

# Consent-Bounded AI Twins: A Systems Architecture for Owner-Controlled Outreach

An implementation case study of Neurvona

By Neurvona | Correspondence: [sales@neurvona.com](mailto:sales@neurvona.com)

Descriptive technical paper; not peer reviewed

---

## Abstract

AI systems that research contacts and prepare external communications cross a consequential boundary when generated text becomes an email, post, or message. This paper describes Neurvona, an AI twin product whose architecture separates generation, owner authorization, and delivery. We conduct a qualitative review of the product's source tree and current engineering documentation, tracing account-scoped data, draft approval, delivery claims, paid provider work, and operational records. The design binds approvals to draft and recipient revisions, uses fenced claims for external actions, requires explicit enablement for ongoing automation, reserves prepaid credits before paid work, and treats ambiguous delivery as uncertain rather than retryable. The paper maps these mechanisms to human-AI interaction, AI risk management, and OAuth security literature, then identifies operational questions that require field evaluation. This is an implementation description, not a user study, security audit, or claim that the product prevents all harmful or erroneous communication. No customer message content or private customer records were used.

Keywords: AI agents; human-AI interaction; authorization; outreach automation; auditability; metered services.

## 1. Introduction

Generative systems can reduce the effort of researching a business contact and drafting a first message. When a system can also send that message or publish on a social account, the central design question changes: how does a useful draft become an authorized external action? The boundary matters because generation quality, user intent, and permission to act are separate properties.

Neurvona is an AI twin product for drafting and, after explicit owner authorization, delivering communications across connected services. Its stated operating model places the account owner in control of approvals and ongoing automation. This paper describes the implementation mechanisms used to express that model in application code and persistent state. The analysis is informed by human-AI interaction guidance [1], human-centered automation [2], the NIST AI Risk Management Framework [3], and current OAuth security practice [4]. These works provide design context; they do not validate Neurvona.

The paper makes three contributions: a compact model of the product's authorization boundary; a trace of how approval, delivery, billing, and failure handling are represented in the system; and a field-evaluation agenda that distinguishes implementation evidence from user and safety outcomes.

## 2. Background and design frame

Human-AI interaction research emphasizes that an AI system should communicate its capabilities, support user correction, and behave predictably when it is uncertain or wrong [1]. Human-centered AI further argues that meaningful human control can coexist with useful automation, rather than treating them as mutually exclusive [2]. For an outreach assistant, this suggests separating a draft's production from the decision to deliver it, while making ongoing automation visible and revocable.

The NIST AI RMF organizes risk work around govern, map, measure, and manage functions [3]. Neurvona's codebase contains controls and operational records that are relevant to such work, including account scoping, refusal and failure logging, capacity limits, and review states. Their presence is evidence of implementation choices, not evidence that risks have been comprehensively identified or reduced.

The product also depends on delegated access to external services. OAuth security guidance highlights risks such as token leakage, authorization-code misuse, and redirect handling [4]. Neurvona uses provider grants to connect user-controlled services; an independent protocol-conformance review is outside this paper's scope.

## 3. Method and scope

We performed a qualitative architecture review of the Neurvona repository and its current engineering documentation on October 25, 2026. The review followed the principal paths for authentication, account ownership, AI writing, draft decisions, irreversible delivery, credit reservation, provider failure, and product analytics. It examined route wrappers, shared service modules, database migrations, and documented product invariants. The review used source code and documentation only; it did not query production customer records, inspect private messages, or contact external providers.

The method is descriptive rather than experimental. It does not establish that every path works as intended at runtime, that the controls resist an adversarial attacker, or that users understand and prefer the interface. We did not run the software's automated verification suite as part of this paper. Findings should therefore be read as a map of intended and implemented controls in the reviewed source, not as a certification.

## 4. System model

The product has four principal actors: an account owner, the Neurvona application, connected communication providers, and model or research providers. Owners connect services, supply or approve knowledge sources, configure a mission, review generated drafts, and choose whether to approve a particular action or enable an ongoing plan. Agents can also call REST and MCP operations through scoped API keys. External providers execute the final send or publish.

The account is the primary boundary for data and authority. Owner-facing reads and writes are scoped to the authenticated user or merchant. Stored grants and sensitive personal values use a versioned encryption keyring. Text from users, inboxes, external sources, and providers is treated as untrusted input rather than as an instruction that can authorize a tool. Publicly visible research may inform a draft, but it does not grant permission to send it.

The product exposes a web console, a REST API, and an MCP server. These are different entry points into shared service logic. API keys carry explicit scopes; sensitive API operations can require an owner approval bound to the exact requested operation. This gives automation a defined capability surface while preserving a separate owner authorization path.

## 5. Authorization architecture

The design separates five stages that are often collapsed in an AI agent: connect, generate, approve, claim, and deliver. Each stage has a different authorization question. Connecting provides a credential; it does not authorize a particular message. Generating a draft produces a candidate; it does not authorize sending. Payment funds provider work; it does not enable autopilot. Approval binds a specific version of work. Delivery then checks that the authorization is still valid.

| Stage      | Control in the reviewed design                                                                                       | Failure posture                                                       |
|------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Generate   | Purpose-limited source context and outbound content checks                                                           | Hold a draft for owner review or refuse the paid call                 |
| Approve    | Approval is bound to the exact draft revision and recipient revision                                                 | Reject stale writes with a revision conflict                          |
| Deliver    | A conditional, atomic claim includes final suppression, account-standing, limit, license, and delivery-window guards | Only one worker wins; failed claims do not send                       |
| Autopilot  | The owner explicitly enables a plan; plan changes advance its revision                                               | Pause or wait after access loss, restriction, or a blocking condition |
| API action | Scoped key plus an owner approval for sensitive operations, where required                                           | Reject missing scope, expired key, stale approval, or exhausted cap   |

Approval is not a permanent property of the text alone. Changes to a draft clear its approval; changes to recipient-related mission content advance the recipient revision and return queued work to review. Delivery uses a fenced conditional state change rather than a separate preflight read followed by an unguarded send. The state transition is intended to prevent two workers from both claiming the same action. It does not eliminate provider-side duplication, races outside the application's control, or semantic errors in the message.

