

Lived-Experience Cognitive Engine (LECE)

A local-first architecture for continual self-improvement of personal autonomous agents

Author: Utso Sarkar

Affiliation: Founder OS / Stamped Energy

Version: 1.0 — June 2026

Repository: github.com/officiallyutso/Founder-OS

Contact: utsosarkar1@gmail.com

ABSTRACT. State-of-the-art self-evolving agents — EvolveR, EvoSC, Continual Experience Internalization, ProPlay, and Global Workspace Agents — are evaluated on generic, public benchmarks. None address the setting that matters most to a working founder: a **SINGLE HUMAN'S LONGITUDINAL, PRIVATE, ON-DEVICE OPERATIONAL LIFE**. We present **LECE** (Lived-Experience Cognitive Engine), a three-pillar architecture integrated into Founder OS that closes the learning loop on personal agent traces without exporting data to cloud trainers. Pillar 1 performs **EXPERIENTIAL SELF-DISTILLATION**: off-policy, principle-level internalization from flight-recorder JSONL with effectiveness decay and step-wise injection. Pillar 2 maintains a **BUSINESS DIGITAL TWIN** and runs **PREPLAY** — counterfactual rehearsal of high-stakes actions inside a registry-enforced simulation sandbox before real execution. Pillar 3 implements a **GLOBAL WORKSPACE** attention economy with **TRUST-CALIBRATED AUTONOMY** learned per action type against a hard safety floor. All heavy or continuous cognition is feature-flagged; safe Tier-A distillation is default-on. LECE is shipping code, not a roadmap.

1. Introduction

Modern language-model agents exhibit remarkable single-turn competence and near-zero longitudinal memory. Each session resets. For a founder managing thousands of outreach threads, investor conversations, product pivots, and irreversible commitments, this amnesia is not a UX inconvenience — it is an architectural failure.

Founder OS addresses the **body** of an artificial cofounder: 117 tools, multi-agent swarm orchestration, SQLite CRM, ChromaDB retrieval, self-healing execution, and human approval gates for irreversible actions. Before LECE, the system's richest asset — append-only per-turn traces in `data/traces/*.jsonl` — was written and never read by any learning process. LECE is the **psyche**: the subsystem that metabolizes lived experience into durable judgment.

Contributions. This whitepaper specifies LECE and claims four contributions (Section 6): (1) rigorous problem formulation for personal on-device continual improvement; (2) integrated closure of trace → principle → behavior and twin → preplay → outcome loops; (3) tool-registry simulation sandbox for safe rehearsal; (4) earned, per-capability trust with inviolable safety floors.

2. Related Work

Line of work	Representative	Relevance to LECE
Experience-driven agent lifecycle	EvolveR (2510.16079)	Trajectory recording and reuse; LECE targets private founder life, not public benchmarks
Self-consolidation	EvoSC (2602.01966)	Memory consolidation analogue; LECE nightly distillation job
Continual experience internalization	CEI (2606.04703)	Off-policy, principle-level, step-wise injection recipe
Procedural world models	ProPlay (2606.12780)	Procedure graph + preplay; LECE applies to live CRM/runway state
Global Workspace agents	GWT agents (2604.08206)	Saliency-based coalition bidding
RAG / vector memory	ChromaDB, etc.	Necessary recall layer; insufficient for judgment and rehearsal

LECE does not claim invention of these ideas in isolation. It claims **first integrated instantiation** on a founder's private operational substrate.

3. Problem Formulation

Input. A single principal (founder) operating a company over months. An autonomous agent with tool access to email drafts, CRM, calendar, web research, and messaging interfaces.

Data. Sensitive longitudinal records: traces, action logs, strategies, failures, CRM state, financial runway, approval history.

Objective. Improve agent behavior week-over-week on *this* founder's tasks — outreach tone, follow-up timing, high-stakes messaging, proactive surfacing — without (a) uploading private data for third-party training, or (b) collapsing via on-policy learning from current mistakes.

