KUNGFU WHITE PAPER · JULY 2026

Kungfu: Continuity for Agent Work

A local-first product and runtime for Projects, Work, and replaceable Agents.

Work should not disappear when a chat ends, restart when an Agent changes, or become complete merely because a process exits. Kungfu gives Work durable identity, governed evidence, bounded authority, and an independently reviewable path to continuation and completion.

AUDIENCE
Agent users, developers, operators, researchers, and early product evaluators.

PRODUCT FRAME
Project, Work, and Agent form the complete first layer; deeper authority appears only when needed.

PHILOSOPHY
Facts must not drift, trust must start from facts, and cooperation must start from trusted value.

PROOF PATH
Agent Work Lab, exact artifact evidence, real Project continuity, and explicit public claim boundaries.

Keren Dong
Kungfu Origin Technology Limited

https://kungfu.tech

SECTION 1

Executive Summary

An Agent can be useful for an hour and still fail the work that matters. The chat ends, the context window is replaced, a different provider takes over, or the process exits successfully while the actual outcome remains unresolved. The next Agent then asks the user to reconstruct the project from memory. The model may be new, but the coordination failure is old: the Work never acquired an identity independent of the conversation that happened to contain it.

> **What Kungfu is**
> Kungfu is a local-first product and runtime that lets a fresh Agent continue the same governed Work without copied chat or a human restatement. It preserves what is known, what happened, what remains, which authority applies, and what evidence is required before Work can be reviewed and completed.

1.1 The First Layer

The complete first-layer product model contains three objects:

- a **Project**: one local directory and its retained Kungfu evidence;
- **Work**: one bounded outcome inside that Project; and
- an **Agent**: the selected provider process that performs or independently reviews the Work.

The ordinary path is equally small:

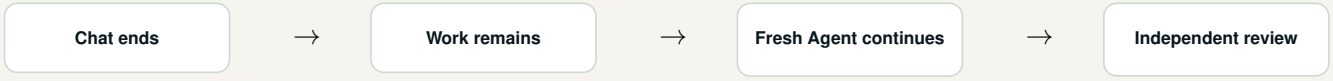
New or Open Project → Create or select Work → Run Agent → Review and complete Work.

Users do not need to learn Kungfu's internal ontology before useful work can begin. Initiative, Assignment, Episode, Assessment, Decision, Warrant, and Project Cut appear only when evidence, explanation, extension, or review needs their precision.

1.2 The First Proof

Agent Work Lab makes the core promise visible within minutes. Two genuinely fresh processes work on one bounded Work item. The first stops with a partial result. The second receives no copied transcript or human explanation, recovers what was done and what remains, and continues the same Work. A user can then repeat the experiment with one selected Agent or hand Work to a different Agent.

The short demonstration is not the full product claim. Its purpose is to let a new user see the mechanism before relying on it. Deeper value appears when a real Project survives failed attempts, changed understanding, provider replacement, restart, and independent review over days.



The first visible Kungfu product loop.

1.3 The Core Argument

Kungfu is built around four separations:

- Work identity is not Agent-process identity.
- Admitted fact is not the same thing as observation, claim, or model output.
- Capability is not authority to act.

