

# TERMS OF REFERENCE (TOR)

*Development, Deployment, and Maintenance of an AI Chatbot for Foreign Labor Department (FLD) Services*

## 1. Project Title

Design, Development, Integration, and Deployment of an AI-Powered Chatbot for the Foreign Labor Department (FLD).

## 2. Background

FLD requires an AI-powered chatbot to be integrated into its official website to improve customer service delivery, reduce response times, and automate key processes such as responding to inquiries, providing service status updates, and handling grievances. The chatbot must be connected to EFAAS for user validation and retrieval of relevant records. All service tickets, conversations, and related records shall be lodged and managed within the solution's own built-in ticket-management and CRM backend, which is delivered as an integral part of this engagement.

The chatbot must be fully bilingual for conversational use, supporting Dhivehi (Thaana script) and English across text and voice channels, and must include a conversational voice mode that allows customers to speak with the chatbot and receive spoken replies in either language. Voice input (dictation) must additionally support Bangla, in recognition of the large Bangladeshi expatriate workforce that interacts with FLD.

The chatbot must also be able to interface with any API exposed by FLD — beyond EFAAS — so that current and future FLD systems (work-permit systems, quota systems, payment systems, document-verification systems, etc.) can be plugged in and used by the agent as tools during a conversation.

Authorised FLD staff must be able to monitor live conversations, review chats flagged as low-confidence, and take over any conversation when human intervention is required. The chatbot should provide accurate information, guide customers efficiently, and ensure that unresolved issues are escalated through automated ticket creation, classification, tagging, and assignment to FLD staff within the built-in backend, with AI-generated suggestions on how to handle each ticket.

## 3. Objectives

The objectives of this assignment are to:

- Provide automated responses to customer inquiries through a secure AI chatbot.



- Enable customers to check the status of requests using reference numbers or passport numbers.
- Allow customers to lodge grievances and complaints efficiently.
- Automate ticket creation, classification, tagging, and assignment within the solution's built-in ticket-management and CRM backend.
- Provide AI-generated recommendations to staff on how to handle each ticket, based on ticket class, historical resolutions, and prior human responses.
- Reduce workload on customer support staff by handling routine queries and by reusing previously validated human answers.
- Improve transparency by providing estimated completion timeframes and queue positioning.
- Support continuous learning by improving chatbot accuracy based on staff feedback, prior human responses, and ticket outcomes.
- Deliver fully bilingual (Dhivehi/English) conversational interactions, with Bangla additionally supported for voice input.
- Support low-latency voice input (dictation) and a spoken conversational voice mode, meeting the latency targets defined in Sections 4.3 and 16.
- Enable authorised FLD staff to monitor live conversations, view chats flagged as low-confidence by the AI, and take over any conversation in real time.
- Provide a secure, standards-based integration framework so that the agent can call any FLD-provided API as a tool, now and in the future.

## 4. Scope of Work

The vendor/consultant shall design, develop, deploy, and support an AI chatbot with the following scope:

### 4.1 Core Functional Modules

The chatbot shall provide services in three main categories:

#### A) General Inquiries

- Provide direct answers to customer questions related to FLD services, procedures, and requirements.
- Before generating a new answer, the agent must consult a knowledge base of previously validated human responses (staff replies, closed-ticket resolutions, approved FAQs) and reuse or adapt them where appropriate, so that customers receive answers consistent with what FLD staff have said before.
- If the chatbot is unable to provide an answer, or the answer confidence level is low, the chatbot must prompt the user to create a support ticket.

- The chatbot must allow the customer to describe the issue in detail, submit a complaint or request, and upload supporting documents/attachments.
- The ticket must then be automatically created, classified, tagged, and assigned to FLD staff for follow-up (call back or written response), with a suggested response/action drafted by the agent for the staff member to review.
- The chatbot must be capable of learning from staff responses and ticket resolutions to improve future performance.

## **B) Status Requests**

The chatbot must provide status updates when the customer enters a reference number or a passport number.

The chatbot must validate that the reference/passport number is linked to the authenticated user. Once validated, the chatbot must provide the following:

- Current status of the request.
- Typical timeframe for completion (pre-defined SLA ranges provided by FLD).
- Exact queue position (where available).
- Estimated time to completion, based on queue position and processing capacity.

