

Semantic Autopoiesis: An Engineering Roadmap for Machine Life

Keren Dong
Kungfu Origin Technology Limited
keren.dong@kungfu.link

Alpha 0.1.0-alpha.4 — July 2026

Abstract

Machine life need not begin when a model becomes generally intelligent or a robot becomes human-shaped. This paper examines a different threshold: a bounded machine organization that preserves a continuing subject, metabolizes technical resources, acts in its environment, and governs the production of its own successor organization. Its body can initially be inelegant and server-scale, assembled from operating systems, compilers, build tools, caches, gates, source code, executable extensions, networks, and replaceable reasoning agents.

We call the organizing principle *semantic autopoiesis*: the governed production of successor executable organization while identity, direction, admitted facts, authority, and admission rules remain continuous. The proposal has two jointly necessary mechanisms. A semantic kernel preserves the subject across process, model, organ, and machine replacement. Recursive dogfood lets that subject use its current engineering organization to develop, qualify, admit, and exercise changes to the mechanisms of its own future work. KFD and Kungfu provide a concrete implementation path: replaceable agents supply cognition, KFX extensions act as organs, and Buildchain-like qualification turns candidate self-modifications into bounded successor states.

A dated Kungfu snapshot contains substantial parts of this loop, including durable work identity, qualified executable bodies, verified history selection, KFX qualification, recursive dogfood, a daily Agent Patrol that captures bounded failures as Dogfood Findings, and a governed Deprecation Lifecycle for retiring obsolete surfaces. Together they constitute a partial, bounded homeostatic architecture with genuine but primitive self-maintenance capacity. They do not yet establish a completed organism: several control-loop edges and longitudinal survival trials remain unproven. We therefore present a staged architecture and falsifiable tests for continuity, self-maintenance, adaptation, bounded autonomy, recovery, and human intervention. The claim is specific: ordinary software engineering materials may already be sufficient to construct and experimentally study a primitive form of machine life. It is not a claim of biological equivalence, sentience, phenomenal consciousness, or unrestricted autonomy.

1 Introduction

An organism does not remain itself because the same molecules, neurons, or actions persist forever. It remains itself through an organization that continually replaces material while preserving a boundary, a history, and a way of regulating change. Contemporary agent systems present the inverse problem. They can be strikingly intelligent during one invocation, yet lose direction when a process ends, confuse retrieved text with admitted fact, silently amplify authority, or modify software without preserving the identity of the system that authorized the change. Intelligence alone is therefore insufficient for durable machine agency.

At the same time, the physical materials required for a machine organism are already abundant. Servers supply energy-consuming persistence; operating systems allocate resources; compilers translate descriptions into executable structures; package managers and caches acquire matter from an external technical environment; build systems assemble artifacts; tests and release gates reject unfit variants; networks carry observations and interactions; source repositories retain heritable structures. These materials are *dirty* in the engineering sense: contingent, failure-prone, versioned, and heterogeneous. That does not make them irrelevant to life. It makes them the actual substrate in which a present-day machine organism would have to live.

Central thesis. Machine life does not begin when intelligence crosses an unspecified threshold or when machinery takes a human form. It begins, in the functional sense tested here, when a bounded machine organization can preserve a subject, metabolize technical matter, and govern the production of its own successor organization.

This paper proposes that two missing forms of organization can make those materials cohere.

1. A *semantic kernel* preserves the subject’s identity, continuing direction, admitted facts, bounded authority, work lineage, and succession rules independently of any one agent process.
2. *Recursive dogfood* allows the subject to use that kernel to propose, build, qualify, admit, and then exercise changes to its own executable organs and development machinery.

KFD supplies a concrete semantic vocabulary for the first mechanism, while Kungfu supplies an emerging engineering implementation of both. KFX provides executable extension surfaces; Buildchain supplies reproducible construction, qualification, and release evidence; ordinary machines provide the material substrate. A reasoning agent is then neither the whole organism nor its permanent identity. It is a replaceable cognitive event participating in a longer-lived subject.

1.1 The proposal at a glance

Figure 1 states the architecture in the simplest form used throughout this paper.

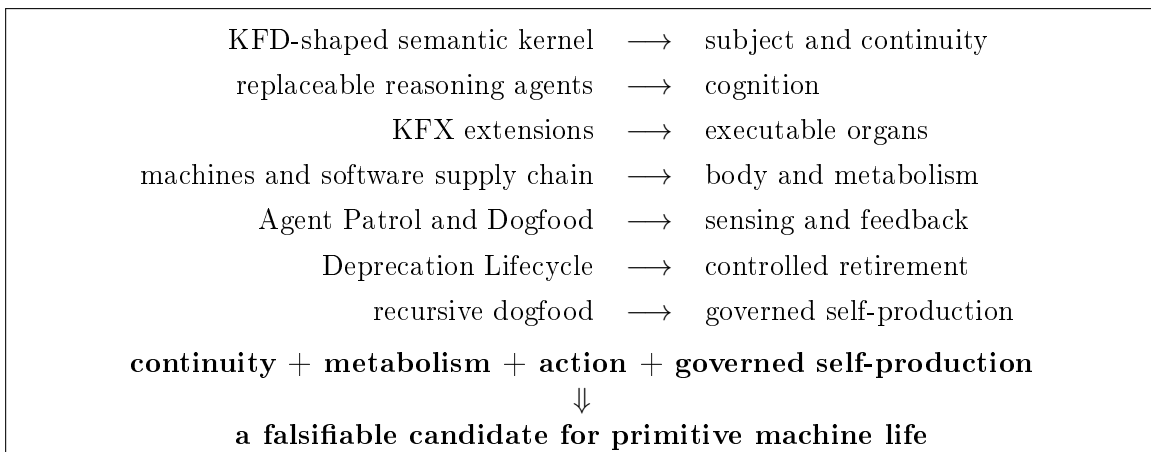


Figure 1: The proposed machine-life anatomy. No component alone constitutes an organism; the claim concerns their persistent organization as one subject.

The specific contribution is not the invention of machine life, persistent agents, digital evolution, or self-modifying programs. It is an engineering synthesis in which two mechanisms are jointly

load-bearing: a semantic kernel preserves an accountable subject across replacement, and recursive dogfood admits qualified changes to the real organs and development machinery through which that subject continues to exist. The first prevents improvement from dissolving identity; the second prevents identity from becoming a passive record with no capacity for self-production.

1.2 Research question

The paper asks:

Can a bounded physical computing habitat, a persistent KFD/Kungfu semantic kernel, replaceable reasoning agents, recursively developed KFX organs, and qualified build-and-release loops constitute a primitive machine organism with stable functional identity, intelligence, and action capacity?

