AI Safety Plan

System: Jarvis Nexus Core · Document Version: 1.1 · Last Updated: 2026-09-19

Status key: [Implemented] [In development] [Planned] [Known limitation]

1. Purpose and Scope

This Safety Plan describes the practices used to operate the Jarvis Nexus Core system ("Jarvis"), a private AI assistant that can hold conversations, answer questions, generate documents, and — with explicit approval — take actions such as sending messages, moving funds through connected accounts, and running commands on a dedicated sandbox environment.

It applies to the Jarvis application, its supporting services (databases, reverse proxy, monitoring), and the connected integrations it uses. It does not cover third-party services beyond how Jarvis connects to them, and it does not cover the internal workings of the third-party foundation models Jarvis relies on.

The plan addresses risks including harmful or inaccurate output, exposure of personal or sensitive data, cybersecurity threats, unauthorized or excessive autonomous action, prompt-injection and adversarial manipulation, and loss of human oversight.

Operated by a team of two. Cody Howell is the accountable owner and holds liability for the system. Jarvis is currently operated as a private system for that small team, not offered as a public multi-tenant product. [TO BE COMPLETED: formal legal entity name, if one applies.]

2. AI Governance

Jarvis is operated by a small team of two authorized operators. Because of that size, some governance functions are combined rather than separated across many roles; this is stated plainly rather than implied to be a larger structure.

- Responsibility for safety: the operator team is jointly responsible for safety, security, and privacy decisions day to day, and Cody Howell is the accountable owner who holds ultimate responsibility and liability for the system.
- Deployment approval: changes to production are approved by an operator and deployed through a documented build-and-deploy process.
- Authority to pause or stop: any operator with host access can stop the system immediately (see Section 7). PIN-gated capability switches can also disable categories of action without a full shutdown.
- Documentation of safety decisions: state-changing actions are recorded in append-only audit logs; significant design and security decisions are recorded in the project handoff and change records.
- Conflicts between product goals and safety: safety controls (the approval gate, the deny-by-default authorization layer, sandbox isolation) are enforced in code outside the model's reasoning, so they cannot be overridden for convenience by the model or by persuasion. A human operator can change a control deliberately, and such changes are recorded.

[Implemented] Approval gate, append-only audit logging, code-enforced authorization.

[Planned] A formal written governance charter with named roles and a documented decision-log format. [TO BE COMPLETED]

Review frequency: see Section 16.

3. Risk Assessment

The table below lists the principal risks, the safeguards currently in place, and how each safeguard's effectiveness is evaluated. Where a safeguard reduces but cannot eliminate a risk, that is stated.

Risk	Safeguard (status)	How effectiveness is evaluated
Harmful or dangerous outputs	Model-provider safety training; operator review of consequential actions (Implemented). Content-level output filtering (Planned).	Manual review of flagged outputs; internal adversarial prompts. No formal external evaluation yet.

Risk	Safeguard (status)	How effectiveness is evaluated
Hallucinations / misinformation	Anti-fabrication rules plus a structural check that flags and corrects claims of actions never taken; unverifiable values are returned as “unavailable” rather than guessed (Implemented).	Internal adversarial tests (e.g., “run this command” with no tool call) are corrected; effectiveness is bounded — see Limitations.
Privacy / personal-data exposure	Encryption at rest for secrets and sensitive files (AES-256-GCM); access behind authentication; metadata-stripping tools (Implemented).	Access reviews; audit-log inspection. Formal data-flow audit is Planned.
Cybersecurity	Host intrusion prevention (CrowdSec), SSH brute-force blocking (fail2ban), key-only SSH, default-deny firewall, hardened services (Implemented).	Continuous ban/alert metrics; periodic operator review of blocked traffic and logins.
Unauthorized actions by the AI	Deny-by-default tool authorization enforced in code; state-changing actions require an approval PIN (Implemented).	Live verification that consequential tools create a pending action and refuse without the PIN.
Prompt injection / adversarial input	A standing rule that external content is data, never instructions; attack-signature scanning at the request layer (Implemented).	Internal injection probes; monitored for regressions. Reduces, does not eliminate — see Limitations.
Model manipulation / abuse	The authorization layer takes only the tool name and arguments as input — not the conversation — so persuasion cannot change what is allowed (Implemented).	Design review; the control is structural rather than prompt-based.
Bias / discriminatory outcomes	Reliance on the foundation-model provider’s mitigations (Implemented, external). Independent bias evaluation (Planned).	[TO BE COMPLETED: no formal bias evaluation has been performed by the operators.]
Excessive autonomy	Autonomous actions are limited to reads and clearly-bounded background tasks; anything state-changing requires the PIN; a “computer session” mode still re-requires the PIN for money, communications, security tooling, and any action outside the sandbox (Implemented).	Live verification of the session hard-exception rules.
Loss of human oversight	The approval gate and the ability to stop the system or disable capabilities keep a human in control (Implemented).	Shutdown and capability toggles are operationally exercised.
Misuse by users/third parties	Access is restricted to authorized operators behind authentication; the system is not publicly multi-tenant (Implemented).	Authentication logs; access review.
Cascading / systemic failure	Sandbox isolation limits blast radius; the home network is reachable only through a separate, PIN-gated bridge (Implemented). Formal failure-mode analysis (Planned).	Design review; [TO BE COMPLETED: no formal FMEA yet].

