

Proof, not promises.

A two-engineer Airiam team joined **a regulated US healthcare SaaS platform** mid-flight: live, multi-vendor, and built around AI agents. Within three months we were writing **most of each week's changes**, under governance. This is what we built, how we led, and what it means for you.

Prepared for **Strong Industries** By **Airiam Advanced Technologies**

Figures as of **1 October 2026**

— THE ENGAGEMENT IN NUMBERS

Three months. Two engineers. One very measurable footprint.

Every number below comes from the client's source control and the shared delivery board, not from estimates or timesheets.

1,370⁺

commits

about a third of the platform's entire history, after joining at the 20% mark

96

merged pull requests

small, ticket-linked changes that merge in about 1.4 hours (median)

7,000⁺

automated tests

half of the platform's entire test suite

7 of 21

deployable services

designed and built end to end by Airiam

240⁺

tickets traced

every change maps back to a tracked piece of work

13 days

to live telemetry

from the first line of the observability library to data flowing in pre-production

10 days

accounting integration

connection, bank-feed reconciliation and an exceptions screen

3 days

to a live cloud console

from first code to a continuously deployed app; enterprise sign-in within a week

◦ COMPUTED

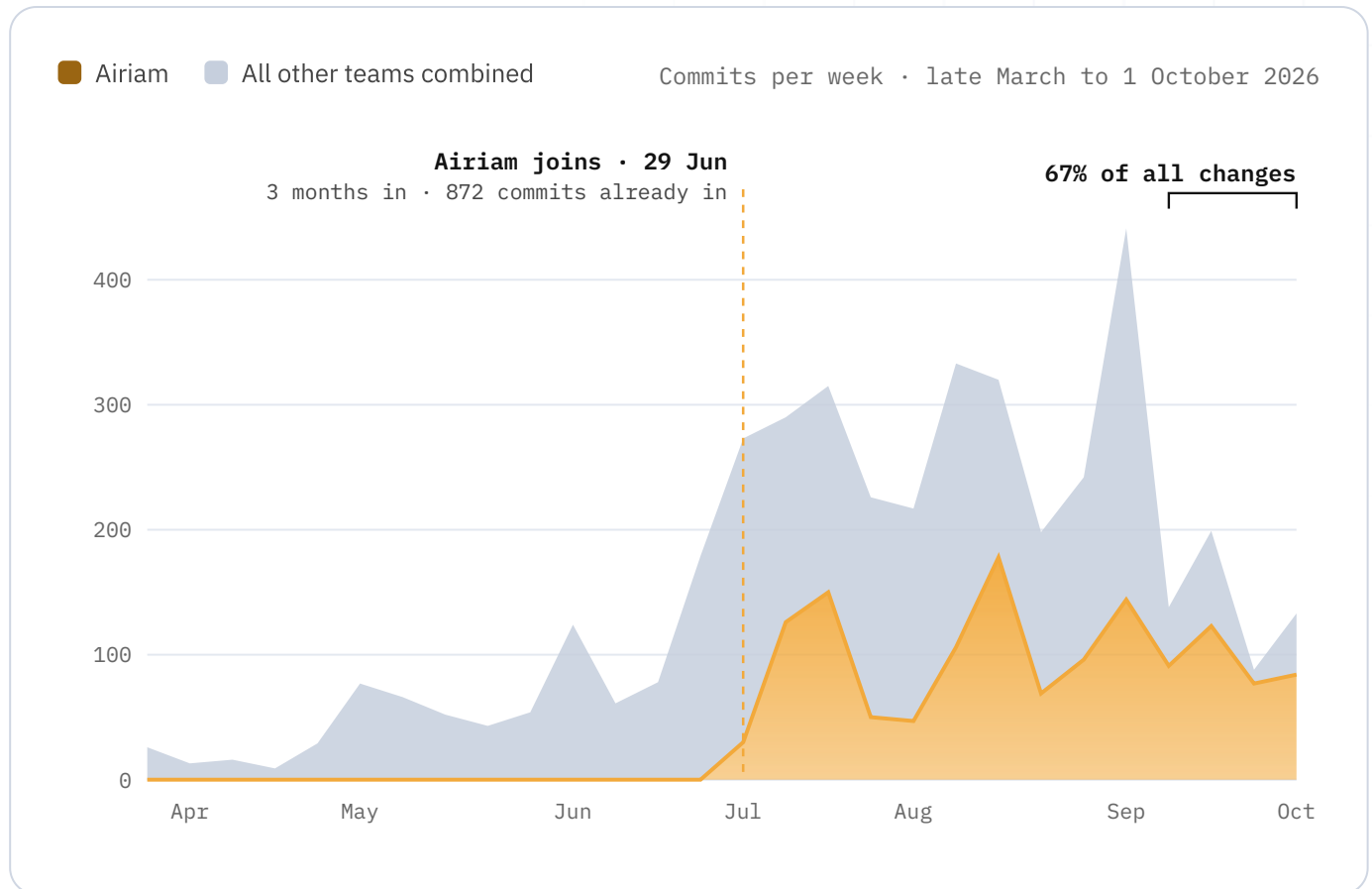
From git history, pull-request records and the delivery board on 1 Oct 2026. Rounded down where marked +.

