

[CLIENT LOGO]

# [CLIENT NAME]

## AI READINESS ARCHITECTURE ASSESSMENT

**Prepared by:** ITX Solutions Architecture team
**Engagement:** Tier 1 + 2 - code, documentation, and interviews

5 October 2026

# TABLE OF CONTENTS


1. EXECUTIVE SUMMARY ..... 3
2. OUR APPROACH ..... 5
3. THE PRIORITIZED USE CASES ..... 6
4. AI READINESS SCORECARD ..... 8
5. FINDINGS BY VIEW ..... 9
6. PRIORITIZED RECOMMENDATIONS & ROADMAP ..... 12
7. NEXT STEPS ..... 14
APPENDIX A - SCOPE & ENGAGEMENT TIER ..... 14
APPENDIX B - EVIDENCE SOURCES & TOOLS ..... 14
APPENDIX C - HOW WE SCORED ..... 14

# 1. EXECUTIVE SUMMARY

The Company asked a direct question: for the three prioritized use cases - a de-identified data foundation (UC-5), AI-driven concept discovery (UC-3), and agentic operational orchestration (UC-1) - can the Platform carry the ambition as it stands, where are the gaps, and where is re-platforming warranted? This report answers that question with evidence, so leadership can invest with confidence.

One pattern runs through every view: the modern tier is a genuine reuse asset, while the legacy tier and every AI-specific capability are greenfield. The NestJS API, the data pipeline, a defined Snowflake foundation, and an above-average AI governance policy are real strengths to build on. The work concentrates on the legacy UC-1 path, the data foundation itself (de-identification, consolidation, and quality), and the AI layers that do not exist yet.

The answer to invest versus a full re-platform: invest in the data foundation now - UC-5 is achievable with focused work and unlocks the other two. Treat UC-3 and UC-1 as sequenced, net-new AI builds on top of it. The AI layers are new either way. Two things deserve attention immediately, independent of any use case: secrets present in source-control history, and delivery pipelines that run no automated security or test gate - both straightforward to remediate.

## What's working - the reuse assets

- **A clean, modern API.** The modern NestJS API is well-modularized - five circular dependencies across 901 files, with clear domain and infrastructure layering and a global JWT guard. It is a sound surface to extend for agent use.
- **A modern, tested data pipeline.** The data pipeline is modular, carries real error handling and integration tests, and moves data reliably between Snowflake and MongoDB.
- **A deterministic scaffold for agents.** Today's rules engine already codifies the actions an agent would take (status changes, letters, inventory assignment), runs event-driven, persists rich audit state, and enforces rule-level guardrails.
- **A defined, well-structured Snowflake foundation.** The Source database is layered and constrained (RAW → STG → CURATED), and leadership has already defined the centralized data foundation as a core initiative.
- **Above-average AI governance policy.** An approved AI acceptable-use policy, a governance committee, and a HIPAA-eligible Bedrock-under-BAA sandbox give the Company a real policy head start.
- **Disciplined engineering standards.** ESLint, Prettier, and tsconfig are enforced consistently across the TypeScript estate.

## Where the work is

- **The legacy UC-1 path.** The legacy Express server carries 90 circular dependencies concentrated in the exact operational services an agent would drive, alongside god-files (a 7,904-line invoice repository, and an 898-change app module) and 371 disabled tests.

