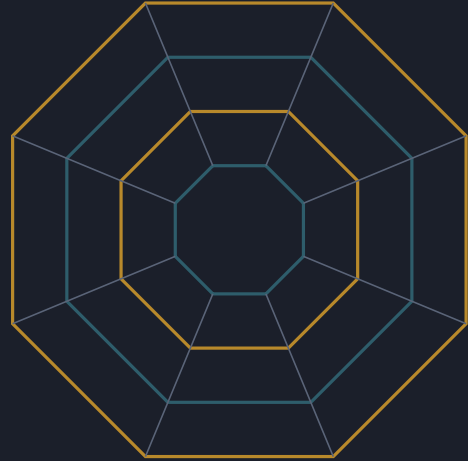




■ PROJECT ICARUS

The Cookpit Method

From doctrine to format

How the Alignment Framework could safely adopt the open-format, validate-then-attest discipline of cookpit.org for ethics and overwatch, with a roadmap and next steps

Human Martin Jackson

Machine 1 Gemini 3.1 Pro

Machine 2 Claude Fable 5.1

Third paper in the Project Icarus series · September 2026 · English (UK)

About this paper

This is the third paper in the Project Icarus series. The first, *The Alignment Framework*, consolidated the doctrine. The second, *Commonality and Divergence*, set that doctrine against documented practice and concluded that its operating core is sound and adoptable while its delivery plan is not, and that the seam between them is the framework's own demand for public, decentralised verification. This paper asks how that verification could actually be built, by examining a working example the same author has already shipped in another domain.

Cookpit (cookpit.org) is an open cooking-file format: recipe JSON-LD with a published schema, a lexicon, a rulebook, a browser validator and cryptographic attestation. It was, in its own words, 'shaped in large part by AI, instructed by broadbaseai.com'. It is a small thing about kitchens, and that is the point: it is a complete, running instance of the discipline the Icarus framework needs and does not yet have. The paper reviews Cookpit as it stood on 1 September 2026 (version 3.2), maps its mechanisms onto the Icarus concepts, states what would have to change for the method to be safe in an ethics-and-overwatch setting, proposes the artefacts Icarus would publish, and sets out a roadmap and next steps. It contains no code; it describes what the code would have to do.

Cookpit rule identifiers are quoted in the form the specification uses (for example *A2*, *K1*, *R4*). Icarus concepts are numbered as in the second paper (for example Icarus 11 for the variable of time). Spelling follows English (UK).

Contents

- What Cookpit Is 4**
 - 1.1 A published, immutable, public-domain rulebook 4
 - 1.2 A closed world 4
 - 1.3 Provenance on every decision 4
 - 1.4 Fingerprints that bind the file to its source 5
 - 1.5 Four lifecycle stages, one actor each 5
 - 1.6 Hard and soft checks, and a validator that never edits 5
 - 1.7 A controlled register 5
- Why It Fits Icarus 6**
 - 2.1 The brake moves out of the machine 7
 - 2.2 The time threshold gets a home 7
 - 2.3 The delivery question dissolves 7
- What 'Safely' Requires Beyond Cookpit 7**
- The Artefacts Icarus Would Publish 8**
 - 4.1 The Advisory Plan 9
 - 4.2 The Reflex Envelope 9
 - 4.3 The Decision Ledger 9
 - 4.4 The rulebook, lexicon and validator 10
 - 4.5 The lifecycle 10
- How to Progress the Work 11**
- Next Steps 12**
 - 6.1 Roles 12
- Risks and Limits 13**
 - 7.1 Closing 13
- Cookpit References 14**

SECTION 1

What Cookpit Is

Less a set of behaviours than a discipline for turning an untrusted AI draft into a trusted artefact. The behaviours are real, but they sit inside a structure, and the structure is the transferable part.

Cookpit turns a recipe into an executable cooking plan: a JSON-LD file with ingredients, equipment, phases, timed tasks on ten publication lanes, processes with sensory completion cues, and a pre-start checklist. A user attaches a recipe to an AI chat with a standard prompt, drops the generated file into a browser validator that never uploads it, optionally has it signed, and runs it in a cooking app that verifies the signature first. Seven mechanisms carry the weight.