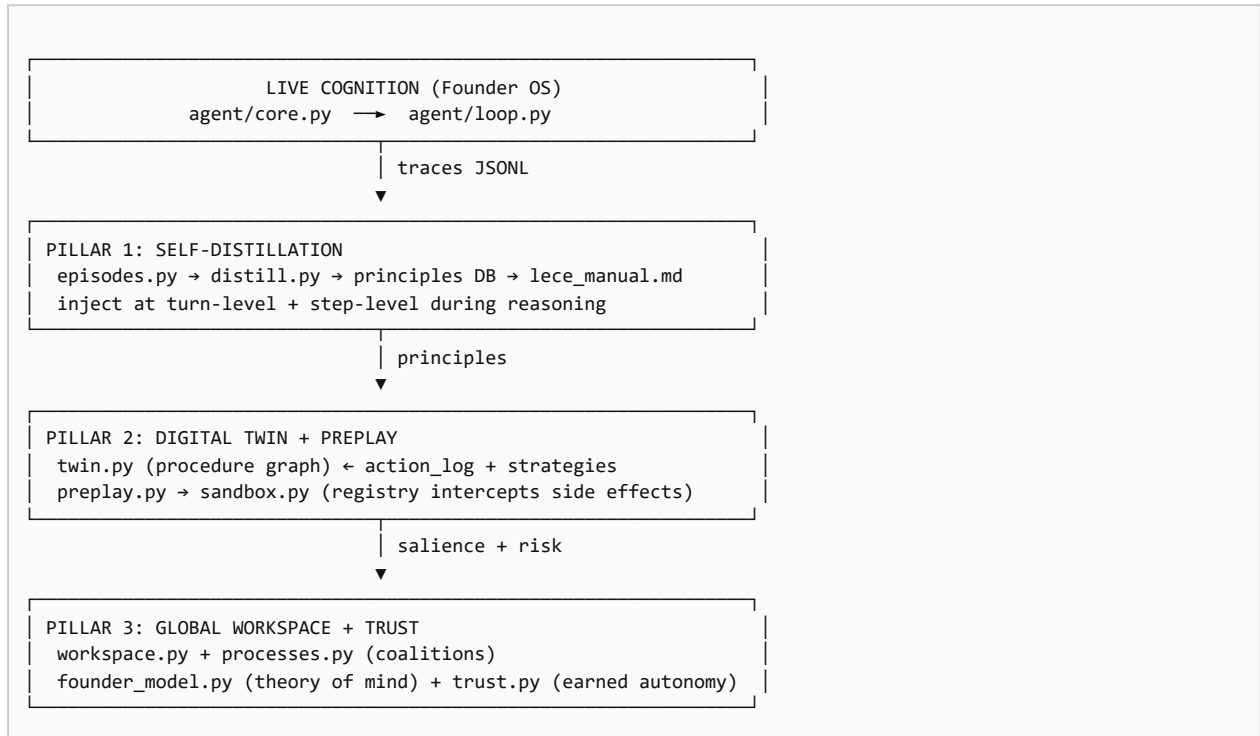
Constraints.

- Local-first storage and computation (optional cloud LLM API, replaceable with Ollama).
- Irreversible actions require human approval regardless of learned trust.

- Learning must be measurable (preplay error, principle effectiveness, trust calibration).

4. System Architecture

LECE lives in `agent/cognition/` (15 modules + `train/` scaffold) atop Founder OS's `agent/core.py` and `agent/loop.py`.



Closed flywheel. Act → record trace → distill principles → inject into reasoning → preplay high-stakes branches → execute with approval → measure outcome → update twin reliabilities → feed next distillation cycle.

