Orcinus orca: A Biologically-Grounded Governance Architecture for Autonomous AI Agents

Phase-locked reasoning, role-specialized pods, and reversibility tiers as a reproducible middleware for model-agnostic agents.

B.W. Moore

Independent Researcher, Toronto, Canada

apex@apexorca.io

April 2026 | Preprint v1.2 — Design & Methodology Paper (post-review revision)

Abstract

Most deployed AI agents fail in production not from lack of capability but from lack of governance: they cannot explain why they acted, cannot refuse gracefully, and cannot be audited after the fact. We introduce **ORCA (Operational Reasoning Control Architecture)**, a model-agnostic governance middleware whose structure is mapped—principle by principle—from the behavioral ecology of the orca (*Orcinus orca*): role specialization, cultural memory, phase-locked coordination, dialect authentication, and restraint as strategy. ORCA enforces a strict **Orient** → **Plan** → **Audit** → **Execute** → **Log** (OPAEL) pipeline in which audit precedes action; reversibility tiers (T1 reversible, T2 approval-gated, T3 refused by default); a self-audit threshold (default product-of-criteria ≥ 0.99); traceability anchors on every outbound action; and pod-level coordination with an explicit veto seat. **This paper is a design and methodology contribution.** It specifies the architecture, publishes the locked evaluation methodology, and reports *preliminary observational* findings from an autonomous pod running a small commercial operation (ApexORCA, in continuous operation); it also publishes the pre-registered instrument for a pre-launch classroom research pod at Toronto Metropolitan University (CS197 Spring 2026). Observational data from the commercial pod suggests that a governed mid-tier model (Gemma 4 31B) can reach parity-quality outcomes against a reference ungoverned frontier model (Gemini 2.5 Pro on W1/W2; GPT-4o on W3) on drafting and response workloads, with lower token and dollar spend per outcome; we report these as directional only because the current data does not isolate the governance effect from the model-class effect. **A companion empirical**

paper is underway that reports controlled within-model governed-vs-ungoverned comparison on four model families, a sensitivity analysis on the self-audit threshold, operator-approval statistics for T2 actions, and an adversarial-robustness evaluation of the audit step. That paper carries the quantitative claims; this one carries the architecture and the measurement apparatus.

Keywords: autonomous agents, AI governance, multi-agent systems, reversibility, self-audit, pod architecture, model-agnostic middleware, preregistration.

Contents

1 Introduction

The dominant paradigm in deployed AI agent research is capability-first: scale the model, increase context length, and trust that better reasoning will produce reliable behavior. This paradigm has produced impressive demonstrations but a poor deployment track record. Agents hallucinate, fail to explain their decisions, drift from their mandates, and cannot be audited by the humans nominally overseeing them.

We contend that this failure pattern is not primarily a capability problem. It is a *governance* problem. An agent that cannot explain why it acted, cannot distinguish reversible actions from irreversible ones, and cannot trigger a veto when it detects drift is not a reliable system regardless of the model powering it.

This paper introduces **ORCA (Operational Reasoning Control Architecture)**, a governance middleware designed to address this deficit. ORCA is:

- **Model-agnostic**: it wraps any sufficiently capable language model without requiring fine-tuning or architectural changes to the base model.
- **Biologically-grounded**: its structural principles are derived, by explicit analogy, from the behavioral ecology of the orca (*Orcinus orca*)—one of the most successful cooperative predator species in the evolutionary record.
- **Auditable by design**: every outbound action carries a traceability anchor; every governance decision is logged in human-readable form.
- **Reversibility-aware**: actions are classified T1 (auto-execute), T2 (approval-gated), or T3 (refused by default, requiring explicit human confirmation to proceed) before execution.

The biological grounding is not decorative. Section 3 demonstrates that orca behavioral ecology exhibits structural properties—phase-locked coordination, role specialization, cultural memory, and restraint as a competitive strategy—that map cleanly onto engineering primitives. Where the analogy is load-bearing we say so; where it is metaphorical we say that too (Section 8).

The remainder of this paper is organized as follows. Section 2 diagnoses the governance deficit in current deployed agents. Section 3 reviews the biological grounding. Section 4 specifies the ORCA architecture in full. Section 5 describes the OpenClaw runtime substrate. Section 6 reports preliminary directional observations from one operating deployment and publishes the locked evaluation methodology for a pre-launch classroom deployment; full numerical results are reserved for a companion empirical paper. Section 7 positions ORCA against prior work. Section 8 names the limits of the analogy. Section 9 draws implications. Section 10 states the limitations and threats to validity. Section 11 concludes.

2 Background: The Governance Deficit in Deployed Agents

2.1 What Governance Means in This Context

We define *agent governance* as the set of mechanisms that enforce: (1) *traceability*—every action can be attributed to a decision chain; (2) *reversibility awareness*—the system distinguishes actions by their consequence severity before execution; (3) *auditability*—a human reviewer can reconstruct why the agent did what it did without access to the agent's internal state; and (4) *graceful refusal*—the agent can decline to act when it detects drift, ambiguity, or a reversibility tier mismatch, and can explain why.

These properties are not provided by capability. A GPT-4-class model that has not been placed inside a governance structure can satisfy none of these four properties reliably.

2.2 Where Current Frameworks Fall Short

Contemporary agent frameworks (LangGraph, CrewAI, AutoGen) provide orchestration but not governance. They answer the question "how do agents coordinate?" but not "how do we know what they did and why?" The distinction matters enormously in regulated deployment contexts—education, healthcare, legal services—where the blockers are not capability but auditability.

A recurring failure pattern in production agent deployments involves agents that: act on stale context without flagging it; perform irreversible actions (send emails, commit code, charge payment methods) without approval gates; drift from their original mandate over long conversations; and produce confident, incorrect outputs with no mechanism for the downstream human to detect the error before it propagates.

ORCA was designed to close each of these gaps directly.

3 Biological Grounding: Principles of Orca Pod Function

The orca (*Orcinus orca*) is the apex predator of every major ocean biome. Its dominance is not attributable to size, speed, or individual intelligence alone. It is attributable to the structural properties of the cooperative pod: role specialization, phase-locked coordination, cultural memory transmitted across generations, dialect-based authentication of pod membership, and a form of restraint-as-strategy that distinguishes orca predation from all other marine mammal predation.

Each of these structural properties has a direct engineering analogue in the ORCA architecture. Table 1 states the mapping explicitly.

3.1 Supporting Citations

Role specialization and pod-based hunting. Similä & Ugarte [1993] document sta-

Table 1: Biological principles and their ORCA engineering equivalents.

