

The Horeca Digital Operations Playbook

How restaurant groups and independent venues consolidate reservations, guest transaction data, review routing, and SMS/loyalty automation into a unified operating system.

Unified Guest Memory

Consolidate preferences, reservation milestones, and transaction summaries in Supabase instead of third-party silos.

Automated Reputation Control

Identify critical dining reviews privately on-shift, routing positive responses to public maps while managing concerns locally.

Commission-Free Retention

Convert transactional diners into repeat loyalty guests using seasonal menus, automated text gifts, and local content.

01

The Fragmented Guest Relationship

Traditional hospitality operations suffer from "OTA dependency." Venues rent their guest relationships from reservation sites (like OpenTable) which charge heavy seat commissions, gate guest profiles, and promote competitive venues.

The gap is relational. Guest preference logs, transactional histories, and reservation states are scattered across point-of-sale systems (Lightspeed POS) and checkouts. Front-of-house staff cannot personalize the dining experience.

Fine-dining groups cannot deliver premium services if guest preference memory is fragmented across three disconnected software layers.

Consolidating reservation APIs, POS checkouts, and reputation nodes into a single relational database enables independent operators to bypass third-party margins entirely.

THE PLATFORM GAP

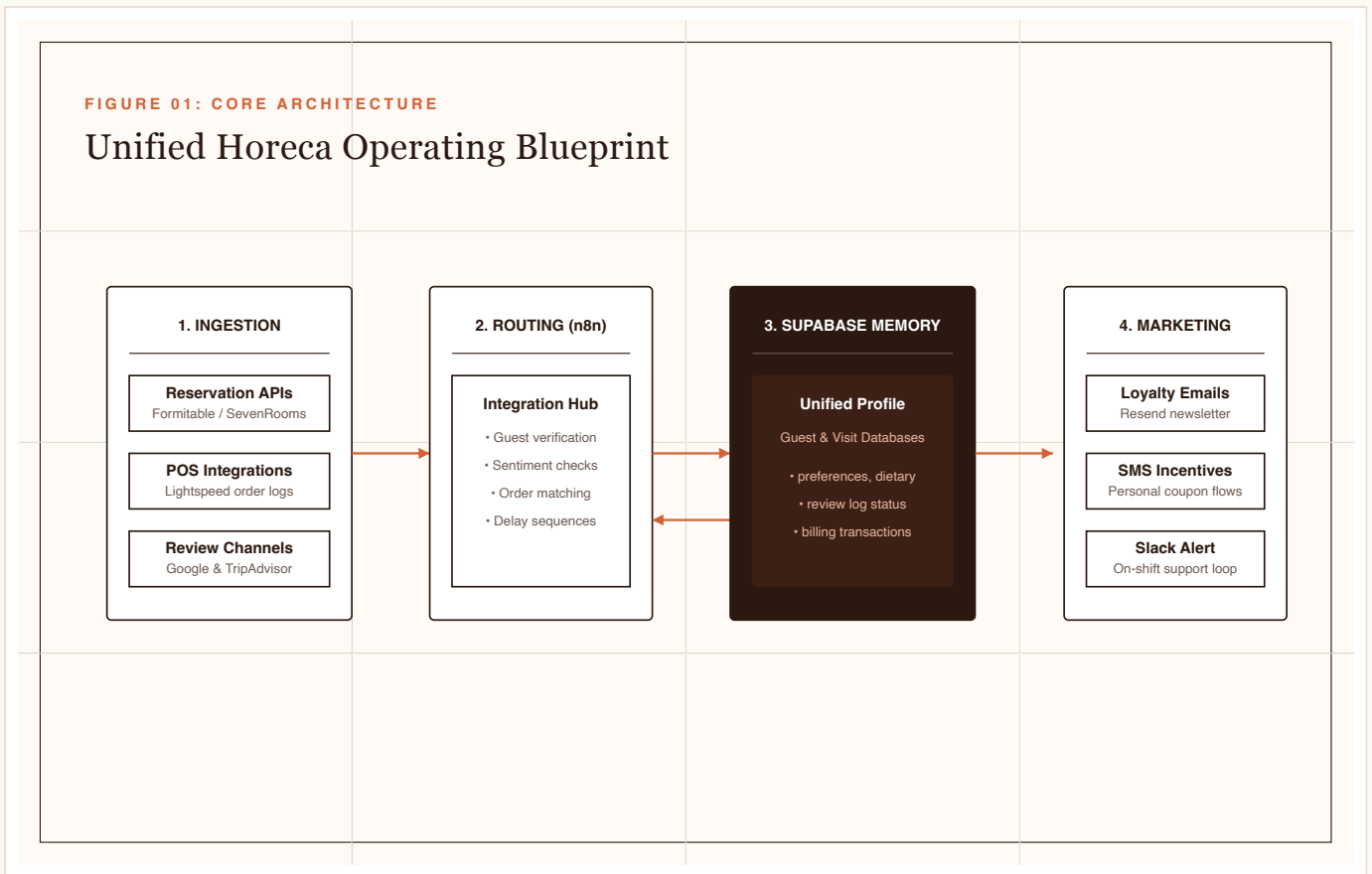
Traditional Setup

- OpenTable charges cover commissions.
- Lightspeed POS spend logs are completely disconnected from profile tags.
- Unhappy guests post straight to Google Maps before management is alerted.

Integrated System State

- Frictionless Formitable booking widget has \$0 commission.
- n8n maps receipt summaries directly to database profiles.
- Twilio routes low ratings to managers on shift.

Hospitality System Architecture



The operational data flow connects reservation webhooks, checkout transaction webhooks, and SMS reputation routes. n8n serves as the integration engine, verifying checkouts and matching tables in a master **Supabase** PostgreSQL database in real-time.

03

Centralized Horeca SQL Schema

Guest preferences are highly relational. A single visitor profile maps to many reservations, visit dates, and receipt line items (such as wine pairings or allergy exclusions).

Enforcing table constraints in PostgreSQL guarantees that dietary warnings trigger kitchen print notifications automatically, and links transaction totals to profile lifetime value attributes.

