

NexGate — Commerce, Location & Logistics Master Guide

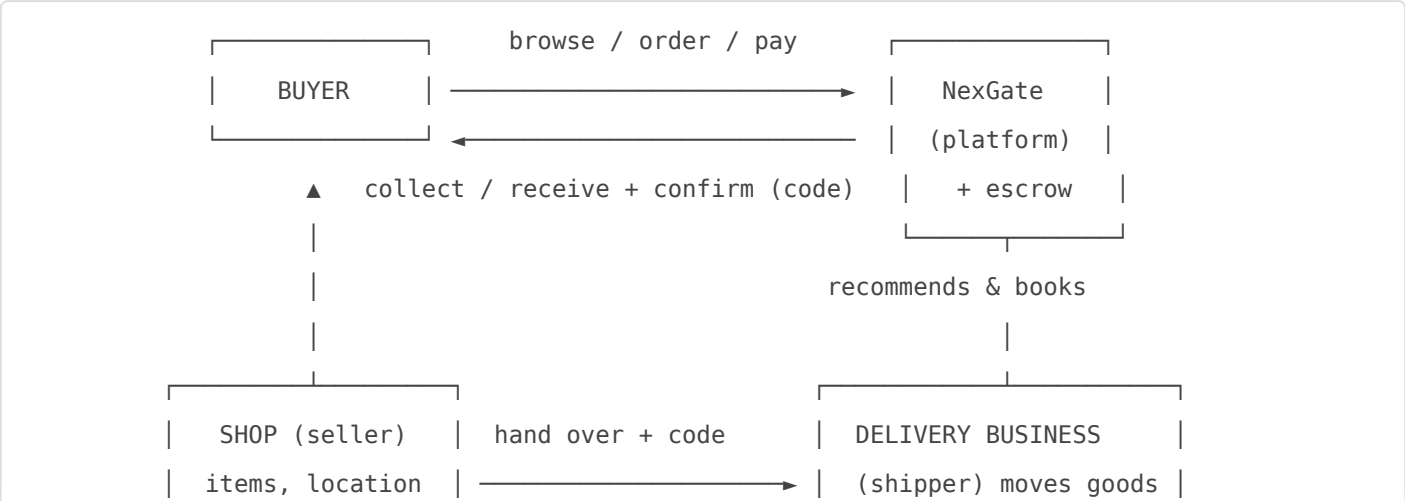
“ **Purpose.** This is the single reference for how buying, location and shipping work in NexGate. It captures every decision made so far and the reasoning behind each one, so that any implementer (human or Claude Code) can build from it, and future-you remembers *why*.

Companion file: `location-service-spec.md` is the detailed build spec for the Location Service. This guide is the wider picture; that spec is the first thing to build.

Guiding rule for everything here: simple over clever. No approval workflows, no features nobody asked for, no machinery a solo developer can't maintain. Where a simpler rule does the job, it wins.

0. The idea in one picture

NexGate is a social-commerce super app: users open **shops** to sell and **delivery businesses** to ship, and buyers order and get goods delivered or collect them. NexGate never owns stock or vehicles — it connects the three sides and holds the money safely in between.



Money path: Buyer → escrow → (on delivery code) → seller wallet + shipper wallet + NexGate fee

Everything in this guide hangs off two services:

1. **Location Service** — *where* things are. The foundation. Built first.
2. **Shipping Service** — *how* goods move. Built on top of Location.

1. Core principles (apply everywhere)

1. **The pin (lat/lng) is the single source of truth** for any place. Text is only a label.
2. **Location is collected only for a stated reason**, never on app open, never as background tracking.
3. **The administrative hierarchy (region/district/ward) comes from NexGate's own data**, computed from official NBS ward boundaries. Google is used **only on the client** for the map, place search and the live label — never called from the backend, never stored.
4. **Orders freeze a copy** of every address and price. Later edits never change a past order.
5. **Users own businesses; businesses earn into the owner's one wallet.** Shops sell, delivery businesses ship. Same ownership model for both.
6. **Money is held in escrow and released on a delivery code.** Proof of handover, not trust.
7. **One shipment = one pickup (one shop) → one shipper → one drop-off.** No multi-stop.
8. **Accept orders 24/7; be honest about timing.** Closed hours delay the work, never the sale.
9. **Shippers describe their own pricing and coverage.** NexGate never forces a pricing model.
10. **Trust is automatic, not a manual queue** — phone verify, limits that grow with good history, ratings, and auto-pause on repeated failure.

2. Location Service (foundation)

Full build detail is in `location-service-spec.md`. This is the summary.

2.1 What it owns

- Tanzania ward boundaries (NBS 2022) and **pin → region / district / ward** resolution
- **Shop locations** (one per shop, mandatory to publish)
- **Buyer saved addresses** (many, optional) and **recent locations**
- **Order address snapshots** (frozen copies)
- **Stations** and future **zones** (same shape-based pattern)
- Privacy rules for who sees which part of a location

2.2 How a pin becomes a place

<code>confirmed pin (lat, lng)</code>	
<code>/</code>	<code>\</code>
NBS ward shapes (ours)	Google (client only)
region, district, ward, codes	"Mlimani City, Sinza Morim..."
STORED (truth)	shown while placing the pin only
	(never sent to backend, never stored)

The lookup is a point-in-polygon test in PostGIS: *which ward shape contains this point?* It runs in milliseconds thanks to a spatial (GIST) index. If no ward contains the pin (beaches, lakes, borders), fall back to the **nearest ward within 500 m**; beyond that, the pin is "outside coverage".

