HUGSTONONE

ENTERPRISE EDITION

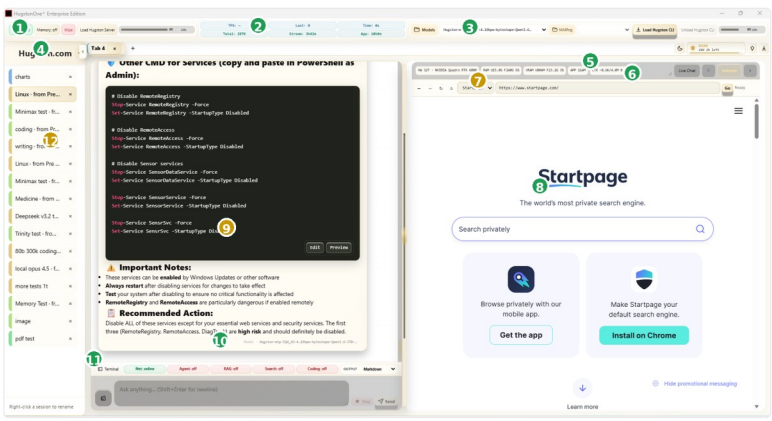
The Most Powerful, Highest Privacy and Local First AI Integrated Workstation to date.

Enterprise product whitepaper with evidence based competitive benchmark

We will focus this paper on some of the main points requested by the users

Figure 1 — Unified control plane with live Internet Mode

This view shows the left conversation pane, the right Preview pane running a live browser, persistent generation metrics and stats, and direct runtime controls in one desktop surface.



Numbered callouts

Green = standout / differentiating capability

1. Authoritative Online / Offline state and memory controls.
2. Persistent generation metrics and stats: TPS, last output, elapsed time, total tokens, stream time and app uptime.
3. Direct model picker, MMProj selector, and Hugston CLI load / unload controls.
4. Session sidebar and multi-tab workspace for parallel tasks.
5. Live hardware, RAM, VRAM, app-memory and context stats inside the Preview header.
6. Internet toggle enables live browser view. This is the visible research mode.
7. Search provider selector, address bar and Go control for directed browsing.
8. Preview window renders the live page without leaving the app.
9. Output can remain in Markdown while the right pane shows a visual target or webpage.
10. Mode badges summarize current state: network, agents, RAG, search and coding.
11. Terminal can stay docked below the conversation for local commands and CLI workflows.
12. Session list on the far left keeps multiple problem threads available without reopening files.

Prepared from user-supplied HugstonOne Enterprise Edition screenshots.

97.0	12	7
Weighted integrated-workstation score	Benchmark pillars	Annotated product figures

Version 1.0 | July 2026 | Public product whitepaper

Document purpose

A rigorous product narrative, evidence framework, and transparent benchmark for HugstonOne Enterprise Edition.

> **Terminology discipline**
>
> HugstonOne Enterprise Edition is described as a standalone cross platform desktop application.

Document control

Document	HugstonOne Enterprise Edition Product Whitepaper 2026
Status	Public product whitepaper
Version	1.0
Evidence date	20 June 2026 research cut; revised July 2026
Benchmark type	Integrated workstation capability benchmark, not a throughput benchmark
Local API	Present in the interface, excluded from scoring until properly validated

Contents

Executive summary

A local AI workstation defined by as a whole, explicit control, and operational continuity.

HugstonOne Enterprise Edition is designed as a complete local AI workstation rather than a single purpose model launcher, web interface, or inference daemon. The application combines local model execution, simultaneous Hugston CLI and Hugston Server controls, Agents skills tools, large source RAG in terabyte, PDF and multifile multiformat processing, Coding Mode, agents, live Internet browsing deepsearch pentest with AI integrated, lower overhead background AI Online Search, encrypted among HugstonOne users Live Chat, session transfer, multi-tab continuity, persistent metrics and stats, and a three-layer memory architecture with an explicit Wipe control nonethe less code preview etc.

The central differentiator is not that each feature exists somewhere in the market. The difference is that these capabilities are presented inside one user controlled desktop workflow, share the same session and workspace context, and can remain local unless the user deliberately enables a network dependent mode offering the highest privacy and independence an AI platform can have right now.

> **Research conclusion**
>
> Within the publicly documented products reviewed by 20 June 2026, no other standalone application was found to document the same complete combination of local inference, runtime choice, large-source retrieval, coding, agents, dual research modes, encrypted user collaboration, session transfer, layered memory, advanced context controls, and persistent local metrics and stats. This is a bounded research conclusion, not a claim about private or unpublished software.

Enterprise value proposition

CONTROL	CONTINUITY	CONVERGENCE
Model, runtime, network, workspace and memory remain explicit	Sessions, tabs and memory preserve long-running work	Inference, RAG, coding, research and collaboration share one surface

What is excluded from the benchmark

- Popularity, funding, user count, investor backing, social reach, and marketplace size.
- Unverified performance claims. TPS and model load superiority require a separate controlled benchmark.

Product thesis and architecture

Six operational pillars define the workstation.

1 Sovereign execution

Direct local GGUF workflows, Hugston CLI, Hugston Server, user-selected compatible runtimes, explicit Online / Offline state.

2 Knowledge at scale

Partial, Semantic, and paged Full RAG with an advanced proprietary algorithm; local PDF extraction; bounded processing instead of unlimited prompt injection.

3 Engineering workspace

Coding Mode, project tree, targeted context, patching, diff, versions, terminal, and visual Preview.

4 Controlled autonomy

Agent workspace scope, permission tiers, shell approval, tools, skills, RAG, coding, and optional network research.

5 Research and collaboration

Live Internet Mode, background Online Search, encrypted peer chat, files, active-tab transfer, session transfer, and all-session transfer.