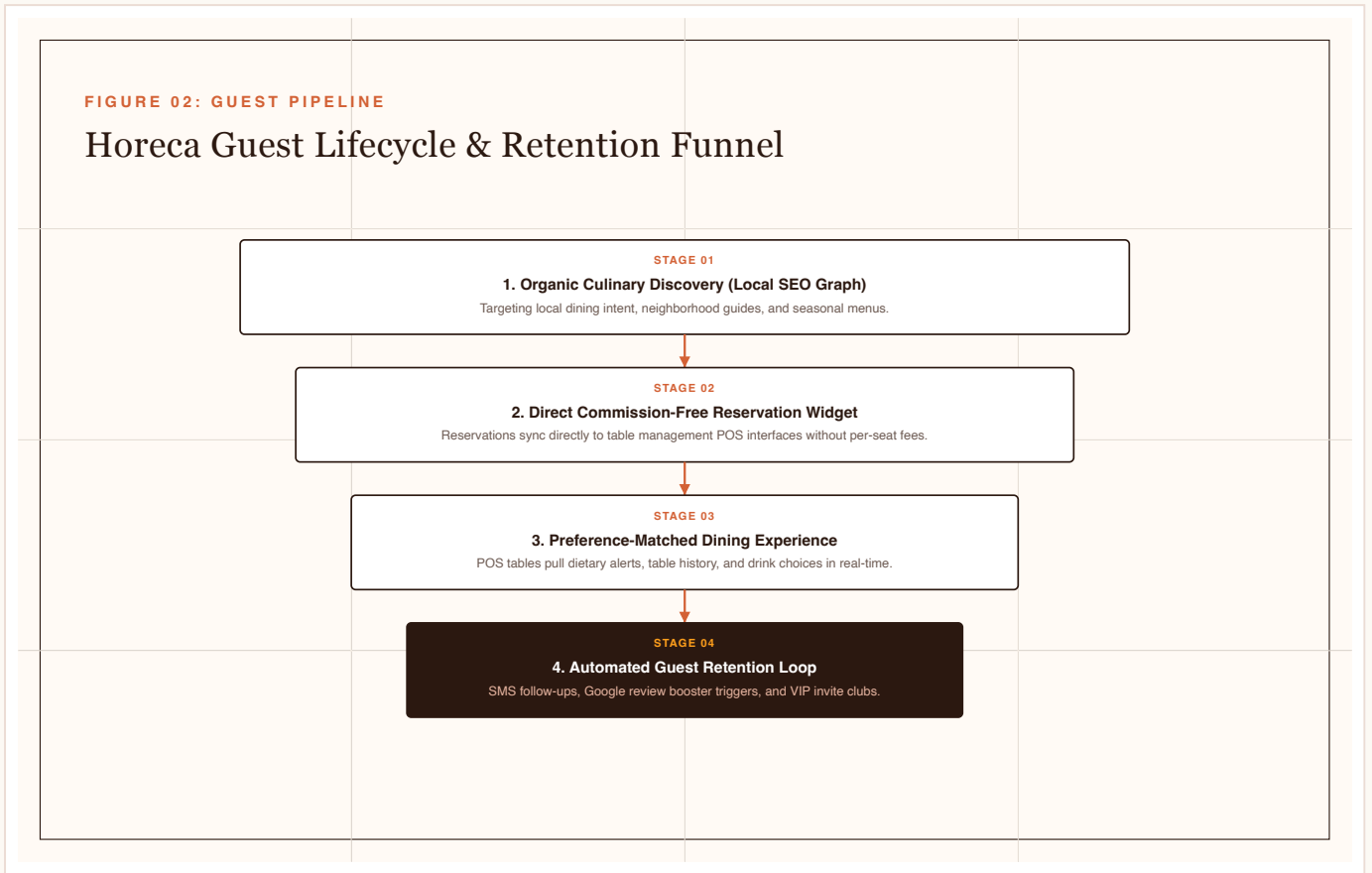
By using index arrays, floor terminals can query guest preference records in under 50 milliseconds.

POSTGRESQL TABLES

```
CREATE TABLE guests (  
  id uuid PRIMARY KEY DEFAULT  
  gen_random_uuid(),  
  email text UNIQUE NOT NULL,  
  first_name text NOT NULL,  
  last_name text NOT NULL,  
  preferences jsonb -- allergies, seating  
);  
  
CREATE TABLE checkouts (  
  id uuid PRIMARY KEY DEFAULT  
  gen_random_uuid(),  
  guest_id uuid REFERENCES guests(id),  
  pos_receipt_id text UNIQUE,  
  amount_spent numeric NOT NULL,  
  dine_date date NOT NULL  
);  
  
-- Fast lookup index for FOH terminals  
CREATE INDEX idx_guests_pref  
ON guests USING gin(preferences);
```

04

Guest Lifecycle & Retention Funnel



Converting single checkouts into recurring bookings requires a systematic journey. This model maps out guest acquisition, direct commission-free reservation engines, transaction profile matching, and loyalty campaigns.

05

Local Culinary SEO & Menu Graph

Most restaurants rely on PDFs for menus, which search engines cannot index. This destroys organic search visibility for high-intent queries (e.g., "gluten-free dining near me").

By structuring menu items inside a Headless CMS (such as Contentful or Webflow), each dish becomes an indexed node in your local SEO graph. n8n dynamically synchronizes these items to Google Business Profile, Facebook Local, and the website.

When a seasonal menu updates, the changes propagate globally, maintaining accurate search metadata.

CMS MENU SYNC SCHEMA

```
// Webflow CMS API payload mapping
{
  "collection_id": "menu_items",
  "fields": {
    "name": "Dry Aged Ribeye",
    "price": "45.00",
    "allergens": ["dairy-free", "gluten-free"],
    "active": true,
    "pairing_suggestion": "Barolo 2018"
  }
}
```

06

POS checkout Webhooks

When a transaction check is paid in Lightspeed POS, it emits a real-time webhook. n8n intercepts the payload, extracts the guest email from the table registration, and logs the spend metrics to the database.