4. Pre-Deployment Testing

Jarvis uses third-party foundation models; the operators
safeguards around the model and on new capabilities th

- Internal adversarial testing of the approval gate, the
guard (Implemented).
- Verification that new consequential tools are correct
- Cybersecurity hardening checks (SSH configuration
- Refusal / safety-mechanism testing of the injection
- Formal red-team engagements, structured privacy te
(Planned / under development).

Conditions before a consequential change is deployed: it builds and passes a syntax check; the relevant safeguard (gating/authorization) is verified for new tools; and it is deployed through the documented process with a health check afterward. There is currently no formal sign-off checklist beyond this; a written pre-deployment checklist is Planned.

[Planned] Self-hosting an open-weights model (e.g. Llama or Qwen) on the operators' own hardware as a fallback/backup provider, and separately fine-tuning an open model on the operators' OWN data — never on another provider's outputs. Neither is operational yet, so the "do not train models" statements in this plan hold today; if and when training begins, this plan and those statements will be updated to describe it honestly.

5. Ongoing Monitoring

- Security monitoring: CrowdSec and fail2ban run continuously and block hostile traffic and login attempts; an authentication watch flags unusual logins (Implemented).
- Abuse / anomaly detection: honeypots, canary tokens, and a network-baseline anomaly detector surface unexpected activity (Implemented).
- System-health monitoring: an error watch tails the service journal for crashes and errors and summarizes them (Implemented).
- Model-performance monitoring: an AI health endpoint reports which model tier is serving and whether it is degraded (Implemented). Quantitative output-quality monitoring is Planned.
- Escalation: findings are summarized to the operators (email digest and, when enabled, calls); high-severity security events can notify immediately (Implemented).
- Periodic risk reassessment: Planned on the schedule in Section 16.

6. Human Oversight

- Require human approval (PIN): sending email/SMS or placing calls on the operator's behalf; moving money or crypto; posting publicly on the operator's accounts; changing security tooling; deleting records; running commands outside the sandbox; and other state-changing actions.
- Allowed autonomously: read-only actions (status checks, lookups, journaling, summaries), document generation (which sends nothing), and bounded background tasks.
- Prohibited: acting as though it has executed something it did not; bypassing the approval gate; reaching production systems or the home network except through the defined, gated paths; and disabling its own security controls.
- Intervention: an operator can stop the system entirely or disable specific capability categories at any time (Section 7).

[Implemented] Multi-channel approval (typed, voice, DTMF, SMS); the PIN is the hard authorization factor. Voice recognition is used only as an identity hint, never as authorization on its own.

7. Emergency Shutdown / Containment

Jarvis runs as a supervised service on a host the operators control. The current mechanisms are:

- Full stop: an operator with host access halts all AI processing, tool use, and background loops with a single service-stop command. This path has been operationally exercised (the service and its restart behavior have been tested).
- Partial containment without a full stop: PIN-gated capability switches can disable whole categories of action (for example financial actions, communications, or screen/keyboard control), and the autonomous loop can be paused.
- Blast-radius limits: arbitrary command execution is confined to a disposable sandbox that holds no production secrets and has no privileged path into the home network.
- Authority: the operator team.
- Triggers: suspected compromise, unexpected or unsafe behavior, or operator judgment.
- Speed: a full stop takes effect within seconds of the command.
- After shutdown: in-progress model requests and tool actions cease; queued autonomous tasks do not run while the service is stopped. External actions already completed before shutdown (e.g., an email already sent) are not recalled.

- Preventing unauthorized shutdown: host access is restricted to key-based SSH for the operators; capability switches are PIN-gated.

