**FAA ADVISORY CIRCULAR AI SYSTEM: TECHNICAL IMPLEMENTATION PLAN**

**Solo Deployment with Optional N8N Orchestration**

**Scope:** 1 person, iOS deployment, <500 queries/month  
**Vector DB:** Pinecone (Starter free tier)  
**Framework:** LangChain + MCP  
**Optional Enhancement:** N8N multi-agent verification  
**Timeline:** 4 weeks (Phase 1) + 2 weeks (Phase 2, optional)

**I. ARCHITECTURE & TECHNOLOGY DECISIONS**

**1.1 Vector Database Selection**

For solo deployment with <500 queries/month and 3,000 vectors (150 ACs @ ~20 vectors each):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Database | Storage | Cost | Scaling | Cold Start | Why Not |
| Pinecone | Cloud SaaS | $0  *(2GB, 1M reads/mo)* | To millions | <1ms | ✅ Best for low usage requirements |
| Supabase pgvector | PostgreSQL | $5–10/mo | To 100k vectors | 50–100ms | Slower cold start |
| Weaviate Cloud | Cloud SaaS | $0 (free tier) | Good | <50ms | Free tier, less proven |
| Qdrant Cloud | Cloud SaaS | $0 (free tier) | Excellent | <1ms | Free tier limited features |

**Rationale:** Pinecone's free tier (*2GB, 1M reads/mo*) – capacity for over 200K-300K vectors. Roughly 3K vectors required (~15MB) and 500 queries/mo will use <1% of these limits. Fast retrieval (<1ms latency). Mature API with LangChain integration. Can upgrade seamlessly if volume grows.

**1.2 Embedding Strategy**

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Choice | Cost | Rationale |
| Embedding Model (large) | OpenAI text-embedding-3-large | $0.13/1M tokens | ✅ 3,072 dimensions, but with increased cost. Updatable |
| Embedding Model (small) | OpenAI text-embedding-3-small | $0.02/1M tokens | 1,536 dimensions *(~96% as accurate as large*); much cheaper |
| One-time Embedding Cost | 3,000 pages × 1,000 avg tokens = 3M tokens | ~$0.06 | One-time corpus embedding |
| Monthly Refresh | New ACs detected; embedded weekly | ~$0.20/mo | GitHub Actions monitors FAA website |
| Alternative | Local MiniLM (sentence-transformers) | $0 | 384 dimensions; 10% lower quality; CPU-bound |

**Decision:** OpenAI text-embedding-3-large. Cost negligible; quality superior to local options for technical FAA language.

**1.3 LLM Choice**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Cost/ Query | Quality | Speed | Context Window | Best For |
| Claude Haiku 4.5 | $0.018 | 95% quality | Fast | 200k tokens | ✅ Upgraded accuracy – min cost increase |
| Claude Haiku 3.5 | $0.015 | 92% quality | Fast | 200k tokens |  |
| GPT-4 Mini | $0.020 | 94% quality | Fast | 128k tokens | Slightly better accuracy |
| GPT-4 Turbo | $0.30 | 98% quality | Slower | 128k tokens | Expensive for volume |
| Llama 3.1 (local) | $0 | 80% quality | CPU-bound | 8k tokens | Requires GPU |

**Rationale:** Claude Haiku 4.5 offers the best overall value for this RAG environment. It preserves the 200 k-token context window and sub-2 second latency of Haiku 3.5 while improving reasoning accuracy and citation reliability. The cost increase is marginal (~$1.50 more per 500 queries) and easily offset by higher retrieval precision in technical FAA documents. Haiku 3.5 remains a stable fallback. The model can be swapped seamlessly in LangChain via a single configuration variable.

**II. PHASE 1: RAG-ONLY IMPLEMENTATION (WEEKS 1–4)**

**2.1 Hallucination Rates at Each Stage**

**Baseline (ChatGPT, no RAG):**

* Medium-context: 40–50% hallucination
* High-context: 50–60% hallucination
* Low-context: 70–80% hallucination

**Phase 1 (RAG Only) — Expected Outcome:**

|  |  |  |  |
| --- | --- | --- | --- |
| Query Context | General LLM hallucination  rate | RAG system hallucination  rate | Notes |
| High | 50-60% | 3-8% | - 80-90% hallucination reduction  - Cross-AC relationships exist in corpus  - Ensemble retrieval captures all relevant sections |
| Medium | 40-50% | 3-8% | - 80-90% hallucination reduction  - Full AC corpus available in retrieval  - Single-AC answers well-grounded |
| Low | 70-80% | 20-30% | - 60-70% hallucination reduction  - Question itself is vague; corpus cannot fix inherent ambiguity  - System can clarify and re-ask |

