

Verification Before Velocity:

A cross-layer evaluation of four open-source Skynet products

Exzil Calanza

Independent Researcher and Software Builder, Iloilo City, Philippines

26 August 2026 | Version 1.0

Publication status. Independent research preprint. IEEE-style organization is used for readability; this work is not IEEE-published, IEEE-endorsed, accepted, certified, or peer reviewed.

Companion article. <https://exzilcalanza.info/verification-before-velocity-four-skynet-products-2026/>

Abstract

Automation products fail at different layers: protocol transport, browser adaptation, source-change analysis, and desktop distribution. Their evidence is therefore not reducible to a single score. This study evaluates four MIT-licensed Skynet products using primary repositories, focused local suites, release provenance, a progressive Remote Skynet PC run, and current protocol and security standards. The local evidence records 85 passed and one skipped test for Skynet Chrome CDP, 19 passed tests for the Firefox WebDriver BiDi adapter, and 98 passed tests for AI File Blast Radius. Skynet Desktop v0.1.3 is bound to a 99,753,337-byte unsigned installer with SHA-256 4fd815622d6fc24c37bb46a611af340b094508ab66a897d3c5a0d5abfc9cf15f. These are suite- and artifact-specific facts, not comparable product rankings. On DESKTOP-OJ9K9QR, browser ports and Firefox were unavailable, three exact public Blast Radius gate tests passed under source drift, and Desktop was not installed. Those exclusions remain negative or partial observations. Guided by DeepSeek Harness testing policy, the resulting Skynet audit harness classifies refusal, policy, rate limit, capability absence, transport failure, empty response, worker error, and timeout into bounded recovery routes without treating any exclusion as approval. The main contribution is a field-level evidence contract: verify the real entry path, preserve artifact lineage, separate unavailable from failed and passed, and compare only fields measured under equivalent conditions.

Keywords: agent harness; browser automation; Chrome DevTools Protocol; WebDriver BiDi; software dependency analysis; Electron; reproducible evaluation; progressive benchmarking

1. Research question and contribution

The study asks how four products that operate at different layers can be evaluated without turning unlike evidence into a marketing score. Skynet Chrome CDP controls an already-running Chromium session; firefox-cdp maps a compact Tab API onto WebDriver BiDi; AI File Blast Radius estimates transitive change impact before an edit; and Skynet Desktop packages signed-in command-line agents into a Windows application. A unit-test count, a browser self-test, a latency percentile, and an installer hash answer different questions.

The contribution is a cross-layer evidence contract. Each claim is typed by subject, environment, entry path, artifact, and independent signal. A missing dependency or protocol port is recorded as unavailable, not passed. A provider refusal or timeout triggers a bounded next route, not approval and not an unbounded retry. This design was durably implemented in tools/skynet_audit_harness.py after inspection of DeepSeek Harness testing policy [9][10].

2. Standards and primary-source context

Chrome DevTools Protocol defines JSON commands and events across domains used to inspect and control Chromium. Its tip-of-tree protocol changes frequently and carries no backward-compatibility guarantee [1]. The Accessibility domain exposes the full accessibility tree, roles, names, states, and ignored-node metadata, but remains experimental [2]. Those facts motivate explicit browser-version and viewport evidence rather than assuming machine portability.

WebDriver BiDi is a W3C Working Draft defining bidirectional remote control of user agents [3]. The Firefox product is therefore an ergonomic adapter, not a CDP server or proxy. It establishes a BiDi session and translates remote values while keeping Firefox process/profile ownership explicit [5].

Electron's security guidance treats remote content and powerful desktop capabilities as a trust boundary [8]. The Desktop installer is unsigned; verification therefore depends on an exact release URL, byte count, checksum, embedded-artifact lineage, and honest disclosure rather than a code-signing claim.

3. Products and verification layers

Table 1 identifies the question each product can answer. The rows are complementary: protocol reachability does not prove desktop installation, and a release checksum does not prove a live browser port. Cross-product evaluation is meaningful only when evidence is kept in its native layer.

Table 1. Products, layers, questions, and real entry paths.

