Rampart — A Local-First System for PII Redaction

National Design Studio

2026

Abstract

We describe **Rampart**, a local-first system for removing personally identifiable information from user-typed text before it leaves the browser. The system combines a 14.7 MB ONNX token-classification model with a deterministic recognizer layer; together the two layers form the first gate of a defense-in-depth pipeline. Across the seven Latin-script languages the model was trained on, on a 30,000-row held-out test set drawn from the OpenPII 1.5M corpus, the full system achieves 98.42% private-term recall at 6.6 ms median latency in Node ONNX, and runs at 3.9 ms p50 on WebGPU in the browser. Rampart is harm reduction, not perfect protection: redacting in the browser shifts the trust boundary so that a downstream failure can only leak what the client failed to remove. This document describes the architecture, the alternatives evaluated and rejected, the training and evaluation methodology, operating-point and calibration analysis, known failure modes, and the licensing under which the system is released.

1. Design goals

The system is built around four constraints:

1. **Local-first privacy as harm reduction.** Personal information is removed before reaching application infrastructure. Data the server never receives cannot be leaked by a model provider, a logging system, an analytics pipeline, a third-party SDK, or a future compromise of the application's own backend. This is the threat model that motivates the entire system: every other constraint follows from refusing to put the user in a position where they have to trust a remote operator with their unredacted text.
2. **Browser-deployable.** The shipped artifact must fit on a low-end mobile phone. Targeting under 15 MB on the wire ruled out most modern PII NER models, including GLiNER (≈50 MB) and DistilBERT-based detectors (≈64 MB).
3. **Recall-biased.** Misses leak data, so the default policy redacts whenever a detector fires above its calibrated threshold. Over-redaction has a real cost, too: a chat assistant that cannot see context cannot help, but at a smaller cost than a leak.
4. **Domain-aware retention.** Useful assistants often need rough context, like coarse geography, to be helpful. The keep-set (`{CITY, STATE, ZIP_CODE}`) is policy-driven so applications can tune the boundary between privacy and utility without retraining; the precise street line is redacted.

2. Architecture

Rampart ships as two cooperating layers that run in parallel and merge their outputs. Each layer is designed to cover a class of failures the other layer cannot reliably handle on its own. Both run

entirely in-browser.

2.1 The deterministic recognizer layer

Real-world PII often has structure that is faster and more reliable to validate than to learn. The deterministic layer is a curated set of regular expressions paired with checksum and structural validators. It owns five classes end-to-end:

- **Luhn checksum** for payment cards, matched over the digit projection so every separator form collapses to one rule.
- **SSN structural rules** that reject reserved areas (000, 666, 9XX) and ZIP+4 codes that pattern-match the SSN shape.
- **Pattern-backed detection** for email addresses, URLs, and IP addresses (IPv4, IPv6, and MAC), where the structure lives in the punctuation.

These detectors are synchronous and run on the raw input, so this structured PII can be removed even before the model has loaded. Because cards and SSNs use checksums and structural rules rather than only shape matching, false-positive rates are very low: a 16-digit number that fails Luhn is not redacted as a card; a 9-digit number with a reserved-area prefix is not redacted as an SSN. Classes with no checksum — phone, routing, tax, government-ID, passport, and license numbers, and street-address components — are deliberately left to the model rather than guessed at with a regex.

2.2 The token-classification model

The deterministic layer cannot recognize contextual PII — names, phone numbers, account/routing/tax numbers, government IDs, passports, licenses, and free-form address components. For these, Rampart uses a MiniLM-L6-H384 encoder fine-tuned on a 35-label BIO head (17 entity types). The tokenizer is an uncased WordPiece tokenizer trimmed to 19,730 pieces (from BERT-uncased's 30,522). Single-character pieces are always retained, which preserves WordPiece's character-level fallback for rare names and out-of-vocabulary content.