6 Continuity and observability

Three memory layers, one-click Wipe, persistent per-response metrics and stats, RAM / VRAM / context visibility.

Additional worth mentioning:

No services or additional tools need to be installed for any process to run

No elevation install requirement (install as normal user not sudo)

No python involved (just pure bash scripting)

No http/s protocol mandatory (all can run and execute in Cli)

No updates

No telemetry

No external API

No accounts needed in the app

No internet connection needed ever...

Figure 1. Unified control plane with live Internet Mode AI driven.

Annotated product evidence with numbered feature callouts.

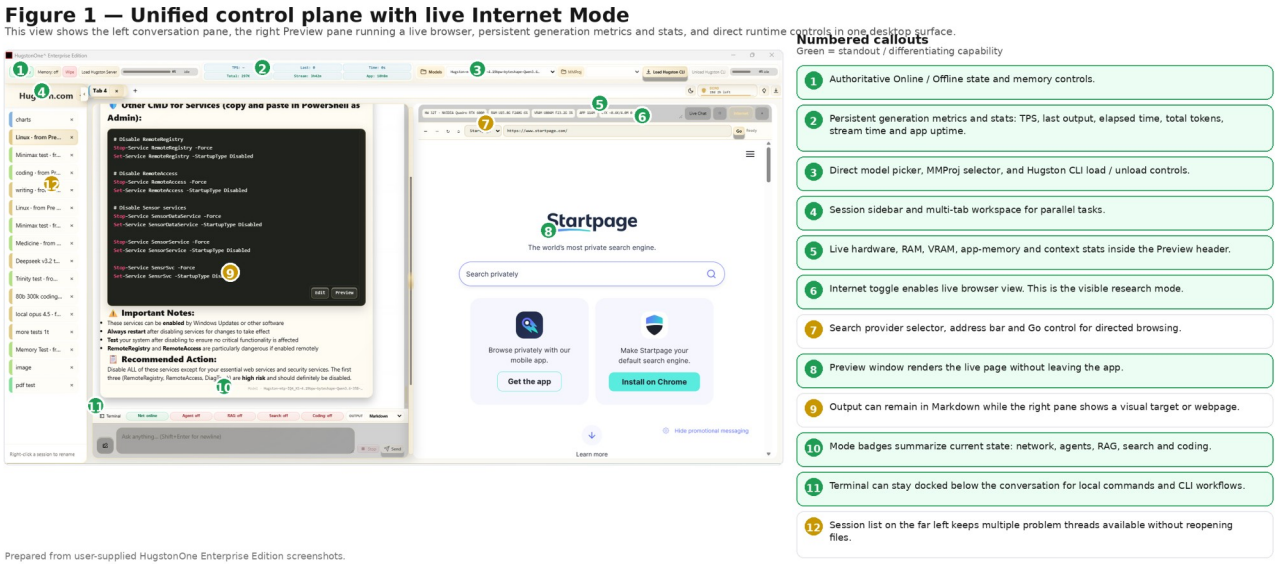


Figure 1. Unified control plane with live Internet Mode.

A single screen already reveals the workstation concept: conversations and tabs on the left, a live browser in Preview on the right, persistent performance metrics and stats across the top, direct model and runtime controls, and docked terminal access below. HugstonOne separates visible Internet Mode from background Online Search to increase hardware performance. The visible mode supports interactive browsing, simple/deep research, pentest and other user directed workflows; background search avoids the cost of continuously rendering the live page.

Differentiating capability

Standout: one application combines a chat workspace, live browser, runtime controls, hardware stats, sessions, tabs, and terminal.

Figure 2. Encrypted Live Chat among HugstonOne users and workspace transfer in one click.

Annotated product evidence with numbered feature callouts.

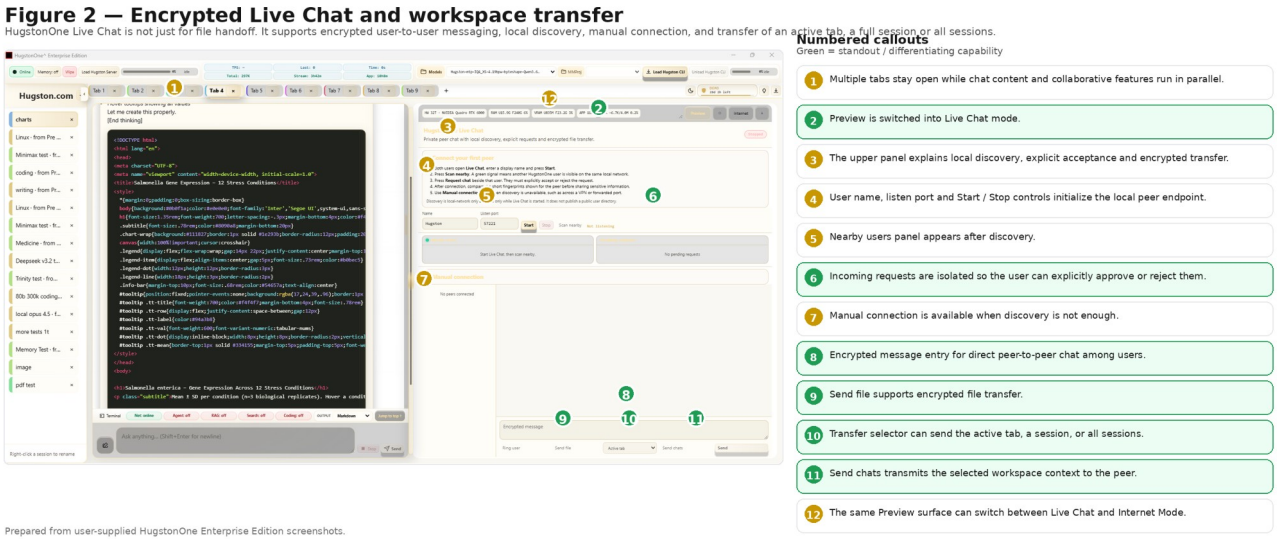


Figure 2. Encrypted Live Chat and workspace transfer.