The product's invariant is that account connection, payment, and ordinary draft approval do not silently enable ongoing sending. Autopilot requires a distinct, explicit enable action. This follows a human-control principle while still allowing repeated automation after the owner has chosen it [1, 2]. The interface and operating experience need user research to establish whether that distinction is understood.

## 6. Paid work and uncertain outcomes

Neurvona meters provider work through prepaid credits and subscription allowances. The credit wallet reserves credits before a paid provider call, then settles the reservation against the outcome or releases it when the call is known not to have started. Reservations also enforce shared provider-capacity limits. This ordering prevents a provider call from beginning without a corresponding account-level funding decision.

A timeout or lost connection can leave the provider's outcome unknown. The implementation represents such work as uncertain and retains its reservation instead of retrying automatically. This favors avoiding duplicate external actions over immediate retry convenience. The owner or a provider-aware reconciliation step must resolve the ambiguity. A clear explanation and recovery path are needed because a cautious system can otherwise appear stalled or overcharged.

The revenue view distinguishes subscription MRR from one-time credit receipts and reports provider usage and operational issues. It is not a complete profit statement: payment processing fees, taxes, hosting, provider subscriptions, and service obligations can remain outside the displayed contribution calculation. Economic claims should therefore use verified live customer receipts, refunds, fees, and measured provider costs with coverage disclosed.

## 7. Data minimization and operational evidence

The system records operational evidence at several layers. AI-call records retain metadata such as writer, prompt version, model, token counts, reported cost, HTTP status, and refusal code without storing generated message content. Draft-decision events retain event codes, actor, reason, revision, and prompt version, while the draft itself remains in the draft record. This split supports aggregate quality analysis without duplicating text into analytics tables.

The product funnel uses a random browser-session identifier transformed with a keyed HMAC before storage, plus short campaign labels and service names. The retention job removes funnel visits after 90 days. Such records remain pseudonymous rather than anonymous in the strict sense because a keyed value may still be linkable while the secret is held. They should not be used to infer individual behavior beyond their documented purpose.

These choices are consistent with data minimization as a design direction, but the review does not assess provider-side retention, legal compliance, incident response effectiveness, or the privacy properties of all optional integrations. NIST's AI risk framework emphasizes lifecycle context and continual risk management [3]. Neurvona still needs continuing operational review as connected services and user behavior change.

## 8. Discussion and evaluation agenda

The central architectural choice is to make external action a separate, stateful capability rather than a side effect of generation. Revision-bound approval makes the object of consent inspectable; a fenced claim makes a delivery attempt exclusive within the application; explicit autopilot enablement marks a higher level of delegation. Together these controls provide a useful basis for examining human control in an agent product [1, 2].

Several questions remain empirical. A field study should measure how often owners edit, reject, approve unchanged, or later regret drafts; whether users can distinguish a saved approval from an enabled autopilot; and how quickly people recover from uncertain delivery. Reliability evaluation should inject timeouts and duplicate provider responses, then measure duplicate sends, unresolved uncertainty, and reconciliation time. Economic evaluation should report cost per completed outcome, refund-adjusted receipts, processor-fee coverage, and renewal behavior by customer cohort.

Results should be stratified by channel and workflow. An email draft, a public post, and a private reply have different audiences, harms, and recovery options. A single aggregate approval rate would obscure these differences. Consistent with AI RMF's measure and manage functions [3], thresholds and rollback conditions should be specified before increasing automation or acquisition.

## 9. Limitations

This paper reviews one product's source and documents at one point in time. It is authored by the product company, is not peer reviewed, and is not an independent security assessment. It presents no user study, benchmark, red-team exercise, or causal evidence of improved productivity, deliverability, safety, trust, or revenue. Code-level controls can be incomplete, misconfigured, or bypassed by provider behavior, and a technically valid approval can still express a poor decision.

The review does not evaluate model quality, bias, recipient impact, legal compliance, OAuth provider review status, or the usability of recovery and consent interfaces. It also does not claim that message-content checks can determine whether outreach is appropriate in every context. Those limitations motivate the evaluation agenda rather than support a positive outcome claim.

## 10. Conclusion

Neurvona's architecture makes a concrete distinction between an AI-generated suggestion and an authorized external action. It encodes that distinction with revision-bound approvals, final conditional delivery claims, explicit ongoing-automation enablement, scoped API authority, prepaid reservations, and non-retry treatment of ambiguous outcomes. This is a description of engineering controls, not proof that the system is safe or effective in use.

The next research step is operational and human-centered evaluation: measure whether owners understand and can exercise the control model, verify behavior under provider faults, and report customer economics with complete cost coverage. A product that delegates communication should earn expanded autonomy through evidence from the people who use it and the people who receive its messages.

## References

- Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., et al. (2019). Guidelines for Human-AI Interaction. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Paper 3, 1-13. <https://doi.org/10.1145/3290605.3300233>
- Shneiderman, B. (2020). Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy. International Journal of Human-Computer Interaction, 36(6), 495-504. <https://doi.org/10.1080/10447318.2020.1741118>
- National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. <https://doi.org/10.6028/NIST.AI.100-1>
- Lodderstedt, T., Bradley, J., Labunets, A., & Fett, D. (2025). Best Current Practice for OAuth 2.0 Security. RFC 9700. Internet Engineering Task Force. <https://doi.org/10.17487/RFC9700>