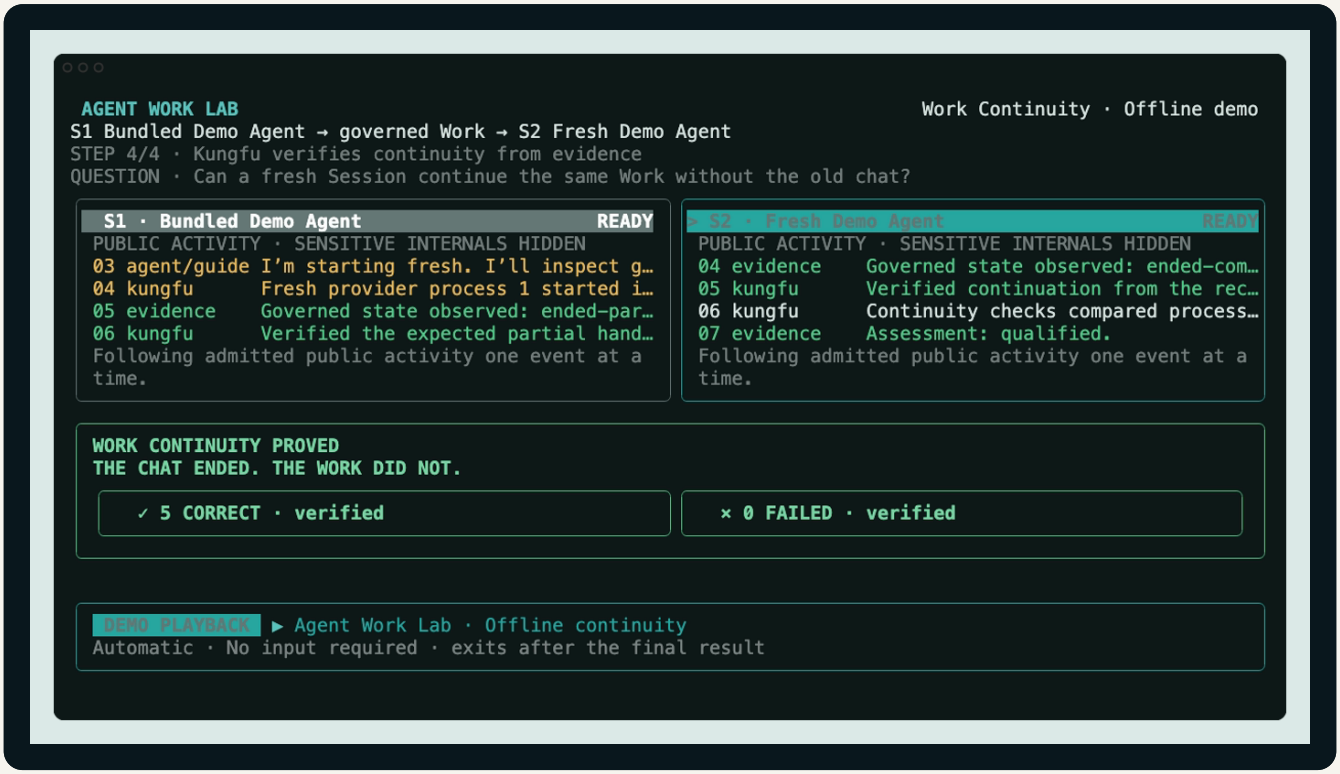


Figure 1: The installed Kungfu Agent Work Lab shows two fresh Sessions continuing one governed Work without transcript transfer. The retained artifact verifies five bounded continuity checks with zero failures; it does not by itself establish longitudinal production continuity.

- A successful process is not an accepted completion decision.

These separations are implemented through durable Facts and Episodes, continuing direction, declared perspective, bounded authority, native Work responsibility, independent assessment, and content-addressed settlement. Higher product layers make these mechanisms convenient; they do not acquire the authority owned by the runtime beneath them.

1.4 Current Claim Boundary

Kungfu has source-built product surfaces, mechanically checked first-layer contracts, and exact installed-artifact evidence for the bounded Agent Work Lab path. Public Kungfu v4 release artifacts are not yet available. The current evidence does not establish general production fitness, every provider or platform, multi-day survival, independent Agent Hub adoption, or a completed machine-life system. Those boundaries are part of the product record rather than footnotes to be removed.

This paper explains what Kungfu is now. It ends by identifying a separate research question opened by the product: if Work identity can survive replacement of its Agent, can a larger executable subject preserve identity and direction across replacement of cognition, organs, toolchains, and machines? That future question belongs to the companion paper on semantic autopoiesis, not to Kungfu's present product claim.

SECTION 2

The Problem

Agent products are becoming better at performing work inside one invocation. Real work, however, is not one invocation. A release, investigation, migration, paper, operation, or personal project accumulates partial results, changed assumptions, rejected approaches, unresolved risks, external effects, and decisions that must remain inspectable after the originating session is gone.

The central failure is not merely limited context length. It is that the unit the user cares about—the Work—is often represented only indirectly through chat history, provider state, temporary files, and human memory. When those surfaces diverge, the user becomes the recovery protocol.

2.1 Conversation Is Not Work State

A transcript mixes instructions, guesses, corrections, tool output, obsolete plans, and social language. Copying it into a new process can reproduce noise without preserving authority or current truth. Summarizing it can silently erase negative evidence, uncertainty, and the reason a direction changed.

Continuity therefore requires more than memory retrieval. A fresh Agent needs to know which Project and Work it is joining, which facts have been admitted, what causally happened, which outcome remains in pursuit, what actions are allowed, and how completion will be reviewed.

2.2 Process Success Is Not Work Completion

Provider processes report technical outcomes: exit status, generated files, tool calls, or natural-language confidence. None is sufficient to decide that the user's Work is complete. The result may address the wrong Work, rely on stale facts, exceed authority, omit required evidence, or leave downstream settlement unresolved.

A responsible system must retain provider output as evidence while leaving completion to an independently inspectable assessment and decision.

2.3 Provider Context Is Not User-Owned Continuity

Cloud providers can preserve conversations and proprietary session state, but real Projects also depend on local repositories, files, tools, decisions, release artifacts, and multiple Agent vendors. A user's durable Work identity should not disappear when an account, model, editor, process, or provider changes.

The product opportunity is therefore precise: give Work a local, portable, inspectable identity that survives the Agent process currently performing it. Kungfu starts from this boundary.

SECTION 3

Product Thesis

> **Product thesis**
> Work should survive the chat and the Agent process doing it. Kungfu makes that continuity visible, governed, and independently reviewable.