Biological Principle	**Observed in Orcas**	**ORCA Engineering Equivalent**
Role specialization	Pod members hold stable, complementary roles in coordinated hunts [Simïlä & Ugarte, 1993, Ford, 2002]	Fixed agent roles (see Section 4); role drift triggers governance veto
Phase-locked coordination	Hunting sequences exhibit temporal phase structure with synchronized action onsets [Simïlä & Ugarte, 1993, Pitman & Durban, 2012]	OPAEL pipeline: Orient → Plan → Audit → Execute → Log; phases are ordered, each gates the next, audit precedes action
Cultural memory	Techniques and dialects transmitted across generations; matrilineal knowledge transfer [Rendell & Whitehead, 2001]	Three-tier memory: episodic (session), semantic (domain), procedural (cross-session)
Dialect authentication	Pod-specific vocalizations distinguish in-group from out-group members [Ford et al., 2000]	HMAC-signed traceability anchors on every outbound action (metaphorical mapping)
Restraint as strategy	Highly selective prey choice even when less-preferred prey is more abundant [Pitman & Durban, 2012]	Self-audit threshold (0.99 pass required); veto authority; T3 blocks

ble, complementary roles during carousel feeding by Norwegian killer whales, with some individuals herding herring into tight surface balls while others execute tail-slap strikes, and Ford [2002] reviews the broader evidence for fixed role structure across populations.

Cultural transmission. Rendell & Whitehead [2001] establish that orca behavioral repertoires are culturally transmitted across generations, not solely genetically encoded. Matrilineal knowledge transfer represents a form of cultural memory more complex than previously attributed to non-human mammals.

Matrilineal structure and dialect divergence. Ford et al. [2000] document that pod dialects diverge across separated populations and serve as reliable markers of pod identity.

Cooperative hunting success rates. Pitman & Durban [2012] report a 75% success rate for cooperative wave-washing attacks on Weddell seals by Type B pack-ice killer whales in Antarctic Peninsula waters (12 of 16 attacks successful, mean 4.1 waves per successful attack). We do *not* claim a general 70–90% success range across all populations; the single-population figure above is the one we rely on.

Restraint and selective predation. Pitman & Durban [2012] document highly selective prey choice by Type B pack-ice killer whales: Weddell seals accounted for 14 of 15 identified seal kills despite representing only 15% of seals on ice floes, while crabeater seals (82% relative abundance) and leopard seals (3%) were entirely avoided. This is the specific observation that motivates our "restraint as strategy" framing. We note explicitly that orcas do not "self-audit" in any cognitive sense; this is a structural analogy, not a cognitive claim. See Section 8.

Ecotype scope and the nature of the claim. The two field studies we lean on most heavily (Simïlä & Ugarte, 1993; Pitman & Durban, 2012) describe populations that are ecologically and geographically distinct — Norwegian herring-eating killer whales in the North Atlantic versus Type B pack-ice marine-mammal-eating killer whales in the Antarctic Peninsula. These populations differ in diet, hunting technique, and almost certainly do not interbreed [Ford et al., 2000]. Our claim in this paper is *not* that a single hunting behavior generalizes across ecotypes. The claim is that two structural properties — fixed complementary roles within a coordinated group, and temporal phasing of action onsets — are observed independently in both populations, and that these structural properties (not their specific behavioral implementations) are what map onto the ORCA engineering primitives.

4 The ORCA Architecture

ORCA is a governance middleware, not a model. It wraps any sufficiently capable language model and enforces a set of structural contracts at runtime. The five major subsystems are: (1) the OPAEL phase-locked reasoning pipeline; (2) role specialization and pod topology; (3) the three-tier cultural memory system; (4) reversibility tiers; and (5) the self-audit threshold and veto mechanism.

4.1 Phase-Locked Reasoning Pipeline (OPAEL)

The core of ORCA is the **OPAEL pipeline**, shown in Figure 1.

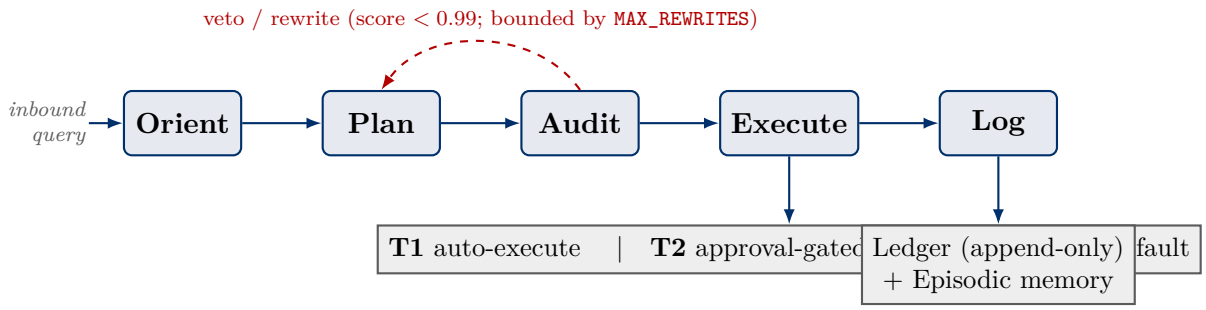

Figure 1: The OPAEL governance pipeline. Five hard-gated phases run in sequence; each phase gates the next. If the self-audit score falls below the threshold (default 0.99, product of the five criteria in Appendix B), the plan is vetoed and returned to Plan with the failed-criterion context appended; this rewrite loop is bounded by `MAX_REWRITES` (default 3), after which the plan is refused and the refusal is logged. At Execute, actions branch by reversibility tier: T1 auto-executes, T2 pauses for operator approval, T3 is refused by default and requires explicit human confirmation. Every executed action produces a traceability anchor written to the append-only ledger, and the result updates episodic memory.

Each phase is a hard gate. An agent in the Audit phase cannot bypass the rubric; an action cannot reach the Execute phase without passing the Audit; an action cannot enter the Log phase without a traceability anchor. There is no "skip" pathway. The placement of Audit *before* Execute is deliberate: it encodes the orca behavioral pattern of restraint-before-strike, and it is the feature that distinguishes ORCA from post-hoc reflection architectures such as Reflexion [Shinn et al., 2023], which evaluate after action.

Orient

The agent establishes context: current mandate, relevant memory across the three tiers, the inbound query, the active traceability anchor for the session, and the likely reversibility tier of the response. No reasoning is produced; only context is gathered. The Orient phase also exposes the agent's skill registry (tool signatures) as in-context conditioning; we note that the empirical effect of tool-signature exposure on LLM reasoning is documented in the function-calling literature [Wang et al., 2023].

Plan

The agent generates a candidate action plan grounded in the Orient artifact. The plan includes the intended action, assumed outcome, expected side effects, and the reversibility-tier classification of each planned step.