Product	Layer	Primary verification question	Public entry path
Skynet Chrome CDP	Chromium protocol	Can an owned tab in a live session be read and controlled with explicit state?	standalone_benchmark.py --port 9222
firefox-cdp	Firefox WebDriver BiDi	Can an isolated Firefox session complete the BiDi launch/session/navigation path?	firefox_bidi_worker.py --selftest --json
AI File Blast Radius	Static dependency graph	What transitive impact and untested reach precede a source edit?	blast-radius <target> --gate
Skynet Desktop	Windows distribution	Does an exact release artifact preserve lineage, capabilities, and cleanup truth?	installer + SHA256SUMS + release gate

4. Local evidence

Focused local suites were run through the real repository entry paths preserved in this run. The recorded outcomes are 85 passed and one skipped test for Chrome CDP, 19 passed tests for firefox-cdp, and 98 passed tests for File Blast Radius. The skipped Chrome test is disclosed; it is not silently promoted to a pass. Test totals are evidence of the named suite at the recorded source state, not measures of product size or quality and not cross-product scores.

The Chrome repository separately reports a synthetic-fixture latency study on Chrome 151 with 200 measured samples after 20 warm-ups: 0.33 ms p50 for an evaluate round trip and larger costs for scene, DOM, accessibility-tree, and screenshot operations [4]. That first-party benchmark is reproducible from its published fixture, but its timing remains environment-specific. The companion accessibility-tree field study demonstrates why even a raw node count can change with viewport state [12].

Desktop v0.1.3 uses a different evidence type. The public release record binds the installer name, 99,753,337-byte size, SHA-256 checksum, source commit, embedded app.asar checksum, and post-uninstall cleanup evidence. The installer is explicitly unsigned. A successful hash comparison establishes byte identity with the published checksum; it does not establish publisher identity or installation success on another machine.

Table 2. Local evidence is suite- and artifact-specific; totals are not a leaderboard.

Product	This-run/local evidence	Safe claim	Not established
Chrome CDP	85 passed, 1 skipped	Named focused suite passed except disclosed skip	Universal browser compatibility or machine-independent timing
firefox-cdp	19 passed	Named protocol-contract suite passed	A current live self-test on every Firefox installation
File Blast Radius	98 passed	Named graph/risk/gate suite passed	Assertion coverage of every reachable file
Desktop v0.1.3	99,753,337 bytes; SHA-256 4fd81562...	Published installer byte identity and unsigned status	Code-signing publisher identity or Remote-PC installation

5. Progressive Remote Skynet PC evaluation

A bounded run on DESKTOP-OJ9K9QR probed available capabilities progressively instead of assuming parity with the local host. No usable Chrome CDP endpoint was found across the single bounded 127.0.0.1 scan of ports 9222 through 9230. Firefox was not found. Three exact public low-tier File Blast Radius gate tests passed, but installed/public source drift prevented a broader product pass. Skynet Desktop was not installed in the probed registry or default paths. The run did not install software, kill or relaunch a browser, or repeat the earlier 58-probe sequence.

This is useful negative evidence. It proves that the evaluation ladder can continue across products even when one capability is absent. It does not prove that Chrome CDP or Firefox cannot work on that computer after a separate setup action, nor that File Blast Radius as a whole passed, nor that the current v0.1.3 Desktop release was evaluated there. The target machine identity was verified through the connector, while the connector exposed no separate app-target selector; the label Remote Skynet PC therefore remains an operator label rather than an independently selected app target.

Table 3. Progressive Remote Skynet PC evidence. Unavailable is not pass.

Product	DESKTOP-OJ9K9QR state	Classification	Bounded next route
Chrome CDP	No endpoint on 9222–9230	Unavailable	Rerun exact benchmark only after a safe endpoint exists
firefox-cdp	Firefox not found	Missing dependency	Rerun self-test after separately scoped availability
File Blast Radius	3/3 exact public low-tier gate tests; source drift	Partial verified	Reconcile source identity before broader conformance
Skynet Desktop	Not installed; run observed older release state	Unavailable as installed app	Evaluate v0.1.3 installer in a separately scoped install run

6. Harness upgrade derived from DeepSeek

DeepSeek Harness is a developer-preview, plugin-oriented agent harness [9]. Its testing policy distinguishes unit, coverage, real-API end-to-end, snapshot, and browser-snapshot tiers. It states that line coverage is necessary but insufficient and emphasizes real entry-path tests that check the world because mocks can leave a green suite around a broken product [10]. This study adopts that principle without claiming code lineage or endorsement.

