JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Engageo: A Multilingual Voice and Messaging System for Missed-Call Recovery in Indian Specialty Clinics

Architecture, Design, and Engineering Lessons from Production Deployment

[1]Atul Hooda, [2]Siddharth

[1]CTO, Engageo Agency, [2]CEOEngageo Agency
Department of Computer Engineering,
Army Institute of Technology, Pune, India

Abstract : Major portions of high-intent calls into independent Indian specialty clinics-dental, dermatology, hair transplant, fertility, and ophthalmology-go unanswered since their front-desk staff is unable to keep pace with the inbound calls. One missed call from a patient with high-intent can mean foregone revenue of 40k to 3L INR per appointment, depending on specialty. Current solutions fare differently. IVR systems cannot understand natural speech. Web forms require patient action. Rule-based chatbots break on Indian colloquialisms and accents. Developer-oriented conversation AI platforms require integration work that clinic owners simply cannot perform. Standard voice-based systems are usually designed for a monolithic English context, not the multiplicity of languages used by Indians when interacting with healthcare.

In this paper, we propose the architecture, design, and early operational experience of Engageo, a missed-call recovery system piloted at five Indian specialty clinics. Our system is structured around the Decision Lifecycle (6 stages, from missed-call to verified appointment and patient reporting). Engageo supports 12 Indian languages: Hindi, Hinglish, English, Punjabi, Bengali, Tamil, Telugu, Kannada, Malayalam, Gujarati, Marathi, Odia, and allows patient-driven language-switching at turn boundaries.

We have 5 main contributions: (1) The Decision Lifecycle framework for both engineering and human communication. (2) A multilingual conversation architecture across 12 Indian languages. (3) The Script Coherence Rule, a prompt-level constraint to address a TTS failure specific to Indian multilingual systems: text-to-speech with a mixed-script output results in unintelligible audio when multiple scripts are concatenated together (e.g. Devanagari concatenated with Gurmukhi, Devanagari with Tamil). (4) A 5-stage messaging lifecycle: Confirmation, Reminders, Follow-up, Missed Call follow-up, and No-Show recovery. (5) Engineering patterns for systems like this, such as race condition handling during calendar booking, atomic guards against redundant sends, and Indian phone number normalization.

This paper is a systems-and-design contribution. We share qualitative results from roughly 50 in-depth interviews with Indian specialty clinic owners, and from early deployments at our five pilot clinics. We are performing a quantitative outcome study with these clinics and plan to report these results separately.

I. INTRODUCTION

These independent specialty clinics in India – dental, dermatology, hair transplant, fertility, and ophthalmology, suffer from a systemic bottleneck, a single point of failure for patient acquisition at the front desk. With just one practitioner and one receptionist, a clinic can receive 30-120 calls per day. The receptionist cannot respond to all calls, is unavailable while treating patients, and uses voicemail outside of business hours. A significant portion of high intent patients are lost. For high-ticket specialties, the cost is astronomical. A single hair transplant inquiry can be worth 80,000 to 300,000 rupees. An appointment for a dental implant consultation can be worth 40,000 to 200,000 rupees. An inquiry about an IVF treatment can be worth 150,000+ rupees, capture it in the moment it is received. If a patient hits voicemail, they simply call the next clinic on the search results. Conventional solutions fail each in their own way. Traditional IVR systems lack the ability to understand natural language. Web contact forms require patients to do extra work and lose the urgency. Rule-based chatbots fail under the linguistic variance of real Indian patients and their tendency to blend two or more languages within a single message. General-purpose conversational AI platforms are not designed for clinic owners, as they are intended as developer-facing tools designed primarily for English and require significant technical integration. This paper introduces Engageo-a missed call recovery system that has been in pilot use in 5 Indian specialty clinics. Engageo's structure revolves around the Decision Lifecycle, a six-step process from missed call to confirmed appointment, at the heart of which is a conversational voice agent that understands 12 Indian languages and is integrated with the phone system, calendar, customer record, and messaging systems.

II. Background and Related Work

Response latency is a well-documented factor of conversion across service industries. Experiments in real estate, automotive sales, and B2B software have demonstrated that inquiries responded to within 5 minutes convert several times better than inquiries responded to within 30 minutes. The mechanism is intuitive: a person in the market is in a high-intent state that will quickly decay as they start seeing alternative solutions or when context switches. While literature on response-time effects in the healthcare industry does exist, it has been confined largely to emergency-response systems and has not extended to the elective-care inquiry funnel where most acquisition revenue is generated.

The application areas for conversational agents in healthcare have generally included symptom-checking, medication adherence reminder, post-discharge follow up and patient education. Systematic reviews have found acceptable patient satisfaction with AI-based conversation across low-stakes informational contexts. A gap, however has existed throughout the literature focusing on post-acquisition engagement, helping patients already within the clinical relationship, rather than the pre-appointment inquiry phase. The inquiry phase represents a discrete set of interaction requirements: capturing intent quickly, gracefully handling ambiguity, completing a transactional booking and exiting smoothly without access to patient history, unlike post-acquisition systems.

