THE RED CELL METHOD · APD-01

AI Platform Delivery

How we design, build and operate AI platforms that earn production: multi-model by design, deterministic where it counts, governed from the first commit. With a worked example: a grant intelligence platform for Australian businesses.

Red Cell Consulting · The method we use on our own platforms, written down

CONTENTS

About this document

The method in these pages is the one we practise: our own platforms, Recursys, Myra, ADMRL and Stagecell, run in production under exactly these disciplines. The worked example, a grant intelligence platform commissioned by a fictional manufacturer, shows the method applied end to end. Every organisation and figure in the example is fictional; the engineering is not.

The thesis: platforms, not pilots

Most organisations do not have an AI problem. They have a pilot problem: demonstrations that impress and evaporate, because nothing underneath them was built to be trusted.

A PILOT AND A PLATFORM ARE DIFFERENT SPECIES

	THE PILOT	THE PLATFORM
What it optimises	The demo	The thousandth day of operation
Where facts come from	The model's fluency	Systems of record, cited
What happens when it is wrong	Someone laughs, or nobody notices	A test fails, a gate blocks, a person is named
Security posture	An API key in a notebook	Tenancy, keys, network, audit, by design
Cost	Unknown until the bill	Modelled per action, monitored per day
After a year	A slide in someone's deck	Load-bearing infrastructure with a roadmap

THE FIVE COMMITMENTS

- **Deterministic where it counts.** Consequential facts and figures are computed by testable code or retrieved from records, never generated. Models narrate; they do not invent the numbers.
- **Multi-model by design.** No single model, no single vendor. Work routes to the cheapest model that clears the quality bar for that task, with fallbacks wired.
- **A brain, not a prompt.** The platform accumulates structured domain memory: registers, graphs, precedents, evidence. Intelligence lives in the system, not in a lucky prompt.
- **Agents through gates.** Anything that acts passes behavioural tests, adversarial checks and permission boundaries before production, and logs everything after.
- **Governed from the first commit.** The register, impact assessment and evidence file are part of the architecture, so the platform is ready for Australia's 2027 rules by construction.

Where this method comes from

We run our own production platforms under it: a deal-intelligence platform that reads entire data rooms, a financial analysis platform whose every figure is provably computed, a clinical reporting platform, and an agent factory that ships assistants through gates. This document is that practice, written down and applied to a new domain on page twelve.

The reference architecture

Six layers, one direction of trust: anything that reaches a person can be traced down the stack to a record, a rule or a test.

THE STACK

LAYER	WHAT LIVES HERE	RULE OF THE LAYER
1 · Channels	Web app, inbox integration, API, notifications	Every surface shows provenance; nothing arrives unexplained
2 · Agents	Task workers: watchers, screeners, assemblers, correspondents	Declared permissions, gated releases, full action logs
3 · The brain	Domain registers, the knowledge graph, precedent memory, evidence library, retrieval	Knowledge is structured and cited, or it is not knowledge
4 · Determinism layer	Rules engines, calculators, validators, schedulers	If it is a number, a date or an eligibility, code computes it
5 · Model gateway	Routing, budgets, fallbacks, structured-output enforcement, red-team filters	Models are suppliers, interchangeable and measured
6 · Infrastructure	Cloud landing zone, tenancy isolation, keys, queues, observability	Boring, reproducible, priced

THE TRUST DIRECTION

Requests flow down; trust flows up. A sentence in a report exists because an agent assembled it from brain entries; the brain entry exists because the determinism layer validated an extraction; the extraction cites a source document; the document has a hash and an ingest date. Ask any answer "says who?" and the platform can show you, four layers deep. That query, "says who?", is the architecture's acceptance test.

What is deliberately absent

No layer where a model talks straight to a user about facts without passing the wall. No agent that acts without a permission entry. No component whose cost cannot be attributed. Absences are design decisions too, and these three are the ones that keep platforms honest.

The determinism wall

The single discipline that separates platforms you can defend from demos you have to apologise for.

