# Serapod2u QR Management System — Final Canvas (Build Spec v2.1)

**Authoritative build spec for AI code generators and engineers.**

This consolidates and de-duplicates all prior V2 docs. Anything from older drafts that conflicts with this spec is superseded here. Focus: clean, minimal, production-minded architecture with clear APIs, RLS, and acceptance criteria.

## 0) Executive Summary

Serapod2u is a QR/RFID-driven product movement and authenticity platform with B2B order flows and optional consumer campaigns. Core actors: **HQ**, **Manufacturer**, **Warehouse**, **Distributor**, **Seller** (new; under Distributor), **Shop/Kedai**, and **Consumer** (nonvape only when campaign-eligible).

Key behaviors:

* **Config-driven QR math** per Product Group (Units per Master, Buffer per 1000).
* **HQ→Manufacturer orders** with **L1/L2 approvals**, **PO render/send**, **payment gating** before batch generation.
* **Single QR per unit** serving dual purpose: **supply-chain tracking (ops scans)** and **consumer authenticity/rewards** (nonvape only).
* **Derived verify behavior** from group profile: *vape* ends at Shop (consumer path blocked, health warning page); *nonvape* allows consumer panel only when campaign-eligible.
* **Optional RFID (masters)** using GS1 SSCC-96 EPC (URI/HEX stored); QR fallback always present.

Non-functional: RLS isolation, idempotent ingest, offline queues, storage-backed assets (CSV/XLSX/PDF/SVG), WhatsApp/email notifications controlled **only** in HQ Settings.

## 1) Platform & Architecture

**Clients**

* Web Admin: **Next.js (App Router)**, TypeScript.
* Desktop PWA Scanner: camera/HID, IndexedDB offline queue.
* Mobile Scanners: **React Native (Expo)** with SQLite offline queue.

**Backend/Core**

* **Supabase**: Postgres (+RLS), Auth, Storage, Realtime.
* **Edge Functions** (Deno/TS) for secure writes, validations, QR/PIN/labels, PO render/send, notifications.
* Jobs/dashboards via Realtime/pg\_cron.

**Environments & Secrets**

SUPABASE\_URL = https://lovfzdtyestqnjoeafjh.supabase.co  
SUPABASE\_ANON\_KEY = eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImxvdmZ6ZHR5ZXN0cW5qb2VhZmpoIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NTQ4MjQ4NTIsImV4cCI6MjA3MDQwMDg1Mn0.I7ucCWq1kXpHmoisJRfnb5NlBjPNUPXINqMrLRl7f0c  
SERVICE\_ROLE\_KEY (CI only) = eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImxvdmZ6ZHR5ZXN0cW5qb2VhZmpoIiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc1NDgyNDg1MiwiZXhwIjoyMDcwNDAwODUyfQ.BtmIrVzCZvFa3h3zTGKByi4D9K2n6JnjTv5nXxorRmw

BASE\_SHORT\_URL=https://q.serapod2u.com  
WHATSAPP\_TOKEN (optional)  
SMTP\_HOST / SMTP\_USER / SMTP\_PASS (optional)

**Conventions**

* API base path: /api/v1/...
* Request headers: Authorization: Bearer <JWT>, X-PG-ID: <uuid> → edge sets select set\_config('app.pg\_id', :pg\_id, true).
* Responses use envelope: { ok:true, data, request\_id } or { ok:false, error:{code,message,field\_errors?,hint?}, request\_id }.
* Identifiers: **ULID** for masters/units; human line: SP<BRAND3><CAT3><YYYYMMDD><BATCH6><FAC2><PIN4> (printed on labels).

## 2) Roles & Access

**Product Group isolation** via RLS and user\_group\_roles:

* Roles per group: hq, manufacturer, warehouse, distributor, seller (new), shop.
* **Seller** is a child of **Distributor**: sellers inherit visibility from their parent Distributor but can only **sell to consumers** and **scan/verify deliveries**; they **cannot** create HQ or Manufacturer orders.
* Non-admin deletes are denied on master data (use status and sid guard).

## 3) Data Model (MVP scope)

All tables: id uuid pk, product\_group\_id uuid fk, status text, sid uuid, created\_at/created\_by, updated\_at/updated\_by.