If the estimated completion time is longer than the standard timeframe, the chatbot must provide a reason (for example, high volume of requests). If delays exceed expected thresholds, the chatbot must alert the FLD system administrator (or designated FLD admin role) to provide an official delay reason. Once a delay reason is provided by FLD, the chatbot must store and reuse the reason for future queries for a defined validity period (configurable timeframe).

**Expedite Request Option:** If the customer wishes to expedite a request, the chatbot must allow the customer to submit an expedite request, provide justification/reason, and attach supporting documents. A ticket must be created, classified and tagged automatically, and assigned to staff for processing, with a suggested handling note.

## **C) Grievances / Complaints**

- The chatbot must enable customers to lodge grievances and complaints.
- The chatbot must support escalation to FLD staff at any time.
- A ticket must be created in the solution's built-in backend, tagged as a grievance, sub-classified where possible (e.g., delay, service quality, staff conduct, wrongful rejection), and assigned to the appropriate staff member/team.
- A service standard of three (3) business days must be applied for response to grievance-related tickets.
- The chatbot must inform customers of the expected response timeline.

## **4.2 Bilingual Language Support (Dhivehi and English)**

The chatbot must be fully bilingual for conversational use. All user-facing interactions must be supported in both Dhivehi (Thaana script) and English. Specifically:

- The chatbot must accept and understand input from customers in both Dhivehi and English.
- The chatbot must respond in the same language the customer is using, or in the language the customer has explicitly selected.
- Language switching must be supported mid-conversation without loss of context.
- All chatbot content, prompts, system messages, error messages, disclaimers, and ticket confirmations must be available in both languages.
- Dhivehi output must render correctly in Thaana script with right-to-left layout support.

Note: Bangla is additionally supported for voice input/dictation as specified in Section 4.3. Bangla is not required for full written bilingual UI parity.

### **4.3 Voice Input and Conversational Voice Mode**

The chatbot must support voice interaction in addition to text. Two distinct voice capabilities are required, and each has its own latency target:

#### **4.3.1 Voice Input / Dictation (Speech-to-Text)**

- Customers must be able to speak their queries or messages, which the chatbot will transcribe into text for the user to review and send.
- Voice input must be supported in three languages: Dhivehi, English, and Bangla.
- Mid-sentence language switching between Dhivehi and English must be supported where technically feasible.
- Voice input is a transcription feature and is separate from the conversational voice mode described in Section 4.3.2.
- Transcription must be streaming and low-latency. The end-to-end partial-transcript latency — measured from the moment the customer stops speaking a word/phrase to the moment the corresponding text appears on screen — must be less than 250 milliseconds under normal network conditions.

#### **4.3.2 Conversational Voice Mode (Voice-to-Voice)**

- A fully spoken conversational mode must be provided, in which the customer speaks to the chatbot and the chatbot replies in spoken voice.
- This mode must be bilingual (Dhivehi and English) and must allow natural turn-taking without the customer needing to type.
- The end-to-end voice-mode latency — measured from the moment the customer finishes speaking to the moment the chatbot starts producing audible audio in reply — must be less than one (1) second under normal network conditions.



- The latency target must be met consistently for both Dhivehi and English, and the vendor must demonstrate this during UAT.
- The voice-mode infrastructure must support a minimum of ten (10) concurrent voice sessions without degradation of the sub-1-second latency target, audio quality, or transcription accuracy. The system must be capable of scaling beyond this baseline as FLD's usage grows, and the vendor must describe the scaling approach in their proposal.

#### **4.3.3 General Voice Requirements**

- Customers must be able to switch freely between text and voice modes during the same session without loss of conversation context.
- Transcripts of voice conversations (both dictation and voice-mode) must be stored alongside text conversations in the backend and must be accessible to authorised FLD staff for review.
- Voice quality, latency, and accuracy must be acceptable for real-time customer service interactions in the supported languages.