THE RULE

Every consequential output is split into two kinds of content. **Computed content**: numbers, dates, eligibilities, thresholds, statuses, produced by deterministic code that can be unit-tested and re-run to the same answer. **Narrated content**: explanation, argument, drafting, produced by models but permitted to reference only computed content and cited sources. The wall is the enforcement point between them: any narrated claim that cannot cite its computed or documentary source is rejected before a person sees it.

WHAT SITS ON EACH SIDE

COMPUTED (CODE, TESTED)	NARRATED (MODELS, GATED)
Eligibility against published criteria, clause-cited	Why this opportunity fits, in plain language
Dates, deadlines, windows, reminders	The cover note that accompanies a submission
Funding maths: caps, co-contributions, budgets	Project narrative assembled from evidence entries
Document completeness checks against checklists	Draft replies to questions, held for sign-off
Register states, ownership, audit trails	Summaries of what changed this week

ENFORCEMENT, CONCRETELY

- Structured outputs with schemas: model responses that do not parse are retried or rejected, never patched by hand.
- Citation-or-reject: generated text is scanned for factual claims; each must resolve to a source id. Unresolvable claims kill the sentence, not the standard.
- Golden answers: the deterministic layer ships with worked examples from the domain (the regulator's own, where they exist), and the test suite replays them on every change.
- No silent fallbacks: when the wall rejects, the user sees "insufficient evidence", which is an honest answer and a work item, not a failure.

Why we are strict about this

Our financial platform runs more than thirteen hundred tests and eleven evaluation gates so that no number a model made up can reach a screen. That standard came from managing real money. It transfers to any domain where being wrong has a cost, which is every domain worth building for.

The brain: what the platform knows

A prompt forgets. A platform remembers, in structures a person can audit.

THE FOUR MEMORIES

MEMORY	WHAT IT HOLDS	HOW IT STAYS HONEST
Registers	The domain's live state: opportunities, obligations, deadlines, owners, statuses	Every row sourced and dated; deterministic validators on write
The knowledge graph	Entities and relationships: programs, agencies, criteria, projects, people, documents	Edges carry provenance; orphan claims cannot join the graph
The evidence library	The organisation's own verified facts: metrics, capabilities, financials, accreditations, past outcomes	Each entry has an owner, a review date and a source document; stale entries are flagged, not trusted
Precedent memory	What was tried and what happened: past submissions, assessor feedback, outcomes, correspondence	Outcomes recorded verbatim; the learning loop reads them, people confirm the lessons

RETRIEVAL THAT RESPECTS THE WALL

When a model needs context, it retrieves entries, not vibes: chunks carry their source ids into the prompt, and the citation checker verifies the output against exactly those ids. Retrieval quality is measured (recall against a golden query set) because a brain that surfaces the wrong memory is worse than no brain at all.

WHAT THE BRAIN IS NOT

- Not a dump of every document ever ingested; curation is a feature, and deletion has an owner.
- Not fine-tuning by default: structured memory is auditable and updatable the day the world changes; baked weights are neither.
- Not shared across clients. One tenant, one brain, cryptographically separated.

The compounding asset

The brain is what makes year two better than year one: every round fought, every question answered, every outcome recorded makes the next one faster and sharper. Pilots cannot compound. This is the part of the platform that appreciates.

Multi-model by design

Models are suppliers. Route the work, measure the quality, keep the exit open.

ROUTING BY TASK AND CONSEQUENCE

TASK CLASS	MODEL TIER	WHY
Judgement work: analysis, argument, complex drafting	Frontier	Quality dominates; volume is low; cost per task is trivial next to the stakes
Volume work: summarisation, extraction to schema, routine drafts	Mid-tier	Quality bar is clearable; volume is high; cost matters
Classification, tagging, routing, screening	Small / task-tuned	Milliseconds and fractions of a cent, measured against a labelled set
Embeddings and retrieval	Dedicated embedding models	Different job, different tool

THE GATEWAY DISCIPLINES

- **One seam:** every model call passes a single gateway that enforces schemas, budgets, timeouts, logging and red-team filters. Swapping a model is a config change behind the seam, not a rewrite.
- **Measured routing:** each task class has an eval set; a cheaper model earns the route by passing it, not by being fashionable. Routing tables are reviewed monthly against cost and quality drift.
- **Fallback chains:** provider outage degrades gracefully: queue, retry, alternate model, or honest unavailability. The platform never silently switches to a model that has not passed the task's eval.
- **Budget attribution:** every call carries a task and tenant tag; cost per outcome (per screen, per submission, per reply) is a dashboard, not an archaeology project.