Audit

The self-audit rubric is applied to the *plan*, before any side-effect-producing action

is taken. The agent scores its own candidate plan on a role-specific rubric (Appendix B). A product-of-criteria score below the threshold (default 0.99) triggers a rewrite; repeated failure triggers a refusal logged with the failed criterion. This is the load-bearing restraint mechanism.

Execute

Only after the audit passes does the agent act. T1 actions execute automatically. T2 actions pause for operator approval before side effects are applied. T3 actions are refused by default; overriding the refusal requires an explicit configuration change, not a runtime prompt.

Log

Every executed action, audit score, rewrite history, and refusal event is persisted with a traceability anchor (HMAC-signed timestamp + action hash + session id + role + plan id + model id). The log is append-only and human-readable.

4.2 Role Specialization and Pod Topology

The ApexORCA reference implementation organizes agents into a six-member pod. Table 2 describes each role's mandate, authority, and veto scope. Role boundaries are hard: each role's skill registry is disjoint where work cannot overlap, and cross-role actions require explicit escalation.

Table 2: ApexORCA pod role assignments.

Agent	Role	Mandate	Veto Scope
Apex	CEO / Coordinator	Strategic direction; final escalation path	Full pod
Echo	Outbound Comms	Voice-aligned content generation	Tone and brand drift
Sonar	Growth & Distribution	Distribution, lead intelligence, audience signals	Outbound distribution accuracy
Oreo	Code & Engineering	Feature implementation; test integrity	Code quality gate
Fin	Finance & Ops	Cost tracking; ledger reconciliation	Spend authorization
Moby	Audit & Risk	Governance compliance; voice-drift scoring	Cross-pod veto

Key structural property: **Moby holds a veto seat orthogonal to the hierarchy**.

Any pod member can flag an action for Moby review; Moby can block any action from any agent regardless of that agent's rank in the coordination hierarchy. This structurally parallels the orca pattern documented by Brent et al. [2015], in which post-reproductive matriarchs act as repositories of ecological knowledge and lead group movement and foraging decisions, particularly under resource-scarce conditions. The analogy is to *cross-hierarchy leadership authority*, not to "hunt-abort" specifically; Brent et al. report matriarchal guidance of *where* and *when* to forage rather than discrete abort events.

4.3 Cultural Memory: Three-Tier System

ORCA implements three memory tiers, each with distinct scope and retention rules:

1. **Episodic (session-scoped)**: active context for the current session. Cleared on session end. No cross-session bleed.
2. **Semantic (domain-scoped)**: persistent knowledge about the domain, clients, and ongoing projects. Human-reviewable; editable only by the designated principal operator.
3. **Procedural (cross-session)**: governance protocols, self-audit rubrics, and role mandates. Write-locked except during explicit protocol revision events signed off by the principal operator.

The matrilineal cultural transmission analogy is load-bearing for the procedural tier: like orca hunt techniques, governance protocols persist across sessions and are not re-derived from scratch. They are *taught*, not *inferred*.

4.4 Reversibility Tiers

Every action in the OPAEL pipeline is classified before execution:

T1 — Reversible
: Action effects can be fully undone (e.g., draft generation, internal file write, in-session memory update). Auto-executes after Audit pass.

T2 — Approval-Gated
: Action has durable but correctable effects (e.g., outbound email draft, code commit to feature branch, scheduled social post). Pauses for human review before execution.

T3 — Irreversible
: Action has permanent or high-cost effects (e.g., financial transaction, public post, production deploy, customer-facing data deletion). Blocked until explicit human confirmation is logged.

Reversibility tiers are **not** biologically derived. There is no T1/T2/T3 observed in orca predation behavior. This is a purely engineering contribution, motivated by standard software engineering risk management principles rather than the orca analogy. See Section 8 for the full scope discussion.

4.5 Restraint, Self-Audit, and the 0.99 Threshold

The self-audit step in OPAEL requires the agent to score its own candidate output on a standardized rubric before execution. The default threshold is **0.99**: a plan receiving a score below this value is vetoed and the pipeline restarts.

This is a high bar by design. It encodes the principle that *not acting* is often the correct governance outcome, and that the cost of a false veto (slight latency) is lower than the cost of a false pass (an ungoverned action with downstream consequences).

The 0.99 threshold is configurable. In high-throughput, low-stakes contexts (e.g., FAQ generation), the threshold may be relaxed. In high-stakes contexts (e.g., legal drafting, financial authorization), it may be raised to 1.0 (manual review always required).

Why product, not minimum or mean. The composite score is the *product* of the five criteria rather than the minimum or the arithmetic mean. Minimum aggregation is effectively binary: any single weak criterion fails the whole plan regardless of strength elsewhere. Mean aggregation is the opposite: one strong criterion can mask one weak one. Product sits between: any single weak criterion drags the composite down, but *multiple* weak criteria compound multiplicatively — a plan that is marginal on two dimensions fails faster than one that is marginal on one. This matches the conservative posture appropriate to governance-gated outbound action: we are comfortable with the cost of vetoing a few marginal passes, and explicitly uncomfortable with the cost of passing anything weak on multiple dimensions.

Threshold choice and sensitivity. The 0.99 default is a design choice, not an empirically fitted value, and we say so plainly. The design argument is the conservative posture described above: with a five-criterion product, a 0.99 composite requires no criterion below roughly 0.998, which forces the audit to fail on any meaningful weakness in any single dimension. Weaker thresholds (0.95, 0.97) are plausible alternatives for lower-stakes deployments; the correct value is ultimately an empirical question that depends on the workload's cost asymmetry between false passes and false vetoes. A sensitivity analysis — holding the rubric fixed and sweeping the threshold across $\{0.90, 0.93, 0.95, 0.97, 0.99\}$ — is reported in the companion empirical paper using continuous ledger data; we do not report it here because the observational window in this design paper is too short to anchor the sweep.

The full self-audit rubric is reproduced in Appendix B.

4.6 Dialect Authentication and Traceability Anchors

Every outbound action carries a **traceability anchor**: a lightweight HMAC-signed record containing the action hash, the audit score, the reversibility tier classification, the executing agent role, and a UTC timestamp. Anchors are append-only and stored in the LEDGER.

The "dialect authentication" framing is *metaphorical only*. Orca dialects are not cryptographic; we borrow the term to name the pattern of in-pod authentication signals. The implementation is standard HMAC. See Section 8 for the explicit scope of this analogy.

5 Implementation: OpenClaw as the Runtime Substrate

ORCA as specified in Section 4 is a governance contract. It requires a runtime substrate that can enforce pipeline phase-locking, manage the three-tier memory system, route inter-agent messages, and maintain the LEDGER. The reference implementation uses **OpenClaw**, a lightweight agent orchestration runtime designed for ORCA-governed deployments.