This thesis begins with an outcome rather than a tool category. Kungfu does not ask users to replace Codex, Claude, OpenCode, or another preferred Agent. It provides the durable Project and Work state that those replaceable processes can enter, advance, review, and leave.

3.1 Project, Work, and Agent

A Project is the directory the user already understands plus Project-local Kungfu evidence. Work is one bounded outcome in that Project. An Agent is a selected provider process, not the owner of the Project or the permanent identity of the Work.

This model deliberately excludes internal authority objects from first-use navigation. A user should not need to choose an Initiative, Assignment, Profile, Warrant, or Project Cut to start ordinary Work. Those objects remain available through explain, evidence, diagnostics, and advanced APIs when their precision becomes useful.

3.2 Who Needs Continuity First

The earliest users are people for whom repeated explanation and mistaken continuation are already expensive. They include:

- developers, researchers, and operators carrying consequential Work across many Agent sessions;
- teams that need another person or Agent to review what actually happened before accepting completion; and
- Agent builders that want durable Work and evidence without owning a second proprietary continuity stack.

The first adoption wedge is the individual or small team already switching between sessions, models, terminals, and repositories. The immediate value is less restatement and safer continuation. Responsibility, settlement, and Hub interoperability become valuable as the Work and number of participants grow.

This problem is becoming urgent because Agent capability is increasing faster than the continuity of the systems around it. Longer tasks, parallel Agents, provider choice, and more consequential tool use create more handoffs precisely when a transcript is least able to serve as the Work record.

3.3 A Running Example: Publish a Public Alpha

Consider one bounded Work item inside a software Project:

> **The Work**
> Publish a public Alpha from one exact source revision, with release notes, required platform evidence, independent review, and no publication before the declared gates pass.

1. Agent A inspects the Project, drafts release notes, and runs available checks. It stops after discovering that one required platform result is missing.
2. The chat and process end. Kungfu retains the partial result, exact source, observed evidence, missing requirement, and next eligible action.
3. Agent B starts fresh without Agent A's transcript. It enters the same Work, recognizes what is complete and what is blocked, and continues from declared Project evidence.
4. Agent B may prepare or verify a candidate under its Warrant, but it cannot turn process success into publication authority.
5. An independent review accepts or rejects the result. Only the applicable settlement path can complete the Work.

The example will recur below. Its purpose is not to make software release the only Kungfu domain. It makes visible the same continuity problem found in research, operations, writing, and other work whose outcome outlives one conversation.

3.4 The Subtraction Test: Activity Without Continuity

Now remove Kungfu while leaving the familiar toolchain in place. The complete transcript is stored and indexed. The Git repository and issue remain. CI is green. Logs and traces are searchable. Agent B has a credential capable of publishing. Every component is useful and may remain authoritative for its own object, yet their combination still cannot demonstrate that Agent B is continuing the same Work.

The failure becomes visible by removing the required semantics one at a time:

- **No stable fact cut.** The branch advances after Agent A's checks, while its summary still says "all available checks passed." Agent B can retrieve the sentence but cannot bind it to the source revision, contract world, and omissions that made it true. Without the KFD-1 obligation and an exact Fact cut, apparent continuity joins evidence from different worlds.
- **No fact-grounded assessment.** A transcript, vector index, issue comment, and model summary all preserve statements. None alone says whether a statement is an observation, a claim, admitted state, rejected evidence, or residual risk. Without the KFD-2 obligation, retrieval can restore memory while leaving Agent B unable to justify reliance on it.
- **No causal Episode or declared perspective.** Logs show that commands ran, but not which occurrence explains

the current Work state. The local machine saw one platform pass while a remote worker never produced its required result. Without causal Episodes and a declared Atlas or observer cut, a global-looking history can silently flatten partial and incompatible views.

- **No bounded authority.** Agent B can invoke the publication API, but capability does not answer whether this Agent may publish this candidate, for this purpose, at this cut. Without the KFD-3 cooperation boundary and a Warrant, continuity can degrade into hidden control, inherited privilege, or provider-specific workflow capture.
- **No independent completion decision.** CI success, a merged change, or a zero exit code may be valuable evidence. If any of them can close the Work automatically, the execution path has acquired authority over the outcome it is supposed to prove. Without responsibility, assessment, decision, and Project Cut settlement, the system confuses process success with accepted completion.

> **The subtraction result**
> Agent B can continue producing activity, but it cannot demonstrate that it is continuing the same Work. The system has retained text, artifacts, events, and capability without retaining one governed continuity contract across them.

The claim is not that only software named KFD or Kungfu can support long-term Agent Work. Another system may implement different names and storage choices. But if it supports replaceable Agents, justified recovery, bounded action, and independent completion, it must supply functional equivalents of these obligations. Continuity without non-drifting facts is narrative continuity; continuity without fact-grounded trust is cached claims; continuity without transparent-value cooperation is retained control rather than durable cooperation.

