WhatsApp Lead Bot architecture via Meta Cloud API

Yestoryd can build a Gemini-powered WhatsApp lead bot for under ₹500/month at early stage, scaling to ~₹4,000 at 1,000 conversations/month — an 80%+ savings versus going through a BSP like AiSensy. The key enabler: Meta made all service (inbound) conversations completely free in November 2024, Greenadsglobal and since the lead bot handles prospects who message first, the WhatsApp messaging cost is effectively zero. This plan covers every step from Meta app creation to Supabase schema design, with specific attention to India's DPDP Act requirements around children's data — the single highest compliance risk for Yestoryd.

The recommended stack is a **Fastify + TypeScript server on Hetzner CX22 (Singapore)**, calling **Gemini 2.5 Flash** for AI responses, storing everything in **Supabase PostgreSQL**, and using Meta's Cloud API directly for WhatsApp messaging. Setup takes 40–80 developer hours; ongoing maintenance runs 5–10 hours/month.

A. Meta Cloud API setup: from zero to production

Creating the Meta App and WABA

The setup begins at developers.facebook.com. Create a new app by selecting **"Other"** as use case, then **"Business"** as app type. ExpertFlow Link it to your Meta Business Manager account (create one if needed). Once created, add the **WhatsApp product** from the app dashboard — this triggers automatic creation of a WhatsApp Business Account (WABA) and provides a test phone number with five pre-registered recipient slots. DevOpsSchool.com

The test number is send-only and cannot receive messages — it exists purely for API validation. For the lead bot, you need to register Yestoryd's new SIM immediately.

Registering the Indian SIM

Navigate to **App Dashboard → WhatsApp → API Setup → "Add phone number"**. DevOpsSchool.com The number must NOT be currently registered on WhatsApp or WhatsApp Business — uninstall WhatsApp completely and restart the phone before proceeding. Anjoktechnologies Enter the number in +91 format, complete OTP verification (SMS or voice call), and set a display name that matches your legal business name. You'll receive a **Phone Number ID** — store this alongside your **WABA ID**, as both are required for every API call. DevOpsSchool.com

Critical constraint: Without business verification, the number is limited to **250 unique contacts per day**. Salesforce After verification, this jumps to Tier 1 (1,000/day) with automatic upgrades based on volume and quality. WANotifier

Business verification for Indian companies

Go to **Business Settings → Security Center → Start Verification**. DevOpsSchool.com Meta cross-checks with MCA (Ministry of Corporate Affairs) records, so all details must match exactly. Required documents for Indian businesses:

- **Certificate of Incorporation** or **GST Certificate** (Heltar) (primary identity proof)
- **UDYAM/MSME Registration** or utility bill under company name (address proof) (Heltar)
- **Active website with SSL** displaying the legal business name, address, and phone matching Business Manager (Gallabox)
- **Domain email** (e.g., amit@yestoryd.com) — strongly recommended

Approval typically takes under 1 working day (Jalpi) but can stretch to 14 days. There is **no fee** from Meta for verification.

Permanent access token via system user

Temporary tokens expire in ~1 hour and are unsuitable for production. (Webflow) Instead: go to **Business Settings → Users → System Users**, create a system user with "Employee" access, assign it your WhatsApp app and WABA with **Full Control**, then generate a token with (whatsapp_business_messaging) and (whatsapp_business_management) permissions. This token does not expire. Store it in environment variables — never in client-side code or Git repos. (Supadex)

Messaging tiers and throughput

Tier	Unique contacts/24h	How to reach
Unverified	250	Default
Tier 1	1,000	Business verified
Tier 2	10,000	Send to ≥500 unique users in 7 days, maintain medium+ quality
Tier 3	100,000	Send to ≥5,000 unique users in 7 days
Unlimited	No limit	Sustained high quality and volume

Default throughput is **~80 messages per second** per number, auto-upgrading to ~1,000 mps for eligible accounts. (Gadget Hacks) For an early-stage lead bot handling dozens of concurrent conversations, this is more than sufficient.

