

# InboxLens

*A Gmail sales-engagement & tracking toolkit competing on features with Mailsuite, MailTrack, Yesware, Streak, and Mixmax.*

By Niraj Kumar · [nirajkr.com](https://nirajkr.com) · May 2026

**TL;DR** A Chrome extension + backend that brings premium sales-engagement features — real-time open tracking, link & attachment analytics, AI-assisted compose, automated follow-ups, mail merge, and inline cal.com booking — into Gmail at near-zero infra cost. Built across evenings/weekends in 6 weeks; 25 active testers; 99%+ open-tracking accuracy; sub-1.2s p95 latency on open notifications.

## 1. The problem

---

Email is the most-used product on the planet and arguably the least-instrumented surface in a power user's day. A founder sends an investor update and has no idea if it was opened. A recruiter ships ten offer letters and can't tell which candidates are stalling. A job seeker fires off cold pitches into the void.

Commercial answers exist — **Mailsuite (formerly MailTrack), Yesware, Streak, Mixmax, HubSpot Sales** — but they bundle the read-receipt layer with enterprise sales suites. The cheapest tier with real-time notifications and analytics tends to start at **\$10-\$15 / user / month**; the genuinely useful tiers (sequences, mail merge, team dashboards) sit at **\$25-\$100**.

I built **InboxLens** to put the most-loved 80% of those features in one Chrome extension, design every feature with privacy as a first-class concern, and run it at a cost structure (\$8/month at 100 users) that an individual contributor can absorb out of pocket.

### Who it's for

- **Job seekers** sending cold pitches and follow-ups during a search.
- **Recruiters** managing candidate pipelines outside a heavyweight ATS.
- **Founders** sending investor updates and partnership cold-emails.
- **Sales reps and freelancers** running lightweight outbound (< 500 emails/month).
- **Anyone who treats email as a system, not a UI.**

## 2. Feature matrix vs. incumbents

---

The goal was not to out-feature an enterprise sales stack — it was to deliver every feature an individual power user actually uses, then add a few the incumbents don't.

Feature	InboxLens	Mailsuite	MailTrack	Yesware	Streak
Real-time open tracking	✓	✓	✓	✓	✓
Read receipts in compose (✓✓)	✓	✓	✓	—	Partial
Link-click tracking	✓	✓	Partial	✓	✓
Attachment-view tracking	✓	✓	—	✓	—
Send-later / schedule	✓	✓	—	✓	✓
Auto follow-ups (no-reply)	✓	Partial	—	✓	✓
Mail merge / sequences	✓	Partial	—	✓	✓
Templates & snippets	✓	✓	—	✓	✓
AI compose / smart reply	✓	—	—	Partial	—
Inline cal.com / Calendly slot	✓	—	—	Partial	Partial
Skip internal-domain tracking	✓	✓	—	✓	—
Multi-device open dedup	✓	Partial	Partial	Partial	—
Mobile vs desktop open split	✓	—	—	—	—
Per-recipient analytics	✓	Partial	—	✓	✓
Reply-quality scoring (AI)	✓ (beta)	—	—	—	—

**Why this matrix matters** InboxLens is at parity with Mailsuite on the read-receipt + tracking core, matches Yesware/Streak on outbound automation, and adds three things none of the incumbents do well today: AI-native compose, inline scheduling links, and reply-quality scoring.

## 3. Product flow

---

### 3.1 Onboarding — 3 steps, under 60 seconds

- **Install** from the Chrome Web Store. The extension lands a single banner at the top of Gmail: *“InboxLens is on — start composing.”*
- **OAuth** the minimum Gmail scope set: ``gmail.compose``, ``gmail.send``, and ``gmail.metadata`` (we do not request ``gmail.readonly`` on message bodies).
- **Pick defaults** in a one-screen settings panel: auto-track all outbound (default on), exclude internal domain, default cal.com link, notification preference (toast, email digest, or Slack DM).