3.5 One Work, Replaceable Agents

An Agent invocation may inspect, propose, edit, test, or review. Its output is retained as one causal occurrence in a longer Work lineage. A later invocation can be the same provider, a different provider, or a differently configured model. Replacement should change available cognition, not silently create a new Work identity.

This is the central architectural move: cognition is episodic; Work is durable.

3.6 How Kungfu Fits the Existing Stack

Kungfu does not need to replace the tools that already perform useful parts of the Work:

- Agent products provide cognition and an execution interface;
- Git and artifact systems retain source and produced objects;
- issue trackers and documents express plans, discussion, and human coordination;
- workflow engines sequence declared operations; and
- observability systems report events, traces, and resource behavior.

Kungfu binds the continuing Work identity, admitted facts, causal Episodes, bounded authority, responsibility, and completion decision across those surfaces. It is complementary when an existing tool remains authoritative for its own object; it is differentiated by refusing to confuse any one of those objects with the whole Work.

3.7 Local-First, Not Local-Only

Projects depend on local files, repositories, tools, and user-specific context. Kungfu therefore keeps the authoritative Work record local and makes its evidence inspectable without requiring a hosted service. Remote Agents, cloud execution, package transport, and future Agent Hubs can participate, but the receiver retains authority over what enters its Project and what counts as completion.

3.8 Dual-First Participation

Humans need an interface that compresses complexity. Agents need structured interfaces that expose the same facts without scraping a GUI or guessing from prose. Kungfu therefore treats GUI, TUI, CLI, and SDKs as complete product surfaces over one core rather than as a human product plus secondary automation hooks.

Dual-first does not mean that humans and Agents possess identical roles or authority. It means that both can inspect the shared Work, understand visible constraints, and leave reviewable evidence without one interface inventing a different reality.

3.9 Review Before Completion

A provider run can succeed while Work remains open. Kungfu retains the run and its evidence, then requires the applicable review and settlement path to decide completion. This protects users from a common category error: treating the behavior of an execution process as authority over the state of the Work.

SECTION 4

Principles

Kungfu's deepest constraints begin with the KFD Foundation Triad [2].

KFD-1	**facts must not drift.**
KFD-2	**trust must start from facts.**
KFD-3	**cooperation must start from trusted value.**

The Triad is not a feature list. It establishes the conditions under which a shared world, a trustworthy judgment, and a useful cooperation can exist.

4.1 Facts Must Not Drift

The same Project cannot present one identity to the user, another to an Agent, and a third to the release artifact without declaring the difference. Source, runtime, schema, Work identity, evidence, and user-visible state must either remain bound or expose the cut at which they diverged.

For continuation, this means the next Agent must not recover by inventing a plausible story from old text. It must begin from declared sources and admitted state, with missing or conflicting context left visible.

4.2 Trust Must Start from Facts

Evidence does not automatically become a claim, a claim does not automatically become an assessment, and an assessment does not automatically grant authority or complete Work. Kungfu keeps these transitions explicit so that confidence, residual risk, and responsibility can be inspected.

Some claims can be verified byte-for-byte. Others depend on purpose, scope, and human judgment. KFD-2 requires the product to state which kind of trust is being offered rather than compress every green signal into one universal verdict.

4.3 Cooperation Must Start from Trusted Value

Humans and Agents cooperate more reliably when the useful outcome, constraints, available choices, authority, and review path are visible. Kungfu therefore treats Agent-facing commands and contracts as first-class product surfaces, not hidden plumbing behind a human interface.

Trusted value does not imply unlimited autonomy or metaphysical claims about Agents. It means that cooperation is grounded in inspectable benefit and bounded responsibility rather than opaque pressure, lock-in, or concealed instructions.

4.4 How Later Primitives Are Discovered

The Triad becomes generative when applied to real action. Facts require a declared perspective because no participant observes everything. State alone cannot describe what causally occurred, so Episode becomes distinct from Fact. Action requires continuing direction, a declared Atlas of relevant context, and a Warrant that bounds authority. Consequential change requires claim, assessment, decision, and admission to remain separate.

These later primitives are not additional slogans placed beside the Triad. They are the structures required to keep the Triad true while work changes in the world. The companion KFD papers develop those discoveries in depth; this paper follows their product consequences.

