Referral Tracking — how it works, and the evidence

This document sets out how a partner's referrals are recorded and attributed (the mechanism), together with the results of actually running that mechanism (the evidence). If your concern is "I referred them — is it actually being counted?", this answers it with both the design and measurements.

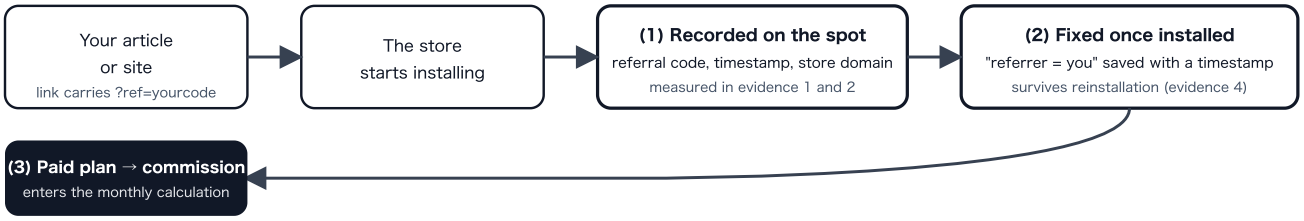
> **Our commitment**: the recorded data shown here becomes, unchanged, the line items on your quarterly commission statement. Numbers do not move on our side alone.

---

1. Attribution rules at a glance

Attribution follows the priority order set out in Article 3 of the Mumu Labo Partner Program Terms of Service. Where more than one method applies, **the earliest recorded timestamp wins.**

Priority	Method	What decides it	What gets recorded	Status
①	Referral link	An install arriving through your referral link (`?ref=yourcode`). Installs that route via the App Store are carried through by cookie	Referral code, timestamp, store domain	**Live**
②	Pre-registration match	Automatic matching against the email address and store domain given at pre-registration (email first, then domain)	Registered email, registered domain, registration date	**Live**
③	Onboarding survey	The store answers "where did you hear about us?" at install (not shown to stores already attributed by ① or ②)	The answer and its timestamp	**Live**
④	Manual declaration	An emailed declaration within 30 days of the referral, which we verify	The declaration and when it was received	**Live**



What is recorded reaches you unchanged, as the line items on your quarterly statement

The format of a referral code

- Permitted characters: alphanumerics plus `-` (hyphen) and `_` (underscore).
- Recommended shape: `partnername-channel-yearmonth` (for example `yourname-blog-202607`).
- **We issue and record the code.** Partners cannot choose their own, which is what prevents collisions and impersonation.

---

2. Verification (end-to-end test report)

These are the results of actually running the system to verify mechanism ① above. **One row is one verification**, recording "action → expected result → actual result".

#	What was verified	Action	Expected result	Result	Evidence
1	A referral code is accepted	Start an install from a link carrying `?ref=demo-media-202607`	The install proceeds and the referral code is recorded	✅	Evidence 1, 2
2	A malformed code is rejected	Access with an invalid referral code (symbols, whitespace)	The request is refused and **nothing is recorded**	✅	Evidence 3
3	The attribution record persists	Inspect the record directly in the database	Store domain, referral code and timestamp are stored	✅	Evidence 2
4	Continuous automated verification	Run the full automated test suite for attribution logic	All pass (install confirmation, priority order, protection on reinstall)	✅	Evidence 4

Why verification #2 matters: refusing any code that fails the format check is what proves **a third party cannot hijack your results or pollute the records with a fabricated code.** Your attribution is protected by it.

Evidence 1: accepting an install carrying a referral code (raw output)

```
$ curl -si "…/auth/install?shop=demo-partner.myshopify.com&ref=demo-media-202607"
HTTP/1.1 302 Found        ← accepted, redirecting on to Shopify's authorisation screen
```

Evidence 2: the attribution record stored in the database

```
SELECT shop_domain, ref, created_at FROM install_attributions;

demo-partner.myshopify.com | demo-media-202607 | 2026-07-11 21:41:02
(1 row)                    ↑ store          ↑ referral code  ↑ recorded at (UTC)
```

(Timestamps are UTC. That is 2026-07-12 06:41 Japan time.)