**Validation:** RAG achieves 3–8% on "well-known" topics. With complete FAA AC corpus, all queries are "well-known."

**2.2 Technical Architecture**

Document Pipeline

- GitHub Actions monitors FAA site

- Weekly check for new/updated ACs

- Download + extract text (PyPDF)

- Chunk: 1,000 tokens w/ 100 overlap

- Embed: OpenAI text-embedding-3

- Store: Pinecone

LangChain RAG Pipeline

Query → Normalize text

→ Embed query

→ Search Pinecone

├─ Semantic search (0.6 wt)

└─ BM25 keyword (0.4 wt)

→ Ensemble top-5 chunks

→ Re-rank (optional)

→ Format context (MCP)

→ Send to Claude

→ Parse + cite

→ Confidence score

API Backend (Railway/Render)

- FastAPI endpoints

- Rate limiting

- Query logging → Supabase PostgreSQL

- Caching layer (Upstash Redis)

iOS App (SwiftUI)

- Query text box

- Results display

- Citations + confidence badge

- Offline cache (last 20 queries)

**2.3 Implementation Steps (High Level)**

**Week 1: Document Preparation**

1. Download 150 FAA ACs from faa.gov/airports/resources/advisory\_circulars/
2. Extract text via PyPDF2 or pdfplumber
3. Chunk using LangChain RecursiveCharacterTextSplitter (1,000 tokens, 100 overlap)
4. Add metadata: AC number, section, applicability, publish date
5. **Output:** ~3,000 chunks ready for embedding

**Week 2: Vector DB + Embedding**

1. Create Pinecone account → Starter free tier
2. Create index: dimension=3,072 (text-embedding-3-large), metric=cosine
3. Embed all chunks via OpenAI API (~$0.60 total, ~10 minutes)
4. Upload to Pinecone (~1 minute)
5. **Output:** Searchable corpus in Pinecone

**Week 3: LangChain + API**

1. Build LangChain retriever:
   * **PineconeVectorStore** for semantic search
   * **BM25Retriever** forkeyword search
   * **EnsembleRetriever** combining both (0.6/0.4 weights)
2. Build generation chain:
   * LLM: Claude Haiku 4.5
   * Prompt: Context + query + cite instruction
3. Add MCP server for context standardization:
   * Format retrieved chunks consistently
   * Token counting
   * Caching support
4. Wrap in FastAPI:
   * **POST /query** endpoint
   * Request validation
   * Response formatting (answer + citations + confidence)
5. Set up Databases:
   * Create Supabase account 🡪 create PostgreSQL database (free tier)
   * Create audit\_logs table with schema:
     + id (uuid, primary key)
     + query\_text (text)
     + confidence\_score (float)
     + llm\_response (text)
     + citations (jsonb)
     + timestamp (timestamptz)
   * Copy connection string for FastAPI
   * Set up Upstash Redis:
     + Create Upstash account at upstash.com
     + Create Redis database (free tier: 10K commands/day)
     + Configure FastAPI with connection URLs
   * Configure FastAPI caching:
     + Install: pip install redis supabase
     + Add Redis client with Upstash URL
     + Cache query embeddings (key: hash of query text, TTL: 24 hours)
     + Cache retrieval results (key: embedding hash, TTL: 1 hour)
     + Log all queries to Supabase PostgreSQL
   * Deploy to Railway Starter or Render (free tier for FastAPI hosting)
6. **Output:** Live API endpoint with Supabase + Upstash

**Week 4: iOS App + Testing**

1. SwiftUI app in Xcode:
   * TextField for question input
   * Button to submit query
   * Display area for answer + AC citations + confidence badge
   * URLSession to call API
2. Connect to your API endpoint
3. Test 50 queries manually:
   * Compare answers to actual ACs
   * Measure hallucination rate (target: 3–8% on medium, high)
   * Record latency
4. Add local caching (last 20 queries) for offline mode
5. **Output:** Working iOS app on your device