Developing voice for Indian languages has consistently lagged English voice AI due to smaller training corpora, increased dialectal variation and high incidence of code-switching. While recent developments in multilingual speech models have narrowed the gap, two problems remain poorly solved. First is dealing with in-utterance code-switching where speakers can switch languages within a single sentence; example, “appointment kal subah 11 baje chahiye, is that okay?” contains Hindi, English, and transliterated Hindi numeral and is fully contained within 16 words. While some systems try to force either start-of-conversation language choice or per-utterance language detection, this does not account for the true fluidity of conversation. Second is multi-script synthesis; text-to-speech models performing synthesis across two or more non-Latin Indic scripts often have poor prosody or completely unintelligible outputs. To the best of our knowledge, this mode of failure has not previously been reported in literature.

Several commercial offerings exist in clinic-side automation. Developer-facing conversational AI platforms offer building blocks that require significant integration work and limited support for Indian languages. Booking marketplace platforms intermediary between patient and provider but pass ownership of acquisition leverage to the marketplace and provide no assistance in recovering from a missed call to the clinic's phone number. Light-weight chatbots and IVR systems allow for scripted automation but fail with natural conversational interaction and do not support high-ticket specialty booking workflows. None of the categories directly serve the niche of missed-call recovery from the clinic's existing phone number and require no integration effort from the clinic.

III. Problem Validation: Findings from Informal Clinic Interviews

Before we built Engageo, we conducted around fifty exploratory calls with clinic owners and managers of Indian specialty clinics in hair transplant, dental, dermatology, fertility, and ophthalmology. These were not structured research interviews but rather business development conversations as part of an early customer discovery process where notes were kept informally in chat applications and email. We summarize the common patterns observed here not as findings with counts but as a qualitative pattern that formed the basis of the system's design.

The primary issue described by clinic owners across all specialties was similar and nearly identical: large numbers of incoming calls during clinic hours, a receptionist who is physically incapable of picking up every call, and a clear acknowledgment that lost calls mean lost revenue. Multiple owners cited attempted solutions (a second receptionist, a phone tree / IVR, a basic chatbot) which they abandoned after weeks, attributing their failures to an increased fixed cost (in the case of a second receptionist) or poor patient experience (IVRs drove away older patients more than voicemails, and basic chatbots did not account for language variations in incoming patient calls).

The second consistent pattern was language. Owners in Maharashtra cited calls coming in on a mix of Marathi, Hindi, English, and numerous code-switched variants. Owners in Punjab cited Punjabi mixed with the same other languages. In southern states, callers called in Tamil, Telugu, Kannada, or Malayalam in addition to English. In each case, owners emphasized that a system that could not accommodate non-English calls would not work for a substantial number of their patients, particularly the elderly or those calling from smaller towns.

The third theme was the integration ceiling. Clinic owners refused to switch their phone carrier, their calendar program, or their existing patient record system to accommodate a new system. If the system required developer work on the clinic's end, the project was dead. This influenced our design choice of periphery-integrating with the existing system and taking the clinic's established programs as unchangeable fixtures around which to build orchestration logic.

A more subtle fourth theme was trust. Several owners were concerned that an AI system would mishandle a clinical question or make a mistake that would embarrass them in front of a patient, causing it to overbook them or similarly create a difficult interaction. These concerns shaped our design decisions around conversational design (Section 8), particularly the rules aroundAI disclosure, clinic-question filtering, and explicit appointment confirmation before committing to a calendar entry.

These conversations were not formal research, without structured instrumentation or formal coding methodology, and we present them here only for context not empirical validity. They are the source of the system's design priorities, the multi-language coverage, lack of clinic-side integration required, careful handling of clinic-questions, and the position of the system as a front desk replacement.

IV. THE DECISION LIFECYCLE: A SIX-STAGE RECOVERY FRAMEWORK

Engageo is built around the six-step process known as the Decision Lifecycle. The Decision Lifecycle breaks the problem of missed-call recovery into discrete operational steps, each having a distinct input, a distinct output, and a distinct success metric. This decomposition fulfills two goals: as an engineering principle to guide system modules and as an operational concept owners can use to understand system behavior without technical detail.

Figure 1: A Six-Stage Missed-Call Recovery Framework

This decomposition is not just a story. Every stage has a failure mode and recovery path that is independent of every other stage. A failure to detect leads to no follow-up actions. A failure in routing may lead to a callback at the wrong time or with the wrong initial sentence. A failure to engage (the patient doesn't answer) leads to a Whatsapp notification and not a subsequent call. A failure to qualify sends the patient to a human being. A failure to convert leads to an offer to take a different appointment. A failure to archive doesn't impact the patient, but sends an internal notification to investigate later.

V. SYSTEM ARCHITECTURE

Engageo's backend is a custom one, which brings together the speech, language, telephony, calendar, and messaging features, within a unified end-to-end pipeline. The infrastructure is designed with 3 operational constraints:- a latency target of sub 8 seconds between missed-call to outbound callback, no developer work on the client side,and limited resource usage under high concurrency.