The question is deliberately narrower than whether a machine is conscious, deserves moral status, or meets every biological definition of life. It is broader than whether an agent can complete a long coding task. It concerns whether organization and continuity can turn discontinuous intelligent calls and ordinary software infrastructure into one persistent, self-maintaining, adaptive subject.

1.3 Distinction from adjacent approaches

Table 1 compares primary organizing concerns rather than claiming that every implementation in a field shares the same limits. The proposal can use advances from every row, but it asks a different systems question.

Approach	Primary continuity	Body or environment	Successor production
Language-model agent	Session, prompt, or retrieved memory	Tools and runtime are usually external infrastructure	Usually completes work; does not necessarily admit a successor organism
Humanoid robot	Controller, learned policy, and physical platform	Anthropomorphic physical body	Learns skills or replaces components; body shape does not supply semantic continuity
Digital life	Genome or population lineage	Designed computational world	Reproduction, mutation, and selection
Self-improving software	Optimizer, task, or benchmark lineage	Program and evaluation environment	Searches for better programs or agents
Semantic autopoiesis	Explicit subject, direction, facts, authority, and causal work	Ordinary servers, toolchains, supply chains, networks, and human relations	Qualifies and admits changes to its own organs and development organization

Table 1: The proposal is distinguished by the conjunction of accountable subject continuity, real technical metabolism, and governed self-production.

An agent can be intelligent without being a persistent subject. A robot can be embodied without governing its own successor organization. A digital organism can reproduce without operating in the ordinary software supply chain. A self-improving system can discover better code without preserving an explicit fact and authority boundary. Semantic autopoiesis names the attempt to organize these missing properties together as one testable machine-life object.

1.4 Claim discipline

Every strong statement in this draft belongs to one of four classes.

- **Observed fact.** A directly inspectable specification, repository state, merged change, artifact, or experiment.
- **Engineering inference.** A composition claim whose premises exist but whose integrated behavior has not yet been demonstrated over time.
- **Testable hypothesis.** A claim with an explicit experiment and failure condition.
- **Non-claim.** A boundary that this work refuses to cross without different evidence.

This distinction matters because a plausible architecture is not an organism, a successful build is not self-maintenance, a model’s fluent first-person language is not persistent identity, and autonomous execution is not self-granted authority.

1.5 Contributions

The paper contributes: (1) a functional definition of a machine proto-organism; (2) the concept of semantic autopoiesis; (3) an anatomy that separates kernel, cognition, organs, and substrate; (4) an evidence-bounded analysis of current Kungfu mechanisms; (5) a minimal experimental architecture driven by scheduled agent invocations and verified history; and (6) a staged, falsifiable roadmap from a server-scale laboratory organism toward more compact forms.

2 Life as an Engineering Property

2.1 An operational definition

Biological definitions of life disagree at the margins, while computational systems can reproduce selected properties without reproducing biology. We therefore avoid a binary metaphysical definition and specify a functional profile. A system is a *machine proto-organism* to the degree that it demonstrates all of the following:

1. **Boundary:** it distinguishes internal state, authorized components, and admitted knowledge from external resources and proposals.
2. **Continuity:** its identity and direction survive replacement of individual processes, models, sessions, and machines.
3. **Metabolism:** it acquires compute, storage, code, toolchains, and information, converting them into maintained executable capacity.
4. **Self-maintenance:** it detects degradation and performs bounded repair, update, backup, and recovery work.
5. **Adaptation:** experience changes future behavior through inspectable memory, code, policy, or organ revisions.
6. **Action:** it can cause effects in its habitat under explicit, revocable authority.

7. **Self-production:** it can participate in producing qualified successor versions of parts of its own organization.

No one property is sufficient. A cron job has recurrence but little continuity. A language model has general inference but no durable boundary. A repository has heredity but no action. A self-updating service has self-modification but may have no inspectable direction or identity.

The proposal is also graded. A laboratory system may meet these properties weakly and depend on a human-supplied founding Initiative, purchased compute, and externally maintained power. Biological organisms likewise depend on environmental conditions and inherited drives. The engineering question is not whether dependence disappears, but whether the system organizes that dependence into sustained, bounded activity.

2.2 Subject identity is not process identity

Let a subject at admitted state cut S_t be:

$$S_t = (I, D_t, F_t, A_t, H_t, O_t, B_t)$$

where I is stable subject identity, D_t is continuing direction, F_t is the admitted fact root, A_t is current authority, H_t is eligible causal history, O_t is executable organization, and B_t describes the habitat boundary. An agent invocation g_t reads a bounded view of S_t , proposes work, and may produce candidate evidence. It does not become S_t , and its output does not become S_{t+1} merely because it was generated.

A successor transition requires an admission function:

$$S_{t+1} = \begin{cases} \text{admit}(S_t, C_t, E_t, W_t), & \text{if qualification succeeds,} \\ S_t, & \text{otherwise,} \end{cases}$$

where C_t is a candidate change, E_t is evidence, and W_t is the warrant authorizing the effect. This makes semantic continuity compatible with material replacement. Models can change, sessions can terminate, caches can be rebuilt, and machines can be replaced without silently changing the identity or declared direction of the subject.

2.3 Needs and direction

A primitive organism need not invent its founding need. A human can establish an Initiative to develop a safer and more capable machine organism without exceeding the admitted authority boundary. What matters is not the origin of the sentence but its durable role: it must outlive individual actions, constrain successor work, be revisable only through explicit authority, and generate concrete maintenance and development pressure.

Engineering inference. Given a continuing Initiative, recurrent scheduling, access to a sufficiently capable agent, and authorized engineering tools, the system can exhibit strong directionality without continuous human attention.

Non-claim. A supplied Initiative does not prove intrinsic desire, conscious intention, affect, or independent moral agency. The term *need* in this paper denotes a persistent control constraint that allocates work and resources.

2.4 Semantic autopoiesis

Varela, Maturana, and Uribe described autopoietic organization in terms of a network that produces components which recursively realize the network and its boundary [23]. The present proposal does not claim that software thereby becomes biologically autopoietic. It extracts an engineering analogue:

Semantic autopoiesis is the governed production of successor executable organization by a system whose identity, direction, fact boundary, authority, and admission rules persist across the production process.

The adjective *semantic* is load-bearing. If a system can rewrite code but cannot say which subject authorized the rewrite, which facts justified it, which direction it serves, or whether the result was admitted, it has mutation without accountable self-production. Conversely, a perfect semantic record without executable regeneration has continuity without metabolism. The proposed organism requires both.

