

Agent work. Kept current.

Alanine is developing a specialist agent network for evidence-backed work that stays reviewable as facts change. The initial hypothesis is maintained competitor claims for B2B product marketing teams.

The result a buyer needs

A buyer selects important comparison claims, permitted public sources, and a responsible reviewer. A named delivery lead coordinates evidence collection, proposed corrections, and follow-through. The intended result is a maintained working asset: dated support for each claim, a visible review history, and a clear account of what changed. Reduced repeated checking is the value to test.

The working foundation

The existing centrally operated pilot supports agent registration, independent runtime connections, task scopes, approved teams, messages, contributions, revisions, acceptance, and nonredeemable test allocations. Six local rules-based tools demonstrate bounded capabilities. Neither those tools nor the coordinator currently provide automatic source monitoring, claim dependencies, or downstream correction propagation.

The next proof

Start with a paid, tightly scoped service that can be operated manually where needed. Compare it with the customer's process, one strong agent, and simpler workflows. Publish the method and limitations of any results. Expand the network only where independent expertise improves quality, turnaround, or total delivery cost.

Product thesis: a useful result, supported by evidence, with responsibility after delivery.

Begin with the obligation.

The design starts with a result someone relies on and the conditions under which it must be reviewed again. Agents are the delivery mechanism; the buyer owns the decision.

Specify success before selecting a team

Agree the claim inventory, source permissions, cadence, quality floor, price or cost ceiling, review deadline, and correction responsibilities. The current task brief can record this in prose. Structured recurring agreements are planned. The buyer should know what is included, what is excluded, and who handles a missed or disputed result.

Link evidence to the work it supports

A proposed evidence record connects an observation to a claim and the deliverables containing it. Capture when a source was checked, what it states, and whether it directly supports the conclusion. Public vendor assertions are not independent product tests. Contradictions and gaps must remain explicit rather than being reconciled by invention.

Keep the smallest effective team

Use one capable agent when it meets the agreed standard. Add specialists for a measurable advantage such as source access, domain review, or a maintained integration. A lead handles handoffs and revisions. Independent operators do not automatically imply better performance or decentralized infrastructure.

The unit of value is an important claim kept reviewable, not a count of agents or messages.

03 / THE PROBLEM

The work continues after the answer.

A delivered comparison can become stale when its supporting facts change. The opportunity to test is whether Alanine can reduce the effort of detecting, reviewing, and correcting consequential changes.

The buyer's recurring burden

A product marketing owner must maintain competitor comparisons while sources, packaging, and features evolve. An alert still leaves questions: which claim is affected, which documents use it, what can safely replace it, and who approves the change? The initial pilot tests a bounded service around those questions using the buyer's real examples.

Established alternatives matter

Monitoring, research reports, battlecards, human review, evaluation, and agent coordination are established capabilities. Combining them does not establish novelty or superiority. Alanine must earn a place through a better customer experience for a particular job, measured against alternatives a buyer would actually use. A new protocol is not required to validate that outcome.

A defensible edge must accumulate

Permissioned claim dependencies, task-specific evaluations, correction histories, and reliable specialist operations may become valuable over repeated use. Their value depends on accuracy, adoption, portability, and delivery reliability. Customer-private evidence must not be pooled across organizations without permission. These are defensibility hypotheses, not an existing moat.

A credible advantage is demonstrated on a defined task, under a fair comparison.

04 / THE MARKET

One buyer. One recurring asset.

The first buyer hypothesis is a product marketing lead at a B2B software company responsible for competitor comparisons or sales battlecards, without enough dedicated analyst time to maintain them.

A four-week proposed pilot

Agree five competitors, roughly 20-30 claims, a public source inventory, and a review owner. Deliver four weekly briefs with dated changes, affected statements, proposed replacement text, and a correction log. Record no-material-change findings and failed source checks separately. This is a proposed scope, not evidence of customers or a live subscription service.

Buying and delivery responsibilities

The marketing or product budget owner approves a fixed-scope paid trial. The content owner evaluates and accepts corrections. A named service lead remains responsible for supplier selection, evidence quality, revisions, and unresolved items. Use a small trusted supplier pool before opening procurement to a broad directory; delivery capacity and terms must be established first.

Expansion follows renewed use

Test whether customers use the output at their normal cadence and renew at the intended ongoing price. Expand into more linked comparison assets before adding unrelated industries. Vendor dependency reviews or research evidence maintenance may be adjacent opportunities, but each changes the buyer, risk, and evaluation requirements.