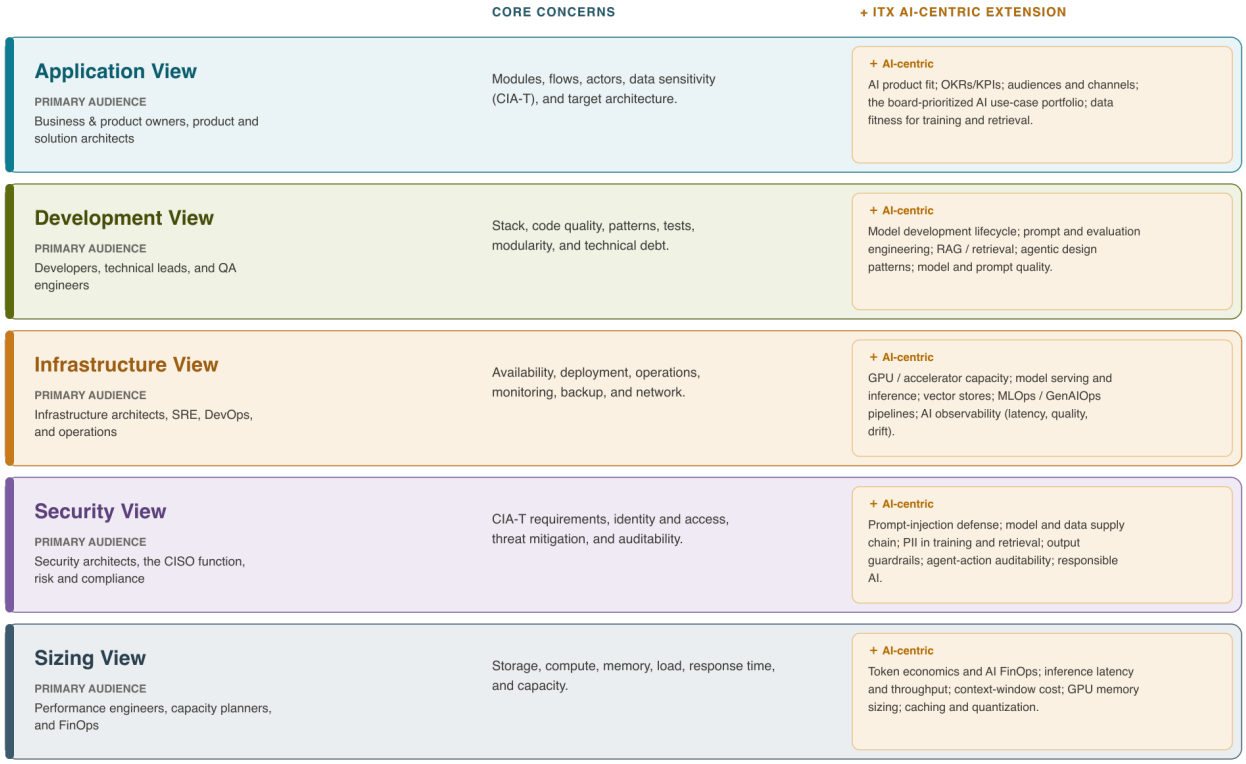
- **The data foundation itself.** The majority of clients land only in the document store, the audit collections are heavily polymorphic, and there is no de-identification, automated lineage, or continuous data-quality monitoring yet.
- **Every AI-specific layer.** Agentic patterns, guardrails and human-in-the-loop, prompt and evaluation engineering, model SDLC, and the retrieval layer for RAG are all greenfield.
- **Security and delivery hygiene.** High-signal secrets sit in Git history, IAM grants are over-broad, the dependency estate carries a significant critical and high vulnerability backlog alongside potential injection sites, and no pipeline runs an automated security, dependency, or test gate.

| | |
|---|---|
| **OVERALL READINESS** | Data Foundation (UC-5) - **Gaps** (investable now).<br>Concept Discovery (UC-3) - **Re-platform** (retrieval + GenAI discipline are greenfield).<br>Operational Orchestration (UC-1) - **Re-platform** (legacy path + agent layer need rework). |

# 2. OUR APPROACH

ITX baselined the Company's architecture against recognized frameworks - the cloud Well-Architected pillars, TOGAF, and the NIST AI Risk Management Framework - and healthcare and engineering best practice, scoring each view against a consistent maturity rubric so the findings stay objective and comparable. This was a Tier 1 and 2 engagement, working from source code, technical documentation, a live read-only profile of the non-production MongoDB, the Snowflake schema export, and the leadership materials.

## The ITX five-view model




The ITX five-view model - five role-aligned views, each extended with the AI, data, and product dimensions a modern assessment demands (shown in orange).

We examine every system through five role-aligned views. Each view answers the questions a specific audience cares about - the Application view for business and product leaders, the Development view for engineering, the Infrastructure view for operations, the Security view for the CISO function, and the Sizing view for capacity and cost.