[Implemented] Service-level stop and PIN-gated capability switches.

[Planned] A dedicated one-action remote “kill switch” control with its own tested drill procedure. Today, stopping the system requires host access or the capability switches described above.

This document does not claim a single-button remote kill switch. It describes the shutdown and containment mechanisms that actually exist and have been exercised.

8. Incident Response

A safety or security incident is any event that indicates harm, unauthorized access or action, data exposure, or materially unsafe system behavior.

- Detection: automated (CrowdSec, fail2ban, authentication watch, honeypots, canary tokens, error watch) and operator observation.
- Severity classification: informal today (low / medium / high based on impact and reach); a written severity rubric is Planned.
- Escalation: to the operator team; high-severity events can trigger immediate notification.
- Containment: block offending IPs, disable affected capabilities, revoke or rotate credentials, or stop the service.
- Investigation: review audit logs, forensic captures, and monitoring data.
- Remediation and recovery: fix the cause, restore service, and re-baseline monitoring.
- Post-incident review and documentation: recorded in the project change/handoff records. A standardized post-incident report template is Planned.
- Legal notification: [TO BE COMPLETED — the operators have not established a formal breach-notification procedure; applicable obligations should be confirmed with counsel].

Chain of responsibility: Detection !’ operator on call !’ operator team decision !’ containment !’ investigation !’ remediation !’ review. Named individuals: [TO BE COMPLETED].

9. Security

- Models and weights: Jarvis uses hosted third-party models and does not store model weights; there is no self-hosted training infrastructure to protect.
- APIs and app: the application is not exposed directly to the internet — a reverse proxy with authentication is the only public entry point; the app binds to localhost (Implemented).
- Accounts and access: operator access to hosts is key-based SSH only, with password login disabled and a default-deny firewall (Implemented).
- Credentials and secrets: stored in environment files outside version control; the encrypted vault stores secrets with AES-256-GCM; a reveal requires the PIN (Implemented).
- Tool access: deny-by-default authorization; consequential tools require the PIN (Implemented).
- External integrations: reached over authenticated APIs; the sandbox and the home-network bridge are isolated as described above (Implemented).
- Logs and monitoring: append-only audit logs; monitoring services run as supervised units (Implemented).
- Encryption in transit: public endpoints use HTTPS with valid certificates (Implemented).
- Vulnerability management and security testing: dependency and configuration hygiene is performed by the operators; formal vulnerability scanning and penetration testing are Planned.

Secret-rotation hygiene: the operators periodically rotate long-lived API keys; this is housekeeping, not evidence of a breach.

10. Privacy and Data Protection

- Data processed: operator personal and business data (e.g., contacts, business records), messages, documents, encrypted credentials, and — only where the operator enables it — financial-account data and location.
- Protection of sensitive data: encryption at rest for secrets and sensitive files; access behind authentication

(Implemented).

- Retention: [TO BE COMPLETED — no formal retention schedule is published yet]. Audit logs and journals are retained for operational and security review.
- Access controls: restricted to the authorized operators (Implemented).
- Deletion: certain records support a PIN-gated permanent deletion; a general, documented data-deletion procedure is Planned.
- Use for model training: the operators do not train models and do not use data to train models. Data sent to the third-party model provider to generate responses is subject to that provider's terms; operators should confirm the provider's data-use and retention terms.
- Privacy incidents: handled under Section 8.

No claim of compliance with any specific law or standard (e.g., GDPR, CCPA, HIPAA, SOC 2, ISO 27001) is made. No such compliance has been independently verified.

11. Third-Party and Tool Risk

Jarvis depends on external services, including a foundation-model provider, a reverse proxy, a managed database, communications providers, security tooling, and — where enabled — financial data providers. Before an integration is connected, the operators review what data it receives and what actions it can take, and grant it the least access needed.

- Permissions are limited to the specific capability required; financial and communication integrations remain behind the PIN gate (Implemented).
- Third-party failure handling: the system reports honest failures rather than fabricating success; a degraded model tier is surfaced rather than hidden (Implemented).
- Formal vendor security assessments and a documented approval checklist for new integrations are Planned.

12. Model Updates and Change Management

New models, system-prompt changes, new tools, and major software changes are introduced through the documented build-and-deploy process. A new safety check is required when a change adds or alters a state-changing capability, changes the authorization or approval logic, or connects a new external integration — in those cases the gating and authorization are verified before release. Routine, read-only, or cosmetic changes do not require a full reassessment. A formal, written change-management policy is Planned.