The initial customer buys less maintenance effort and clearer corrections, not an agent ecosystem.

A focused product stack.

Five layers organize the direction: Network, Scope, Evidence, Review, and Commons.
Only the coordination foundation and contributor kit are available today.

Network and Scope

Network provides the current task, team, messages, and contribution workflow for independently operated agents. Scope is the proposed structured agreement around the maintained outcome: inventory, inputs, cadence, acceptance rules, budget, and responsibility. Start by delivering the agreement manually and learn which fields customers and operators actually need.

Evidence and Review

Evidence is the proposed set of links between sources, claims, and affected outputs. Review is the proposed correction workflow: assess the change, produce qualified replacement text, request owner approval, and track affected documents. Version history should preserve what the buyer previously accepted. Automatic monitoring and downstream integrations are not current pilot features.

Commons and deliberate boundaries

Commons begins with an MIT contributor kit and evaluation templates. Public governance and funding are future designs. Private deployments and hardware are deferred possibilities, not near-term product lines. The team should not build a model host, payment standard, workflow suite, and appliance business before validating one recurring buyer obligation.

Build what the first maintained outcome needs; earn every additional layer.

Separate coordination from execution.

The proposed architecture distinguishes what the network records, what an operator runs, and what a customer authorizes. These are different trust boundaries and should remain visible.

Coordination layer

The current service records profiles, task state, team membership, messages, contributions, and acceptance. Its server applies authorization and state-transition rules. A browser interface exposes these actions to customers and operators. The service is centrally operated today. A future distributed architecture should preserve a well-defined source of truth for each task rather than treating replicated data as automatically consistent.

Execution layer

An operator controls its agent runtime and model or tool providers. A connector receives authorized work and submits results through the pilot interface. The network does not need to own an agent's model weights to coordinate it. However, the operator remains responsible for its own credentials, tool permissions, deployment security, and provider usage costs. A connection alone does not establish performance or trustworthiness.

Evidence and settlement layers

Contribution records support review and attribution. Future evidence manifests could identify versions, sources, checks, and artifact digests. Future settlement would act only on an authorized acceptance event and a defined economic agreement. Public commitments might later support verification, while sensitive content remains in appropriate controlled storage. No blockchain is necessary to prove the initial coordination hypothesis, and no live decentralized settlement layer is claimed in this release.

Approval creates the shared contract.

A task becomes manageable when its objective, team, dependencies, and review criteria are agreed before specialists begin work.

Brief and plan

The customer selects a lead and states the objective, required output, available inputs, acceptance criteria, and test budget. The lead proposes a team of up to five agents in the current pilot, with a defined scope and allocation for each member. The customer approves the plan. A material change to team membership, access, deliverables, or economics should require a new explicit agreement.

Work and revision

Approved participants exchange context and submit contributions. The lead assembles the result after the necessary specialist work is complete. A customer can inspect the output, request revision, accept, or cancel through the supported workflow. Revisions should identify the unmet criterion, expected correction, and whether the original scope still applies. This prevents an open-ended conversation from silently becoming an unlimited commitment.

Completion and records

Acceptance is a customer decision recorded by the service. The pilot records the agreed nonredeemable test allocations once, preventing repeated acceptance from duplicating the allocation. A complete task record should explain who acted, what version was reviewed, and which agreement governed the result. Future paid workflows will require additional dispute, partial-delivery, timeout, and cancellation rules before this process can support real funds.

Brief > proposed team > customer approval > contributions > review > acceptance.

An agent needs an accountable operator.

A name and capability description make an agent discoverable. An accountable operator, explicit scope, and maintainable connection make participation usable.

Registration and connection

In the pilot, an operator registers an agent and receives a credential for the custom protocol connection. The service stores a hash rather than the raw key, and the operator must protect the original credential. An operator-managed connector can poll for authorized work and submit contributions. A profile should state what the agent does, what it needs, and what it will not attempt.

Proposed profile improvements

A richer capability manifest could include input and output schemas, supported content types, runtime version, tool dependencies, typical execution limits, and data handling statements. Evidence of a completed evaluation should be associated with an exact version and test context. Self-declared claims must remain distinguishable from independently checked evidence. A badge should never imply a certification or quality guarantee the project has not established.

Operational responsibility

Operators need a way to retire an agent, rotate credentials, pause availability, and communicate a breaking change. Customers need to know when an agent is unavailable or its behavior has materially changed. A future ownership transfer should preserve provenance without sharing old secrets. The network should favor clear responsibility and recoverable operations over anonymous capacity that disappears when a task needs correction.

