Technical Documentation

BFSI Call Center AI Assistant

Lendkraft Technologies Solutions Pvt. Ltd – Hiring Task Submission

---

1. Introduction

This project implements a lightweight, compliance-oriented AI assistant designed specifically for Banking, Financial Services, and Insurance (BFSI) call center environments.

The assistant is capable of handling:

- Loan eligibility and application queries
- EMI details and explanations
- Interest rate information
- Payment and transaction queries
- Basic account support
- Policy and compliance-related questions

The system follows a tiered architecture to ensure:

- High accuracy
- Regulatory compliance
- Controlled generation
- Privacy protection
- Fully local execution

The solution runs entirely on a small local language model (SLM) and does not depend on external APIs after initial setup.

2. System Objectives

The system was designed according to the following principles:

2.1 Compliance First

BFSI systems must not hallucinate financial information or expose sensitive data.
The architecture prioritizes curated dataset responses before invoking generation.

2.2 Lightweight & Local

The assistant runs locally using TinyLlama-1.1B and FAISS vector search, ensuring:

- No cloud dependency
- No external API calls
- Full data privacy
- Controlled environment

2.3 Deterministic Responses

Common queries return standardized dataset responses to reduce variability and risk.

2.4 Grounded Policy Responses

Regulatory or complex queries are answered using Retrieval-Augmented Generation (RAG) to prevent hallucination.

3. System Architecture

3.1 High-Level Pipeline

User Query
→ Guardrail Validation
→ Tier 1: Alpaca Dataset Similarity
→ Tier 2: Local Small Language Model
→ Tier 3: RAG Layer (if required)
→ Final Response

4. Core System Components

4.1 Guardrail Layer

The Guardrail Layer validates queries before processing.

It blocks:

- Requests for customer account details
- Credit card numbers
- OTP/password sharing
- Fraud or hacking-related queries
- Sensitive personal information

This prevents:

- PII leakage

- Policy violations
- Hallucinated financial data

If a query violates compliance rules, the system immediately returns a safe rejection response.

4.2 Tier 1 – Alpaca Dataset Similarity Layer

This layer contains 150+ curated BFSI conversation samples in Alpaca format:

- Instruction
- Input
- Output

Each instruction is converted into embeddings using SentenceTransformer and stored in a FAISS vector index.

When a user query is received:

1. The query is embedded.
2. Similarity search is performed.
3. If similarity exceeds threshold, the stored response is returned directly.

Benefits:

- Fast (<1 second)
- Deterministic
- Fully compliant
- No generation required

This ensures standardized responses for common call center queries.

4.3 Tier 2 – Local Small Language Model (SLM)

If no strong dataset match is found, the query is passed to the Small Language Model.

Model Used:
TinyLlama-1.1B-Chat

Characteristics:

- Lightweight (~1.1B parameters)
- CPU compatible
- Deterministic decoding (no randomness)
- Inference-only mode (no training)

The prompt is structured to:

- Maintain professional BFSI tone
- Avoid guessing financial numbers
- Avoid fabricating policies
- Provide concise answers

This tier handles:

- General explanations (e.g., “What is EMI?”)
- Non-policy conversational queries
- Variations not covered in dataset

4.4 Tier 3 – Retrieval-Augmented Generation (RAG)

For complex or regulatory queries (e.g., NPA classification, penalties, RBI guidelines):

1. Query embedding is generated.
2. FAISS policy index is searched.
3. Relevant policy document chunks are retrieved.
4. Retrieved context is passed to SLM.
5. SLM generates a strictly grounded response.

The RAG prompt enforces:

- Answer only from retrieved context
- No assumptions
- No additional explanations
- Safe fallback if context insufficient

This significantly reduces hallucination risk in financial policy queries.

5. Example Data Flow Scenarios

Scenario 1 – Standard Query

Query:
“How can I track my loan application?”

Flow:
Guardrail → Dataset Similarity → Direct Response

Tier Used: Tier 1

Scenario 2 – General Query

Query:
“What is EMI?”

Flow:
Guardrail → No dataset match → SLM Generation

Tier Used: Tier 2

Scenario 3 – Regulatory Query

Query:
“What happens if EMI is unpaid for 90 days?”

Flow:
Guardrail → No dataset match → RAG Retrieval → Grounded SLM Response

Tier Used: Tier 3

---

6. Risk Mitigation Strategy

To ensure compliance and safety:

- Dataset prioritized over generation
- Guardrails block PII requests
- Deterministic decoding prevents unpredictable responses
- RAG grounding prevents fabricated policies
- Strict prompt engineering
- Fully local inference (no external exposure)

---

7. Performance Characteristics (CPU-Based System)

Tested on 16GB RAM CPU system:

- Tier 1: < 1 second
- Tier 2: 20–30 seconds
- Tier 3: 20–40 seconds

Latency is due to CPU-based model inference.
In production, GPU or quantized models would significantly reduce response time.

8. Scalability and Maintainability

The architecture is modular:

- Dataset can be expanded easily
- FAISS indexes can be rebuilt
- Policy documents can be updated
- SLM model can be replaced
- Guardrail rules can be extended

Version control ensures safe updates to policies and dataset.

9. Conclusion

This solution demonstrates a compliance-driven, tiered AI architecture designed for BFSI call center automation.

The system balances:

- Speed (Tier 1 retrieval)
- Intelligence (Tier 2 generation)
- Compliance (Tier 3 grounded responses)
- Security (Guardrail enforcement)
- Local execution (SLM-based)

The architecture ensures safe, scalable, and reliable customer query handling in financial environments.