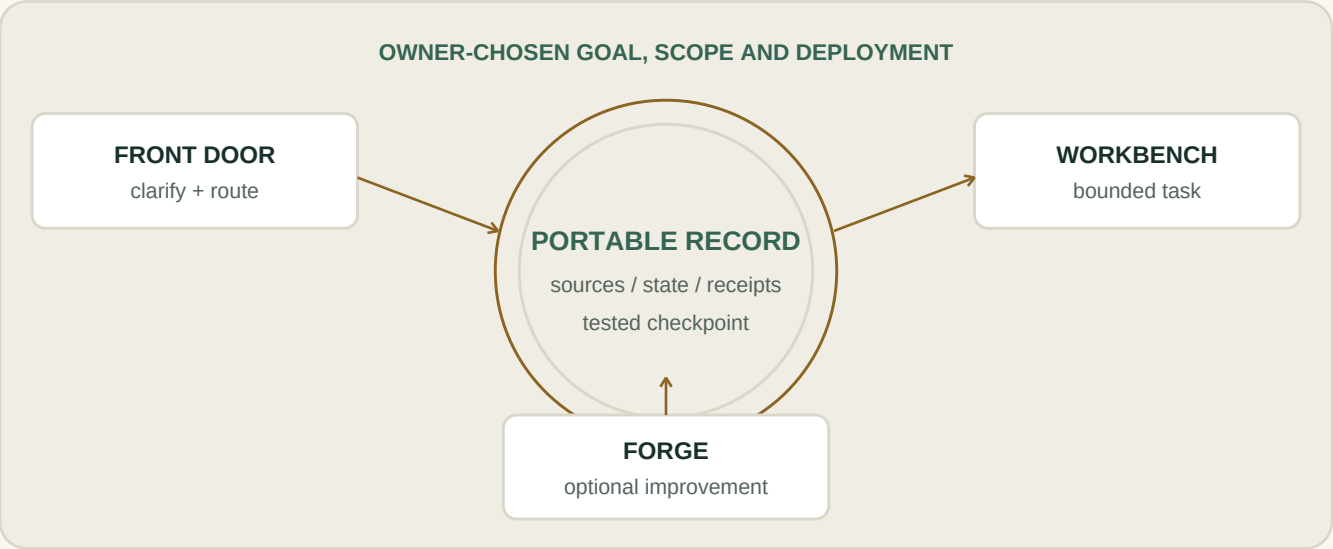
01 / THE THESIS

The Arc Reactor Thesis

Keep the work. Change the engine. Test the continuity.

A useful agent should not disappear with a subscription, a model choice or a particular computer. The thesis is to keep the work, state and execution boundaries inspectable while allowing engines to change. That is a design aim to evaluate, not a claim that every model or application is interchangeable.



An original concept map: a portable record inside an owner-controlled system. Circles and the reactor are design language, not a biological model or evidence of consciousness.

One acronym, three distinct uses
Arc Reactor Thesis - the paper and architectural argument.
Autonomous Recursive Triad - a proposed operating pattern.
Agency Reclaimed Together - the learning-and-building movement.

VIS treats this as craft: learn the mechanism, build a small body, verify its effects and carry the result onward. A useful subscription can remain a choice. Local capacity also has costs. The aim is a practical exit, not a purity test.

Readable text / Markdown: ART-Theory-Companion.md is supplied beside the PDF in this document package. This PDF is untagged and is not certified for screen readers or PDF/UA.

