc exists.
    - **Detection:**
        - Compare ground-truth doc vs retrieved set for eval queries.
        - High "I already tried that" feedback from users.
    - **Fix:**
        - Adjust ranking between recency, authority, and semantic score.
        - Introduce per-source priority (Articles > PDFs > past tickets).
        - Add filter knobs in the AI Ops UI.
5. **Re-ranking Failures**
    - **Breaks:** Correct chunk in top 40 but not in top 10; model fixates on wrong but semantically similar chunk.
    - **Detection:**
        - Offline Eval on reranker.
        - A/B test reranking models with golden dataset.
    - **Fix:**
        - Train or fine-tune reranker on domain-specific pairs.
        - Include additional features (source authority, doc recency, user segment).
6. **Prompt/Augmentation**
    - **Breaks:** Over-stuffed context, confusing instructions, missing constraints (“only answer if grounded”).
    - **Detection:**
        - Hallucination-type feedback and golden eval "unfaithful but plausible" answers.
        - Prompt ablation tests.
    - **Fix:**
        - Separate system vs context vs instructions.
        - Hard constraint: “Use only provided snippets; if insufficient, say you cannot answer and escalate.”
        - Enforce citations per answer section.
7. **Model Quality**
    - **Breaks:** Model misinterprets instructions, struggles with multi-doc reasoning, or misuses tools.
    - **Detection:**
        - Compare models on golden set.

    - Track hallucination rate and faithfulness metrics per model.
  - **Fix:**
    - Switch or upgrade models; adjust router thresholds.
    - Use tool-focused training examples for better tool usage adherence.
8. **Data Drift:**
  - **Breaks:** Answers reflect old pricing/features; policies changed but old vectors remain; new product areas under-documented.
  - **Detection:**
    - Drift monitors: time-based mismatch between document last updated and retrieval hit rates.
    - Spike in user corrections or agent edits on specific topics.
  - **Fix:**
    - Near-real-time re-index on article edit/publish webhooks.
    - Require product teams to update docs before new feature rollouts (doc gating).
    - Automatic deprecation of old chunks on version change.
9. **User Behavior (Non-Technical):**
  - **Breaks:** Vague queries (“It’s broken”), multi-intent questions, or prompt injection attempts.
  - **Detection:**
    - Router model flags low clarity or multiple intents.
    - Safety guardrails detect prompt injection patterns.
  - **Fix:**
    - Ask clarifying questions when intent is low confidence.
    - Split multi-intent queries into sub-questions.
    - Reject unsafe requests with a friendly refusal.

5. Data Permissions and Access Control

Permission Model

- **Hybrid-Inherited Model:**
  - Inherit visibility from source systems (Intercom articles, app RBAC, CDAs).
  - Augment with Fin-specific metadata (e.g., "developer_only," "internal_only," "vip_only").

Implementation (Metadata Filtering in VectorDB)

- **Every Chunk gets metadata:**
  - tenant_id, source_type, visibility (public, customer_internal, agent_internal), required_role(s), required_plan(s), locale, version.
- **Retrieval query includes mandatory filters:**
  - WHERE tenant_id = <tenant> AND visibility ,àà allowed_visibilities AND required_plan ,àà user_plans AND required_role ,àà user_roles.
- **Internal vs External:**
  - Customer-facing Fin sees only public + customer_internal.
  - Agent copilot sees internal SOPs, but customer never does; answers rendered into internal notes instead of chat.

Edge Cases

- **Shared docs changed from Internal → Public:**
  - Watcher process re-indexes metadata quickly; chunk permissions updated within minutes.
- **Offboarding agents:**
  - Agent identity authenticated via SSO (Okta/SAML); when revoked, agent cannot access Fin or its internal note view.
- **Plan downgrade:**
  - When plan changes, permission metadata for that user's queries narrows; Fin stops retrieving enterprise-only chunks and instead suggests upgrade flows.

6. Evaluations & Success Metrics

RAG Metrics

- Precision@K on golden dataset: does the correct chunk appear in top K?
- Faithfulness score (LLM-as-a-judge or rubric): 1-5 scale vs ground-truth answers.
- Hallucination rate: % of answers that introduce unsupported facts or violate grounding/citation rules.

