WhatsApp Integration — Technical Playbook

Event management SaaS · OTP + transactional utility, one-way, no marketing, no chatbot · Admin template management · Node/React on AWS · Cloud API v23.0 · 2026-08-28

Your use case — and why it's the easy path

You are sending: OTP at registration; acceptance/rejection decisions; event updates & reminders; room allocation; and nudges to collect extra info (travel details, etc.) so organisers can arrange facilities. **No marketing. One-way. No chatbot / auto-replies.**

This is the **cleanest possible WhatsApp profile**:

- Everything is either **Authentication** (OTP) or **Utility** (all the transactional messages). You never touch the Marketing category — so no 24–48h approvals, no US pause, lowest cost, easiest approvals.
- One-way + no chatbot means you skip conversational automation entirely. You only need: send templates out, and a webhook to (a) capture delivery status, (b) record inbound replies when applicants send the info you asked for, and (c) honor STOP.
- Cost in India is tiny: Utility ≈ ₹0.145/msg, Auth ≈ ₹0.145/msg, and Utility sent inside an open 24h window is **free**.

> **Key subtlety for your "collect additional info" flow:** when an applicant taps a button or replies with their travel details, that reply opens a 24-hour window. Reading and storing that reply is **data collection, not a chatbot** — fully compatible with your one-way stance. You can even send follow-up Utility messages free inside that window. You just don't need auto-reply logic; a human/organiser or your DB consumes the reply.

1. Message inventory — category per use case

Your message	Category	Notes
OTP at registration	Authentication	Fixed Meta-defined body + copy-code button. You cannot free-form it.
Application accepted	Utility	Transactional decision notice
Application rejected	Utility	Keep neutral/factual tone (avoid "abusive/threatening" rejection)
Event update / schedule change	Utility	Follow-up to their registration
Event reminder (T-24h / T-1h)	Utility	Factual only — no promo or it flips to Marketing
Room / seat allocation	Utility	Can carry a document header (allocation slip PDF)
Collect travel details / extra info	Utility	Body asks for info + a URL button to a form, OR quick-reply buttons

> **Golden rule:** if a message is a direct, expected follow-up to the applicant's registration, it's Utility. The moment it promotes anything, it's Marketing. Since you have no marketing, every template you author is Authentication or Utility.

2. Step-by-step procedure (end to end)

1. **Meta app + WABA.** Create a Business-type Meta App with the WhatsApp use case. Note `WABA_ID` and the test `PHONE_NUMBER_ID`.
2. **Business Verification.** Submit your company for Meta Business Verification (needed to leave test limits and raise tiers).
3. **Accept the DPA.** Accept Meta's WhatsApp Business Data Processing Terms (your GDPR/DPDP processor agreement).
4. **Permanent token.** Create a System User in Business Manager; mint a permanent token with `whatsapp_business_messaging` + `whatsapp_business_management`. Store in AWS Secrets Manager.
5. **Consent capture.** Add an unbundled WhatsApp opt-in checkbox to your React registration form; persist a consent record (text shown, source, timestamp, IP, locale). This is your legal basis under Meta policy + DPDP + GDPR.
6. **Author templates.** Create your Authentication (OTP) + Utility templates — from the admin dashboard via the Template Management API (section 4). Wait for APPROVED (usually 1–5 min for these categories).
7. **Webhook.** Stand up the HTTPS webhook (verification handshake + HMAC verify). Subscribe to `messages` (inbound + statuses) and `message_template_status_update` (approval events for your dashboard).
8. **Real number.** Add + verify your business number (OTP), set a 6-digit 2FA PIN, get the display name approved.
9. **Wire sending.** Registration → OTP send; decision → accept/reject Utility; scheduler → reminders; allocation → room Utility; info-request → Utility with form link/buttons. All behind SQS (section 5).
10. **Capture replies.** When an applicant replies (travel info), the webhook stores the message against their application record; the 24h window opens so any follow-up Utility is free.
11. **Smoke test + go live.** Send to your own number, confirm delivery webhooks fire and inbound capture works, then open the gates.