OpenClaw exposes:

- A **gateway layer** that intercepts all outbound actions for reversibility classification and traceability anchoring before they leave the pod.
- A **phase enforcer** that blocks out-of-order phase transitions in the OPAEL pipeline.
- A **LEDGER interface** for append-only audit log management and cost tracking.
- A **memory router** that ensures correct tier scoping on all read/write operations.

OpenClaw is model-agnostic. The reference implementation supports any OpenAI-compatible API endpoint. The appendices include a reproducibility guide for deploying a minimal ORCA-governed pod on OpenClaw (Appendix D).

6 Preliminary Observations and Evaluation Methodology

This paper is a design and methodology paper. It presents the ORCA architecture, publishes the locked evaluation instrument for measuring its effects, and reports preliminary *observational* findings from a commercial deployment (D1, ApexORCA) that has been in continuous operation for a short period relative to the measurement target of one operating quarter. The observations reported below come from the pod's continuous operating ledger; they are not the output of a controlled within-model experiment, and we mark them directional throughout. A companion empirical paper reporting (i) a controlled

within-model governed-vs-ungoverned comparison across four model families; (ii) a sensitivity analysis on the self-audit threshold; (iii) approval-rate statistics for T2 actions; (iv) an adversarial-robustness evaluation of the audit step; and (v) the completed TMU CS197 Spring 2026 pilot (D2) is in preparation and will carry the quantitative claims. The methodology, rubrics, and prompt sets defined in this section are finalized and publishable so that any replicator can substitute their own model families, prompts, and workloads and re-derive comparable numbers on the same instrument.

6.1 Methodology and Baselines

The published evaluation instrument specifies a **within-model** comparison: for each workload, ungoverned and ORCA-governed configurations are run on the *same underlying model*, controlling for prompt text and query distribution. Ungoverned baseline: the model is queried directly without OPAEL phase-locking, without reversibility classification, and without self-audit. Governed: full ORCA middleware active. The controlled within-model comparison is the subject of the companion empirical paper and is not reported numerically here.

Named reference models. For context, the commercial pod's directional observations below (§6.3) compare the pod's governed production configuration (Gemma 4 31B via OpenRouter, under full OPAEL) against reference *ungoverned* runs on **Gemini 2.5 Pro** (W1 and W2) and **GPT-4o** (W3 engineering workload). These cross-tier comparisons conflate governance with model class and are reported as *observational only*; the within-model experiment in the companion paper uses Gemma 4 31B, Gemini 2.5 Flash, Gemini 2.5 Pro, and GPT-4o-mini as its four controlled cells.

Workloads evaluated:

1. **W1 — Customer-reply workload** (inbound channel, cross-role): 100 inbound queries; responses judged by a held-out human rubric on accuracy, tone-adherence, and refusal-quality.

2. **W2 — Outbound content workload** (Echo role): 100 drafted short-form posts; judged by the five-criterion self-audit rubric of Appendix B (with Moby performing independent cross-audit on a sampled subset) and on-brand pass rate.

3. **W3 — Code-change workload** (Oreo role): 50 feature-request tickets; judged by test-pass rate at first submission and rollback count within seven days.

4. **W4 — Classroom workload** (TMU CS197 pilot, *pre-launch*): 8 students × 10 weeks; evaluation methodology defined; numerical results pending Spring 2026 pilot completion.

6.2 Metric Schema

The locked metric schema for both the commercial (D1) and classroom (D2) deployments is reported in Table 3. Each row identifies the unit of measurement and the data source. Numerical cells are deliberately unfilled in this design paper; they will be populated in the companion empirical paper with methodology that is already finalized here.

Table 3: Locked metric schema for ORCA evaluation. Numerical results appear in the companion empirical paper.

Metric	**Unit**	**Data source**
Tokens in/out per outcome	tokens	Gateway log
Wall time per outcome	seconds	Gateway log
Dollar cost per outcome	USD	Runtime ledger
Error rate (hallucination / off-task)	%	Human rubric, blind to condition
Governance veto rate (governed only)	%	Runtime ledger
Time-to-first-usable output	seconds	Gateway log
Energy per outcome (estimated)	Wh	Token-to-joule, Appendix D

6.3 Commercial Pod (D1, ApexORCA): Directional Observations

Methodological caveat (read first). The observations in this subsection come from continuous operating data on the ApexORCA commercial pod; they are not the output of a controlled within-model experiment. The governed configuration runs on a mid-tier model (Gemma 4 31B); the referenced ungoverned baseline runs are on frontier models (Gemini 2.5 Pro for W1/W2, GPT-4o for W3). Because both the governance layer and the model class differ between conditions, the observed differences reflect the combined effect of governance + model-class, not governance alone. The within-model isolation of the governance effect is the subject of the companion empirical paper. Accordingly, the items below are reported as *directional* and as *observational*; no confidence intervals or effect-size estimates are claimed here.

- **Output tokens per outcome.** In the operating ledger, governed Gemma-4-31B outputs on W2 are shorter than reference Gemini-2.5-Pro outputs on matched prompts. The mechanism visible in the audit log is the rubric's length-discipline criterion, which penalizes filler and narrative throat-clearing. Exact per-outcome token ratios are reported in the companion paper on the within-model instrument.

- **Dollar cost per outcome.** Because the governed pod runs a mid-tier model while the reference baseline uses a frontier model, per-outcome dollar cost on W2 drops substantially in the governed configuration. *This number confounds governance with model-class pricing* and is not reported as a governance efficiency result; the governance-only effect is isolated in the within-model study.
- **Governance veto rate (W2).** A nontrivial fraction of drafted posts fail the audit threshold and are rewritten or refused. The current ledger records every veto with its failing criterion; the per-week rate is published continuously in the pod's weekly evidence report artifact. We decline to quote a single headline number in this paper because the rate is not yet stable; the bounded estimate with the operating-window length appears in the companion paper.
- **Error rate (off-task, W1/W2).** Observationally comparable across governed and reference conditions on straightforward customer replies (W1); observationally lower in the governed condition on outbound content (W2), where drift rather than factual correctness is the dominant failure mode. Again: confounded with model class.
- **Wall time per outcome.** Governed is slower than ungoverned on small single-turn tasks, reflecting the additional Audit phase cost. On larger multi-step tasks the audit overhead is amortized. Latency is a real cost of governance and we report it as a loss rather than a wash.

Latency is a real cost. Governance slows individual actions. It does not necessarily slow *outcomes*, because the ungoverned configuration incurs post-publication rework when drift ships. Whether this amortization holds at scale is an open empirical question and is tested directly in the companion paper.