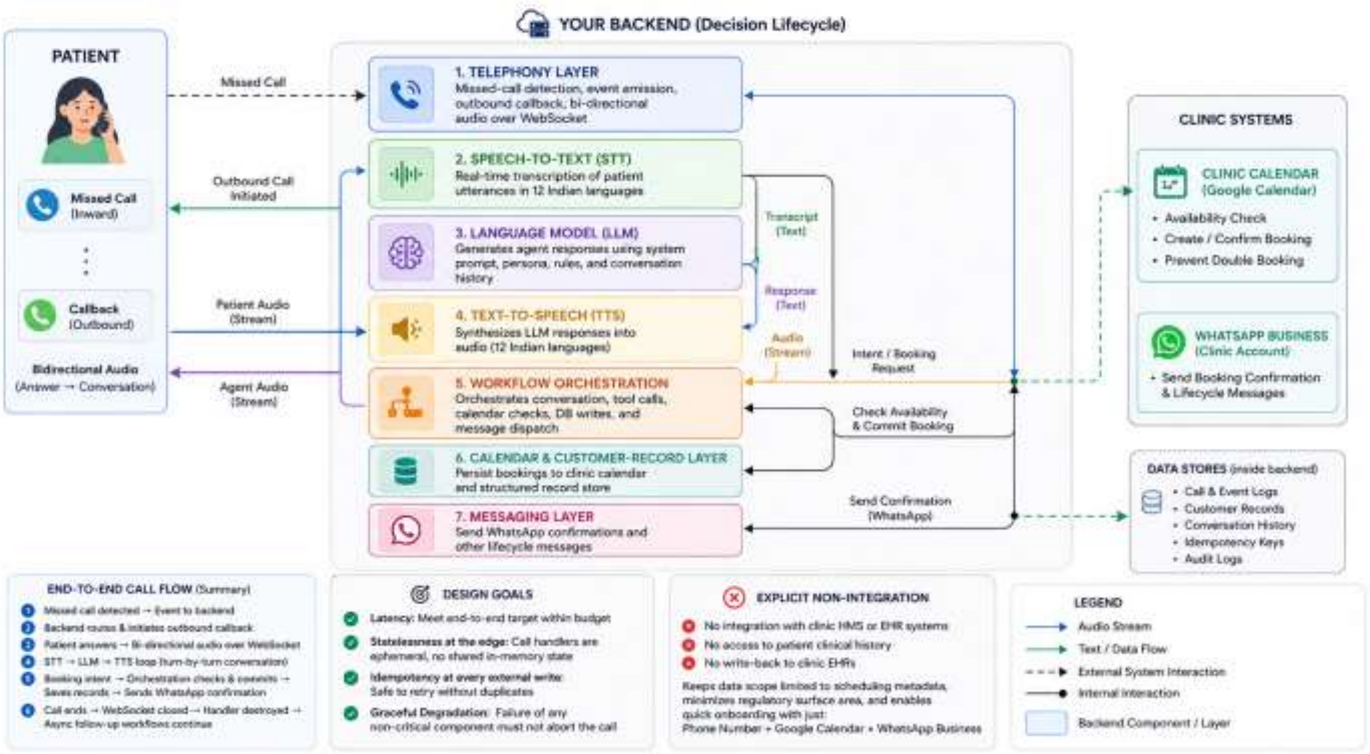


Figure 2: End-to-End System Architecture of the Engageo Conversational Voice AI Pipeline

5.5 Comparison with Existing Approaches

Dimension	Traditional Receptionist	IVR Systems	Rule-Based Chatbots	Marketplace Platforms	Developer Voice AI Tools	Engageo
Response Time	2+ hours (varies)	Immediate (menu)	Minutes	10–30 minutes	Seconds	< 8 seconds
After-Hours Coverage	None	Partial	Full	Limited	Full	Full
Natural Language	Yes	No	Limited	Yes (text only)	Yes	Yes
Multilingual Support	Depends on staff	Limited	Rare	Limited	English-first	12 Indian languages
Code-Switching	Native	No	No	No	Limited	Native
Voice Channel	Yes	Yes	No	No	Yes	Yes

Messaging Channel	Manual	No	Yes	Yes	No	Five-stage lifecycle
Calendar Integration	Manual	None	Manual	Platform-locked	Custom build	Native
Clinic-Side Setup	Hiring	Telephony config	Developer work	Account creation	Engineering team	None
Deployment Time	Weeks (hiring)	Days	Days–weeks	Days	Months	4 days
Patient Channel Preference	Familiar	Disliked by older patients	Preferred by younger	Mixed	Preferred by tech-savvy	Voice + WhatsApp

Table 1: Comparison of Patient Inquiry Handling Approaches for Indian Specialty Clinics

VI.THE VOICE LAYER: AUDIO PIPELINE AND REAL-TIME CONVERSATION

The voice layer is the workhorse of the system. Its responsibilities include taking an incoming audio stream from a patient via telephone call, transforming it into meaningful conversational turns, creating appropriate responses, and streaming the synthesized audio back to the patient, all with low latency that simulates a natural conversational cadence, rather than an intermittent, disconnected dialog.