This webhook script parses receipt lines to detect preference trends (e.g., guest ordered red wine three times), updating guest tags automatically for future targeted menus.

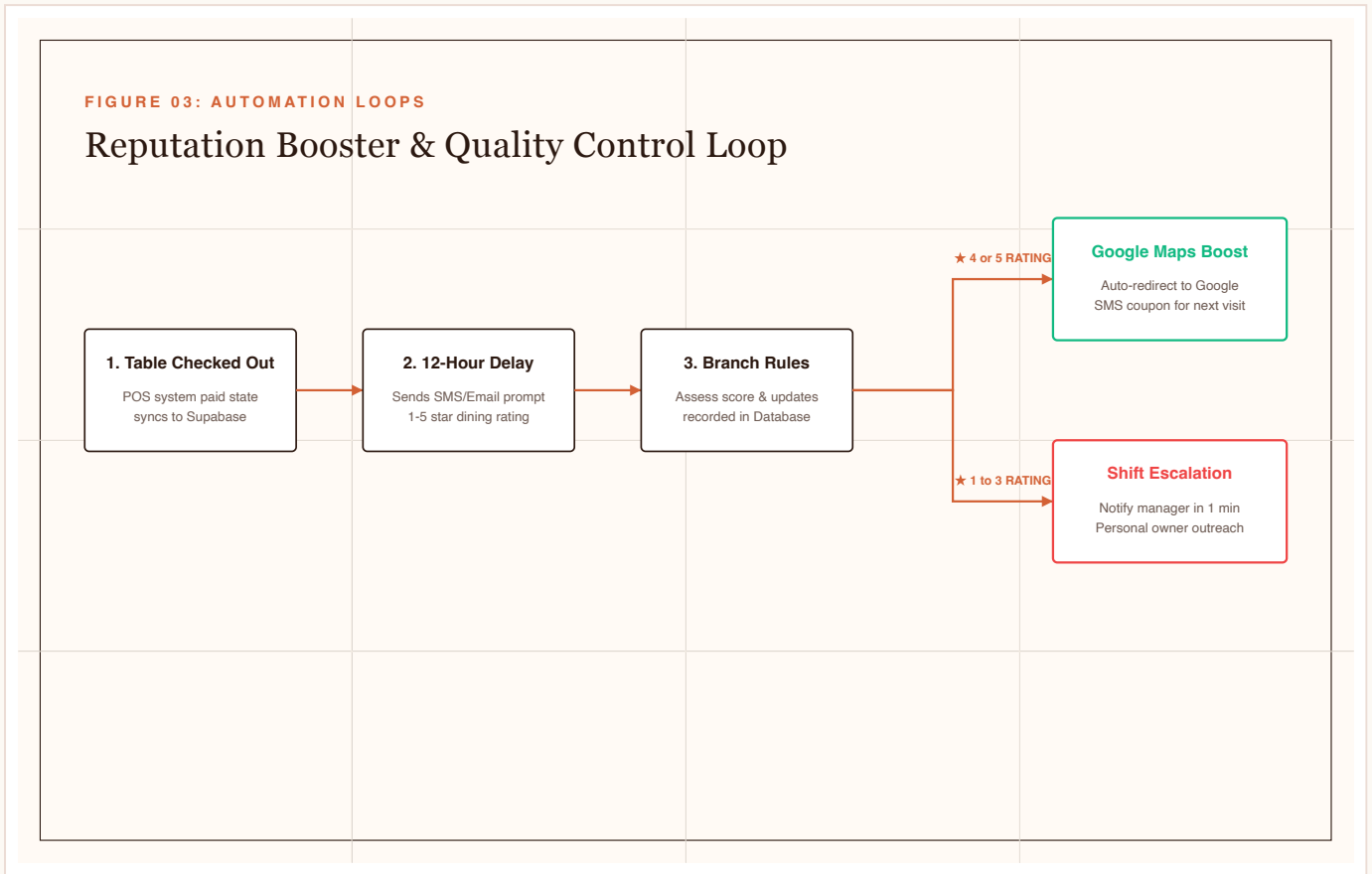
POS WEBHOOK ROUTER

```
// n8n JavaScript Node
const checkout = items[0].json;
const totalSpent =
parseFloat(checkout.receipt.total);
const itemsBought =
checkout.receipt.line_items;

// Detect if guest ordered wine pairings
const boughtWine = itemsBought.some(item =>
  item.category === 'Wine' && item.price >
15
);

return {
  posReceiptId: checkout.receipt.id,
  amount: totalSpent,
  isWineLover: boughtWine
};
```