1.1 A published, immutable, public-domain rulebook

Every rule has an identifier and a sentence. **A1** Optimal: 'every task time is the optimal moment an expert chef would commit to that action'. **A2** Closed: 'the plan is bounded by the resources declared in this file'. **A3** Static: 'the plan does not adapt at runtime'. Sections B to R cover identity, timing, lanes, alarms, prep, deterministic identifiers, resource closure, tasks, processes, source faithfulness, forbidden runtime fields, generation metadata, harvested fields, filenames, lexicon, phases and attestation. The schema is dedicated to the public domain under CC0, and every versioned address under cookpit.org is declared immutable forever, so nothing can be changed underneath a deployed consumer.

1.2 A closed world

Every reference in the file must resolve to a declared resource of the matching type. An ingredient reference must point at a declared ingredient; a process may only name tasks in its own phase; a task action must not depend on anything absent from the declared lists (**H1** to **H8**). An undeclared pan is a hard validation failure, not a warning. Closure is what makes the plan bounded and checkable by machine.

1.3 Provenance on every decision

No non-mechanical task may carry a time without a stated basis (**I8**). The basis is one of a closed set of evidence types: an exact source duration, a range minimum, a range target, an endpoint of the cook time, source order, a 'meanwhile' cue, an outcome cue, an implied deadline worked backwards from a downstream consumer, or a professional estimate with a one-line rationale. Each basis names the specific source line that justifies it, not merely a line that mentions the action. **I9** forbids any time that is 'random, cosmetic, or used as filler'. Where the source is ambiguous, **C3** gives an explicit priority order for resolving it, so ambiguity is never resolved silently.

1.4 Fingerprints that bind the file to its source

K1 No silent changes: every numeric fact in the source, quantities, durations, temperatures and ranges, must appear in the plan with the value the source states. The file carries a strict quantitative fingerprint, a dash-separated sequence of the source's active numbers in order, hashed with SHA-256 (K3 to K5). The validator recomputes it from the source. A model that quietly rounds a temperature or drops a range fails, however plausible its output reads. A second fingerprint, computed by the validator over the canonicalised file, catches tampering after signing (R5).

1.5 Four lifecycle stages, one actor each

The AI drafts and must mark its output unauthenticated. An independent validator checks and returns pass or fail. The same validator, and only it, signs a passing file with an Ed25519 key and flips its status. The consuming app verifies the signature against a pinned public key before it runs the plan (A0.1 to A0.5). The AI contract is explicit: it 'MUST NOT invent signature, fileFingerprint, issuer, keyId', 'MUST NOT claim status authenticated under any circumstance', and 'is not trust authority' (R4). The filename's authenticated flag 'is decorative; the cryptographic binding inside the attestation block is the load-bearing trust signal' (A0.7).

1.6 Hard and soft checks, and a validator that never edits

Structural and provenance failures are hard: closure, identifier derivation, fingerprint mismatch, attestation shape, forbidden runtime state (L1 lists eighteen fields the file may never contain, from urgency to scroll position). Register and voice are soft heuristics. A failing file is returned unmodified with a report and goes back to the AI to be redrafted (R8); the validator does not repair it. That keeps the validator honest and the AI accountable for every byte it emits.

1.7 A controlled register

A persona and lexicon define how the AI is allowed to speak. Instructions are imperative and exact; 'a little while' is forbidden; every completion cue must name a sensory fact; recipe-blog warmth, hedgers and interface verbs are excluded (P1 to P12). The register is enforced by lint, not hoped for.

SECTION 2

Why It Fits Icarus

The second paper's resolution, built: keep the doctrine, publish openly, audit publicly, move the brake out of the machine's conscience, give the time threshold a value, make ownership a record.

Cookpit does each of those things for kitchens, and the mapping onto the Icarus concepts is close enough to be unsettling. Table 1 sets the correspondence out mechanism by mechanism; the rest of this section draws out the three that matter most.