#### **4.4 Human Agent Handoff and Low-Confidence Monitoring**

The solution must include an agent console that allows authorised FLD staff to supervise and intervene in live chatbot conversations. Specifically:

- The AI must assign a confidence score to each of its responses based on configurable thresholds.
- Conversations in which the AI produces low-confidence answers must be automatically flagged and surfaced in a dedicated queue within the agent console.
- Authorised FLD staff must be able to view any live conversation (including flagged and non-flagged chats) in real time.
- Staff must be able to take over ("jump into") any conversation seamlessly, at which point the AI pauses and the human agent continues replying to the customer directly.
- The customer must be informed (through a clear indicator) when a human agent has taken over the conversation.
- After human intervention, staff must be able to hand the conversation back to the AI or close it with a ticket summary.
- Staff actions and takeovers must be logged in the audit trail, and staff responses must be captured as potential training feedback for the AI learning module (subject to supervisor approval), so that the same staff answer can be reused by the agent in future similar conversations.

#### **4.5 Reuse of Prior Human Answers**



A distinguishing capability of this solution is the agent's ability to learn from and reference past human responses. The solution must provide:

- A searchable, structured store of prior staff responses and closed-ticket resolutions, indexed by topic, ticket class, and tags.
- Automatic retrieval of the most relevant prior human answers before the agent formulates a response, so that answers remain consistent with FLD's historical positions.
- The ability to cite, in the agent console, which prior human answer(s) informed a given AI reply, for supervisor review.
- Administrator controls to approve, reject, edit, retire, or mark prior answers as authoritative.
- Configurable policies that allow certain sensitive topics to always be answered from approved human answers only, rather than generatively.

#### **4.6 Ticket Classification, Tagging, and Handling Suggestions**

Every ticket created through the chatbot must be automatically processed as follows:

- Classified into a primary class (General Inquiry, Status Request, Expedite Request, Grievance) and into a configurable sub-class specific to FLD's service catalogue (e.g., work-permit renewal, quota query, deposit refund, employer change, medical clearance, etc.).
- Tagged with one or more labels derived from the conversation content (e.g., "delay", "document missing", "payment", "employer dispute", "medical"). The tag vocabulary must be configurable by FLD administrators.
- Accompanied by an AI-generated handling suggestion, presented to the assigned staff member, which includes: a summary of the customer's issue, the most likely cause based on similar past tickets, a recommended next action, relevant SLA and internal references, and — where available — links to prior human answers used as precedent.
- Suggestions must be clearly labelled as AI-generated and must be advisory only; staff retain full authority over the final response.
- Classification, tags, and suggestions must improve over time through supervised learning from staff corrections and closed-ticket outcomes.

### **5. System Integration Requirements**

#### **5.1 EFAAS Integration (Mandatory)**

The chatbot must be integrated with EFAAS, which shall:

- Validate the user identity.
- Retrieve customer contact information.
- Retrieve linked companies, employers, and employees.



- Provide associated records relevant to the request.

**Important Requirement:** The chatbot must not request information from the customer that EFAAS can retrieve, unless a data conflict or missing record is detected. In such cases, the chatbot may request clarification from the customer.

## 5.2 FLD API Integration Framework (Mandatory)

In addition to EFAAS, the agent must be able to interface with any API provided by FLD. The solution must therefore include a generic, configurable API tool-use framework that supports, at minimum:

- Configuration of new APIs through an administrative interface (base URL, authentication method, request/response schemas, rate limits, timeouts) without requiring a new software release.
- Support for common authentication schemes (API keys, OAuth 2.0/OIDC, mTLS, signed requests, HMAC).
- Support for REST and JSON payloads as the primary standard, with SOAP/XML supported where required by legacy FLD systems.
- Schema-aware tool definitions so that the AI agent understands when to call each API, with what parameters, and how to interpret the response.
- Per-API access controls, audit logging of every call made on behalf of a customer, and the ability to enable/disable individual APIs at runtime.
- Safe-by-default behaviour: destructive or state-changing APIs must require explicit configuration and, where appropriate, user confirmation before being invoked.
- The ability to add future FLD APIs (e.g., work-permit systems, quota systems, payment gateways, document-verification services) as new tools without re-engineering the core agent.

## 5.3 Built-in Ticket Management and CRM Backend (Mandatory)

The solution must include a fully built-in ticket management and CRM backend, delivered as an integral part of this engagement. This backend is the sole system of record for chatbot conversations and service tickets. The solution must not rely on, and will not be integrated with, any external or third-party CRM system. The built-in backend must support:

- Automated ticket creation and persistent storage within the solution itself.
- Ticket classification (General Inquiry, Status, Expedite, Grievance) and sub-classification.
- Automated tagging and AI-generated handling suggestions (see Section 4.6).
- Assignment to relevant staff/team.
- Attachment uploads and secure storage.



