Incident Postmortem — Ledger Desynchronization (2024-10-14)

Severity: SEV-1 • Duration: 08:22–11:55 UTC • Detected by: Metrics (delta alert)

Impact: 2,314 ledger rows out of sync (missing/duplicated), 0 funds lost, 41 merchants delayed settlement reporting.

1. Executive Summary

A deployment introduced a race condition between the payouts ledger writer and the reconciliation importer. Under burst traffic, duplicate provider_ref rows were admitted, and the importer skipped later corrections, causing visible mismatches in merchant dashboards. Funds movement itself remained correct, but reporting and exports diverged. We remediated by backfilling, locking the writer around provider ref insertion, and adding a unique index with a partial filter.

2. Timeline

- 08:22 Alert “ledger_delta_gt_50” fired (payouts_sum vs provider_sum)

- 08:26 On-call acknowledged; throttled reconciliation importer

- 08:41 Identified abnormal duplicate provider_ref in payouts_ledger

- 09:05 Rolled back writer-2024.10.14-1 to previous build

- 09:18 Wrote migration to add UNIQUE(provider_ref, merchant_id) WHERE provider_ref IS NOT NULL

- 10:12 Ran backfill for T-1..T-0 – 8h; diff reduced by 97%

- 11:55 Verified zero delta for last 60 minutes; resolved

3. Root Cause

The writer change batched upserts without guarding against duplicated provider_ref during a short window when provider callbacks arrived out of order. The reconciliation importer used a “first write wins” heuristic and ignored later corrections if a row existed.

Contributing Factors

- Missing unique index on (provider_ref, merchant_id)

- Callback queue burst: retries collapsed due to narrow jitter, generating mini-storms

- Lack of end-to-end idempotency on reconciliation importer

4. Customer Impact

- Delayed settlement totals and “paid vs pending” graphs

- 18 support tickets (merchants suspected payout losses)

- No financial loss (balances correct); only reporting artifacts

5. Mitigation & Resolution

- Throttled importer to reduce write contention
- Rolled back writer build; added process-level lock around provider_ref writes
- Added unique index with partial filter for null-safety
- Replayed provider callbacks and ran targeted backfill

6. Action Items

ID	Owner	Description	Due
AI-01	Eng	Unique index (provider_ref, merchant_id)	2024-10-16
AI-02	Eng	Writer lock + test for duplicate callback; chaos test	2024-10-18
AI-03	SRE	Widen retry jitter; add queue burst alarms	2024-10-20
AI-04	FinOps	Reconcile T-7..T-0, attach diffs to Jira	2024-10-21
AI-05	PM	Merchant comms + RCA summary	2024-10-17

7. Technical Details

7.1 Schema

```
payouts_ledger(
  id PK, merchant_id, payout_id, provider_ref, amount, currency,
  state ENUM(created, submitted, succeeded, failed),
  created_at, updated_at, source_event_id
)
```

7.2 Faulty Upsert (before)

```
INSERT INTO payouts_ledger AS l (merchant_id, payout_id, provider_ref, amount, currency, state, source_event
VALUES ($1,$2,$3,$4,$5,$6,$7)
ON CONFLICT (payout_id) DO UPDATE
  SET state = EXCLUDED.state, provider_ref = COALESCE(l.provider_ref, EXCLUDED.provider_ref);
-- No uniqueness on (provider_ref, merchant_id)
```

7.3 Fixed Upsert (after)

```
-- Unique guard; allow NULL (provider might not return ref for failures)
CREATE UNIQUE INDEX CONCURRENTLY IF NOT EXISTS idx_provider_ref_unique
  ON payouts_ledger (provider_ref, merchant_id) WHERE provider_ref IS NOT NULL;

INSERT INTO payouts_ledger AS l (merchant_id, payout_id, provider_ref, amount, currency, state, source_event
VALUES ($1,$2,$3,$4,$5,$6,$7)
ON CONFLICT (provider_ref, merchant_id) WHERE provider_ref IS NOT NULL
DO UPDATE SET state = EXCLUDED.state;  -- never overwrite provider_ref with NULL
```

7.4 Importer Idempotency

Importer now tracks (source_event_id, provider_ref) and will re-apply corrections if a later event supersedes an earlier write.

8. Observability

- Dashboards: ledger_delta (provider vs internal), duplicates_per_min

- Logs: structured lines with payout_id, provider_ref, source_event_id

- Traces: span from callback receive → ledger write → importer recheck

9. Lessons Learned

- Enforce uniqueness at the boundary where external ids enter the system

- Avoid “first write wins” in eventually consistent imports

- Jitter windows must scale under bursty retries

10. Appendix

10.1 Backfill Script (excerpt)

```
-- Identify duplicates
SELECT provider_ref, merchant_id, COUNT(*) c
FROM payouts_ledger
WHERE created_at > now() - interval '7 days'
  AND provider_ref IS NOT NULL
GROUP BY 1,2 HAVING COUNT(*) > 1;

-- Delete surplus rows keeping latest succeeded OR latest updated_at
```

10.2 Merchant Comms Template

We identified a reporting error affecting settlement totals between 08:22–11:55 UTC on Oct 14. Funds movement remained correct; however, some dashboards showed incorrect totals. The issue is resolved and data has been backfilled. We’re adding additional safeguards to prevent recurrence. We’re sorry for the confusion caused.