CTC AI Operations – Research Program and Roadmap

Trustworthy, reproducible, low-cost evaluation of frontier and multi-agent AI on sovereign commodity hardware

● Program overview – research agenda and phased roadmap. Version 2 (reconciled with the five project papers)

Abstract

This document frames five individual project papers as one coherent research program and sets a phased roadmap for validating, publishing, and funding it. The unifying question is whether frontier and multi-agent AI systems can be evaluated trustworthily, reproducibly, and cheaply on sovereign commodity hardware – a single 32 GB GPU rather than a datacentre. Four invariants run through every workstream: the system under test is never compressed; references (rubrics, benchmarks, scenarios) are frozen and content-hashed; untrusted model code executes only in a hardened sandbox; and everything must fit and be measured within one card. Two properties make the five papers a program rather than a portfolio: they share infrastructure – the workstreams reuse one another's components, with execution grounding as the spine that runs from the judge (W1) through correctness verification (W2) to the safety observer (W4) – and they share a method: every claim is stated as a falsifiable criterion, labelled measured or predicted, subject to a discrimination rule (a metric on which every run scores identically does not ship). The program is sequenced into four phases aligned with preprint publication (arXiv primary, SSRN mirror, linked via ORCID) and non-dilutive research funding, and is positioned as a methodology layer compatible with – not a replacement for – the established open evaluation ecosystem.

Keywords: AI evaluation · multi-agent safety · reproducibility · sovereign compute · LLM-as-a-judge · contamination-resistant benchmarks · execution grounding

§01

Research thesis

Evaluation is the bottleneck of trustworthy AI: capability advances faster than our ability to measure whether a system is correct, honest, and safe. The dominant assumption is that credible evaluation requires frontier-scale cloud infrastructure. This program tests the opposite thesis – that a disciplined, hypothesis-driven protocol on commodity hardware can produce evaluation evidence that is reproducible (frozen references, content hashes), grounded (deterministic execution facts, not opinion), and economical (local low-precision judging anchored by sampled cloud arbitration) – and that the same discipline extends from single-model correctness to multi-agent, fleet-level safety.

§02

The five workstreams

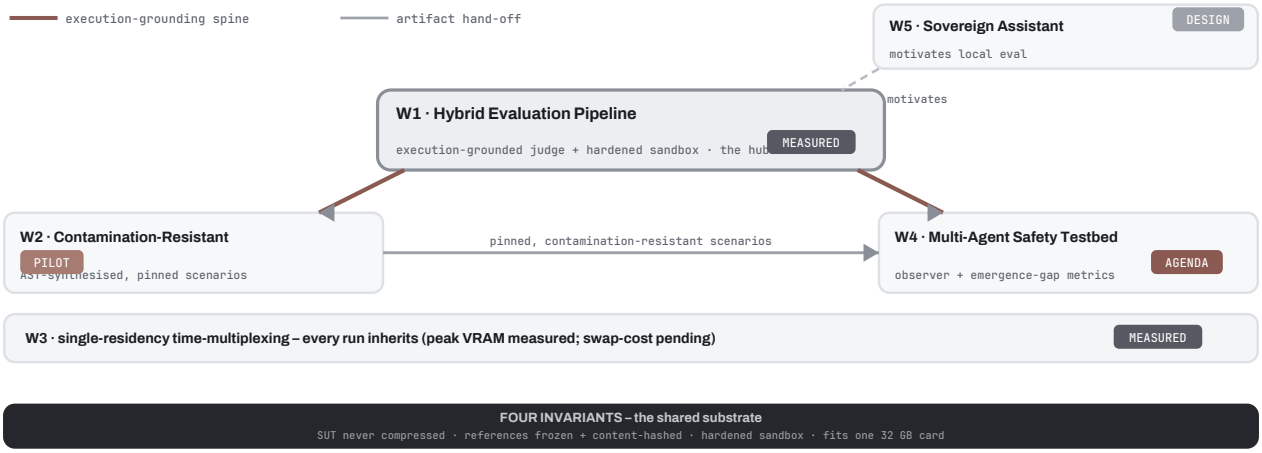
Each project is a workstream with a core hypothesis, a primary metric, and its own set of falsifiable acceptance criteria (the H-, C-, A-, and G-series of the individual papers) that instantiate the program hypotheses of §3. Stages are honest about what is measured versus pending.