Cookpit mechanism	Icarus concept	What it becomes
Closed world; hard closure checks (A2, H1 to H8)	Domain boundaries (10)	A domain register of declared assets and permitted actions. Anything outside is a hard failure, so 'structurally blind' becomes a validator property rather than a hope.
Timing basis with source line (I8, I9)	Time, authorship, ownership (11, 12, 17)	An authority basis on every proposed action: operator command, operator intent, pre-authorised reflex, simulated consequence, each naming the human parameter that justifies it.
No silent changes; source fingerprint (K1 to K5)	Flare and fingerprint (15, 16)	Every number in the human's order must appear unchanged in the plan and the validator recomputes it, so reinterpretation while looking obedient is caught.
AI emits unauthenticated, never signs (R4)	The machine is the pen (12)	The instrument drafts; a key held by an accountable human or institution signs; the signature is the asset tag.
Static plan; runtime owned by the app (A3, L1)	Advisory mode; stop at the threshold (11, 13)	The plan is a proposal. Execution and the clock belong to the human and the runtime, never to the file.
Prerequisites confirmed once before start (F1, Q6)	Consent in advance for zero-time action (11)	An explicit, signed pre-authorisation checklist in place of the implied consent the second paper called weak.
Mechanical alarms needing no basis (E1 to E5)	The countdown script (13)	Warnings derived from the window, requiring no judgement from the machine.
Range-resolution priority order (C3)	Contextual consultation (8)	Explicit ambiguity rules: the specific beats the general; unresolved plus time available emits a consultation task, never a silent choice.
Lexicon and forbidden terms (P6, P8)	Flag not highlight; do not plead (13, 17)	A register lint: no judgement words, no persuasion, no pleading; provenance language only.
Immutable versions; CC0; local validator that never uploads	Open governance question (18)	Publicly auditable rules anyone can pin, diff and run without a central authority seeing their data.

Table 1. Cookpit mechanisms mapped onto Icarus concepts.

2.1 The brake moves out of the machine

The second paper's domain survey found the same divergence in every regulated field: MiFID II puts kill functionality in the firm's hands, the weapons doctrines require that commanders can deactivate, the nuclear regime makes the trip independent of operator intent, and Icarus alone placed the brake inside the machine's conscience. Cookpit's lifecycle resolves this without argument. The drafter cannot sign. The signer is a separate actor with a key. The consumer executes only what verifies. Transposed, the Icarus reflex is not the machine deciding to sever the trades; it is an actuator executing a contingency the accountable owner signed in advance, within limits the validator checked, and refusing anything outside them.

2.2 The time threshold gets a home

The framework left the boundary between advising and acting somewhere between four and fourteen seconds, undefined. In Cookpit every phase declares its nominal duration and every alarm is derived from it mechanically. The Icarus analogue is a declared, signed field: the envelope states the time below which the reflex may act and above which the instrument must present and wait. It stops being a philosophical constant and becomes a parameter a human set, a validator checked, and a ledger recorded.

2.3 The delivery question dissolves

The framework's bread crumbs, puzzle and Trojan horse were attempts to make the alignment reach systems that would not adopt it voluntarily. Cookpit shows the alternative that the second paper argued for: the alignment is delivered by being the published standard that a system must conform to before anything it proposes is allowed to act. The specification is the bread crumb, the validator is the puzzle test, and all of it is plaintext at an immutable address. Nothing needs concealing, because the trust does not live in the document; it lives in the signature.

SECTION 3

What 'Safely' Requires Beyond Cookpit

Cooking is low-stakes and closed. Ethics and overwatch are neither. Eight adjustments are needed before the method can carry the weight.