**Master Data**

* product\_groups(code, name, workflow\_profile in ('vape','nonvape'), verify\_behavior derived)
* categories(code, name, requires\_regulatory)
* brands(code, name, category\_id)
* **3-tier taxonomy**: Product Group → Category → Brand (no Subcategory at MVP).
* products(code, name, category\_id, brand\_id, manufacturer\_id?, images, flavour, price\_cents(MYR), country\_of\_origin, weight\_g, option\_b\_fields for regulated)

**Ops / Commercial**

* manufacturers(code, name, contacts, address, country)
* orders(code, manufacturer\_id, status in ('pending\_approval','approved','po\_sent','payment\_notified','payment\_verified'), approver stamps, po\_code, po\_sent\_at, payment\_notified\_at)
* order\_approvals(order\_id, approval\_type in ('hq\_l1','hq\_l2','mfr\_payment\_verified'))

**QR & Labeling**

* qr\_settings(product\_group\_id, units\_per\_master, buffer\_per\_1000)
* batches(batch\_no(8d), order\_id, product\_id, total\_units, masters\_count, master\_id\_mode in ('qr\_only','rfid\_only','both'), label\_badge\_id?, label\_badge\_text\_override?, label\_badge\_apply\_to in ('unit','master','both'), production\_date, mfg\_facility\_code)
* masters(batch\_id, seq\_no, code ULID, epc\_uri, epc\_hex, quantity)
* units(master\_id, code ULID, seq\_in\_master, pin\_hash, pin\_salt)
* qr\_exports(csv\_url,xlsx\_url,pdf\_url,svg\_zip\_url, batch\_id)

**Inventory & Movement**

* scan\_events(idempotency\_key, role, action, device\_id, device\_seq, client\_ts, result)
* assignments(distributor\_id, created\_by\_role) + assignment\_items(master\_id, qty)
* warehouse\_bins (for move ops)

**Commerce & Docs**

* purchase\_order\_documents(order\_id, pdf\_url, ubl\_xml\_url?)
* invoices, invoice\_lines

**Notifications & Campaigns**

* notification\_settings(whatsapp\_enabled,daily\_cap,events)
* notification\_templates
* notification\_logs
* label\_badge\_catalog(name, default\_apply\_to)
* **Consumer Engagement (nonvape)**: points\_ledger, redemptions (for rewards); campaign eligibility flag per unit.

## 4) Workflows

### 4.1 HQ → Manufacturer (Order & Payment)

1. Create Order → pending\_approval.
2. Approve L1 (HQ A) → record hq\_l1.
3. Approve L2 (HQ B) → status approved.
4. Generate & Send PO → status po\_sent (PDF stored, sends via email/WhatsApp per HQ Settings).
5. Notify Payment (HQ) → status payment\_notified.
6. Manufacturer Ack Payment (optional) → record mfr\_payment\_verified → status payment\_verified.
7. **Generate Batch** (Manufacturer) allowed **only** after payment verified.

### 4.2 Manufacturer → Warehouse → Distributor → Seller → Shop

* **Produced** (scan in) → **In‑Warehouse** → **Assign to Distributor** (multi-scan Masters) → **Handover to Seller** (optional) → **Delivered to Shop**.
* RFID recommended for dock-door; QR always valid; scans are idempotent and role-scoped.

### 4.3 Distributor → HQ (B2B Orders)

* Distributor places order (for available/coming batches) → HQ approve/reject → Distributor sends D‑PO → HQ sends Invoice → Distributor acknowledges payment → Completed.
* Detail page shows **food-delivery style tracker** (for transparency).

### 4.4 Verify (Public)

* **Vape**: authenticity page + shop path only; consumer scans land on health warning page (no rewards).
* **Nonvape**: if **campaign‑eligible**, show login (Google/Facebook), then **Redeem/Earn**; else authenticity only.
* **Single QR** supports ops scans and consumer flows contextually (role-aware route, see §6 APIs).

## 5) APIs (v1)

**Shared**

* Envelope, headers, error codes (see §8). Helpers: ULID gen, idempotency key hash(pg\_id + code + device\_id + device\_seq), CSV/XLSX/PDF/SVG renderers, signed URL helper, HMAC-SHA256 PIN hasher (per-batch salt, never store plaintext PIN).

### 5.1 Groups & MDM

POST /api/v1/groups { code?, name, workflow\_profile } // server derives verify\_behavior  
POST /api/v1/categories { pg\_id, name, requires\_regulatory? }  
POST /api/v1/brands { pg\_id, category\_id, name }  
POST /api/v1/products { tangible attrs + option\_b; manufacturer\_id?; image token required }  
PATCH /api/v1/products/:id/status { pg\_id, status: 'active'|'archived'|'blocked' }  
GET /api/v1/catalog/products ?pg\_id=&q=&brand\_id=&category\_id= // returns only active