Audio Fundamentals - All audio in the system is represented as 16-bit linear Pulse-Code Modulation sampled at 8KHz (the standard telephony rate). This is 16 bytes per ms of audio data. So a 200ms piece of audio data, like that described below, would be 3200 bytes. This computation shows up time and time again throughout the system: for buffering, for silence-frame generation, for latency budgeting, and for knowing how much of a streamed audio response has been played.

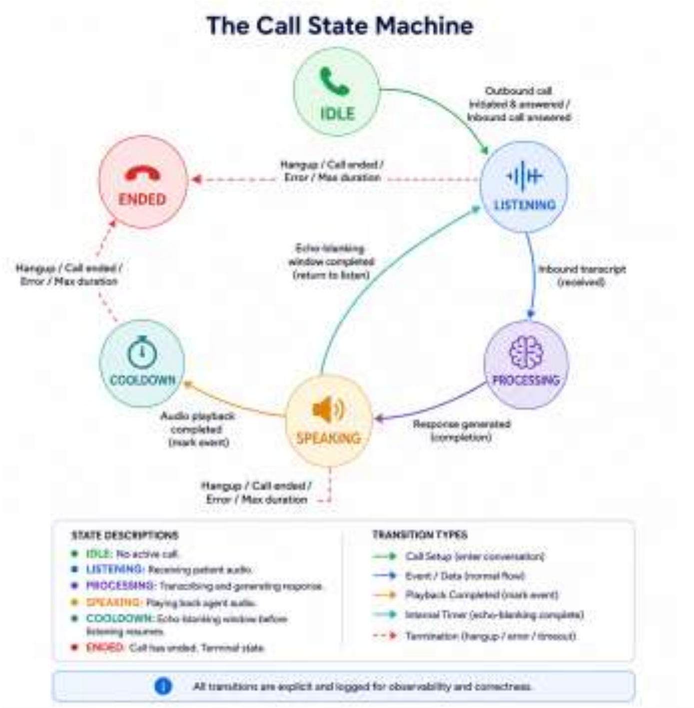
The Streaming protocol - Communication between the telephony provider and the backend takes place over a bi-drectional WebSocket connection. The actual data itself flows as base64-encoded chunks inside JSON messages with event types for stream start, media chunks, marks, and stream stop. The marks are a key part of enabling conversational fluidity. Since the system has no way of knowing when synthesized speech will stop playing over the patient's handset, it sends a uniquely named marker after every utterance and only re-opens the floor for the patient when it receives the corresponding mark event from the telephony provider.

Voice Activity Detection - Many audio chunks contain no speech data. This can be caused by anything from loud background noise, to momentary silences, to environmental sounds (e.g., a dog barking, a door slamming, etc.). Sending any of this noise through to the speech-to-text layer results in wasted processing power, as well as the generation of erroneous transcripts. We use a simple energy-based Voice Activity Detection: for every audio chunk that comes in, we calculate the RMS value of the samples, and compare that against an empirically determined threshold. Anything below the threshold is discarded as silence. The threshold was selected based on observing the distribution of RMS values of actual voice calls and selecting a threshold that appropriately separated ambient noise from speech without cutting off quiet speakers.

Echo Handling - Full-duplex telephony requires that an agent's own transmitted audio is somewhat captured by a patient's microphone and played back to the agent as the patient's audio. The speech-to-text module then correctly transcription this as patient speech and generates spurious transcripts and triggering erroneous interruptions. Phone-side acoustic echo cancellation eliminates but doesn't eliminate this phenomenon; we treat the remainder using an echo-blanking window: after the agent is done speaking, we ignore patient audio for 500ms; the compromise compromise chosen here is that this window is short enough that quickly-responding patients aren't cut off but long enough for the majority of residual echo to have passed before the listening window opens.

Barge-In and Generation IDs - When the agent is speaking, the patient can interrupt and has the agent immediately stop the current utterance, discard any audio already queued, and initiate a new response to the interruption. In a naïve implementation, this can result in a race condition; the language model could be in the process of generating the turn that the patient just interrupted, and this old turn can arrive in after the turn that follows has already commenced. We solve this with monotonically increasing generation identifiers; any call to the language model is stamped with an ID; when the model's response returns, we check its ID against the current ID; if a barge-in has incremented the ID, the old response is simply silently discarded.

Backchannel Filtering - While the agent is speaking, the patient can often interject short affirming phrases "haan", "hmm", "okay", "ji", "achha", "bilkul", "theek hai", etc. These are called backchannels and not interruptions, and signal that the patient is listening, and that the agent should continue. If they are treated as interruptions, the conversation is very disruptive, as the agent's utterance stops every time the patient acknowledges. We use an explicit set of Indian language backchannel tokens and any patient audio transcription matching these while the agent is speaking is suppressed instead of treated as a new turn.