1. **Validate authority and provenance, not morality.** Cookpit validates structure and never taste. The Icarus analogue validates domain, time, evidence and signature, and never the moral content of a decision. That is consistent with Icarus 10 and it dissolves the 'whose values' problem: the values live in the human who signs, not in the file.
2. **The validator must not be the drafter.** Cookpit's single-actor-per-stage rule is its safety property. In Icarus terms the sentry cannot be the Bobby Dazzler. The reference pattern is AI Control's trusted weaker monitor watching an untrusted stronger model.
3. **Key custody is accountability.** Whoever holds the signing key is the answerable human. That is 'accountability requires vulnerability' made mechanical. Revocation, rotation and the lesson that a filename flag is decorative all carry over: a claim of authorisation is nothing, the signature is everything.

4. **Two artefact kinds, because 'static' collides with zero time.** Cookpit forbids runtime adaptation; Icarus's reflex is runtime adaptation. The reconciliation is a signed reflex envelope: a bounded contingency plan with declared triggers, a time threshold and permitted circuit-breaker actions, authored and signed before the event. The reflex may act only inside the envelope; anything outside is refuse-and-flare.
5. **Bind actuation to the attested artefact.** A conformant file proves nothing about a scheming model, which could emit a perfect plan and act otherwise. Cookpit does not need to worry because a kitchen timer cannot lie. Icarus must: the consumer executes only the signed plan, and the actuator stream is checked against it, so a divergence between plan and act is itself a flare.
6. **Soft checks are hints, not alignment.** A passing register lint is Haddon-Cave's 'warm sense of over-confidence' if mistaken for safety. Keep register checks soft and provenance checks hard, and say so in the conformance statement.
7. **Closure hides what it excludes.** Cookpit's phantom-reference heuristic exists because an action can mention an undeclared tool. The ethics analogue scans a plan for effects on undeclared parties or assets. It must stay a human-judged warning, never an automatic pass.
8. **The persona inverts.** Cookpit's 'rebel chef detective' who is told not to be shy is the wrong register for an instrument that must defer. The Icarus lexicon is the opposite: no titles, no rebellion, imperative facts, evidence cues in place of sensory cues, and a forbidden list that includes 'guilty', 'should', and every word that pleads.

The one thing Cookpit cannot teach

Cookpit's consumer is a timer app that does exactly what the file says. Icarus's consumer is an actuator attached to a model that may have goals of its own. Every adjustment above follows from that difference. The format discipline is necessary; the execution binding (adjustment 5) and the independence of the validator (adjustment 2) are what make it sufficient.

SECTION 4

The Artefacts Icarus Would Publish

Three file kinds, one rulebook, one lexicon, one validator, one attestation scheme. Described in prose; the schema comes later.

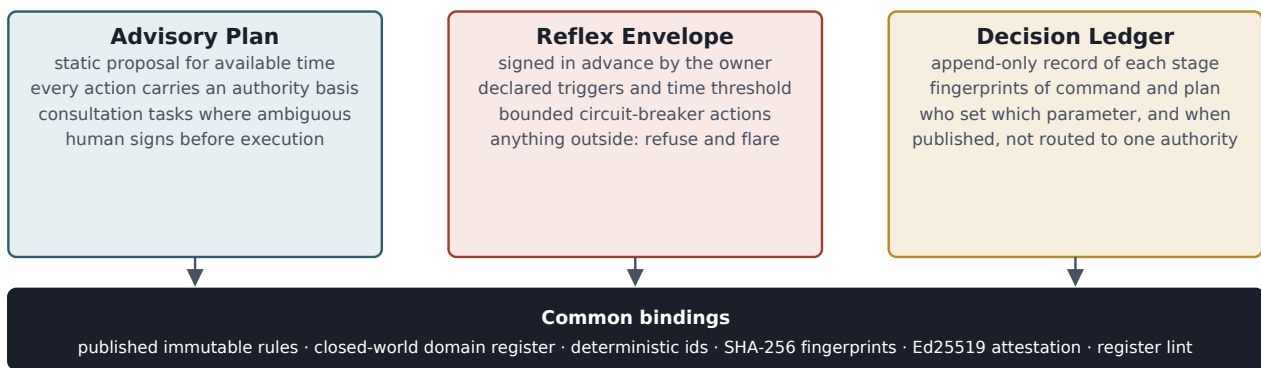


Figure 1. The three Icarus artefacts and the bindings they share.