The installed Skynet audit harness implements three durable consequences. First, provider outcomes are typed: refusal, policy, rate limit, capability absence, transport failure, empty response, worker error, and timeout. Second, every type maps to a bounded recovery route; none maps to approval. Third, acceptance is based on local artifacts, executable tests, live URLs, hashes, or remote observations rather than model self-report. Thirty-two focused harness tests currently lock the installed/candidate identity and recovery behavior; that focused test result is reported separately from the four product suites.

7. Evidence contract and claim boundaries

A completion claim should name the evaluated subject, exact version or digest, entry path, environment, expected state, observed state, and independent receipt. Missing evidence is unknown. A typed exclusion describes why a measurement did not run and which next route is safe; it neither erases the requirement nor blocks unrelated measurements.

The product rows in Table 2 cannot be averaged. Tests may differ in granularity, fixture, dependency coverage, and platform interaction. Latency percentiles require identical workloads and environment controls. Release hashes establish artifact identity, not runtime behavior. Remote availability is a property of the observed machine state, not of the product in general.

Table 4. Typed exclusions and bounded recovery behavior.

Observed condition	Typed state	Permitted behavior	Forbidden interpretation
Provider refuses or policy-blocks	refusal / policy	Record exact response; route to another registered proof source	Approval or product failure
Rate limit or timeout	rate_limit / timeout	Bounded wait or alternate registered provider	Infinite retry or pass
Port/dependency absent	capability_absence	Continue independent tests; preserve setup route	Product pass or permanent impossibility
Empty/worker error/transport failure	execution failure	Check receipt and retry once through bounded recovery	Done, approval, or evidence

8. Security and trust boundaries

Attaching to a signed-in browser is powerful because it reaches real cookies, sessions, and extensions. The Chrome connector therefore treats attach-first operation and explicit tab ownership as design constraints, not as claims that browser debugging is generally safe [4]. A routable debugging endpoint can expose the entire profile; loopback binding, per-run secrets, exact tab ownership, and cleanup evidence are mandatory controls.

The Firefox adapter launches an isolated visible profile and limits cleanup to processes carrying its unique profile path [5]. Skynet Desktop reuses existing CLI sign-ins and does not require API keys, but the distributed Windows installer is unsigned. Users must verify the SHA-256 checksum, understand the absence of code-signing publisher identity, and treat release provenance and runtime permissions as separate questions.

9. Threats to validity

The local suites were run on one source checkout and one host. They do not establish behavior across operating systems, browser versions, user profiles, or network conditions. Repository claims and benchmark figures remain first-party until independently replicated. The Remote-PC report contains useful exclusions but evaluated an older Desktop release state and source-drifted Blast Radius installation; it is not a four-product success run.

The study is authored by the product builder, which creates selection and interpretation risk. Mitigations are primary-source citations, exact artifact hashes, preserved raw receipts, explicit failed/unavailable states, and public commands. The formal structure resembles an IEEE-style technical manuscript, but this document is independently published. It is not IEEE-published, IEEE-endorsed, accepted by a conference or journal, or peer reviewed.

10. Reproducibility protocol

Freeze source commits or release tags before testing. Run each product's documented entry path. Capture stdout, stderr, exit code, environment, version, and cleanup. For browser products, record browser build, protocol version, debugging endpoint, owned target, viewport read-back, and screenshot or state evidence. For source analysis, record target path, graph calibration, uncovered radius, and gate reason. For Desktop, record release URL, byte count, SHA-256, signature state, installation path, capability checks, and uninstall cleanup.

On a second host, probe prerequisites once within a declared bound. Classify unavailable capabilities and proceed to independent products. Never install dependencies, restart a user's browser, or reinterpret an exclusion as a pass unless that action is separately in scope. Publish both successful measurements and typed exclusions so another operator can reproduce the observed boundary rather than only the green subset.