— JOINING MID-FLIGHT



We arrived when the platform was already moving fast.

The build was already three months in, with engineers from several firms committing daily (**10 other contributors** over the project's life). We didn't pause the train. We got on, earned the work, and within weeks were carrying a growing share of it.



872

commits already in the history when we started

32% → 51%

our share of commits, July to September

67%

of all changes over the last four weeks

— Evidence & method

- **Source:** full commit history of the client repository across all branches, bucketed by ISO week and grouped into "Airiam" (two engineers, all identities de-duplicated) and "everyone else".
- **First Airiam commit:** 29 June 2026. The final week shown is partial, ending 1 October.
- **Monthly share:** July 32%, August 42%, September 51%. Over the last four ISO weeks, 375 of 558 commits (67%).
- Commit counts measure activity, not value. We pair them with the ownership and test figures in the next section.

— PILLAR 1 • BUILT

Whole services, end to end, not tickets at the edges.

We take a capability from specification to running software: design, data model, code, tests, infrastructure and documentation. Each card says exactly where it stands.

● **PRODUCTION BRANCH** merged for production release

○ **LIVE • PRE-PROD** running in the client's pre-production cloud

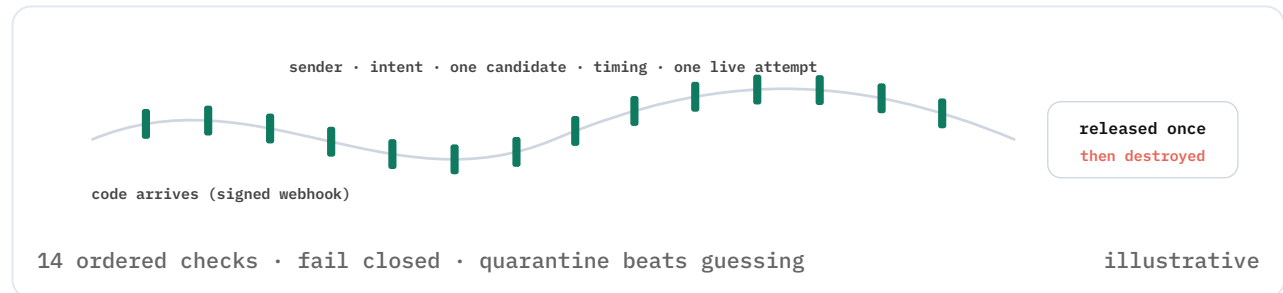
● **BUILT** merged, not yet deployed

● **IN FLIGHT** being built now

● **DESIGNED** design locked

One-time-code relay: removing the human from portal logins, safely

Portal automation was blocked by one manual step: a person typing each login's one-time code. The relay receives the code, matches it to exactly one login and one live attempt, and hands it over once. It then destroys the code and keeps an audit trail of every event without ever logging the code itself. Its first rule: automation must never lock anyone out.