Reputation Booster & Quality Control



A critical Google Maps review can drop local visibility by 20%. n8n triggers feedback requests 12 hours post-visit. Satisfied guests route directly to public review systems, while unsatisfied responses trigger shift-manager escalations.

08

Loyalty SMS & Coupon Security

To prevent coupon theft and duplicate redemptions, reward coupon codes are hashed and validated against guest IDs in Supabase.

When a guest presents an SMS coupon code at the POS terminal, the FOH screen queries the SQL function. If valid, the POS applies the deduction and flags the database entry as redeemed.

SQL COUPON VALIDATION

```
CREATE OR REPLACE FUNCTION validate_coupon(  
  code_input text,  
  guest_id_input uuid  
) RETURNS boolean AS $$  
DECLARE  
  coupon_valid boolean;  
BEGIN  
  SELECT (active AND NOT redeemed) INTO  
  coupon_valid  
  FROM coupons  
  WHERE code = code_input  
  AND guest_id = guest_id_input;  
  
  IF coupon_valid THEN  
    UPDATE coupons  
    SET redeemed = true  
    WHERE code = code_input;  
    RETURN true;  
  END IF;  
  
  RETURN false;  
END;  
$$ LANGUAGE plpgsql;
```

09

Guest Privacy & Data Governance

Educational and hospitality datasets track private notes, email databases, and critical allergy requirements. Under GDPR guidelines, floor staff must not have access to offline profiles.

Deploying Row-Level Security (RLS) policies checks active reservation logs, limiting query results exclusively to guests dining on the current calendar date.

SUPABASE RLS POLICY

```
-- Lock guest profiles to on-duty staff
CREATE POLICY "FOH Staff Access"
ON guests
FOR SELECT
USING (
  EXISTS (
    SELECT 1 FROM visits
    WHERE visits.guest_id = guests.id
    AND visits.visit_date = CURRENT_DATE
  )
);
```

10

The Horeca Operations Scorecard

Operational Metric	Traditional State	Systemized State	Action Required
Reservation Tracking	Rented OpenTable / Yelp profiles	Unified database sync via n8n	Connect Formitable/SevenRooms webhook
Diner Preferences	Trapped in booking notes or staff memory	Central profile synced to shift dashboard	Build Supabase guest preferences table
Reputation Control	Diners post directly to Google Maps	12h rating gates for recovery routing	Set up automatic SMS surveys in n8n
Loyalty Marketing	Batch newsletter blasts manually	Contextual SMS coupons (Twilio)	Configure unique coupon database checks
Dietary Privacy	Staff prints raw spreadsheets	GDPR-compliant storage with RLS policies	Configure row-level security on profiles

90-Day Implementation Roadmap

Transitioning restaurant operations requires careful, phased executions. Do not launch loyalty SMS campaigns before reservation and POS checkout logs sync correctly.

Establish transaction memory first. You cannot run smart loyalty campaigns on disconnected database fields.

PHASE TIMELINE

Phase 1: Ingestion (Days 1-30)

- Deploy Supabase guest schema.
- Connect booking widget APIs (Formitable).
- Sync Lightspeed POS receipt logs.

Phase 2: Review (Days 31-60)

- Deploy 12-hour feedback loops.
- Connect Google Maps booster routes.
- Establish manager recovery alerts.

Phase 3: Loyalty (Days 61-90)

- Launch Headless CMS Menu Graph.
- Deploy SMS verification functions.
- Enforce Row-Level Security policies.