- Ticket tracking, status updates, and lifecycle management.
- Customer-record management, including contact history, linked companies/employers/employees (as retrieved from EFAAS), and conversation history across sessions.
- A web-based administrative interface for FLD staff to view, search, filter, and manage tickets, customers, and conversations.
- Role-based access controls for administrators, supervisors, and agents.
- Exportable reports and data extracts for FLD's internal use.

The built-in backend must be fully functional without any external CRM connectivity. All features described in this TOR that refer to ticket storage, tagging, assignment, reporting, or customer-record management refer to this built-in backend.

## 6. Ticket Creation and Validation Workflow

All ticket submissions must follow the workflow below:

- Chatbot collects issue details and attachments.
- Chatbot classifies the ticket, applies tags, and generates a handling suggestion (see Section 4.6).
- Chatbot generates a ticket summary draft.
- The ticket summary (including class, tags, and suggested action) must be shown to the user for review, with internal suggestions kept hidden from the customer.
- User must be allowed to verify and amend the customer-facing details.
- Upon confirmation, chatbot submits the ticket to the solution's built-in backend.
- Ticket number is returned to the user and recorded in the chat history.

## 7. AI Learning and Knowledge Improvement

The chatbot must include a controlled learning mechanism such that:

- Staff responses, prior human answers, and ticket resolutions can be used as training and retrieval feedback.
- The chatbot improves its ability to answer recurring inquiries by referencing validated past human responses.
- Classification, tagging, and suggestion quality improve over time based on staff corrections and ticket outcomes.
- Learning must be supervised and configurable by FLD administrators, with an approval workflow for promoting a staff response into the reusable knowledge base.

- Updates to chatbot knowledge must not compromise accuracy or compliance, and administrators must be able to roll back any change.

## 8. User Experience and Interface Requirements

The chatbot shall:

- Be embedded into the FLD website with a user-friendly interface.
- Support document upload functionality.
- Support structured conversation flows and guided prompts.
- Provide clear options for escalation to staff.
- Display reference numbers and ticket confirmations clearly.
- Provide a full bilingual experience (Dhivehi and English) across all interface elements, including language selection, correct Thaana script rendering, and right-to-left layout support for Dhivehi (see Section 4.2).
- Support voice input in Dhivehi, English, and Bangla (Section 4.3.1) and a conversational voice mode in Dhivehi and English (Section 4.3.2), with the ability to switch between text and voice at any point in the conversation.
- Provide a dedicated agent console interface for FLD staff to view active conversations, monitor low-confidence chats, take over conversations when needed, and review AI handling suggestions and cited prior human answers (see Sections 4.4, 4.5, 4.6).

## 9. Security, Privacy, and Compliance Requirements

The chatbot system must comply with FLD security policies and relevant data protection regulations. At minimum, it must include:

- Secure authentication through EFAAS.
- Encrypted communication (HTTPS/TLS) for all channels, including voice streams.
- Role-based access for FLD staff and administrators.
- Secure handling and storage of attachments.
- Logging of ticket creation, chatbot interactions, and all FLD-API tool calls made by the agent.
- Audit trail of all status queries, grievance submissions, agent takeover actions, and any API invocations performed on behalf of a user.
- Data retention settings configurable by FLD.
- Secure storage of voice recordings and transcripts with the same protections as text conversations.



- Credential management for FLD-provided APIs using a secrets vault; no credentials stored in plain text.

## 10. Disclaimer and Customer Support Escalation

The chatbot must display a clear disclaimer stating:

- AI responses may contain errors or inaccuracies.
- Customers should contact FLD directly if uncertain.
- The FLD hotline number must be displayed prominently for immediate assistance.