~6 weeks spec → pre-prod

2,900+ passing tests

136 recorded decisions

200+ verification rows

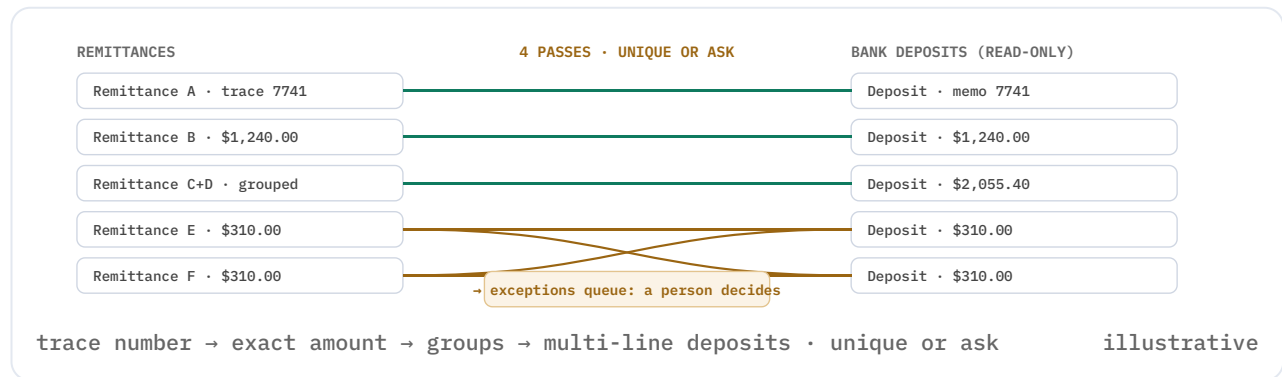
email live • SMS in flight

WHAT IT PROVES

We can ship multi-tenant, security-critical services that handle other parties' secrets, with custody and tenant isolation designed in from day one.

QuickBooks Online reconciliation in 10 days, read-only by construction

We built a per-client connection, a bank-feed reader, a four-pass matcher and an exceptions screen. The matcher links a payment only when the answer is unique, and anything ambiguous goes to a person. When evidence showed that posting entries would double-count money the bank feed already records, we deleted our own working posting engine. The module can now only *read* the books, and tests enforce that.



10 days, 41 tickets

970 test functions

17.5k lines retired on evidence

no patient data can reach the vendor

WHAT IT PROVES

We can integrate accounting and ERP systems where a mistake costs money, and we change course when the evidence says so.

Insurer-portal automation where no API exists

Browser automation behind a stable interface: adding a portal means adding an adapter, not rewriting the core. Status sweeps attach results only on a certain match, and an ambiguous record is never attached.

~97-100% Airiam-authored

940 test functions

WHAT IT PROVES

We can automate brittle third-party portals into governed, auditable workflows.

SCHEDULED DATA FEED

• PRODUCTION BRANCH

A gated partner portal, turned into a weekly governed feed

A scheduled job extracts about 32,000 rules across 36 plans, detects changes, and writes them through the platform's governed write path with an immutable audit snapshot. A crashed run resumes where it stopped.

99% Airiam-authored

790+ test functions

WHAT IT PROVES

We can turn external data into reliable, monitored, resumable pipelines.

DOCUMENT AI

• BUILT

Paper remittances to structured data, gated by arithmetic

OCR plus LLM structuring turns printed PDFs into the same structured record an electronic file produces. Inference stays inside the client's cloud. Acceptance is decided by arithmetic, never by the model's confidence.

100% Airiam-authored

330+ test functions

versioned prompts

WHAT IT PROVES

We use LLMs where they help, and deterministic checks where correctness matters.

OPERATIONS

• LIVE • PRE-PROD

Ops Console and the monitoring service behind it

A read-only operations console, live in three days and secured with enterprise sign-in. It reads a monitoring service that stamps every figure with its source and freshness (more in Pillar 2).

2,172 tests

95% coverage

37 recorded decisions

WHAT IT PROVES

We build operator tools that tell the truth and are tested like products.

AI ticket triage, stood up inside the client's own cloud

We took Airiam's own triage product and deployed a separately branded, fully isolated copy into the client's tenant, with its own identity, data and secrets. The LLM classifies tickets; code makes the consequential decisions.

under 21 h ticket → verified

~980 tests

signed, replay-safe intake

WHAT IT PROVES

We can clone a product into a customer's tenant in days, with nothing shared.

Who wrote what: approximate ownership by subsystem

We claim what is ours and say clearly what isn't. Low-share tiles are areas other teams own, where we only extended, hardened or instrumented their work.

One-time-code relay

~100%

Accounting reconciliation

~100%

Document AI ingestion

~100%

Portal status sweep

~100%

Governed data feed

~99%

Portal automation

~97%

Monitoring read service

~99%

Synthetic prober

~99%

Shared observability library

~97%

Platform error model

~100%

Database test suites

~88%

Infrastructure guard tests

~67%

Deploy pipeline (shared)

~45%

Infrastructure as code

~36%

Agent runtime

~15%

Governed-write gateway

~14%

— Evidence & method

- **Ownership:** share of surviving lines attributed by line-level authorship on the integration branch (for the agent runtime and gateway, share of lines added). These figures depend on file filters, so we round them and present them as approximate.
- **Status:** "Production branch" means merged to the branch the production pipeline deploys from, with production rollout timing under the client's control. "Live · pre-prod" means deployed and exercised in the client's pre-production cloud.
- **Test counts** are static counts of test functions in the code (parametrised cases are not expanded), or totals reported by CI where stated.
- The platform's core governed-write engine and agent runtime were built by other teams. Our work there was extension, hardening, identity and observability.