### 3.2 Compose surface

Inside Gmail’s compose window, InboxLens injects three lightweight controls:

- **Auto-tracking pill** — green when on, can be toggled off per email. This is the *“Auto-tracking enabled / Tracks opens via invisible pixel”* affordance shown to the sender.
- **Read-receipt indicator** — the ✓✓ marker in the toolbar mirrors the WhatsApp metaphor: one tick on send, double tick when opened, blue when clicked.
- **AI compose** — sparkle button calls a server-side LLM endpoint, which generates a draft conditioned on the recipient’s last 3 threads (subject lines only, never body).

### 3.3 Tracking pixel and link wrapping

Two parallel mechanisms run on every tracked send:

- **Pixel.** A 1×1 transparent GIF served from ``https://t.inboxlens.io/{msg-id}.gif``. The endpoint logs the recipient hash, timestamp, device-class (from UA), city-level geo (from IP, never finer), and forwards a write to the events store.
- **Link wrapping.** Outbound URLs in the email body are rewritten through ``https://l.inboxlens.io/{msg-id}/{hash}?u={url}`` which logs the click and 302s the recipient to the original URL. Click latency budget: < 80 ms p95.

### 3.4 Open event flow

- Pixel hit → backend writes ``(msg-id, recipient, timestamp, device, geo)`` to Postgres and increments an in-memory counter in Redis.
- A **12-minute dedup window** suppresses preview-pane re-fetches and gmail-image-proxy double-counting — the single most common source of false-positive open counts in competing tools.
- Server pushes the event to the user’s extension over **Server-Sent Events**; a desktop toast notification appears within 1.2 s p95 of the actual open.

### 3.5 Dashboard

- Per-email cards show recipient, open count, last-opened, device split, link clicks, and attachment views.

- Bulk view filters by template, recipient, status, age. Power-user keyboard nav (j/k/o).
- A single AI-summary banner at the top: “5 high-priority threads waiting on you (opened, no reply ≥ 48 h).” Built on top of OpenAI tool-calling over the events table.

### 3.6 Automated follow-ups

- Configurable rule: if no reply in N days, draft a follow-up using the prior thread context.
- Default behavior is **approval-required** — the draft sits in the user’s inbox marked “InboxLens suggestion.” Power users can flip on auto-send with per-recipient guardrails.
- Hard cap: **3 follow-ups per recipient** ever, unless the user replies and resets. This is a feature, not a bug.

### 3.7 Mail merge & sequences

- CSV upload → preview the first 5 merged emails → schedule send window + per-merge variables and fallbacks.
- Send-rate limiter respects Gmail’s 500/day cap and applies a configurable jitter to avoid the spam-trap pattern of identical-spacing bursts.
- Per-recipient tracking: every send in a merge is a first-class trackable email in the dashboard.

### 3.8 Inline scheduling (an incumbent doesn’t do this well)

- Insert a cal.com or Calendly link into a compose with a single click. The link is rendered server-side as a **branded card preview** in the email, not a raw URL.
- When the recipient books, the event fires back into the dashboard as a “booked” signal — closing the loop from outbound → booked-meeting.

## 4. Architecture

---

### 4.1 System diagram (in prose)

- **Chrome extension (Manifest V3)** — content script injects controls into the Gmail DOM, background service-worker handles auth refresh and SSE.
- **Edge API** — `t.inboxlens.io`` (pixel) and `l.inboxlens.io`` (link wrap) run on **Cloudflare Workers**. Pixel is the single hottest endpoint in the system; running it at the edge keeps p95 < 25 ms globally and side-steps the cost of always-on AWS.
- **Core API** — Node.js + Express on AWS ECS Fargate behind CloudFront, talking to Postgres (RDS) and Redis (Elasticache).
- **Events store** — Postgres with monthly partitions on the events table. pgvector for the embedding store used by sequence-template suggestions.
- **AI** — OpenAI for compose drafts (4o-mini for cost) and reply-quality scoring; Gemini Flash for batch summarization. Prompt caching on shared prefixes.
- **Observability** — Sentry for errors, PostHog for product analytics, CloudWatch for infra.

### 4.2 Why edge for tracking endpoints

A pixel endpoint serves a 1×1 image and writes one row. It's the perfect Cloudflare Workers workload: tiny payloads, global low latency, and pay-per-request rather than always-on EC2. Running just the pixel and link endpoints at the edge drops the marginal cost of a tracked email from ~\$0.0002 to under \$0.00002.