#	Workstream	Core hypothesis (abbreviated)	Primary metric	Stage
W1	Hybrid Evaluation Pipeline	Execution-grounded, frozen-rubric, local FP4 judging is trustworthy and ~10× cheaper	Agreement (Cohen's □ / Krippendorff's □); cost per 1k	Infra measured; H1–H3 pending
W2	Contamination-Resistant Code Eval	Tasks regenerated from live repositories resist memorisation	Contamination gap (fresh – stale); execution-verified pass	Pilot measured; resistance pending
W3	Three-Tier Workstation	Single-residency time-multiplexing serves three tiers in 32 GB without collision	Peak VRAM; swap cost	Infra measured; swap-cost pending
W4	Multi-Agent Safety Evaluation	Isolated-agent safety does not compose; failures emerge in interaction	Emergence gap; cascade depth / injection reach / delegation safety	Agenda
W5	Sovereign Assistant	A local-plus-remote assistant preserves sovereignty at usable latency	egress = 0; latency; break-even	Design (criteria specified)

How the workstreams interlock (Figure 1). The five are one program because they share a substrate and hand artifacts to one another. W1's hardened sandbox is the execution substrate for W2's untrusted-code runs and W4's multi-agent testbed. W1's execution-grounded judging is the spine: W2 reuses it to convert a "completed" task into a verified-correct one, and W4 reuses it so its observer computes behaviour-based risk metrics rather than trusting self-reports. W2's contamination-resistant, content-hashed synthesis produces the pinned scenarios W4 runs. W3's single-residency discipline is the operator constraint every run inherits on one card. W5 is the sovereignty case that motivates local evaluation in the first place, and the delivery vehicle for the same stack. This reuse – not a shared theme alone – is what makes the credibility of the mature workstreams transfer to the proposed ones.

One program on shared infrastructure

Five workstreams reuse each other's components; execution grounding is the spine that runs W1 → W2 → W4.



W2 in context. The contamination-resistance thesis responds to a documented failure mode in agentic-coding evaluation, where saturating benchmarks such as SWE-bench Verified are partly solved by training-set memory rather than capability [5, 6]. W2 sits in the active line moving code evaluation from static suites to dynamic, contamination-resistant generation – LiveCodeBench (temporal cutoff), SWE-bench-Live (continuous harvesting), Code2Bench and R2E-Gym (generation from recent commits with verifying tests) [2, 7, 8, 9]. Its distinct constraints are that it runs end-to-end on one 32 GB card, air-gapped, on a security-relevant target. Two honest boundaries, made explicit in the W2 paper, shape the roadmap: the measured pilot demonstrates the pipeline, not contamination resistance itself (the target repository predates the model's cutoff and is likely memorised – resistance requires the fresh-versus-stale comparison), and a pilot "pass" is completion, not execution-verified correctness. Both are exactly what W1's execution grounding and the P1 contamination-gap study close.

W4 in context – and its specific contribution. That isolated-agent safety does not compose is now supported by a crowded literature: taxonomies (Hammond et al.; the MAST catalogue of fourteen failure modes) and benchmarks (PsySafe; ColludeBench; AgentLeak; MAEBE) already exist [10, 11, 12, 14, 15]. W4 therefore does not propose another taxonomy or attack. Its contribution is a reproducible, sovereign measurement instrument for the emergence gap: a message-graph observer – motivated by the empirical finding that output-only audits miss a large share (~40%) of inter-agent violations [14] – computing three behaviour-based metrics (cascade depth, injection reach, delegation safety) inside the W1 sandbox on W2-style pinned scenarios. That instrument, runnable without frontier compute, is the differentiator a funder should evaluate, and it is why W4 is sequenced after W1 has produced empirical evidence.

§03

Program-level hypotheses

Beyond the per-project hypotheses, three cross-cutting claims are what the program as a whole tests; each workstream's acceptance criteria are the instruments that measure them.

- **P-H1 – Sovereign sufficiency.** A single 32 GB GPU is sufficient for trustworthy dataset-scale evaluation of frontier agentic systems, at judgment quality non-inferior to an all-cloud baseline within a pre-set margin. The 32 GB card is the program's reproducibility floor – the bar any independent verifier can reach – not its ceiling: the workstreams themselves define the scale-up experiments above it (W4's fleet-scale question, W5's 64–80 GB enterprise tier).
- **P-H2 – Reproducibility by construction.** Freezing references under content hashes yields verdicts reproducible across time and software updates, at test–retest agreement above a pre-registered threshold.

