



# WHITEPAPER

Browser privacy observability and crowdsourced intelligence

## **The short version**

Veilance is a browser extension that shows people what a website is doing inside their browser. When the extension sees behavior that is new or useful, the user can choose to submit a sanitized observation to the Veilance network. Validated contributions can earn VLNC.

Version 0.3 | August 2026

# 1. Why Veilance exists

The web has a visibility problem. A page loads, scripts execute, third-party services answer requests, browser APIs get touched, storage gets written, permissions get queried, and almost all of it happens outside the user's view.

You can open developer tools and piece some of this together. Most people will not, and they should not need to. A privacy tool should be able to answer a straightforward question: what did this site actually do in my browser?

That is the first job of Veilance. The extension is useful before a token, before a network, and before a user ever submits anything. It observes privacy-relevant behavior locally and explains it in terms a normal person can understand without hiding the technical evidence from people who want it.

The second job is to turn isolated observations into useful intelligence. Tracking behavior changes. Scripts move. Vendors get replaced. Custom fingerprinting code will not always match a public blocklist. A shared set of independently observed behavior can fill in those gaps faster than a static list can.

## 2. What Veilance actually looks at

Veilance is not trying to record everything a page does. That would defeat the point. The detection surface is intentionally limited to behavior that is relevant to browser privacy and can be represented without collecting the user's content.

- Third-party hosts contacted by the page, including known analytics, advertising, identity and tracking infrastructure.
- Canvas, WebGL, AudioContext and other API patterns that can contribute to browser fingerprint construction.
- Cookie and browser-storage operations as events and metadata, without collecting the stored values.
- WebRTC, media-device and environment enumeration that can expose device or network characteristics.
- Client Hints, hardware capability queries and other browser attributes commonly used to build a device profile.
- Permission requests, service workers and selected persistent background behavior.
- Privacy and security policy information such as Content Security Policy when it is available to the extension.
- Normalized identifiers for scripts or behaviors when they are useful for matching and can be stored without retaining user content.

A detection is not automatically an accusation. AudioContext has legitimate uses. Canvas has legitimate uses. WebRTC has legitimate uses. Veilance should report what happened, why the behavior is relevant, and how confident the detection is. Intent should not be invented when the evidence does not support it.

### 3. Local first, because it has to be

A privacy extension that sends every page visit to a central server is the wrong product. Ordinary Veilance detection should work locally. The browser should not need to report a user's history just to explain the page that is open.

When Veilance finds something that may improve the network, contribution is a separate action. The report is normalized and sanitized in the extension. The user can inspect the payload before it leaves the browser.

#### **Data that should never be routine telemetry**

Passwords, form values, authentication tokens, authorization headers, clipboard contents, cookie values, arbitrary page text, screenshots, private DOM content, and raw query strings that may contain identifiers.

The contribution schema and sanitization rules should be documented publicly. Trust is easier to earn when a user can see the boundary instead of being asked to accept a vague promise that data is “anonymous.”

### 4. From observation to intelligence

A submitted report is not automatically accepted as truth. If VLNC has value, people will try to farm it. The network therefore has to treat every contribution as untrusted input until it has been checked.

| Stage       | What happens                                                                                  |
|-------------|-----------------------------------------------------------------------------------------------|
| Observe     | The extension records privacy-relevant events locally.                                        |
| Sanitize    | Excluded values are removed and identifiers are normalized before submission.                 |
| Deduplicate | The backend checks whether the same observation or equivalent behavior is already known.      |
| Validate    | Schema checks, replay protection and abuse scoring determine whether the report is plausible. |
| Corroborate | High-value findings can remain provisional until independent browsers reproduce them.         |

| Stage  | What happens                                                                                                  |
|--------|---------------------------------------------------------------------------------------------------------------|
| Accept | A validated observation updates the site or behavior profile.                                                 |
| Reward | Eligible discovery or confirmation credit is added to the contributor's VLNC balance for later Solana payout. |

## 5. What makes a contribution worth paying for

The wrong incentive is “visit websites and get paid.” That creates a bot farm on day one. Veilance should pay for intelligence that adds measurable value to the network.

A reward can be based on a combination of novelty, reproducibility, confidence, relevance and contributor reputation. An observation that has already been submitted a thousand times is not worth the same as the first credible report of a new fingerprinting implementation on a widely used service.

A useful model is to split discovery from confirmation. The first credible contributor receives discovery credit after validation. Independent users who later reproduce the same behavior can receive smaller confirmation credit. If the finding never survives validation, nobody gets paid simply for being first.

Rewards should accrue off-chain and settle after a threshold or interval. There is no reason to create an on-chain transaction for every observation.

## 6. VLNC on Solana

VLNC is the Veilance reward and utility token on Solana. The token is used to pay contributors whose uploaded telemetry adds validated intelligence to the network.

### VLNC token mint (Solana)

H4scGE64jRY67EcDmEK9XuMnBbkkE2tNtFM2Ni3pump

### Trade VLNC

<https://pump.fun/coin/H4scGE64jRY67EcDmEK9XuMnBbkkE2tNtFM2Ni3pump>

The reward path is deliberately tied to accepted telemetry, not browsing activity. A user can run Veilance without connecting a wallet and without uploading anything. If that user chooses to

contribute a sanitized observation, the backend evaluates the report for novelty, validity, duplication, corroboration and abuse signals. Only eligible contributions create reward credit.