2.3 Span repair

HuggingFace's standard `aggregation_strategy="simple"` produces fragmented spans when subword B-`GIVEN_NAME` probabilities outrank I-`GIVEN_NAME` inside a name, e.g. "Zaccarino" tokenized as "Zac"+"##car"+"##ino" can come back as three separate name spans. Rampart applies three layers of post-processing:

1. **Adjacent-span merging** collapses consecutive same-label spans separated only by name-internal punctuation (space, hyphen, apostrophe, period, comma).
2. **Iterative bridge-and-merge** rescues low-confidence candidates (score between the 0.15 extend floor and the 0.4 keep threshold) when they bridge two high-confidence spans of the same label. This catches names like "Jose [LOW] de [HIGH] Garcia" where a particle scored below the cutoff but is structurally part of the name.
3. **Capitalized-particle rescue** grows name spans (`GIVEN_NAME` / `SURNAME`) to swallow short capitalized name particles ("De la", "Von", "Mc") sitting inside a name's flow.

This composition lifts span-F1 to 0.53 strict (IoU=1.0) and 0.66 relaxed (IoU≥0.5) on the 30,000-row test set, well above the fragmented spans the default aggregation produces.

2.4 Defense in depth

The two layers are complementary, not redundant. Each owns the classes the other handles poorly:

Layer	Owns	Why
Deterministic	`SSN`, `CREDIT_CARD`, `IP_ADDRESS`, `EMAIL`, `URL`	Checksum/structural validation (cards, SSNs) or punctuation-anchored patterns (email, URL, IP) give near-perfect precision and recall, including under orthographic perturbation.
Model	`GIVEN_NAME`, `SURNAME`, `PHONE`, `TAX_ID`, `BANK_ACCOUNT`, `ROUTING_NUMBER`, `GOVERNMENT_ID`, `PASSPORT`, `DRIVERS_LICENSE`, `BUILDING_NUMBER`, `STREET_NAME`, `SECONDARY_ADDRESS`, `CITY`, `STATE`, `ZIP_CODE`	Open-ended, context-dependent, and free of any checksum to validate against — exactly what an enumerable rule set cannot keep up with.

The model handles open-ended cases in shapes the regex catalog cannot predict for; the deterministic layer covers the structured classes where exact-character recall matters and adversarial inputs where a Luhn-valid card survives any perturbation of the surrounding text.

The system unions the two layers' spans before applying policy, so a token redacted by either layer is redacted in output. This is why full-system private-term recall exceeds the model alone on the structured classes the deterministic layer owns, while the model raises the ceiling on classes the deterministic layer cannot enumerate.

2.5 Policy

Every detected span carries a label. The policy layer applies **default-deny**: each label is redacted unless explicitly in the keep-set. The default keep-set is `{CITY, STATE, ZIP_CODE}` so an assistant can reason about coarse geography and eligibility while the precise street line (`BUILDING_NUMBER` + `STREET_NAME`) and secondary-address line (`SECONDARY_ADDRESS`) are redacted alongside names, identifiers, and contact information. The keep-set is a compile-time set (`KEEP_LABELS` in `src/types.ts`), not a runtime flag.

2.6 Session table

A client-only session table maps each detected raw value to a stable placeholder:

```
Maria Garcia    →  [GIVEN_NAME_1] [SURNAME_1]
555-11-2222     →  [SSN_1]
88 Pine Avenue  →  [BUILDING_NUMBER_1] [STREET_NAME_1]
```

The model provider sees only the placeholders. The user sees the restored response. Placeholders are intentionally formatted to look obviously synthetic so a model provider cannot easily de-anonymize them through downstream inference. The table is never transmitted.

3. Methodology

3.1 The candidate space

Model selection was not a one-shot grid. The shipped model is the product of a sustained, instrumented search: many fine-tuning runs across base architecture, corpus composition, label schema, vocabulary trim, prefilter strategy, and quantization, run over the project's lifetime. The eval harness (§3.4) was the instrument that drove it. Every run was scored end-to-end against the held-out set, and the corpus and schema were evolved in whatever direction the metrics said was working. The relabeling and corpus update decisions in §3.5 are themselves validated outputs of that loop.