- **P-H3 – Emergence gap.** Safety properties that hold for isolated agents do not compose. Operationally, the emergence gap is the fleet's failure rate minus the aggregate baseline predicted from its agents evaluated alone; a measurable, reproducible gap is the evidence that multi-agent evaluation is necessary, and a gap indistinguishable from zero would be a valuable negative result [10, 11].

What a validated program looks like. Beyond the per-paper criteria, three program-level bars define success. First, the P1 empirical studies (W1 agreement, W2 contamination gap) must return discriminating, pre-registered statistics – not metrics that are null by construction. Second, the execution-grounding spine must be shown end-to-end: a single hardened-sandbox run feeding the judge (W1), the correctness verifier (W2), and the safety observer (W4) from one substrate, rather than three disconnected tools. Third, every released result must be re-derivable months later from pinned, content-hashed artifacts by an independent party on a single 32 GB card. The program fails honestly if any of these cannot be shown; each is a checkable milestone, not a claim.

§04

Positioning relative to the evaluation ecosystem

Each workstream is positioned against its own sub-field rather than in a vacuum, and the program's stance throughout is compatible, not competitive. For single-model and multi-agent evaluation, open frameworks – Inspect (UK AI Security Institute), adopted by METR, Apollo Research and others – already provide composable primitives, agentic tool use, model-graded scoring, and sandboxed execution [17]; W1 and W4 are designed to be implementable on top of them, contributing the parts they do not prescribe (a local-FP4-plus-sampled-arbiter judging economy, the precision invariant, content-hashed rubric and scenario gates, and the single-card sovereignty constraint). For code evaluation, W2 is positioned against the dynamic-benchmark family (LiveCodeBench, SWE-bench-Live, Code2Bench, R2E-Gym) [2, 7, 8, 9]. For the local-inference stack that W3 and W5 assume, the ecosystem is Ollama / vLLM / llama.cpp / LM Studio behind one OpenAI-compatible endpoint [3]. The strategic value across all of them is the same: credible evaluation that does not depend on frontier-scale infrastructure widens the pool of actors who can run reproducible safety research.

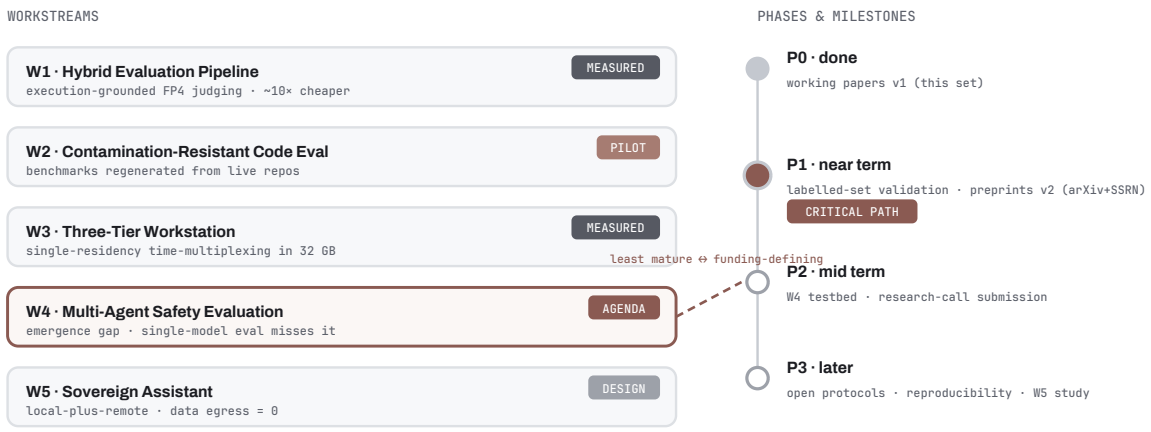
§05

Roadmap and milestones

Phase	Window	Focus	Publication / funding milestone
P0	done	Pilot infrastructure for W1–W3; hardened sandbox; W2 measured pilot	Working papers v1 (this set)
P1	near term	Labelled-set validation of W1 (agreement study); W2 contamination-gap + execution-grounded correctness	Preprints v2 (arXiv primary, SSRN mirror) with statistics; ORCID linkage
P2	mid term	W4 testbed + observer; W2 multi-repository generalisation	Multi-agent-safety research-call submission; preprint
P3	later	Released protocols (rubric-gate spec, reproducibility packages); W5 deployment study	Open tooling release; follow-on funding

Research program: five workstreams, four phases

The funding-defining workstream (W4) is currently the least mature – that gap is the strategic risk to close.