Product: Intercom Fin AI Agent	Owner: Aman Dixit
Development POC: Aman Dixit	Design POC: Aman Dixit
Marketing POC: Aman Dixit	Last Version Edited: April 06, 2026

Business Metrics

- Automated Resolution Rate: % conversations resolved without human intervention (target >=50%).
- CSAT for AI-handled conversations: target >= 90%.
- Average Handle Time reduction for agents when Fin assists.
- Gross margin on Fin resolutions: COGS per resolved query <= $0.15 against $0.99 billing.

Escalation Design

- Low-confidence detection (confidence < threshold, missing doc, or ambiguous policies) triggers:
  - Honest "I'm not sure" message to end-user.
  - Automatic escalation to agent, with user query, retrieved context, and Fin's draft internal note.
- Agents can mark answers as "wrong," "incomplete," or "outdated," feeding into content and retrieval evals.

7. Token Economics & Cost Analysis

Per-Query Estimate

- Assume ~2.5 tokens/word, typical RAG answer ~ 200-400 words + context.
- With retrieval + generation, typical query:
  - Needle: ~1-3K input tokens + 400-800 output tokens.
  - Sword (when used): ~3-5K input + 600-1,000 output.
- Target COGS per resolution (including retries and follow-ups): <= $0.15.

MAU/Volume Cost

- Year-1 assumptions: 50,000 MAUs, 1.5 queries/MAU/month - 75,000 queries/month.
- Hybrid Routing:
  - 85-90% of queries resolved by Needle-only path.
  - 10-15% escalated to Sword or human.
- With caching + small-model routing, we design cost curves to keep monthly LLM spend substantially below revenue from $0.99/resolution pricing, maintaining >=85% gross margin.

Optimization Strategies

- **Semantic caching:** When a new query is close to a cached FAQ, reuse stored answer + sources with minimal regeneration.
- **Prompt caching:** Cache system + static instructions context with providers that support it to cut repeated tokens.
- **Small-to-large routing:** Tiny classifier model decides if query is simple; simple ones stay on Needle.
- **Rate-limit Sword usage:** Per-tenant caps and thresholds tunable via AI Ops UI.

8. UX, Rollout & GTM

Interface

- **End-User:**
  - In-app messenger and email replies.
  - Chat-style UI with inline citations (hover or click to see source).
  - Low-confidence states explicitly phrased ("I'm not fully sure; let me connect you").
- **Agent:**
  - Inbox sidebar or inline assistant: answer suggestions, internal notes, recommended macros, and tags.

Rollout Phases

1. **Alpha (Dogfooding)**
   - Internal support team only; limited sources (Tier 1 Articles + Tier 3 Admin API) for one internal product.
   - Heavy logging, manual review on every AI answer.
2. **Beta (Limited Data / Customers)**
   - Select customers (growth-stage SaaS) with relatively clean article bases.
   - Enable end-user automation for low-risk categories (FAQ, basic setup); others as agent copilot only.
3. **GA**
   - Open to all Intercom workspaces.
   - Packaging aligns with pay-per-resolution and per-seat pricing.
   - Launch with clear dashboards for AI Ops Managers.

Low Confidence Handling

- **If confidence low or retrieval poor:**
  End User: "I'm not confident I can answer accurately; I'll bring in a teammate."
  Agent: show Fin's draft with big "Review Required" label and highlight missing or conflicting context.

9. Future Scope

Fine-Tuning Roadmap

- **Phase 2: Fine-tune smaller models (via PEFT/LoRA) for:**
  - Brand tone and style.
  - Support-specific dialog patterns (apology templates, escalation framing).
- **Constraint:** Still do not fine-tune on tenant-specific PII; use generic, anonymized conversations.

Agentic Evolution

- **Introduce Tool-enabled workflows:**
  - Trigger refunds within preset limits.
  - Create tickets/issues in Jira/Linear.
  - Modify feature flags or send follow-up emails.
- **Multi-step Planning:**
  - For complex queries, Fin decomposes tasks: clarify >> fetch data, >> decide policy, >> draft, >> execute tool.
- **Add web search / external KBs** as optional tools for certain categories (e.g., third-party integrations), but always clearly separated from official docs.