What makes the frame AI-centric is the extension shown in orange. Each view keeps its established core concerns and adds the dimensions modern AI demands - data fitness and the AI use-case portfolio in the Application view; retrieval, prompt and evaluation engineering, and agentic patterns in Development;

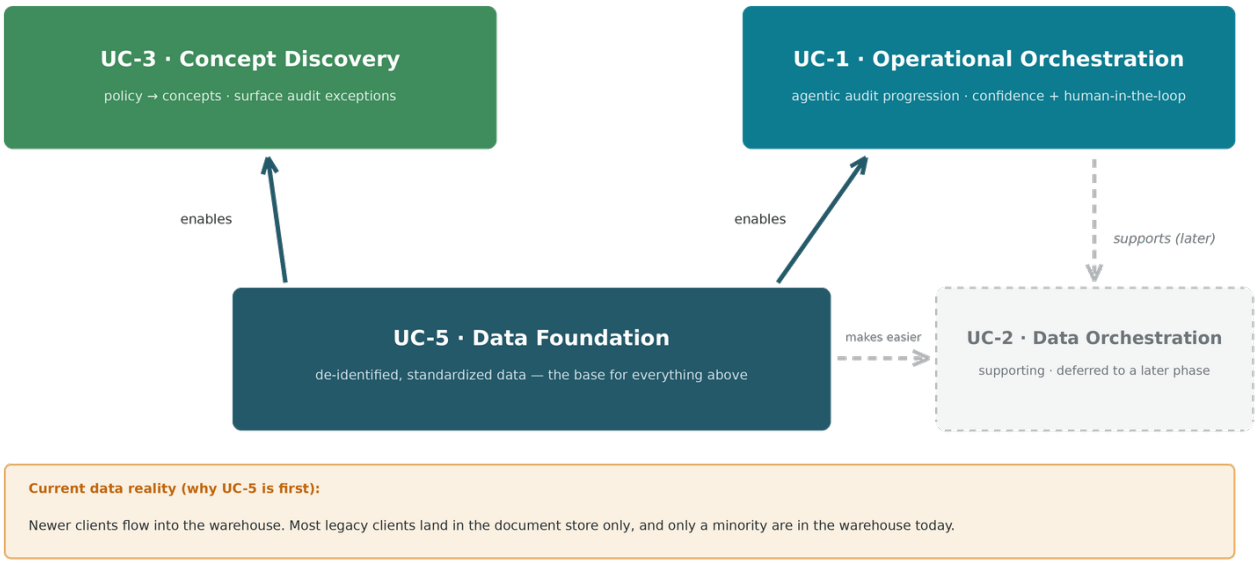
model serving, vector stores, and AI observability in Infrastructure; prompt-injection defense, output guardrails, and agent auditability in Security; and token economics and inference cost in Sizing.

| | |
|---|---|
| **SCOPE** | In scope: the Platform and NestJS API, the data pipeline, the .NET rules and ETL family, the Snowflake warehouse (schema export), the MongoDB operational store, and infrastructure-as-code. Evidence tier: Tier 1 + 2 (code, docs, interviews). No live-environment or production-data access to this engagement. The legacy Automation repository is reference-only. |



# 3. THE PRIORITIZED USE CASES

The assessment is measured against three use cases, confirmed with the Company's Product and Engineering leadership. They are sequenced in priority order - UC-5 (Data Foundation), then UC-3 (Concept Discovery), then UC-1 (Operational Orchestration) - because the data foundation is the base the other two build on. UC-2 (Data Orchestration) was considered and deferred to a later phase.




How the three use cases connect - UC-5 (Data Foundation) is the base that enables both UC-3 and UC-1. UC-2 is supporting and deferred.

## UC-5 · Data Foundation - Priority 1 · core enabler

- **What it is.** A de-identified, cross-client data foundation spanning the Company's MongoDB and Snowflake estate - the standardized ground truth for benchmarking, analytics, and the AI behind the other two use cases.

