delivr.ai

TECHNICAL FEATURE SHEET

JavaScript Identity Resolution Pixel

First-party browser tag. Resolves anonymous web traffic to hashed identity. Captures behavior. Zero plaintext PII leaves the page.

DEPLOY dynamic, per-pixel · Vanilla JS, no deps · ESBuild-minified + obfuscated

9	3	39	0	180d
EVENT TYPES	HEM HASH VARIANTS	MASKED-FIELD PATTERNS	PLAINTEXT LEAKS	IDENTITY PERSISTENCE

Identity resolution

- **Hashed email (HEM)** · SHA-256 of normalized email; never plaintext
- Emits **SHA-256, SHA-1, MD5** variants for partner match
- Normalized: trim + lowercase, RFC-pragmatic validation
- **First-party cookie ID** · 64-bit server-issued snowflake
- Multi-HEM per cookie; atomic dedup-merge backend-side
- IP, User-Agent, referrer captured server-side

Form & field capture

- Global **submit listener** · captures all form submissions
- Harvests field names, values, form id/name/action/method
- Email masked to ***** before leaving page
- **Password inputs always masked** · type-based fail-safe
- Blocklist masks card, SSN, bank, CVV `39 patterns`
- Values truncated to 100 chars; custom blocklist override

White-label & first-party

- Serve the pixel from **your own domain** · custom CloudFront host, not cdn.delivr.ai
- Fully **first-party** context; no third-party domain
- Per-org / project hostname mapping; tenants isolated by S3 prefix
- Host → project verified via tenant registry · unmapped hosts rejected (no spoofing)
- Dedicated white-label deploy lane; registry refresh every 5 min

Behavioral events · all 9

- **Page view** `page_view` · URL, referrer, title, viewport
- **All clicks** `all_clicks` · element, selector, text, x/y
- **Form submissions** `all_form_submissions` · masked fields + form metadata
- **Deep scroll** `deep_scroll` · threshold (default 50%)
- **File downloads** `file_downloads` · file-type link clicks (pdf, doc…)
- **Exit intent** `exit_intent` · top-edge mouseleave, once/page
- **Idle user** `idle_user` · inactivity timer (default 30s)
- **Copy** `copy` · copy event + selected text (≤200 chars)
- **Video engagement** `video_engagement` · play/pause/ended; id, src, time

Each type toggled per pixel via the `Active` flag.

Performance & loading

- Vanilla JS · **no jQuery, no external deps**
- ESBuild minify + obfuscate; per-pixel dynamic gen
- **Double-load guard** per pixel ID (idempotent)
- Multi-pixel isolation · independent SDK instances
- Auto-init (toggle via `data-auto-init`)
- Debounced scroll; image-beacon redirect chain

Privacy & protection

- **Zero plaintext email** · only HEM leaves the browser
- Client-side field masking before POST
- Backend PII classifier `46 patterns`
- Sensitive-field audit flags card / bank / password
- **Bot filtering** via User-Agent antibot engine
- Data-URL / sandboxed-iframe events suppressed

Cookie & storage

- First-party cookie set server-side, refreshed each request
- Backend store: cookie → HEM list `180-day TTL`
- **Cookieless mode** · opt-in per org, no cookie issued
- Detects **partitioned cookies** (Privacy Sandbox)
- Bots get no cookie / empty HEM
- No localStorage · cookie-only footprint

Browser resilience

- Cookieless fallback for **ITP / ETP** environments
- Partitioned-cookie path for Safari / Firefox blocking
- No fingerprinting · cookie-ID + HEM only
- **SPA-aware** · wraps pushState / replaceState
- Detects popstate + hashchange route changes
- Per-client abuse guard · 100/60s (backend autoscales)

Integration & config

- One-line snippet `GTM-ready`
- **GA4** · reads gtag client_id + session_id
- **Microsoft Clarity** · userId + sessionId
- Partner EID enrichment (444 / Lijit callback)
- Global + per-event params via SDK or data-attrs
- Config priority: JSON → data-attrs → window → script

Payload & transport

- `fetch` POST, JSON, `credentials:include`
- Event: type, data, pixel_id, organization_id
- HEM payload: SHA-1[], MD5[], SHA-256[], cookie_id
- Server redirect chain via **image beacon**
- try/catch boundaries; console-logged errors
- Global + event params merged into payload

DETERMINISTIC BY DESIGN

The identity flywheel

It starts with your ICP, sized and segmented (TAM/SAM/SOM) in **Delivr.ai Segmentation**. We serve ads to those known people, and because the pixel can be served through the ad server, every impression fires it on load and writes that identity to the device or browser. Ours resolves to the actual person, **deterministically**, not a guess from a cookie. The next time that user lands on any property running the pixel, **domain resolution lifts**. Each deterministic identity we seed raises match rates everywhere the pixel runs. The loop feeds itself.

ICP TAM/SAM/SOM · Delivr.ai Segmentation → Ad served to known person → Pixel fires on impression → Identity written to device → Visits any pixel property → Domain resolution ↑ → **Match rate ↑** ↻ compounds

Gregory Kotovos
Co-founder & Chief Evangelist
gk@delivr.ai