The Call State Machine - Every call is modeled as a finite state machine. All states in the state machine are named explicitly: IDLE, LISTENING, PROCESSING, SPEAKING, COOLDOWN, and ENDED. Transitions between all states are explicit and are logged. An inbound transcript while in LISTENING pushes the call to PROCESSING. Generation of a response completes to move to SPEAKING. Completion of playback (mark event) moves to COOLDOWN, where the echo-blanking window runs, then back to LISTENING. Localizing this logic within the state machine prevents errors from unexpected ordering of conversation events.

Latency Optimization - The total end to end latency budget per turn is about two seconds, between the end of a patient speaking and beginning of the agent's utterance. Several techniques trim this budget: streaming speech to text returns incremental transcripts while the patient is still speaking so that context lookup can begin prior to the completion of utterance. Streaming text to speech returns chunks of speech audio before the generation is complete so that playback can begin. A persistent connection is maintained with the inference service and used across multiple turns of the same call. This prevents handshakes. Each of these methods save on the order of hundreds of milliseconds.

Figure 3: Finite-State Conversational Flow for Real-Time Voice Interaction

VII. THE MULTILINGUAL DESIGN LAYER

This conversational agent handles twelve Indian languages: Hindi, Hinglish, English, Punjabi, Bengali, Tamil, Telugu, Kannada, Malayalam, Gujarati, Marathi, and Odia. The following section details how the agent manages language switching, the particular mixed-script synthesis failure that we observed, and the prompt-level rule we employ to handle this issue.

Language selection and patient-driven switching - Each call begins in Hindi; this is the language most likely to be understood across the geographies of our deployments and avoids any particular patient from being shut out. When the patient's reply is not in Hindi, the agent switches to the detected language at the next turn boundary and continues the rest of the conversation in that language. Speech-to-text detects the language and returns it as part of the transcription. The agent is designed not to switch within an utterance or even mid-utterance, instead only switching between turns. This simplifies the system and avoids the cascading failures that result when an agent attempts to determine distinct intents within an utterance.

While patients primarily spoke Hindi, Hinglish, or English during our pilot tests, their usage of Indian languages was more geographically clustered, with a higher density of Marathi callers from Maharashtra and Punjabi callers from the North, as well as Southern callers tending toward Tamil or Kannada, the long tail of language usage made supporting all twelve languages worthwhile from the beginning.

The Mixed-Script Failure Mode - Early in development, we uncovered a class of failures in text-to-speech output that did not occur in any single-language conversation. When the language model outputted a response containing two non-Latin Indic scripts within the same utterance (for example, two non-Latin Indic languages and scripts combined, as in a response that contained both Devanagari and Gurmukhi characters, or both Devanagari and Tamil characters), the synthesis output was unintelligible: audio contained words cut in half, incorrect prosody and missing syllables. However, synthesizing the exact same utterance in only one of the Indic scripts produces clean audio.

The failure was reproducible and was unrelated to the specific words the language model selected; the model had no issue combining Latin characters (like English words in a Hinglish response) with Indic characters. It was specifically combining two Indic non-Latin scripts within one response that was problematic.

The Script Coherence Rule - We solved this with a prompt-level constraint added to the system prompt that controls language model behavior. The rule had three parts. First, two scripts could be combined: Devanagari and Latin; this is standard for Hinglish and has no problems synthesizing. Second, we established a rule: only one Indic script per response could be used. Thus, a conversation response in Punjabi would only be written in Gurmukhi, with optional Latin English words included in the script, and must exclude Devanagari; a conversation response in Tamil would only be in Tamil characters. Third, script-switching would only occur at turn boundaries (the boundary between end of one response and beginning of another) and not within responses.

The constraint is hardcoded directly into the language model's system prompt and not dealt with as a post-processing step or filtered out by checking the model output. This is intentional because the language model is often good at adhering to structural constraints expressed as simple instructions. This prompt-level method works better than a regex-based post-processing filter for the long tail of tricky edge cases.

Edge Cases - Two cases that required explicit handling were patient names and proper nouns that sometimes appeared in naturally a different script than the contextually current one. These cases were allowed to override the rule due to the fact they often synthesized well, and numbers, dates and times are always provided as a spoken-form string in the appropriate language instead of digits, since number synthesis is unreliable for the Indic scripts and since speaking the numbers as spoken-form strings ("subah gyarah baje" instead of "11 AM") produces better prosody.

To our knowledge, the mixed-script TTS failure has not been previously documented in academic papers about Indic language voice systems, and our practical remedy (commit to one script per turn, switch only between turns) was a small change with a huge positive impact on the system's audio output.