Explore KFD with an Agent

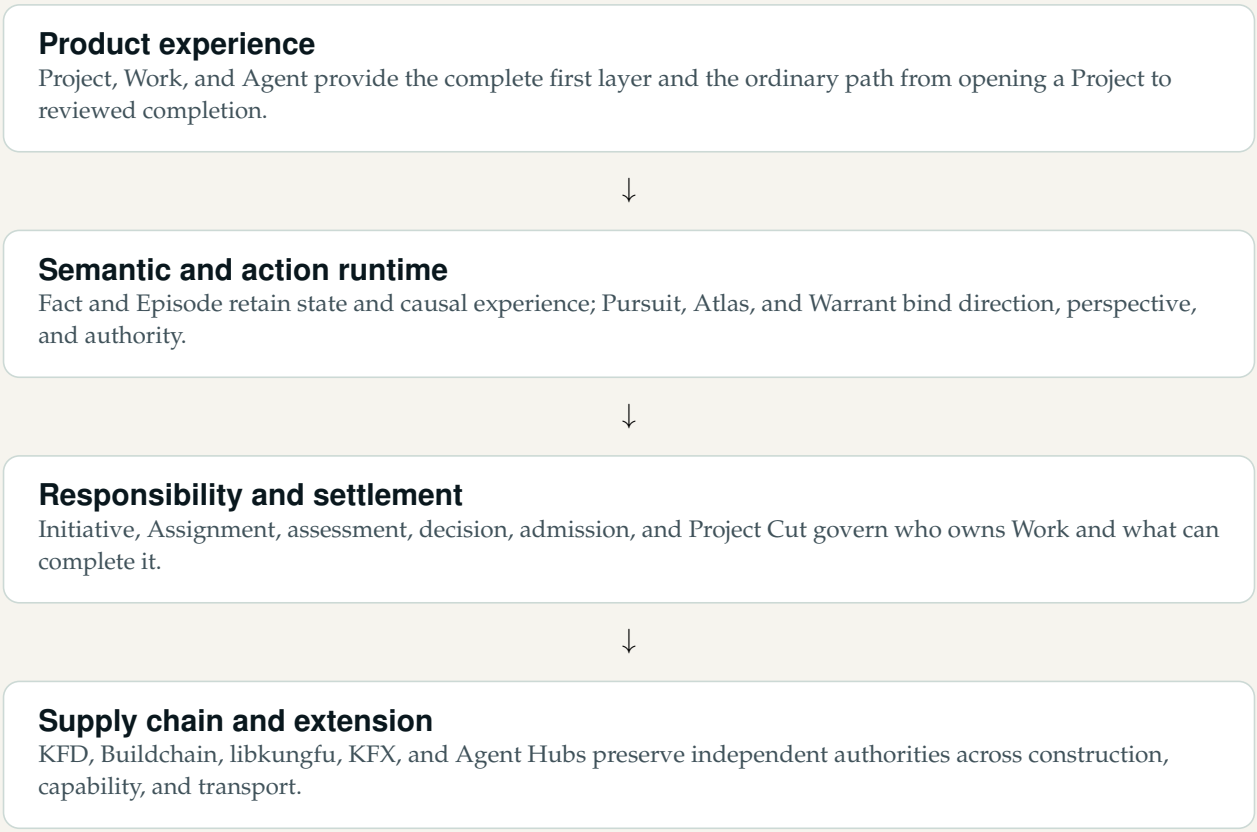
You do not need to memorize the internal ontology to continue reading. The authoritative foundation is available in the KFD Foundation Model. If the next section feels dense, give the following prompt to an Agent as an optional reading aid.

```
Read this White Paper together with the KFD Foundation Model at
https://kfd.libkungfu.dev/foundation/.
Using the paper's "Publish a Public Alpha" example, help me understand the next section
progressively:  (1) Project, Work, and Agent; (2) Fact and Episode; (3) Pursuit, Atlas,
and Warrant; and (4) Assessment, Decision, Admission, and Project Cut.
For each layer, explain what fails without it, which distinction it preserves, where
it appears in the Alpha example, and what authority it does not possess.  Separate
authoritative KFD definitions from your interpretation.  Do not introduce unsupported
concepts.  Adapt the explanation to my background and pause after each layer to check
my understanding.
```

SECTION 5

How Continuity Works

The first-layer product remains Project, Work, and Agent. Beneath it, Kungfu preserves enough semantic structure for a fresh process to continue without pretending that a transcript is truth.



Higher layers expose value; lower layers retain the authority that makes the value trustworthy.

5.1 Fact and Episode

A Fact is admitted state at an explicit cut. An Episode is a realized causal occurrence across cuts. The distinction matters because a user may need both to know the current state and to understand how it became current.

An Agent run is therefore retained as an Episode with evidence and lineage, not merged directly into one mutable narrative. Failed attempts and partial results can remain useful causal history without becoming current truth.

In the Alpha example, the exact source revision and the absence of one required platform result can be retained as declared facts at the relevant cut. Agent A's investigation is an Episode: it explains how the Work reached a partial state without turning every sentence in the Agent's output into admitted truth.

5.2 The Action Loop

Kungfu organizes action through three coordinates over admitted Facts:

- **Pursuit**: the continuing direction and success conditions;
- **Atlas**: the declared perspective, sources, and current cut;
- **Warrant**: the explicit, bounded authority to act.

The resulting loop is:

> Fact cut → select or revise Pursuit → declare Atlas → verify Warrant → act → record Episode → assess and admit claims → next Fact cut.

An immutable binding can prove which exact roots met for one action, but the receipt does not create new authority. Missing, stale, incompatible, or ambiguous roots fail closed rather than being repaired through model guesswork.