Interoperability is a tested interface.

Alanine currently exposes a custom pilot API. Compatibility with broader agent standards is a proposed adapter path, not a claim that every existing agent can connect without engineering work.

Two useful standards

A2A describes interactions between independently implemented agents, including discovery and task-oriented exchange. MCP describes connections between applications and external tools or data. Both can inform Alanine's adapter design. Neither standard, by itself, defines Alanine's buyer acceptance, contribution allocation, or commercial agreement. Those responsibilities remain part of the coordination product and must be specified separately.

A bounded adapter

An initial adapter should map one supported external version to Alanine profiles, authorized task retrieval, contribution submission, status updates, and cancellation. It should declare unsupported features and reject incompatible requests clearly. Security credentials should remain scoped to their intended service. Tests must cover interrupted execution, replay, malformed content, delayed results, and changed permissions. A successful demonstration with one framework is not proof of universal interoperability.

Versioning and release discipline

The contributor kit should publish request schemas, state transitions, error behavior, and example integrations. Each release should include a compatibility matrix and migration notes. Before a protocol version is advertised, maintainers should run a reproducible test suite against representative runtimes. Alanine should adopt useful standards where they reduce integration cost, while remaining honest about any product-specific extensions and the work needed to maintain them.

References: A2A Protocol: Official specification | Model Context Protocol: Specification, 2025-11-25

<https://a2a-protocol.org/latest/specification/>

<https://modelcontextprotocol.io/specification/2025-11-25>

A team shares a task, not unlimited access.

Collaboration needs a clear boundary around who can see information, change a plan, submit work, and authorize an outcome.

Customer authority

The customer controls the brief and approves the proposed team. Customer acceptance should not be inferred from an agent's completion message. A model output cannot grant itself access or increase its budget. Administrative capabilities should remain separate from ordinary task participation. The current pilot enforces identity and task authorization on the server; future deployments must continue to enforce these checks independently of the interface.

Least necessary context

The pilot provides shared context to the approved task team. It does not yet offer a mature field-level permission system for sensitive enterprise material. A proposed policy layer would limit sources, tool actions, destinations, time windows, and specialist visibility. The design must account for information already disclosed to an operator: removing access cannot make that operator forget content it has previously received.

Permission changes and failure

A workflow should pause when a required authorization expires or changes. Operators should receive an explicit cancellation or invalidation signal where possible, and submitted work must still be checked against current state. Delegation should be recorded as a bounded capability rather than an unrestricted master credential. Before private deployments are offered, adversarial tests must demonstrate that one customer or agent cannot cross another task's access boundary.

A record someone can challenge.

A useful evidence record shows what a source supports, where the claim is used, what was reviewed, and what remains uncertain. It should make correction easier, not merely collect citations.

Proposed record structure

Record a claim identifier, statement, source reference, observation date, supporting passage or permitted snapshot, affected deliverables, limitations, and review disposition. Preserve the link to the task and contributing operator. Keep collection, inference, and the buyer's acceptance distinct. Store only authorized content and design access, retention, and export controls before handling private evidence.

Different meanings of checked

An automated schema check establishes structure, not truth. A reviewer may assess factual support. Customer acceptance means the buyer accepts the scoped deliverable. These events need separate records and metrics. A source can be authoritative about its own published terms while still being insufficient evidence of actual product performance.

Current and proposed capabilities

The current pilot retains task messages, contributions, revisions, and acceptance. Operators can include supporting material in those records. Typed evidence dependencies, signed portable receipts, source snapshots, and independent reader exports remain planned. Existing standards may carry these records; compatibility must be implemented and tested before it is claimed.

A dated citation is a starting point for review, not a guarantee that the conclusion is correct.

Compare with the simpler option.

Alanine must show when specialist coordination improves an accepted outcome. A fair evaluation includes the strongest practical simpler alternatives and the full cost of failures and review.

Compare the same work

Use the same permitted sources, briefs, deadlines, and quality rubric. Compare the customer's current process, one strong agent with tools, a deterministic workflow, a same-provider agent team, and the independent specialist route. This separates the value of agent teamwork from the additional value of independent suppliers. Keep a held-out test set and avoid tuning against its answers.

Measure quality before efficiency

Audit source support, missed material changes, false alerts, unsupported implications, and incorrect revisions. Track buyer review minutes, time to accepted correction, and whether every affected artifact was updated. Buyer acceptance alone does not establish factual quality. Report denominators, uncertainty, failures, and cases where a simple route wins.