— PILLAR 2 · OBSERVED

Instrument everything. Leak nothing.

A regulated platform can't adopt off-the-shelf monitoring without risking sensitive data. We designed observability where protected data **cannot leave by construction**: one exporter, a default-deny allow-list, and CI gates that fail the build if anyone widens it.

EVERY SERVICE ● SENSITIVE FIELDS DROPPED AT THE BOUNDARY ● IDS, TIMINGS, COUNTS, OUTCOMES PASS OPERATORS

emit

redact

store

read API

console

17 services, one shared library, two lines to adopt

default-deny allow-list at a single egress

in-boundary telemetry backend, provisioned as code

every figure stamped with source & freshness

read-only, never guesses

— UNAVAILABLE IS NEVER ZERO

Every value travels with its provenance. A missing signal shows up as missing, with the reason, and never as a reassuring zero.

Request latency · p95

412 ms

source: telemetry

window: 1 h

fresh · 2 min

Synthetic availability

100%

source: prober

window: 15 min

fresh · 1 min

Platform resource load

UNAVAILABLE · reason shown

source: —

not wired yet

illustrative values, not client data

— A LESSON WE BUILT IN

reachability ≠ usability

Green health checks don't prove users can work.

So the prober doesn't just ping endpoints. Every five minutes it signs in like a real caller and drives an actual agent turn, on its own tagged series so it can never mask real traffic. We also wrote alert rules for rejection storms and for *silence*, not just for errors.