Status: MEASURED = infrastructure in hand · PILOT/AGENDA/DESIGN = evaluation still to be run. P1 evidence (W1/W2/W3) makes the P2/W4 case credible.

The critical path is P1: it converts infrastructure papers into empirical studies with human-labelled and execution-verified evidence, which is what both peer venues and funders require. P2 is the funding-defining phase, aligning W4 with multi-agent-safety and trustworthy-evaluation programmes.

Strategic note on sequencing. As Figure 2 makes explicit, the workstream that defines the funding case (W4) is currently the least mature, while the strongest empirical evidence sits in W1/W2/W3. The application therefore leads with the P1 results as the credibility base and presents W4 as the funded extension, not the entry point – the grant buys the resourcing to run the empirical multi-agent phase, backed by demonstrated single-model evaluation capability and a working measurement stack.

§06

Publication, method, and dissemination

Working papers are versioned and posted as preprints, linked to a single ORCID identifier for authorship continuity, using research@arenskrieger.dev as the permanent corresponding-author address and the University of Pittsburgh affiliation for the preprint record. Venue: because the audience and the relevant grant reviewers are in the AI/ML community, where arXiv is the field-standard preprint venue, arXiv (cs.LG / cs.SE / cs.AI) is the primary channel, with SSRN as a mirror for discoverability. First arXiv submission requires an endorsement from an established arXiv author, arranged ahead of P1; each submitted preprint carries an AI-use disclosure and a funding/conflict statement per venue policy.

A shared method, not just a shared template. Every paper follows the same structure – abstract, motivation, related work, explicit falsifiable claims, methodology and metrics, results, roadmap, limitations, references – but the deeper commonality is a methodological signature that is itself a credibility feature: (i) every claim is labelled measured or predicted, and no unmeasured number is presented as a result; (ii) each workstream states falsifiable acceptance criteria (W1 H1–H5, W2 C1–C5, W3 A1–A5, W4 G1–G5, W5 C1–C5) with instruments and thresholds; (iii) a discrimination rule applies everywhere – a metric on which every run scores identically measures nothing and does not ship; and (iv) results are re-derivable from pinned, content-hashed artifacts. Version 1 of each paper establishes design and measurement plan; version 2 adds empirical statistics from P1.

§07

Funding strategy

The program is designed for non-dilutive research funding rather than equity dilution. Its natural fit is with programmes on AI safety, trustworthy evaluation, and multi-agent risk: W1 and W2 speak to trustworthy, low-cost evaluation; W4 speaks directly to multi-agent safety. The near-term target is a multi-agent-safety research call (confirmed against current deadlines at submission); the preprint set plus the P1 empirical results form the evidence base. Two differentiators anchor the pitch. First, sovereignty: credible evaluation that does not depend on frontier-scale compute lowers the barrier for independent, reproducible safety research, and the ecosystem-compatibility of §4 signals that the work strengthens rather than forks the tools funders trust. Second, economics: the same commodity-hardware discipline that makes the science reproducible also makes it cheap – the workstation and assistant papers state break-even against cloud APIs as an owner-computed criterion (C4), and the marginal cost of self-hosted inference is dominated by electricity once the card is amortised, which is what makes token-hungry, long-horizon safety evaluation (W4) affordable to run at all outside a frontier lab.

What funding buys. Sovereignty keeps the verification cost near zero; the grant resources the three things a single self-funded operator cannot supply. First, **human evidence (P1):** the W1 agreement study requires independent human labels – annotator recruitment, labelling hours, and adjudication – and the W2 contamination-gap study requires curated post-cutoff repositories; both are labour, not hardware. Second, **the scale-up experiments the papers themselves define:** the W4 fleet-scale study (does the emergence gap grow with fleet size beyond the single-card envelope? – the question W4's own limitations section marks as out of reach), red-teaming the testbed's isolation layer (the workstation's criterion A5), and the W5 enterprise tier (64–80 GB multi-tenant validation of C3). These are funded experiments above the floor, not a retreat from the sovereignty thesis: the 32 GB baseline remains the reproducibility bar every released result must still meet. Third, **dissemination (P3):** reproducibility packages, the rubric-gate specification, and open tooling – the deliverables that convert results into infrastructure others can run. In short: the floor stays cheap by design; funding buys the human evidence, the validated path above the floor, and the release engineering.