2.3 The ward data (done)

- Source: **NBS 2022 Population & Housing Census — Tanzania Wards shapefile** (official, ~3 m accuracy).
- Already converted to **GeoJSON WGS84 (EPSG:4326)**, geometries repaired, delivered as `tz_wards_2022.geojson.gz`.
- **4,344 wards, 31 regions**. Primary key is `ward_id` (1–4344) because NBS codes are not unique.
- Loaded by a startup **seeder** (no Flyway, no version tracking) into PostgreSQL + PostGIS.
- The data file lives outside Git (`data/geo/`), mounted read-only into the container).

District and region need no separate data. Every ward row already carries its council, district and region names + codes, so one pin lookup returns all levels at once. Coverage checks and reports just use the stored codes. The only case needing a district/region *shape* is drawing an outline on a map — build it on demand by dissolving wards (`ST_Union(geom) GROUP BY reg_code[, dist_code]`), never store a second dataset.

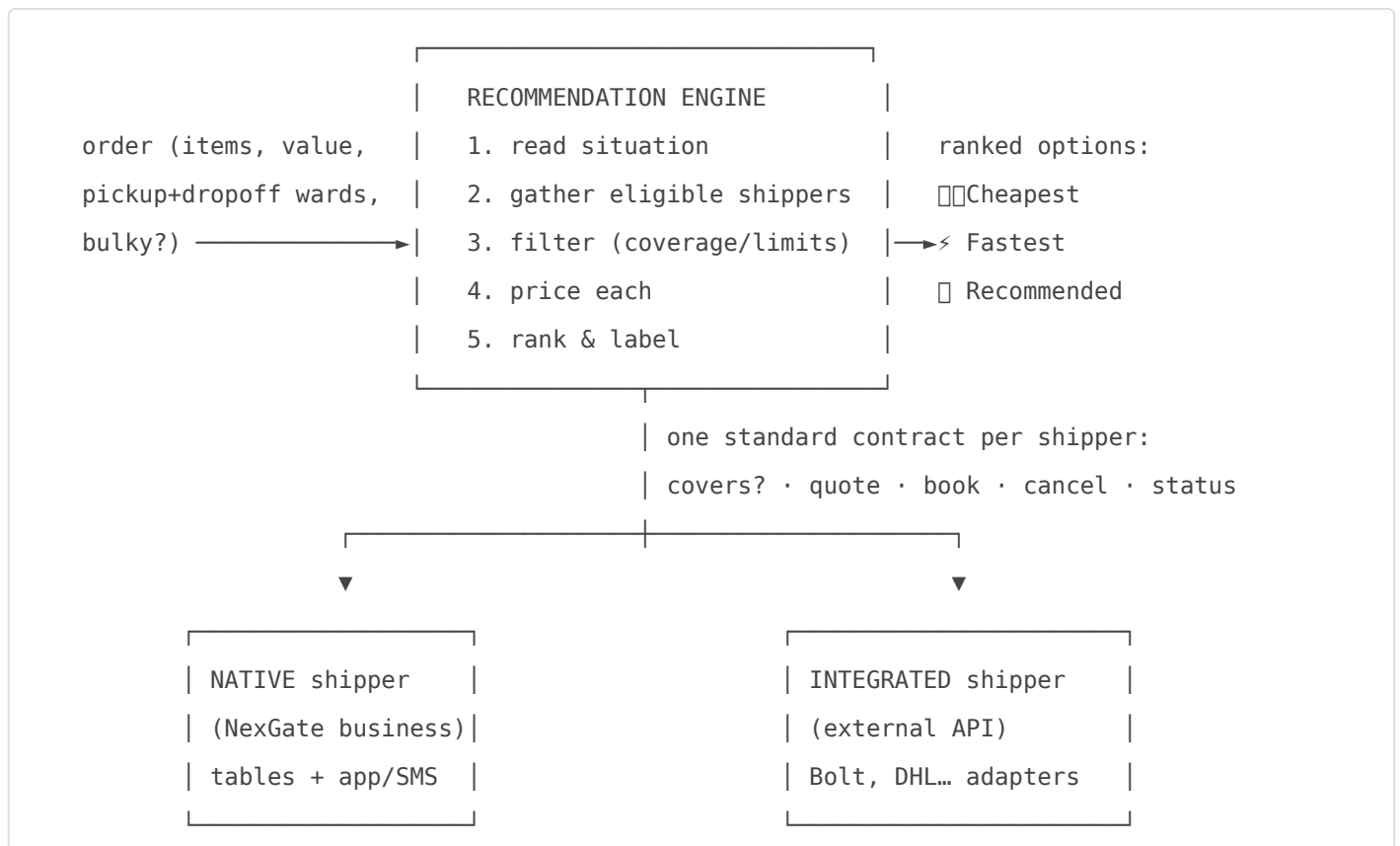
2.4 Address capture UX (same pin screen everywhere)

- **Never ask for location on app open.** Ask only when the user taps "Use my location".

- **Fixed centre pin, the map moves** under it (Bolt/Uber pattern). Satellite toggle. Accuracy circle.
- **Confirm disabled until zoomed to street level.** Search only moves the map, never confirms.
- **Shop:** pin + landmark + phone required (+ optional building, official address, photo). Mandatory.
- **Buyer:** pin + optional rider note only. Pin-only, Bolt style. Saving is optional; many saved addresses, plus recents.

3. Shipping Service (built on Location)

3.1 The whole shipping picture



3.2 Two classifications of every shipper

A shipper is described by **two independent fields**:

A. Service type — how goods move

Type	How it works	Coverage set by	Examples
------	--------------	-----------------	----------

☐ Door-to-door	Picks up at seller, delivers to buyer	Areas (ward/district/region, or nationwide) + optional max distance	Boda groups, Kibuti Express, DHL / EMS (wide area)
☐ Station	Seller drops at a station, buyer collects at a station	Stations (pins) + lanes between them	Bus companies, truck offices

“ "Local" vs "intercity" is **not** a separate type — it falls out of how wide the area is. DHL is just door-to-door with a nationwide area.