### 4.3 Privacy as a system property

- **Email body is never stored server-side.** Only message-ids, the sender's hash of the recipient address, and event metadata.
- **Cookieless infrastructure.** No third-party cookies are set when the pixel or link endpoint is hit.
- **Skip-tracking domains.** Internal-domain emails are excluded by default; users can add more.
- **Recipient deletion.** Public `/forget`` endpoint lets any recipient remove the hash of their address from the events store. GDPR-compatible by design, not by retrofit.

### 4.4 Resilience to Gmail UI changes

Gmail rewrites its DOM frequently. The extension uses semantic selectors (data attributes and ARIA roles, never CSS class hashes), feature-detects each control on mount, and ships a "safe fallback" that disables InboxLens UI and surfaces a one-line warning when Gmail changes break the inject points. Mean time to recovery on a Gmail-side change has been < **4 hours** across the three Gmail UI refreshes since launch.

## 5. Tech stack & build choices

---

- **Extension** — Manifest V3, vanilla TS + a small render layer; Tailwind for the popup and settings panel. No React in the content script (keeps the bundle under 30 KB gz and avoids Gmail's frame-script-CSP quirks).
- **Backend** — Node.js, Express, Postgres, Redis. Drizzle ORM for type-safe queries; Zod for request validation.
- **Edge** — Cloudflare Workers (TS), Workers KV for hot path rate-limits.
- **AI** — OpenAI 4o-mini for sub-second compose; Gemini Flash 1.5 for nightly batch summaries; pgvector for template retrieval.
- **Infra** — AWS ECS Fargate + RDS + ElastiCache, GitHub Actions CI/CD, Pulumi for IaC.
- **Observability** — Sentry + PostHog + CloudWatch + Grafana on a single Coolify-hosted dashboard.

## 6. Early metrics

---

- **Build time:** ~6 weeks across evenings and weekends.
- **Test users:** 25 (peers, ex-Fitterfly colleagues, two recruiter friends).
- **Open-tracking accuracy:** > 99% in controlled A/B against Mailsuite over a 200-email run.
- **Time-to-first-open notification:** < 1.2 s p95.
- **Edge p95 pixel latency:** 23 ms globally.
- **Estimated cost to run at 100 users:** ~\$8 / month including OpenAI and AWS.
- **Bug surface area since launch:** 7 issues opened, 7 closed; MTTR on Gmail-UI breakage < 4 h.

## 7. Roadmap — what I'd build next

---

- **Outlook + Apple Mail** support (extension shares ~70% of the core; SMTP-side hooks differ).
- **iOS share extension** — send via InboxLens from any app on mobile, with the same tracking guarantees.
- **Team mode** — shared open / click analytics across a hiring team, with role-based visibility.
- **Reply-quality scoring** — LLM-based signal ("enthusiastic / neutral / declining") graded against a labeled set of recruiter replies. Early prototype shipped to 3 beta users.
- **A/B subject-line testing** — split mail-merge sends across 2-3 subject variants and report opens-to-replies.
- **CRM webhooks** — HubSpot, Pipedrive, and Notion as first-class destinations for events.

## 8. What this taught me

---

A side project is a license to make the calls a product team usually defers. A few that stood out:

- **Real-time event systems need careful dedup.** What looks like one open can be 4 fetches from preview panes, image-proxy prefetch, and a second device. The 12-minute dedup window is the difference between a useful product and a noisy one.

- **Privacy is product surface, not policy.** Every feature has a \*‘‘would I want this done to me?’’\* moment, and most of the time the right answer reshapes the feature rather than killing it.
- **The Gmail DOM is unstable.** Resilience to Gmail’s own UI changes is a feature, not a bug — the extension treats Gmail like a third-party API that can break weekly.
- **AI compose lives or dies on latency.** Above 800 ms, people stop using it. 4o-mini, prompt caching, and pre-warming the model are the difference between an adopted feature and a curiosity.
- **Edge changes the cost equation.** The single decision to move the pixel and link endpoints to Workers dropped the marginal cost-per-email by 10× and would have been invisible in a slide on day one.