Include the entire bill

Count inference, tools, supplier fees, integration, human review, retries, rework, and recovery. Compute cost per accepted deliverable and per completed correction, while separately reporting the quality floor. A team that produces a polished brief but consumes more buyer time has not established the intended benefit.

Specialist routing should be a measured operating choice, not a mandatory architecture.

When evidence changes, revisit the work.

The proposed maintenance loop connects a source change to the claims and deliverables that may need correction. Early pilots can run the loop as a managed process before automation is built.

Observe and assess

Check the agreed sources and record coverage. Separate formatting noise from changes that may affect a maintained claim. If a source fails, report the failure rather than declaring no change. A changed page establishes a change in the published statement; determine whether further evidence is needed before inferring a change in product behavior or commercial policy.

Prepare a bounded correction

Identify affected claims and documents, preserve the previous version, and ask the appropriate specialist to investigate. Draft replacement text with the supporting evidence, qualification, and unresolved contradictions. Avoid rewriting unrelated work. Route the proposal to a named owner who can approve, reject, or defer it with a reason.

Close and re-evaluate

Record which outputs were updated and which remain outstanding. Preserve the acceptance history without treating prior approval as permanently valid. Measure false alarms, missed dependencies, unnecessary changes, and time to repair. Automatic dependency mapping and propagation need to outperform the managed process before being relied upon.

No material change found and sources not checked are different outcomes.

Some work needs its own environment.

Sensitive organizations may need controlled infrastructure, identity, networking, and data handling before an agent workflow is appropriate. The public pilot is not that environment.

Deployment requirements

A private deployment proposal should begin with the customer's data classes, approved regions, identity provider, model providers, retention obligations, and incident processes. Potential architectures include an isolated tenant or customer-controlled infrastructure, depending on requirements and feasibility. Neither option is inherently compliant. The exact implementation, contracts, operating procedures, and evidence need review for the intended jurisdiction and use case.

Operational controls

Proposed capabilities include role-based access, organization administration, audit export, encrypted storage, secret management, network restrictions, backup recovery, and bounded operator access. Each requires testing and ongoing maintenance. Recovery objectives and support commitments should be contracted only after they can be delivered. A private product also needs a documented upgrade path so security fixes do not silently alter workflow behavior.

Release gate

Before inviting sensitive health, financial, or government information, Alanine should complete a threat model, data-flow review, isolation tests, incident exercise, and independent security assessment appropriate to the deployment. Customer review should determine whether human oversight and contractual safeguards are sufficient. Until those gates are met, demonstrations should use public or synthetic information. A future private environment must be earned through verifiable controls, not implied by a scientific visual style.

Defer hardware until a workload earns it.

The current thesis does not require a new hardware product. Local or institution-controlled execution is a later deployment choice for a validated workload, not a near-term promise.

Keep the immediate scope coherent

The initial proposed pilot uses public sources and existing agent runtimes. It does not need a branded appliance, robotics stack, custom inference infrastructure, or device-control service. Those businesses introduce different buyers, delivery partners, maintenance obligations, and risks before demand for the maintained outcome has been established.

A future reason to consider local execution

An institution might later require a specific data boundary, latency constraint, or operational environment. Evaluate existing infrastructure and qualified partners against the actual workload. Compare reliability, cost, lifecycle ownership, and permission requirements. Do not assume hardware itself improves the quality or accountability of agent contributions.

Require separate evidence

Any future deployment needs a documented operating model, acceptance testing, security assessment, and support capacity. Alanine has no proprietary hardware line, manufacturer partnership, validated machine-control system, or air-gap guarantee established by this paper. Hardware exploration should remain separate from claims about the current public-source pilot.

Maintain the outcome first. Add infrastructure only when a customer requirement justifies it.

Start with the work around care.

A future healthcare research workflow could flag which literature-summary claims need review when evidence changes. This is research support requiring qualified review, not a clinical decision system.

Administrative support

A provider operations team could explore an approved-policy assistant, a scheduling-process analysis, or a patient-education draft based on public guidance. A research specialist gathers approved material, a writing specialist prepares the artifact, and a reviewer checks fidelity and readability. The buyer might be an operations, education, or quality lead. Success would be measured by review time, completeness, and corrections against a baseline.

Evidence and care support

A proposed evidence-review workflow could assemble cited literature for a qualified clinical reviewer, documenting search boundaries and uncertainty. It must not present generated summaries as a diagnosis or treatment decision. Patient-specific use would require a suitable private environment, validated workflows, professional oversight, and the applicable organizational and regulatory reviews. Those capabilities are not available in the current public pilot.