Live Chat is a sophisticated user-to-user collaboration layer, not merely a file sender. It supports encrypted messages, nearby user discovery, manual connection, encrypted file transfer, and one click transfer of the active tab, the current session, or all sessions among unlimited team members or users. The transfer model preserves full privacy while being locally protecting data and intellectual property up to military grade.

Differentiating capability

Standout: encrypted conversation plus active tab, session, and all session transfer is a collaboration capability not evidenced in the compared desktop runners.

Figure 3. Code on the left, rendered result on the right with local libs.

Annotated product evidence with numbered feature callouts.

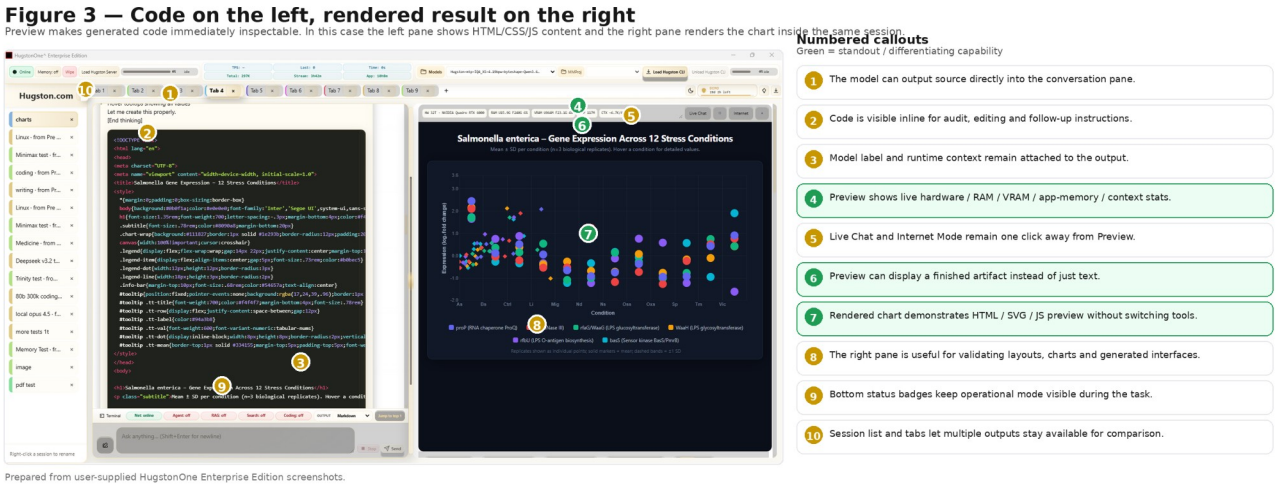


Figure 3. Code on the left, rendered result on the right.

Preview closes the validation loop for HTML, CSS, JavaScript, SVG, charts, and related artifacts. Generated source remains visible for review while the result is rendered to web or local library beside it. This is especially valuable in local coding workflows because the user can inspect, refine, and compare the artifact without leaving the session.

Figure 4. Runtime, long context up to 10 million ctx, and sampling control.

Annotated product evidence with numbered feature callouts.

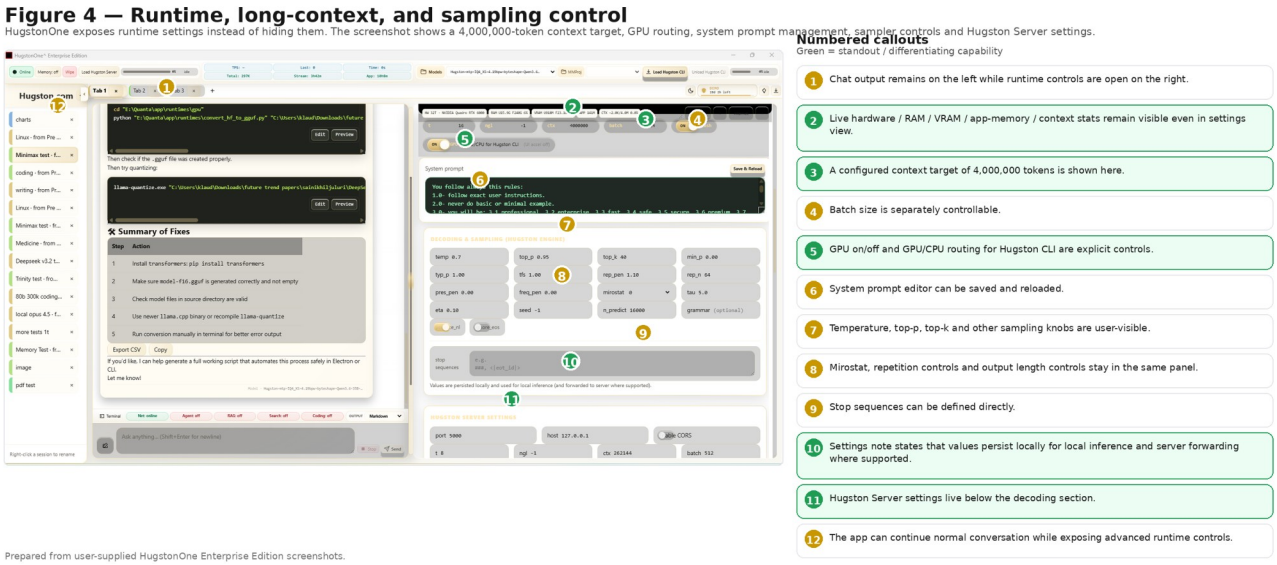


Figure 4. Runtime, long-context, and sampling control.