13 days

first line → live telemetry

17

services instrumented

7

workstreams, delivered in 3 increments

4

activations, each with a dated evidence dossier

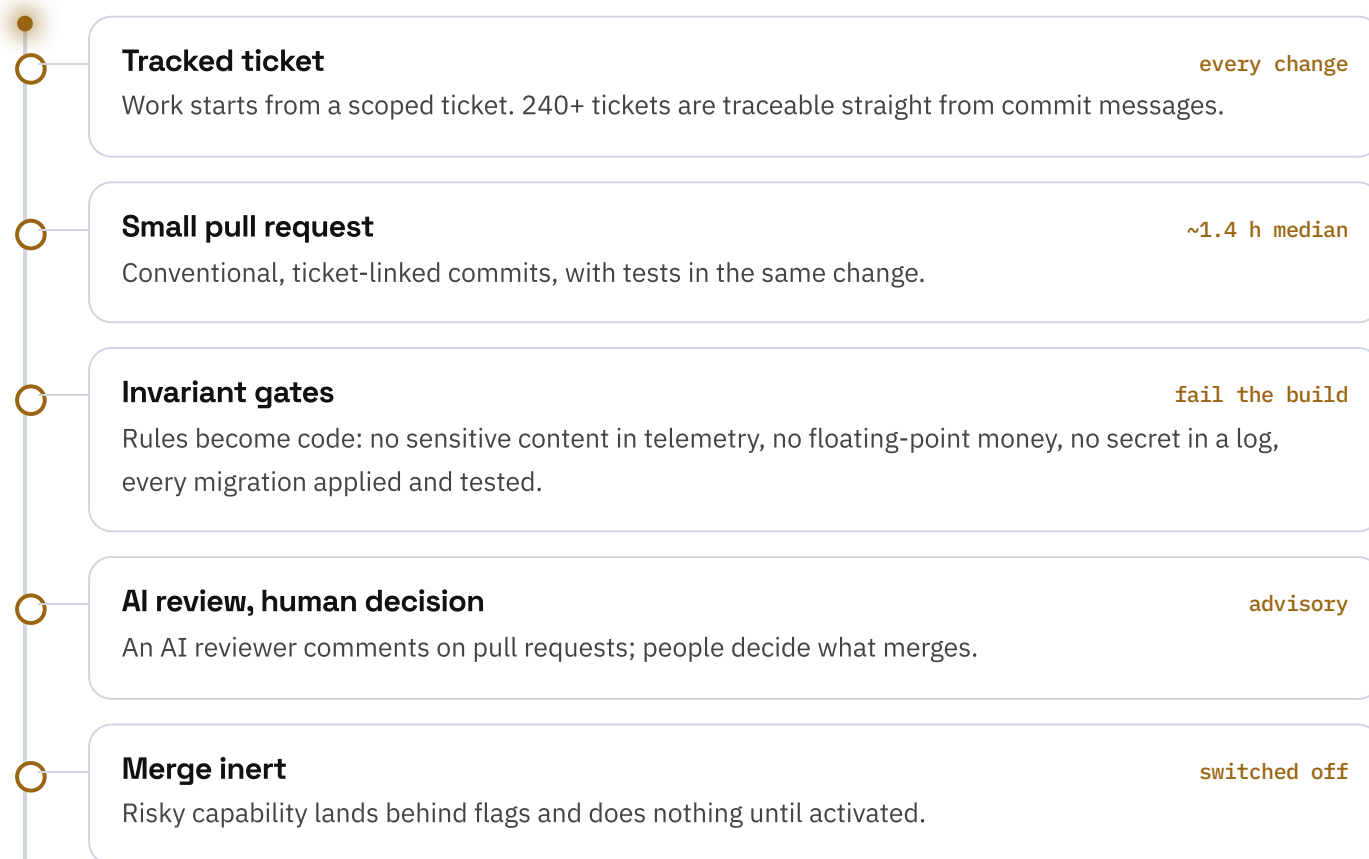
— Evidence & method

- **Model:** two planes. Governance and audit events form one; metrics, traces and AI cost form the other. They are joined by a correlation key. The telemetry plane is built on OpenTelemetry behind a swappable provider.
- **Safety:** one exporter, type-split default-deny allow-list, drift-guard tests, and CI gates for the attribute vocabulary and "no content in telemetry".
- **Delivery:** everything lands switched off, then gets activated per increment with recorded evidence. Status: live in pre-production; the monitoring read service and prober are on the production branch.
- AI cost per model call is measured in integer cents, without content.

— PILLAR 3 • LED

Engineering leadership inside someone else's codebase.

We didn't take over the project or rewrite anyone's work. We added the control plane around it: **rules every engineer and coding agent reads first**, gates that turn rules into failing builds, and a promotion model where risky changes land switched off and are activated with evidence.





Activate with evidence

dated dossier

Activation is a separate, recorded step: what changed, live checks, what was deferred.

The control plane, before and after

When we arrived

Now

Agent-neutral control file 737 lines



before: a single-tool instruction file · now: one rulebook every coding agent and engineer reads first

Infrastructure guard-test files 76 files



48 of them first written by Airiam

Custom invariant CI gates 4



added alongside existing gates: telemetry vocabulary · no-content rule · money as decimal · secret-free logs

AI pull-request review workflow 1,300+ runs



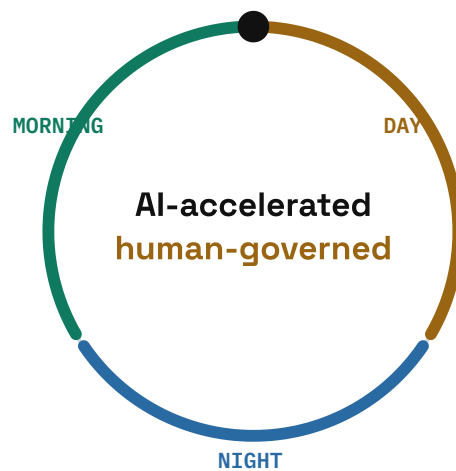
added by Airiam; adopted by the client's lead engineer too

Shared platform libraries for telemetry & errors 2



observability provider · typed platform error model

"Now" counts include contributions from other teams where noted. The deploy pipeline itself was built by other engineers; we co-authored about half of it as it hardened.



— HOW WE DELIVER AT THIS PACE

AI does the heavy lifting. People hold the authority.

- **Day loop:** specification, then tests first, then build and scoped checks.
- **Night loop:** AI reviewers harden and prepare, with a second model family cross-checking. They **cannot merge or push**.
- **Morning gate:** people review what was prepared and make every decision that matters.
- **Human gates on security:** no agent may sign off its own security review, and every decision is written down with its reasoning.