Evidence 3: rejecting a malformed referral code

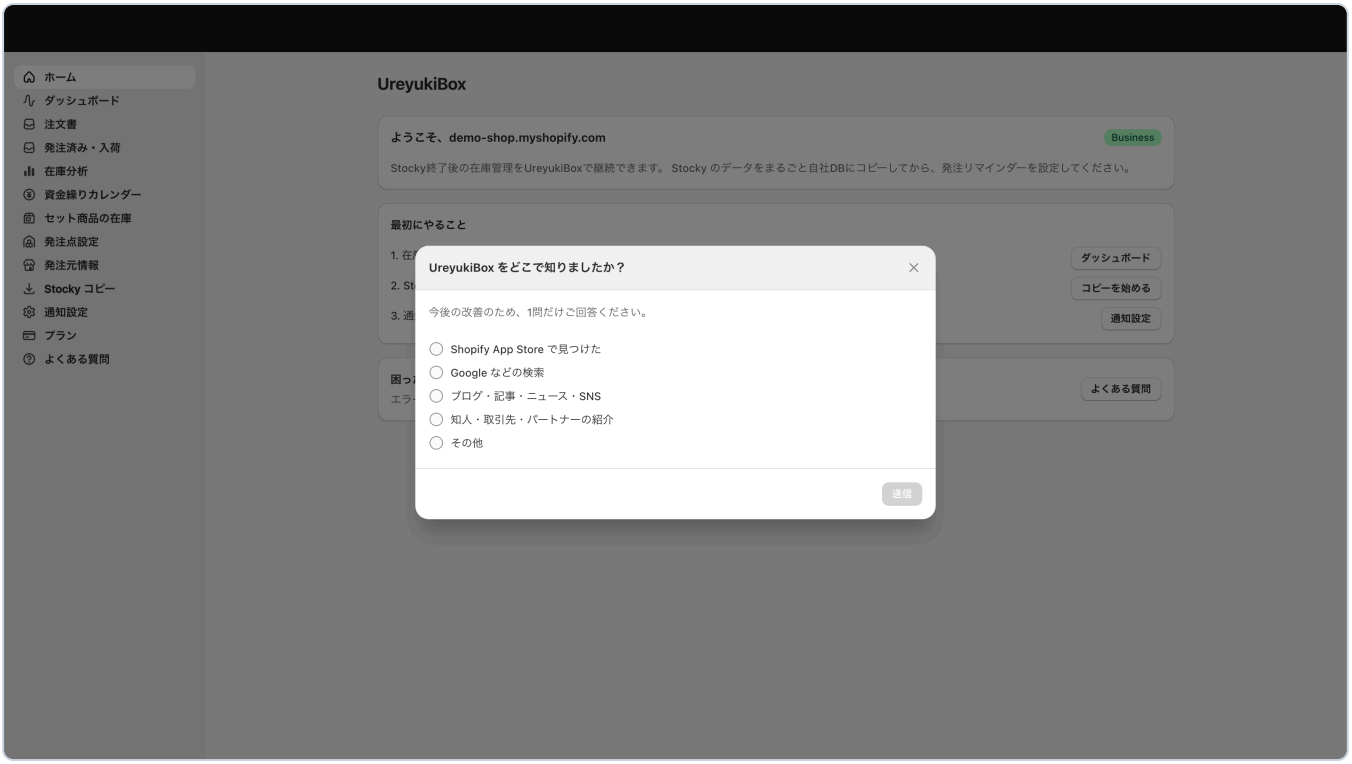
```
$ curl -si "…/auth/install?shop=demo-partner.myshopify.com&ref=bad%20ref!"
HTTP/1.1 400 Bad Request
{"detail":"invalid ref"}   ← refused with an error; no record is written
```

Evidence 4: automated tests (attribution logic — all 58 passing)

```
tests/test_install_attribution.py
  test_install_stores_valid_ref                    PASSED  ← valid codes are stored
  test_install_rejects_invalid_ref                 PASSED  ← invalid codes are rejected
  test_install_with_ref_sets_install_source        PASSED  ← the referrer is fixed on install
  test_install_ref_takes_precedence_over_waitlist  PASSED  ← priority ① over ② holds
  test_reinstall_does_not_overwrite_install_source PASSED  ← reinstalling does not clear the referrer
  …and 53 more
  ======================== 58 passed ========================
```

"Reinstalling does not clear the referrer" matters in particular: it proves that once your attribution is fixed, nothing the store subsequently does can lose it.

> **An honest note**: the evidence above was captured by running the pre-release app in a development environment on the same code as production. The onboarding survey (attribution ③) is also live, and that survey answers — a declared referral code — flow into partner commission aggregation is verified by automated test (`tests/test_partner_payout.py`).



3. How recorded data maps to your statement

What the app records automatically	Where it appears on the statement
Store domain	The "store domain" column of the line item
The attribution method that was decided (① to ④)	The "attribution method" column
When attribution was fixed	The "attribution timestamp" column (also used to adjudicate conflicts)
The referral code (for ①)	A note in the "attribution method" column
The pre-registration date (for ②)	Determines whether the early rate applies

At each quarterly payment we send you the statement built from these records, as data. For the statement format and a worked example, see the separate document **"Commission Statement"**. Questions or objections can be raised within 30 days of receiving a statement (Article 6 of the terms).