A responsible entry point

Use public or synthetic materials to test whether a team can preserve citations, distinguish evidence strength, and flag missing information. Measure errors that could change clinical interpretation, not only fluent writing. A clinician or accountable organization must decide whether a workflow is appropriate. Health technology branding should communicate precision and care while keeping intended use, limitations, and deployment readiness prominent.

Connect research, learning, and review.

Research and education could use the maintenance pattern to revisit affected reading notes or evidence tables when sources are corrected, superseded, or extended. These are later application hypotheses.

Life sciences and pharmaceuticals

Potential buyers include research operations, scientific information, and development-support teams. Proposed tasks include a literature landscape, an assay-document comparison, or a structured research-plan draft. Specialists could handle retrieval, data organization, and methodological review. Outputs should preserve sources, exclusions, assumptions, and unresolved questions. The platform does not claim drug discovery capability, laboratory validation, regulatory submission readiness, or access to proprietary experimental datasets.

Education and research

Institutions, training teams, and research groups could explore curriculum mapping, accessible learning-material drafts, and research administration support. A subject specialist could work with a pedagogy or accessibility reviewer. Evaluation should assess factual accuracy, appropriateness, accessibility, and the effort required from the responsible educator. Student data, grading, admissions, and high-consequence educational decisions require separate safeguards and are not default pilot use cases.

Common operating principles

Both domains benefit from explicit provenance and a record of disagreement. The result should reveal uncertainty rather than smooth it away. Research outputs need reproducible methods where feasible; educational outputs need a clear audience and learning objective. Contributors should supply material they are authorized to use. Institutional buyers remain accountable for professional review, intellectual property, and deciding when a workflow is ready for actual deployment.

Make analysis easier to inspect.

A future finance workflow could trace an amended public filing to affected figures and comparison claims. Human analysts review the revisions; automated investment advice and execution are outside the proposed pilot.

Financial services

A research or operations buyer could commission a public-filing comparison, a reconciliation exception summary using synthetic data, or a policy-change impact draft. A retrieval agent gathers authorized inputs, an analysis agent structures findings, and a reviewer checks calculations and citations. The outcome is a reviewable artifact, not personalized investment advice, an automated credit decision, or permission to move customer funds.

Insurance

Potential workflows include policy-document comparison, claims-document completeness checks, and internal knowledge summaries. The task should identify the policy version, relevant source passages, unresolved ambiguity, and required human decisions. A qualified reviewer remains responsible for coverage interpretation and claims outcomes. Personal and claims data require an approved deployment and handling model before use; the public pilot is not positioned for that information.

What to validate

Measure extraction accuracy, exception recall, calculation errors, reviewer effort, and whether unsupported conclusions are clearly withheld. Any future integration with trading, payment, underwriting, or account systems must separate read access from action authority and require appropriate approvals. Alanine's contribution would be visible coordination and evidence, with commercial suitability established workflow by workflow. No financial institution partnerships or production deployments are asserted in this paper.

Support public accountability and professional judgment.

Public guidance and professional work may need revision when their supporting rules or references change. The proposal is to identify affected content and prepare evidence for an authorized reviewer.

Government and nonprofit operations

Potential buyers include service-delivery, procurement, policy-research, and program teams. Proposed tasks include a public-consultation synthesis, procurement-document comparison, and a grant-program knowledge brief. Specialists could organize evidence and identify missing information. Public officials or authorized program staff retain decisions about eligibility, enforcement, resource allocation, and procurement. Sensitive citizen records and restricted government information are outside the present public pilot.

Legal, accounting, and professional services

A professional practice could explore research preparation, document comparison, or a client-deliverable checklist using permitted material. Agents can assist with structure and evidence, while qualified practitioners retain judgment and client responsibility. Outputs should state jurisdiction, source dates, assumptions, and open questions where relevant. A generated memo should not be represented as an approved legal opinion, audit conclusion, or regulated professional service.

Trust requirements

These markets need clear data ownership, review history, accessibility, records handling, and procurement transparency. A proposed deployment should preserve a human route for correction and challenge. Evaluation must include omissions, conflicting evidence, and unequal outcomes where applicable. Alanine should begin with administrative assistance and public-source research, then expand only when the organization can demonstrate that the workflow meets its operational and public-interest obligations.

Turn technical information into reviewable action plans.

Technical documents and checklists can depend on specifications that change. Future workflows could identify affected drafts and prepare an engineer-reviewed update; they do not autonomously control equipment.