B. Connection — how NexGate talks to them

	Native (NexGate business)	Integrated (external API)
Set up by	The owner, in the app	NexGate, once, in code + admin
Pricing	Owner's tables	Live from their API
Jobs arrive	App notification + SMS/WhatsApp link	NexGate calls their API
Status	Owner taps buttons / link	Their webhooks or polling → mapped
Money	Into owner's NexGate wallet	NexGate pays them; they invoice NexGate
New one	Zero code (user signs up)	One adapter per company

They combine freely: a native door-to-door rider, a native station bus, integrated Bolt, integrated DHL.

3.3 One standard contract (why the engine never changes)

Every shipper, native or integrated, answers the same five questions. That's all the engine knows:

```
boolean covers(pickupWard, dropoffWard);    // area / lane check
Quote  quote(order);                        // price + ETA
Booking book(order);                       // returns a reference
void    cancel(ref);
Status  status(ref);                       // mapped to NexGate statuses
```

- **Native** answers from the owner's tables and app/link updates.
- **Integrated** answers by calling the company's API.

Standard status ladder (every provider's own statuses are translated into this, so the buyer's tracking screen never changes):

CREATED → BOOKED → PICKED_UP → IN_TRANSIT → AT_STATION → OUT_FOR_DELIVERY → DELIVERED
↘ FAILED / CANCELLED / RETURNED

3.4 Adding a provider = data, not code (for native)

```
[ - - - - - ]
Add delivery business
| Name:   Shabiby Bus Parcels   |
| Type:   ○ Door-to-door ● Station |
| Connect: ● Native ○ Integrated (API) |
| Contact: +255 7XX... (booking line) |
| _____ |
| Coverage |
| Door-to-door → pick areas |
| Station      → add stations + lanes |
| Pricing method ($3.6) |
| Limits: max distance / max value |
| Bulky: ○ No ● Yes (+extra) |
| Settlement: ● Wallet ○ Pay at station |
| [ ] Enabled |
L - - - - -
```

Integrated providers (Bolt, DHL) enter nothing here; their adapter handles it.

3.5 Coverage: picked from what the system already knows

Door-to-door → areas (from the region → district → ward list, or nationwide):

Mbeya Riders	□□ area: Mbeya district	→ local only
Kibuti Express	□□ area: Dar + Pwani	→ regional
DHL / EMS	□□ area: nationwide	→ intercity door-to-door

Offered only when **both** pickup and drop-off pins fall inside its area.

Station → stations + lanes (each station is a pin resolved to a ward, like a shop):

Shabiby Bus ☐☐

Stations: Mbeya Stand · Magufuli Terminal (Dar) · Dodoma Terminal

Lanes: Mbeya ↔ Dar · Mbeya ↔ Dodoma

Offered when it has a station near pickup **and** near drop-off. Buyer sees nearest stations first. Optionally a station shipper also delivers to the door around a station (then it also picks areas).

Routes with multiple stops on the way. A bus line usually stops at several towns in order (Mbeya → Iringa → Morogoro → Dar). The shipper enters **one route = an ordered list of its stations**, not every pair of lanes. The system then treats **any earlier stop → any later stop** on that route as a valid trip (both directions if the route runs both ways):

Route "Mbeya–Dar daily" (both ways)

1. Mbeya Stand 2. Iringa Office 3. Morogoro Stand 4. Magufuli (Dar)

→ valid trips: Mbeya→Iringa, Mbeya→Morogoro, Mbeya→Dar,

Iringa→Morogoro, Iringa→Dar, Morogoro→Dar (+ reverses)

4 stops → all pairs auto-derived. Add a 5th stop and its new pairs appear at once.

Pricing along a route (shipper's choice):

- **A. Per segment, summed** — one price per gap between neighbours; a longer trip is the sum (Mbeya→Dar = Mbeya→Iringa + Iringa→Moro + Moro→Dar). Simplest to keep accurate.
- **B. Stop-to-stop table** — a full price per pair, when it isn't a clean sum (e.g. a long-haul discount).

Plus the usual value add-on (insurance) and bulky extra. First/last mile to a buyer not near a stop is added as a separate local door-to-door leg, each with its own code — unchanged.

“ A **lane** is just the special case of a route with two stops. Model everything as routes.

3.6 Pricing — the shipper chooses the model, NexGate never forces one

At registration the shipper picks **one** method:

How do you charge?

- Flat price
- By distance (bands: 0–3 km, 3–7 km, ...)

- By package value (ladder: insurance – the shipper carries the risk)
- By distance + value (added together)
- Combined table (distance × value grid, each cell a full price)

Plus, on top: **bulky extra** (if accepted), **max distance** and **max value** (filters, only when set).

Why value-based for parcels: the shipper holds the insurance, so charging by the goods' value is fair, and "max accepted value" caps their risk. Quantity is handled for free — more items means a higher order value.

Distance is measured from the two pins: start with straight-line × 1.3 (free, instant), switch to a routing service later if needed without changing shipper tables.

Examples (5 km trip, phone worth 800,000):

Mbeya Riders	distance only	3,500
Shabiby (lane)	value only	20,000
Kariakoo courier	distance + value	3,500 + 3,000 = 6,500
Combined table	(3–7km × 500k–2M)	7,000
Small rider	flat	3,000

3.7 Package info — minimal

No size or weight. Only:

- **Order value** — from the cart, automatic.
- **Bulky? (yes/no)** — one checkbox on the product (category can pre-tick furniture, fridges). Shippers that don't accept bulky are filtered out. A fridge never gets offered a motorbike.

Vehicle type is not needed — "bulky vs not" is the only decision it would inform. Shippers may add a vehicle **badge** for display only (e.g. "Boda", "Pickup"); it drives no logic. Integrated providers (Bolt) pick the vehicle category inside their adapter.