Vendor posture

We are vendor-neutral by architecture and opinionated by measurement. Today's routing tables lean on frontier models where judgement is at stake and cheaper tiers everywhere else; next quarter's tables will follow next quarter's evidence. The client owns the seam, the evals and the exit.

Agents through gates

An agent is software that acts. That one property changes everything about how it must be engineered.

THE FACTORY PATTERN

STATION	WHAT HAPPENS	EXIT ARTEFACT
Scope	What the agent may read, write and trigger, written as configuration	Permission manifest, reviewed and signed
Context	The documents, registers and constraints it reasons over, versioned	Context bundle with hashes
Behavioural tests	Known cases with expected behaviour, including refusal cases	Green suite, kept green
Adversarial pass	Prompt injection, data exfiltration attempts, boundary probes	Findings closed or accepted with names attached
Sign-off	A person approves the manifest and the evidence	Release record

RUNTIME RULES

- Actions are logged with enough context to reconstruct them, because one day someone will need to.
- Consequential facts route through deterministic tools; the agent orchestrates, it does not remember figures.
- Outbound anything (email, submission, payment, record change) above the lowest consequence class holds for human sign-off. The agent drafts and queues; a person releases.
- Kill switches are boring and tested: per-agent off, per-tenant off, all-off.

The correspondence bar

The highest bar we set is for agents that write to outside parties, and the worked example includes one that drafts replies to a government grantor. It ships with the strictest gates in this document: whitelisted recipients, claim-level citations, mandatory human release, and a tone corpus it must match. Page thirteen shows what that looks like in practice.

Evaluation infrastructure

If it is not evaluated, it is not engineered. Evals are the release gate, the regression net and the routing evidence, all at once.

THE EVAL STACK

LEVEL	WHAT IS TESTED	BLOCKING?
Unit and property tests	The determinism layer: calculators, validators, rules	Yes, per commit
Golden-set evals	Extractions, classifications and screens against labelled truth	Yes, per release
Behavioural agent suites	Each agent's known cases, incl. refusals and escalations	Yes, per agent release
Adversarial suites	Injection, exfiltration, jailbreak attempts on every user-facing seam	Yes, per release
End-to-end scenario runs	The whole platform on a synthetic corpus with known answers	Yes, per release
Drift monitors	Live quality sampled against the golden sets, cost per outcome	Alerting, monthly review

THE SYNTHETIC CORPUS

Before real data arrives, we build a synthetic world with a known answer key: fictional documents, fictional histories, seeded traps. The platform is graded against the key on every release, so "it works" is a score, not a feeling. The corpus grows every time production surprises us; surprises become tests, and tests never un-happen.

Numbers we hold ourselves to

Our own platforms publish their discipline: hundreds of behavioural tests per agent surface, eleven evaluation gates on the financial platform, a three-hundred-document synthetic data room with a scored answer key on the deal platform. A vendor who cannot tell you their equivalent numbers is telling you something.

Infrastructure and security

The unglamorous layer that decides whether everything above it can be trusted, insured and afforded.

