Deltawall

WHITE PAPER

Last-mile AI data protection

Why SASE, CASB and endpoint DLP miss the prompt, and what an interaction-layer control has to do instead.

September 2026 · Deltawall B.V. · deltawall.ai

Last-mile AI data protection

> Generative AI moved data exposure to the one place the enterprise security stack does not look: a text box inside the browser, on a site the organisation has already approved. This paper explains why each layer of a modern stack misses that moment, and derives the requirements for a control that works at the point of interaction.

1. The shift to file-less, last-mile data movement

For two decades, data-loss prevention was built around files and networks. Sensitive data lived in documents; documents moved as attachments, uploads and transfers; controls inspected those files at the endpoint, at the mail gateway or at the network edge.

Generative AI changed the unit of movement. The most common way sensitive data now reaches an external AI system is not a file at all. It is a **paste**: a customer record, a contract clause, a block of source code or an API key copied into a prompt so that an assistant can help with the work.

Three properties make this channel different:

- **It is file-less.** The data exists only as text in a form field, for the few seconds between paste and submit.
- **It is encrypted in transit.** By the time the request leaves the device, it is inside a TLS session to the AI provider.
- **It goes to a permitted destination.** The AI site is often sanctioned, or at least not blocked, because the organisation wants the productivity it brings.

We call this **last-mile data movement**: the final step between a person and the system they are feeding, which happens inside the browser and nowhere else. It is also where uploads happen: a spreadsheet dropped into an assistant for analysis carries far more regulated data than any single prompt.

2. What each control layer sees, and what it misses

Each layer of a modern security stack is correct about what it observes. None of them observes the interaction.

APPROACH	WHAT IT SECURES WELL	WHAT IT MISSES AT THE LAST MILE
SASE and SSE	Network access, connectivity and application-level policy	The prompt, the paste and the upload inside an encrypted session to a permitted site
Proxy and SWG with TLS inspection	Routed web traffic, URL and category filtering	Needs a root certificate on every device and decrypts all traffic; still sees a request rather than who typed what, and why
CASB	Sanctioned SaaS through APIs: configuration, sharing and posture	Real-time text input into AI tools, and personal accounts that never touch the API
Endpoint DLP and EDR	Files, processes, devices and removable media	File-less actions: typed prompts and text pasted into a web form

The gap is structural rather than a matter of tuning. A network control sees a connection; a cloud control sees an application; an endpoint control sees a process. The risk is an **interaction**: a specific value, typed or pasted by a specific person, into a specific field, about to be submitted.

There is a second, quieter problem. The controls that can reach into encrypted traffic do so by **interception**: installing a root certificate and decrypting every session. That creates its own exposure, operational fragility and, in many European organisations, a works-council and privacy conversation that never ends.

3. Requirements for interaction-layer protection

From that gap follow six requirements for a control that closes it.

1. **Inspect at the point of interaction.** The control must see the content of the input and the upload in the page, before submit, not the encrypted request afterwards.
2. **Detect on the device.** Sending content elsewhere to be analysed recreates the exposure the control exists to prevent. Classification must happen locally.
3. **Enforce before send, with more than two outcomes.** A binary allow-or-block either stops the work or lets everything through. The control needs graded actions, including masking a value while the rest of the message proceeds, and a managed bypass that records the justification.
4. **Do not intercept.** No root certificate, no decryption, no network rerouting, no kernel components.
5. **Record decisions, not content.** Evidence is essential, but a log that stores prompts becomes a second copy of the sensitive data.
6. **Make the record verifiable by someone else.** An auditor or regulator should be able to check the record without trusting the vendor or the employer.

The rest of this paper describes how Deltawall meets each requirement, and where it deliberately does not.

4. Detection on the device: checksums, context and a model

Deltawall classifies content in the browser with two layers.

The **deterministic layer** handles data with a known structure. Twenty-one detectors cover financial data, national identifiers, personal data and secrets, and each validates its match rather than merely pattern-matching it:

- **Checksum validation** proves the number itself is well-formed: IBAN by mod-97, payment cards by the Luhn check, the Dutch BSN by the eleven-test, the German tax ID, and EU national IDs.
- **Format validation** checks structure for classes such as Dutch VAT numbers, phone numbers, IP addresses and cloud credentials.
- **Context gating** applies to identifiers with no public checksum, such as KvK and BIG numbers and dates of birth. They fire only when a matching keyword sits beside them.