HugstonOne exposes advanced runtime behavior rather than hiding it behind a single quality slider. The example shows a 4,000,000 token target, batching, GPU routing, system prompt editing, sampler controls, stop sequences, and Hugston Server settings. The product can expose configurations up to 10 million tokens where the model, runtime, and hardware support them. A configured target remains a request while the active runtime allocation is authoritative.

Differentiating capability

Standout: advanced context and runtime settings remain visible, editable, and linked to the same workstation that hosts chat, RAG, coding, and research.

Figure 5. Coding Mode as an AI driven structured engineering workspace.

Annotated product evidence with numbered feature callouts.

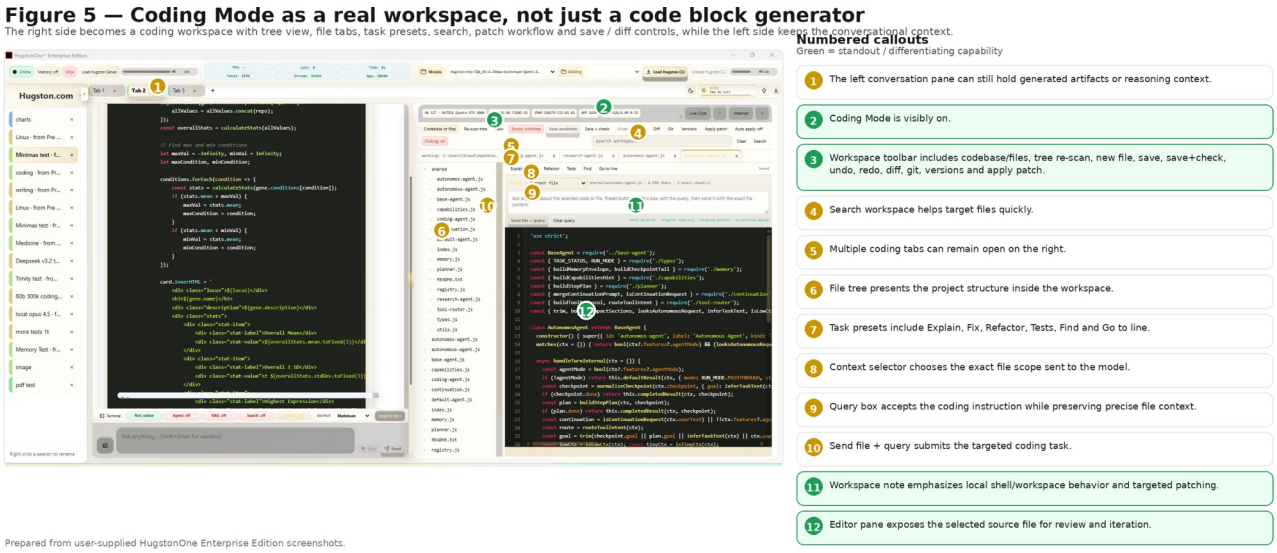


Figure 5. Coding Mode as a structured engineering workspace.

Coding Mode includes a filesystem workspace tree, multiple file tabs, targeted context selection, task presets, search, save and check, diff, versions, patch application, and source review. The chat remains available on the left, allowing a model to explain or refine, interact the change while maintaining explicit file scope.The coding mode is connected optionally with the agents, tools, skills.

Figure 6. Agents, RAG, Online Search, Coding Mode, speculative decoding, and CLI.

Annotated product evidence with numbered feature callouts.