- **Today.** Newer customers flow core records into the warehouse first, then the document store. The majority of legacy customers land directly in the document store and are not in the warehouse at all.
- **Target outcome.** A standardized, higher-quality, well-understood data foundation that de-risks the invest-versus-re-platform decision and unlocks UC-3 and UC-1.

## UC-3 · Concept Discovery - Priority 2

- **What it is.** AI over transactional history and policy documents to generate audit concepts and surface audit exceptions.
- **Today.** Concept discovery is rules-and-queries-driven and performed by the Services team per client - manual and bespoke, not yet productized.
- **Target outcome.** More exceptions identified, broader coverage, and a productized, normalized concept-generation capability across the client base.

## UC-1 · Operational Orchestration - Priority 3

- **What it is.** Agents move an audit through its lifecycle - evaluating status, applying the next step, and assigning and prioritizing inventory - with a human in the loop.
- **Today.** Operational flow is event-driven and rules-based. A newer rules engine exists but covers only some customers and is binary (pass or fail).
- **Target outcome.** Higher auditor throughput, faster cycle time, and less manual review, with confidence-scored decisions and a human in the loop.

| | |
|---|---|
| **PRIORITY & SEQUENCE** | UC-5 enables both UC-3 and UC-1, so the foundation comes first. UC-2 (Data Orchestration) is supporting and deferred to a later phase. |

# 4. AI READINESS SCORECARD

The scorecard shows readiness for each prioritized use case across the five views. Read a column for a use case's readiness. Read a row to see whether a theme affects one use case or several.

| VIEW | UC-5 · DATA FOUNDATION | UC-3 · CONCEPT DISCOVERY | UC-1 · OPERATIONAL ORCHESTRATION |
|---|---|---|---|
| Application | Gaps | Re-platform | Gaps |
| Development | Gaps | Re-platform | Re-platform |
| Infrastructure | Gaps | Gaps | Gaps |
| Security | Gaps | Re-platform | Re-platform |
| Sizing & cost | Gaps | Gaps | Gaps |
| OVERALL | Gaps | Re-platform | Re-platform |
| HOW TO READ THIS | | No use case is Ready as-is. The Data Foundation is the most achievable. Concept Discovery and Operational Orchestration are held to Re-platform by the Development and Security rows. | |

## The maturity scale (1–5)

Each dimension of every view is scored on a five-level scale. The levels are deliberately general, so the same scale applies consistently across all five views.

| LEVEL | NAME | WHAT IT MEANS |
|---|---|---|
| 1 | INITIAL | Ad hoc or absent. The practice is missing, undocumented, or applied inconsistently by individuals. |
| 2 | DEVELOPING | Partial. Some practices exist, but they are inconsistent, manual, or incomplete. |
| 3 | DEFINED | Documented and generally followed. Repeatable, though not always measured or enforced. |
| 4 | MANAGED | Consistent, measured, and enforced - for example, gated in the pipeline. |
| 5 | OPTIMIZED | Continuously improved and largely automated. A genuine strength to build on. |

## From maturity to readiness

The scorecard reports readiness per use case, not raw maturity. We translate the maturity of the dimensions that matter for each use case into one of three plain-language ratings:

| READINESS | WHEN WE ASSIGN IT |
|---|---|
| **READY** | The dimensions critical to this use case are at Level 4-5, with no blocking risks. Usable as-is. |
| **GAPS** | The critical dimensions sit at Level 2-3. Usable after defined, scoped improvements before scaling. |
| **RE-PLATFORM** | One or more critical dimensions are at Level 1, or a blocking risk is present. Material investment or replacement is the economical path. |
| **BLOCKING RISKS** | Blocking risks override the average. Some findings cap readiness regardless of the other scores - for example an active exposed secret, sensitive data that is unfit or unsafe to use, or an unbounded autonomous agent action. We flag these explicitly and let them set the rating. |



# 5. FINDINGS BY VIEW

## 5.1 Application view - Readiness: Gaps (maturity 3/5)

**Summary.** A defined, well-structured Snowflake foundation and a clean, typed NestJS API make the platform a viable base. The audit data model and the retrieval layer are what hold the AI use cases back.