Meta's current pricing for India (post-July 2025 per-message model)

Meta shifted from per-conversation to **per-message pricing** on July 1, 2025. (Eesel AI +2) The critical insight for Yestoryd:

Category	Rate (INR)	When it applies
Service	**₹0 (FREE)**	All replies within 24h of user's last message
Utility (within 24h window)	**₹0 (FREE)**	Since July 2025
Utility (outside window)	~₹0.13	Order updates, confirmations sent proactively
Marketing	**₹0.86**	Proactive promotions, re-engagement templates
Authentication	~₹0.13	OTP verification

Since the lead bot responds to inbound messages from prospects, **80%+ of messaging costs are zero**. Only follow-up marketing templates sent after the 24-hour window closes incur the ₹0.86 charge. This is what makes direct Cloud API dramatically cheaper than BSP-mediated approaches.

B. Webhook architecture: receiving and processing messages

Verification and subscription

Meta verifies your webhook endpoint by sending a GET request with `hub.mode=subscribe`, `hub.verify_token` (your secret string), and `hub.challenge`. Your server must validate the token and return the challenge value with HTTP 200. After verification, subscribe to the **`messages`** webhook field — this covers both incoming messages and delivery status updates. GitHub

```javascript
// Webhook verification (GET)
app.get('/webhook', (req, res) => {
  const mode = req.query['hub.mode'];
  const token = req.query['hub.verify_token'];
  const challenge = req.query['hub.challenge'];
  if (mode === 'subscribe' && token === process.env.VERIFY_TOKEN) {
    res.status(200).send(challenge);
  } else {
    res.sendStatus(403);
  }
});
```

Incoming message payload structure

All webhooks arrive as POST requests. WhatsApp Extract the sender phone number from `entry[0].changes[0].value.messages[0].from`, the message body from `entry[0].changes[0].value.messages[0].text.body` (for text), and the WhatsApp message ID from `entry[0].changes[0].value.messages[0].id` npm for deduplication.

Button replies arrive under `interactive.button_reply.id` and `interactive.button_reply.title`; GitHub list replies under `interactive.list_reply.id`.

Message sending API

All outbound messages go through `POST https://graph.facebook.com/v21.0/{PHONE_NUMBER_ID}/messages`. The bot will primarily use three message types:

Text messages for natural conversation responses. **Interactive reply buttons** (max 3 buttons, 20-char titles) Cognigy for structured choices like "Yes, tell me more" / "Talk to a human" / "Take the assessment". **Interactive list messages** (max 10 items across sections) Whatsapp for presenting program options or FAQ categories. Template messages are needed only for re-engagement outside the 24-hour window.

Reliability best practices

The single most important rule: **respond with HTTP 200 immediately, then process asynchronously**. 360dialog Meta expects a response within 5 seconds and retries with exponential backoff for up to 7 days if it doesn't receive one. Use the WhatsApp message ID (`wamid.*`) as a deduplication key — duplicate webhook deliveries are common, especially for status updates. Validate the `X-Hub-Signature-256` header using your App Secret via HMAC-SHA256 to confirm payloads actually originate from Meta. GitHub

---

C. Hetzner Singapore beats Vercel for this use case

Why serverless falls short for conversational AI

Vercel's serverless functions suffer **1–3 second cold starts** GitHub when the bot hasn't received traffic recently DEV Community — a real problem for a WhatsApp bot where parents might message at 10 PM after a quiet period. Adding Gemini API latency (0.8–1.5s) on top of a cold start means the first response could take **4–5 seconds**, creating a noticeably sluggish experience. Vercel Pro ($20/month) provides adequate execution time limits Flexprice (300s standard, 800s with Fluid Compute), Inngest but the cold start issue is fundamental to serverless architecture.

Hetzner CX22 Singapore is the clear winner

Factor	Hetzner CX22 (Singapore)	Vercel Pro
Monthly cost	**₹400** (~€4.50)	₹1,680 ($20)
Cold starts	**None — always warm**	1–3s after inactivity
Latency to India	**30–60ms**	Variable (nearest edge)
Background jobs	Full cron, workers, queues	Limited (waitUntil)
In-memory state	Supported	Not possible
DevOps overhead	Medium (server mgmt)	Near-zero

Hetzner Singapore provides 2 vCPUs, 4GB RAM, 40GB NVMe SSD, Hetzner and 20TB bandwidth Hetzner for ~€4.50/month. Hetzner The Singapore data center delivers **30–60ms latency to India** — excellent for a conversational bot. Setup involves Ubuntu, Node.js, Caddy (automatic HTTPS via Let's Encrypt), and PM2 for process management. Initial setup takes 2–4 hours; ongoing maintenance is ~30 minutes/month for security updates.