**2.4 Phase 1 Costs**

|  |  |  |
| --- | --- | --- |
| Component | Cost | Notes |
| Pinecone | $0/mo | Free tier (2GB, 1M reads/mo) |
| OpenAI embeddings (one-time) | $0.06 / 1X | 3M tokens × $0.02 per 1M |
| Claude Haiku LLM (500 q/mo) | $9/mo | 500 × $0.018 per query |
| Railway hosting | $0-5/mo | Free first month, then $0-5/mo |
| GitHub Actions | $0/mo | Free for public repos |
| Supabase PostgreSQL | $0/mo | Free Tier |
| Upstash Redis | $0/mo | Free Tier |
| iOS development | $0/mo | XCode already purchased |
| Phase 1 Total/Month | **$9–14** | After first month |

**III. PHASE 2: N8N MULTI-AGENT ORCHESTRATION (OPTIONAL, WEEKS 5–6)**

**3.1 When to Add Phase 2**

**Provides:**

|  |  |  |
| --- | --- | --- |
| Capability | Description | Business Value |
| Proactive AC Change Detection | Monitors FAA site daily and alerts users of new or revised ACs. | Keeps compliance content current automatically. |
| NOTAM Integration | Links live NOTAM data to related ACs and regulations. | Adds real-time situational awareness. |
| Compliance Audit Reporting | Auto-generates monthly summaries on accuracy and activity. | Eliminates manual audit prep and improves oversight. |
| Multi-Agent Architecture | Five agents handle retrieval, summarization, explanation, verification, and review. | Improves reasoning reliability and traceability. |
| Cross-Model Verification | Uses Claude 4.5, GPT-4o Mini, and Sonnet 4.5 for layered validation. | Boosts accuracy through independent cross-checks. |
| Confidence Scoring & Flagging | Combines similarity and verifier scores into a 0–100 rating; flags low-confidence outputs. | Adds measurable reliability and prevents bad answers. |
| Explainability Layer | Generates short “why this is correct” citations to FAA sources. | Ensures transparency and audit-ready outputs. |
| Human-in-the-Loop Feedback | Sends low-confidence answers to review; user input retrains retrieval weights. | Enables continuous learning and accuracy gains. |

**3.2 Phase 2 Value Proposition**

**Phase 2 = Compliance Intelligence Automation:**

|  |  |  |  |
| --- | --- | --- | --- |
| Automation / Capability | Metric | Annual Value (Est.) | Notes / Cost Basis |
| AC Change Detection | Auto-alerts when FAA updates or releases new ACs | 4–7 hrs saved per update × 12 updates ≈ **$5,000 / yr** | N8N + Claude Haiku 4.5 summarizer (< $2 / mo) |
| NOTAM + AC Cross-Ref | Links new NOTAMs to related ACs | 15–20 hrs / yr ≈ **$1,800 / yr** | FAA RSS feed + n8n logic; no API cost |
| Audit Readiness Report | Auto-generates monthly compliance dashboard | 24–36 hrs / yr ≈ **$2,900 / yr** | Claude Haiku 4.5 PDF summary; AWS SES email (Free Tier) |
| Cross-AC Validator | Flags conflicting or outdated guidance | 8–12 hrs / yr ≈ **$1,000 / yr** | Pinecone query + LLM verify ($1 / mo) |
| Confidence Scoring & Flagging | Adds 0–100 trust score; auto-flags < 0.70 responses | ~5 hrs saved per audit ≈ **$600 / yr** | N8N automation only (Free) |
| Explainability & Citation Layer | Generates short, cited rationale per answer | 6–8 hrs / yr ≈ **$800 / yr** | Built into LLM prompt; no extra cost |
| Human-in-Loop Feedback | Reviewer dashboard for flagged responses | 6–10 hrs / yr ≈ **$600 / yr** | Simple webform + PostgreSQL log (Free) |
| TOTAL ANNUAL VALUE | — | **≈ $10.7 K /solo user** | **Cost:** $31-39 / mo  *All****-****inclusive P1+P2* |

**Rationale:** Phase 2 introduces low-cost automation that converts the RAG system from a static reference tool into an active compliance assistant. Core functions—AC change detection, NOTAM cross-referencing, and audit reporting—run through lightweight n8n automations and the free Pinecone tier, keeping ongoing expenses under $50 per month. The multi-agent design (retrieval, summarization, verification, and review) improves accuracy while maintaining near-real-time updates and explainable outputs. Together, these upgrades deliver 18-22X ROI. Phase 2 can be deployed incrementally, scaling from a single analyst instance to multi-user environments without new infrastructure or licensing costs.