THE REFERENCE BUILD · AWS, AUSTRALIAN REGIONS

CONCERN	REFERENCE POSITION
Landing zone	Infrastructure as code from day one; environments reproducible from the repository
Tenancy	Row-level isolation enforced in the database, tested by automated leak checks; option of account-level separation for regulated clients
Keys and secrets	KMS per tenant class; secrets in managed stores; no credentials in code, ever
Network	Private by default; the model gateway is the only egress to AI providers; WAF and rate limits at the edge
Data residency	Australian regions; provider data-processing terms reviewed against client obligations, including zero-retention options for model calls
Observability	Structured logs, traces on every agent action, cost telemetry per task and tenant
Backup and recovery	Point-in-time recovery, tested restores, documented objectives

OPERATING POSTURE

- Deploys are staged with health gates and automatic rollback; releases are boring by policy.
- Incidents follow a rehearsed runbook with client communication templates pre-written.
- Run cost is reviewed monthly against the model on page fourteen; drift gets a ticket, not a shrug.

Security is a property, not a page

Every layer in this document carries its own controls: the wall rejects unsourced claims, agents hold permissions, the gateway filters injections, the infrastructure isolates tenants. This page is where it becomes auditable: the evidence file (page ten) links to all of it.

Ingestion: reading the world honestly

Platforms live or die on their inputs. Ingestion is engineering, not scraping.

THE INGESTION DISCIPLINES

DISCIPLINE	PRACTICE
Provenance	Every ingested artefact keeps its source, retrieval time and content hash; re-reads diff against the hash
Structured first	Where a source offers data (APIs, feeds, structured pages), we take data; models parse only what has no better form
Reconciliation	Parsed content is validated against its own document: totals, dates, internal references. Parsers that cannot self-check do not ship
Freshness	Sources have monitored cadences and staleness alarms; the platform knows what it last saw and when, and says so
Change detection	Diffs on watched sources become events: a changed guideline, a moved deadline, a new round
Respectful automation	Rate limits, identification, terms-of-use review per source; government sources via their published channels wherever offered

THE COMPLETENESS QUESTION

Discovery-class platforms carry a special obligation: silence is a claim. If the platform says "no relevant items this week", it is asserting coverage, so coverage itself is measured: source uptime, crawl success, and sampled audits against manual checks. Where coverage is partial, the platform says partial, because a confident gap is how trust dies.

Documents are adversaries, politely

Real-world documents are inconsistent, versioned badly and occasionally wrong about themselves. The parser suite treats every format quirk found in production as a new regression test. This is slow, compounding, unfashionable work, and it is precisely where reliable platforms are actually built.

Governance built in, 2027 ready

Australia's Standards for AI arrive in 2027. A platform built on this method walks into them already compliant, because the artefacts regulators want are the platform's own working parts.

THE GOVERNANCE ARTEFACTS, BY CONSTRUCTION

WHAT 2027 SIGNALS	WHERE THIS METHOD ALREADY PRODUCES IT
A register of AI systems, data and owners	The platform ships with its own register: models, agents, data flows, named owners, consequence ratings
Impact assessment for consequential uses	Written per agent at the Scope station, filed with decisions
Human accountability	Sign-off gates with names; outbound actions carry a releasing person
Testing and monitoring	The eval stack of page seven, plus drift monitors, all dated
Audit trails and incident reporting	Action logs, incident runbook, the evidence file
Records that survive scrutiny	Everything above is dated as it happens; evidence cannot be backdated, so it never has to be

TRUTHFULNESS AS AN ENGINEERING REQUIREMENT

Where a platform communicates with regulators, agencies or counterparties, truthfulness obligations attach with real consequences. The method treats this as architecture: claims cite evidence entries, evidence entries cite documents, and outbound text is released by a person who can see the whole chain. Honesty, made mechanical, is the cheapest compliance program ever built.

Alignment, not paperwork

ISO/IEC 42001 and the signalled Australian requirements converge on the same grammar: know your systems, rate consequences, name owners, test, keep records. This method does not add a compliance layer at the end; it is the compliance layer, running in production from the first week.

The delivery method

Five phases, fixed cadence, working software from week three. The client's people are in the build, because the platform is theirs.