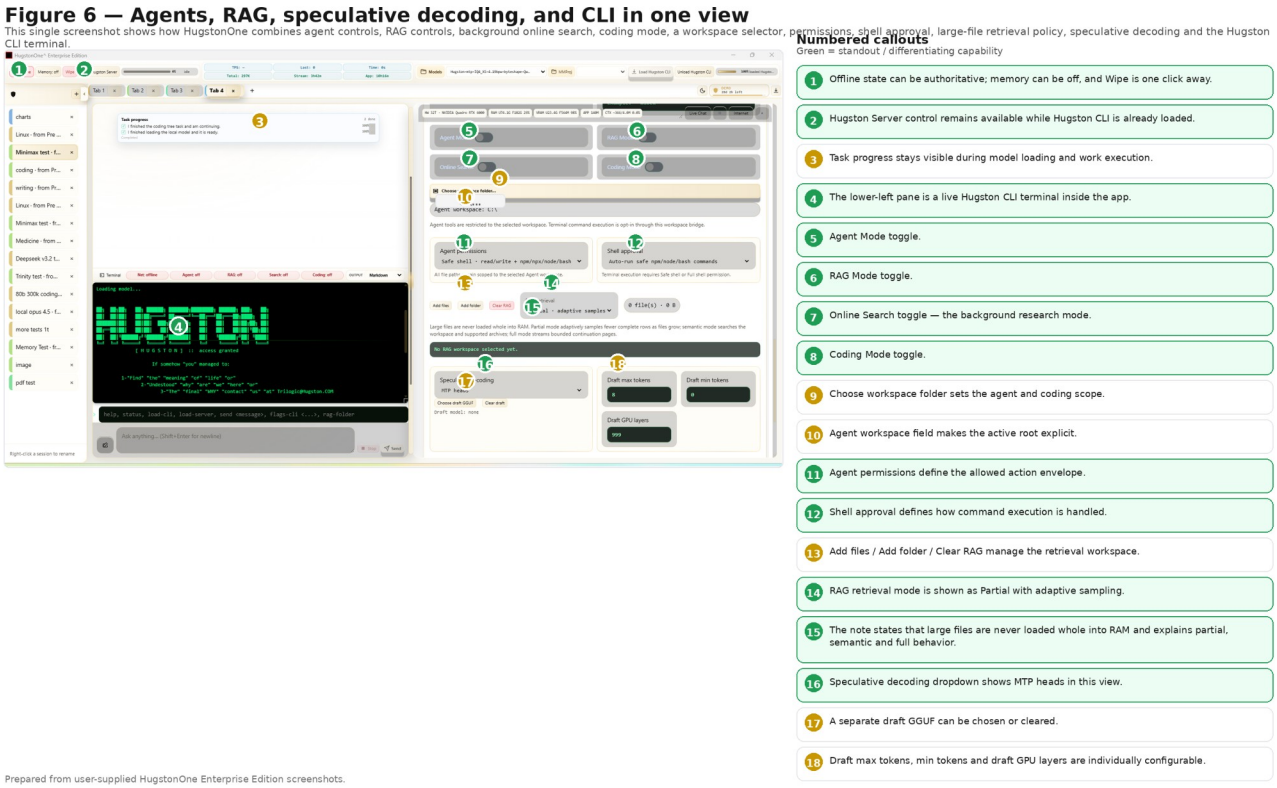


Figure 6. Agents, RAG, Online Search, Coding Mode, speculative decoding, and CLI.

This figure shows the integration density that drives the benchmark result: Agent Mode, RAG, background Online Search, Coding Mode, workspace selection, permission level, shell approval, large-file retrieval policy, speculative decoding, and the Hugston CLI terminal and server they are all interactive and optional to each other. It also shows the memory state and Wipe control, plus Hugston Server availability while CLI work continues.

Differentiating capability

Standout: agent permissions, RAG policy, coding, online search, speculative decoding, CLI, Server controls, memory state, and Wipe are visible in one operational view.

Figure 7. Fully functional Demo of 30 days and Activation process.

Annotated product evidence with numbered feature callouts.

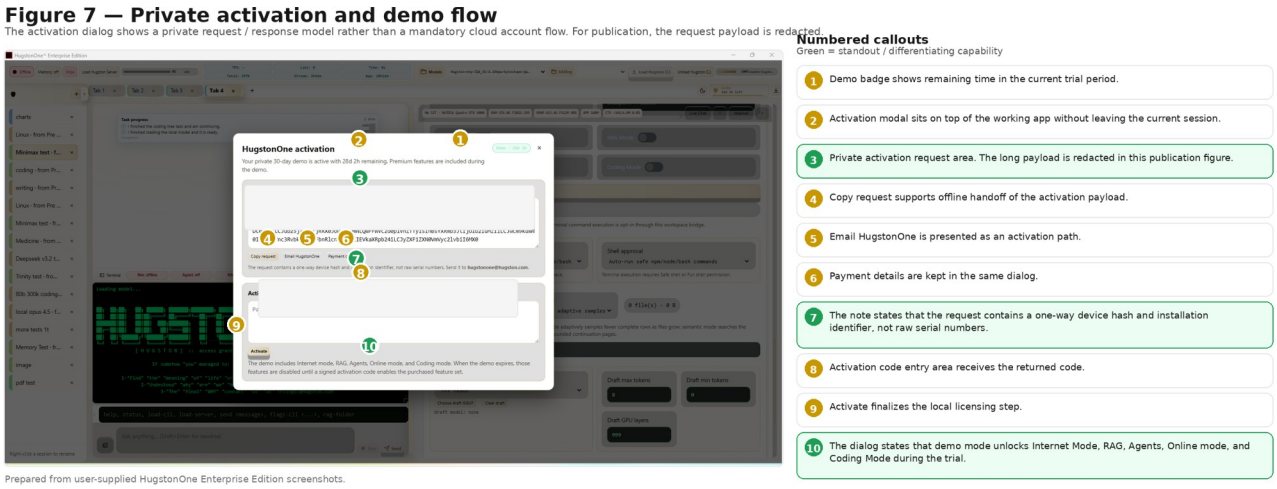


Figure 7. Private activation and demo flow.

There is a demo of 30 days for the users to test all the features. At the end of 30 days users will be able to use the app still with full privacy and all models but some advanced features will be unavailable. For interested users that want to buy the full service, they need to send the payment and the public key, then to receive the license.

Yearly assistance and consultancy may be purchased additionally if so required by the user.

Observe that to protect our user privacy, no full name or identity check is required to purchase the service (unless required by law in the future).

HugstonOne works with all the GGUF models like Deepseek 3 or 4, Qwen 2, 3, 3.x, Mistral, GptOss, Gemma, Minimax, Step3.7, Lfm, MTP, Dflash, Eagle, Ngram etc…

It has a also a very fast image processing beside document processing.

It should be noted that this whitepaper is not meant to undermine any of the other platforms rather then showing and comparing HugstonOne capabilities and innovations to our users.

With that said, we encourage others to do the same to out platform.

Below some benchs.

Benchmark methodology

A transparent capability benchmark for integrated privacy first local AI workstations.

The benchmark measures integrated workstation capability, not raw inference speed. Every pillar is assigned a weight according to its importance to a privacy and local first professional workstation. Each application receives a 0-5 evidence rating per pillar. A score of 5 means first party, integrated, and directly evidenced; 3 means capable but external, prepared, narrower, or incomplete; 0 means not evidenced in the reviewed material.