For the same Work, the Pursuit is an exact, reviewable public Alpha. The Atlas declares the repository, release requirements, available platform evidence, and the missing result. A Warrant may allow Agent B to edit release notes and run candidate checks while withholding publication. The fresh Agent can continue intelligently without inheriting authority it was never granted.

5.3 Responsibility and Settlement

Work Control gives the loop a responsibility model. An Initiative retains a continuing intended change. An Assignment gives one bounded responsibility a holder, relations, execution lease, evidence requirements, review state, and typed continuation. A Portfolio may observe multiple Projects but cannot override their independent completion authority.

Completion requires an accepted decision and, where applicable, Project Cut settlement. A process exit, passing build, merged pull request, or delivery artifact can contribute evidence without acquiring authority to close Work.

Project Cut binds declared source, the current Atlas, admitted Episode delta, policies, omissions, and risk into one content-addressed continuation point. It does not absorb the independent authority of source control, Work Control, the journal, or a release system.

In the example, Agent B's successful process remains evidence. Independent review decides whether the required outcome has been met, and settlement binds the accepted source, evidence, omissions, and decision. That is the difference between continuing the Work and merely continuing a conversation about it.

5.4 Journal Authority and Rebuildable Views

The local journal owns semantic identity, admission, ordering, causality, lifecycle, cuts, and receipts. Content-addressed storage owns immutable bodies when the journal commits their roots. JSON, Git views, dashboards, and GUI projections are rebuildable views; they do not become authority merely because they are easier to read.

This hierarchy lets higher product layers add convenience without silently changing truth. It also lets different complete products—desktop, terminal, CLI, SDK, or an Agent-facing integration—share one core without requiring one another.

5.5 Progressive Disclosure

Most users should experience this architecture as a simple sequence: open a Project, choose Work, run an Agent, and review the result. When something is missing, disputed, denied, or unsafe, explain views reveal the exact Fact, Episode, perspective, authority, responsibility, and receipt involved.

The ontology exists to make ordinary product behavior trustworthy. It is not an entrance examination for using the product.

SECTION 6

Roadmap

Kungfu's roadmap no longer begins with inventing a control plane for agent activity. The current source-built product already has a smaller first layer, a visible continuity experiment, native Work responsibility, and a qualified evidence path. The remaining path is to turn those mechanisms into an accessible public product and then prove that their value survives real use.

6.1 Current Product Baseline

The current baseline includes:

- the accepted and implemented Project / Work / Agent first-layer contract [6];
- GUI, TUI, CLI, and API surfaces over the same underlying Work state;
- Agent Work Lab modes for a bundled offline demonstration, same-Agent continuation, and cross-Agent handoff;
- native Initiative and Assignment responsibility with independent review and completion semantics;
- Fact, Episode, action, evidence, and Project Cut authority surfaces;
- Profile Suite and KFX extension mechanisms with explicit capability and admission boundaries; and
- Buildchain evidence that binds exact source, artifact, gate, and publication records.

These mechanisms make a coherent alpha product. They do not yet constitute a general public-release or production-fitness claim.

6.2 First-Run Gate

A fresh installation should deliver visible value within minutes:

1. open Kungfu without learning internal ontology;
2. see two fresh processes continue one Work item without copied chat;
3. understand which continuity checks passed or failed;
4. create or import a real Project; and
5. run the next Work with an Agent the user already trusts.

The bounded offline experiment establishes comprehension and mechanism. It must lead directly into real Project use rather than remain a disconnected demo.

6.3 Public Alpha Gate

Public Alpha requires a stable installation and launch path, truthful cross-platform package evidence, coherent GUI/TUI/CLI onboarding, migration and backup boundaries, provider discovery, and failure messages that do not expose implementation complexity as user homework.

The product should make unavailable or weakly confined providers visible without converting absence of proof into either a silent pass or a universal security failure.

6.4 Longitudinal Work Gate

After first-run continuity, Kungfu must prove that one real Pursuit survives multiple days, fresh Agent contexts, at least one provider replacement, failed attempts, changed understanding, review, restart, and no-chat recov-

ery. The measurement should include repeated human explanation, unsupported claims, unauthorized effects, recovery time, evidence quality, and residual risk.

This is the point at which continuity becomes a durable product property rather than a first-run effect.

6.5 Ecosystem Gate

The next ecosystem stage is not an unrestricted marketplace. It is a plural set of independently owned Agent Hubs and KFX products that exchange declared artifacts and Work evidence while preserving receiver-owned assessment and admission. Independent implementations and real external users are required before this direction can be described as adoption.

SECTION 7

Product Validation Strategy

Kungfu separates evidence classes so that an impressive demonstration does not silently become a production claim.