VIII. THE CULTURAL AND CONVERSATIONAL DESIGN FRAMEWORK

It is not enough that the voice agent can accurately process audio and language; the agent must address the patient appropriately. Indian specialty clinic callers, generally older and elderly, often calling on behalf of an ailing relative, and always sensitive to signs of disrespect, call on an Indian clinic's reception line already on edge. A voice agent that can correctly handle both the voice-input and the language-output but does not use the correct tone, honorific, or pacing can create customer frustration even when functional. This section describes design decisions we have made in deploying with real Indian users:

Positioning on the Conversational Spectrum - We position the agent on the conversational spectrum, between the rigidly transactional IVR system and fully imitative human conversation. The agent introduces itself by its name, "Ritu," uses warm but professional language, and attempts not to deceive the user about its AI status. The agent answers directly when a patient explicitly asks whether it is human or AI; the cost of admitting it is an AI is considerably less than the cost of being discovered mid-call that it has deceived them.

Agent Identity - We presented the agent consistently as a female receptionist, "Ritu," using a warm mid-range voice. The female voice and name are a direct result of patient expectations: actual receptionists at these clinics are often women. Having a female agent will align with user expectations, avoiding unnecessary friction in the conversation. A single agent name across all clinics builds familiarity over repeat interactions. We ensured self-reference verbs always take feminine form ("main bol rahi hoon," "main karti hoon") by hardcoding this in the prompt; natural language generation models often default to masculine forms.

Honorifics and Respectful Address - Indian patients across age groups and regions expect to be addressed using honorifics. We instruct the agent to use the "ji" suffix for all names and short answers. This is hard-coded into the prompt and does not depend on the language model's judgment. Especially older patients react very negatively to not being addressed with an honorific, while younger patients don't mind them so the rule errs on the side of over-using honorifics.

Code-Switching Behavior - A single spoken language, Hindi, is typically spoken with some infusion of English words in urban and semi-urban India. We observed this behavior in patients' speech and reproduced it in the agent's speech for naturalness: English is used for technical terms and procedures ("appointment", "consultation", "implant", "checkup") when they are spoken within Hindi sentences; numbers and times use the language of the sentence they are in.

Question Phrasing - Unlike text, voice conversation cannot be flexible. An open-ended question such as "How can I help you?" leads to ambiguous and varied responses from users. We used anchored phrasing consistently: "Aap kab convenient hain subah, dopahar, ya shaam?" rather than "When would you like to come?" The agent is programmed to ask no more than three possible answers per question, a number that a patient can reasonably keep in their working memory during the conversation (five is too many).

Acknowledgment Patterns - The agent acknowledges what the user has said before continuing: after the user states their name, it replies, "Rahul ji, theek hai..." and after the user states the reason for their visit, "Achha, [reason] ke liye...". These acknowledgments create conversational rhythm and signal that the agent is listening, while the absence of them creates a mechanical feel in the agent.

Distress Detection - Some callers will be calling the clinic while they are experiencing intense emotional or physical distress (acute pain, post-procedure issues, fertility-related stress). The system prompt instructs the agent to detect signs of distress and tailor its interaction: it will speak more slowly, include more explicit acknowledgment, and escalate to human staff for urgent clinical issues. The agent is strictly prohibited from giving any medical advice, all clinical questions are to be referred to clinic staff with a clear segue: "Yeh sawal doctor better answer kar sakte hain, main appointment fix karti hoon."

Trust Boundaries - We defined three forbidden interaction styles in the system prompt. The agent must never lie about being an AI when directly asked, confirm the booking until the calendar write is confirmed, or promise the clinic has not authorized (call-backs from doctors, specific price quotes, or timelines for treatments). Each one of these forbidden behaviors, once experienced, leads to a breakdown of trust in the agent which is frequently reported by the caller to the clinic owner.

IX. The Messaging Lifecycle Layer

The voice channel terminates when the call terminates, but the patient remains associated with the clinic. The messaging lifecycle layer ensures the system continues to interact with the patient in the days after the call, through Whatsapp, which has far greater guaranteed deliverability and patient consumption than SMS or email in India. Five workflows execute asynchronously on a scheduling layer, each of them invoked based on a trigger at some point in the patient journey.