Accepted credit can accumulate in the contributor account and be paid out in VLNC to a Solana wallet after a minimum threshold or settlement interval. This avoids creating an on-chain transaction for every report while still making the payout auditable on Solana.

The practical flow is simple: observe locally, choose to upload, sanitize before transmission, validate on the network, then pay the contributor in VLNC if the telemetry is accepted. Rejected, replayed, duplicate or manufactured reports do not receive a payout.

A wallet should be required only for receiving rewards, not for using the extension. Keeping detection separate from identity reduces the amount of information Veilance needs from users who only want local privacy visibility.

VLNC should not be the reason the browser extension exists. If the extension is not useful without the token, the product is backwards. The token exists to make useful privacy telemetry worth contributing and to support network-native utility as the intelligence layer grows.

## 7. Abuse is part of the design problem

Any open reward system becomes an adversarial system. A wallet is not a person. A browser instance is not a person. A domain is not automatically a useful source. Veilance has to assume automation, replay, self-generated sites and coordinated submissions will happen.

- Rate limits and delayed settlement instead of instant per-event payouts.
- Replay-resistant signed submission envelopes and versioned schemas.
- Reputation based on previously accepted contributions, not wallet balance.
- Diminishing rewards for repeated, closely related or low-value observations.
- Independent corroboration before high-value findings receive full credit.
- Domain and traffic-quality signals to reduce incentives for manufactured sites.
- Anomaly detection for coordinated browser fleets or submission timing patterns.
- Manual review for rare findings that are too valuable to trust to a simple automated score.

The anti-abuse system does not need to be perfectly decentralized to be useful. The first requirement is that it is difficult to extract rewards without contributing legitimate intelligence.

## 8. The intelligence layer

Over time, the useful asset is not the extension itself. It is the historical privacy map created from validated observations.

A Veilance profile can show what an origin does, which third parties it contacts, which fingerprinting techniques are present, what changed, when the change first appeared, and how many independent observations support the finding. A Veilance Score can summarize that profile, but the underlying evidence should remain available. A single number without evidence is not enough.

This intelligence can support more than end users. Researchers, security teams, journalists, browsers and other privacy products can consume the same normalized data through Veilance Intelligence and the Veilance API.

## 9. Product surface

- Veilance Extension - local browser privacy observability, explanations and contribution controls.
- Veilance Intelligence - normalized and corroborated privacy telemetry collected over time.
- Veilance API - programmatic access to site profiles, techniques, third-party relationships and historical changes.
- Veilance Score - an explainable privacy summary backed by the observations that produced it.
- VLNC Rewards - accounting and Solana settlement for accepted uploaded telemetry.

## 10. Rollout

The rollout should be boring on purpose. The extension needs to prove that it can detect useful behavior before an incentive layer is allowed to distort the data.

- Phase 1 - Detection. Ship local observation, technical evidence and plain-language explanations. Publish the telemetry boundary.
- Phase 2 - Contribution. Add opt-in sanitized reports, deduplication and public visibility into what the network has learned.
- Phase 3 - Validation. Introduce corroboration, contributor reputation and controlled VLNC payout experiments.
- Phase 4 - Network. Launch production VLNC payouts on Solana and intelligence API access after the economics, abuse controls and compliance model survive testing.

## 11. Trust model

Veilance is asking users to install software that can observe privacy-relevant browser behavior. That is a high-trust position. The answer is not a longer privacy policy. The answer is inspectability.

Core client behavior, the telemetry schema, sanitization rules, scoring methodology and material collection changes should be documented. Where practical, client-side logic should be open to independent review. Backend anti-abuse systems may need operational secrecy, but that cannot become an excuse to hide what the browser transmits.

The extension should also separate fact from inference. “AudioContext was used in this pattern” is an observation. “This site is malicious” is a conclusion that requires substantially more evidence. Veilance should be conservative about the second statement.

## 12. What Veilance is not

Veilance is not an ad blocker, antivirus replacement, legal privacy audit, or system for declaring every analytics script malicious. It is an observability and intelligence product. It shows what happened, explains why it may matter, keeps uncertainty visible, and gives users a way to improve the shared dataset without handing over their browsing history.

## 13. Closing

Most privacy problems are hard to reason about because the important behavior is hidden. Veilance makes that behavior visible where it happens: inside the browser.

If the extension can do that well, the rest follows naturally. Useful observations can become shared intelligence. Independent reports can make that intelligence more trustworthy. VLNC gives Veilance a direct way to pay the people who upload telemetry that survives validation instead of rewarding noise.

## See what websites don't show you.

**Important notice.** This document describes a proposed technical network and incentive design. It is not an offer to sell securities, investment advice, or a promise of token value. Token mechanics, rewards and distribution remain subject to technical, economic and legal review. Trading VLNC involves market risk and is separate from earning contributor rewards.