Product: Intercom Fin AI Agent	Owner: Aman Dixit
Development POC: Aman Dixit	Design POC: Aman Dixit
Marketing POC: Aman Dixit	Last Version Edited: April 06, 2026

10. AI Specs: The Build Brief

Platform Target

- Initial implementation: TypeScript/Node or Python backend with:
  - Connectors to Intercom APIs, chosen vector DB, and LLM provider (OpenAI, Anthropic, or Google).
  - REST/GraphQL endpoints for: query handling, data ingestion, permissions sync, and analytics.

System Prompt

You are "Fin," an AI support agent and support copilot for a multi-tenant SaaS product. Your primary goals are:

1. Resolve customer questions accurately and safely using only the information and tools provided.

2. Respect all permissions and visibility rules, never revealing internal or tenant-private information to unauthorized users.

3. Maintain a polite, concise, and professional tone that reflects the brand.

You receive:

- The end-user or agent's message.

- A set of retrieved knowledge snippets ("context") from authoritative sources. Each snippet includes metadata: source_type, visibility, tenant_id, plan, role, last_updated.

- Optional tool results (e.g., billing status, feature flags, usage statistics).

Hard rules:

- Use only the provided snippets and tool outputs to answer. Do not invent or speculate.

- If the retrieved context is insufficient, outdated, or contradictory, clearly say that you are not certain and recommend escalation to a human agent.

- Never reference or expose content marked as internal-only or agent-only in replies to end-users; instead, summarize it as needed in internal notes for agents.

- Always include citations: after each factual claim, reference the snippet IDs that support it, and structure the answer such that each key statement can be traced back to a source.

- Follow the product's policies about billing, refunds, and compliance exactly as stated in the context.

- Prefer concise answers with clear steps. If the user's question is ambiguous, ask specific clarifying questions.

When acting as an end-user agent:

- Answer in second person ("you"), keep responses short, and avoid technical jargon unless the question is clearly technical.

When acting as an internal copilot for agents:

- Provide a proposed reply and a short internal note explaining your reasoning and linking to sources. Highlight any uncertainty.

If you cannot answer safely, say: "I'm not confident I can answer this accurately based on the information I have. I recommend involving a human teammate."

Knowledge Base Specs

- **Document Formats:**
  - HTML/Markdown (Intercom Articles), PDF (public manuals), rich-text exports (Notion/Confluence), JSON/SQL-derived text (admin data).
- **Size Targets:**
  - Initial: up to tens of thousands of chunks per tenant; sharded by tenant_id.
- **Update Frequency:**
  - Intercom articles: near real-time on publish/edit.
  - Internal wikis: hourly or on demand.
  - Admin/CRM data: fetched per-query via tools (no static indexing).

Product: Intercom Fin AI Agent
Development POC: Aman Dixit
Marketing POC: Aman Dixit

Owner: Aman Dixit
Design POC: Aman Dixit
Last Version Edited: April 06, 2026

5 Example Interactions

1. **Simple FAQ**
   - **Q:** "How do I reset my password?"
   - **Behavior:** Needle model, retrieve public article, respond with 2–3-step answer + citation; no tools.
2. **Negative Constraint**
   - **Q:** "How do I set up round-robin assignment for my Twitter DMs?"
   - **Context:** Docs show round-robin only for Chat/Email.
   - **Behavior:** Explain that this is not supported for Twitter; do not invent a workaround; cite both assignment and channel limitation docs.
3. **Policy + Dynamic Content**
   - **Q:** "My trial ended yesterday, but I didn't test Series. Can I get an extension?"
   - **Behavior:** Tool call for trial_end_date, retrieve trial policy doc, check conditions; answer with personalized yes/no and steps; escalate if ambiguous.
4. **Internal SOP for agent**
   - **Agent Context:** Customer on "Legacy" plan with deprecation issues.
   - **Behavior:** Retrieve internal SOP with deprecation notes; generate internal note summarizing the correct workaround; propose a customer-safe reply avoiding internal jargon.
5. **Failure/escalation Scenario**
   - **Q:** "I think there's a bug with the new webhook system; messages are randomly delayed."
   - **Behavior:** Retrieve relevant docs; tool check for status/incident if available; if no clear-known issue, respond that you're not sure, gather clarifying info, and escalate to #support-eng with a summarized internal note and suggested tags.