### What's working

- A layered, constrained Snowflake Source schema and clean MongoDB members/providers collections - a real substrate for the data foundation.
- The modern NestJS API is typed, documented with Swagger annotations, and secured by a global JWT guard.

### Opportunities

- **The MongoDB audit collections are heavily polymorphic** - the primary audit collection carries 88 mixed-type fields across 1,189.
- **The retrieval layer is greenfield** - zero vector columns, no Cortex Search, and no chunkable text.

### Recommendations

- Publish and version the OpenAPI contract. Add idempotency and rate-limiting before agents call the API.
- Define normalized, typed serving projections over the polymorphic audit collections.
- Stand up a retrieval layer (vector/Cortex) and a de-identified corpus for UC-3.

**Evidence & confidence.** Tier 1 (NestJS/Swagger code, Snowflake DDL, live read-only Mongo $sample) + Tier 2 (aligned use cases). High confidence.

## 5.2 Development view - Readiness: Re-platform (maturity 2/5)

**Summary.** The modern repositories are healthy and tested. The legacy path an agent would drive is the most debt-laden, most tightly-coupled, and least-tested code in the estate.

### What's working

- The NestJS API and the data pipeline are well-modularized and carry real integration tests - the reuse assets.

### Opportunities

- The legacy Express server has 90 circular dependencies through the UC-1 operational services (versus five in the modern API), alongside god-files.
- The legacy repository carries 371 disabled tests - the safety net is weaker than it looks.

### Recommendations

- Treat UC-1 as a re-platform on the server tier: build the agent against the clean NestJS API.
- Establish a test-coverage baseline in CI and triage the 371 disabled tests.
- Stand up prompt versioning and an evaluation harness before GenAI ships.

**Evidence & confidence.** Tier 1 (SonarQube, madge dependency graphs, Git-history hotspots, coverage inventory) + Tier 2 (interviews). High confidence.

## 5.3 Infrastructure view - Readiness: Gaps (maturity 3/5)

**Summary.** Deployment mechanics are mature and codified. The gaps are consistent infrastructure hardening, unhardened containers, and the complete absence of automated delivery gates.

### What's working

- Infrastructure is codified as modular, per-customer Terraform. The platform runs on ECS/Fargate with no Kubernetes complexity to manage.

### Opportunities

- **IaC hardening gaps** - a recurring set of infrastructure-as-code misconfigurations, repeated across shared modules.
- Container images are unhardened: several run as root, several lack a health check.

### Recommendations

- Fix the IaC findings once at the shared-module level. Add an IaC scan gate.
- Harden container base images and move build-arg secrets to runtime injection.
- Add security, dependency, and test gates to the pipelines.

**Evidence & confidence.** Tier 1 (Terraform, Dockerfiles, pipeline definitions, Checkov and Trivy) + Tier 2 (interviews). Live runtime posture is inferred.

## 5.4 Security view - Readiness: Re-platform (maturity 2/5)

**Summary.** The AI governance policy is a genuine strength. Operational controls are the gap: high-signal secrets in source history, over-broad access, injection on the critical path, and no runtime guardrails.

### What's working

- An approved AI acceptable-use policy, a governance committee, and a Bedrock-under-BAA sandbox - an above-average governance foundation.

### Opportunities

- **High-signal secrets sit in Git history** - cloud credentials, private keys, and package-registry tokens.
- **AI controls are greenfield** - no content-safety guardrails, no human-in-the-loop boundary for UC-1's autonomous actions.

### Recommendations

- Rotate exposed secrets, purge Git history, and move secrets to a vault. Add a CI secrets gate.
- Before UC-1 acts on live records: tool allow-lists, human-in-the-loop, and full agent action logging.
- Right-size IAM to least privilege. De-identify PHI for any AI use.

**Evidence & confidence.** Tier 1 (gitleaks, Semgrep, Syft/Grype, Checkov/Trivy, LLM-integration scan) + Tier 2 (policy documents, interviews).

## 5.5 Sizing & cost view - Readiness: Gaps (maturity 2/5)

**Summary.** Platform sizing is adequate and Snowflake already delivered a major efficiency win, but there is no cost model for the AI use cases.