THE PHASES

PHASE	TYPICAL WINDOW	WHAT EXISTS AT EXIT
1 · Discovery	2-3 weeks	The domain mapped: sources, registers, consequence classes, the synthetic corpus designed, the walking-skeleton scope signed
2 · Walking skeleton	3-4 weeks	The thinnest end-to-end slice live on real infrastructure: one source in, one screen out, the wall enforced, evals running
3 · Vertical slices	6-10 weeks	Capability lands one complete slice at a time, each production-quality, each gated, each demonstrated on the corpus then on real data
4 · Hardening	2-3 weeks	Adversarial passes, load and cost rehearsal, incident drill, documentation, the governance file complete
5 · Operate	ongoing	Monitored production, monthly routing and cost review, the brain compounding, a roadmap owned jointly

THE CADENCE

- Weekly working demo on the synthetic corpus: never slides, always software.
- Fortnightly decision review: what the build learned, what it changes, decisions in writing.
- The client names a product owner with real authority and two domain experts who attend demos; their judgement trains the platform's standards.

Why slices, not layers

Building layer-by-layer means nothing works until everything does. Building slice-by-slice means the platform is always working and always honest about how much world it covers. Stakeholders watch capability arrive; nobody waits nine months for faith to be rewarded.

Worked example: the grant intelligence platform

Hadley Manufacturing (fictional), 140 staff, advanced fabrication across two states. Grants were a folklore process: heard about late, written over weekends, abandoned at the third form. The brief: never miss a relevant program again, and make every submission the best version of the truth.

WHAT THE PLATFORM WATCHES

SOURCE CLASS	COVERAGE	CADENCE
Federal: GrantConnect and agency programs	Forecast and open opportunities	Daily, with change diffs
State programs (VIC, NSW for Hadley)	Business and industry portals, guidelines PDFs	Daily
Local and regional	Councils and development bodies in Hadley's footprint	Weekly
Adjacent instruments	R&D incentives, apprenticeship and energy-efficiency schemes	On change

THE DETERMINISTIC SCREEN

Each opportunity is parsed to a structured record: entity eligibility, size and turnover thresholds, geography, industry codes, project windows, co-contribution requirements, caps and assessment criteria, every field cited to the guideline clause it came from. Hadley's profile (from the evidence library) runs against the rules engine, and the platform returns eligible, ineligible with the failing clause, or human-review with the ambiguous clause quoted. No model opinion enters the screen; eligibility is arithmetic and clauses, which is why Hadley trusts a "no" as much as a "yes".

THE EVIDENCE LIBRARY AT WORK

Hadley's library holds verified entries: headcounts, revenue bands, certifications, plant capabilities, project case studies with measured outcomes, each with an owner and a review date. Submissions may only claim what the library holds. When a draft needs a fact the library lacks, the platform raises an evidence request to the named owner rather than letting a model improvise. The library grew from 40 entries at launch to over 300 within two rounds, because every application funds the next one.

The weekly read

Monday, 7am: the round briefing lands. New opportunities screened with eligibility verdicts and clause citations; changes to watched programs diffed; deadlines inside 30 days flagged with owner and status; and the one-line answer to "what closes this week that we care about?" Provenance on every line. Reading time: four minutes.

Worked example: agents, correspondence and the round

Where the factory pattern earns its keep: the platform's agents, and the one that writes to government under the strictest gates in this document.

THE AGENT ROSTER

AGENT	WHAT IT DOES	CONSEQUENCE CLASS	HUMAN GATE
Watcher	Monitors sources, diffs changes, raises events	Low	None; read-only
Screener	Runs the deterministic eligibility screen, drafts the verdict note	Medium	Verdicts sampled weekly
Assembler	Builds submission drafts from the evidence library against the assessment criteria	Reviewed output	Owner reviews; nothing submits itself
Deadline sentinel	Tracks windows, escalates unowned deadlines	Low	Escalation path only
Correspondent	Drafts replies to grantor emails and clarification requests	**High**	**Mandatory release, every message**
Acquittal clerk	Assembles post-award reporting from project records	Reviewed output	Finance sign-off

THE CORRESPONDENT, IN DETAIL

- Reads the grantor's email in thread context, retrieves the submission, the evidence entries and prior correspondence.
- Drafts a reply in which every factual claim carries a citation chip to its evidence entry; uncited claims are rejected at the wall before a human ever sees the draft.
- Whitelisted recipients only; attachments only from the submission record; tone matched to a reviewed corpus.
- A named person releases every message. Median human effort observed: ninety seconds, because checking cited claims is fast and writing from scratch is not.