The recommended hybrid architecture: Keep Yestoryd's main website on Vercel. Deploy the WhatsApp bot as a separate Fastify service on Hetzner Singapore. Both share Supabase as the common database. This keeps deployment simple for the marketing site while giving the bot the always-on reliability it needs.

D. Bot architecture: state machine meets Gemini

Hybrid conversation state management

Pure AI-driven conversation is unpredictable; pure decision trees feel robotic. The solution is a **hybrid state machine** where structured states ensure all lead data gets captured, while Gemini handles the natural language within each state.

Conversation states: GREETING → CHILD_INFO → READING_CONCERNS → PARENT_INFO → URGENCY_ASSESSMENT → QUALIFICATION_COMPLETE → optionally HUMAN_HANDOFF. State is stored in Supabase's conversations table (not in-memory — server restarts would lose state, and not Redis — unnecessary complexity at this scale). Each incoming message triggers: fetch state → build context → call Gemini → extract structured data → update state → send response. Total processing time: **1–3 seconds**.

Gemini 2.5 Flash integration

Gemini 2.5 Flash delivers **0.36–0.38s time-to-first-token** Artificial Analysis and **200+ tokens/second output speed**, Artificial Analysis Apidog making it fast enough for WhatsApp's conversational pace. Use Google's official @google/genai SDK: Google AI

```typescript
const response = await ai.models.generateContent({
  model: "gemini-2.5-flash",
  contents: conversationHistory,
  systemInstruction: systemPrompt,
  generationConfig: { maxOutputTokens: 150, temperature: 0.7 }
});
```

Key system prompt design principles: Instruct Gemini to keep responses under 3 short sentences (WhatsApp-friendly). Ask ONE question at a time. Mirror the user's language — if they write Hinglish ("mera beta 5 saal ka hai"), respond in Hinglish. Include the current conversation state and collected data in the prompt so Gemini knows what information still needs to be gathered. Set maxOutputTokens: 150 to prevent verbose responses. Send only the last 10–15 messages as context to limit input token costs.

For **structured data extraction**, instruct Gemini to output a JSON block alongside its conversational response containing any newly extracted data: `{"child_name": "Arjun", "child_age": 6, "concerns": "doesn't like reading, prefers videos", "urgency": "medium"}`. Parse this server-side to update the lead record.

Supabase schema design

Three core tables power the system:

leads — the primary business entity with `phone_number` (unique), `parent_name`, `child_name`, `child_age`, `reading_concerns`, `urgency` (high/medium/low), `lead_score`, and `status` (new → qualifying → qualified → handed_off → converted → lost).

conversations — tracks session state with `current_state`, `collected_data` (JSONB), `is_bot_active` (boolean for human handoff), `assigned_agent_id`, and `last_message_at` for timeout management.

messages — full conversation history with `direction` (inbound/outbound), `sender_type` (user/bot/agent), `content`, `wa_message_id` (for deduplication), and `metadata` (JSONB for button IDs, extracted data, etc.).

Index on `phone_number`, `conversation_id`, and `created_at` for query performance. Enable Row Level Security on all tables. `Supabase` The bot server uses Supabase's `service_role` key (bypasses RLS); the agent dashboard uses authenticated role with read-only policies.

Human handoff mechanism

Escalation triggers (checked in order): explicit request ("talk to someone", "agent", "insaan se baat karo"), AI-detected frustration (Gemini outputs an `ESCALATE` flag), keyword detection ("complaint", "refund", "shikayat"), conversation stuck (3+ messages without state progression), or high-value lead ready for closing. `Connverz`

When escalation fires: set `conversations.is_bot_active = false`, send the parent a message ("Connecting you with a Yestoryd reading expert — they'll reply within 10 minutes! 😊 "), and notify agents via Supabase Realtime push to a dashboard + WhatsApp message to Rucha/Amit's phones via Channel 1 (AiSensy). All subsequent messages for this conversation skip Gemini and go straight to the agent inbox. `Connverz` The agent clicks "Return to bot" in the dashboard to re-enable automation.