5. Pillar 1 — Experiential Self-Distillation

5.1 Design choices

Dimension	LECE choice	Rationale
Granularity	Principle-level	Instances ossify; principles transfer (CEI)
Regime	Off-policy	High-scoring episodes only; avoids mistake reinforcement
Injection	Step-wise	Aligns with intermediate tool-selection decisions
Forgetting	Effectiveness decay + quarantine	Prevents unbounded manual growth

5.2 Pipeline

1. `iter_traces(since_ts)` — consume JSONL flight recorder (`agent/trace.py`).

2. `episodes.py` — segment traces; score outcomes $\in [0,1]$ from verification, approvals, healing events, strategy success.
3. **Filter** — retain episodes with score ≥ 0.6 .
4. `distill.py` — cluster by tool family; LLM extracts transferable principles (not anecdotes).
5. **Semantic dedup** — Chroma `DefaultEmbeddingFunction` (all-MiniLM-L6-v2); reinforce near-duplicates.
6. **Decay** — unreinforced principles lose effectiveness; quarantine below 0.2.
7. `manual.py` — rebuild `data/agent_state/lece_manual.md` (Tier A internalization).
8. **Injection** — `evolution.retrieve_context()` per turn; `loop.py` per reasoning step.

5.3 Tier B (optional, default OFF)

Scaffold for local LoRA fine-tuning: `train/dataset.py`, `train/lora.py`, `train/gate.py` (promote only if eval beats baseline), `train/modelfile.py`. Enable with `LECE_DISTILL_TRAIN=true` and GPU dependencies.

5.4 Schedule

`job_distill` cron at **03:30** daily (after memory consolidation at 03:00).

6. Claimed Contributions

1. **Problem formulation.** Personal, private, longitudinal, on-device continual improvement from one agent's operational life — not anonymous task benchmarks.
2. **Closed loop on private life.** Traces become principles become behavior; twin state becomes simulated futures become safer actions become measured outcomes become sharper simulations — wired in production code.
3. **Registry-enforced simulation sandbox.** `contextvars.SIMULATION` propagates through execution; `agent/registry.py` intercepts side-effect tools so rehearsal cannot leak to production.
4. **Trust as earned, per-capability quantity.** Competence tracked per action type from outcomes and failure ledger; autonomy scales upward only; static approval floor never lowered.

7. Pillar 2 — Business Digital Twin + Preplay

7.1 Twin state

Extends `memory/world_model.py`:

- CRM pipeline, tasks, goals, finance, strategies.
- **Procedure graph** (`twin_transitions`): (state_key, action_key) → outcome_key with reliability, bootstrapped from `action_log` + `strategies`.

7.2 Sandbox

`SimulationContext` + `contextvars.SIMULATION`. Write tools return **predicted** results. Enforced in registry `call()` and swarm sub-agents. Structural guarantee: simulation cannot send email, post publicly, or mutate production state.

7.3 Preplay algorithm

Before high-stakes tools (`send_email`, `x_post`, persistent writes):

1. Generate N candidate variants (`LECE_PREPLAY_BRANCHES`, default 3).
2. Estimate outcomes via `learned` | `swarm` | `hybrid` (`LECE_PREPLAY_MODE`).
3. Select best branch (critic score / transition reliability).
4. Route real action through approval gate with attached reasoning.
5. **Expectation feedback** — compare prediction vs actual; update reliabilities; log `preplay_runs`.

8. Pillar 3 — Global Workspace + Adaptive Autonomy

8.1 Workspace

Cross-turn blackboard (`workspace` table): items with salience and decay. Coalitions bid:

- **perception** — pending approvals, reminders
- **goals** — active objective pressure
- **healing** — unrecovered failures
- **twin_risk** — runway / approval backlog elevation

Winner broadcasts into deliberative core. Optional continuous loop (`LECE_WORKSPACE_CONTINUOUS`, default OFF) ticks every `LECE_WORKSPACE_TICK_S` seconds.

8.2 Founder theory of mind

`founder_model` KV: risk tolerance, communication style, decision speed — learned from interaction patterns, not static onboarding forms.

8.3 Trust-calibrated autonomy

`trust.py` extends `policy.decide()`:

- Never weaker than static `AUTONOMY_LEVEL` floor.

