Optimization & Scalability Strategy

Monitor and Improve Performance Over Time

1. Multi-Layer Observability Stack

Infrastructure & Application Metrics (Prometheus + OpenTelemetry)

Key Metrics:

- Request Rate: refill_requests_per_minute (by intent type, PA role)
- Latency Percentiles: p50, p95, p99 for end-to-end flow
- Error Rate: failed_requests_total (by failure type: circuit_breaker, validation_error, ehr_timeout)
- Agent Utilization: Active conversations, queue depth, agent saturation

LLM & Agent Performance (LangFuse)

Purpose: Track AI decision quality, prompt effectiveness, and conversation flows

Tracked Dimensions:

1. Intent Classification Accuracy
    - Confidence score distribution
    - Misclassification rate (compare predicted vs. human-labeled ground truth)
    - Fallback to circuit breaker frequency
2. Entity Extraction Quality
    - Slot fill success rate (per entity type: drug_name, dose, quantity)
    - Disambiguation frequency (how often RAG vector search is triggered)
    - Clarification turns required (target: <2 turns for 90% of requests)
3. Safety Validation Outcomes
    - Escalation rate by trigger type (allergy, DDI, controlled substance)
    - False positive rate (escalations overridden by physicians)
    - Blocked refills (auto-reject due to major contraindications)
4. Prompt Performance
    - Token usage per prompt template
    - LLM response time by model (Claude Sonnet 4 vs alternatives)

  - ○ Cost per conversation (prompt + completion tokens)

Test & Evaluation Loops

Unit Testing with Synthetic Data. FHIR fixtures for edge cases

Integration Testing with Policy Fuzzing. Randomize medication combinations to stress-test policy engine

Human-in-the-Loop Eval (Weekly Review)

1. Sample 1-2% of escalations randomly

Feedback Loop:

- False positives → Adjust policy thresholds (e.g., moderate DDI threshold)
- Missed risks → Add safety rules, retrain intent classifier
- Track improvement: Target <2% false positive rate, 0% missed risks

DSPy for Prompt Optimization. Hand-crafted prompts degrade over time as language patterns shift

Scale the Architecture to Support Multiple Workflows

Stateless Agent Design. Agents hold no local state; all context stored externally

Conversational memory and state Storage: Redis Cluster

Support refill workflow + future workflows (pre-op screening, discharge planning)

Workflow registry + dynamic routing

Shared Components (avoid duplication):

- EHR Agent (used by all workflows)
- RAG Vector Store (formulary, policies, clinical guidelines)
- Escalation orchestrator (physician handoff)
- Audit logging

Hybrid Retrieval Strategy. Pure vector search misses exact matches; sparse search misses semantic similarity. Combine BM25 (sparse) + dense embeddings

Document Versioning & Provenance. Policies change; need audit trail of which version informed decisions

Automated re-indexing pipeline