Session timeout handling

If a user goes silent for **4+ hours**: mark conversation as `PAUSED`, preserve state. If they return within 24 hours: resume from saved state with "Welcome back!" context. If they return after 24 hours: send a pre-approved template message to re-open the window `n8n` `Wetarseel` ("Hi {parent_name}, would you like to continue exploring reading options for {child_name}?"). After **7 days** of inactivity: mark as `STALE` and create the lead record with whatever data was collected.

Recommended tech stack

Layer	Technology	Why
Runtime	Node.js 20+ / TypeScript	Team familiarity, strong Gemini/Supabase SDK support

Layer	Technology	Why
HTTP framework	Fastify 5.x	2–3× faster than Express, built-in validation, Pino logging
AI	@google/genai SDK	Official Gemini client
Database	@supabase/supabase-js 2.x	Supabase client with realtime support
Validation	Zod 3.x	Runtime validation for webhook payloads
Process manager	PM2	Auto-restart, cluster mode, log management
Reverse proxy	Caddy	Automatic HTTPS, zero-config
Testing	Vitest	Fast TypeScript-native testing

E. Cost analysis: direct API saves 77–89% versus BSP

Per-conversation cost breakdown

A typical lead qualification conversation involves **15–20 message exchanges**. With Gemini 2.5 Flash, this consumes approximately **33,000 input tokens** (growing context) and **2,000–3,500 output tokens** (including thinking), costing roughly **₹1.26–₹1.60 per conversation** on the paid tier. On Google's free tier (500 requests/day, Hostbor 15 RPM), cost is zero — usable for early stage but risky for production due to rate limits.

Total monthly cost at three scales

Component	100 convos/mo	500 convos/mo	1,000 convos/mo
WhatsApp (80% inbound = free)	₹17	₹86	₹173
Gemini 2.5 Flash	₹0 (free tier)	₹630	₹1,260
Hetzner CX22 Singapore	₹378	₹378	₹378
Supabase	₹0 (free tier)	₹0 (free tier)	₹2,100 (Pro)
Domain	₹80	₹80	₹80
Total (Direct + Hetzner)	**₹475**	**₹1,174**	**₹3,991**
Total via AiSensy + AI	**₹3,520–4,520**	**₹5,137–6,007**	**₹5,876–6,616**
Savings	**86–89%**	**77–80%**	**32–40%**

The WhatsApp messaging cost line assumes 20% of conversations require marketing template follow-ups at ₹0.86 each. At early stage, this is likely lower. **The single largest variable cost is Gemini API usage**, which can be reduced ~75% by switching to **Gemini 2.5 Flash-Lite** ($0.10/$0.40 per 1M tokens Artificial Analysis +2 vs $0.30/$2.50) at the cost of slightly less capable reasoning.

What you trade by skipping AiSensy

AiSensy provides a ready-made dashboard, campaign management, analytics, template management UI, agent routing, and customer support — plus a no-code chatbot builder (₹1,999/month add-on). Going direct means building webhook handling, message sending logic, conversation analytics, and agent dashboard from scratch. The **40–80 hours of initial developer time** is the real hidden cost. At ₹1,000/hour, that's ₹40,000–80,000 one-time — which pays for itself within 2–4 months of BSP subscription savings.

F. Compliance: children's data is the critical risk

DPDP Act and Yestoryd's specific exposure

India's Digital Personal Data Protection Act 2023, DLA Piper CookieYes with Rules published January 2025, EY Lexology defines a child as anyone **under 18** — stricter than GDPR (16), COPPA (13), King Stubb & Kasiva or any peer regulation. Full enforcement begins **May 13, 2027**, The Wire but the Data Protection Board is already operational since November 2025. CookieYes Penalties reach **₹200 crore** for non-compliance with children's data provisions. Atlassystems

Since Yestoryd collects children's names, ages, and reading abilities, **verifiable parental consent is mandatory** before processing any child's data. King Stubb & Kasiva The parent chatting on WhatsApp provides a baseline, but you need an explicit consent mechanism — a message like "I confirm I am the parent/guardian of [child] and consent to collection of their reading profile" with an affirmative reply button. This must happen before the bot collects child-specific information.

