

UNDERGROVE

PRODUCT WHITEPAPER

Intelligence runs deeper.

Private AI. Space for independent thought.

Undergrove is in development. This paper describes the product direction, a proposed architecture, and the evidence needed before a live service launches.

VERSION 0.1

19 OCTOBER 2026

01 / THE PREMISE

A place for work that matters.

An early idea needs room. So does a difficult question, a draft that is not ready to share, or a decision with no obvious answer. AI can help with that work. People should be able to understand what they give it in return.

Undergrove is being built as a private AI workspace: a place to think, write, research, and build, with clear choices about models and personal knowledge. Its purpose is to make useful intelligence easier to work with while keeping the person using it in control.

The product, today

The current release is a working interface preview. It shows the intended experience through scripted conversations, model-family choices, agent presets, and document metadata. It is not a deployed private LLM.

Available now	Proposed next	Still to be decided
Website and brand system; scripted workspace; session controls	One live model route; authenticated chat; published data handling	Model and hosting partners; launch timing; service plans

How to read this paper

Architecture and release gates are proposals. They should change when implementation, testing, or user research shows a better approach. No live model, production storage, token, funding round, adoption figure, or performance result is asserted here.

02 / THE PEOPLE AND THE WORK

Start with a useful task.

The first audience hypothesis is independent builders, researchers, and small teams doing sustained work with ideas they do not want to publish. That hypothesis needs interviews and observed use. It is not a claim of market demand.

01	Research	Turn an uncertain question into a plan. Keep source material close to the answer, distinguish evidence from interpretation, and make it easy to revisit an assumption.
02	Writing	Move between a rough thought and a clear draft. Preserve the author's voice, keep changes understandable, and let the user choose what context the model sees.
03	Building	Explain unfamiliar code, compare approaches, and plan a small first version. Any future tool access should have a defined scope and visible permissions.
04	Decisions	Break a complex choice into options and tradeoffs. The product should help people exercise judgment, with uncertainty and limits made clear.

Begin with these everyday uses. Evaluate completed tasks rather than counting features or models. A broad catalog will only be valuable if people can choose sensibly and get reliable work done.

03 / A CLEARER RELATIONSHIP

Privacy should be understandable.

Private AI is already a competitive category. Venice publishes model-specific privacy modes spanning anonymized provider access, contractual non-retention, hardware isolation, and end-to-end encryption. These categories involve different trust assumptions. Its published descriptions are useful context, not an independent verification of the service. [1]

Undergrove's proposed focus is a calm workspace for sustained work, portable knowledge, and clear data choices. That focus must become observable product behavior. It is not evidence that another service lacks those capabilities.

Before a request

Show the selected model, who operates it, where processing happens, what is retained, and which optional features send information elsewhere. Explain a material change before the user submits content.

During the work

Keep memory and connected tools intentional. Let users see what is included, limit access to a project, and stop a task without searching through settings.

After the work

Support useful export and understandable deletion. A person should be able to leave with their work and know what remains in the service.

PROOF BEFORE PROMISES

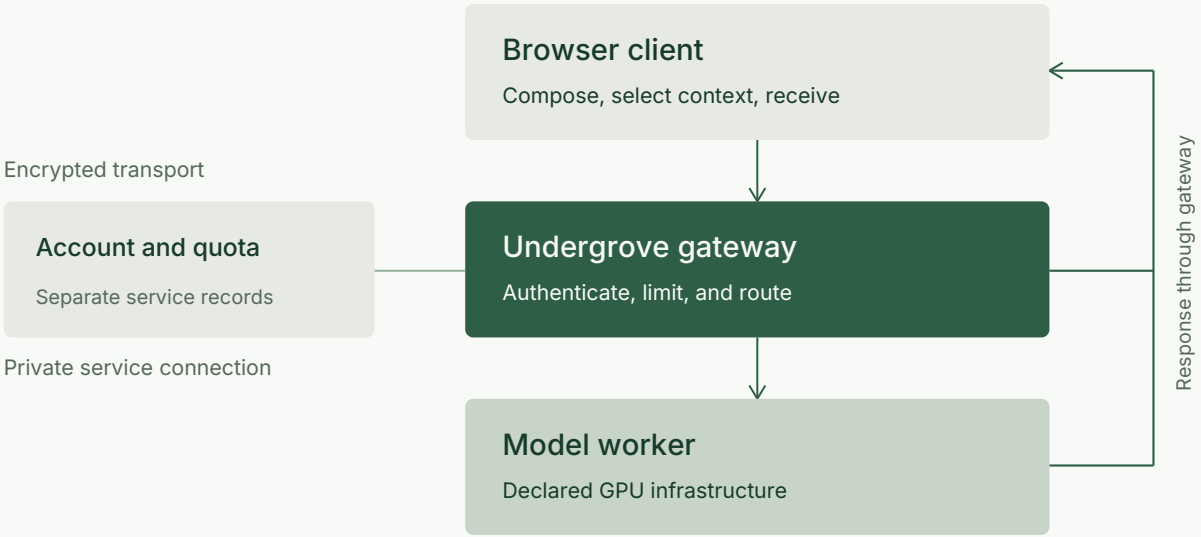
A private-looking interface does not prove private computation. Every published guarantee needs a defined scope, implementation evidence, and a review date.