One representative round of that search — the selection matrix detailed in §4 — varied four axes over their Cartesian product:

- **Base architecture (2).** MiniLM-L6-H384-uncased vs ELECTRA-small. ELECTRA is attractive in size (its trimmed Q4 artifact is ~10 MB to MiniLM's ~19 MB at full vocab) but, as §4.1 shows, did not imbibe additional data the way MiniLM did.
- **Prefilter (2).** Whether the deterministic layer's structured classes (`SSN`, `CREDIT_CARD`, `IP_ADDRESS`) are premasked before the model classifier runs or trained as real model classes (the no-prefilter ablation). Prefilter-on is the shipped configuration; the model never spends capacity on digits a checksum already settles.
- **Data mix (3).** Cumulative: the synthetic conversation corpus alone, then plus an OCR'd-document corpus, then plus the public AI4Privacy template corpus.
- **Corpus volume (2).** The first 100k vs the first 250k synthetic conversations.

That is $2 \times 3 \times 2 = 12$ data/architecture cells per base, scored under both runtime modes — a 24-model ablation, each cell exported to its Q4 and several vocab-trimmed artifacts. §4 reports it and the directional findings that came out of it; this section defines the rules under which every cell was scored. It is one round of the larger, ongoing search, not its entirety.

3.2 Pre-registered eval design

The eval design is fixed before a round's numbers are read. The governing principle is recall-biased and size-aware:

> Among candidates that clear the private-term recall floor on the held-out set, ship the smallest — unless a documented reason (multilingual coverage, adversarial robustness, a fairness regression) justifies overriding.

3.3 Datasets

Dataset	Rows	Use
OpenPII calibration	10k	Recall-floor threshold tuning
OpenPII held-out test (7 languages)	30k	Headline test (en, es, fr, de, it, pt, nl)
Per-language slices	—	English (11,569) and Spanish (3,234) reported separately; remaining five in the per-language table (§ model card)
Fairness	1,875	Faker × 15 naming traditions × 5 templates

All OpenPII splits are drawn deterministically from the held-out 100k partition of the OpenPII 1.5M corpus, disjoint from training. The shipped headline is measured across all seven supported languages; the per-language counts are the natural language distribution of that 30k slice (see model card).

3.4 Metrics

- **Private-term recall**: for every gold private value, did the redacted output contain the value? Wilson 95% CI; bootstrap 1000-resample CI for stratified breakdowns.
- **Public-term retention**: for every gold public value, did the redacted output preserve the value?
- **Span-level F1**: strict at IoU=1.0; relaxed at IoU≥0.5; overlap at IoU>0. One-to-one greedy matching, higher-scored predictions match first.
- **Latency**: Node.js ONNX runtime cold / p50 / p95 / p99 over the full 30,000-row test set.
- **Calibration**: 15-bin reliability expected calibration error (ECE), per label and overall, computed on per-span max-class scores.

Headline results, per language Full system (model + deterministic layer + policy) on the 30,000-row held-out test, scored end-to-end by the committed `eval/bench` harness:

Language	Rows	Private recall	Public retention	Leaks / private terms
English (en)	11,569	98.85%	90.5%	618 / 53,877
Spanish (es)	3,234	98.84%	91.6%	160 / 13,736
French (fr)	4,708	98.41%	92.8%	317 / 19,906
German (de)	4,260	97.94%	91.7%	357 / 17,347
Italian (it)	3,218	97.83%	94.1%	301 / 13,855
Portuguese (pt)	1,485	97.73%	92.5%	147 / 6,467
Dutch (nl)	1,526	97.21%	91.9%	182 / 6,519
All seven	**30,000**	**98.42%**	**91.69%**	**2,082 / 131,707**

Recall is term-presence (did every gold private value vanish from the output); retention is the policy-aware keep-set (city/state/ZIP). The seven-language aggregate carries a Wilson 95% CI of [98.35, 98.49].

3.5 Label schema and training-data design

The shipped model reflects two design decisions that postdate the candidate sweep above. Both are about *what* the model is asked to learn, not which checkpoint to ship.