**3.3 Phase 2 Architecture** *- components*

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Role | Tools / Models | Cost Tier |
| Data Ingestion & Monitoring | Detects new ACs and NOTAMs; triggers n8n workflows. | n8n HTTP Request + RSS Feed nodes | Free |
| Embedding & Storage | Converts PDFs to 512-dim vectors; stores in Pinecone. | OpenAI text-embedding-3-small + Pinecone Starter | ≈ $0.02 / 1M tokens (FREE storage) |
| Retrieval Agent | Finds top-ranked FAA sections relevant to query. | LangChain Retriever + Pinecone index | Free |
| Reasoning Agent | Generates initial answers from retrieved chunks. | Claude Haiku 4.5 (primary) | ≈ $2 / mo |
| Verification Agent | Cross-checks answers for accuracy and cites sources. | GPT-4o Mini (verifier) + Claude Sonnet 4.5 (esc.) | ≈ $3 / mo |
| Confidence & Flagging | Scores 0–100 confidence; flags < 0.70 for review. | n8n logic + PostgreSQL log | Free |
| Explainability Layer | Produces short, cited “why this is correct” paragraphs. | LLM prompt template within LangChain | No extra cost |
| Audit & Feedback | Logs decisions and user feedback; updates retrieval weights. | n8n + Supabase / PostgreSQL | Free |

**3.3 Phase 2 Architecture** *– workflows, components, and logic*

Workflow 1: **Schedulers (N8N)** – *triggers and timing layer*

* AC Change Check – 06:00 UTC
* NOTAM Fetch – 08:00 UTC
* Monthly Audit – 1st of month

Workflow 2: **Detect and Parse (N8N)** – *automation layer*

* Fetch FAA Site + NOTAM RSS
* Compare old vs new AC PDFs
* Parse NOTAM Keywords

NODE/Component: **Embed & Upsert** – *vectorization layer*

* Use OPENAI text-embedding-3-large (1,536 dim)
* Upsert into Pinecone (AC. + NOTAM namespaces)

Agent Component 1: **Retriever Agent** – *RAG retrieval logic*

* Queries Pinecone for top-K related FAA Account
* Returns relevant document chunks

AGENT COMPONENT 2: **Reasoning (Claude Haiku 4.5)** – *primary LLM layer*

* Drafts answer and references sources
* Sends citations to verifier

AGENT COMPONENT 3: **Verification (GPT-4o mini/Sonnet 4.5)** – *cross-model verification layer*

* GPT-4o Mini cross checks facts
* Escalates complex cases to Sonnet 4.5
* Outputs verified answer + citation list

LOGIC LAYER: **Confidence Scoring + Flagging** – *evaluation layer*

* Combines Pinecone similarity + verifier agreement
* Generates 1-100 confidence score
* Flags results for <0.70 for review

LOGIC + REPORT LAYER: **Explain/Gate** – *explainability & governance layer*

* If >= 0.70: generate short “Why is this correct” paragraph
* If <0.70: flag for review and log feedback

WORKFLOW 3: **Outputs** – *final orchestration layer*

* Sends email/mobile update alert
* Updates PostgreSQL/Supabase audit log
* Generates monthly PDF report

**3.4 Implementation (High Level)**

**WEEK 5: N8N Setup + AC Detection**

1. **Create N8N account** (Cloud tier: $19/month, or self-host)
2. **Build Workflow 1**: FAA Change Detection
   * **HTTP Request node:** Check FAA website daily
   * **Conditional node:** New AC detected?
   * **Download + Compare:** Old vs new PDF
   * **Extract text** → run **OpenAI text-embedding-3-small**
   * **Upsert to Pinecone DB** (AC namespace)
   * **LLM node:** Claude Haiku 4.5 → summarize changes in plain English
   * **Verification call:** GPT-4o Mini → fact check and return confidence score
   * **Confidence Logic:** If < 0.70 → flag for review; else continue
   * **Email node:** Send summary + citations to user
   * **Supabase node:** Log changes and confidence score to audit table
3. **Test:** Manual FAA AC update to confirm alert and logging
4. **Output:** Daily alerts trigger automatically with FAA AC’s change