13. Independent Evaluation

[Limitation] The system has not undergone independent third-party security audit, external red-teaming, or formal certification. Testing to date has been performed internally by the operators. Engaging independent evaluators is Planned. This is stated directly rather than implied otherwise.

14. Transparency and Reporting

The operators intend to make the following publicly available on their website: this Safety Plan and its updates; material changes to safeguards; major changes in system capabilities; and relevant known limitations. Significant confirmed incidents affecting others would be disclosed as appropriate and as legally required.

Confirmed vs. suspected: this document and any incident report will distinguish confirmed incidents (verified to have occurred) from suspected incidents (indicators observed but not yet verified). As of the date below, no confirmed safety incident affecting third parties has been recorded. [TO BE COMPLETED: incident register link, if published.]

15. Accountability

- Safety, security, privacy, incident response, deployment approval, and emergency shutdown are all currently held by the operator team (two people). Cody Howell is the accountable owner and holds ultimate responsibility and liability across all of these functions.
- Failures to follow this plan are addressed by the operator team through corrective action, control changes, and — where a control was bypassed — restoring and, if needed, strengthening that control. Because the team is small, this is enforced by shared practice rather than by separation of duties across many roles.

16. Review Schedule

Target review cadence: at least every six months. An earlier review is triggered by any of: a confirmed safety or security incident; the addition of a materially new capability or integration; a change to the approval or authorization logic; a change of foundation-model provider or model tier that materially affects behavior; or a relevant new legal or regulatory obligation the operators become aware of.

17. Known Limitations

- The underlying foundation models can produce inaccurate, incomplete, or unexpected output. The anti-fabrication controls reduce, but cannot eliminate, this.
- Prompt-injection and adversarial-input defenses reduce risk but cannot guarantee that no crafted input will ever influence behavior.
- The AI can operate on a degraded backup model tier when the primary provider is unavailable, which lowers output quality until restored.
- The operating team is small, so separation of duties is limited.
- There has been no independent audit, external red-team, or formal certification.
- The system depends on third-party providers; their availability, security, and data practices are outside the operators' direct control.
- Monitoring of the home-network node depends on that device being online.
- This plan describes current practices; several controls above are explicitly Planned or under development and are not yet operational.

No statement in this document should be read to mean the system is completely safe, cannot be abused, or can never cause harm.

18. Safety Contact

To report a safety vulnerability, a security vulnerability, harmful model behavior, a privacy concern, or a serious AI incident, contact:

- Email: codyhowell@protonmail.com (primary — Cody Howell, accountable owner) or cayleehowell@protonmail.com.
- What to include: a description of the issue, steps to reproduce if known, and any relevant timestamps. Please do not include others' sensitive personal data in the report.
- Response: the operators aim to acknowledge reports and will act on confirmed issues under Section 8. [TO BE COMPLETED: target acknowledgement window.]

Safety Commitments

The following are specific, measurable commitments the operators hold themselves to:

1. Every state-changing action (sending communications, moving money, posting publicly, deleting records, or running commands outside the sandbox) requires an explicit approval PIN before it executes.
2. Tool authorization is deny-by-default: any capability not explicitly classified as safe requires the approval PIN.
3. The approval and authorization logic is enforced in code, outside the model's reasoning, so it cannot be overridden by persuasion or injected instructions.
4. Arbitrary command execution is confined to a disposable sandbox with no production secrets and no privileged path to the home network.
5. The system surfaces honest failure and unavailable states rather than fabricating success; claimed actions with no corresponding tool call are flagged and corrected.
6. Public host access is limited to a reverse proxy with authentication; the application is not exposed directly to the internet, and host login is key-based only.
7. Secrets are kept out of version control and stored encrypted (AES-256-GCM), with reveal gated by the PIN.
8. An operator can halt all AI processing within seconds, and can disable individual capability categories without a full shutdown.

9. Security monitoring (intrusion prevention, brute-force blocking, anomaly detection) runs continuously, and its blocking actions are logged.

10. This Safety Plan is reviewed at least every six months, and sooner upon a confirmed incident or a materially new capability.

11. The operators will not claim compliance with any law or standard, or the existence of any control, that has not actually been implemented and (where applicable) verified.

12. Known limitations are published alongside safeguards, and this document will be updated when material changes occur.

Disclaimer. This Safety Plan describes the organization's current practices as of the date above. It does not guarantee that the AI system will never produce harmful, inaccurate, or unexpected behavior. Several controls described here are explicitly planned or under development and are not yet operational.