### What's working

- Moving to the cloud warehouse cut client data-load time by roughly an order of magnitude - a strong efficiency signal.

### Opportunities

- No token or inference cost model for any AI use case. Agent and RAG economics are unquantified.
- Cost-runaway signals: Lambdas lack concurrency limits and the API lacks rate limits.

### Recommendations

- Build a per-use-case cost model and validate against a metered pilot.
- Set budgets and rate limits before wider AI exposure.
- Run a Tier 3 FinOps review once live usage data exists.

**Evidence & confidence.** Tier 1–2 estimate from architecture, IaC, and interviews. Runtime cost is inferred.

# 6. PRIORITIZED RECOMMENDATIONS & ROADMAP

All 30 prioritized recommendations, sequenced to make the foundation safe and usable first, then to close the delivery and security gaps that protect every use case, then to build the AI capabilities. Each maps to a finding and to the Technical Findings & Remediation Guide (same REC IDs). Effort is shown in AI-augmented person-weeks (how the Company's team actually builds) - a high-level SWAG, not a quote; the full register totals roughly 132 to 323 AI-augmented person-weeks.

| REC | RECOMMENDATION | VIEW | WHEN | EFFORT (AI PW) | OUTCOME |
|---|---|---|---|---|---|
| **REC-01** | Confirm & remove committed client data | Sec / App | Now | ~1.5-3 pw | No client data in source control. |
| **REC-02** | De-identify PHI for AI use | App / Sec | Now | ~12.5-30.5 pw | PHI usable safely for production AI. |
| **REC-03** | Consolidate Mongo-only clients into Snowflake | App / Infra | Next | ~9.5-22.5 pw | A complete cross-client ground truth. |
| **REC-04** | Complete customer AI-use authorization | Sec | Now | ~2-6 pw | Legal basis to run AI on client data. |
| **REC-05** | Export & assess the Analytics database | App | Now | ~1.5-3.5 pw | Fastest safe path for UC-3. |
| **REC-06** | Typed serving projections over audit data | App / Dev | Now | ~5.5-13.5 pw | Deterministic agent grounding for UC-1. |
| **REC-08** | Event-driven triggers + a DAG orchestrator | Infra / Dev | Next | ~5-13 pw | Lower latency + reliable event triggers for agentic orchestration. |
| **REC-09** | Automated lineage + data-quality tests | Dev / App | Next | ~5-12 pw | Auditable provenance + proactive quality for AI trust (CIA-T 'T'). |
| **REC-10** | Data-quality profiling + freshness SLAs | Dev / Infra | Next | ~2-5 pw | Drift/quality caught before it reaches AI. |
| **REC-11** | Publish & version the OpenAPI contract | App / Dev | Now | ~1.5-3 pw | A stable, discoverable contract for agent callers. |
| **REC-12** | Idempotency, versioning & rate limiting | App / Sec | Now | ~4-10 pw | Safe agent retries + stable contracts. |
| **REC-13** | Route agents to the NestJS API; retire legacy | App / Dev | Next | ~2-5 pw | One consistent contract. |
| **REC-14** | Quality gates + targeted debt paydown | Dev | Now | ~6-14.5 pw | Safer, cheaper AI-era change. |
| **REC-15** | De-duplicate hotspots; extract shared modules | Dev | Next | ~2-5.5 pw | A maintainable base for AI-assisted change. |
| **REC-16** | Add a CI code-quality gate | Dev / Infra | Now | ~1-2 pw | A quality gate that blocks regressions. |
| **REC-17** | Split the UC-1 god-files | Dev | Now | ~8.5-21 pw | Lower change-risk on the agent path. |
| **REC-18** | Break the legacy Express coupling | Dev | Now | ~8-19 pw | A safe, extensible agent path. |
| **REC-19** | Establish a test-coverage baseline | Dev | Now | ~3.5-9.5 pw | A safety net that de-risks re-platform. |
| **REC-20** | Triage the 371 disabled tests | Dev | Next | ~1-2 pw | An honest, trustworthy safety net. |