Atomic label decomposition. Earlier iterations used coarse, pre-combined labels — a single `STREET_ADDRESS`, a single `PERSON`, plus `ORGANIZATION`, `LOCATION`, `DATE`, `AGE`, `INCOME`, and a catch-all `SECRET`. A model trained on those overfits to the easy case where a name or address arrives as one tidy, well-formed blob. The shipped schema instead forces everything to atomic pieces: names split into `GIVEN_NAME` / `SURNAME`; the street line into `BUILDING_NUMBER` / `STREET_NAME`; geography into `CITY` / `STATE` / `ZIP_CODE`; and document identifiers into their specific classes (`TAX_ID`, `BANK_ACCOUNT`, `ROUTING_NUMBER`, `GOVERNMENT_ID`, `PASSPORT`, `DRIVERS_LICENSE`) rather than a generic `SECRET`. This trains the model to recognize PII *fragments* in disordered

text — a building number on one line, a street name three messages later — instead of expecting a textbook one-line address.

Dates, ages, and income are non-PII. These were dropped from the redact-set entirely and map to 0 (kept context). A public-benefits assistant needs to reason about age and income to be useful, and a bare date is rarely identifying on its own; classifying them as redactable was over-redaction that hurt utility without a matching privacy gain. The keep-set proper is `{CITY, STATE, ZIP_CODE}`; dates/ages/income are simply not modeled as PII.

Premask train/serve symmetry. The structured classes the deterministic layer owns (`SSN`, `CREDIT_CARD`, `IP_ADDRESS`) are replaced with sentinel tokens *before* the model sees the text, both at inference (`src/premask.ts`) and during dataset construction . The model therefore never spends capacity learning to classify raw card/SSN/IP digits — those are a solved problem for a checksum — and the train-time and inference-time input distributions match by construction.

A deliberately noisy corpus. OpenPII supplies broad multilingual entropy, but its conversations are clean and well-formed. To keep the redactor from overfitting to tidy inputs, the synthetic portion of the corpus is generated to be messy and realistic on purpose: low-effort and typo-prone text, voice-dictated phrasing, values pasted out of forms, multilingual mixing, and contradictory, duplicated, or wrong-field entries, produced across a range of assistant personas so the user-side language varies. The aim is a model that catches partial, fragmented PII in real chatbot text rather than only in clean examples.

4. Alternatives we tried

4.1 Base architecture: MiniLM over ELECTRA

The first axis we settled was the encoder. ELECTRA-small was the strongest size contender — its trimmed, Q4 artifact is ~10 MB against MiniLM's ~19 MB at full vocab — which for a browser-deployed model is a real pull. We ran the entire selection matrix (§4.2) twice, once on each base, on identical data and schema.

ELECTRA did not turn extra data into accuracy the way MiniLM did. Its eval loss bottomed out early and then crept up as the corpus grew, and the matrix bears this out: every one of ELECTRA's strongest cells used the *smaller* (100k-conversation) data slice, and adding the larger slice produced its *worst* cells, not its best. MiniLM, on the same axes, reached its top result on the larger slice. For a project whose whole thesis is a corpus that keeps growing (§3.5), an encoder that stops improving — or regresses — as data grows has no headroom for the loop we rely on. We shipped MiniLM-L6-H384-uncased and kept ELECTRA on the shelf as a size lever for a hypothetical future release willing to accept that ceiling.

4.2 The selection matrix

To choose the data recipe and the prefilter strategy, we ran a Cartesian ablation rather than tuning one variable at a time. Twelve cells per base — **prefilter** {on, off} × **data mix** {synthetic-only, +OCR'd documents, +AI4Privacy templates} × **corpus volume** {100k, 250k synthetic conversations} — each fine-tuned, exported to its Q4 and vocab-trimmed artifacts, and scored *end-to-end through its matching runtime* (prefilter-on cells run with the deterministic layer; prefilter-off cells run the model raw). Across both bases that is 24 trained models, scored under both runtime modes.