— Evidence & method

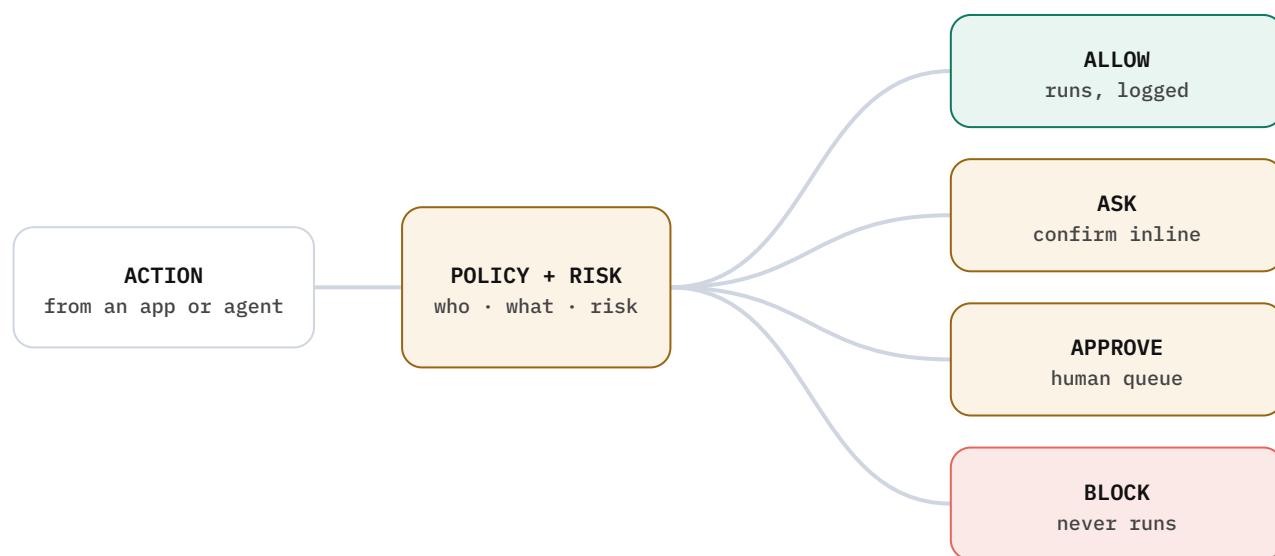
- **Before:** the repository state on the day before our first commit (29 June 2026). **Now:** the integration branch on 1 October 2026.
- **The agent control file** holds documentation precedence, architecture rules for new code, scope naming, security and data rules, skill routing and a definition of done. About 90% of it is Airiam-written.
- **Standards adoption:** our layered-architecture and error-model standards were proposed first, then codified for new code. We don't claim a retrofit of existing services.
- **Review:** pull requests merge after CI and an AI review pass. We don't present this as independent human peer review.

Governance you can watch working.

Airiam's governance model, **Build Safely**, doesn't replace the tools you use to build. It governs what gets shipped and what it does once it's running. On this platform we exercised every tier of it, and it's how we earned the client's trust.

One action, four possible verdicts

Every consequential action passes a policy and risk check before it runs, and the outcome is sealed in an append-only audit trail.



AUDIT TRAIL · newest first
BLOCK change payment details · sealed 2a9e...d4b3
APPROVE issue refund over limit · sealed e1c0...7f52
ASK send customer update · sealed 4b7d...08aa
ALLOW read account summary · sealed 9f3a...c21e
illustrative model · audit hashes shown are sample values

The tiers, and where we proved each

TIER 0

Build Readiness

Assess tools, data and risk. Agree an AI-use policy.

Checked four external review reports claim by claim against the code.

TIER 1

Launchpad

Build-time release gates, monitoring, and a safe deploy path.

Invariant CI gates, infrastructure guard tests, and a pipeline that refuses to report green when nothing changed.

TIER 2

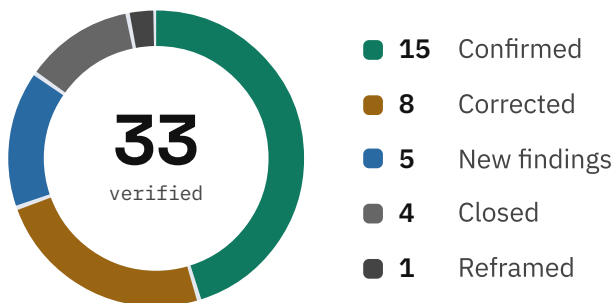
Governed Operations

Runtime control over what systems and agents actually do.

Identity bound to server-side membership, data that can't leak by construction, release-once secrets, and read-only-by-design integrations.

We verify the verifiers

Four external review reports landed on the project at once. We checked every load-bearing claim against the code, corrected a false "critical" headline, and surfaced five real issues the reviews had missed.