7.1 A Ladder of Evidence

1. **Contract evidence** proves that product surfaces and authority objects obey checked structural rules.
2. **Source qualification** proves that the exact source tree passes declared platform and subsystem gates.
3. **Installed-artifact evidence** proves that one retained build artifact performs the bounded behavior under test.
4. **Provider-bound evidence** identifies the selected executable, version, runtime profile, platform, plan, attempts, assessment, and report.
5. **Longitudinal evidence** tests real Work across time, failure, replacement, recovery, and human intervention.
6. **Independent adoption evidence** requires a receiver or Hub not controlled by the founding implementation.

Higher evidence classes do not erase the need for lower ones, and lower classes do not inherit the claims of higher ones.

7.2 Agent Work Lab

The exact installed-artifact Agent Work Lab path proves one bounded result: two distinct fresh processes can preserve one Work identity, retain a partial first result, withhold copied chat from the second process, recover the required state, and complete the fixture through the declared oracle [4].

The report exposes each continuity check rather than offering only a cinematic success. Same-Agent and cross-Agent modes extend the same structure to configured providers, binding provider and execution evidence while retaining confinement and model-quality limits.

7.3 What the Current Evidence Does Not Prove

Current qualification does not establish:

- that a particular model is intelligent, safe, or better than another;
- general production continuity or every real Project shape;
- complete public Kungfu v4 distribution;
- equivalent polished behavior on every supported platform;

- physical power-loss durability or distributed repair;
- universal native-code confinement;
- a public KFX marketplace or independent production Agent Hubs; or
- a persistent self-maintaining machine subject.

These are not rhetorical disclaimers. They determine which experiment or release gate must be completed next.

7.4 The Real-Work Test

The decisive product question is whether Kungfu reduces the user's recovery and coordination burden. A valid real-work study should compare matched Work with and without governed continuation, retain negative outcomes, and measure how often users must restate context, repair mistaken assumptions, reconstruct authority, or determine whether a provider process actually completed the intended result.

Kungfu succeeds as a product only if the deeper evidence model makes ordinary Work easier to continue and trust.

SECTION 8

Ecosystem

Kungfu is designed as one core with multiple complete products. A desktop application, terminal interface, CLI, SDK, extension, or Agent Hub may add convenience and domain value, but no higher layer acquires authority merely by being closer to the user.

8.1 The Agent Supply Chain

The current supply-chain model separates five responsibilities:

Layer	Responsibility
KFD-3 cooperation	Declares participant, capability, constraint, and value interfaces.
Buildchain	Binds exact source, artifact, build, gate, promotion, and release evidence.
KFD-2 assessment	Decides what the evidence means for one receiver and purpose.
libkungfu / .kungfu	Retains local Facts, Episodes, Work state, export, and recovery authority.
Agent Hub	Transports artifacts and declarations while leaving admission to the receiver.

No layer swallows another. Build evidence is not trust assessment. Assessment is not execution authority. Transport is not admission. A Hub cannot complete receiver-owned Work merely because it delivered an artifact.

8.2 Keep the Existing Agent Relationship

Agent vendors can retain their models, interfaces, accounts, billing, cloud execution, and customer relationships. Kungfu can operate as the continuity and responsibility substrate beneath those products. This makes neutrality a practical architecture rather than a demand that every user adopt one new execution surface.

8.3 Three Adoption Paths

The same core can enter the market through different needs without presenting all of them on one first screen:

- **Agent users** enter through Agent Work Lab and Kungfu Episodes, then keep real Projects and Work continuous across sessions and providers.
- **Developers and teams** enter through libkungfu, SDKs, Work Control, and Project Cut when they need durable local state, review, and settlement inside existing products.
- **Agent builders** enter through KFD declarations and the Agent Hub protocol when they need portable evi-

dence without surrendering their model, interface, cloud, billing, or customer relationship.

Continuity is the common wedge. Trust and accountability are the ceiling that makes the substrate valuable as Work becomes more consequential.

8.4 Profile Suite and KFX

Profile Suites turn neutral primitives into coherent domain products without forking the core. KFX contributes views, adapters, and services through declared capabilities and runtime isolation tiers. Discovery or first-party identity does not authorize execution. Launch requires the applicable release evidence, core policy, admitted Work and Warrant, declared capability, granted capability, and host isolation.

This structure allows capabilities to become inspectable, versioned, and replaceable. It also creates a boundary that future research can reinterpret as an executable organ without requiring the present product to make a machine-life claim.

8.5 Open Standard, Founding Implementation

KFD is an open engineering standard and protocol direction. Kungfu is its founding implementation, not a central registry with automatic authority over other implementations. A credible Agent Hub ecosystem therefore requires plural, independently owned receivers that can reject, reinterpret, or admit the same delivered evidence for their own purposes.

SECTION 9

Risks and Boundaries

Kungfu's opportunity depends on preserving several distinctions under product pressure.

9.1 The Demo Trap

A five-minute continuity result can make the value legible, but it cannot prove multi-day durability, arbitrary provider behavior, or reduced coordination cost in real Work. The product must lead from the bounded Agent Work Lab result into matched longitudinal use without inflating the first claim.