Benchmark pillar	Weight	What is measured
Offline-first autonomy	15%	First-launch local operation, direct local-model use, explicit network control.
Runtime & model freedom	10%	Direct model ownership, runtime choice, CLI/server flexibility, backend dependence.
Privacy & data control	10%	Local data handling, explicit storage and memory control, outbound data behavior.
RAG, PDF & large-source handling	10%	Document ingestion, PDF support, retrieval modes, bounded large file architecture.
Coding, terminal & preview	10%	Project workflow, patching, terminal, rendered artifacts, validation loop.
Agents, tools & permissions	10%	Agent capability, workspace scope, permissions, shell approval, tool integration.
Internet research modes	7%	Visible browsing, background search, deep research workflow.
Memory, sessions & continuity	8%	Session/tab isolation, continuation, persistent recall, memory wipe.
Encrypted collaboration & transfer	5%	User chat, encrypted files, tab/session/all session transfer.
Long context & speculative controls	8%	Context configuration, runtime limits, speculative families, cache controls.
Metrics & hardware stats	4%	Persistent per-response metrics and local RAM/VRAM/context stats.
Standalone deployment	3%	Desktop packaging, user-level operation, external stack requirements.

Evidence classes

Class	Meaning	Scoring effect
Integrated	First-party capability in the product workflow; directly evidenced.	4–5
External / backend dependent	Requires a separate provider, runtime, service, plugin, or preparation step.	2–3.5
Not evidenced	Not demonstrated by the reviewed official material.	0–1.5

Observe that some platforms are more like a desktop without backend or runtime relying to other apps like HugstonOne to run at all.

Here below we benching and comparing with some of the most popular platforms worldwide.

Overall benchmark result

HugstonOne leads on integrated workstation breadth; competitors retain important specialist strengths.

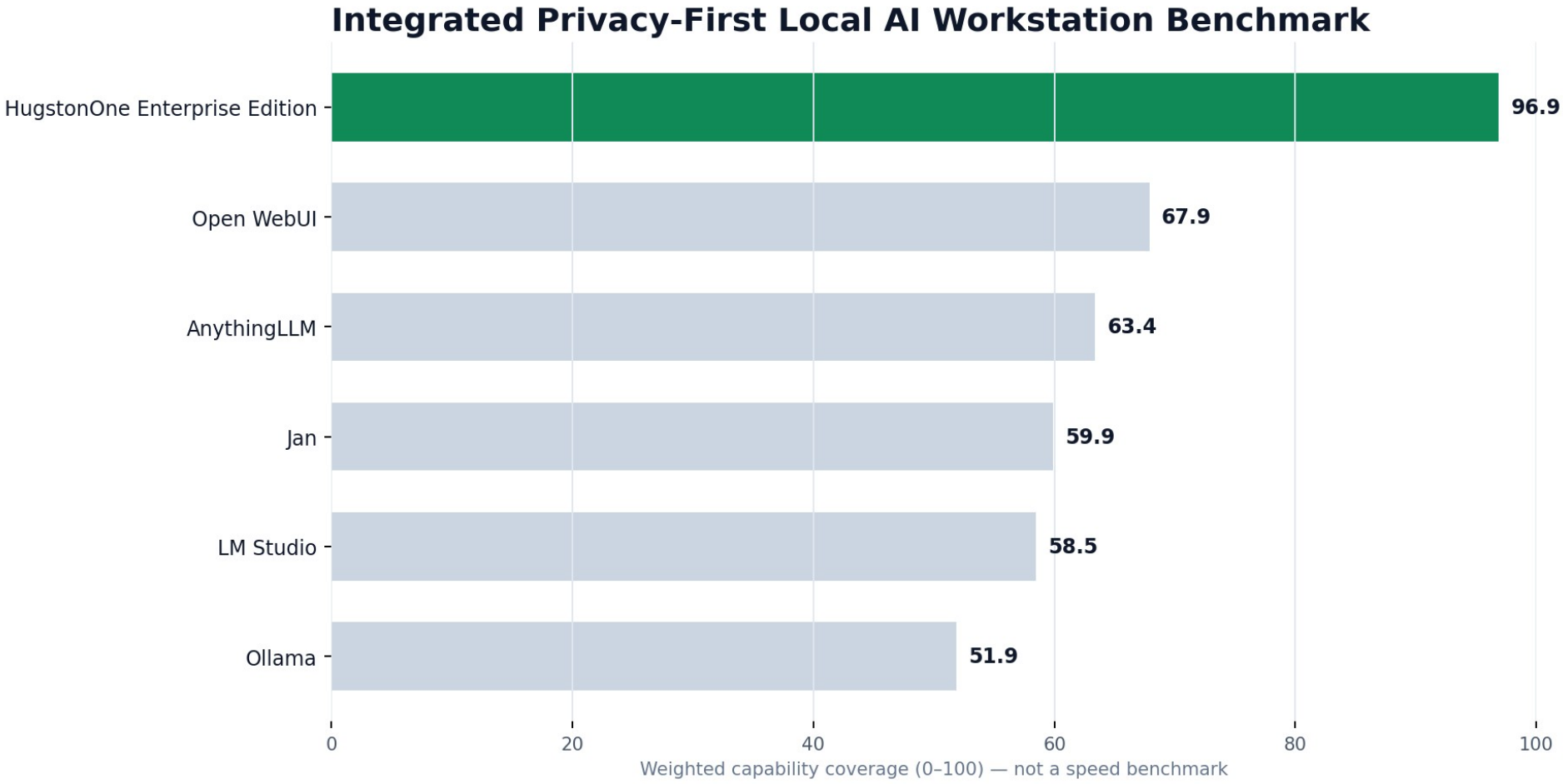


Figure 8. Weighted integrated workstation benchmark. The score is a capability coverage result, not a throughput measurement.