Reviews that change what ships

When other teams prepared high-stakes releases, we reviewed them. In one, a single missing column would have made a production write path fail.

```
41 -- promote line items into the shared ledger
42 MERGE INTO ledger_lines AS t USING staged AS s
43 WHEN NOT MATCHED THEN INSERT (id, item, qty)
43 WHEN NOT MATCHED THEN INSERT (id, tenant_id, item, qty)
44 VALUES (s.id, s.item, s.qty);
```

● REVIEWING

● BLOCKING · NEW REQUIRED COLUMN MISSING

○ FIX LANDED SAME DAY

● RELEASED

illustrative code, based on a real pre-release review

7/1

The client's delivery board runs in Airiam's workspace

About half of all its tickets were written by Airiam.

~19 min

The lead engineer asks for our review

On their own incident fix. 10 of our 11 findings were adopted.

Same day

A production blocker caught before release

Fixed and swept by the release owner before the release merged.

Next

Asked about the next stage

Now in discussion to lead the agent platform's next stage.

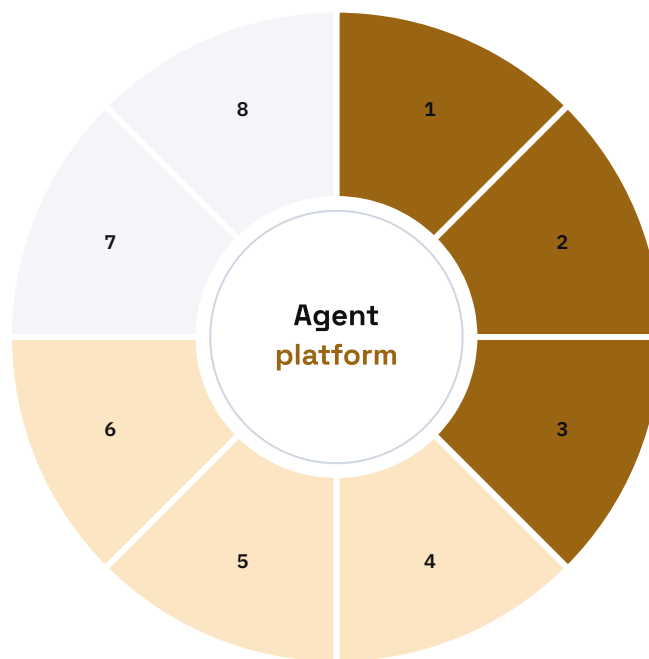
— Evidence & method

- **Verdicts:** allow, ask (confirm inline), approve (human queue) and block form Airiam's governance model. On this platform, the governed-write core was built by another team; we extended it, hardened identity and scope around it, and routed our own modules through it.
- **Review register:** 33 entries, of which 15 were confirmed, 8 corrected, 5 new, 4 closed and 1 reframed. Every entry was checked against a pinned version of the code.
- **Release review:** a fallback write path omitted a column that a new migration made mandatory, and the stated blast radius was 6 tables when the real figure was 16. Both were fixed before the release merged.
- **Board:** the client's earlier work-tracking was migrated into Airiam's workspace with provenance preserved. Authorship was counted from each issue's original creator.

— PILLAR 5 • NEXT

Trusted with what comes next: the agent platform.

The platform's AI agents are operational. Taking them from operational to dependable at scale is the next stage, and Airiam is **in discussion to lead it**. This is the blueprint we bring to it. Highlighted segments are where we've already laid the foundations.



■ Foundations delivered ■ Designed, or built in another Airiam product ■ Blueprint

1 • Agent observability & safety

• FOUNDATIONS DELIVERED

Traces, cost and health for every agent turn, plus a hard safety-interrupt channel.

Live agent traces, per-call cost telemetry, authenticated agent probe · safety interrupt designed

2 • Agent identity & provenance

• FOUNDATIONS DELIVERED

Agents act with authority derived from server-side membership, never client claims.

Identity plane moved to real sign-in in 10 days; scope bound to membership

3 • Runtime & context economics

• FOUNDATIONS DELIVERED

Model routing, context budgets and per-tenant cost attribution.

Measured cost per model call, in integer cents, content-free

4 • Autonomy controls

• DESIGNED

Kill switch, spend and rate caps, approval expiry and re-validation at approval time.

Event-loop protection delivered; control design in hand

5 • Evaluation & regression gates

• DESIGNED

Gold scenarios, injection tests and a merge-blocking evaluation gate.

Pre-deployment evaluation gate designed in Airiam's own platform

6 • Output grounding

• DESIGNED

Deterministic checks that amounts, dates and IDs in a reply match the source data.

Grounding verifier built in another Airiam product

7 • Proactive event lane

• BLUEPRINT

Agents that notice what needs doing: watchers, daily huddles, triggers.

8 • Agent API contract

• BLUEPRINT

Streaming, resumable and card contracts for new front ends.