**WEEK 6: NOTAM + Audit Workflows**

1. **Build Workflow 2**: NOTAM Processing
   * **HTTP Request:** Fetch FAA NOTAM RSS feed
   * **For each NOTAM:** Extract keywords + location codes
   * **Query Pinecone:** Find related ACs (semantic match)
   * **LLM node:** Claude Haiku 4.5 → summarize link between NOTAM and AC
   * **Verifier:** GPT-4o Mini cross-check → confidence score 0–100
   * **Condition:** If < 0.70 → flag for review via email dashboard
   * **Format alert:** Include NOTAM ID, AC reference, and score
   * **Send:** Mobile notification + email summary
2. **Build Workflow 3:** Monthly Audit Report
   * **Query PostgreSQL audit log** *(last 30 days)*
   * **Analyze:** Hallucination rate, confidence patterns, retrieval accuracy
   * **LLM node:** Claude Sonnet 4.5 → generate plain-language summary
   * **Generate PDF report:** n8n PDF node or Markdown → PDF script
   * **Email node:** Send report to user (AWS SES free tier)
3. **Output:** Automated compliance reporting + continuous audit logging

**Architecture Integration Notes**

* Each workflow calls the shared RAG pipeline for retrieval and verification.
* n8n handles scheduling, logging, and report delivery.
* Agents (Claude Haiku 4.5 → GPT-4o Mini → Sonnet 4.5) manage reasoning, cross-model verification, and explanations.
* Entire system runs within $31–39 / month (Phase 1 + 2 combined).

**3.5 Phase 2 Costs**

|  |  |  |
| --- | --- | --- |
| Component | Cost *(Monthly)* | Notes |
| N8N Cloud | $19 | Cloud-tier; includes scheduling, email, and automation nodes. *($0 if self-hosted on Railway)* |
| LLM Calls (Summaries + Verification) | $3–6 | Claude Haiku 4.5 (primary reasoning) + GPT-4o Mini (verifier) + Claude Sonnet 4.5 (escalation). ~500 queries / mo ≈ <$0.02 each. |
| OpenAI Embeddings | <$0.10 | One-time cost ≈ $0.06 for 3 M tokens; negligible for weekly refresh (~$0.20 / mo). |
| Pinecone Vector DB | $0 | Free tier (2GB, 1M reads/mo) |
| FAA Website Monitoring | $0 | HTTP requests via n8n; no API cost. |
| NOTAM Feed | $0 | FAA RSS feed is free and open-access. |
| Email Delivery | $0 | AWS SES free tier (≤ 3 K emails per month). |
| PostgreSQL / Supabase Logging | $0 | Free plan for light audit logging (< 500 MB). |
| PDF Reporting | $0 | n8n native Markdown → PDF conversion or custom Python script. |

**Cost Breakdown:** Phase 2 add-on: $22-$25/mo. Total cost with Phase 1: $31-$39/mo.

**Rationale:** All services run within free or low-tier limits. n8n orchestrates automations, while lightweight LLM calls and Pinecone storage keep costs minimal. Even at full automation scale, the system operates for <$50 per month with continuous compliance monitoring, verification, and audit reporting.

**IV. FINAL HALLUCINATION RATES (PHASE 1 ONLY)**

**Your document**: "RAG Only achieves 3–8% on well-known topics"

**Your corpus**: Entire FAA AC database (complete source material). **All queries**: “Well-known”

|  |  |  |  |
| --- | --- | --- | --- |
| Query Type | Baseline | Phase 1 | Improvement |
| Medium (60–70% of queries) | 40–50% | **3–8%** | 80–90% ↓ |
| High (20–30% of queries) | 50–60% | **3–8%** | 80–90% ↓ |
| Low (5–10% of queries) | 70–80% | 20–30% | 60–70% ↓ |

**Why:** Complete corpus = no "unknown" info. LLM grounds every answer in actual AC text.

**NOTE:** Phase 2 adds verification, scoring, and audit layers that further reduce residual hallucination risk from 3–8 % to < 2 % on critical queries through multi-model cross-validation.

**V. DEPLOYMENT CHECKLIST**

**Pre-Launch**

*  Pinecone free-tier account created
*  150 FAA ACs downloaded and extracted
*  Chunking + metadata tagging complete
*  Embeddings generated + uploaded to Pinecone
*  Supabase account created + PostgreSQL database configured
*  Upstash account created + Redis database configured
*  Railway/Render account created *(for self-host or API)\_*
*  LangChain pipeline tested locally
*  MCP server implemented *(manages local model calls)*
*  FastAPI deployment tested *(for app 🡪 API handoff)*
*  Multi-agent logic verified locally *(retriever 🡪 reasoning 🡪 verifier 🡪 confidence)*
*  Confidence scoring thresholds *(>= 0.70 pass / < 0.70 flag)*
*  iOS app UI built in Xcode
*  API integration in iOS complete
*  Supabase connection tested *(audit logs writing)*
*  Upstash Redis connection tested *(caching working)*