04 / PROPOSED SYSTEM

A small, inspectable first architecture.

The starting proposal is one suitable licensed open-weight model on declared GPU infrastructure, connected to an Undergrove gateway and a simple browser client. Operating an existing model is a different undertaking from training a new foundation model. A foundation-model programme is outside this first release proposal.

PROPOSED REQUEST AND RESPONSE PATH



Separate responsibilities

Account and quota services should not need conversation content to do their job. Operational telemetry should be designed around explicit fields, with prompt bodies excluded by default. Provider credentials belong on the service side.

The trust boundary

In this proposed first architecture, the gateway and model worker may process plaintext. Transport encryption protects the connection; it does not make content inaccessible to the systems that process it. The cloud operator, deployment configuration, and operational access remain part of the trust model.

Record the model license and version, deployment route, processing location, retention behavior, and operational owners. Publish the actual route when it exists. This diagram is a design proposal, not a deployed system.

05 / TRUST BOUNDARIES

Know what each boundary protects.

The intended protection is for user content, credentials, and workspace data across ordinary use, failure, and recovery. The first release should be explicit about what it can enforce and where it still depends on people, providers, or the user's device.

Boundary	Risk to examine	Proposed response
Device and browser	An extension, shared device, or compromised client exposes content	Minimize sensitive persistence; clear-session controls; describe device limits
Service and operators	Debugging, logs, support access, or misconfiguration exposes prompts	Restricted access; content-excluding telemetry; retention tests and reviews
Other users	A permission, cache, or retrieval error crosses account boundaries	Enforced isolation and negative access tests at every data boundary
Models and tools	Retrieved text changes instructions or triggers unauthorized access	Treat content as untrusted; validate tool calls outside the model [2]

Stronger confidentiality

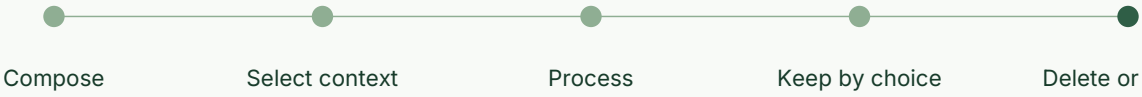
Confidential computing and client-to-enclave encryption are later research options. They require a concrete threat model, verified attestation, key-management design, and independent assessment. They are not implied by the initial hosting proposal.

A compromised endpoint can expose data before encryption. A provider contract is not a hardware guarantee. A blocked prompt is not proof that every injection attempt will fail. Availability and recovery also need their own tests.

06 / DATA LIFECYCLE

Give data a defined life.

Collect deliberately, use for a stated purpose, and retain only what the published policy permits. Content, account data, and operational metadata need separate rules. A blanket claim of "no logs" would hide those distinctions.



Data	Proposed handling	Evidence needed
Prompts and responses	Exclude from training by policy; define processing and retention per route	Deployed configuration, provider commitments, and retention-path tests
Saved work and memory	Optional storage with visible controls; scope to the user or workspace	Access, export, deletion, and recovery tests
Files and embeddings	Treat originals and every derived copy as part of the same lifecycle	Inventory covering text extraction, chunks, indexes, summaries, and backups
Account and usage records	Keep necessary service records separate from conversation content	Field-level purposes, retention periods, access owners, and deletion rules

Test the forgotten paths

Check queues, caches, crash reports, tracing, support systems, and backups as well as the main database. Synthetic test prompts can reveal accidental retention without putting real private conversations into the test. OWASP's logging guidance supports deliberate treatment of sensitive data in logs. [3]

The preview holds conversations in page memory. File names, sizes, and types are used; contents stay unread. Refreshing clears the session. This is not hosted inference.

07 / MODEL EVALUATION

Choose models with evidence.

A model name is not a product specification. The selected version, license, context limits, serving configuration, and hosting route all affect what the service can honestly offer. The first catalog should be small enough to evaluate and explain well.

Dimension	Question to answer before release
Useful work	Does the model complete representative writing, research, reasoning, and coding tasks?
Accuracy	Where does it make unsupported claims, miss evidence, or conceal uncertainty?
Experience	How do first-token latency, response speed, cancellation, and errors feel under load?
Data handling	Does the deployed route behave as the published policy describes?
Economics	What does a completed useful task cost, including retries and operational overhead?

A repeatable comparison

Build a versioned evaluation set from intended uses and consented or synthetic examples. Record the model, prompt, settings, result, and review method. Keep private customer conversations out of evaluation collection by default.

