Eli Lilly BoB Platform – Design Document

# 1. Overview

The Eli Lilly BoB (Bot-of-Bots) Platform is a Streamlit-based orchestration system built to route healthcare queries to the most relevant AI agent using AWS Bedrock. It integrates multiple domain-specific agents, pre-routing logic, semantic similarity, and LLM-based evaluation to return high-confidence responses in real time.

# 2. Objectives

- Automate intelligent routing of healthcare-related queries.  
- Optimize query processing time using pre-routing and parallel execution.  
- Evaluate and score agent responses using embeddings and LLM judgment.  
- Enable clear visualization of metrics and outcomes for decision support.

# 3. System Architecture

The architecture includes:  
- Frontend: Streamlit app for user interaction and visualization.  
- Backend: Python-based logic invoking AWS Bedrock agents and models.  
- Embedding: Amazon Titan for semantic comparison.  
- Evaluation: LLM-as-a-judge using Titan Text Premier.  
- Storage: In-memory state for query history (extendable to DynamoDB).

# 4. Key Components

- Agent Registry: Metadata config for each Bedrock agent.  
- Orchestration Engine: Runs agents in parallel, captures responses.  
- Pre-routing Logic: Bypasses orchestration for known ticket-related keywords.  
- Confidence + Relevance Scoring: Embedding similarity with reference answers and query.  
- LLM Evaluator: Judges best agent output based on reference correctness and clarity.

# 5. Data Flow

1. User enters a query in Streamlit.  
2. If it matches pre-routing keywords, it is sent directly to a ticket agent.  
3. Otherwise, it is passed to all agents in parallel using ThreadPoolExecutor.  
4. Their responses are scored using:  
 - Confidence (vs. reference)  
 - Relevance (vs. query)  
5. The best agent is selected and its answer evaluated using an LLM.  
6. Results, evaluation, and scores are displayed in the UI.

# 6. Technologies Used

- Streamlit (Python UI framework)  
- AWS Bedrock Agents & Runtime  
- Amazon Titan Embedding & LLM models  
- Boto3 SDK for AWS interaction  
- Plotly for visualization

# 7. Future Enhancements

- Add user authentication and session logging  
- Store full chat and metrics in a DynamoDB table  
- Incorporate feedback loop to fine-tune reference answers  
- Support agent chaining or follow-up generation

# 8. Architecture diagram