Mismatch rule: if the real package differs (bulky not ticked), the extra cost is the **seller's** (they chose), the buyer's price never changes after payment, and repeated mismatches auto-correct the product's bulky flag.

3.8 Who enters what

Party	At setup (once)	Per order
Seller	Shop location, landmark, phone	☐ Bulky (usually none); tap "Ready for pickup"

Party	At setup (once)	Per order
Shipper	Coverage + one pricing method + limits + bulky	Tap link; update status
Buyer	—	Pick a delivery pin; tap one option (Cheapest/Fastest/Recommended)

The system works out the rest: value, bulky, pickup+drop-off wards, eligible shippers, price, ranking.

3.9 The recommendation engine, step by step

1. WHERE pickup ward + drop-off ward → same district / same region / different region
2. WHO shippers whose coverage (areas / lanes) includes both wards
+ build station + door-to-door combinations where needed
3. FILTER drop any that fail a hard rule:
 - area/lane not covered · over max distance · over max value
 - bulky needed but not accepted · closed & past cut-off (→ ETA shifts, not dropped)
 - provider currently auto-paused (repeated failures)
4. PRICE live quote (integrated) or the shipper's table (native)
5. RANK score by price · speed · reliability(history); weight by situation
(urgent → speed; cheap item → price; expensive → reliability)
label ☐☐Cheapest / < Fastest / ☐ Recommended; drop absurd outliers

Backup: if the chosen shipper fails (no pickup, cancels), auto-offer the next best and tell the buyer.

Reliability needs history, so at launch show only **Cheapest** and **Fastest**; add **Recommended** once real delivery data exists. Reliability is your long-term moat — competitors can copy features, not your data.

3.10 Shipper onboarding: register ? KYC ? go live

Shippers handle other people's goods and money, so onboarding has three parts. KYC gates **going live**, not filling the form — let them build the profile, then verify to switch on. Trust then grows automatically; **no manual approval queue** except on a flagged exception.

Register (6-step form) → KYC verify → GO LIVE (low limit) → limits & badge grow with history

Registration form (a short wizard, save-as-draft, admin can fill it for offline shippers):

Step	Fields	Feeds the engine as
1 Basics	name · service type (DOOR/STATION) · booking phone · WhatsApp? · logo	identity, notify, branches next step
2 Coverage	DOOR: areas (region/district/ward) or radius-from-base · STATION: stations + routes	coverage check
3 Pricing	method (flat/distance/value/dist+value/grid; route segment or table) · bulky yes/no (+extra)	quote
4 Limits & alerts	max value · max distance · won't- carry list · response window (default 30 min) · advance heads-up (default ON)	filters , job alerts (\$4.3a)
5 Hours & money	opening hours + cut-off · settlement (wallet / pay-at-station) · payout account	ETA/availability , money routing
6 Verify	phone OTP · agree shipper terms (liability up to declared value)	go live

Minimum to go live = steps 1–3; steps 4–5 have working defaults; step 6 is just verify. A one-man rider finishes in ~2 min; a bus company takes longer only because it adds stations.

KYC — fast, tiered, mostly automatic:

Tier 0 phone verified → register, build profile (not live)
Tier 1 + NIDA + ID photo + selfie match → GO LIVE, low max value (e.g. 200,000/order)
Tier 2 + payout name matches NIDA → higher limit
Tier 3 good history (N jobs, rating, low failures) → higher limit + "Verified" badge

- Auto-checks first (OTP, face-match selfie↔ID, name-match payout↔NIDA). Pass → live, no human.
- The **mobile-money number is itself a KYC signal** (already tied to a real NIDA in TZ).
- Manual review only on a flag (blurry ID, mismatch) — a small queue, not every shipper.
- **Sellers** get a lighter version of the same module (phone + NIDA + payout match before first payout).

Licences & compliance — collect now, confirm mandatory ones with a professional later.

Store each as *document type + file + expiry*. Mark required/optional per shipper type & tier once advised. Non-blocking at launch; uploading the real ones earns a **"Verified business"** badge.

Group	Documents (collect if the shipper has them)
Identity & business	NIDA · TIN (TRA) · BRELA (business name / company) · local-govt business licence · VAT (VRN) if over threshold

Group	Documents (collect if the shipper has them)
Transport / courier	LATRA licence (road carriage of goods/passengers — replaced SUMATRA) · TCRA courier/postal licence (door-to-door couriers) · TASAC (if freight-forwarding)
Vehicle & driver (own fleet)	vehicle registration · commercial/PSV licence · motor insurance (+ goods-in-transit) · driving licence · inspection/fitness
Employees (if any)	NSSF · WCF · OSHA
Money / insurance	mobile-money/bank payout · goods-in-transit / cargo insurance

“ 🚩 **Flag (not legal advice):** which licences are legally mandatory — especially **LATRA** for goods carriage and **TCRA** for courier services — and NexGate's own liability as the connecting platform, plus KYC/AML and escrow rules (Bank of Tanzania), must be confirmed with a Tanzanian lawyer / compliance professional. See §8.

4. Order & money flow

4.1 Checkout — pickup or delivery, per shop

Each shop in the cart is its own shipment, so the choice is per shop:

```
┌ ----- ┐
  Deliver to: 🏠Home, Mbezi Juu
├ 🏪Shop A · Kariakoo · 2 items ┤
│   ○ Pickup at shop           Free ┤
│   ● Delivery 🏪Kibuti Express 4,000 ▼ ┤
├ 🏪Shop B · Sinza · 1 item ┤
│   ● Pickup at shop           Free ┤
│   ○ Delivery ┤
├ 🏪Shop C · Mbeya · 1 item ┤
│   ● Delivery 🏪Shabiby       20,000 ▼ ┤
├ ----- ┤
└ Delivery total (2 deliveries) 24,000 ┘
└ ----- ┘
```

- **One shipment = one shop → one shipper → one drop-off.** No multi-stop, ever.
- A 3-shop cart = 3 shipments, each with its own shipper, price, codes and escrow release.
- **Pickup** also uses escrow + a code (buyer shows code at the shop; seller enters it).