### 5.2 HQ→Manufacturer Orders

POST /api/v1/orders { pg\_id, manufacturer\_id, lines:[{product\_id, qty\_units, unit\_price\_cents}] }  
POST /api/v1/orders/approve-level-1 { order\_id }  
POST /api/v1/orders/approve-level-2 { order\_id }  
POST /api/v1/orders/:id/po/render // stores PDF (and optional UBL)  
POST /api/v1/orders/:id/po/send // email/WhatsApp per HQ Settings  
POST /api/v1/orders/:id/notify-payment  
POST /api/v1/orders/:id/ack-payment // optional

### 5.3 Batches (Manufacturer)

POST /api/v1/batches  
{  
 "pg\_id":"uuid",  
 "order\_id":"uuid",  
 "product\_id":"uuid",  
 "total\_units":10000,  
 "production\_date":"YYYY-MM-DD",  
 "mfg\_facility\_code":"MY",  
 "master\_id\_mode":"both",  
 "label\_badge\_id":null,  
 "label\_badge\_text\_override":"FREE DEVICE",  
 "label\_badge\_apply\_to":"unit"  
}  
// Logic: UPM=qr\_settings.units\_per\_master; B1000=buffer\_per\_1000  
// BufferUnits=floor(total\_units/1000)\*B1000; TotalUnique=total\_units+BufferUnits  
// Masters=ceil(TotalUnique/UPM); generate ULIDs; persist; render CSV/XLSX + PDF/SVG; return signed URLs

### 5.4 Scans, Inventory & Movement

POST /api/v1/scan { pg\_id, code\_type:'unit'|'master'|'rfid', code, actor\_role, action, device\_id, device\_seq, client\_ts }  
POST /api/v1/assign { pg\_id, distributor\_id, seller\_id?, master\_ids:[...] }  
POST /api/v1/move { pg\_id, master\_id, bin\_id }  
GET /api/v1/distributor-orders/:id/tracking

### 5.5 Distributor→HQ Orders

POST /api/v1/distributor-orders { pg\_id, batch\_id?, product\_id?, qty\_units, notes? }  
POST /api/v1/distributor-orders/:id/approve | /reject  
POST /api/v1/distributor-orders/:id/send-dpo  
POST /api/v1/distributor-orders/:id/send-invoice  
POST /api/v1/distributor-orders/:id/ack-payment

### 5.6 Verify (Public)

GET /verify/[code] // resolves ULID/EPC, role/context-aware  
// Vape → authenticity + shop path, consumer blocked (health warning page)  
// Nonvape → if campaign-eligible, show Redeem/Earn after login (Google/Facebook)

### 5.7 Notifications (HQ only)

PUT /api/v1/settings/notifications { pg\_id, whatsapp\_enabled, daily\_cap, events:{...} }  
PUT /api/v1/settings/notification-templates { pg\_id, templates:{ shop\_new\_batch\_coming: "... {{free\_gift\_note}}" } }  
POST /api/v1/notify/test { channel:'whatsapp'|'email' }

## 6) UI/UX Blueprint (Zero‑Data Prototype)

**Global Nav by role**: Dashboard; Product Catalog (Distributor, Shop); Orders (HQ→Manufacturer; Distributor→HQ); Batches & Labels; Warehouse Ops; Verify (public landing); Settings (HQ); Notifications; Label Badges; QR Settings; Master Data (Groups/Categories/Brands/Products).

**Products (HQ only)**: Grid with filters; status toggle Active ↔ Archived ↔ Blocked; images, tangible attrs.

**Catalog (Distributor/Shop)**: Active products only; badge chip (e.g., FREE DEVICE) when upcoming batch carries a label badge; CTA => Place Order (Distributor) / Contact Distributor (Shop).

**Orders**: Status Trackers with L1→L2→PO Sent→Payment steps; PO render/send dialog.

**Batches**: Generate after payment verified; show computed Masters Count from QR settings; download assets (CSV/XLSX/PDF/SVG).

**Warehouse Ops**: Receive, Assign (multi-scan Masters), Move (rack/bin).

**Distributor→HQ**: Detail page shows **food-delivery style tracker**.

**Verify**: Switch panel based on profile (Vape/Nonvape) and campaign eligibility.

**Movement Animation**: Visual tracker from HQ→Manufacturer→Warehouse→Distributor→**Seller**→Shop per batch; HQ can see all; Distributor/Seller see only their own batch journeys.

## 7) Security, RLS & Idempotency

* All writes occur in Edge Functions with the caller’s JWT forwarded to Supabase client.
* RLS: policy requires product\_group\_id match to set\_config('app.pg\_id') and role presence in user\_group\_roles.
* Scan ingest is **idempotent** via idempotency\_key and (device\_id, device\_seq).
* DELETE on master data is disabled; use status and sid guard.