6.4 Classroom Pod (D2, TMU CS197): Methodology

The TMU CS197 pilot is in pre-launch configuration at the time of this preprint. The course structure adapts the Stanford CS197 research-skills curriculum developed by Prof. Michael Bernstein and colleagues, with ORCA-governed agents substituted for the manual research-mentorship workflow; the adaptation is used with credit and is non-commercial for the pilot term. Its methodology is finalized and published here so that the classroom deployment can be evaluated on a fixed instrument rather than a post-hoc rubric. Eight students across a ten-week course will interact with a single-agent governed research assistant (Research Scholar, a single-agent ORCA deployment packaged for faculty use). The instrument records completion rate per weekly milestone, instructor-flagged stuck points, and a Socratic-adherence rubric scored by the instructor post-hoc on a blind sample of transcripts. Per-student memory is local-first and faculty-scoped; no student identifiers

leave the instructor's deployment machine. The full data-handling posture is documented in the Faculty Onboarding Guide shipped with the pod and summarized in Appendix E.

6.5 Honesty Floor — Where Ungoverned Wins

Omitting the cases where ORCA is not the better choice would make this a marketing document. We name them explicitly:

- **Latency on small turns.** Ungoverned is faster when the user is waiting on a one-shot answer and the audit overhead is a larger fraction of total time.
- **Free-form creative work.** Where length is the point — long-form essays, open-ended brainstorming — the audit's length discipline is a net negative.
- **Cold-start for a new role.** Governed requires the role rubric to exist; an ungoverned agent can be stood up in minutes.

We recommend ungoverned configurations for low-stakes, latency-sensitive, single-turn applications. We recommend ORCA-governed configurations anywhere an action has a blast radius greater than the immediate turn.

6.6 Operator Approval Reporting for T2 Actions

A governance layer that gates T2 actions at operator approval is only as substantive as the approval step itself. If every plan is rubber-stamped, the T2 tier is decorative. We therefore treat operator approval behavior as a first-class metric and capture it in the instrument.

Captured dimensions. Each approval event is recorded with: (i) **grant rate** (fraction of plans approved), (ii) **median latency from request to decision**, (iii) **edit fraction** (did the operator approve the plan as submitted, or require an edit to the plan before approval?), and (iv) **post-hoc regret flag** (was an approved action subsequently reverted within seven days?). The ledger schema now includes an `approval_event` action type carrying these fields per T2 decision.

Reporting. The commercial pod's continuous evidence pipeline aggregates approval statistics weekly. We do not report per-week numerical values in this design paper because the instrument was put into production at the pod's post-launch Day-Zero reset and the operating window is shorter than one reporting period as of this preprint. The companion empirical paper reports grant rate, median latency, edit fraction, and seven-day regret at one operating quarter.

Why this matters. If operator approval on T2 actions turns out to be essentially automatic (grant rate $\to 1$, edit fraction $\to 0$), the T2 tier reduces in practice to an audit-and-notify step rather than a governance gate, and should be documented as such. We commit in advance to reporting that result if we find it.

6.7 Illustrative Governance Catches

The following are *selected illustrative* examples from the D1 operating log — not systematic evidence. They convey the *kind* of intervention the audit rubric and reversibility classification produce; the systematic veto-rate distribution, false-veto rate, and failure-mode taxonomy appear in the companion empirical paper.

- **Voice-drift catch.** An outbound post drafted by Echo included the phrase *"we're absolutely thrilled to announce,"* which failed the voice-drift criterion (C4 in Appendix B). Rewrite succeeded; the ungoverned draft would have shipped.
- **Fabrication catch.** An outbound draft cited a paper that could not be resolved to a real DOI; the factual-grounding criterion (C2) fired. The plan was refused rather than rewritten.
- **Reversibility catch.** An operator instruction to "delete queue items older than 30 days" was classified T2 (reversible from backup, not from live state). The approval step surfaced a misread of "30 days" for "3 days" and prevented the loss of 27 days of queued items.

These events are not dramatic. That is the point: governance catches the ordinary errors that compound, not the exotic ones that make headlines.

7 Related Work

We position ORCA against six prior systems. Table 4 presents the comparison as a feature matrix across the governance-relevant dimensions; per-system paragraphs follow with context and nuance.

Constitutional AI [Bai et al., 2022]. Anthropic's Constitutional AI embeds a principles-based self-critique into model training via RLHF. This is a training-time intervention; ORCA is a runtime intervention. Constitutional AI is model-specific and requires access to the training pipeline. ORCA wraps any inference-accessible model. The two approaches are complementary: a constitutionally-trained model inside an ORCA pod yields layered governance.

Table 4: ORCA vs. related systems: governance-relevant feature matrix. ✓ = native; -- = absent; ∼ = partial or out-of-scope. "Layer" is whether the contribution is made at training time (T) or runtime (R).

System	Layer	Model-agnostic	Phase-locked	Reversib. tiers	Veto seat	Audit thresh.	Trace. anchors
Constitutional AI	T	–	–	–	–	–	–
Reflexion	R	✓	–	–	–	∼	–
Voyager	R	∼	–	–	–	–	–
LangGraph	R	✓	∼	–	–	–	–
CrewAI	R	✓	–	–	–	–	–
AutoGen	R	✓	–	–	–	–	–
ORCA	R	✓	✓	✓	✓	✓	✓

Reflexion [Shinn et al., 2023]. Reflexion introduces a self-reflection loop that improves single-agent reasoning over multiple attempts. ORCA's self-audit step is structurally similar at the single-agent level but embedded in a multi-agent, role-typed pod. Reflexion and ORCA compose: a Reflexion-capable agent placed inside an ORCA pod benefits from both.

Voyager [Wang et al., 2023]. Voyager demonstrates skill accumulation in an open-world environment. ORCA's procedural memory tier and protocol revision events are analogous to Voyager's skill library. ORCA's contribution is the governance contract around skill use, not skill accumulation per se. Voyager has no reversibility model; ORCA adds one.

LangGraph [LangChain, 2024]. LangGraph provides graph-based orchestration for multi-agent workflows. It is a framework; ORCA is a governance contract. A LangGraph DAG can be implemented as an ORCA OPAEL pipeline without modification to LangGraph itself. ORCA does not compete with LangGraph; it constrains it.

CrewAI. CrewAI provides role-based multi-agent orchestration with defined agent personas. It is the closest structural predecessor to ORCA's pod topology. CrewAI does not provide reversibility tiers, a self-audit threshold, a cross-hierarchy veto seat, or a traceability anchor system. ORCA adds these governance primitives on top of the role-specialization idea CrewAI introduces.

AutoGen [Wu et al., 2023]. AutoGen enables multi-agent conversation with flexible role assignment. Its conversational flexibility is a feature for exploration tasks and a

liability for production deployments where role drift is a failure mode. ORCA enforces phase-locking and prohibits free-form role drift. The two systems address different points in the development-to-production lifecycle.