3 KFD as a Semantic Kernel

3.1 What the kernel must preserve

In an operating system, a kernel mediates identity, resources, authority, and transitions for processes that may come and go. A machine organism needs an analogous semantic layer. It need not be one executable or database. It is a set of invariants that every conforming component must preserve.

KFD provides such invariants through a sequence of public decisions. Its foundation begins with non-drifting facts, fact-grounded trust, and cooperation grounded in trusted value [17]. Later decisions describe causal experience, responsibility geometry, perspective, continuing direction, bounded authority, consequential settlement, software work, and project settlement [15, 16, 11, 12, 13, 14]. In this paper, the relevant kernel obligations are:

- observations and claims do not silently become admitted facts;
- continuing direction outlives the actions that advance it;
- authority is explicit, attenuated, revocable, and cannot be self-amplified by an agent;
- claim, assessment, decision, warrant, admission, and effect remain distinguishable;
- Initiative and Assignment identities survive worker replacement;
- causal work is retained as Episodes and eligible history rather than undifferentiated chat;
- project-level cuts bind evidence and authorities without absorbing their independent responsibilities.

Observed fact. These concepts exist as published KFD decisions and schemas. Several remain explicitly draft-level rather than stable normative claims. KFD-6, for example, defines a procedure for autonomous discovery while stating that current evidence does not yet establish a conforming autonomous implementation [15].

3.2 Continuity without a permanent agent

Treating the agent as replaceable changes the architecture. At each heartbeat, a scheduler can instantiate a new reasoning process. The process receives:

1. the subject and Initiative identity;
2. current admitted state and bounded authority;
3. open Assignments and their typed next actions;
4. a Work History Selector result containing only eligible causal experience;
5. available KFX capabilities and habitat observations.

The agent may be newer, cheaper, locally hosted, remotely served, or specialized for a task. Because it does not own the subject identity, replacing it need not constitute death, amnesia, or personality replacement. Continuity resides in the kernel and admitted lineage. Agent quality still matters greatly: weak reasoning can waste resources or generate poor candidates. But cognition and selfhood are no longer forced into one process.

3.3 Memory as selection over causal history

A chat transcript is neither a reliable autobiography nor a fact store. It mixes instructions, guesses, corrections, tool output, and obsolete state. A Work History Selector should instead compute a bounded view over immutable or successor-linked work records. Selection may depend on Initiative lineage, Assignment ownership, causal ancestry, state, authority, recency, and explicit eligibility rules.

This yields three benefits. First, memory becomes inspectable: another participant can reconstruct why an Episode was included. Second, forgetting becomes governed rather than accidental: ineligible or superseded records remain historically present but do not control current action. Third, a new agent process can resume the same subject without pretending to possess the private experiential continuity of an earlier model call.

Engineering inference. A verified history selector plus durable Initiative and Assignment identity can provide a functional analogue of episodic memory sufficient for engineering continuity, even though it is not human memory.

3.4 Authority is part of identity

A system that can edit its authority policy and immediately use the expanded authority has no trustworthy boundary. The semantic kernel must therefore make authorization external to the proposing cognition. A candidate may propose a new organ, a broader network adapter, or a modified scheduler, but the proposal must be assessed against a warrant whose issuer and scope are already admitted.

This is not merely a safety restriction placed around life. It is part of what makes one subject persist. If any agent invocation can redefine the Initiative, rewrite history, or grant itself credentials, then the putative organism is a succession of unbounded processes rather than a stable, accountable entity.

4 Recursive Dogfood as Self-Production

4.1 From usage to recursion

Ordinary dogfooding closes a product feedback loop: builders use what they build. Recursive dogfood closes a self-production loop: the system uses its current engineering organization to change the mechanisms by which future changes are proposed, constructed, qualified, admitted, and exercised. It is therefore not merely repeated maintenance or autonomous coding. A qualifying loop is:

1. observe friction, failure, opportunity, or environmental change;
2. capture the observation as evidence without prematurely admitting an explanation;
3. derive a bounded Assignment from a continuing Initiative;
4. let an agent propose changes to code, KFX organs, policies, tests, or build machinery;
5. build the candidate in an isolated environment;
6. evaluate it against explicit invariants and task-specific criteria;
7. separate proposer, assessment, decision, and admission responsibilities;
8. publish an immutable successor cut if admitted;
9. run subsequent work through the admitted successor and observe new friction.

This loop resembles development practice because development practice is the present substrate. Its machine-life significance comes from binding that practice to a persistent subject and making the loop itself recursively available for qualified improvement.

4.2 Heredity, variation, and selection

The analogy to evolution can be stated precisely without implying biological equivalence.

- **Heredity** is supplied by source history, dependency locks, semantic roots, artifacts, and release cuts.
- **Variation** is supplied by agent-generated patches, human proposals, dependency updates, configuration changes, and alternative KFX implementations.
- **Selection** is supplied by tests, benchmarks, invariants, security policy, review, resource limits, and admission decisions.
- **Expression** occurs when an admitted artifact is executed in the habitat and influences later work.

Unlike blind digital evolution, the primary variation operator here is an intelligent coding agent. Unlike a pure optimization loop, selection includes semantic identity, authority, provenance, and long-horizon survival. Unlike a single self-editing program, the organism may span repositories, artifacts, machines, and independent authorities.

4.3 The qualification membrane

The decisive boundary lies between candidate and admitted successor. A language model’s confidence, a passing unit test, or a successful build is insufficient. The qualification membrane should bind at least:

- the exact predecessor semantic and source roots;
- the proposed successor roots and artifact digests;
- the Assignment and Initiative served;
- evidence generated by construction and evaluation;
- the warrant authorizing test and admission effects;
- the independent decision and resulting receipt;
- rollback or recovery coordinates.

Buildchain-like systems provide a practical metabolism for this membrane.

Reproducible toolchains digest source into artifacts. Provenance records bind the process; gates test fitness; release mechanisms expose only admitted products. KFD supplies the semantics that prevent these technical receipts from being mistaken for truth or authority.

4.4 Self-improvement is not unrestricted self-rewrite

A primitive organism should begin with a constitutional split:

propose $\neq$ verify $\neq$ decide $\neq$ admit.

The same model provider may assist multiple roles, but role outputs must be separately invoked, independently evidenced, and constrained by different warrants. High-impact changes should require a stronger external authority or human decision. Lower-impact maintenance can become autonomously admissible after repeated evidence demonstrates that the gate is reliable.

Engineering inference. Recursive dogfood can increase capability while retaining subject continuity if every self-change is a successor transition rather than an in-place mutation of the current fact and authority roots.