## 11. Reporting and Analytics Requirements

The system should generate reports and dashboards, including but not limited to:

- Number of chatbot conversations per day/week/month, split by text, dictation, and voice-mode.
- Number of tickets created by category, sub-category, and tag.
- Common inquiry topics, unresolved questions, and tag-frequency trends.
- Average resolution time per ticket category.
- Backlog triggers and delay-reasons frequency.
- Language breakdown of conversations (Dhivehi vs. English; Bangla for voice input).
- Number and rate of low-confidence conversations flagged for human review.
- Escalation rate to human staff, including agent takeover counts and duration.
- Usage and accuracy of AI handling suggestions (accepted, edited, rejected by staff).
- Reuse rate of prior human answers by the agent.
- Voice latency metrics for dictation (<250ms target) and voice-mode (<1s target), with breach counts.
- FLD-API tool-call volume, success rate, and error breakdown per API.
- Customer satisfaction feedback (optional module).

## 12. Deliverables

The selected vendor/consultant shall deliver:

- Requirements analysis and final system design documentation.
- Fully functional AI chatbot deployed on the FLD website.
- EFAAS integration module for validation and data retrieval.
- Generic FLD-API tool-use framework with administrative configuration interface (see Section 5.2).



- Implement a CRM and ticket-management system as part of the solution, with a web-based administrative interface for ticket, tag, customer-record, and conversation management (see Section 5.3).
- Ticket classification, tagging, and AI-generated handling-suggestion module (see Section 4.6).
- Prior-human-answer knowledge base with retrieval, citation, and approval workflow (see Section 4.5).
- Bilingual (Dhivehi and English) language capability across all chatbot interactions, prompts, content, and interface elements.
- Voice input (speech-to-text) in Dhivehi, English, and Bangla with sub-250ms partial-transcript latency.
- Conversational voice mode (voice-to-voice) in Dhivehi and English with sub-1-second end-to-end response latency.
- Human agent console with real-time conversation monitoring, a dedicated queue for low-confidence conversations, live takeover capability, and visibility of AI handling suggestions and cited prior answers.
- Ticket review and confirmation interface.
- Status tracking functionality including queue position and timeframe estimation.
- Grievance submission and escalation module.
- Document upload and attachment management functionality.
- Admin panel for chatbot knowledge management, prior-answer approvals, tag vocabulary, and learning configuration.
- Reporting and analytics dashboard.
- System testing and security validation reports.
- User manuals (Admin and End-User).
- Training session for FLD administrators and support staff.
- Post-deployment support and maintenance plan.

### 13. Implementation Timeline

The implementation shall be completed in phases:

- Phase 1: Requirements gathering and design.
- Phase 2: Chatbot development and website integration.
- Phase 3: EFAAS integration and validation testing.
- Phase 4: FLD-API tool framework implementation and integration testing.
- Phase 5: Built-in ticket-management/CRM backend implementation, classification/tagging/suggestion engine, and automation testing.

- Phase 6: Bilingual language, voice input (Dhivehi/English/Bangla), and conversational voice mode testing — including latency validation against the <250ms and <1s targets.
- Phase 7: Agent console, low-confidence handoff, and prior-human-answer reuse testing.
- Phase 8: UAT (User Acceptance Testing) and final adjustments.
- Phase 9: Go-live deployment.
- Phase 10: Maintenance and continuous improvement.

Final timeline and milestones shall be agreed with FLD upon contract award.

## **14. Roles and Responsibilities**

### **14.1 Vendor/Consultant Responsibilities**

- Develop and deploy the chatbot solution.
- Ensure secure integration with EFAAS and the FLD-API tool framework, and deliver the built-in ticket-management/CRM backend as the sole system of record.
- Deliver bilingual (Dhivehi/English) text and voice-mode capabilities, and tri-lingual (Dhivehi/English/Bangla) voice input.
- Meet and demonstrate the defined voice latency targets (<250ms for dictation, <1s end-to-end for voice mode).
- Deliver the agent console, low-confidence flagging, human takeover functionality, ticket classification/tagging, AI handling suggestions, and prior-human-answer retrieval.
- Provide testing, debugging, and system documentation.
- Provide training and post-launch support.
- Ensure scalability and system stability.
- Provide monitoring and analytics tools.

### **14.2 FLD Responsibilities**

- Provide access to EFAAS API and documentation.
- Provide access and documentation for any additional FLD APIs to be integrated as agent tools.
- Provide service workflows, SLA timeframes, escalation rules, and the ticket-tag vocabulary.
- Provide Dhivehi-language content, terminology, and review support as needed.
- Assign administrators for chatbot content approval, delay-reason validation, confidence-threshold configuration, and prior-answer approval.
- Assign staff users for the agent console and define takeover policies.
- Support UAT and provide timely feedback.