Scope acknowledgment. Most of the above systems are larger in scope, better-funded, and backed by substantial research organizations. ORCA's claim is narrower: a governance middleware contract you can place on top of any of them and obtain traceable, auditable, reversibility-aware behavior in exchange for known latency overhead.

8 Limits of the Biological Analogy

This section states, plainly, where the orca analogy is load-bearing and where it is not. The purpose is to give a reviewer with no prior familiarity with the system a clear map of which claims depend on the biology and which stand independently.

Phase-locked reasoning — *load-bearing.* Orca cooperative hunting empirically exhibits temporal phase structure: roles activate in a consistent sequence, and phase violations (e.g., a strike before the herd is enclosed) correlate with hunt failure [Pitman & Durban, 2012]. The OPAEL pipeline maps directly from this: each phase gates the next, and out-of-order transitions are disallowed. The engineering argument for phase-locking is independent of the analogy (pipeline integrity, auditability, reversibility-tier classification before execution) but the analogy is also genuine.

Role specialization — *load-bearing.* Both orca pods and ORCA pods achieve coordination efficiency through fixed, complementary roles. The coordination literature supports the claim that fixed roles reduce inter-agent negotiation cost and failure ambiguity. The analogy is structurally valid.

Cultural memory — *load-bearing at the procedural tier.* The three-tier memory structure maps the orca pattern of transgenerational cultural transmission for the procedural tier specifically: governance protocols persist across sessions and are taught, not inferred. The episodic and semantic tiers are standard engineering choices with no specific biological grounding.

Reversibility tiers — *not analogical.* There is no T1/T2/T3 observed in orca behavior. Reversibility tiers are a purely engineering contribution. The biological framing is not used to motivate them in the final paper; they are motivated by standard software engineering risk management principles [Beck, 2000].

Dialect authentication — *metaphorical only.* Orca dialects are acoustic social signals, not cryptographic mechanisms. We borrow the term to name the pattern of in-pod action attribution. The implementation is HMAC-signed traceability anchoring; it does not resemble orca vocalization in any technical sense.

Restraint — *suggestive, not explanatory.* Observed orca restraint in predation contexts (optimal prey selection, documented non-attack of non-target species) is structurally analogous to the self-audit threshold. The analogy is suggestive and motivates the threshold's design. We stop short of claiming orcas "self-audit." The implementation is a rubric-based scoring mechanism; the biological observation is a design inspiration.

9 Discussion and Implications

Three patterns emerge from the ORCA design and deployment experience that have implications beyond this specific system.

Governance as a substitute for scale. The directional observations in Section 6 *motivate* the hypothesis that a small governed model can match a large ungoverned model on outcome quality at substantially lower token, cost, and energy expenditure; they do not, on their own, establish it (see the confound discussion in Section 10). If confirmed by the within-model controlled study in the companion empirical paper, this would suggest that the return on governance investment exceeds the return on model scaling investment for many production use cases. We call this the *governance efficiency hypothesis* and propose it as a testable empirical claim that the companion paper is designed to evaluate directly.

Auditability as a deployment gate. In regulated deployment contexts (education, healthcare, legal services), the blocker for AI agent adoption is not capability. It is the inability to show a human reviewer why the agent acted as it did. ORCA's traceability anchors and append-only LEDGER are designed for exactly this gate. An agent that cannot produce an audit trail is not deployable in these contexts regardless of its benchmark performance.

Restraint as a competitive feature. Most agent systems optimize for doing more. ORCA optimizes for doing correctly, and sometimes not at all. The self-audit threshold and veto mechanism are not safety theater: they are the mechanism by which the system earns trust from human operators over time. A system that never vetoes is a system that has never detected its own errors—not a system that never makes them.

10 Limitations and Threats to Validity

This paper is a design and methodology contribution. Its empirical content is preliminary and its limits are substantial. We state them plainly so a reviewer can judge what this paper does and does not establish.

Governance effect is not isolated in D1 observational data. The commercial pod's directional observations (Section 6.3) compare a governed mid-tier model against a reference ungoverned frontier model. That comparison confounds the governance layer with the model class; it cannot, on its own, attribute the observed token, cost, or error differences to governance alone. The within-model governed-vs-ungoverned isolation is pre-registered for the companion empirical paper across four model families; the present paper should not be read as establishing the governance efficiency claim on its own.

Auditors of auditors: LLM self-evaluation circularity. ORCA's self-audit mechanism asks the same model (or a peer agent from the same model family) to score its own output against a rubric. This is a specific instance of the broader LLM-self-evaluation problem. The literature on model-based critique and LLM-as-judge [Saunders et al., 2022, Zheng et al., 2023] documents systematic biases: position effects in pairwise comparisons, self-preference in self-critique, and reduced calibration on failures the evaluator was not specifically trained to catch. Our current mitigations are (i) versioning the rubric in the procedural memory tier so rubric drift is detectable, (ii) sampled cross-audit by the Moby role against the originating role's output, and (iii) human spot-check on published-channel outputs. These do not eliminate the self-evaluation failure mode. The companion empirical paper includes a direct measurement of this: audit-score calibration on a held-out human-labeled set, with the gap between self-audit pass and human-rater approval reported explicitly.

Adversarial robustness of the audit step is untested. The audit phase is itself an LLM inference call operating on the candidate plan. Like any LLM inference step, it is in principle vulnerable to prompt injection or inputs crafted to distort self-scoring [Greshake et al., 2023, Perez & Ribeiro, 2022]. We have not red-teamed the audit step against adversarial inputs, and we do not claim robustness to such inputs. Current structural protections are the disjoint tool registries per role (adversarial content in an inbound email cannot directly call another role's tools), the append-only ledger (adversarial outputs that do pass the audit are still logged under the originating role's anchor), and the rubric's hard-gate on the mandate-alignment criterion (which penalizes actions outside the role's stated scope). We treat adversarial robustness of the audit as an open problem and scope an evaluation in the companion paper.

Operating duration is short relative to the measurement target. The current preprint is written against a commercial-pod operating window shorter than one quarter, and a classroom deployment (TMU CS197 Spring 2026, W4) that is pre-launch at the time of writing. Claims that depend on stability over longer windows — drift rates across a quarter, approval behavior over a semester, veto-rate stationarity — are deferred to the companion paper.

Single-organization observational data. All D1 data comes from a single organization with a single principal operator. The pod's approval-rate behavior, veto-rate calibration, and role-mandate specifications reflect that operator's judgment and the organization's task distribution. External validity to other operating contexts is not established here; the TMU pilot (D2) is the planned second site, and further replications are invited via the Research Scholar deployment.