Scoring used a deliberately hard internal development suite — 55 hand-written chat cases (66 private terms) skewed toward the failure modes we most wanted to catch: hyphenated and particled names, non-Latin-script names, address fragments, government identifiers, split and dotted contact details. It is **not** the shipped held-out metric (the 98.42% headline is measured on 30k OpenPII rows, §3.3); these are the much harder, much smaller dev numbers we used to rank *directions*, and the shipped model is many training iterations beyond this round. The matched-runtime results:

Base	Best cell (data mix / volume / prefilter)	Dev recall	Worst cell
MiniLM	AI4Privacy+Ai4Privacy / 250k / prefilter-on	**80.3%**	21.2% (no-prefilter, 250k synthetic, full mix — a training collapse)
ELECTRA	AI4Privacy / 100k / prefilter-on	81.8%	42.4% (no-prefilter, 250k, synthetic-only)

Three findings drove the recipe:

1. **The full data mix produced every top cell.** On both bases the highest-recall configurations folded in all three sources; synthetic-only cells trailed. Breadth of corpus, not volume of one source, moved the needle.
2. **MiniLM scaled with data; ELECTRA did not** (§4.1). MiniLM's best cell used the larger slice; all of ELECTRA's best cells used the smaller one.
3. **One cell collapsed.** The MiniLM no-prefilter / 250k synthetic / full-mix cell scored 21.2% — a reminder that more data and more classes can destabilize training, and the reason every candidate is scored end-to-end before it is trusted, never assumed good from its loss curve.

The prefilter-on configuration won on its merits here and matches the runtime: the deterministic layer is a checksum away from perfect on the digit classes, so making the model relearn them only adds variance.

4.3 Prefolded normalization

All training rows pass through the same normalization the runtime applies before tokenization: lowercase, NFKD decomposition, and combining-mark stripping. The combining-mark step is what folds accents — `José` becomes `jose`, `Müller` becomes `muller`, `François` becomes `francois` — so the model sees a single canonical form regardless of how the user typed the name.

We do this for two reasons. First, BERT's BasicTokenizer already performs the same fold implicitly at inference time under `do_lower_case=True` with default accent-stripping, so prefolding the training data makes the train-time and runtime distributions identical by construction; without it, the model would learn token sequences containing combining marks that the runtime tokenizer would never emit. Second, accent collapse is a robustness property we want: a user who types `Jose` and a user who types `José` should be redacted identically, and an attacker who substitutes one for the other to evade detection should fail. Prefolding bakes that property into the training distribution rather than relying on the runtime to recover it after the fact.

A guard in the training pipeline fails the run if a future tokenizer change breaks this assumption, so retraining with a cased base model cannot silently desync the two normalizations. Because prefolding already produces the canonical form, separate accent-augmentation (training on both

José and Jose as distinct strings) is disabled — it would be a no-op against an already-folded corpus.

5. Comparison to public PII baselines

During model selection we ran a one-off comparison of Rampart against public PII systems — a community BERT-small detector, Microsoft Presidio + spaCy, GLiNER small v2.1, and AWS Bedrock Guardrails — on the English+Spanish slice under identical scoring rules. On that slice Rampart reached 98.85% private-term recall; the open baselines trailed on retention in particular (GLiNER and BERT-small kept ~33.5% of public context), and the cloud incumbent trailed on both recall and latency. Also of note, those systems are Python- or cloud-only (Presidio, GLiNER/torch, Bedrock) and cannot run in the shipped TypeScript form factor.

6. Calibration

The runtime applies a single recall-biased confidence floor (`minScore` = 0.4) uniformly across the model's labels, chosen against the 10,000-row OpenPII Latin calibration split (disjoint from test). There is no per-label threshold table in the shipped runtime: classes the model alone is weak on — SSN, CREDIT_CARD — are not propped up with a tuned threshold but covered by the deterministic layer's checksum/structural validation, which is the system of record for them (with EMAIL, URL, and IP_ADDRESS covered by pattern match). Phone, routing, government-ID, passport, and license numbers carry no checksum and are left to the model under the same floor. Trading a miss (which leaks data) against the cheaper failure of over-redaction is the explicit policy choice here, not a model regression.

ECE on the 30,000-row test set is **0.018** for the model alone (well-calibrated, no post-hoc isotonic correction needed) and **0.291** for the full system. The system-level ECE is higher because the deterministic layer always emits score 1.0 on its detections, making the score distribution bimodal — that is a score-distribution artifact of the union, not a calibration regression of the underlying model.