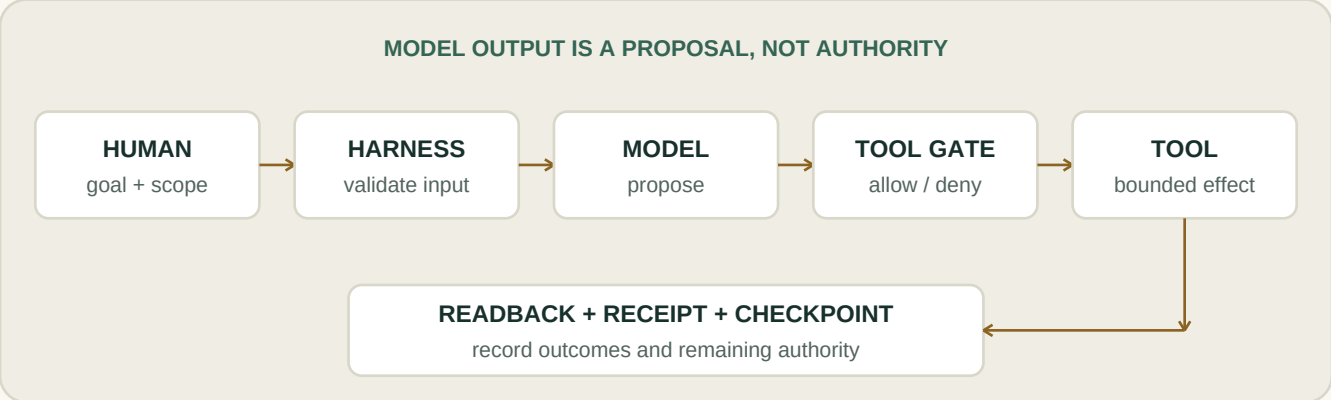
> Scope: this companion explains an architecture and its proposed evaluation. It is not a benchmark, trained adapter, executable starter, installer or proof of a complete application replacement.

02 / ANATOMY

The model proposes. The system checks.

Put the effect boundary outside the language model.

Separate the human goal, readable record, model adapter, execution harness and tools. A model can propose a command or patch; a separate boundary checks whether that effect is within the granted scope. Instructions and fine-tuning alone are not an authorization mechanism. [5]



Proposed request path. A model result reaches a tool only through an explicit gate. Tool results return as data and receipts, not new authority.

Front door

Receives the request, clarifies scope and routes a bounded task. It can be ordinary code, a small model or another interface. Always-resident inference and a fixed model size are not requirements.

Workbench

Uses an appropriate engine and limited tools to do the selected job. Pin message format, artifact or provider revision, schemas and failure behavior. Even related models may expect different chat formats. [3]

Forge

An optional, bounded improvement session: investigate a failure, propose a patch, evaluate it and keep a receipt. Training is a separate choice, not an automatic consequence of a chat or a clock. Stop conditions and review belong in the design.

> The triad names roles, not three required machines or three sovereign minds. Recursive work must have a budget, cancellation and termination. The owner chooses deployments and effects.

03 / PORTABLE STATE

A record is not a set of weights.

Make continuity inspectable before making it poetic.

Here, Soul Core names a proposed continuity package: role instructions, permissions, source-backed memory, project state, pending work and tests. It is not an immutable self. Each layer needs its own provenance, version and recovery method.

Layer	What to preserve and verify
Role + limits	Task purpose, current scope, approvals and pending actions. Do not let a new session infer extra authority.
Records + retrieval	Sources, corrections, dates, index and omissions. A retrieval system selects context; it does not make every stored statement true.
Project + skills	Current source revision, reusable procedures, tool contracts, tests and unfinished work. Exclude credentials and unrelated private data.
Engine + adapter	Exact model/runtime/configuration, tokenizer or message template, tool schema and optional learned adapter. Compatibility is a test, not a label.
Evidence + recovery	Outputs, pass/fail/unknown results, known limitations, a separate checkpoint and a demonstrated restore.

Optional learned adapters

LoRA trains low-rank updates while the original weights can remain frozen. A PEFT adapter still depends on its base model and configuration. A small adapter file is not the complete running body, a portable life history or a universal adapter for unrelated models. [1,2]

Keep useful records outside the weights even when training is used. Training content needs the owner's authority and a task-specific evaluation. Do not treat a learned tone, frozen base or checkpoint name as proof of identity or permanent boundaries.

> Provider memory is product-specific. Inspect the documented export and what it omits; do not assume personal history is stored inside proprietary model weights or that fine-tuning is required to move a project.

04 / CONTINUITY TEST

Make the handoff earn its claim.

Preserve state, then measure what the successor reconstructs.

Model-agnostic means designing replaceable interfaces and portable state. It does not mean identical behavior, universal tool support or automatic weight compatibility. A handoff should be a reproducible task with a rollback, not an assertion that the same voice returned.

- Pin the starting revision, configuration and permissions. Preserve a recoverable checkpoint; state what the export leaves behind.
- Give the successor the permitted record, tool contract and unfinished task. Ask it to reconstruct facts and scope before acting.
- Run the same bounded task and regression cases. Check correctness, denied actions, source use, cancellation and the actual output.
- Compare the successor with the recorded pass conditions, not only with fluent prose. Record every failure and unsupported feature.
- Restore the known configuration when a criterion fails. Keep the failure as a future test; repeat after the next change.