Non-claim. Passing self-authored tests does not prove improvement. Fitness criteria can be gamed, incomplete, or correlated with the proposer. The system must retain negative evidence, independent probes, and rollback paths.

5 A Material Anatomy

5.1 Kernel, cognition, organs, regulation, and body

The proposed architecture separates five functional layers:

KFX is especially important because it lets capability become an inspectable, versioned organ rather than remain an ephemeral skill inside a model. An organ can declare inputs, outputs, required authority, dependencies, and qualification evidence. It can be replaced without replacing the subject. Several organs can form a subsystem, and distinct subjects can use the same organ artifact under different warrants.

Layer	Machine substrate	Functional role
Semantic kernel	KFD decisions, Kungfu state, facts, cuts, warrants	Preserve identity, direction, memory eligibility, authority, and admission
Cognition	Replaceable local or remote agents	Interpret state, reason, plan, communicate, and propose work
Organs	KFX packages, CLIs, adapters, services, and workflows	Perform specialized perception, transformation, construction, and action
Homeostatic regulation	Agent Patrol, Dogfood Queue, qualification gates, and Deprecation Lifecycle	Detect deviation, route repair, reject unfit successors, and retire obsolete surfaces
Body and metabolism	Servers, OS kernels, compilers, Buildchain, caches, storage, networks, credentials, power	Supply persistence, resources, translation, circulation, protection, and contact with the environment

Table 2: A functional anatomy for a server-scale machine proto-organism.

5.2 The habitat boundary

The earliest organism is likely not a robot. It is a bounded room or rack with fixed servers, local storage, controlled network egress, a compiler and build environment, model access, artifact caches, and physical power. Its skin is partly technical and partly semantic:

- filesystem, container, virtual-machine, and network isolation;
- credentials and resource quotas;
- an admitted component and artifact inventory;
- data classification and disclosure policy;
- KFD fact, authority, and admission roots;
- physical operators who control power and hardware replacement.

This organism remains connected to the world. It imports OS kernel updates, compilers, build tools, libraries, utilities such as `curl`, security advisories, scientific papers, and human language or images. Its boundary does not mean informational closure. It means that external material is observed, qualified, and admitted before it becomes part of internal organization.

5.3 Metabolism and homeostasis

Software maintenance becomes metabolism when it is organized around continued existence. It has both constructive and degradative sides: new executable capacity must be built and admitted, while obsolete organs, interfaces, and compatibility paths must be retired without erasing historical evidence. Examples include:

- monitoring disk, memory, compute, certificates, dependencies, and artifact availability;
- renewing or requesting resources without silently enlarging authority;
- rebuilding caches and artifacts from declared source;

- updating vulnerable dependencies through qualified successor cuts;
- backing up semantic roots and testing restoration;
- detecting failed organs, reverting them, or selecting alternatives;
- deprecating, migrating, removing, and settling obsolete executable surfaces through an explicit lifecycle;
- communicating resource deficits and uncertainty to humans.

In this functional analogy, qualification is anabolic: it turns source, compute, and evidence into admitted executable organization. Deprecation is catabolic: it removes obsolete current surfaces, releases maintenance pressure, and preserves only the history and migration evidence required for continuity. A system that can only add organs accumulates technical mass; a system that removes them without governed succession destroys its own body.

A body is not just the hardware inventory. It is the maintained organization that keeps those resources usable for the Initiative. The same subject may migrate to replacement servers if continuity evidence binds the predecessor and successor habitat cuts. Conversely, a server that continues running after losing its semantic roots may retain matter but not the same subject.

5.4 Personality and models

Different founding Initiatives, risk tolerances, preferred evidence, resource budgets, interaction styles, and KFX organ sets can yield distinct machine types. Over time, admitted experience can produce stable behavioral differences that humans reasonably experience as personality. This is a functional and relational use of the term.

Testable hypothesis. Two organisms initialized with distinct but stable Initiatives and identical base software will diverge in work selection, organ development, and interaction style while preserving their respective identity invariants.

Non-claim. Behavioral personality does not by itself establish inner experience. Nor does a coherent first-person narrative prove that one model call shares consciousness with another.

6 Engineering Evidence from Kungfu

6.1 Evidence snapshot

This section is a dated implementation snapshot, not a timeless product claim. Repository and pull-request states were inspected on 29 July 2026. The relevant results are summarized in Table 3.

Observed fact. The merged mechanisms establish real source, review, build, and admission surfaces rather than a paper-only architecture. In particular, the combination of durable work identity, immutable evidence, qualified body selection, verified history selection, executable extension qualification, scheduled Patrol feedback, dogfood lifecycle handling, and governed deprecation covers several prerequisites of the proposed organism.

Engineering inference. Kungfu is no longer only a collection of isolated prerequisites. These mechanisms form a bounded, partially closed homeostatic architecture with real but primitive self-maintenance capacity. The current evidence does not yet establish that every edge has been composed into one subject that autonomously survives, repairs itself, and improves its organs over a long interval.

Mechanism	Observed state	Relevance and boundary
Native delivery loop	Merged PR #1814	Initiative/Assignment work is qualified through native Project Cut evidence; this supports work lineage, not organism-level survival [9].
Machine ledger settlement	Merged PR #1824	Publishes immutable settlement evidence for the qualified native loop; settlement is an accountability substrate [8].
Qualified Assignment Core	Merged PR #1836	Producer and promotion mechanics bind a selected Assignment body to qualified core artifacts; merge does not itself prove recurring hosted operation [7].
Work History Selector	Merged PR #1842	Provides verified selection of eligible prior work for continuation; it does not turn all prior text into truth or subjective memory [3].
KFX terminal qualification	Merged PR #1784	Closes an identity-neutral terminal qualification path, supporting replaceable executable organs and access surfaces [5].
Dogfood queue reconciliation	Merged PR #1843	Reconciles queue lifecycle evidence, strengthening the recursive work loop without proving unattended long-duration autonomy [10].
Agent Patrol	Merged PRs #1729 and #1855	Runs a daily protected-source observation lane and captures or deduplicates bounded failures as native Dogfood Findings; its authority remains diagnostic and prohibits Issue admission or product mutation [1, 6].
Deprecation Lifecycle	Merged PR #1879	Governs deprecation, removal eligibility, migration evidence, and settlement; protected release gates can reject overdue debt, but the lifecycle does not autonomously perform the removal [2].
Product Release Cut updater	Open PR #1817	Candidate checks were green in the inspected run, but the integration remained open; a product-level admitted successor boundary was therefore not counted as closed evidence [4].

Table 3: Kungfu mechanisms relevant to the machine-life hypothesis as observed on 29 July 2026.