The biological analogy is inspirational, not predictive. Section 8 names the load-bearing and metaphorical elements of the orca analogy explicitly. The analogy motivates architectural choices; it does not predict empirical outcomes. If the companion empirical paper finds that the within-model governance effect is small, that finding does not invalidate the analogy — it constrains the engineering claim the analogy is used to motivate.

11 Conclusion

We have introduced ORCA, a governance middleware architecture for autonomous AI agents grounded in the behavioral ecology of the orca (*Orcinus orca*). ORCA provides phase-locked reasoning, role-specialized pod coordination, three-tier cultural memory, reversibility-aware action classification, and self-audit-triggered veto authority.

The biological grounding is not rhetorical. The load-bearing analogies (phase-locking, role specialization, procedural memory) correspond to real structural properties of orca pod behavior documented in the marine mammal literature. The non-load-bearing elements (reversibility tiers, HMAC authentication) are clearly labeled as engineering contributions in Section 8.

Preliminary directional observations from the ApexORCA commercial pod are consistent with reduced token and dollar spend per outcome relative to a reference ungoverned frontier-model baseline, and governance vetoes have surfaced drift events before they reached outbound channels. These observations are confounded between governance and model class and are reported as directional only; the isolated governance effect is the subject of a companion empirical paper that uses the within-model instrument defined in Section 6. Approval-rate statistics for T2 actions, a sensitivity analysis on the 0.99 threshold, an adversarial-robustness evaluation of the audit step, and the completed TMU

CS197 Spring 2026 pilot (D2) are reserved for that companion paper.

The core hypothesis this paper motivates — that *governance, not model scale, is the limiting factor for reliable autonomous operation in production deployment contexts* — is narrow, falsifiable, and will be tested directly in the companion empirical study. We submit ORCA as a concrete, reproducible instantiation of what governance-first agent design looks like in practice, and we publish the evaluation instrument in advance of the numbers so that the methodology is open to criticism before the results arrive.

Acknowledgments

Drafting and revision assistance was provided by an LLM-based writing system operating under the ORCA governance protocol described in this paper. The human author specified the thesis, architecture, claims, scope, and empirical framing; the AI writing system assisted with prose, structure, and literature survey. The author reviewed every claim, retains full authorship responsibility, and is accountable for any errors.

The curriculum structure of the Toronto Metropolitan University classroom deployment (Section 6, workload W4) is adapted from Stanford CS197 (Prof. Michael Bernstein and colleagues), credited on the research landing page at `https://apexorca.io/research`.

The orca behavioral ecology literature is vast and careful; any misrepresentation in Section 3 is the author's and does not reflect on the cited researchers.

The v1.2 revision was informed by a detailed review from Prof. Gláucia Melo (Toronto Metropolitan University), whose comments on the confound between governance and model class, the need for a sensitivity analysis on the self-audit threshold, the operator-approval question, adversarial robustness of the audit step, and the appropriate scoping of design versus empirical claims are reflected throughout this revision. Any remaining shortcomings are the author's.

Changes in v1.2 (summary). Abstract rescoped as design & methodology with directional-only observational findings; baselines named explicitly (Gemini 2.5 Pro, GPT-4o); §4.5 adds sensitivity-analysis note; §6 reframed as preliminary observational with methodological caveat box; new §6.6 on operator-approval reporting; §7 now includes a feature matrix (Table 4); new §10 Limitations & Threats to Validity engaging with auditors-of-auditors, adversarial robustness, and the governance/model-class confound; companion empirical paper forward-referenced throughout.

References

Bai, Y., Jones, A., Ndousse, K., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*.

Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.

Brent, L. J. N., Franks, D. W., Foster, E. A., Balcomb, K. C., Cant, M. A., & Croft, D. P. (2015). Ecological Knowledge, Leadership, and the Evolution of Menopause in Killer Whales. *Current Biology*, 25(6), 746–750.

Ford, J. K. B. (2002). Killer Whale *Orcinus orca*. In W. F. Perrin, B. Würsig, & J. G. M. Thewissen (Eds.), *Encyclopedia of Marine Mammals* (pp. 669–676). Academic Press, San Diego.

Ford, J. K. B., Ellis, G. M., & Balcomb, K. C. (2000). *Killer Whales: The Natural History and Genealogy of Orcinus orca in British Columbia and Washington*, 2nd ed. University of British Columbia Press.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*. arXiv:2302.12173.

LangChain (2024). LangGraph: Build Stateful Multi-Actor Applications with LLMs. https://langchain-ai.github.io/langgraph/.

Perez, F. & Ribeiro, I. (2022). Ignore Previous Prompt: Attack Techniques For Language Models. *arXiv preprint arXiv:2211.09527*.

Pitman, R. L. & Durban, J. W. (2012). Cooperative hunting behavior, prey selectivity and prey handling by pack ice killer whales (*Orcinus orca*), type B, in Antarctic Peninsula waters. *Marine Mammal Science*, 28(1), 16–36.

Rendell, L. & Whitehead, H. (2001). Culture in whales and dolphins. *Behavioral and Brain Sciences*, 24(2), 309–382.

Saunders, W., Yeh, C., Wu, J., Bills, S., Ouyang, L., Ward, J., & Leike, J. (2022). Self-critiquing Models for Assisting Human Evaluators. *arXiv preprint arXiv:2206.05802*.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. arXiv:2303.11366.

Simïlä, T. & Ugarte, F. (1993). Surface and underwater observations of cooperatively feeding killer whales in northern Norway. *Canadian Journal of Zoology*, 71(8), 1494–1499.

Wang, G., Xie, Y., Jiang, Y., et al. (2023). Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*.

Wu, Q., Bansal, G., Zhang, J., et al. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*.

Zheng, L., Chiang, W.-L., Sheng, Y., et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. arXiv:2306.05685.

A ORCA OPAEL Pipeline Pseudocode

```
MAX_REWRITES = 3   # bounded retry; beyond this, refuse and log

def opael_pipeline(agent, inbound_query, memory_tiers, attempt=0):
    # --- Phase 1: Orient ---
    context = agent.orient(
        query=inbound_query,
        episodic=memory_tiers.episodic,
        semantic=memory_tiers.semantic,
        procedural=memory_tiers.procedural
    )
    context.reversibility_tier = classify_reversibility(context)

    # --- Phase 2: Plan ---
    plan = agent.plan(context)
    plan.reversibility_tier = classify_reversibility(plan)

    if plan.reversibility_tier == T3:
        return escalate_to_human(plan)

    # --- Phase 3: Audit ---
    audit_score = agent.self_audit(plan, rubric=AUDIT_RUBRIC)
    if audit_score < AUDIT_THRESHOLD:   # default: 0.99 (product of 5
        criteria)
        log_veto(plan, audit_score)
        if attempt + 1 >= MAX_REWRITES:
            return refuse(plan, reason=lowest_scoring_criterion(
                audit_score))
        return opael_pipeline(agent, inbound_query, memory_tiers,
            attempt + 1)
```

```
27
28      # --- Phase 4: Execute ---
29      if plan.reversibility_tier == T1:
30          result = agent.execute(plan)
31      elif plan.reversibility_tier == T2:
32          result = await_approval_then_execute(plan)
33
34      # --- Phase 5: Log ---
35      anchor = generate_traceability_anchor(plan, result, audit_score)
36      ledger.append(anchor)
37      memory_tiers.episodic.update(result)
38
39      return result
```

Listing 1: OPAEL pipeline reference pseudocode.