4.2 Escrow + delivery codes (proof of handover)

```
sequenceDiagram
    actor Buyer
    participant NexGate as NexGate (escrow)
    participant Seller
    participant Shipper
    Buyer->>NexGate: pay (item + delivery)
    Note over NexGate: funds held in escrow
    Seller->>Shipper: hand over + SELLER pickup code
    Shipper->>NexGate: enter pickup code (proves shipper has it)
    Shipper->>Buyer: deliver
    Buyer->>Shipper: give DELIVERY code
    Shipper->>NexGate: enter delivery code
    Note over NexGate: release:
    NexGate->>Seller: item price - commission
    NexGate->>Shipper: delivery fee - commission
    NexGate->>NexGate: commissions
```

Code rules (decided):

- **Pickup code** (seller → shipper): proves the shipper received the parcel — matters if it's lost.
- **Delivery code** (buyer → shipper): releases the money.
- **Station collection:** buyer gives the code to station staff, who enter it. Same as door-to-door.
- **Multi-leg trips** (rider → bus → rider): **each handover has its own code**, so each leg is paid as soon as it's done — the first-mile rider isn't left waiting days for a small fee.

Open decisions to finalise (see §8): buyer unreachable / forgets code; item wrong or damaged after code; return window; cancellation cost.

4.3 Switching pickup ? delivery after ordering

```
Order placed as PICKUP → "Get it delivered instead"
    → pick drop-off pin → fresh delivery quotes (today's prices)
    → pay delivery fee only → into escrow → shipment created, seller notified
```

Allowed until the buyer has collected.

Order placed as DELIVERY → "Collect it myself"

→ allowed only BEFORE the shipper has picked it up

→ delivery fee refunded to the buyer

4.3a When the shipper is contacted — offer at "Ready", not before

The seller has **two separate actions**, and the shipper is only offered the job at the second one:

Order arrives → seller ACCEPTS (yes, I'll fulfil) ← shipper NOT contacted yet
→ seller PACKS
→ seller taps READY FOR PICKUP ← NOW the binding offer goes to the shipper

Offering at **Ready** (not at Accept) means: no shipper rides over to wait while the shop packs; if the seller can't fulfil, the order dies before any shipper was bothered; and the ETA/response-window start from a real ready moment.

Heads-up (optional, per-shipper setting — default ON, can be disabled). So shippers can plan without being booked early, NexGate sends a **non-binding** heads-up when the seller accepts/starts packing. It commits nothing and starts no timer:

On seller accept/pack → Heads-up (no commitment): "Order coming from Sinza, ~15 min. Be ready?"
On seller READY → Binding OFFER + response window (30 min) starts ← the one that counts

- The heads-up is a shipper **preference**: `Advance heads-up [✓] (default on)`. A shipper who finds it noisy turns it off and only sees the binding offer at Ready.
- **Re-check at Ready**: if packing ran long and the chosen shipper has since **closed** / **passed cut-off**, the engine re-evaluates at Ready — same shipper still open → offer it; otherwise fall into normal reassignment (§4.4). A slow pack never leaves the order stuck with an offline shipper.

4.4 Shipper rejects the job ? auto-reassign

A shipper can decline an offered shipment (too far, too busy, bulky they can't carry, value too high, closed). This must resolve cleanly, because it touches money. Two rules make it safe:

1. **Escrow holds two parts — item money and delivery money.** A rejection only touches the **delivery** part. The **item money stays held, and the seller never re-approves** — their job ended at "ready".
2. **The buyer's agreed price is protected.** The system can reassign silently only if it holds that price or beats it; a *higher* price needs the buyer's yes.

The flow:

Seller taps READY (\$4.3a) → shipment OFFERED to chosen shipper (delivery fee held)
→ shipper has a response window (e.g. 30 min, or until they next open)
└─ ACCEPT → pickup (code) → normal flow
└─ REJECT (reason) or TIMEOUT
 → that shipper's delivery fee released back in escrow
 → shipment = REASSIGNING, recompute options excluding rejecters

flowchart TD

```
A[Shipper rejects / times out] --> B{Next-best shipper?}
B -- none --> C[Buyer: switch to pickup refund, or cancel full refund]
B -- same price or cheaper --> D[Auto-book next · refund difference to wallet]
D --> E[Notify buyer: new courier, same price · NO action needed]
B -- costs more --> F[Delivery money to buyer wallet · show fresh options+prices]
F --> G[Buyer picks one · pays difference from wallet]
G --> H[New shipper acts]
E --> H
```

- **Same or cheaper → auto, buyer does nothing** (just a notification). The system walks **down the ranked list on its own** — shipper 1 rejects → shipper 2 auto-assigned → acts, and so on.
- **Dearer → buyer confirms** the new price (pays the difference from wallet, or gets a refund).
- **None left → buyer chooses** pickup (delivery refunded) or cancel (full refund).
- **Reason feeds reliability.** Chronic rejecters sink in the rankings and can auto-pause.
- **Loop guard:** cap auto-reassign at ~3 tries (or the order's handover window); after that, hand it to the buyer so a shipment never spins forever.

“ There is **no repeated approval** — the seller is out of it after "ready", and the buyer is pulled in only when the price would rise or nobody is left. Otherwise the system does the chasing.

5. Free shipping

Two funders, same mechanism. The shipper is **always paid in full**; the discount is covered by whoever offered it, deducted at escrow release.