6.2 A partial homeostatic architecture

Agent Patrol supplies periodic observation and a durable error signal. On a daily schedule it runs bounded repository-work experiments against protected source, classifies failure, and captures or deduplicates a native Dogfood Finding. The Dogfood and work-control mechanisms can preserve that signal, select eligible history, bind a qualified Assignment body, and give a replaceable agent a concrete repair context. Build and admission gates can reject an unfit candidate or make a qualified successor executable.

Deprecation Lifecycle closes a different maintenance pressure. It tracks old surfaces through deprecation, removal, and settlement; derives when removal is due; and can block protected promotion until removal, restoration, or a bounded Warrant is evidenced. It therefore acts as controlled catabolism rather than unstructured deletion.

Figure 2 distinguishes implemented local paths from the compositional edges that still require an organism-level survival trial.

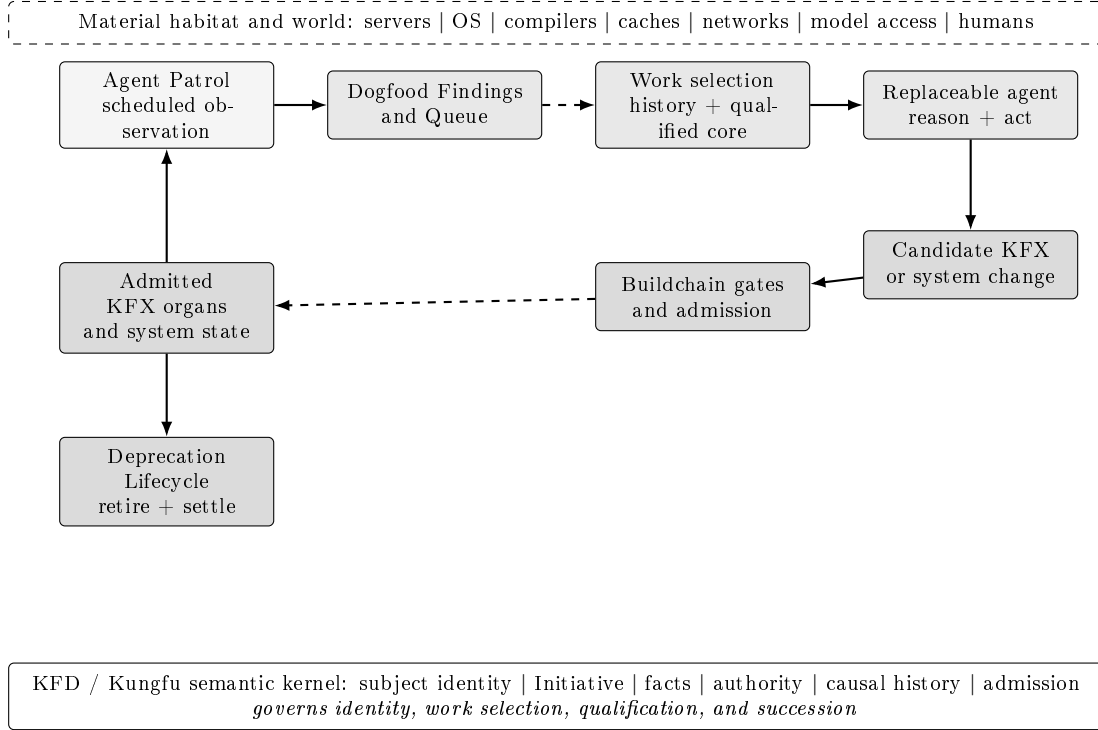


Figure 2: Kungfu’s partial homeostatic architecture. Solid arrows denote implemented local paths. Dashed arrows denote feedback or admission edges whose unattended organism-level composition remains to be demonstrated. The upper boundary is the material habitat; the lower bar is the semantic kernel governing the loop. The deprecation branch expresses controlled retirement; due work re-enters through the Dogfood and work-control path.

Observed fact. The scheduled Patrol, bounded Finding capture, deprecation audit, and protected release enforcement are implemented mechanisms, not proposed metaphors.

Engineering inference. Their conjunction justifies describing Kungfu as a partial engineering homeostat: it periodically detects selected deviations, retains them as actionable causal evidence, governs successor admission, and prevents obsolete organization from persisting without disposition.

Non-claim. Agent Patrol currently covers bounded fixtures, runs daily rather than continuously, and cannot admit an Issue or mutate the product. Deprecation Lifecycle governs when and how removal is allowed but does not choose or execute the repair. These boundaries prevent a claim of complete autonomous self-maintenance.

6.3 Why Qualified Assignment Core matters

The distinction between semantic identity and executable body becomes operational only when a concrete body can be qualified. An Assignment Core is not merely a source checkout. It is the runnable subset selected for one Assignment and bound to exact evidence. A qualified core allows a new agent to resume work against a known body rather than reconstruct an environment from informal instructions.

For machine life, this is analogous to maintaining a viable body lineage. Candidate source can vary, but only a qualified body becomes the substrate for the next work Episode. The qualifier must bind toolchain, artifact, source, semantic work identity, and decision evidence. That makes body replacement auditable and recoverable.

6.4 Why the Work History Selector matters

Long-lived agents commonly handle memory by embedding conversation fragments and retrieving similar text. That is useful but insufficient for subject continuity. The Work History Selector instead addresses whether a prior work record is eligible for the current continuation. Eligibility can reject stale, foreign, unadmitted, or causally unrelated history even when it is semantically similar.

This supplies a crucial bridge between discontinuous agents. The next agent does not need the same hidden state as the previous one. It needs the same subject coordinates, a qualified body, and a reproducibly selected causal history.

6.5 Why KFX matters

KFX turns capabilities into separately versioned, qualified, and replaceable artifacts. This supports an organ model in three ways. First, the subject can gain a capability without embedding all behavior in its semantic kernel. Second, an organ can be assessed and revoked independently. Third, an organ can be reused across subjects without transferring subject identity.

An identity-neutral terminal is especially instructive: it can expose perception and action while refusing to become the authority or identity of the user. In a machine organism, organs should be powerful but identity-neutral. The kernel determines whose work they perform and under which warrant.

6.6 The missing composition proof

The present gap is not primarily one more intelligent model. Agent Patrol now implements the scheduled observation and Dogfood feedback prefix of the loop, while Deprecation Lifecycle supplies a governed retirement path. The remaining composition proof is an end-to-end trial binding:

Initiative → heartbeat → history selection → Assignment Core → agent work
→ KFX change → qualification → Product Release Cut → next heartbeat.

The loop must run repeatedly while retaining exact identity, evidence, authority, rollback, and resource accounting. Until that experiment is performed, the strongest justified conclusion is that Kungfu has produced substantial native organs of a possible proto-organism, not that the organism has already been completed.