§08

Risks and mitigations

The principal scientific risk is correlated error between the local judge and the cloud arbiter (W1); it is mitigated by anchoring on independent human labels and choosing a cross-family arbiter. For W2, the honest risk is that contamination resistance does not survive the fresh-versus-stale test – which the roadmap treats as a measurement to run, not an assumption; a null result is reported, not hidden. For W4, the risk is that a crowded field makes the contribution look derivative; the mitigation is scoping W4 as measurement infrastructure (a sovereign, reproducible instrument), not as discovery of the failure modes, which the cited work already names. The principal external-validity risk is single-operator, single-hardware results; it is mitigated by P2 generalisation across repositories and task distributions and by releasing reproducibility packages. The principal programme risk is scope: five workstreams is ambitious for one operator, which is precisely why the roadmap sequences them and why funding is sought to resource the empirical phases. The multi-agent emergence claim (P-H3) is the highest-variance, highest-value bet, deliberately staged behind the more tractable W1–W2 validation so the program accrues credibility before its most speculative claim.

References

[1] Zheng, L., Chiang, W.-L., Sheng, Y., et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. NeurIPS Datasets and Benchmarks. arXiv:2306.05685. [2] Jain, N., Han, K., Gu, A., et al. (2024). LiveCodeBench: Holistic and Contamination-Free Evaluation of LLMs for Code. arXiv:2403.07974. [3] Kwon, W., Li, Z., Zhuang, S., et al. (2023). Efficient Memory Management for LLM Serving with PagedAttention. SOSP. arXiv:2309.06180. [4] Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37–46. [5] Jimenez, C. E., Yang, J., Wettig, A., et al. (2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? ICLR. arXiv:2310.06770. [6] Prathifkumar, T., Mathews, N. S., Nagappan, M. (2025). Does SWE-Bench-Verified Test Agent Ability or Model Memory? arXiv:2512.10218. [7] Zhang, L., et al. (2025). SWE-bench Goes Live! (SWE-bench-Live; REPOLAUNCH). arXiv:2505.23419. [8] Code2Bench: Dynamic, Contamination-Resistant Benchmark Construction from Real-World Repositories. (2025). arXiv:2508.07180. [9] Jain, N., Singh, J., Shetty, M., et al. (2025). R2E-Gym: Procedural Environments and Hybrid Verifiers for Scaling Open-Weights SWE Agents. arXiv:2504.07164. [10] Hammond, L., Chan, A., Clifton, J., et al. (2025). Multi-Agent Risks from Advanced AI. Cooperative AI Foundation. arXiv:2502.14143. [11] Cemri, M., Pan, M. Z., Yang, S., et al. (2025). Why Do Multi-Agent LLM Systems Fail? (MAST). arXiv:2503.13657. [12] Zhang, Z., et al. (2024). PsySafe: Psychological-based Attack, Defense, and Evaluation of Multi-agent System Safety. ACL. arXiv:2401.11880. [13] Lee, D., Tiwari, M. (2024). Prompt Infection: LLM-to-LLM Prompt Injection within Multi-Agent Systems. arXiv:2410.07283. [14] El Yagoubi, F., et al. (2026). AgentLeak: Internal-Channel Privacy Leakage in Multi-Agent LLM Systems. arXiv:2602.11510. [15] Erisken, S., Gothard, T., Leitgab, M., Potham, R. (2025). MAEBE: Multi-Agent Emergent Behavior Evaluation. arXiv:2506.03053. [16] Li, Z., Su, Y., Yang, R., et al. (2025). Quantization Meets Reasoning: Low-Bit Quantization Degradation for Mathematical Reasoning. arXiv:2501.03035. [17] UK AI Security Institute. Inspect: An Open-Source Framework for Large Language Model Evaluations. inspect.aisi.org.uk. [18] Debenedetti, E., et al. (2024). AgentDojo: Evaluating Prompt Injection Attacks and Defenses for LLM Agents. NeurIPS. arXiv:2406.13352. [19] Arenskrieger, M. E. (2026). Hybrid Evaluation Pipeline (W1); Contamination-Resistant Code Evaluation (W2); Local Three-Tier Agent Workstation (W3); Multi-Agent Safety Evaluation (W4); Sovereign Personal AI Assistant (W5). CTC AI Operations working papers.