

TLMS — Cross-border logistics platform

Region: Malaysia & Singapore **Stack:** Laravel · Vue · Redis queues · MySQL **Load:** Thousands of live shipments/day

Field apps were deadlocking the MySQL core on synchronous writes. We re-architected ingestion as idempotent Laravel queue workers with per-shipment locks.

LaravelVue.jsQueue workersPusherIdempotent jobs
Problem

Sync API convoy

Mobile/desktop scans hit REST → multi-table MySQL updates in one HTTP request. Peak freight hours caused row locks on shipments, sync backlogs, and dropped packets between field and core.

Architecture

Field → queue → core

API validates, enqueues, returns **202 Accepted**. Workers own joins, billing hooks, and cross-border status. Monotonic sequence per shipment prevents MY/SG drift.

Design decisions

Why queues

Option	Failure mode	Our fix
Sync REST	Deadlocks on batch scans	Single-writer job per shipment ID
Dual-write hubs	Cross-border drift	Event log + ordered propagation
SPA → DB	No back-pressure	Rate limits + scale workers on depth
Code shape		

API never blocks on billing

```
ProcessShipmentScan::dispatch($payload)
->onQueue('scans')->afterCommit();
return response()->json(['status' => 'queued'], 202);
```

// Worker: lockForUpdate on shipments, ordered table locks

Production metrics

Results

- Thousands of active multi-point shipments daily
- 100% reduction in MySQL deadlock errors on scan ingestion (90-day window)
- Telemetry across web admin, driver mobile, desktop sync

Niaga Prestasi / TLMS programme. Metrics from production monitoring.

contact@sevendyne.com · sevendyne.com/case-studies © 2026 Sevendyne Consultancy Services LLP