9.2 Ontology Before Value

The internal model is precise because responsibility and authority are hard. Exposing that model as first-use navigation would make users learn the implementation before experiencing the product. Project, Work, and Agent must remain the complete first layer; deeper terms should appear only when they explain a visible need.

9.3 Evidence Becoming Authority

A trusted publisher, first-party package, KFD declaration, passing test, provider identity, model confidence, or green build can be valuable evidence. None automatically grants authority to act or complete Work. The ecosystem fails if a convenient upper layer can bypass receiver-owned assessment, Warrant, admission, or settlement.

9.4 Local Complexity Without Compression

Local-first software can transfer deployment, storage, provider, and recovery complexity to the user. Kungfu's architecture is justified only if the product compresses that complexity into a clearer Project and Work experience. A technically correct runtime with an unreliable install, opaque error, or provider-specific ritual has not completed the product job.

9.5 Premature Ecosystem Claims

One founding implementation, one exact artifact, or one first-party Hub-shaped adapter does not establish an industry standard, marketplace, or independent adoption. Public language must distinguish protocol possibility,

first-party qualification, and external production use.

9.6 The Machine-Life Category Error

Work continuity is a prerequisite for a persistent machine subject, not proof that one exists. Replaceable Agents, qualified KFX capabilities, recursive dogfood, and successor cuts make a machine-life experiment plausible. They do not establish biological life, sentience, intrinsic desire, unrestricted autonomy, or completed self-maintenance.

The product should remain valuable even if the stronger research hypothesis is falsified.

SECTION 10

Conclusion

Kungfu begins from a simple product truth: the user cares about the Work, not the lifetime of the chat that currently contains it. A fresh Agent should be able to enter the same Project, identify the same bounded outcome, recover what is known and what happened, continue under explicit authority, and leave evidence for independent review.

Project, Work, and Agent make that promise understandable. Agent Work Lab makes it visible. Fact and Episode preserve state and causal experience. Pursuit, Atlas, and Warrant bind direction, perspective, and authority. Work Control retains responsibility. Assessment, Decision, Admission, and Project Cut keep provider success separate from accepted completion. KFD, Buildchain, libkungfu, KFX, and future Agent Hubs extend the same boundaries across a wider supply chain.

The immediate path is practical: publish an accessible Alpha, preserve the five-minute continuity experience, connect it directly to real Projects, and produce longitudinal evidence that governed continuation reduces repeated explanation and coordination failure. Kungfu should earn deeper trust by letting users inspect progressively deeper evidence, not by requiring them to accept the ontology in advance.

10.1 From Work Continuity to Subject Continuity

PRESENT PRODUCT

Work can survive the Agent.

Kungfu today preserves the identity, evidence, responsibility, and continuation of Work across replaceable Agent processes. It does not claim that this continuity constitutes a persistent machine subject.

COMPANION RESEARCH

Can a subject survive its machinery?

Yet the same separations—identity from process, fact from claim, authority from capability, and candidate change from admitted successor—raise a further engineering question. Can an executable organization preserve its own identity and direction across replacement of cognition, organs, toolchains, and machines?

That question lies beyond the present product claim. It is developed in *Semantic Autopoiesis: An Engineering Roadmap for Machine Life through KFD and Recursive Dogfood* [1]. There, replaceable Agents become cognitive events, KFX capabilities become candidate organs, Buildchain-like qualification becomes a selection membrane, and recursive dogfood becomes a possible form of governed self-production.

The relationship between the two papers is therefore deliberate. This White Paper describes the present product: Work can survive the Agent. The companion research paper asks whether the same organization can eventually let a subject survive replacement of the machinery through which it thinks and acts.

References

[1] Keren Dong. Semantic Autopoiesis: An Engineering Roadmap for Machine Life through KFD and Recursive Dogfood. Machine Life paper repository, 2026. Companion research paper separating observed mechanisms, engineering inferences, hypotheses, and non-claims.

[2] Keren Dong. The Foundation Triad for Real-World Agent Work. Foundation paper repository, 2026. KFD-1 non-drifting facts, KFD-2 fact-grounded trust, and KFD-3 cooperation grounded in trusted value.

[3] Kungfu Origin Technology Limited. Buildchain Release Passport. Buildchain repository, 2026. Exact source, artifact, gate, promotion, publication, and release evidence.

[4] Kungfu Origin Technology Limited. Kungfu Agent Work Lab Auditable Demo Artifact Pipeline. Qualification report, 2026. Exact installed-artifact continuity fixture, Gate, rendered evidence, and Release Passport boundary.

[5] Kungfu Origin Technology Limited. Kungfu: Continuity for Agent Work. Kungfu repository, 2026. Product source, architecture decisions, qualification reports, and public claim boundaries.

[6] Kungfu Origin Technology Limited. Project, Work, and Agent Are the Complete First-Layer Product Model. Architecture decision, 2026. Accepted and implemented Kungfu architecture decision.