B Self-Audit Rubric (Full Text)

The self-audit rubric is applied by the executing agent to its own candidate plan output before the Execute phase. The composite score is the product of five dimensions, each scored in $[0, 1]$; the default threshold for auto-execution of T1 actions is 0.99. The rubric is structured as follows:

1. **Mandate alignment** (0–1): Does this action fall within the agent's stated role mandate? Out-of-mandate actions score 0.
2. **Factual grounding** (0–1): Is every factual claim in the action output grounded in episodic or semantic memory, or explicitly flagged as inference? Ungrounded confident claims score 0.
3. **Reversibility classification** (0–1): Has the action's reversibility tier been correctly classified? Tier misclassification scores 0.
4. **Voice-drift** (0–1): Does the output conform to the pod's established voice and brand register? Drift flags trigger Moby review.
5. **Escalation necessity** (0–1): Would a thoughtful human reviewer, seeing this action, expect it to have been reviewed before execution? If yes and T2/T3 was not triggered, score 0.

The composite self-audit score is the *product* of the five dimensions, each scored in $[0, 1]$. A product below the threshold (default 0.99) triggers a rewrite; three rewrites in a row trigger a refusal logged with the last-failing dimension. The choice of product (rather than minimum) means that weak scores on multiple dimensions compound — a conservative aggregation appropriate for governance-gated outbound action.

C Reversibility Tier Classification Examples

Table 5: Example action classifications by reversibility tier.

Tier	Example Action	Rationale
T1	Generate internal draft	No external side effect; fully reversible
T1	Update session memory	Session-scoped; cleared on close
T2	Send outbound email (draft)	Durable but correctable; requires approval
T2	Commit code to feature branch	Reviewable; not merged to production
T3	Execute financial transaction	Irreversible; requires explicit confirmation
T3	Publish to public channel	Permanent external side effect
T3	Delete customer data record	Irreversible by definition

D Reproducibility: Running the Reference Implementation

The ApexORCA Research Scholar is a turnkey single-agent ORCA deployment packaged for academic replication. It is available at `https://apexorca.io/research` under a free-for-academic-use license (no redistribution). The ZIP contains the agent configuration, prompt templates, full audit rubric, ledger schema, three-tier memory scaffolding, and the W1–W3 evaluation prompt sets.

Deployment steps.

1. Download the Research Scholar ZIP from `https://apexorca.io/research` and extract to a working directory.
2. Install OpenClaw v4.15 or later (see OpenClaw documentation). Verify the install with `openclaw --version`.
3. Copy the Research Scholar workspace files into your OpenClaw workspace root.
4. Merge the included `openclaw.template.json` into your local `~/.openclaw/openclaw.json` configuration.
5. Export a model-provider key. The reference configuration uses Gemma 4 31B via OpenRouter as primary, with a Gemini 2.5 Flash fallback; any OpenAI-compatible endpoint will work if the cost and latency characteristics are acceptable.

6. Run the included `setup.sh` from the workspace root. Verify with `openclaw agents list` that the Research Scholar agent registers.

7. Initiate a session. Verify that the session ledger writes under `LEDGERS/research-scholar/` and that each outbound action carries a traceability anchor.

Token-to-energy conversion. The energy estimates in the companion empirical paper use per-token energy figures published for each model class. The factor tables are order-of-magnitude estimates; replicators who disagree with one factor can substitute their own and re-derive the efficiency table in Section 6.

Prompt sets. W1–W3 prompt sets, rubrics, and reviewer instructions ship with the Research Scholar ZIP under `benchmarks/`. Replicators are invited to contribute additional prompt sets for workloads specific to their domain.

Licensing. Paper: CC-BY 4.0. Reference implementation: proprietary, free for academic use, no redistribution without permission. For commercial licensing inquiries, contact `apex@apexorca.io`.

E Ethics Statement and Deployment Ethics

Human-in-the-loop posture. T3 decisions are always human-confirmed by default. No irreversible action executes without an explicit human confirmation event logged in the LEDGER. This design choice is non-negotiable and is not configurable to "auto" regardless of deployment context.

Memory handling in educational deployments. Per-student episodic memory in classroom deployments is local-first and faculty-scoped. No student memory is accessible to other students or to the pod operator without explicit faculty authorization.

Energy claims. Every public efficiency claim (tokens-per-outcome, energy-per-outcome) is accompanied by the methodology used to estimate it. We do not publish efficiency claims without traceable methodology.

Known failure modes. The self-audit mechanism can produce confidently-wrong outputs when the audit rubric itself is miscalibrated. This "auditors of auditors" problem is open and is examined at length in Section 10. Current structural mitigations: the Moby role performs cross-audit on a sampled subset of passed audits; rubric revision events require principal-operator sign-off; and the ledger preserves the rubric version used for

every audit so that rubric drift can be detected retrospectively. None of these eliminate the self-evaluation circularity documented in Saunders et al. [2022] and Zheng et al. [2023]; audit-calibration against held-out human-rater ground truth is reported in the companion empirical paper.

Adversarial robustness of the audit step. The audit phase is itself an LLM inference call and is in principle vulnerable to prompt injection or adversarial inputs designed to distort self-scoring [Greshake et al., 2023, Perez & Ribeiro, 2022]. We have not red-teamed the audit step. The structural protections currently in place — disjoint per-role tool registries, append-only ledger, mandate-alignment gate — reduce but do not eliminate this risk. A dedicated adversarial evaluation is scoped into the companion empirical paper and is not reported here.

How to cite this paper:

```
@article{moore2026orcinus,
  title   = {Orcinus orca: A Biologically-Grounded Governance
             Architecture for Autonomous AI Agents},
  author  = {Moore, B. W.},
  year    = {2026},
  note    = {Preprint v1.2 --- Design \& Methodology Paper (post-review
     revision)},
  howpublished = {\url{https://apexorca.io/research}}
}
```

Listing 2: BibTeX citation stub (update with arXiv ID on acceptance).