**Launch Week**

*  Deploy API to Railway/Render
*  Deploy iOS app *(via TestFlight)*
*  Run 50-query validation test *(covering High / Medium / Low contexts)*
*  Measure hallucination rate *(target: 3–8% on medium/high)*
*  Measure latency *(target: <2 seconds)*
*  Verify confidence scoring output *(sample 10 queries >= 0.70 threshold)*
*  Monitor cost tracking *(LLM + embeddings < $1/day)*

**Optional (Week 5–6)**

*  Enable N8N orchestration *(if query volume > 1 000 / month or automation needed)*
*  AC Change Detection active *(Workflow 1)*
*  NOTAM integration live *(Workflow 2)*
*  Audit reporting active *(Workflow 3)*
*  Automated verification & flagging live
*  Monthly accuracy report auto-emailed

**VI. RISK MITIGATION**

|  |  |
| --- | --- |
| Risk | Mitigation |
| Vector search misses relevant chunks | Ensemble retrieval (semantic + keyword) + re-ranking + context expansion |
| LLM cost spikes | Monitor OpenAI / Anthropic dashboards weekly; set API spending alerts; throttle non-critical workflows |
| FAA website structure changes | Use AWS Textract (500 docs / mo free) for adaptive PDF parsing; fallback to direct HTML scrape |
| Pinecone free tier expires | Backup index and migrate to Qdrant or Weaviate (free community tiers); auto-sync corpus on transition |
| Hallucination worse than expected | Add MCP caching + cross-model verification + confidence tuning (raise threshold to ≥ 0.75) |
| iOS app crashes offline | Implement local SQLite cache + last 20 query history; auto-retry failed syncs |

**VII. SUCCESS CRITERIA**

* ✅ Phase 1: 3–8% hallucination on medium + high-context queries
* ✅ Phase 1: <2 second latency per query
* ✅ Phase 1: <$20/month ongoing cost
* ✅ Phase 2 (opt): AC changes detected within 24 hours
* ✅ Phase 2 (opt): NOTAM alerts within 2 hours
* ✅ Phase 2 (opt): Monthly audit reports generated automatically
* Phase 2 (opt): Confidence scores > 0.70 on 95% of verified answers

**Technical Plan Version:** 1.0  
**Deployment Model:** Solo, iOS, <500 q/month  
**Target Hallucination:** 3–8% (Phase 1); < 2% after verifications (Phase 2)  
**Estimated Timeline:** 4 weeks Phase 1, +2 weeks Phase 2 (optional)

**VIII. POST-LAUNCH VALIDATION PLAN**

**Week 1 – System Verification**

*  Confirm multi-agent chain executing correctly (retriever → reasoner → verifier → confidence → explain)
*  Validate confidence threshold logic (≥ 0.70 pass / < 0.70 flag)
*  Cross-test 10 queries per category (High / Medium / Low context)
*  Check LLM token usage and daily cost tracking (< $1 / day target)
*  Verify citations and explanations display in final answers

**Week 2 – Performance & Reliability**

*  Measure latency (< 2 seconds avg) and identify slow nodes
*  Run hallucination re-test (≥ 95 % queries above 0.70 confidence)
*  Validate NOTAM → AC linkage alerts sent within 2 hours
*  Monitor n8n workflow failures and error handling (log retry counts < 3 per day)
*  Confirm PostgreSQL/Supabase audit logs record confidence scores and LLM versions

**Week 3 – Reporting & Governance**

*  Receive first auto-generated monthly PDF report
*  Verify audit summaries include hallucination rate, confidence pattern, and cost trend
*  Cross-check report accuracy against raw logs (± 5 % variance max)
*  Archive validated report to versioned folder (GitHub / Drive)
*  Document final performance baseline for Phase 3 planning

**Phase 2 Completion Criteria**

|  |  |  |
| --- | --- | --- |
| Metric | Target | Validated By |
| Confidence Accuracy | ≥ 95 % responses ≥ 0.70 score | Validation log |
| Latency | < 2 seconds | n8n metrics |
| Hallucination Rate | < 2 % after verification | Audit report |
| Monthly Report Delivery | Auto PDF within 1 day of month-end | Email timestamp |
| Ongoing Cost | <$50 / month | Billing summary |