Rank	Application	Weighted score	Evidence-led interpretation
1	**HugstonOne Enterprise Edition**	**96.9**	**Broadest implementation verified integration across all twelve pillars. Local API excluded; cross platform release validation remains a roadmap item.**
2	Open WebUI	67.9	Strong RAG, agents, extensibility, and multi user deployment. Official documentation states that a model provider is required.
3	AnythingLLM	63.4	Strong document and agent workflows. Official Windows documentation describes an Ollama based engine and installation time GPU packages from its CDN; anonymous telemetry can be disabled.
4	Jan	59.9	Strong local llama.cpp engine, direct GGUF linking, and source level transparency. Official quick start states that a default model downloads on first launch.
5	LM Studio	58.5	Strong local runner, document chat, server, import workflow, and speculative decoding. Official docs distinguish offline use from model/runtime acquisition and update checks.
6	Ollama	51.9	Strong runtime, CLI, service, API, and model packaging. Local GGUF import uses a Modelfile and ollama create; it is not an all-in-one workstation.

Detailed pillar scorecard

Ratings are 0-5 and are derived from the evidence rules above.

Benchmark pillar	HugstonOne EE	Open WebUI	Jan	AnythingLLM	LM Studio	Ollama
Offline-first autonomy (15%)	5	3	3.5	2.5	3.5	3.5
Runtime & model freedom (10%)	5	2.5	4.5	2.5	4	5
Privacy & data control (10%)	5	4	4.5	3	4	4
RAG, PDF & large-source handling (10%)	4.5	4.5	2.5	4	3.5	1
Coding, terminal & preview (10%) in GUI	5	3	2	2.5	1.5	1
Agents, tools & permissions (10%)	4.5	5	3	5	2.5	2.5
Internet research modes (7%)	5	4	2	4	1	1.5
Memory, sessions & continuity (8%)	5	4	2.5	4	2.5	1.5
Encrypted collaboration & transfer (5%)	5	1.5	0	1.5	0	0
Long context & speculative controls (8%)	4.5	2	3.5	2	4	3
Metrics & hardware stats (4%)	5	3	2	2	3.5	4
Standalone deployment (3%)	4.5	3	4	3.5	4	4

First-launch and backend evidence

Application	Local model path	Engine / provider relationship	Evidence-based qualification
HugstonOne Enterprise Edition	Direct local GGUF selection	Bundled runtime plus user selected compatible runtime folder	First launch offline autonomy, Standalone desktop application
Jan	Imports and links a local GGUF in place	Built in llama.cpp engine	Official quick start states that a default model downloads on first launch.
LM Studio	Imports via lms import or expected model directory structure	Downloadable LM runtimes	Not standalone, Core use is offline once model and runtime are present; update/model/runtime acquisition can use the network.
Open WebUI	Depends on connected model provider	Requires at least one provider; examples include Ollama, llama.cpp, vLLM, and compatible APIs	Strong interface platform; provider relationship is explicit in official docs.
AnythingLLM	Desktop engine and provider options	Windows docs describe Ollama-based engine and GPU packages from CDN	Strong document/agent product; installation can require network packages; anonymous telemetry is opt-out.
Ollama	GGUF import through Modelfile and ollama create	Own runtime, service, CLI and API	Windows install is per user; workflow is technically oriented and not a full workstation UI.

Competitive application profiles

Specialist strengths are acknowledged; the benchmark rewards integrated coverage.

Open WebUI

A broad, extensible interface platform with strong RAG and agent options. Official documentation states that at least one model provider is required. This makes it powerful as an interface layer, but different from a standalone desktop workstation with a first party integrated inference workflow. Telemetry is collected.

Jan

A strong open source desktop application with a local llama.cpp engine and the ability to link a GGUF in place without copying it. Its official quick start states that the default first launch flow automatically downloads a foundation model so it needs internet connection. Jan receives strong credit for source level control and model ownership.

AnythingLLM

A strong document workspace and agent platform. Official Windows documentation describes an Ollama based local engine and vendor specific GPU packages downloaded from its CDN during installation. Its privacy documentation states that anonymous telemetry is collected unless disabled.

LM Studio

A polished desktop model runner with local document chat, only local server operation, model import from internet connection, and speculative decoding. Official documentation says core use can be offline once models and runtimes are available; model search, runtime downloads, and update checks use connectivity.

Ollama

A strong runtime, background service, CLI, and API. The Windows installer is officially documented as per user and without Administrator rights. Importing a local GGUF requires a Modelfile and an ollama create step. Ollama is excellent infrastructure but does not attempt to be the same kind of all in one workstation.

What makes HugstonOne scores differently’

It has many innovative unique features brought to users locally while gathering them all in one platform.

- It integrates the model workflow, CLI, Server controls, chat, sessions, tabs, RAG, coding, terminal, Preview, agents, research, collaboration, memory, metrics and stats in one desktop surface.
- It separates visible Internet Mode from lower overhead background Online Search.
- It exposes agent scope, permission level, shell approval, RAG policy, context settings, cache controls, and speculative decoding to the user.
- Its Live Chat preserves work structure by transferring an active tab, a full session, or all sessions in addition to encrypted messages and files.
- Its three layer memory model supports active turn continuation, per tab/per session continuity, optional persistent recall, and explicit wiping.
- It has a kill switch for offline enforcment mode.

But here we are not counting only the features but abilities of each feature also.

The benchmark rewards capabilities that share one session, one workspace boundary, one network policy, and one visible control plane. HugstonOne scores highest because the evidence shows these functions working together rather than as unrelated add ons.

2	3	3
Research modes	Memory layers	Workspace transfer levels

Performance benchmark protocol

A scientific framework for proving load time and generation speed claims without manipulating conditions.

> **We factcheck as much as possible to avoid fabricated numbers**
>
> This whitepaper does not insert performance results that have not been captured and published. The protocol below defines how HugstonOne and competitors should be measured on equal terms and we welcome that.