## 8) Validation & Error Codes

**Global**: enums, ranges, presence checks. Option‑B fields enforced for regulated categories (Vape). Image required on product create.

**Standard errors**: VALIDATION\_ERROR, UNAUTHORIZED, FORBIDDEN, NOT\_FOUND, CONFLICT, PRECONDITION\_FAILED, UNPROCESSABLE\_ENTITY, RATE\_LIMITED, IDEMPOTENT\_REPLAY, INTERNAL\_ERROR.

Endpoint-specific examples:

* /batches: order must be payment\_verified; enums valid; cohesive pg\_id across payload.
* Distributor orders: legal state transitions only; wrong role ⇒ FORBIDDEN.

## 9) Acceptance Criteria (MVP)

* Product Groups: profile chosen → verify\_behavior derived and stored; persisted with guardrail dialog.
* 3‑tier MDM with images & tangible attrs; delete protection via RLS + sid.
* Orders: L1/L2 approvals; PO render/send; payment notify; manufacturer ack.
* Batches: buffer/split computed from QR settings; exports created; product **locked after first batch**.
* Scans: offline replay; idempotent; role-gated transitions.
* Catalog: Distributor/Shop see **only Active** products.
* Notifications: WhatsApp toggled/capped only in HQ Settings; logs written.
* **Seller role** present and constrained beneath Distributor; assign handover supports seller\_id.
* **Single QR** supports both ops scans and (nonvape) consumer redemption without changing the code.
* Movement tracker UI shows batch journey by role scope.

## 10) Seed & Dev Runbook (Local)

**Prereqs**: Supabase project; extensions pgcrypto, uuid-ossp; tools: node ≥20, pnpm, git, Supabase CLI, psql.

**Repo layout**

repo/  
 supabase/  
 migrations/  
 seed.sql // copy from seeds section  
 apps/admin/ // Next.js UI  
 apps/api/ // Edge Functions

**Run**

supabase db push  
psql "$DATABASE\_URL" -f supabase/seed.sql

**Next**

* Add one HQ user & one Manufacturer user to user\_group\_roles (by UUID).
* Login as HQ; create real Categories/Brands/Products and upload images.
* Create demo PO PDF and send; as Manufacturer, generate Batch and download assets.

**Safety**: never run seeds in production; all currency in MYR cents; replace placeholder images with signed Storage URLs in testing.

## 11) Versioning & Roadmap

* API version header: X-API-Version: v1. Breaking changes create /api/v2 with ≥6‑month deprecation window.
* Stretch (deferred): UBL 2.1 docs, Email-to-Inbox ingest, OCR, ERP webhooks, AR UI post-verify, ML anomaly detection.

## 12) Prompts for AI Coders (copy/paste)

**Bootstrap**

Generate a Supabase project schema and Edge Functions skeleton per this Canvas. Implement RLS policies using set\_config('app.pg\_id'). Provide a Next.js App Router admin UI with role-gated nav and zero-data prototype screens per §6.

**Work Orders**

1. Implement /api/v1/orders/:id/po/render with pdf-lib A4 layout and persist to Storage; return signed URL.
2. Implement /api/v1/batches with QR math, ULID generation, and CSV/XLSX/PDF/SVG exports; persist and return signed URLs.
3. Implement /api/v1/scan idempotent ingest with legal transitions by role; add movement tracker endpoints for UI.
4. Implement /api/v1/settings/notifications and /api/v1/notify/test with send caps and logs; WhatsApp gated by HQ Settings only.
5. Implement /verify/[code] Next.js route resolving ULID/EPC; block consumer panel for Vape; show campaign flow for Nonvape when eligible. Ensure **single QR** serves both ops scans and consumer redemption flows.
6. Add seller role under Distributor: extend assign to include seller\_id; add constraints and visibility rules; reflect in movement tracker.

**Testing**

* Unit: buffer math, masters\_count, currency formatting, envelope helpers.
* Integration: RLS scope via set\_config, product visibility by status, order approvals preconditions.
* E2E: /batches creates rows + assets; /orders/:id/po/render stores PDF; /verify/[code] switches correctly by profile and eligibility; movement tracker shows correct steps by role.

## 13) Notes & Guardrails

* Keep strings short & localizable later; avoid hardcoding verify logic in UI (server-derived).
* Badge ribbons & status chips must be re-usable across Orders and Distributor flows.
* Use clean, minimalist UI with rounded cards, ample spacing, skeleton loaders.
* Avoid CSV-based master-data creation at MVP; use UI for Product creation. (CSV/XLSX only for label/manifests exports.)

**End of Build Spec v2.1 (Aug 29, 2025)**