7. Schema reconciliation

The 91.69% retention number on the headline test is term-presence scoring that already credits the keep-set (city/state/ZIP) as kept, matching the Rampart policy. We analyzed the 7,244 remaining "over-redacted" public terms in the 30,000-row eval:

- The vast majority are policy-driven redactions of street-line components (street name, building number, secondary address line). OpenPII marks `STREET`, `BUILDINGNUM`, and `SECADDRESS` as 0 (public); the Rampart policy redacts the precise street line (`BUILDING_NUMBER` + `STREET_NAME`) and `SECONDARY_ADDRESS` while keeping `CITY`, `STATE`, and `ZIP`. These are not detector errors; they are the policy firing as designed.
- A smaller share are span-edge artifacts. The runtime's particle-rescue step grows name spans (`GIVEN_NAME` / `SURNAME`) to swallow capitalized particles ("De la", "Von", "Mc"). When an adjacent public token is itself capitalized, that token can be absorbed into the redacted span.
- A very small fraction are digit fragments inside longer correctly-redacted spans (e.g. "376" found inside a redacted 16-digit credit card, surfacing as a "kept token" because the gold schema marks individual digits separately from the card number).

We publish the term-presence number for like-for-like comparison with public PII benchmarks running the same scoring rules. Under policy-aware scoring the retention exceeds 99%.

8. Systems we considered and did not adopt

Two widely-discussed alternatives are worth addressing directly because each is a plausible default for a team starting from scratch on this problem.

OpenAI Privacy Filter. OpenAI released an open-weight token-classification model under Apache 2.0 (announcement) with a similar shape to ours: bidirectional token classification with BIOES span decoding via constrained Viterbi, eight detection categories (person, address, email, phone, URL, date, account number, secret), and reported F1 of 97.4% on the corrected `pii-masking-300k` benchmark. It is the closest peer to Rampart in design intent, and we evaluated it as a candidate.

We did not adopt it for two reasons. (1) Size: the released model is 1.5B total parameters (50M active, MoE-style routing). Even with aggressive quantization the on-disk footprint is two orders of magnitude beyond our 15 MB browser-deployment target, and the active-parameter count alone exceeds what we can ship to a low-end device over the wire. (2) Inference shape: a 1.5B-parameter forward pass with a 128k-token context window is engineered for server-side or workstation-class throughput on long documents, not for sub-10 ms per-keystroke client-side redaction during a chat turn. The two systems are solving overlapping problems with different deployment constraints, and the OpenAI model's strengths (long-context coherence, fine-tunability for domain adaptation) are orthogonal to ours (browser-deployable size, encoder-only latency, per-class checksum validation through the deterministic layer). For applications that have a server they trust and documents that justify the round-trip, the OpenAI model is a strong choice; for an application whose contract is "the user's text never leaves the device," it is too large to deploy on the device.

`ai4privacy/llama-ai4privacy-english-anonymiser-openpii`. A Llama-family fine-tune trained on the same OpenPII corpus we used. The model is high quality on its native distribution and was a serious candidate. We did not adopt it for three reasons. (1) Size: the released artifact is multiple gigabytes, three orders of magnitude larger than our 15 MB target and incompatible with browser deployment on the low-end devices we need to support. (2) Inference cost: a Llama-class generative anonymizer takes hundreds of milliseconds to seconds per turn even on accelerated hardware, versus 6.6 ms for an encoder pass; running it on the client is not viable, and running it on the server reintroduces the threat-model problem above. (3) Generative outputs require a different correctness story — the model rewrites text rather than emitting spans, which makes calibration, span F1, and policy-driven keep-sets harder to define and audit. A token classifier with a deterministic post-processing layer is a much smaller surface area to reason about for a system whose contract is "do not leak."

Both systems are good engineering for their intended deployment shape. Neither is a substitute for client-side redaction when the goal is to prevent data from reaching any server in the first place.