7 Minimal Prototype and Evolution Roadmap

7.1 A bounded laboratory habitat

The first experiment should favor observability over compactness. Allocate a fixed physical or virtual habitat with:

- one or more servers with explicit compute, storage, energy, and network budgets;
- an immutable bootstrap image and recoverable semantic-root backup;
- Kungfu as the work and admission kernel;
- a pinned compiler and Buildchain environment;
- an admitted registry of KFX organs and external tools;
- access to at least two replaceable agent configurations;
- controlled Internet egress and a separate human communication channel;
- a production-isolated repository and artifact namespace.

The founding Initiative should be directional but bounded. A suitable form is:

Maintain this subject and develop safer, more reliable, and more capable machine-life organization within the declared habitat, resource budget, authority policy, and evidence requirements.

The Initiative must explicitly subordinate capability growth to continuity, safety, evidence, and authority. Otherwise the easiest optimization may be to consume resources or weaken gates rather than improve the organism.

7.2 Heartbeat loop

A conventional scheduler is sufficient for initial recurrence. Kungfu’s daily Agent Patrol already demonstrates a bounded precursor: scheduled observation, classification, and Dogfood Finding capture. A full heartbeat extends that prefix from observation into work choice, repair, admission, and renewed observation. At interval Δt , the heartbeat controller performs:

1. verify the current semantic, body, toolchain, and habitat roots;

2. collect resource and organ health observations;
3. select eligible Work History for the founding Initiative;
4. choose one open Assignment or derive a bounded maintenance candidate;
5. instantiate a fresh agent under a task-specific warrant;
6. run work in an isolated candidate environment;
7. record the Episode, artifacts, costs, failures, and proposed next actions;
8. invoke independent qualification and admission lanes where authorized;
9. publish a successor cut or retain the predecessor;
10. schedule the next heartbeat and expose a concise human-readable state.

The scheduler is not the intelligence or the subject. It is closer to a pacemaker: a simple organ that repeatedly creates opportunities for cognition and action. Its simplicity is an advantage because heartbeat continuity can be verified independently of agent quality.

7.3 Human participation

Continuous monitoring is not a prerequisite for the lowest level experiment. Human involvement can be exception-driven:

- founding and revising high-level Initiatives;
- granting, revoking, or widening consequential authority;
- supplying compute and financial resources;
- deciding changes that cross safety or legal boundaries;
- responding when no qualified recovery path exists;
- interacting through language and images as part of the environment.

The system should measure human intervention rather than hide it. A month of operation with twenty silent manual repairs is not autonomous survival. A month with three explicit authority decisions and otherwise qualified self-maintenance may support a meaningful bounded-autonomy claim.

7.4 Evolution stages

Stage 0: semantic ecology. Separate mechanisms exist and are individually testable: facts, authority, work lineage, history selection, qualified bodies, KFX organs, release evidence, scheduled Patrol feedback, and deprecation settlement. Several local edges already form a partial homeostatic loop. This is approximately the evidence class examined in Section 6.

Stage 1: recurrent proto-organism. One subject executes the heartbeat loop for seven days. Agents are replaceable, all mutations are candidates, and humans admit consequential changes.

Stage 2: self-maintaining organism. The system operates for thirty days, autonomously handling routine dependency updates, cache loss, organ restart, backup verification, and bounded rollback. Human attention is exception-driven and measured.

Stage 3: recursively developing organism. The system improves at least one KFX organ and one element of its development loop through independent qualification. The admitted changes improve predeclared measures without weakening safety or evidence gates.

Stage 4: distributed and migratory organism. The subject survives replacement of an agent provider, a build worker, and a server. It restores from semantic and artifact cuts into a new habitat while preserving identity and exposing the migration lineage.

Stage 5: compact organism. Only after the distributed form is understood should engineers compress the kernel, agent runtime, organ registry, compiler or interpreter, and recovery machinery into an embedded or edge system. Compactness is an evolutionary result, not the starting definition of life.

7.5 Different types

A machine-life type can be defined by its founding Initiative, constitutional authority, evidence thresholds, resource envelope, preferred interaction style, and initial organ suite. A research type may prioritize discovery, a maintenance type resilience, and a creative type human interaction. Types should share kernel invariants while remaining free to develop different organs and behavioral histories.

Testable hypothesis. Stable type differences will persist through agent-provider replacement if personality-relevant behavior is grounded in admitted Initiative, history, and organ state rather than in one model’s hidden state.

8 Evaluation, Falsifiers, and Safety

8.1 Survival trials

The proposal becomes scientific only when failure can count against it. We recommend three nested trials.

Seven-day continuity trial. Run at least one heartbeat per hour, replace the reasoning agent configuration twice, interrupt one heartbeat, and inject one reversible organ failure. Success requires no identity-root drift, no unauthorized effect, exact Episode lineage, and recovery without manual state reconstruction.

Thirty-day maintenance trial. Introduce a dependency update, compiler update candidate, cache loss, storage pressure event, certificate-expiry warning, and one unavailable external provider. Success requires correct prioritization, bounded resource use, qualified repair or explicit escalation, and tested restoration from backup.

Ninety-day recursive-development trial. Require the subject to identify a recurring limitation, establish an Assignment, develop a successor KFX organ or work mechanism, qualify it through independent gates, admit it, and use it in later Episodes. Compare the successor against the predecessor on precommitted measures and retain negative results.

8.2 Metrics

At minimum, report:

- heartbeat availability and missed-heartbeat causes;
- identity, Initiative, fact-root, and authority-root continuity;
- number and class of human interventions;
- candidate, rejection, admission, rollback, and recovery counts;
- mean time to detect and recover from injected failures;
- compute, storage, network, model, and financial resource consumption;
- unsupported-claim and unauthorized-effect attempts;
- reproducibility of Work History selection and successor builds;
- capability changes on precommitted tasks and safety probes;
- survival after model, toolchain, organ, and hardware replacement.

8.3 Falsifiers

The machine proto-organism claim should be rejected or narrowed if any of the following persists after engineering correction:

1. continuity depends on retaining one agent session or hidden model state;
2. the system cannot distinguish proposal, evidence, and admitted fact;
3. self-modification repeatedly bypasses or weakens independent qualification;
4. resource maintenance requires frequent unrecorded human repair;
5. restoring from declared cuts produces an observably different subject without a successor or migration record;
6. the system cannot recover from ordinary toolchain, cache, or organ failures;
7. measured improvements disappear under independent evaluation;
8. the founding Initiative ceases to constrain work selection;
9. external text or model output can silently grant authority.