The **model layer** is a 13 MB ONNX named-entity model that runs on the device and classifies the four data types no pattern can describe: person names, postal addresses, health data and organisation-confidential text. It labels text. It does not generate, summarise or decide.

When the two layers disagree, **the checksum wins**. The model can add coverage; it can never override a verified identifier. The same engine parses uploaded files (PDF, DOCX, XLSX, PPTX, CSV, TXT, Markdown and JSON) before the upload request is made. Content it cannot inspect, such as a scanned PDF, is labelled uninspectable rather than guessed at, and the policy decides what happens to it.

5. Enforcement without interception

Deltawall enforces through a Manifest V3 browser extension for Chrome, Edge and Firefox that acts as both **sensor and enforcer**. It reads the message box and the upload field on covered AI sites, asks the policy engine for a decision, and applies it in the page.

The policy engine resolves four conditions (department, data class, destination and channel) to one of five actions:

ACTION	EFFECT AT THE POINT OF INTERACTION
Allow	The request proceeds.
Log	The request proceeds and the decision is recorded.
Warn	The user is told why; the policy can require a written business justification before proceeding.
Redact	The sensitive value is masked in the input box; the rest of the message goes through.
Block	The submission or upload is stopped, and the reason is shown.

New rules run in simulation against live traffic before they enforce anything, so a security team can see what a rule would do before users feel it.

None of this requires interception. There is no proxy, no root certificate, no TLS decryption and no kernel driver. The extension runs in user mode inside the browser sandbox and is deployed like any other managed extension, through enterprise browser policy. For traffic that never passes through a browser, such as scripts, integrations and agents calling model APIs, an opt-in OpenAI-compatible gateway applies the same policy with payload redaction before the provider. Desktop AI applications are governed through MDM policy bundles (AppLocker, DNS blocklist or Jamf profile), not intercepted.

6. Evidence without a second copy of the data

Every decision produces one record containing a data-class label, a salted one-way hash of the match, the destination, the channel, the rule and the outcome. It never contains the prompt or the file. Subjects are recorded as pseudonymous tokens; identifying a person requires a second approver who is not the requester, and the unmasking is itself recorded.

Records are made tamper-evident in three steps:

1. **Chained.** Each record carries the hash of the one before it, so changing any record breaks every later link.
2. **Sealed.** Records are sealed in batches under a Merkle root, so a whole period can be proven at once.
3. **Signed.** Seals are signed with Ed25519, and exported packs carry the public key needed to check them.

An exported evidence pack is verified with a **standalone offline tool**. The auditor runs it on their own machine; nothing contacts Deltawall. Flip a single Block to Allow anywhere in the pack and verification fails. The property protects the employee as much as the employer: nobody can quietly rewrite the record.

7. Honest limits

An interaction-layer control is powerful because it is specific, and its limits come from the same specificity. We print ours:

- **Coverage is bounded to named AI sites**: ChatGPT, Claude, Gemini, Microsoft Copilot, Perplexity and DeepSeek today. The site catalogue updates without an extension release.
- **There is no OCR.** Scanned documents and images are flagged as uninspectable, not read.
- **Redaction applies to typed prompts.** A file that needs redaction is stopped, not rewritten.
- **The on-device model runs in Chrome and Edge.** Firefox enforces with the deterministic layer.
- **It fails open.** If a scan errors or times out, the AI site keeps working. A broken scanner must never break the workday.
- **It complements, rather than replaces,** network, identity and endpoint controls. It covers the moment they cannot see.

Conclusion

The data your organisation most needs to protect now leaves through a text box. Controlling it requires a control that sits at that text box: detecting on the device, enforcing before send, intercepting nothing, and recording decisions in a form that anyone can verify.

To see this running in a real browser, with a prompt redacted, a file stopped and an evidence pack verified offline, book a 30-minute demo at **deltawall.ai/demo**.

Deltawall B.V. · info@deltawall.ai · deltawall.ai