How much sensitive data gets past Velum. Measured, not asserted.

A privacy control is only as good as the data it misses. This report scores Velum's detection engine against a labelled gold set of 1,800 spans across nine languages, using the exact models the apps ship, and states the residual leak plainly, including the part that still gets through. Every figure is reproducible from committed data and hash-pinned models; the method is in the last section.

99.8%	0.07%	1,800	0
MUST-MASK F1, MAXIMUM TIER, 9-LANG AVG	RESIDUAL LEAK, MAXIMUM TIER	LABELLED SPANS · 200 × 9 LANGUAGES	NETWORK CALLS IN THE MASKING PATH

// WHAT WE MEASURED

Three numbers, defined.

Residual leak is the share of must-mask spans (people, contacts, national IDs, dates of birth) the engine missed, after the shipped policy keeps context types. This is the privacy number, and the one to drive to zero.

Must-mask F1 is the precision/recall harmonic mean over the types the policy is meant to hide. It excludes organisations and locations, which the policy deliberately keeps, so it reflects the privacy job rather than being depressed by intentionally-kept context.

Over-mask rate is predicted spans that hit no gold span, i.e. masking non-sensitive text. Tracked to keep the output usable; ≤1.4% at the Maximum tier.

// THE TIERS

Three intensities. You trade speed for the last of the leak.

Tier	Engine	Must-mask F1	Residual leak	p50 latency	Footprint
Standard	Regex floor, zero model	73.1%	37%	instant	none
High	Regex + on-device NER	99.0%	1.0% · ~2% ES/CA	~10 ms	bundled, offline
Maximum	Regex + full NER (GLiNER)	**99.8%**	**0.07%**	**~25 ms**	**~1.1 GB model**

Standard is the free regex floor: exact where it fires, blind to contextual names. High is the app default (its ~2% ES/CA figure matches the desktop Settings copy). Maximum adds the full model for the lowest leak measured.

// PER LANGUAGE, MAXIMUM TIER

Where the residual leak lives.

Leak is zero in seven of nine languages and 0.3% in the two the launch market cares most about. The remaining misses are always in the fuzzy, contextual types, a name scored just under threshold, never in a structured identifier.

Language	Residual leak	Must-mask F1
Spanish es	0.3%	99.7%
Catalan ca	0.3%	99.5%
English en	0.0%	99.3%
German de	0.0%	100.0%
French fr	0.0%	100.0%
Italian it	0.0%	100.0%
Turkish tr	0.0%	100.0%
UK English uk	0.0%	100.0%
US English us	0.0%	100.0%

// WHAT NEVER LEAKS

Structured identifiers are matched exactly.

These types are generated from their published checksum algorithms and matched deterministically. Across the whole gold set they scored **100% precision and 100% recall**, so the residual leak is never an account number, an ID or a token.

email · IBAN · credit card · national ID · phone · IP address · AWS / GitHub token · JWT

// THE HONEST CAVEAT

On messy real-world text, more gets through.

> The figures above come from a synthetic gold set. However carefully it is built, synthetic data is shaped like the thing that generated it. So we keep a second, held-out set of deliberately messy real-world text the generator never saw, and score against it too. Leak rises:

Chain	Must-mask F1	Residual leak
regex + on-device NER	92.0%	4.2%
regex + full NER	89.8%	8.3%

A directional check, not a headline: the held-out set is only 10 examples, so a single miss swings it several points. It exists to keep the synthetic numbers honest, and to state plainly that real documents are harder than clean ones.

// METHODOLOGY & REPRODUCTION

Reproduce every figure.

Gold set, committed & seeded

1,800 labelled spans, 200 per language across es · ca · en · de · fr · it · tr · uk · us. Generated by a seeded PRNG and committed to the repository, so the set is byte-identical on every machine.

No circularity

National IDs, IBANs and cards are minted directly from their published checksum algorithms, never brute-forced through Velum's own detector, which would guarantee a perfect score by construction. An independent oracle then re-checks that the regex packs recognise every committed ID: a real cross-check, not a tautology.

Hash-pinned models

Both models are fetched from public HuggingFace repositories and verified against a committed SHA256 pin; the benchmark uses the exact files the apps ship.

Role	Model	File · hash
Full NER (Maximum)	`onnx-community/gliner_multi_pii-v1`	`onnx/model.onnx` `sha256 7704865e414f2459…`
On-device NER (High)	`Xenova/bert-base-multilingual-cased-ner-hrl`	`onnx/model_quantized.onnx` `sha256 5b65139844be260b…`

Machine & matching

Apple M2 Max · 12 cores · 64 GB, CPU execution provider, Node v26.5.0. A single run, no variance bands; latency is machine-specific. Matching is span overlap with loose type, since label vocabularies differ across models.

The command

Fetch the two pinned models, then run the engine's own bench harness against the committed gold:

```
pnpm --filter @velumprivacy/core bench --combined --breakdown
```

The engine makes no network calls at runtime, enforced in the build by a static scan and a runtime network-block test.

---

velumprivacy.com · su.engineering

Velum's masking engine is source-available under the PolyForm Noncommercial License 1.0.0, published to npm as `@velumprivacy/*` on request, so security teams can read exactly what runs on their data. This report and its underlying gold set and pinned models are available for independent reproduction on request. Commercial licensing, pilots and procurement start at velumprivacy.com.

Figures generated 2026-07-23 from a single bench run against the committed multilingual gold set. Synthetic-gold results; held-out real-world results are stated separately above. This document supersedes the "directional until published" caveat in the enterprise brief.