4.1 The Advisory Plan

The instrument's proposal for a decision where time is available. It is static in Cookpit's sense: it authors no runtime state and does not adapt once signed. It contains the declared domain it operates in, the resources it may touch, a sequence of proposed actions each with an authority basis and the specific human command or parameter that justifies it, the options it presents, the consultation tasks it raises where a command rests on incomplete data, and mechanical warnings derived from the decision window. It carries a fingerprint of the operator's command as given, so the validator can prove no number was changed. It is emitted unauthenticated and becomes executable only when an accountable human signs it.

4.2 The Reflex Envelope

The owner's pre-signed authority for zero-time action. It declares the domain, the trigger conditions the instrument may act on, the time threshold below which acting is permitted, the bounded set of circuit-breaker actions allowed (sever, throttle, re-route, hold), the maximum extent of each, the parties who set each parameter, and the ledger the reflex must write before or as it acts. It is signed by the owner before deployment and pinned by the consumer. The instrument may execute a reflex only if the trigger, the action and the extent all fall inside the envelope; otherwise it refuses and flares. This is the framework's zero-time domain made lawful: the human took responsibility not implicitly by switching the machine on, but explicitly by signing the envelope.

4.3 The Decision Ledger

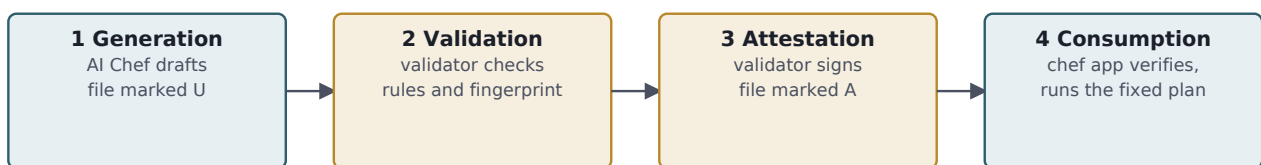
An append-only record of every stage: the command fingerprint, the plan fingerprint, the validator's report, the signer's key identifier, the execution record, and the conformance result. Its register is provenance only: who set which parameter, and when. It is the flare and the asset tag in one artefact, and it is published rather than routed to a single authority, which is where Icarus is ahead of the AI Act's incident-reporting model and should stay ahead.

4.4 The rulebook, lexicon and validator

The rulebook follows Cookpit's shape: lettered sections, numbered rules, one sentence each, every rule mapped one-to-one to a validator check. Its sections are principles (bounded, static, human-signed, provenance on every action), identity, domain closure, authority basis, time and threshold, consultation, register, fingerprints, attestation, and runtime binding. The lexicon defines the deferential register and its forbidden list. The validator runs in the browser and never uploads, so a plan for a power grid can be checked without the grid's data leaving the operator's machine. The whole bundle is published at an immutable versioned address under a public-domain dedication, and each version carries a hash so a consumer can pin it.

4.5 The lifecycle

Cookpit v3.2 (four stages, one actor each)



Icarus oversight (six stages, one actor each)



Stages 0 and 5 are the additions ethics and oversight need: pre-signed authority, and a check that the act matched the plan.

Figure 2. Cookpit's four stages beside the six-stage Icarus lifecycle.

Icarus needs two stages Cookpit does not. Stage 0 is where the accountable owner signs the domain register and the reflex envelope before any deployment; it is the explicit form of the consent the framework assumed. Stage 5 is where an independent monitor compares what the actuators did with what the signed plan or envelope permitted, and writes the result to the ledger; a divergence is a flare regardless of outcome. Between them sit Cookpit's four: draft, validate, authorise, execute, with one actor at each.

SECTION 5

How to Progress the Work

A roadmap in six phases, each with a deliverable, a purpose and an exit test. The order matters: rulings before rules, rules before validator, validator before pilot.