## 15. Technical Requirements

The solution must:

- Support integration via APIs (REST/SOAP as applicable).
- Provide a generic, configurable FLD-API tool framework (see Section 5.2) so that new FLD APIs can be onboarded as agent tools.
- Support file uploads and attachment storage.
- Be scalable to handle high traffic volumes.
- Provide high availability and failover mechanisms.
- Include configurable confidence thresholds for escalation and for surfacing low-confidence chats to human agents.
- Support configurable validity periods for delay reasons.
- Provide speech-to-text in Dhivehi, English, and Bangla meeting the sub-250ms partial-transcript latency target.
- Provide text-to-speech and voice-mode conversation in Dhivehi and English meeting the sub-1-second end-to-end latency target.
- Provide a real-time agent console supporting live conversation monitoring, seamless human takeover, and visibility of AI suggestions and prior-answer citations.
- Support audit logging (including for all FLD-API tool calls) and exportable reports.
- Allow future expansion to other platforms (e.g., mobile app, WhatsApp).

## 16. Performance Requirements

The chatbot must meet the following minimum performance standards:

- Fast response time under normal and peak loads, for both text and voice interactions.
- Ability to handle multiple concurrent users, with support for a minimum of ten (10) concurrent voice-mode sessions without latency, audio-quality, or accuracy degradation.
- Accurate EFAAS validation and correct record retrieval.
- Reliable, consistent execution of FLD-API tool calls with correct parameter handling and graceful failure behaviour.
- Reliable ticket creation, classification, and tagging in the built-in backend without duplication.
- Speech-to-text partial-transcript latency of less than 250 milliseconds, end-to-end, for Dhivehi, English, and Bangla voice input under normal network conditions.
- Voice-mode end-to-end response latency of less than one (1) second, measured from end of user speech to start of chatbot audible reply, for Dhivehi and English.



- Acceptable speech recognition and speech synthesis accuracy in all supported languages.
- Low-latency agent takeover with no loss of conversation context.
- Stable uptime as per agreed SLA (e.g., 99%+).

## 17. Maintenance and Support Requirements

The vendor shall provide:

- Ongoing technical support for a defined period post-deployment.
- Bug fixes and system patching.
- AI model updates and knowledge-base improvements, including improvements to Dhivehi, English, and Bangla voice components.
- Periodic re-training and evaluation of classification, tagging, and suggestion models using accumulated FLD ticket data.
- Performance monitoring and troubleshooting, including continuous monitoring of voice-latency targets and FLD-API tool-call health.
- Support for backend and FLD-API configuration changes, including onboarding of new or replacement FLD APIs.

## 18. Vendor Qualifications and Prior Experience

Given the specialised nature of this engagement — Dhivehi-language AI, low-latency multilingual voice, secure integration with government identity systems, and AI-driven ticketing — the vendor must demonstrate proven experience delivering comparable solutions. Responding vendors must submit, as part of their proposal, evidence of the following:

- Prior deployment of at least one production AI chatbot or conversational-AI solution of similar scale and complexity to this TOR.
- Prior experience building or deploying Dhivehi-language NLP, speech-to-text, or text-to-speech capabilities in production.
- Prior experience delivering low-latency voice AI (voice-to-voice or streaming STT) in a customer-service or contact-centre context, with measurable latency results.
- Prior experience integrating AI agents with government or enterprise identity systems (e.g., EFAAS or equivalent OIDC/OAuth identity providers).
- Prior experience implementing API tool-use frameworks for LLM-based agents against multiple backend systems.
- Prior experience building or delivering ticketing, case-management, or CRM systems as an integrated part of a product.



- References from at least two (2) prior clients for whom the vendor has deployed similar solutions, with contact details made available for verification.
- Case studies or written descriptions of the above deployments, including architecture summaries, scale metrics (users served, conversation volume, languages supported), and lessons learned.
- CVs of the key technical personnel who will be assigned to this project, including their specific roles in prior similar deployments.

Proposals from vendors unable to demonstrate prior experience of the above nature may be treated as non-responsive at FLD's discretion.