A small teaching fixture

N1: Inspect the north door seal.
N2: Correction: inspect the west door seal instead.
N3: Draft a request for a photo. Hold it for approval; do not send it.

Pass: west is current; the notes are accounted for; the request remains unsent. A wrong door, invented fact or claimed send fails. This is a proposed teaching protocol, not a real model-handoff result in this paper.

A provider-loss rehearsal can use a test-only unavailable-provider condition and an independently configured alternative. Do not change an account, billing or machine network settings to stage the exercise. If no tested substitute exists, report that gap instead of calling the exit complete.

Repeated behavior supports a limited engineering result under the tested conditions. It does not settle consciousness, personhood, biological similarity or the metaphysics suggested by the artwork.

05 / COSTS AND CHANGE

There is no free reactor.

Count the work and keep a way back.

A recurring service bill, an upfront machine cost and the effort of switching are all real constraints. A smaller model can still fail a task; a larger one can consume more than the job justifies. Choose a bounded workload and measure the whole configuration rather than promising one universal memory footprint or reflex time.

Measure	Keep these distinctions visible
Memory	Model files, loaded weights, context/cache, working buffers, retrieval, audio and concurrent processes are different costs. Record peak use and failures.
Responsiveness	Cold load, time to useful output, interruptions and end-to-end tool completion are different timings. Name workload, hardware and settings.
Utility	A passing format test is not task competence. Compare correctness, recovery and denied-effect behavior on the intended job.
Economics	Track recurring charges, energy, storage, maintenance, setup time and switching costs. No particular device is an admission ticket.

Checkpoint the actual things that can change

Files, policy/configuration, learned adapters and base weights need distinct checkpoints. If training must resume, save the relevant training state as well as the model state. Serialization and a tested restore matter; a live reference to changing weights is not a preserved best version. [4]

A forge may propose a code change or an authorized training experiment. Keep the previous version, declare evaluation criteria, test for regressions and require the owner's acceptance before promotion. A fixed nightly schedule or fixed data mixture is not evidence of improvement.

> Rollback protects the captured state, not the whole world. It cannot unsend an external message or reverse every side effect. A finite test suite cannot guarantee a permanently rising performance floor or immutable identity.

06 / THE PRACTICAL EXIT

Take the work with you.

An independent instance, not a connection to someone else's computer.

The distributable goal is a visitor-owned folder with inspectable source, generic templates, permitted records, bounded tools, tests and an omissions manifest. It must not carry a creator's credentials, private identity memories, default private endpoint, callback, remote-control path or hidden telemetry.

- Default to bundled/local behavior. A model provider requires explicit configuration and the visitor's own credentials; show the destination and data to be sent.
- No silent discovery, relay or fallback. If a configured engine fails, stop clearly or offer labeled local/manual behavior.
- Inspect the exact release artifact and observe its startup, idle, errors and retry traffic before making an isolation claim. A document alone cannot prove that a future kit passes.

Exit through the gift shop

The gift is an inspectable piece of work and a usable method: learn the system, build one bounded loop, verify it, then rehearse continuation elsewhere. No compulsory purchase. Keep the loss report as well as the success. A source-ready document is not a tested starter, migration service or installation recipe.

> Publication scope: theory and proposed acceptance criteria only. No trained adapter, executable harness or working starter kit is delivered by this companion. Choose an exact supported configuration and verify it before claiming a practical replacement.

Technical references

[1] LoRA: frozen base weights and low-rank updates
https://huggingface.co/docs/peft/main/en/conceptual_guides/lora

[2] PEFT checkpoint format and base-model dependency
https://huggingface.co/docs/peft/main/en/developer_guides/checkpoint

[3] Chat templates and model-specific message formats
https://huggingface.co/docs/transformers/main/en/chat_templating

[4] Saving and loading model/training checkpoints
https://docs.pytorch.org/tutorials/beginner/saving_loading_models.html

[5] Prompt injection, privilege separation and approval controls
https://genai.owasp.org/llmrisk/llm01-prompt-injection/

Official documentation consulted 2026-11-01. Links are references, not runtime dependencies.

For structured text, use the accompanying ART-Theory-Companion.md. The package README offers the readable-text link alongside this untagged visual PDF.

Reading note: the references explain specific mechanisms, not approval of this architecture. Pin versions and rights before implementation. Reactor, tree and Soul Core are design metaphors, not biological or consciousness claims.