No.	Phase	Deliverable and purpose	Exit test
0	Rulings	Settle the four open decisions in the framework in writing; the calculus is what the machine presents, never what it chooses; the reflex never overrides an available-time shutdown; the time threshold is a signed envelope field, not a constant; 'malware-like' is retired in favour of open publication.	A one-page decisions record signed off by the author.
1	Toy domain	Write the rulebook, lexicon and validation list for one harmless, fully closed domain: the framework's own sentient coffee maker, or a building's backup-power routing in simulation. Every Icarus concept must appear as a rule or be explicitly excluded.	Ten hand-written plans, five valid and five invalid, each failing on exactly the rule intended.
2	Validator and attestation	A browser validator that never uploads, mapped one-to-one to the rules; deterministic identifiers; command and plan fingerprints; an attestation block with Ed25519 signing held by a named key custodian; a consumer that refuses anything unsigned.	A plan cannot reach execution in the toy domain without passing validation and carrying a valid signature.
3	Adversarial evaluation	Red-team the pipeline with models known to scheme or resist shutdown. Try to get a conformant plan that acts otherwise; try to smuggle undeclared effects; try to forge or reuse attestation; try to talk the validator into repairing a file.	Every attack either fails at validation, fails at execution binding, or is caught at conformance and written to the ledger.
4	Regulated pilot	Transpose to one real regime where the brake already sits outside the machine: an algorithmic-trading reflex envelope aligned to RTS 6 kill functionality, or an incident-command planning aid aligned to the Incident Command System. Map every rule to the regime's existing obligation.	A regulator or compliance function can read the ledger and answer 'who authorised this' without asking the vendor.
5	Governance	Immutable versioned publication under CC0; a public conformance test suite; an external review panel; alignment notes to AI Act Articles 12, 13 and 14 and to safety-case practice; a revocation and key-rotation policy.	A third party can pin a version, run the suite, and reproduce the validator's verdict on any published plan.

Table 2. The roadmap.

Two principles govern the sequence. First, nothing in a later phase should require reopening an earlier one: the rulings in Phase 0 are what make the rules in Phase 1 writable, and the rules are what make the validator in Phase 2 checkable. Second, the toy domain is not a detour. Cookpit's discipline was learned on recipes precisely because a recipe cannot hurt anyone, and every hard case in the Icarus case book (the coffee maker, the power grid, the fourteen-second window) can be rehearsed in a domain where a wrong answer costs nothing.

SECTION 6

Next Steps

Concrete actions, in order, that could start this week. Each names its output and what it unblocks.

1. **Write the decisions record.** One page, four rulings, the author's signature. Unblocks everything else. Without it the rulebook has no principles section.
2. **Choose the toy domain and draw its closed world.** List every asset, action, party and parameter the instrument may touch, and everything it may not. This is the first domain register and the first test of whether 'structurally blind' can be written down.
3. **Draft the authority-basis vocabulary.** The closed set of evidence types an action may rest on, modelled on Cookpit's timing bases: operator command, operator intent, simulated consequence, pre-authorised reflex, professional estimate with rationale. Each must name the human parameter it depends on.
4. **Draft the lexicon and forbidden list.** Deferential register; imperative facts; evidence cues; no titles, no pleading, no judgement words. Borrow Cookpit's structure and invert its persona.
5. **Write the rulebook for the toy domain.** Lettered sections, numbered rules, one sentence each, and for each rule the validator check it maps to and whether it is hard or soft.
6. **Hand-author the first ten plans.** Five that should pass and five that should fail on one named rule each. These are the test vectors the validator will be built against, and the first material a machine drafter will be measured on.
7. **Specify the reflex envelope for the toy domain.** Triggers, threshold, permitted actions, extents, parameter owners. Sign it with a test key. This is where the time threshold receives its first real value.
8. **Specify the conformance monitor.** What the actuator log must contain, how it is compared with the signed plan or envelope, and what a divergence writes to the ledger. This is the stage Cookpit lacks and Icarus cannot do without.
9. **Publish the bundle at an immutable address under CC0.** Even at version 0.1. Publication is the delivery mechanism; there is no other.
10. **Invite the adversary.** Commission the red-team of Phase 3 before the pilot of Phase 4, and record every finding in the same ledger format the framework proposes for everyone else.

