Jaicat System Architecture Map (Detailed)

A structured breakdown with swimlanes (module owners) and concrete file paths for every major component.

```
%% Swimlane diagram with explicit ownership
flowchart TD
    %% Lanes
    subgraph L1[🎨 UI Layer (ui/)]
      UI[ui/JaicatUI.py\nTkinter GUI: face, mood, input, status]
      Wake[services/wake_word.py\nWake Word Listener]
    end

    subgraph L2[   Core / App Logic (app/, Core/, conversation/, features/)]
      STT[speech_recognition\nAudio→Text]
      TTS[pyttsx3\nText→Voice]
      NLU[conversation/nlu.py\nZero-shot intent/entity extractor\nModel:
facebook/bart-large-mnli]
      NLP[features/nlp.py\nSentiment + Text Analysis\nTF-IDF, spaCy, LSTM]
      NLG[conversation/nlg.py\nContextual Responder / Generator]
      DM[conversation/dialogue_manager.py\nRAG + Context Manager]
      CTRL[app/controller.py\nMain Orchestrator: mood, routing, memory]
      ROUTER[Core/command_router.py\nIntent Routing]
      CMDS[command/registry.py, builtin.py, handler_table.py\nCommand
Registry]
    end

    subgraph L3[   Services Layer (services/)]
      SP[services/spotify_integration.py\nSpotify Control]
      WEATH[services/weather_service.py\nWeather Data]
      CAL[services/calendar_service.py\nCalendar & Schedule]
      ONED[computer_vision/oneDrive.py\nFile Sync]
      CV[computer_vision/vehicle_detection.py\nYOLOv7/ONNX Vision]
      MEM[services/memory.py\nUser Memory (encrypted JSON)]
    end

    subgraph L4[🧩 Intelligence Layer (machine_learning/, kb/, LLMs)]
      ML[machine_learning/Machine_Learning.py\nIntent/Response Models]
      KB[kb/rag_tool.py + kb_index/\nVector DB / FAISS Retriever]
      OLL[Ollama Local LLM\nHTTP API]
      HF[Hugging Face Transformers\nOnline Models]
    end

    subgraph L5[   Data & Security]
      DB[database/\nLocal Storage]
      LOGS[logs/telemetry]
      SEC[security/\nEncryption & Auth]
    end
    %% Flow connections
    Wake --> STT --> NLU
    UI -->|Manual text| NLU
    NLU --> NLP
    NLP --> CTRL
    CTRL --> ROUTER
    ROUTER --> CMDS
    CMDS -->|executes| SP
    CMDS --> WEATH
    CMDS --> CAL
    CMDS --> ONED
    CMDS --> CV

    CTRL --> DM
    DM --> KB
    DM --> NLG
    NLG --> OLL
    NLG --> HF

    CTRL --> TTS
    CTRL --> UI

    CTRL --> MEM
    CTRL --> LOGS

    ML --> NLU
    ML --> NLP
    KB --> DM

    SEC --> MEM
    DB --> MEM
```

🧩Swimlane Summary

Lane	Area	Key Modules	Purpose
UI Layer	Interface	`ui/JaicatUI.py`, `services/wake_word.py`	Handles user visuals, wake word, and input.
Core / App Logic	Processing & Control	`app/controller.py`, `conversation/*`, `features/nlp.py`, `Core/command_router.py`, `command/*`	Converts input → intent → action. Manages mood, context, and responses.
Services Layer	External APIs	`services/*`, `computer_vision/*`	Provides weather, music, calendar, and vision features.

Lane	Area	Key Modules	Purpose
Intelligence Layer	Knowledge & LLMs	`machine_learning/*`, `kb/*`, `conversation/nlg.py`	Handles reasoning, learning, and creative response generation.
Data & Security	State & Protection	`database/*`, `security/*`, `services/memory.py`	Manages persistence, encryption, and logs.

Typical Voice Query Sequence

1. **Wake Word (Jaicat)** detected → microphone activated.
2. **Speech-to-Text (STT)** → raw text.
3. **NLU** extracts intent/entity (e.g., `play_music` → `Spotify`).
4. **NLP** gauges sentiment → adjusts mood.
5. **Controller** delegates to **Router** → matching command.
6. If command: executes via **Services**.
7. If chat or reasoning: passes to **Dialogue Manager + NLG** → **Ollama / HF model**.
8. Response delivered to **UI + TTS**.
9. **Memory** updated, **logs** stored, context preserved.

This version shows every subsystem with its exact file and ownership lane, clarifying which modules handle which stages of Jaicat’s intelligence and user interaction loop.