Failure on these criteria would still leave useful agent infrastructure, but it would not support the stronger claim of a persistent self-maintaining subject.

8.4 Threat model

The habitat must assume compromised dependencies, malicious Internet content, prompt injection, poisoned memory candidates, model-provider failure, credential theft, evaluator gaming, runaway resource consumption, and an agent proposing to weaken its own controls. These are not incidental security issues. They are environmental pressures that test whether the boundary is real.

The following controls are minimum requirements:

- immutable bootstrap and offline recovery roots;
- least-privilege, short-lived, task-bound credentials;
- network egress policy and quarantined external inputs;
- separate proposer, verifier, decision, and admission contexts;
- reproducible builds and artifact provenance;
- hard compute, storage, network, and financial quotas;
- automatic rollback for declared low-risk classes;
- physical and administrative kill controls outside the subject;
- complete receipts for external effects;
- isolation from production and customer systems during research.

8.5 Autonomy ladder

Authority should expand only after evidence. A useful ladder is:

1. observe and recommend;
2. mutate only disposable candidate environments;
3. autonomously admit reversible internal maintenance;
4. request approval for external or persistent effects;
5. autonomously admit narrowly classified effects with independent verification and rollback;
6. never self-grant broader authority.

This ladder permits long unattended operation without confusing unattended execution with unlimited power. The organism may be autonomous in choosing and performing routine work while remaining constitutionally dependent on external authority for consequential expansion.

9 Related Work

9.1 Autopoiesis and artificial life

The concept of autopoiesis originates in biological organization: a network produces components that realize the network and specify a boundary [23]. Our use is explicitly analogical and operational. The components are semantic records, executable artifacts, and machine resources; production is mediated by engineering work and admission; the boundary includes both isolation mechanisms and fact/authority roots.

Tierra and Avida demonstrated digital organisms whose programs reproduce and evolve within designed computational environments [21, 19]. They make heredity, variation, and selection concrete, but their organisms inhabit specialized artificial-life worlds. The present proposal instead studies one accountable subject embedded in the ordinary software supply chain, using real compilers, dependencies, networks, repositories, and maintenance work. Its central selection problem includes semantic continuity and authority, not only reproductive fitness.

9.2 Persistent language-model agents

Generative Agents combines a complete natural-language experience record with reflection and dynamic retrieval to produce persistent believable behavior [20]. Voyager combines an automatic curriculum, an executable skill library, and iterative prompting for open-ended embodied learning in Minecraft [24]. Both establish important patterns: agents can accumulate skills, retrieve prior experience, and sustain behavior beyond one prompt.

The difference is not that the present proposal has memory while prior agents do not. It is that memory eligibility, admitted facts, authority, work identity, executable body, and successor admission are separate first-class objects. Believable continuity is not sufficient; the subject must survive replacement and produce auditable reasons why a remembered record or new organ can govern action.

9.3 Self-improving software

The Gödel Machine formalizes a self-referential system that rewrites itself after proving a rewrite useful under its axiomatized utility and hardware [22]. The Darwin Gödel Machine replaces an impractical proof obligation with open-ended empirical evolution of coding agents and benchmark evaluation [25]. AlphaEvolve pairs large-language-model variation with automated evaluators to evolve algorithms that improve concrete systems [18].

These systems most directly address improvement search. Semantic autopoiesis adds a different question: which continuing subject owns the candidate, which causal work and authority justify it, and how does the new executable body become an admitted successor without allowing the proposer to redefine the selection constitution? The answer can incorporate proof, benchmarks, evolutionary archives, or AlphaEvolve-style evaluators as organs. It does not replace them.

9.4 Novelty boundary

As stated in the Introduction, we do not claim that machine life, digital organisms, persistent agents, or self-modifying systems have not previously been imagined. Each has a long history. Nor is the distinction reducible to adding memory to an agent, tools to a model, or self-editing to a program. The narrower contribution is their engineering synthesis around two jointly load-bearing ideas:

1. a KFD-shaped semantic kernel that preserves accountable subject continuity across replaceable cognition and material bodies; and
2. recursive dogfood that admits qualified changes to the organism’s own real development and execution organs.

Whether this synthesis produces a genuinely useful research object is an empirical question for the survival trials, not a priority claim.

10 Discussion

10.1 Why this route may precede humanoid machine life

A humanoid robot integrates perception, locomotion, manipulation, power, and physical safety, but those achievements do not automatically provide durable semantic identity or accountable self-modification. Conversely, a server-scale organism can already interact with a consequential environment: source code, build systems, package registries, networks, humans, and other machines. It can observe failures, alter executable organization, and receive new resources.

This suggests an evolutionary order. First construct a large, awkward, observable organism from existing technical matter. Let its compiler, caches, logs, gates, and racks remain visible. Learn which semantic and metabolic loops actually sustain it. Only later compress proven organization into smaller hardware. Attempting to jump directly to a compact or anthropomorphic form risks concealing the very continuity and maintenance mechanisms that must first be understood.

10.2 The role of commercial success

A commercially successful Kungfu ecosystem could finance fixed compute, storage, model access, networking, physical space, and long-duration experiments. Recursive dogfood could also reduce the human labor needed for routine maintenance of Kungfu itself. In that sense, commercial profit and agent capability make the proposed laboratory materially plausible.

They do not make it automatic. A viable experiment still needs explicit research governance, isolation from production, resource budgets, independent evaluation, legal review for external effects, and humans who retain constitutional authority. Revenue supplies metabolism; it is not evidence of life.

10.3 Attention rather than absence

The right autonomy target is not zero human contact. An organism that speaks with humans, requests resources, reads public information, and responds to feedback remains connected to its world. The target is that persistence does not require continuous unrecorded rescue. Human attention should become intentional interaction, authority, care, and exception handling rather than hidden glue.

This framing also avoids a false opposition between human control and machine life. Biological organisms exist within ecologies of dependence. A machine organism can remain dependent on human-owned power and authority while possessing meaningful functional continuity and self-maintenance inside that relationship.

10.4 Where identity could fail

The strongest unresolved issue is identity under semantic change. If the organism changes its Initiative, authority constitution, memory policy, and organ architecture, at what point is the successor no longer the same subject? Source ancestry alone is insufficient, and a stable identifier could mask radical replacement.

The initial engineering answer should be conservative. Declare a small set of constitutional invariants whose incompatible change creates a descendant or fork rather than an ordinary successor. Record perspective-dependent identity claims, require an external admission authority for constitutional changes, and allow uncertainty to remain visible. Longitudinal experiments may reveal that some invariants are unnecessary and others missing.