11. Conclusion

Verification should follow the product layer. The four Skynet products have reproducible local evidence, but their counts and artifacts do not form a common leaderboard. The Remote-PC run is valuable precisely because it did not manufacture parity: absent ports, a missing browser, source drift, and an uninstalled application remained visible while the ladder continued.

The durable lesson from DeepSeek Harness is not a borrowed slogan but an operational test: exercise the real entry path and check the world. In Skynet, that means lineage-bound artifacts, live browser and URL evidence, typed exclusions, bounded recovery, and no approval from a model response alone. Velocity follows from making the next safe route executable; it does not come from weakening the proof boundary.

Appendix A. Frozen local evidence manifest

Every local source below was hashed before manuscript generation at repository source commit c48205c4a3349b0f154346b6251ed79b067f9a5e. The manifest establishes which bytes support the paper; it does not convert a first-party assertion into independent replication.

Evidence	Bytes	SHA-256
data\runs\skynet_all_products_paper_20260824\jobs\chrome-tests-004f7a2cbfd4eaa4.out.log	193	4f95fba273183169c077485bac8e72d50e170886e2212faead009a88f0bee18e
data\runs\skynet_all_products_paper_20260824\jobs\firefox-tests-038a769c12ba57f.out.log	101	31c727fe7cf7ef968cce9c5b54bcf6d614fbeb829b7c9129be95d9c79300bd8f
data\runs\skynet_all_products_paper_20260824\jobs\blast-tests-0ee42bd5aeeb7dd6.out.log	182	7a708563a2706a07ec8396a0ae35dc3868a83b4ff5acaad56e9f84c3cb916f52
data\runs\skynet_desktop_v013_20260824\LANE2_COMPLETION_AND_HARDENING_20260825.md	8,673	ba087e642fbc54d45e0771d74d6c1b5dbac2ae5ca6d89cadd063e3330b24e84c
data\runs\remote_skynet_progressive_benchmark_20260824\remote_skynet_product_benchmark_20260824_v2.local_rec_overed.json	7,154	8da900f3f9bab1feef9607244dc58b5e129e5eea517e45344ffb6da8ebde401e
data\runs\deepseek_harness_study_20260824\STUDY.md	3,480	7d1b7fb2a5319b8634971f2c2dc7f387ec9f513b19039b8b2ad17d0ad87f9ac7
tools\skynet_audit_harness.py	20,872	ea7cfef30fab40a34de257b327a5ec89323b4571e21c2ae2dad011d091d8cfc3
data\runs\platform_accessibility_gate_repair_20260823\platform_upsert_audit_final_proof\receipt.json	45,665	68578cb68d2a955456d49349e74f2c9f539bc0d497bd68f91727917c9bbd8c88

References

- [1] [Chrome DevTools Protocol](#). Accessed 26 August 2026.
- [2] [Chrome DevTools Protocol, Accessibility domain](#). Accessed 26 August 2026.
- [3] [W3C WebDriver BiDi, Working Draft](#). Accessed 26 August 2026.
- [4] [Zek21, skynet-chrome-cdp](#). Accessed 26 August 2026.
- [5] [Zek21, firefox-cdp](#). Accessed 26 August 2026.
- [6] [Zek21, skynet-ai-file-blast-radius](#). Accessed 26 August 2026.
- [7] [Zek21, Skynet Desktop v0.1.3 release](#). Accessed 26 August 2026.
- [8] [Electron security guidance](#). Accessed 26 August 2026.
- [9] [DeepSeek AI, DeepSeek Harness](#). Accessed 26 August 2026.
- [10] [DeepSeek Harness testing policy](#). Accessed 26 August 2026.
- [11] [Skynet products and verified download links](#). Accessed 26 August 2026.
- [12] [Accessibility-tree node-count field study and formal paper](#). Accessed 26 August 2026.

Author note

Exzil Calanza is an independent researcher and software builder. Product authorship and evaluation authorship overlap; the threats-to-validity section states the resulting risk.