

Why Identity Sync Matters



Preserving end-to-end user journeys across subdomains, native apps, and webviews

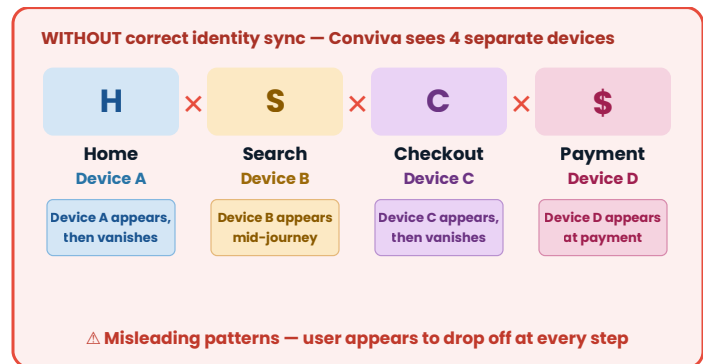
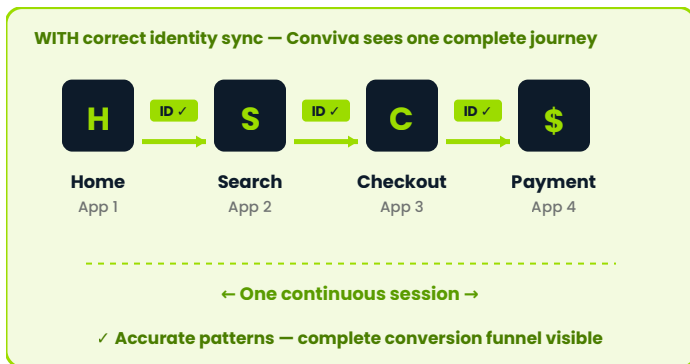
THE ROLE OF THE CLIENT ID

When Conviva's DPI sensor initializes on an app instance for the first time, it generates a **Client ID** — a unique, Conviva-owned identifier used to attribute events to a single, continuous user. It contains **no Personally Identifiable Information (PII)** and exists only to distinguish one user's activity from another's inside Conviva's analytics.

WHY IDENTITY SYNC IS CRITICAL

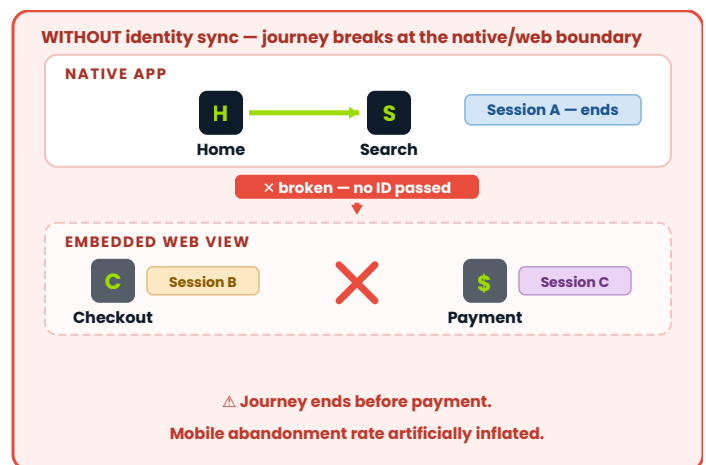
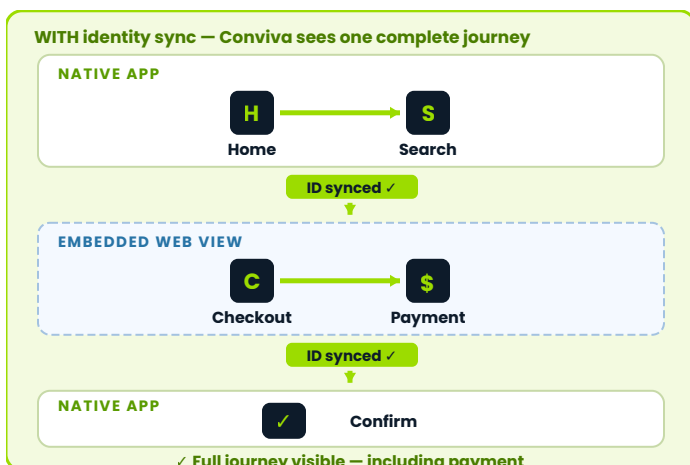
Fragmented Web Apps

Modern websites are often built as separate apps — Home, Search, Checkout, Payment. Without a shared identifier flowing between them, Conviva sees gaps: users appear to skip steps, drop off, or materialise mid-funnel from nowhere. Derived patterns are structurally inaccurate.



Hybrid Mobile Apps

Many mobile apps mix native screens with embedded web-based screens (typically for payment). These two environments use different identifiers by default. Without a deliberate bridge between them, Conviva loses the user at the native/web boundary — payment events disappear entirely and abandonment appears total across all mobile sessions.



Where Sync Is Required & How to Implement It

THE TWO SYNC METHODS

METHOD 1 · WEB ONLY

First-party cookies

Conviva stores the Client ID in a first-party cookie scoped to the parent domain. **As of JS SDK v2.1.0, this happens automatically — no configuration required.** Any subdomain that runs the sensor reads the same cookie and reuses the existing Client ID. If you prefer using Method 2, set `enableClientIdInCookies: false` to opt out.

Note: If your environment uses a WAF, API gateway, or backend security policy that filters or blocks unrecognized cookies, you may need to allowlist Conviva's Client ID cookie (`Conviva_sdkConfig`) to prevent it from being stripped or causing requests to fail.

METHOD 2 · WEB, NATIVE, WEBVIEW

`getClientId()` & `setClientId(clientId)`

The application reads the Client ID from the first context (`getClientId()`) and passes it into the second (`setClientId(clientId)`) before the sensor initializes. Works everywhere — the only option for native-to-webview handoffs.

Which should you use? If your product is purely web and all properties share a parent domain, first-party cookies are the zero-effort default — it just works out of the box. For anything involving a native app, a webview, or cross-domain handoffs, use the `getClientId()` / `setClientId(clientId)` APIs.

WHAT YOU GAIN WITH SYNC IN PLACE

- **Accurate unique user counts** across every surface of your product.
- **Complete funnels and conversion paths**, including the final payment or auth step in a webview.
- **Full experience patterns** — Conviva connects behaviors across subdomains and runtimes to the outcomes you care about.
- **Actionable diagnostics** — session timelines and replays span the full journey, not just a single context.