3. OTP (Authentication) template — special rules

OTP over WhatsApp **must** use a pre-approved Authentication-category template — you cannot send a free-form code. The body is **preset by Meta**; you only choose the button type (copy-code, one-tap autofill, or zero-tap) and pass the code as a parameter at send time.

```
// Create (admin) — Authentication template with copy-code button
POST https://graph.facebook.com/v23.0/{WABA_ID}/message_templates
{
  "name": "registration_otp",
  "language": "en",
  "category": "AUTHENTICATION",
  "components": [
    { "type": "BODY", "add_security_recommendation": true },
    { "type": "FOOTER", "code_expiration_minutes": 10 },
    { "type": "BUTTONS", "buttons": [ { "type": "OTP", "otp_type": "COPY_CODE" } ] }
  ]
}

// Send the OTP
POST https://graph.facebook.com/v23.0/{PHONE_NUMBER_ID}/messages
{
  "messaging_product": "whatsapp",
  "to": "+9198xxxxxxxx",
  "type": "template",
  "template": {
    "name": "registration_otp",
    "language": { "code": "en" },
    "components": [
      { "type": "body",   "parameters": [ { "type": "text", "text": "483920" } ] },
      { "type": "button", "sub_type": "url", "index": "0",
        "parameters": [ { "type": "text", "text": "483920" } ] }
    ]
  }
}
```

Copy-code works on all devices; one-tap autofill / zero-tap are Android-only and fall back to copy-code elsewhere. You generate & verify the code yourself; WhatsApp is only the delivery channel.

4. Admin template management (Template Management API)

You want to create templates from your admin dashboard. Meta exposes full CRUD on `/{WABA_ID}/message_templates`. Wire these into your admin backend (never expose the token to the browser — proxy through your Node API):

Action	Call
Create / submit	`POST /{WABA_ID}/message_templates`
List (with status)	`GET /{WABA_ID}/message_templates?fields=name,status,category,language`
Edit (non-approved / limited fields)	`POST /{TEMPLATE_ID}`
Delete	`DELETE /{WABA_ID}/message_templates?name={name}`

Limits & lifecycle to enforce in your UI:

- Up to **250 template names** per WABA (each name can have multiple language translations, counted as one).

- Statuses: `PENDING → APPROVED / REJECTED`, and later `PAUSED / DISABLED` if quality drops. Only APPROVED templates can be sent.
- Names are lowercase snake_case, unique. You cannot edit an approved template's body without re-review — your dashboard should offer "create v2" rather than in-place edit for approved ones.
- `example` values are mandatory for any component with variables — make them required fields in your template-builder form, or submissions auto-reject with `INVALID_FORMAT`.
- Subscribe to the `message_template_status_update` webhook so your dashboard reflects APPROVED/REJECTED in real time instead of polling. The payload carries `event`, `message_template_id`, `message_template_name`, and a rejection `reason`.

Admin dashboard design (React + Node):

```
React admin ── Template Builder form ──► POST /api/admin/templates (your Node API)
                                            │  (holds the Meta token server-side)
                                            ▼
                                    POST graph.facebook.com/{WABA_ID}/message_templates
                                            │
                          DynamoDB "wa_templates": { name, category, language,
                            components_json, meta_template_id, status, reject_reason,
                            created_by, created_at }
Meta ──► webhook message_template_status_update ──► update status/reject_reason in DynamoDB
                                                      ──► push to admin UI (WebSocket/poll)
```

Guard-rails to bake into the builder: force category = UTILITY or AUTHENTICATION only (you have no marketing); validate variable numbering is sequential from {{1}}; require an example per variable; block editing an APPROVED template (offer clone-to-v2); show live status from the webhook.