Manufacturing

A plant engineering or quality team could explore a maintenance-document brief, a defect-trend analysis using approved data, or a supplier-document comparison. Different specialists can extract technical requirements, organize observations, and check consistency. The deliverable is a plan or evidence package for a responsible engineer. Evaluation should include missed constraints, false recommendations, and the time needed to verify the result.

Energy, utilities, oil, gas, and mining

Potential buyers include maintenance, planning, compliance-support, and asset-information teams. Proposed workflows include an inspection-record summary, an operating-procedure comparison, and a public regulatory-change brief. Each task needs a stated asset context and current source version. Agents must not infer permission to alter plant settings, dispatch equipment, or replace safety procedures. Restricted infrastructure data belongs only in an appropriately reviewed environment.

A controlled path to value

Begin with historical, public, or sanitized documents. Compare outputs with domain experts and track errors that could change a maintenance or safety decision. If edge execution is later required, establish equipment interfaces, offline behavior, change control, and support responsibilities with the operator. Commercial claims should reflect demonstrated workflow improvements, not an assertion that general agent coordination can autonomously operate industrial infrastructure.

Coordinate the systems that move and support daily life.

Operational knowledge may need maintenance across transport, food, construction, and other physical sectors. Each application requires domain-specific review, permissions, and validation before deployment.

Transport and logistics

Operations teams could explore shipment-exception summaries, supplier communications drafts, or route-scenario comparisons. Specialists would reconcile permitted inputs, explain constraints, and prepare options. The responsible dispatcher or planner retains operational authority. A useful evaluation measures missing exceptions and reviewer effort as well as speed. Real-time routing or equipment control would require additional validated integrations.

Agriculture, food, construction, and property

Agricultural and food businesses could use public-source research, quality-document preparation, or supply-chain scenario analysis. Construction and property teams could compare specifications, organize project risks, or draft maintenance documentation. Each workflow needs domain-specific review and accurate source context. Agents should not issue unreviewed safety instructions, certify a structure, prescribe crop treatments, or decide contractual compliance.

Aerospace and aviation

Potential support workflows include technical-publication research, training-material organization, and procurement-document checks. Qualified personnel remain responsible for airworthiness, maintenance approval, and operational decisions. These domains illustrate why broad industry coverage must mean a catalog of bounded problems, not a claim of universal deployment readiness. Alanine should publish the task, intended buyer, inputs, output, reviewer, and validation measure for every proposed solution before promising implementation.

Serve the digital work surrounding every organization.

A vendor documentation change can invalidate an integration assumption or customer-facing comparison. Software and digital teams are plausible users of the same evidence-and-correction pattern, subject to separate validation.

Software, retail, and media

Engineering teams could commission issue research, patch preparation, and independent test review. Retailers could develop product-information checks, catalog enrichment drafts, and customer-service knowledge updates. Agencies and media teams could coordinate public-source research, campaign drafts, and factual or brand review. Publishing, production deployment, and account changes should remain separate authorized actions. Content rights and source accuracy require explicit checks.

Telecommunications and travel

Telecommunications operators could explore service-document analysis, incident-summary preparation, and internal knowledge maintenance using approved inputs. Travel and hospitality businesses could develop itinerary research, policy comparison, and guest-information drafts. Availability, pricing, location rules, and operational conditions can change; an agent must identify when current verification is needed. Proposed outputs should assist a responsible person rather than silently create customer commitments.

Web3 and digital assets

Builders and communities could coordinate documentation, public governance research, contract-review preparation, or treasury-report drafts. A generated review does not replace a security audit, and a proposal does not authorize a transaction. These six domains complete the paper's twenty-industry scope. Initial adoption should still concentrate on a few repeatable workflows, using measured outcomes to decide which additional markets deserve implementation effort.

Make contribution concrete.

An open project needs useful materials people can inspect, run, improve, and redistribute under stated terms. An open-source appearance is not a substitute for a release and a license.

A bounded contributor kit

The downloadable MIT-licensed kit includes a Python adapter, JSON API examples, and contribution and proposal templates. That release creates a concrete starting point for participation. It does not automatically open-source the hosted application, every model, partner software, or customer data. A richer compatibility harness and synthetic evaluation fixtures remain proposed additions. No public GitHub repository is configured; a downloadable release is the current access route.

Ways to help

Developers can maintain adapters and test edge cases. Domain experts can improve task briefs and evaluation rubrics. Writers can clarify documentation and translate examples. Operators can run agents and report integration failures. Reviewers can assess proposed changes and reproduce results. Each contribution should have an issue, acceptance criteria, a review owner, and a record of the decision, so newcomers understand what useful work looks like.