Release and change

Treat a model upgrade or provider change as a release with its own checks. Show material differences in data handling. Do not silently fall back to a route with weaker privacy terms when the preferred route is unavailable.

No launch model lineup, pricing, context size, throughput, or availability target is committed in this paper. Those choices should follow measured quality, capacity, cost, and operational readiness.

08 / KNOWLEDGE AND AGENTS

Context belongs to the person.

Personal knowledge

Memory should be a choice a person can inspect. Proposed controls include view, edit, forget, export, and project scope. The interface should show when saved material will be sent to a model. Keeping a history on a device does not mean that inference also runs there.

Retrieval with boundaries

Document access must be checked before material enters model context. Retrieval should return relevant sources that the user is allowed to see, with citations to the material actually used. Uploaded and retrieved text remains untrusted content. These requirements follow the access and lifecycle concerns described in OWASP's RAG guidance. [4]

Step	User-visible question
Select context	Which documents or memories may this task use?
Inspect a tool	What can the tool read or change?
Approve an action	What will happen outside the conversation?
Review the result	What happened, and can it be undone?

Agents, deliberately

Start with read-only tools. Use narrowly scoped credentials, validate tool arguments outside the model, restrict destinations, and request approval for consequential actions. A document or tool response must not expand the permissions of the task. Prompt-injection defenses need layers and testing; a filter alone is not a complete boundary. [2]

The preview's agent cards are editable prompt starters. They do not run autonomous tools. Its knowledge view demonstrates an interface, not document indexing or AI analysis.

09 / DELIVERY

Advance when the evidence is ready.

Release stages describe decisions and evidence, not calendar promises. The order is intended to keep the first live system understandable, then add capabilities without losing control of its data paths.

Stage	Scope	Gate to advance
0 / Preview	Identity, website, scripted workspace	Honest status, usable navigation, accessible controls
1 / Inference alpha	One model and one declared route	License review, measured quality, data-flow tests, operating ownership
2 / Personal workspace	Saved work, export, optional memory and retrieval	Isolation, citations, deletion, recovery, and user-control tests
3 / Paid beta	Stable service, limits, support, and billing	Capacity, cost, incident handling, and clearly described service terms
4 / Broader choice	More models, API, and scoped tools	Per-route disclosures, compatibility, permission tests, controlled actions

A separate research track

Evaluate local or dedicated deployment, confidential computing, and stronger cryptographic protections when there is a clear user need and a verifiable implementation. These options may have different model, latency, hardware, and feature tradeoffs.

THE NEXT MEANINGFUL STEP

Connect one well-understood model route and demonstrate that its quality and data handling match the published explanation. That is the next meaningful step from a convincing interface to a dependable service.

10 / EVIDENCE

Trust is work that can be checked.

A privacy promise should survive ordinary failures. Verification must cover what happens when requests time out, workers restart, permissions change, or someone asks for help. Design the checks before making the claim.

Scenario	What a release review should establish
Content retention	Synthetic prompts do not appear in undeclared logs, traces, queues, crash reports, or backups
Workspace separation	One user cannot retrieve another user's documents, memory, cached output, or credentials
Deletion and recovery	The published rules match behavior across active stores, derived copies, and backup recovery
Routing failure	An unavailable provider does not cause an undisclosed change in data handling
Tool misuse	Untrusted content cannot authorize a tool or expand the task's allowed access

Evidence to publish

A concise architecture, model and provider register, data-retention table, service limits, incident contact, and a dated record of supported claims. Independent review should identify the scope it assessed and any limits that remain.

Measure useful outcomes

Track task completion, reliability, latency, cost per completed task, and whether users understand the privacy controls. Use the minimum data needed for those measurements. Private content should not become the default price of product feedback.

The interface can be inspected and tested as a scripted preview. It offers no evidence of model quality, production security, service reliability, or private inference. Those are the work ahead.

11 / THE WORK AHEAD

The next step is concrete.

Build a useful first service. Explain how it works. Test the important promises. Give people choices they can understand.

Decisions before launch

Choose the initial model and operator. Define the data policy and operational responsibilities. Set release criteria and test them. Establish access terms and pricing only when the service and its costs are understood.

Version 0.1 is a product and architecture proposal, dated 19 October 2026. It contains no confirmed launch date, token design, investment offer, funding claim, team biography, or customer metrics. Later editions should replace open questions with implementation evidence.

SOURCES / REVIEWED 19 OCTOBER 2026

[1] Venice. Privacy in Venice.

[2] OWASP. LLM Prompt Injection Prevention Cheat Sheet.

[3] OWASP. Logging Cheat Sheet.

[4] OWASP. RAG Security Cheat Sheet.

External guidance informs the proposed controls. It does not validate Undergrove. Product status is based on the current project source and planning documents.

Grow privately.
Think further.