| | | | | | |
|---|---|---|---|---|---|
| **REC-21** | Scope Lambda IAM to least privilege | Sec / Infra | Now | ~0.5-1.5 pw | Least privilege. |
| **REC-22** | Rotate, purge & vault secrets | Sec | Now | ~1.5-3.5 pw | Closes an active credential exposure. |
| **REC-23** | AI security + guardrails + human-in-the-loop + audit | Sec | Now | ~10-25 pw | Safe, accountable autonomous AI actions. |
| **REC-25** | Fix injection/XSS; add SAST to CI | Sec | Now | ~4-9 pw | Close injection/XSS classes before agents drive these paths. |
| **REC-26** | Dependency-update cadence + SCA gate | Sec | Now | ~3.5-8 pw | Known vulnerabilities remediated and gated. |
| **REC-27** | Fix the shared IaC module + CI gate | Infra / Sec | Now | ~1.5-3.5 pw | Hardened, drift-resistant infrastructure. |
| **REC-28** | Harden container images | Infra / Sec | Now | ~1-2 pw | Least-privilege, secret-free containers. |
| **REC-29** | Add security & test gates to CI/CD | Infra / Dev | Now | ~2-5.5 pw | No risk ships ungated; a model/prompt path. |
| **REC-30** | Build the UC-1 agent layer safely | Dev | Now | ~13.5-33 pw | Realizes UC-1 agentic orchestration. |
| **REC-31** | Decide the model strategy | Dev | Next | ~8-19 pw | A clear, reproducible model path. |
| **REC-32** | Prompt versioning + an evaluation harness | Dev | Now | ~4.5-12 pw | Measured, release-gated GenAI quality. |

| | |
|---|---|
| **NEAR-TERM FOCUS** | Start with the items marked Now - the data-foundation, secrets, and delivery-pipeline work. They make the data foundation safe to build AI on, close the secret exposure, and turn the delivery pipeline into a control point. |

# 7. NEXT STEPS

This report is the shared, objective baseline for the work ahead. We recommend taking the “Now” items and the foundation-and-orchestration path into a **Product Roadmap Workshop** with your team, where we sequence them against your goals and timeline and shape the path to the AI outcomes the Company wants to deliver. We look forward to partnering with you on what comes next.

APPENDIX

# APPENDIX A - SCOPE & ENGAGEMENT TIER

In scope: the Platform and NestJS API, the data pipeline, the .NET rules and ETL family, the Snowflake warehouse (schema export), the non-production MongoDB (read-only profile), and infrastructure-as-code. Evidence tier: Tier 1 + 2 (code, documentation, interviews). Out of scope: live-environment access (cloud consoles, production databases, production PHI, runtime telemetry) and the Sizing/observability Tier-3 items, recommended as a follow-up. The legacy Automation repository is reference-only.

# APPENDIX B - EVIDENCE SOURCES & TOOLS

We drew on read-only, non-production evidence at Tier 1–2:

- Repository reconnaissance (scc)
- SonarQube Community - code quality
- Technical-debt and Git-history hotspot analysis
- madge - dependency graphs
- Semgrep - SAST (static application security testing)
- Syft and Grype - SBOM and dependency vulnerabilities
- gitleaks - secret scanning, full history
- Checkov and Trivy - IaC and container posture
- A live read-only MongoDB profile (mongosh, field names only)
- Offline Snowflake DDL analysis

Interviews and leadership materials informed intent. Thirty-four evidence artifacts underpin this report.

# APPENDIX C - HOW WE SCORED

Each view was scored against the ITX AI Readiness Maturity Rubric (1–5), drawn from:

- The cloud Well-Architected pillars
- TOGAF
- The NIST AI Risk Management Framework
- Engineering and healthcare best practice

Readiness per use case (Ready / Gaps / Re-platform) reflects the maturity of the dimensions that matter for that use case and the strength of the evidence. Blocking risks cap readiness regardless of other scores. Non-production data is synthetic, and sensitivity is assessed against the production data model.

# THANK YOU.

This report is the shared baseline for the work ahead - we look forward to partnering with the Company on what comes next.