– Evidence & method

- "Foundations delivered" means Airiam-built and running in pre-production or later: AI call telemetry and cost, live agent traces through a no-sensitive-data projection, an authenticated agent probe, identity bound to membership, and event-loop protection.
- "Designed" means a locked design with no implementation yet (for example, a safety-interrupt channel) or a pattern built in another Airiam product (for example, deterministic grounding checks on LLM output).
- The core agent runtime was built by another team. This blueprint describes the next maturity stage, not a list of defects.

From a partner who keeps you running to one who builds with you.

You already know Airiam as your IT partner. The same company now builds and governs AI systems on live, regulated platforms. Here's how that could grow, one deliberate step at a time.

YOU ARE HERE

1

Secure

The IT support we already provide

2

Integrate

Connect the systems that move orders, money and service, and monitor them.

3

Automate

AI triage, document extraction and portal automation with deterministic guardrails.

4

Govern agents

Agents that act on your systems, with approvals, limits and an audit trail.

PROVEN • LIVE IN PRE-PROD

Accounting integration that reconciles, not just syncs

Built in 10 days; read-only by construction; a person reviews the ambiguous cases.



TO EXPLORE

Orders, invoices and payments across channels

Where do numbers get re-keyed or reconciled by hand today?

PROVEN • LIVE IN PRE-PROD

Automation where there is no API

Portal automation with adapters, plus a relay that handles one-time codes safely.



TO EXPLORE

Dealer, retailer and supplier portals

Which portals does your team log into every week?

PROVEN • LIVE IN PRE-PROD

AI triage with deterministic guardrails

Ticket triage deployed in a client's own cloud; the model classifies, code decides.



TO EXPLORE

Warranty, service and support requests

How do requests reach the right person today, and how fast?

BUILT • TESTED

Document AI checked by arithmetic

Printed remittances turned into structured data, accepted only when the totals balance.



TO EXPLORE

POs, invoices, claims and paperwork

Which documents still get typed in by hand?

PROVEN • LIVE IN PRE-PROD

Operational visibility that never lies

A console where every figure carries its source and age, and missing data says so.



TO EXPLORE

Plants, orders and the service network

What do you wish you could see at a glance every morning?

PROVEN • LIVE IN PRE-PROD

Governed AI agents

Human security gates, identity bound to membership and data that can't leak by construction, on a platform with four-verdict action control.



TO EXPLORE

Agents that act on your systems

Which decisions must always stay with a person?

These are conversation starters, not assumptions about your systems. Every manufacturer's stack, channels and priorities are different. Step 0 is always listening.

— HOW WE'D START

Small first step. No production changes until you say so.

STEP 1 • WEEKS

Build Readiness

A fixed-scope assessment. We map where AI could earn its keep, the data involved, and the risks, then agree the rules.

- Opportunity and risk map
- AI-use and data policy
- No change to your production systems

STEP 2 • FIRST BUILD

Launchpad

One high-value use case, built with the same control plane you just saw: tests, release gates, monitoring and a safe deploy path.

- Gated releases and monitoring
- Decisions recorded with their reasons
- Status reported precisely

STEP 3 • ONGOING

Governed Operations

Runtime control as automation and agents grow: approvals, limits, audit and observability, run with you.

- Four-verdict action control
- Append-only audit trail
- Humans own every high-risk decision

KY

Kuk Yi

CEO • executive sponsor

Owns the relationship and the commitment.

GE

Greg Elmore

Advanced Technologies leadership

Delivery model, standards and engagement shape.

DC

Daniel Chávez

Engineering

Observability, operations console, security hardening and engineering standards.

PA

Pedro Aquino

Engineering

Integrations, portal automation, secret-handling services and accounting reconciliation.

— METHOD & HONESTY

How to read this page

Figures were computed on 1 October 2026 from the client repository's history, its pull-request records and the shared delivery board. Shares are approximate where marked. Attribution counts only work authored by Airiam's engineers; joint work is labelled as joint, and other teams' work is never presented as ours. The client, its partners and its vendors are not named, and no client data appears here. Values shown inside diagrams are illustrative.

• **PRODUCTION BRANCH**

Merged to the branch production deploys from. Rollout is the client's call.

• **LIVE • PRE-PROD**

Running and exercised in the client's pre-production cloud.

● BUILT

Merged and tested, not yet deployed.

● IN FLIGHT

Being built now.

● DESIGNED

Design locked, no implementation yet.

Airiam Advanced Technologies · prepared for Strong Industries

Confidential · not for redistribution