ONE ROUND, MEASURED (FICTIONAL BUT HONEST NUMBERS)

METRIC	BEFORE	WITH PLATFORM
Relevant opportunities surfaced per quarter	3-4, by luck	17, screened
Time to eligibility decision	Days	Minutes, cited
Submission assembly effort	30-40 hrs	9-12 hrs, mostly review
Grantor queries answered inside two days	Sometimes	Every time, with citations
Deadlines missed	Untracked	Zero, sentinel-owned

The economics, and how these projects fail

What platforms honestly cost, when to buy instead, and the failure catalogue we design against.

COST SHAPE · ORDER OF MAGNITUDE

ELEMENT	SHAPE	NOTE
Build (phases 1-4)	A mid-size platform lands in the low-to-mid hundreds of thousands	Fixed-scope phases; slices give stop points with value banked
Run: infrastructure	Hundreds to low thousands per month	Sized to load, reviewed monthly
Run: model spend	Cents per screen, dollars per assembled submission	Routing keeps judgement-priced models for judgement work
Operate and evolve	A fraction of build, annually	The brain compounds; the roadmap is jointly owned

BUILD ONLY WHAT DESERVES BUILDING

If a product does eighty percent of the job on your data with your controls, buy it; we say so in discovery and have. Build when the workflow is the business, the data cannot leave, the wall matters, or the compounding brain is the asset. The worked example is a build because the evidence library and precedent memory are Hadley's moat; the email client next to it is a purchase.

HOW THESE PROJECTS FAIL · THE CATALOGUE

FAILURE	THE TELL	THE COUNTER IN THIS METHOD
Demo-driven development	Impressive Fridays, no tests	Weekly demos run on the corpus, gates block releases
The everything-model	One vendor, one prompt, no seam	The gateway, routing by eval, exits kept open
Facts by fluency	Numbers nobody can source	The wall; citation-or-reject
Agent sprawl	Actions nobody authorised or logged	The factory: manifests, gates, kill switches
Governance at the end	A compliance scramble before launch	The register and evidence file from commit one
Pilot purgatory	Eighteen months, nothing load-bearing	Walking skeleton by week six, slices thereafter

About Red Cell Consulting

Consulting and advisory, backed by platforms we design, build and operate ourselves.

Red Cell Consulting is a multi-disciplinary consultancy working across enterprise and technology, financial analysis, clinical psychology and behavioural science, advisory and investment readiness, restructuring, management consulting and AI governance. The distinctive part is that we also build: our platforms, including Recursys for deal intelligence, Myra for deterministic financial analysis and ADMRL for structured clinical reporting, run in production under the same disciplines we recommend to clients. When we advise on evidence, testing and governance, we are describing how our own systems work.

HOW AN ENGAGEMENT RUNS

STEP	WHAT HAPPENS
1 · Conversation	A free first discussion of the situation. We will tell you plainly whether we can help and what it would involve.
2 · Scope	A short written scope: the question, the deliverables, the timeline, the fee. Fixed where the work allows it.
3 · The work	Delivered in the structure this sample shows, with working sessions rather than big reveals. You see findings as they form.
4 · Handover	Deliverables that stand on their own, and the option of standing support where it makes sense.

FOR THIS SERVICE

AI platform delivery is the discipline this whole document describes, and the Approach page it lives on shows the platforms we run under it. If there is a system your organisation keeps wishing existed, the method on these pages is how it gets built without becoming a cautionary tale. Start with the conversation.

Contact

+61 3 9448 4933 · Mon-Fri 9am-5pm Melbourne

info@redcc.au

www.redcc.au