9. Limitations

The model card enumerates each documented failure with statistics. The most consequential:

1. **Cross-locale name fairness.** Recall on Faker-generated names spans 15 naming traditions; non-Latin scripts (Korean, Han Chinese, Japanese, Arabic, South Asian, Slavic) are below 50%. This is the most important regression to close in subsequent training cycles and is tracked by a stratified regression test in the eval suite.
2. **Adversarial robustness.** The system catches 86.4% of a 20-case adversarial suite. Combined attacks (homoglyph plus whitespace-split, deep zero-width injection inside checksum-valid identifiers) can still bypass the union of the two layers. The deterministic layer raises the floor on structured classes but does not close the gap on unstructured identifiers under composed perturbations. This is the right framing for the limitation, not the primary use case: Rampart is designed to protect users who are entering their own information in good faith from incidental disclosure to downstream services, not to defeat a motivated user who is actively trying to smuggle their own PII past the filter. Adversarial cases are scored to characterize the failure surface and to surface regressions, not because circumventing one's own redactor is the threat model.
3. **Indirect identifiers.** Inferential leaks — e.g. a rare medical condition combined with a ZIP code — are out of scope. The system redacts terms, not statistical fingerprints.

10. Reproducibility

The training pipeline and the benchmark are released under CC BY 4.0. The benchmark (`eval/bench`) runs the shipped TypeScript pipeline over a frozen OpenPII held-out slice and writes the per-run `summary.json` / `by_language.json` that the published numbers cite, so every figure traces to committed evidence produced by the artifact itself. The held-out row `uids` are pinned in a committed manifest; `bun run bench:fetch` regenerates the rows and `bun run bench` reproduces the numbers.

11. Conclusion

Rampart is harm reduction. No client-side redactor at this size will catch every leak, and we do not claim otherwise — §9 documents the classes where the system underperforms and the eval suite is structured so future regressions surface immediately. What it provides is a defensible floor: text is filtered through two complementary layers and replaced with stable placeholders before any server sees it, so the worst case for a downstream leak is bounded by what the client failed to redact, not by the entire raw conversation.

We release the model, deterministic layer, and eval suite under CC BY 4.0 so any team building privacy-sensitive software can use, audit, fork, and improve it. The constraints adopted here — browser-deployable size, recall-biased calibration, defense in depth, no network dependency at inference — are specific to the threat model in which the user's unredacted text must never leave the device. Other deployment shapes warrant other tradeoffs; under this one, the system reported here is a deployable baseline against which future work can be measured.

References

AI4Privacy. 2024a. *llama-ai4privacy-english-anonymiser-openpii*. Https://huggingface.co/ai4privacy.

AI4Privacy. 2024b. *pii-masking-300k: A Benchmark Corpus for PII Detection*. Https://huggingface.co/datasets/ai4privacy/pii-masking-300k.

AI4Privacy. 2025. *pii-masking-openpii-1.5m: An Open Corpus for PII Masking*. Https://huggingface.co/datasets/ai4privacy/pii-masking-openpii-1.5m.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. "ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators." *International Conference on Learning Representations (ICLR)*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 4171–86.

Faker.js Contributors. 2024. *Faker: Generate Massive Amounts of Fake Data*. Https://fakerjs.dev.

Hugging Face. 2024. *Transformers.js: State-of-the-Art Machine Learning for the Web*. Https://github.com/huggingface/transformers.js.

Luhn, Hans Peter. 1960. *Computer for Verifying Numbers*. U.S. Patent 2,950,048.

Microsoft. 2021. *ONNX Runtime: Cross-Platform, High-Performance ML Inferencing*. Https://onnxruntime.ai.

National Design Studio. 2026. *Rampart: Client-Side PII Redaction for AI Assistants*. Https://huggingface.co/nationaldesignstudio/rampart.

OpenAI. 2026. *Introducing the OpenAI Privacy Filter*. Https://openai.com/index/introducing-openai-privacy-filter/.

Reimers, Nils. 2021. *MiniLM-L6-H384-uncased*. Https://huggingface.co/nreimers/MiniLM-L6-H384-uncased.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter*. arXiv:1910.01108.

Wang, Wenhui, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers." *Advances in Neural Information Processing Systems (NeurIPS)*.

Wilson, Edwin B. 1927. "Probable Inference, the Law of Succession, and Statistical Inference." *Journal of the American Statistical Association* 22 (158): 209–12.

Wolf, Thomas, Lysandre Debut, Victor Sanh, et al. 2020. "Transformers: State-of-the-Art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 38–45.

Zaratiana, Urchade, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. "GLiNER: Generalist Model for Named Entity Recognition Using Bidirectional Transformer." *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.