5. Sending architecture (one-way, AWS serverless)

```
Registration (React) ──► API GW ──► Lambda(register)
   ├─ DynamoDB: applications + whatsapp_consent
   ├─ generate OTP → SQS "wa-send" (registration_otp)
   └─ EventBridge "application.decided" / "event.updated" / "room.allocated"
            ▼
        SQS "wa-send"  ──►  Lambda "wa-sender"
             • check whatsapp_opt_in === true
             • pick approved template (from wa_templates config)
             • POST Graph API; 429/5xx → throw → SQS retry → DLQ; 4xx → DLQ+alert
EventBridge Scheduler (T-24h / T-1h) ──► SQS ──► wa-sender (reminders)
Meta ──► API GW ──► Lambda "wa-webhook"
   • verify HMAC, ACK 200 fast
   • statuses → update delivery state (idempotent, keyed on message id)
   • inbound reply (travel details) → store against application, mark window_open_until
   • STOP/UNSUBSCRIBE → set opt_out
```

Because you're one-way and no-chatbot, the webhook's inbound branch just **persists** the applicant's reply (e.g. travel details) against their record for organisers to consume — no auto-response. Reserved concurrency on `wa-sender` caps throughput to your messaging tier; SQS + DLQ absorb event-day bursts.

6. Info-collection nudge (Utility) — without a chatbot

Two clean patterns, both one-way:

- **Link-out (simplest):** a Utility template with a URL button to a secure form on your site (pre-filled with their application id). No inbound parsing at all. Recommended.
- **Reply capture:** a Utility template asking them to reply with the info; your webhook stores the raw reply text against their application for an organiser to review. Still no auto-reply.

```
// Utility: collect travel details via a form link
{
  "name": "collect_travel_details",
  "language": "en",
  "category": "UTILITY",
  "parameter_format": "positional",
  "components": [
    { "type": "BODY",
      "text": "Hi {{1}}, to arrange your stay & travel for *{{2}}*, please share your arrival
details using the secure form below. Booking ID: {{3}}.",
      "example": { "body_text": [["Dixit","Annual Summit 2026","APP-10423"]] } },
    { "type": "FOOTER", "text": "Reply STOP to opt out" },
    { "type": "BUTTONS", "buttons": [
      { "type": "URL", "text": "Add travel details",
        "url": "https://yourevents.com/app/{{1}}/travel", "example": ["APP-10423"] } ]}
  ]
}
```

7. Compliance (unchanged, but lighter for you)

- **Opt-in** collected outside WhatsApp, unbundled, names your business + purpose, timestamped/source-logged. One checkbox satisfies Meta + GDPR + India DPDP.
- Because everything is transactional (OTP + registration follow-ups), you have a strong lawful basis; DPDP's "voluntary provision for a specified purpose" fits acceptance/allocation/updates. Still log consent + honor erasure within 90 days.
- Honor STOP immediately (webhook sets opt_out). One-way sending + strict opt-in keeps your Quality Rating Green.
- India DPDP full enforcement 13 May 2027; startups skip DPO/DPIA/audit. Cross-border transfer to Meta currently permitted.

8. Cost (India, eff. 1 Jan 2026)

Category	Per delivered msg
Authentication (OTP)	₹0.145
Utility	₹0.145 (free inside an open 24h window)
Service (free-form reply)	Free (within 24h window)

Illustration: an event with 1,000 applicants receiving OTP + accept + reminder + allocation (4 messages) ≈ 1000 × 4 × ₹0.145 = **₹580**, less any that land inside a free window after the applicant interacts.

9. Go-live checklist

- Business Verification + DPA accepted

- Real number registered, display name approved, 2FA PIN set
- Permanent token in Secrets Manager (never in the browser)
- OTP (Authentication) + all Utility templates APPROVED
- Admin template management wired to Template Management API + status webhook
- Unbundled consent checkbox + consent record live
- Webhook: HMAC verify, status capture, inbound-reply capture, STOP handling
- SQS + DLQ + reserved concurrency; every send checks opt_in === true
- Erasure workflow (90-day DPDP / GDPR); cost monitoring by category

---

Sources: Meta for Developers (Cloud API, Business Management API authentication templates, message-template management, webhooks, get-started), Meta Terms + Data Processing Terms + platform pricing, SuprSend OTP guide 2026, AiSensy India per-message pricing (eff. 1 Jan 2026), 360dialog template limits/lifecycle, Plivo/EngageLab/Vonage template-management references, Recording Law India DPDP Act 2023 & Rules 2025. Confirm live rates and pin the Graph API version at integration time.