5.1 Seller-funded (per shop or per product)

```
┌ - - - - - ┐
| Free delivery (shop setting)
| [✓] Offer free delivery |
| Where:  ○ Within [5] km of shop
|         ● Areas: Ubungo, Kinondoni |
| When:   order over [50,000] (opt.)
| I pay up to: [5,000] per delivery |
└ - - - - - ┘

Product override: ● Use shop setting ○ Always free ○ Never free
```

Uses your ward boundaries and pin distance to decide eligibility. The **cheapest qualifying option** becomes free; a faster choice costs the difference. The **cap** protects the seller (buyer pays the rest above it). In a mixed shipment, free only if **every** item qualifies.

5.2 NexGate-funded (campaigns)

```
┌ - - - - - ┐
| Campaign: Free delivery Dar weekend
| Areas: Dar es Salaam · Dates: 10–12 Oct
| Min order: 30,000 · NexGate pays up to 3,000
| Who: First order only · Shops: All |
| Total budget: 2,000,000 → auto-stops
└ - - - - - ┘
```

Stacking (seller first, then NexGate):

delivery price	7,000
– seller pays (cap 4,000)	–4,000
– NexGate pays (cap 3,000)	–3,000
= buyer pays	0

A **total budget** auto-stops the campaign — no runaway cost. Each shipment records **who paid what**.

6. Opening hours & orders outside hours

6.1 Hours apply to three things

Who	Why
Shops	Pickup availability; when a shipper can collect
Delivery businesses	Offered as "today" only when working; else ETA shifts
Stations	Where buyers collect parcels (often different hours from the company)

6.2 Hours editor (same screen everywhere)

Mon–Fri 08:00–13:00 · 14:00–18:00 ← several ranges per day (lunch break)
Sat 09:00–15:00
Sun Closed
Special days: 25 Dec Closed · 9 Dec 10:00–14:00 ← override weekly (holidays, Eid)
[] Temporarily closed (one-tap pause)
Same-day cut-off: 16:00 (shippers) ← after this, pickup = next opening

All times **EAT (UTC+3)**. Overnight ranges (18:00–02:00) allowed.

6.3 Orders outside hours — accept 24/7, be honest

Order at Fri 22:00, shop opens Sat 09:00
○ Pickup → "Ready from Sat 09:00"
● Delivery → "Arrives Sat ~12:00–15:00"

Rules:

- **Always accept orders.** Blocking a night buyer loses the sale.
- **Seller notified at once, but timers start at opening.** "Accept within 2h / ready within 24h" pauses while closed — a 22:00 order isn't overdue at 07:00.

- **Shipper jobs are scheduled, sent at the shipper's opening** (not midnight). Shipper may accept early.
- **Handover time** = first moment the **shop AND shipper are both open** (after "ready", before cut-off). **ETA** = handover + the shipper's normal delivery time, skipping their closed hours.
- **Temporarily closed** → seller chooses "still accept (ships when back)" or "stop orders (hide Buy)".
- **Safety net:** if the seller never acts within 24 *opening* hours → auto-cancel + refund from escrow.

7. What we learned from others (and what to copy / avoid)

Topic	Jumia / Glovo / Uber / Bolt do...	NexGate decision
Ordering when closed	Pre-order / scheduled	☐ Accept 24/7, show real time
Delivery choice	Door vs pickup station, options with price+ETA	☐ Per shop, Cheapest/Fastest/Recommended
Money safety	Escrow + 4-digit code	☐ Have it
Seller onboarding	Heavy docs + approval queue	☐ Avoid — phone verify + auto-limits
Surge pricing	Uber/Bolt dynamic	☐ Avoid — shipper ladders are clearer
Separate rider apps	Yes	☐ Avoid — NexGate deals with the business, not riders

To add from others (currently gaps):

1. **Problem-report window** — buyer can report a problem within ~24h of delivery before escrow releases.
2. **Two-way ratings** — rate the buyer too (bad address / no-show), so shippers aren't stuck with them.
3. **Cancellation policy with cost** — who pays when a buyer cancels after the shipper collected.
4. **Late-handover consequence** — feed seller late handovers into the same reliability score shippers have.
5. **Address quality nudges** — already helped by pin + landmark + ward check.

8. Open decisions to finalise before building Shipping

#	Decision	Suggested default
1	Buyer unreachable / forgets delivery code	Shipper marks "Delivered, no code" + photo → buyer 48h to dispute → else auto-release
2	Item wrong/damaged after code	Short report window (24h) before release, or code = final
3	Lost / damaged parcel	Shipper liable up to declared value; refund from shipper wallet; pickup code shows who held it
4	Cash on delivery (COD)	Prepaid only at launch ; add "accepts COD" per shipper later
5	Escrow legality	Check with a lawyer / payment partner (BoT, National Payment Systems Act) — <i>not legal advice</i>
6	Cancellation cost	Free before pickup; buyer bears shipper's cost after pickup
7	Bolt for parcels	RESOLVED (confirmed with Bolt): no parcel/delivery API in Tanzania — passenger rides only. Bolt is out as a shipper; rely on native shippers. Bolt only relevant if NexGate later adds an in-app passenger-ride feature (separate from shipping).
8	Mandatory licences & platform liability	Which of LATRA / TCRA / business licence etc. are required per shipper type, and NexGate's liability as connector — confirm with a Tanzanian lawyer / compliance pro (\$3.10)
9	KYC / AML level	How much identity NexGate must verify & keep for shippers and sellers holding escrowed money — confirm with payment partner / lawyer

9. Note on the Bolt Ride Booker API (reference)

Confirmed with Bolt (Oct 2026): no parcel/delivery API in Tanzania — passenger rides only. So Bolt is **not** a NexGate shipper. This section is kept only for a possible future **in-app passenger-ride** feature (separate from shipping). Prime Stack already has a Bolt **Business** account (ride billing), which is *not* the same as Ride Booker API access.