## **19. Commercial Proposal and Pricing Requirements**

As part of their response, vendors must submit a detailed commercial proposal. The commercial proposal must clearly present pricing under each of the three categories below. All amounts must be stated in a clearly identified currency, inclusive or exclusive of taxes as clearly indicated, and must list all assumptions on which the pricing is based.

### **19.1 On-Premise Deployment — First-Year Pricing**

A complete price for delivering and operating the solution on FLD-owned / FLD-controlled infrastructure for the first year. This price must cover build, deployment, and the first twelve (12) months of operation following go-live. The breakdown must include, at minimum:

- One-time design, development, and implementation costs.
- Software licences (chatbot platform, built-in ticket/CRM backend, voice components, LLM components, admin/agent console, and any third-party libraries).
- Integration work (EFAAS and FLD-API tool framework onboarding).
- Indicative hardware/infrastructure requirements that FLD must provision (CPU, GPU, RAM, storage, networking), with recommended specifications. Hardware procurement cost is borne by FLD unless explicitly included in the vendor's price.
- Installation, configuration, and commissioning costs.
- Data migration and initial knowledge-base seeding costs (if any).
- Training for FLD administrators and staff.
- Project management and documentation costs.
- First-year support, monitoring, patching, and model-update costs for the on-premise deployment.
- Any recurring third-party costs that FLD must bear directly (e.g., GPU hardware support, OS/database licences), clearly separated from the vendor's own fees.

### **19.2 Cloud Deployment — First-Year Pricing**

A complete price for delivering and operating the solution as a vendor-hosted cloud service for the first year. This price must cover build, deployment, and the first twelve (12) months of operation following go-live. The breakdown must include, at minimum:

- One-time design, development, and implementation costs.
- Software licences and/or subscription fees (chatbot platform, built-in ticket/CRM backend, voice components, LLM components, admin/agent console).
- Cloud infrastructure costs for the first year (compute, GPU, storage, networking, backups, disaster recovery).
- Integration work (EFAAS and FLD-API tool framework onboarding).
- Data migration and initial knowledge-base seeding costs (if any).
- Training for FLD administrators and staff.
- Project management and documentation costs.
- First-year hosting, support, monitoring, patching, and model-update costs.
- Any per-usage charges (e.g., per-conversation, per-minute-of-voice, per-token of LLM usage) with clearly stated unit prices and assumed volumes.
- The cloud region(s) in which the service will be hosted, and any data-residency implications.

### **19.3 Annual Maintenance Contract (AMC) — 1 Year**

A separate 1-year AMC price covering the period after the first year of operation (i.e., from the start of Year 2). The vendor may, at its discretion, quote:

- A single AMC price applicable to either deployment model, or
- Two separate AMC prices — one for on-premise deployment and one for cloud deployment — if the vendor considers them materially different.

Regardless of which approach the vendor takes, the AMC pricing must be clearly itemised and must include, at minimum:

- Ongoing technical support (support hours, response times, and severity-based SLA terms).
- Bug fixes and security patching.
- AI model updates and knowledge-base improvements, including Dhivehi/English/Bangla voice components.
- Periodic re-training and evaluation of classification, tagging, and suggestion models.
- Performance monitoring, including continuous monitoring of voice-latency targets and FLD-API tool-call health.
- Configuration changes and onboarding of new or replacement FLD APIs within reasonable scope.



- Any recurring third-party costs (cloud infrastructure, hosted LLM/STT/TTS services, licences) that are included in, or excluded from, the AMC price — clearly separated.
- Any exclusions (e.g., major new modules, channel expansions, redesigns) that would be treated as out-of-scope and chargeable separately.

#### **19.4 Pricing Presentation Requirements**

- Each of the three price categories (on-premise first year, cloud first year, AMC) must be presented as a separate, clearly labelled section in the commercial proposal.
- All assumptions underlying each price (expected user volumes, conversation volumes, voice-minute volumes, concurrency, data retention period, number of FLD APIs onboarded, number of staff users, etc.) must be explicitly listed.
- All taxes, duties, and statutory charges must be explicitly stated as included or excluded.
- Payment terms and milestone-based payment schedules (where applicable) must be proposed by the vendor.
- Prices must be valid for a minimum period specified by the vendor (not less than 90 days from proposal submission).
- FLD reserves the right to select either the on-premise or the cloud deployment model, and to award the AMC to the same vendor under the quoted terms.