Prohibited activities with children's data: no behavioral tracking, no targeted advertising, no profiling that could be detrimental. Atlassystems Yestoryd's use case (recommending books based on age and reading level) is permissible, but **never use this data for ad targeting or behavioral analytics**.

WhatsApp's January 2026 AI chatbot policy

Meta banned **general-purpose AI chatbots** from WhatsApp Business Platform effective January 15, 2026. Cognativ However, **business-specific bots** for customer support, lead qualification, FAQ handling, and booking are explicitly permitted. Eesel AI Yestoryd's lead bot is compliant — it serves a specific business function, not open-ended AI chat. The requirement is that you provide **clear escalation paths to human agents** including live chat, phone number, or email. Social Intents

24-hour window and quality management

When a prospect messages first, a **24-hour customer service window** opens. All bot replies within this window are free and unrestricted. The window resets with each new user message. After 24 hours of silence, only pre-

approved template messages can be sent. (n8n) Design the qualification flow to complete within a single session — most conversations will finish in under 10 minutes.

For quality maintenance: never message users who haven't initiated conversation (the bot is inbound-only — already compliant). Vary response language to avoid sounding repetitive. Marketing templates are capped at **2 per user per 24 hours**. (Gadget Hacks) Monitor quality rating in WhatsApp Manager — if it drops to Yellow, pause all proactive outreach immediately. If Flagged, quality must recover within 7 days or the messaging tier drops. (Brevo Help)

Pre-launch compliance checklist

The non-negotiable items before going live:

- **Parental consent flow**: Explicit "I am the parent/guardian and consent to..." with affirmative button before collecting child data
- **Privacy notice**: Published and linked in WhatsApp — covering data collected, purpose, retention period, rights, contact info, grievance officer
- **AI disclosure**: Opening message must state "I'm Yestoryd's AI assistant" — builds trust and aligns with WhatsApp policy
- **Human escalation path**: Clear way to reach Rucha or Amit at every point in the conversation
- **Opt-out mechanism**: "Type STOP to unsubscribe" honored immediately
- **Supabase RLS**: Row Level Security on all tables, (Supabase) PII fields encrypted at rest
- **Data retention policy**: Define 1–3 year retention, automated deletion, 48-hour pre-erasure notification
- **Grievance officer**: Designated person/email for data principal complaints (DPDP requirement)
- **AI data usage**: Confirm Gemini API is configured to NOT use conversation data for model training (use paid tier, not free)

Hallucination guardrails

The bot will discuss pricing (₹5,999 for 3-month coaching), session format, and coach qualifications. **Never let Gemini generate these facts from memory** — pull them from a verified configuration object and inject them into the system prompt. For any question the bot can't answer confidently, the fallback must be: "Great question — let me connect you with our team for the most accurate answer!" Log all AI-generated responses for quality review.

Conclusion: a lean, production-ready architecture

The architecture is deliberately simple: one Hetzner server, one Supabase database, one Gemini API, one WhatsApp number. At ₹475/month for 100 conversations, it's cheaper than a single AiSensy subscription — and at 1,000 conversations, it's still 35% cheaper while providing complete control over the AI experience and conversation design. The hybrid state machine approach ensures reliable lead data capture while Gemini delivers the natural, Hinglish-fluent conversation that Indian parents expect.

The two highest-priority items for Yestoryd before launch are **parental consent implementation** (DPDP Act's ₹200 crore penalty (Atlassystems) makes this non-negotiable) and **business verification** (unlocks Tier 1 messaging limits and display name approval). Both can run in parallel with the 40–80 hours of development work. A realistic timeline from "new SIM in hand" to "live lead bot" is **2–3 weeks** with a dedicated developer.