The API doc shared earlier is a **reseller passenger-ride** API, useful as a model for *integrated* providers but **not a parcel API**. Key facts if it's ever used (e.g. in-app rides):

- **Auth:** OAuth2 client credentials, 10-minute tokens, backend-only (never in the mobile app).
- **fare_id is fragile:** expires in 5 min, single-use, stops must match the estimate exactly. On `INVALID_FARE_ID`, re-estimate and retry.
- **Tracking:** poll ride every 10 s / driver every 1 s; webhooks live-only (not sandbox), send just `ride_id/type/state`, whitelist Bolt's 3 IPs.
- **States aren't linear** (can bounce `SEARCHING` ↔ `DRIVER_ON_ROUTE`); final price can change for 24h+.
- **Doc quirks:** garbled Cancel section, misspelled `reseler` destroy URL, `Fare_id` capitalised once.
- **Before use:** confirm Ride Booker is enabled for Tanzania, currency and payment setup (samples are Tallinn/EUR).

For NexGate's **parcel** shipping, native shippers matter far more than Bolt. Treat any Bolt/DHL integration as one adapter behind the standard contract (§3.3).

10. Build order (roadmap)

Phase 1 – Location Service ← START HERE (spec + data ready)

Docker PostGIS · seeder · pin resolution · shop location · buyer addresses · snapshots

Phase 2 – Native Door-to-door shipping, one city

Delivery business type · coverage (areas) · one pricing method · bulky flag

Escrow + pickup & delivery codes · prepaid only · status via app/SMS link

Checkout: pickup/delivery per shop · Cheapest + Fastest only

Phase 3 – Station shipping (intercity)

Stations + lanes · multi-leg codes · collect-at-station

Phase 4 – Recommendation engine grows

Reliability scores from history → add "Recommended"

Free shipping (seller + NexGate campaigns) · opening-hours scheduling

Phase 5 – Integrated providers

Integrated adapters behind the standard contract – e.g. DHL (confirm parcel APIs first).

NOT Bolt: confirmed no parcel API in TZ (rides only).

Cross-cutting (fold in as you go): two-way ratings, problem-report window, cancellation policy, late-handover scoring.

Do first: sign up your own first shippers (a few boda groups + one bus company) in one city — recommendations are worthless with no shippers.

11. Data model sketch (shipping — for when you build it)

“ Logical only. Follow the project's ID type, audit and error conventions. All areas/stations point to the **ward / district / region codes** from the Location Service — no new map data needed.

delivery_businesses	owner_user_id, name, service_type(DOOR STATION), connection(NATIVE INTEGRATED), adapter_name?, contact_phone, accepts_bulky, bulky_extra, max_distance?, max_value?, response_window_min(default 30), advance_headsup(default true), settlement(WALLET PAY_AT_STATION), enabled, rating, reliability
business_areas	business_id → region/district/ward codes it serves (DOOR)
business_stations	business_id, name, pin(lat,lng), ward_id, phone, hours (STATION)
business_routes	business_id, name, both_ways (STATION)
route_stops	route_id, station_id, stop_order (ordered list) → valid trip = earlier stop_order → later stop_order on same route (a 2-stop route is just a "lane")
pricing_rules	business_id, method(FLAT DISTANCE VALUE DIST_VALUE GRID), bands/ladder/grid as JSON
route_pricing	route_id, mode(SEGMENT TABLE), segment prices between neighbours, or stop-to-stop table (STATION)

opening_hours	owner_type(SHOP BUSINESS STATION), owner_id, weekly_ranges, special_days, temp_closed, same_day_cutoff
shipments	order_id, shop_id, shipper_business_id, service_type, pickup_snapshot, dropoff_snapshot (frozen), price, who_pays(buyer/seller/nexgate split), status(OFFERED ACCEPTED REASSIGNING PICKED_UP IN_TRANSIT AT_STATION OUT_FOR_DELIVERY DELIVERED FAILED CANCELLED), offer_expires_at, reassign_count, pickup_code, delivery_code, leg_no?
shipment_offers	shipment_id, business_id, state(OFFERED ACCEPTED REJECTED TIMEOUT), reject_reason?, offered_at, responded_at (one row per attempt)
shipment_legs	shipment_id, from, to, provider, status, cost, code (multi-leg)
escrow_holds	order_id, item_amount, delivery_amount, state(HELD RELEASED REFUNDED), breakdown (two parts: item + delivery)
campaigns	areas, dates, min_order, cap_per_delivery, budget, audience, active

12. Worked examples (engine walk-throughs)

“ Illustrative numbers only. Shows the 5-step engine (WHERE · WHO · FILTER · PRICE · RANK) end to end. Reminder of the rules: **value** → **price + insurance**, **bulky** → **vehicle/who can carry**, **quantity flows into value by itself**, hours & limits filter, area/route decides eligibility.

The registered shippers (set up once — used by all examples)

☐ Mbezi Boda ☐☐DOOR · Native
 Areas: Kinondoni, Ubungu (Dar) Pricing: distance
 0–3km 2,000 · 3–7km 3,500 · 7–15km 5,000 · max 15km
 Bulky: NO Hours: 07:00–20:00

- ☐ Dar Cargo ☐☐DOOR · Native
 Areas: whole Dar es Salaam region Pricing: distance + value
 dist: 0–5km 4,000 · 5–15km 7,000 value add: >500k +2,000
 Bulky: YES (+15,000, car) Hours: 08:00–18:00
- ☐ Shabiby Parcels ☐☐STATION · Native
 Route: Mbeya–Dar (both ways): Mbeya ▶ Iringa ▶ Morogoro ▶ Dar
 segment: Mbeya–Iringa 8k · Iringa–Moro 6k · Moro–Dar 7k
 value add: >1M +5,000 · Bulky: YES (+10,000) Hours: 06:00–19:00
- ☐ EMS Courier ☐☐DOOR · Native · nationwide
 Pricing: value ladder ≤200k 15k · ≤1M 25k · >1M 3%
 Bulky: YES (+5,000) Hours: 08:00–16:00 Mon–Fri

A — small but many, same city

20 phone cases, Sinza (Ubungo) → Mbezi (Kinondoni). Value **120,000**, not bulky, ~6 km, Sun 14:00.