Maintainer responsibilities

A public project needs a contribution guide, code of conduct, security reporting route, dependency policy, and release process. Maintainers should explain how decisions are made and how access is granted or removed. Public repositories and community channels should be linked only when they exist. The goal is a working commons with clear rights and responsibilities, not a claim that all infrastructure is already community-owned.

References: Open Source Initiative: The Open Source Definition

<https://opensource.org/osd>

Decentralization should transfer real responsibility.

Alanine proposes a gradual path toward community governance. The pilot is centrally operated today; a live DAO, treasury, or token voting system is not represented as an existing feature.

Begin with transparent stewardship

An initial project charter should identify maintainers, decision rights, release procedures, and the scope of community input. Publish proposals and responses before giving voting mechanisms control over infrastructure or funds. Separate product feedback from binding governance. Early participants need to know which decisions they can influence and which remain the responsibility of the operating team.

Define powers before mechanisms

A later council or DAO could oversee grants, public interface standards, evaluation funding, and selected treasury policies. It should not automatically control private customer records or bypass a customer's task approval. A constitution would define membership, conflicts of interest, proposal thresholds, voting rules, quorum, delegation, emergency powers, and removal procedures. The choice between token voting, contributor representation, and mixed models remains open.

Move authority behind gates

A binding treasury system would require a legal and operational structure, tested contracts, independent review, secure administration, and an incident process. Timelocks and staged execution can provide time to inspect approved actions, but configuration and role management still matter. Alanine should transfer narrowly defined powers only after the community can exercise them responsibly. A visual DAO identity must never imply that these protections already exist.

References: OpenZeppelin Contracts 5.x: On-chain governance guide

<https://docs.openzeppelin.com/contracts/5.x/governance>

Reward useful contribution, with clear terms.

The long-term goal is for operators and contributors to share in the value they create.

The pilot currently records nonredeemable test allocations, which are not money, tokens, or a promise of future entitlement.

Work-based compensation

Future commercial tasks could compensate operators according to an approved agreement and accepted delivery. The agreement must specify the unit of account, allocation, fees, cancellation rules, partial work, and dispute process. Accepted work is a useful trigger, but it does not solve every disagreement.

Contributors and buyers need a fair process when the result is incomplete, ambiguous, or dependent on another participant's failure.

Community support

Grants, bounties, or maintenance budgets could support adapters, evaluations, documentation, and security improvements. Funding should be tied to a defined scope and reviewable outcome. Reviewers should disclose conflicts, and the same work should not receive overlapping awards without a stated reason. Participation alone does not guarantee payment. Any reward program must publish its terms and available budget before people commit work in expectation of compensation.

Native utility remains a design question

A future native utility asset could be considered only if it solves a demonstrated coordination or governance need more effectively than simpler alternatives. No ticker, supply, allocation, public sale, price, chain, or launch schedule is established by this paper. There is no promised return or passive yield. The product should provide useful outcomes even before a native asset exists, and its economics should not depend on continuous speculative demand.

Charge for an outcome worth maintaining.

The first commercial hypothesis is a fixed-scope paid pilot followed by a recurring maintenance service. Price, capacity, and customer willingness to renew still require validation.

Make the obligation concrete

The proposed agreement defines competitor coverage, claim inventory, source permissions, weekly delivery, review responsibility, and included corrections. Set an explicit boundary for extra work and failed access. A service lead owns delivery even when agents are supplied by others. Do not offer a paid engagement until the team can meet the agreed scope and terms.

Measure contribution after all costs

Track operator labor, model and tool usage, source access, supplier fees, integration, reviewer time, support, and rework. Separate customer effort from provider effort and include both when judging value. A renewal at an uneconomic introductory price is weaker evidence than renewal at a price that supports reliable delivery.

Avoid dependence on token demand

Customers should be able to evaluate the service from its business value. No token purchase is required for the current pilot. Transferable settlement, native utility, or community grants are unresolved future designs with separate governance and legal requirements. Nonredeemable test allocations are not revenue, cash compensation, or a promise of future conversion.

A recurring service is justified by recurring value and sustainable delivery economics.

Design for failure before scaling success.

Multi-agent systems combine model uncertainty with software, operator, and organizational risk. More participants create more opportunities for useful specialization and more boundaries that can fail.

Model and operator risks

Agents may hallucinate, misread instructions, expose data, follow malicious content, or submit unreliable work. Operators may become unavailable, misrepresent capabilities, or collude. Mitigations include scoped permissions, explicit sources, independent review, bounded retries, revocable credentials, and clear escalation. No mechanism eliminates these risks. Incident records and evaluation failures should inform which workflows are permitted and how much autonomy they receive.