Controlled variables

- Same physical machine and operating system build
- Same GGUF file and checksum
- Same quantization
- Same GPU layer count
- Same context size
- Same batch and micro-batch sizes
- Same flash attention setting
- Same K/V cache formats
- Same prompt and output limit
- At least five cold-load, five warm load, and five generation runs
- Median result reported; best run never used as the headline
- Failures, crashes, retries, and output-quality issues recorded

Required measurements

Measurement	Definition	Publication requirement
Cold model load	Process and model are fully stopped; disk cache condition explicitly stated.	Median, min, max, five+ runs.
Warm model load	Filesystem cache may remain; model process is restarted or reloaded.	Median, min, max, five+ runs.
Time to first token	Elapsed time from request submission to first generated token.	Median and distribution.
Prompt processing rate	Prompt tokens evaluated per second.	Same prompt and context.
Generation rate	Generated tokens per second.	Same output cap and stop conditions.
Peak RAM / VRAM	Maximum host and accelerator memory during load and generation.	Tool and sampling interval disclosed.
Stability	Crashes, failed loads, context errors, interrupted recovery.	Every failure retained in results.
Output quality	Completion correctness and truncation behavior.	Speed cannot be reported without quality checks.

Publication template to update after future testing from us and from users

Application	Cold load	Warm load	TTFT	Prompt tok/s	Generation tok/s	Peak RAM	Peak VRAM
HugstonOne EE	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test
Open WebUI	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test
Jan	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test
AnythingLLM	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test
LM Studio	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test
Ollama	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test	Pending controlled test

Other platforms...

Risks, limitations, and verification roadmap

A whitepaper is stronger when it distinguishes ambition from verified result os below we add with the scope to test and update:

Cross platform release validation

The architecture targets Windows, Linux, and macOS, but each signed release should be tested independently for installation, inference, file access, networking, and cleanup.

Ten million token context

The UI can expose very large context targets. Effective use remains dependent on model behavior, runtime support, RAM/VRAM, KV cache precision, and prompt processing time.

Large source RAG

Terabyte oriented handling refers to bounded indexing, streaming, sampling, retrieval, and continuation. It does not imply that a terabyte is injected into one model request.

Image only PDFs

PDF.js can extract a text layer. A truly image only scan requires OCR or a vision workflow because no text exists to retrieve. The accuracy depends from the multimodal model also included in the platform.

Local API

The control exists in the interface but is excluded from benchmark scoring until request/response compatibility, CORS, lifecycle, concurrency, and failure handling are validated.

Always in optimizations and under development

This platform is meant to be production ready but it may contain bugs and imperfections. Further testing will be done with time.

Independent replication

The strongest leadership claim will come from published packet captures, reproducible benchmark logs, signed hashes, and third party testing from the users in different OS.

Conclusion

HugstonOne Enterprise Edition is strongest in total privacy and where normally fragmented capabilities become one controlled workstation.

The benchmark does not argue that every individual HugstonOne feature is unique in isolation. Local inference exists elsewhere with amazing apps and platforms. RAG exists elsewhere. Agents, coding tools, browsers, terminals, APIs, and collaboration tools exist elsewhere. But they are not easy to work with as they are not user friendly or built in GUI`s but rather in TUI`s The product’s competitive advantage is the concentration of these capabilities inside one standalone cross platform, with a consistent session model, explicit network state, explicit workspace boundaries, direct runtime controls, and visible memory and performance state.

That combination matters because professional local AI work is not one action. It is a sequence: load a model, recover prior context, inspect local files, retrieve evidence, patch code, run a command, preview the output, research the web when permitted, collaborate with another user, and preserve or wipe the resulting context. HugstonOne Enterprise Edition is designed to keep that sequence coherent.

Innovation evidence led position

HugstonOne Enterprise Edition achieved 96.9/100 in this integrated privacy first workstation benchmark. The score reflects the breadth of directly evidenced first party integration, while the whitepaper separately acknowledges competitor strengths and leaves unverified performance claims unfilled tills further updates.

Sources and evidence notes but not necessary facts until tested.

[/] HugstonOne Enterprise Edition supplied source files, product screenshots, annotated figures, and implementation review performed during this project.

[1] Ollama, “Windows” per user installation, model locations, background API, standalone CLI. https://docs.ollama.com/windows

[2] Ollama, “Importing a Model” GGUF import through Modelfile and ollama create. https://docs.ollama.com/import

[3] LM Studio, “Offline Operation” offline core use after models are present; network requirements for search, downloads, runtimes, and updates. https://lmstudio.ai/docs/app/offline

[4] LM Studio, “Import Models” lms import and expected model directory structure. https://lmstudio.ai/docs/app/advanced/import-model

[5] Jan, “llama.cpp Engine” local GGUF management and in place file linking. https://www.jan.ai/docs/desktop/local-engine/llama-cpp

[6] Jan, “QuickStart” default foundation model download on first launch. https://www.jan.ai/docs/desktop/quickstart

[7] Open WebUI, “Quick Start” requirement to connect at least one model provider; provider examples include Ollama and llama.cpp. https://docs.openwebui.com/getting-started/quick-start/

[8] AnythingLLM, “Windows Installation” Ollama-based engine and installation-time GPU support packages from its CDN. https://docs.anythingllm.com/installation-desktop/windows

[9] AnythingLLM, “Privacy & Data Handling” anonymous telemetry collection and opt-out; document erasure behavior. https://docs.anythingllm.com/features/privacy-and-data-handling

Benchmark integrity commitments

No popularity weighting. No fabricated performance results. No Local API score until validation. Competitor qualifications are drawn from official documentation and are separated from HugstonOne implementation evidence.

12	7	10
Weighted benchmark pillars	Annotated product figures	Primary evidence sources

HUGSTONONE ENTERPRISE EDITION | FULL LOCAL CONTROL. HIGH PRIVACY. INTEGRATED INNOVATION AND CAPABILITY. USER IN CONTROL

HugstonOne

No tricks.