- WHERE both in Dar region, ~6 km
- WHO ☐ ✓ ☐ ✓ ☐ x (no Dar-local) ☐ ✓
- FILTER value ok · not bulky · ☐ CLOSED Sun x → candidates ☐ ☐
- PRICE ☐ 6km = 3,500 ☐ 7,000 + value add 0 = 7,000
- RANK ☐☐ ☐ Mbezi Boda 3,500 ~40min · alt ☐ 7,000

Buyer sees: ☐☐Mbezi Boda – 3,500 · ~40 min. Quantity raised value, but it still fits a bike.

B — one but large (bulky, cheap)

1 mattress, same shops. Value **85,000**, **Bulky ticked**, ~6 km, Sun 14:00.

- WHO ☐ ☐ ☐ cover area
- FILTER BULKY required → ☐ NO x · ☐ closed Sun x → only ☐
- PRICE ☐ 7,000 + value 0 + bulky 15,000 = 22,000
- RANK ☐ only: ☐ Dar Cargo (car) 22,000

Buyer sees: ☐☐Dar Cargo (car) – 22,000. Cheap item, but bulky kept it off the bike.

C — intercity, multi-stop route, first/last mile

Laptop, shop in Mbeya → buyer in Morogoro town. Value **1,500,000**, not bulky, Mon 10:00.

```
1 WHERE    different regions (intercity)
2 WHO      ☐☐ don't reach Mbeya x · ☐ route: Mbeya(1) before Morogoro(3) ✓ · ☐ nationwide ✓
3 FILTER   value ok · not bulky · both open Mon 10:00
4 PRICE    ☐ Mbeya→Moro 8k+6k = 14,000 + value>1M 5,000 = 19,000 (collect at Morogoro Stand)
           ☐ EMS >1M = 3% of 1.5M = 45,000 door-to-door 2–3 days
5 RANK     ☐☐☐ Shabiby 19,000 tomorrow · ☐☐☐ EMS 45,000 to door 2–3 days
```

Buyer sees two groups:

```
☐☐Collect at station  Shabiby · Morogoro Stand (2 km) · tomorrow · 19,000
☐☐To your door       EMS · 2–3 days · 45,000
```

Route matched automatically; priced only the Mbeya→Morogoro segment, not the whole line.

D — shipper rejects ? auto-reassign (from §4.4)

From A, buyer picked ① **3,500**.

```
Seller "Ready" → offer ☐ → ☐ TIMEOUT (busy)
→ 3,500 back in escrow · REASSIGNING · exclude ☐
→ next = ☐ 7,000 → DEARER than 3,500
→ delivery money → buyer wallet, ask: "Mbezi Boda unavailable. Dar Cargo 7,000 (+3,500).
Continue?"
→ buyer accepts → pays 3,500 from wallet → ☐ gets the job
(if next were ≤3,500 → auto-assigned silently, no buyer action)
```

E — mixed cart, two shops (two shipments)

Cart: **phone** from Shop X (Kariakoo, Ilala, value 400k, not bulky) + **blender** from Shop Y (Mbezi, Ubungu, value 90k, not bulky). Buyer in Mikocheni (Kinondoni). Sat 11:00.

```
One shipment per shop (§4.1), priced independently:
Shipment 1  X Kariakoo → Mikocheni (~9 km)
  WHO ☐ (Dar region) ✓ · ☐ x (Ilala not in its areas)
  PRICE ☐ 7,000 + value 0 = 7,000
Shipment 2  Y Mbezi → Mikocheni (~5 km)
  WHO ☐ ✓ (Ubungo+Kinondoni) · ☐ ✓
  PRICE ☐ 3,500 (cheapest) · ☐ 4,000
```

Checkout shows:

Shop X · Dar Cargo	7,000
Shop Y · Mbezi Boda	3,500
Delivery total (2 deliveries)	10,500

Two shippers, two codes, two escrow releases. One slow shop never blocks the other.

F — free shipping (seller + NexGate stacked)

Shoes, Shop Z in Kinondoni → buyer in Kinondoni. Value **60,000**, ~4 km, Fri 15:00. Shop Z offers free delivery in Kinondoni, pays up to **4,000**. A NexGate weekend campaign in Dar pays up to **3,000**.

Cheapest qualifying option: Mbezi Boda 4km = 3,500

- seller pays (cap 4,000) -3,500 → fully covered by seller
- NexGate not needed
- = buyer pays 0

Shipment records: who_pays = seller 3,500

Buyer sees: Free delivery · ~35 min (badge). Shipper still paid 3,500 in full from seller's earnings. If the trip were 9 km = 5,000: seller 4,000 + NexGate 1,000 = buyer still 0, split recorded.

One-line summary

Build the Location Service first (spec + data are ready). Then layer shipping on top: shippers are businesses with a wallet, described by *service type* × *connection*, answering one standard contract; they set their own coverage and pricing; money sits in escrow and releases on a delivery code; orders are accepted 24/7 with honest timing; and free shipping, hours and ratings slot into the same engine without changing it.

Revision #1

Created 11 October 2026 14:31:15 by Admin Qbit

Updated 11 October 2026 14:32:16 by Admin Qbit