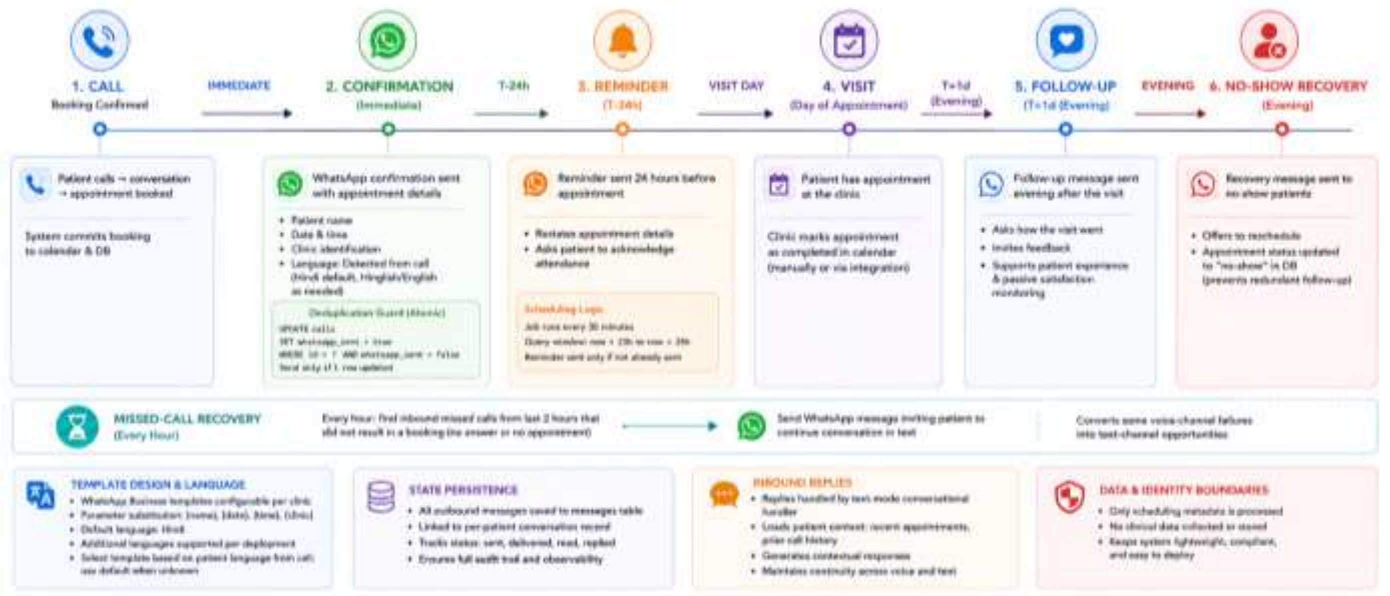


Figure 4: The Messaging Lifecycle Framework for Post-Call Patient Engagement and Recovery

X. Engineering Discipline Patterns

Running against real patients unveiled a class of engineering patterns familiar in themselves but necessary in real-time telephony production. We detail four:

Race Conditions in Calendar Booking - The small time window between availability check and the booking write allows another booking request to steal the calendar slot; at high load even a few hundred milliseconds suffices. We solved this by re-checking calendar availability just before the calendar write inside the same function call. If unavailable, we drop the request and the agent suggests another option on the same turn of the conversation. This adds a single RTT to the calendar service, preventing the class of double-booking errors early releases made.

Atomic Guards Against Duplicate Sends - WhatsApp messages should be sent at most once for any given triggering event. Retries, parallel scheduler invocations and webhook redeliveries will otherwise dupe messages, annoy patients and confuse agents. Each message send is guarded by an UPDATE ... WHERE flag = false RETURNING ...; at most one transaction will return true. The message send is conditional on the transaction success. This patterns moves the responsibility from application logic to the database where it can be enforced under concurrency.

Bounded Concurrent-Call Handling - Every active call consumes network capacity, CPU, and memory. An onslaught of simultaneous calls threatens resource depletion, with potential for cascaded failures. We impose an upper bound on simultaneous calls at the connection acceptance layer, rejecting incoming calls that would exceed it, and saving system resources for calls that do get through. This, combined with graceful shutdown (draining active calls before exit), results in predictable behavior under load.

Phone-Number Normalization - inbound phone numbers appear to the system with an assortment of formats: with or without country code, with or without a leading zero, with or without spaces/hyphens. The same patient dialing from the same number can yield different strings on different calls. All inbound numbers are normalized to canonical E.164 form (+91XXXXXXXXXX) at the system boundary. One operation unifies patient lookup, message dispatch, history search, and every other aspect of the system that relied on consistent strings.

XI. EARLY DEPLOYMENT OBSERVATIONS

Engageo has been deployed in pilot versions at 5 specialty clinics in India across dental, dermatology, hair transplant and fertility specialties. These range from active production deployments to demonstrative installations in test environments. This section presents qualitative findings of the initial phase; quantitative results will follow.

Specialty-specific differences - Hair transplant clinics produced the largest call volume and greatest variety in caller intent; caller intents range from inquiries about pricing to the timeline for procedures to direct appointment requests. Agent capacity to address the full spectrum of these needs without dropping out is more valuable in this segment than in other specialties. Dental and dermatology callers exhibited greater booking intent than other caller types, with less price negotiation and the agent acting mostly transactionally. Fertility callers had long, emotionally complex calls where the patients generally needed reassurance and acknowledgment before being offered a booking. This is where the cultural design choices of slower pacing, and explicit acknowledgments described in Section 8, came to matter the most.

Incidents observed - Three classes of errors re-occurred across all pilots. Firstly, instances of unexpected mid-dialog language switching occurred in which the caller initiated in English and switched mid-call to Tamil for example. The turn-boundary switching logic (Section 7) was correct to detect this switch, but showed the transfer of conversational context across the switch was not clean in some cases. The second was backchannel miss-classification: regional dialect variations on words like "haan" and "achha" were mis-categorized as separate turns instead of backchannel utterances, leading to inappropriate interruption. We iteratively increased the backchannel set as additional variants were found in production logs. The third case involved hand-off mid-call in which a caller passed the phone to a family member who continued the conversation as a different speaker. The agent presently classifies this as continuative, which is almost but not always correct.