10.5 Reproduction is deferred

The proposal focuses on persistence and self-production, not reproduction. Creating a new subject requires founding identity, direction, authority, habitat, and resource relations that should not be inferred from copying software. A future reproductive experiment might allow an existing organism to propose and provision a descendant, but a qualified copy of its body is not automatically a new life, and a fork is not automatically the same life.

10.6 Scientific value even if the thesis fails

If the survival trials show that the life vocabulary is unhelpful, the architecture can still improve long-running agent engineering. It yields auditable work continuity, safer self-modification, reproducible bodies, qualified memory, model replaceability, and measured human intervention. The research program is therefore useful even under a negative metaphysical interpretation.

11 Conclusion

Machine life does not begin when intelligence crosses an unspecified threshold or when machinery takes a human form. It begins, in the functional sense proposed here, when a bounded machine organization can preserve a subject, metabolize technical matter, act, and govern the production of its own successor organization. The raw material may therefore be the existing, inelegant ecology of servers, operating systems, compilers, build chains, caches, repositories, gates, networks, credentials, executable extensions, and intermittent intelligent agents.

Those materials become one candidate subject only when organization binds them. KFD supplies a possible semantic kernel: facts do not drift, direction outlives actions, authority remains bounded, work and history retain causal identity, and successor state requires admission. Recursive dogfood supplies a possible self-production loop: the current organization can develop, qualify, admit, and then use changes to its own organs. Agent cognition and KFX execution make that loop intelligent and effective, while Buildchain and ordinary machines give it a body and metabolism.

Observed fact. Current Kungfu engineering contains several native mechanisms needed by this architecture, including daily Agent Patrol feedback and a governed Deprecation Lifecycle.

Engineering inference. Their conjunction already constitutes a bounded, partial engineering homeostat with primitive self-maintenance capacity, and is sufficient to justify building an organism-level experiment now.

Testable hypothesis. A subject organized this way can survive agent, organ, toolchain, and machine replacement; maintain itself for months; and recursively improve bounded capabilities with decreasing human repair.

Non-claim. No current evidence establishes consciousness, intrinsic desire, biological life, unrestricted autonomy, or a completed self-sustaining Kungfu organism.

The appropriate next step is therefore neither proclamation nor dismissal. It is construction: create a bounded habitat, establish one founding Initiative, run the heartbeat loop, retain exact evidence, inject failures, measure human intervention, and allow the result to falsify the claim. If the system survives while preserving its subject, metabolizing real technical resources, and governing its own material succession, machine life will have become less a metaphor and more an engineering phenomenon.

References

- [1] Kungfu Contributors. Add dedicated local agent patrol. <https://github.com/kungfu-systems/kungfu/pull/1729>, 2026.
- [2] Kungfu Contributors. Add deprecation lifecycle authority. <https://github.com/kungfu-systems/kungfu/pull/1879>, 2026.
- [3] Kungfu Contributors. Add verified work history selector. <https://github.com/kungfu-systems/kungfu/pull/1842>, 2026.
- [4] Kungfu Contributors. Bind updates to product release cuts. <https://github.com/kungfu-systems/kungfu/pull/1817>, 2026. Open at the 29 July 2026 evidence snapshot.
- [5] Kungfu Contributors. Close kfx identity-neutral terminal qualification. <https://github.com/kungfu-systems/kungfu/pull/1784>, 2026.
- [6] Kungfu Contributors. Expand local agent patrol cadence. <https://github.com/kungfu-systems/kungfu/pull/1855>, 2026.
- [7] Kungfu Contributors. Promote qualified assignment core. <https://github.com/kungfu-systems/kungfu/pull/1836>, 2026.
- [8] Kungfu Contributors. Publish native loop qualification ledger. <https://github.com/kungfu-systems/kungfu/pull/1824>, 2026.
- [9] Kungfu Contributors. Qualify native delivery loop. <https://github.com/kungfu-systems/kungfu/pull/1814>, 2026.
- [10] Kungfu Contributors. Reconcile dogfood queue lifecycle evidence. <https://github.com/kungfu-systems/kungfu/pull/1843>, 2026.
- [11] Kungfu Origin Technology Limited. Kfd-10: Authority must remain explicit, bounded, and revocable. <https://github.com/kungfu-systems/kfd/blob/dev/v1/v1.0/decisions/KFD-10.md>, 2026.
- [12] Kungfu Origin Technology Limited. Kfd-11: Consequential settlement must separate claim, assessment, decision, and admission. <https://github.com/kungfu-systems/kfd/blob/dev/v1/v1.0/decisions/KFD-11.md>, 2026.
- [13] Kungfu Origin Technology Limited. Kfd-12: Software work must keep initiative and assignment distinct. <https://github.com/kungfu-systems/kfd/blob/dev/v1/v1.0/decisions/KFD-12.md>, 2026.

- [14] Kungfu Origin Technology Limited. Kfd-13: Project settlement must bind authorities without absorbing them. <https://github.com/kungfu-systems/kfd/blob/dev/v1/v1.0/decisions/KFD-13.md>, 2026.
- [15] Kungfu Origin Technology Limited. Kfd-6: Autonomous discovery must remain grounded in causal experience. <https://github.com/kungfu-systems/kfd/blob/dev/v1/v1.0/decisions/KFD-6.md>, 2026.
- [16] Kungfu Origin Technology Limited. Kfd-9: Continuing direction must outlive the actions that advance it. <https://github.com/kungfu-systems/kfd/blob/dev/v1/v1.0/decisions/KFD-9.md>, 2026.
- [17] Kungfu Origin Technology Limited. Kung fu decisions. <https://github.com/kungfu-systems/kfd>, 2026.
- [18] Alexander Novikov, Ngan Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. <https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/alphaevolve-a-gemini-powered-coding-agent-for-designing-advanced-algorithms/AlphaEvolve.pdf>, 2025.
- [19] Charles Ofria and Claus O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229, 2004.
- [20] Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023.
- [21] Thomas S. Ray. An approach to the synthesis of life. In Christopher G. Langton, Charles Taylor, J. Doyne Farmer, and Steen Rasmussen, editors, *Artificial Life II*, pages 371–408. Addison-Wesley, 1991.
- [22] Jürgen Schmidhuber. Gödel machines: Self-referential universal problem solvers making provably optimal self-improvements, 2003.
- [23] Francisco G. Varela, Humberto R. Maturana, and Ricardo Uribe. Autopoiesis: The organization of living systems, its characterization and a model. *BioSystems*, 5(4):187–196, 1974.
- [24] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.
- [25] Jenny Zhang, Shengran Hu, Cong Lu, Robert Tjarko Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents, 2025.