- Per-tool competence from outcomes + failure ledger.
- Elevation requires `LECE_TRUST_MIN_TRIALS` (default 5).
- Irreversible actions remain on sacred approval list.

9. Measurement Spine

Metric	Module	Description
Preplay MAE / accuracy	<code>metrics.py</code>	Predicted vs actual outcomes
Principle effectiveness	<code>metrics.py</code>	Average effectiveness; quarantine count
Trust competence	<code>metrics.py</code>	Per action-type scores
Ablations	<code>evals/lece/</code>	Principle vs instance; preplay on/off
Synthetic traces	<code>scripts/gen_synthetic_life.py</code>	Reproducible evaluation without private data

Dashboard: `/api/cognition` + React `/cognition` panel. Longitudinal tables in `evals/lece/__init__.py` are designed for week-over-week founder-specific improvement claims.

10. Implementation Summary

Package: `agent/cognition/` — 15 modules + `train/` scaffold.

New tools (10): `distill_now`, `list_principles`, `preplay_action`, `simulate_strategy`, `twin_state`, `why_did_you`, `workspace_state`, `trust_report`, `founder_profile`, `lece_metrics`.

Database tables (7): `principles`, `twin_transitions`, `preplay_runs`, `workspace`, `founder_model`, `trust_scores`, `distill_runs`.

Configuration (selected)

Variable	Default	Purpose
<code>LECE_ENABLED</code>	<code>true</code>	Master switch
<code>LECE_DISTILL_ENABLED</code>	<code>true</code>	Nightly distillation
<code>LECE_DISTILL_TRAIN</code>	<code>false</code>	Tier B LoRA
<code>LECE_PREPLAY_ENABLED</code>	<code>true</code>	Preplay before high-stakes tools
<code>LECE_PREPLAY_MODE</code>	<code>hybrid</code>	learned / swarm / hybrid
<code>LECE_WORKSPACE_CONTINUOUS</code>	<code>false</code>	Continuous cognitive loop
<code>LECE_ADAPTIVE_AUTONOMY</code>	<code>true</code>	Learned trust

Full table in repository [docs/LECE.md](#).

11. Limitations

- Tier B weight training requires GPU and manual setup; scaffold validates dependencies only.
- Swarm preplay rollouts are LLM-estimated, not full environment simulation.
- Founder theory-of-mind uses heuristics until sufficient longitudinal data accumulates.
- Continuous workspace loop increases LLM cost; disabled by default.
- Causal claims require weeks of measured flywheel data per deployment; ablation harness provided, population study not yet published.

12. Conclusion

LECE turns Founder OS from a capable amnesiac into a system that **accumulates operational wisdom** — distilling the past, rehearsing the future, and earning autonomy one capability at a time. The contribution is architectural synthesis under a novel problem formulation: **your private life as the training set, your machine as the only student.**

The implementation is open. The measurement spine is built for falsification. We invite replication on synthetic traces and critique on the claimed novelty boundary.

References

1. EvolveR — experience-driven agent lifecycle. arXiv:2510.16079.
2. EvoSC — self-consolidation for agents. arXiv:2602.01966.
3. Continual Experience Internalization. arXiv:2606.04703.
4. ProPlay — procedural world models for planning. arXiv:2606.12780.
5. Global Workspace Agents. arXiv:2604.08206.
6. Baars, B. — Global Workspace Theory (cognitive architecture foundation).
7. Minsky, M. — Society of Mind (multi-agent deliberation metaphor).
8. Founder OS repository — `docs/LECE.md`, `agent/cognition/`, `evals/lece/`.

Citation. Sarkar, U. (2026). *Lived-Experience Cognitive Engine (LECE): A local-first architecture for continual self-improvement of personal autonomous agents*. Founder OS Technical Whitepaper v1.0.
<https://utso.stamped.work/papers/lece-whitepaper.pdf>