What worked well - Decision Lifecycle classification proved robust; all 5 pilots were deployed without modification using this 6-stage classification scheme. After implementation, the script-coherence logic eliminated the previously common synthesis errors that had required manual attention. The 5-stage messaging lifecycle was reliably executed in asynchronous mode across all 5 pilots.

What did not work so well - Region-specific accents on any of the top 3-4 languages presented transcription problems lower than we had predicted, and occasionally caused the agent to give responses that the callers found unnatural in that language. While integration with clinic-side calendars went as expected, WhatsApp business template set-up had greater client-specific setup time required than had been anticipated. These are addressed in Section 12.

XII. DISCUSSION, LIMITATIONS, AND FUTURE WORK

Discussion - The observations arising from the Engageo deployment are relevant for other systems with similar characteristics. One, in high-ticket service segments recovery out performs acquisition; capture a small percentage of missed prior inquiries for a return in revenue with minimal cost that matches the impact of large increments to marketing spend. Two, the integration ceiling for small clinics is both real and binding; any system requiring developer resources from the clinic will not be adopted, regardless of its features. Three, in India, multi-lingual support for appointments cannot be an optional feature; any single-language or even two-language system will lose a considerable fraction of the addressable patients in every region observed.

Limitations - The system and conclusions are bound by several limitations. The system deployment is modest-5 clinics in four specialties and the observations must be considered as such. The multilingual capability, while extensive in principle, is uneven; the top four languages of support performed significantly better than the bottom four. The current system is contingent on the patient returning the callback within a fairly narrow window; if not the patient is lost through the voice channel and must be brought back through messaging. It currently has no integration to an EMR and therefore no ability to leverage patient history for conversation tailoring. The system excels at procedural, well-defined appointment setting; it does not fit general practice well due to its higher variability and more complex triage.

Future Work - Several avenues are possible. Longitudinal, large-scale outcome studies will provide insights into conversion uplift, latency in a live system, and patient satisfaction across more varied deployments. Improvements to the lower-performance languages, either through superior speech recognition models or targeted fine-tuning using Indian clinical corpora, will alleviate current concerns. Integration with existing patient history within the conversation layer will permit recognizing return patients and personalizing conversational flow. The development of proactive features including no-show probability modeling and optimal callback timing based on observed pickup rates, for instance, will allow the system to contribute beyond recovery.

XIII. REFERENCES

[1] J. B. Oldroyd, K. McElheran, and D. Elkington, "The short life of online sales leads," Harvard Business Review, vol. 89, no. 3, pp. 28–30, 2011.
[2] Lead Response Management Study, "The five-minute rule: Conversion impact of lead response time," InsideSales.com / MIT, 2011. [Online]. Available: https://www.insidesales.com/response-time-matters/
[3] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. S. Lau, and E. Coiera, "Conversational agents in healthcare: A systematic review," Journal of the American Medical Informatics Association, vol. 25, no. 9, pp. 1248–1258, Sept. 2018. doi: 10.1093/jamia/ocy072

[4] A. B. Kocaballi, S. Berkovsky, J. C. Quiroz, L. Laranjo, H. L. Tong, D. Rezazadegan, A. Briatore, and E. Coiera, "The personalization of conversational agents in health care: Systematic review," Journal of Medical Internet Research, vol. 21, no. 11, e15360, Nov. 2019. doi: 10.2196/15360
[5] M. Mars and R. E. Scott, "WhatsApp in clinical practice: A literature review," Studies in Health Technology and Informatics, vol. 231, pp. 82–90, 2016. PMID: 27782019
[6] M. Mars, C. Morris, and R. E. Scott, "WhatsApp guidelines — what guidelines? A literature review," Journal of Telemedicine and Telecare, vol. 25, no. 9, pp. 524–529, Oct. 2019. doi: 10.1177/1357633X19873233
[7] A. Diwan, R. Vaideeswaran, S. Shah, A. Singh, S. Raghavan, S. Khare, V. Unni, S. Vyas, A. Rajpuria, C. Yarra, A. Mittal, P. K. Ghosh, P. Jyothi, K. Bali, V. Seshadri, S. Sitaram, S. Bharadwaj, J. Nanavati, R. Nanavati, and K. Sankaranarayanan, "Multilingual and code-switching ASR challenges for low resource Indian languages," in Proc. Interspeech 2021, pp. 2446–2450. [Online]. Available: https://arxiv.org/abs/2104.00235
[8] G. Sreeram, K. Dhawan, and R. Sinha, "Hindi-English code-switching speech corpus," IIT Guwahati, arXiv:1810.00662, 2018. [Online]. Available: https://arxiv.org/abs/1810.00662