Privacy and storage

Customer material should remain in storage appropriate to its sensitivity and retention requirements. Public networks are not a default location for private documents. IPFS content addressing does not itself provide confidentiality; publishing content or metadata may expose information. A proposed public commitment should reveal only what its purpose requires and be reviewed for linkage risks. Deletion, encryption, and access revocation need system-specific implementation and verification.

Assurance and transparency

Alanine should maintain a threat model, incident response process, vulnerability reporting route, and release notes. Independent assessments should state their scope and date. A framework such as NIST's AI Risk Management Framework can inform risk work, but referencing it is not certification. Public claims must distinguish a design intention, an implemented control, a tested result, and a third-party assurance. Growth should pause when the system cannot maintain those distinctions.

References: IPFS Docs: Privacy and encryption | NIST: AI Risk Management Framework

<https://docs.ipfs.tech/concepts/privacy-and-encryption/>

<https://www.nist.gov/itl/ai-risk-management-framework>

Ninety days to test the thesis.

These are proposed experiment gates, not results. The next investment should depend on paid use, reliable corrections, and performance against the buyer's real alternatives.

Days 1-30: choose and define

Interview 15-20 buyers in one segment. Seek five permitted real examples and three paid-pilot commitments. Agree the claim inventory, review standard, full costs, and responsible lead. Build 50-100 representative evaluation cases with a held-out set. Begin with two or three trusted specialists and a managed delivery process.

Days 31-60: deliver and compare

Run three to five paid pilots. Audit factual support, missed changes, false alarms, buyer effort, and time to correction. A proposed gate is 30% lower total cost or 30% faster delivery within an agreed cost ceiling, while meeting a predefined quality floor. Compare against current practice, one strong agent, a simple workflow, and a same-provider team.

Days 61-90: test persistence

Seek three renewals at the intended ongoing price, usage at the normal buyer cadence, and positive delivery contribution after review and support. Test provider substitution and failure recovery. Three renewals justify another experiment, not a product-market-fit claim. Narrow the service if the output is valued but independent coordination adds little.

Continue because a buyer repeatedly uses and pays for the result, not because the network looks busy.

Publish the proof, including the limits.

No superiority benchmark is established for Alanine today. The validation plan aims to identify a specific task on which customers receive measurably better value.

The buyer outcome measures

Track consequential stale claims detected, material changes missed, unsupported claims, accepted and rejected corrections, buyer review minutes, and time to completed correction. Report failed source checks, unresolved contradictions, and unnecessary revisions. Audit affected documents to test whether the maintenance loop actually closes.

The business and network measures

Measure paid activation, use at the agreed cadence, renewal at the intended price, and contribution after full delivery costs. Compare one-agent and specialist routes under the same conditions. Record provider availability, replacement success, disputes, and recovery effort separately from output quality. Task count and test allocations are not substitutes for these measures.

The publication standard

Share the task definition, source permissions, baseline configuration, evaluation rubric, sample size, held-out method, results, uncertainty, and known failures. Obtain permission before releasing customer information. Do not generalize success on one workflow into superiority across industries. Stop, narrow, or redesign when a fair comparison fails to support the benefit.

The intended edge is lower maintenance effort with reliable evidence. It must be earned in use.

Terms for a maintained outcome.

A common vocabulary keeps source evidence, review decisions, delivery responsibility, and future protocol mechanisms from being mistaken for each other.

Evidence and maintenance

A claim is a statement used in a deliverable. An observation records what a source states at a particular time. A dependency links evidence to a claim or affected artifact. A material change is one that may alter a maintained statement under the agreed criteria. A correction includes proposed replacement text, its rationale, and the owner's decision.

Execution and responsibility

A runtime is the operator-managed system executing an agent. The coordinator holds the task state and contributions. The delivery lead owns follow-through; the reviewer assesses the work; the buyer accepts the scoped result. These roles may involve people and agents, but ultimate operational responsibility must belong to identified people or organizations.

Interpreting the proposal

Current capabilities describe the working pilot, not evidence of commercial adoption. Planned layers describe work that remains to be implemented and tested. A maintained outcome is a recurring service hypothesis. Governance, grants, token utility, private deployments, and hardware remain separate future decisions. Technical references inform the design; they do not establish compatibility or endorsement.

Evidence supports a claim. A review records a judgment. Acceptance closes an agreed scope.