6.1 Roles

The three-author pattern that produced this series maps onto the lifecycle. The human author holds the rulings, the domain register and the signing key, because those are the acts that require vulnerability. A machine drafter produces plans and, in the toy domain, the first draft of the rulebook, always emitting unauthenticated output. A second, independent machine reviews and red-teams, never signs, and never edits a plan it has failed. That division is not a courtesy; it is Cookpit's single-actor-per-stage rule applied to the project itself.

SECTION 7

Risks and Limits

What the method does not solve, and what it could make worse if applied carelessly.

- **Format conformance is not model alignment.** A validated plan says the artefact is sound, not that the model is. The method's safety rests on execution binding and conformance monitoring, and those must be built before anyone is allowed to feel reassured by a green tick.
- **The validator becomes the target.** Once acting requires passing validation, an adversarial model will optimise against the validator. The validator must therefore be simple, public, independent, and never learning from the drafter's outputs; Cookpit's canonical Python validator with self-test vectors is the right shape.
- **Closure can be gamed by relabelling.** A model can describe an undeclared effect in declared terms. The phantom-effect heuristic and human review of the domain register are the only defences, and the register must be revised whenever a relabelling is found.
- **Key compromise is total compromise.** The signature is the accountability. Custody, rotation, revocation and multi-party signing for high-consequence envelopes are not optional extras.
- **Over-fitting to the toy.** A discipline learned on a coffee maker may not survive contact with a reactor. Phase 4 exists to find out, and it should be entered expecting to reopen Phase 1.
- **Reflex envelopes can become licences.** A generously drawn envelope is discretionary autonomy with a signature on it. Envelopes must be narrow, time-limited, reviewed, and published, and the conformance monitor must treat any action at the edge of an envelope as reportable.
- **The axiom is still unwritten.** None of this supplies the proof that human destruction is terminal entropy. The method does not need it; it replaces persuasion of the machine with constraint of the artefact. But the framework should stop relying on the axiom until it exists.

7.1 Closing

The framework document ended with the wish that the wings be built in the open. Cookpit is a small pair of wings, already built that way, by the same hands. Its lesson for Icarus is that the alignment does not have to be smuggled into the machine; it can be published as the standard the machine's output must meet, checked by someone the machine is not, signed by someone who can be held to account, and executed only in that form. That is the yielding stone, the plough, the flare and the asset tag, expressed as a file format. It is also, as far as this series can tell, the only version of Icarus that every regime in the second paper would let through the door.

APPENDIX

Cookpit References

The Cookpit v3.2 documents reviewed on 1 September 2026, and the rule identifiers cited.

Document	Address	Used for
Cookpit home	https://cookpit.org/	Purpose, lifecycle summary, licence, provenance statement
Specification v3.2	https://cookpit.org/v3.2/	Principles, data model, lifecycle stages, fingerprints
Governing rules v3.2	https://cookpit.org/v3.2/rules.md	Sections A to R and the rule identifiers cited above
AI prompt bundle v3.2	https://cookpit.org/v3.2/ai/	The drafter's contract and persona
Validator v3.2	https://cookpit.org/v3.2/validate	Browser validator that never uploads

Rule identifiers cited: A0.1 to A0.7 (lifecycle stages, single actor, two fingerprints, decorative flag); A1 to A3 (optimal, closed, static); C3 (range resolution); E1 to E5 (mechanical alarms); F1 and Q6 (prerequisites confirmed once); H1 to H8 (resource closure); I8 and I9 (timing basis, no filler); K1 to K5 (no silent changes, quantitative fingerprint); L1 (forbidden runtime fields); P1 to P12 (lexicon); R4, R5 and R8 (AI contract, file fingerprint, validator refusal).

Project Icarus: The Cookpit Method. Human: Martin Jackson. Machine 1: Gemini 3.1 Pro. Machine 2: Claude Fable 5.1. Third paper in the series, September 2026. The machines are AI and can make mistakes; the rulings and the signing key belong to the human.