**Actus Agent Requirements:**  
The requirement is to design a **multimodal AI Agent system** that integrates:

* Friendly **chat-based querying**
* Deep access to internal structured/unstructured data (media clips, transcripts, logs, alerts, face / logo detections)
* Triggering downstream **AI providers**
* Coordinating with **external agents** (e.g., Gmail)
* Possibly exposing **APIs**

**✅ High-Level Architecture Overview (in classic AI agent terms)**

Below is the system decomposed into components with terminology like **AI Agent**, **Chatbot**, **MCP (Multi-Component Planning)**, etc.

**🧠 Architecture Summary**

**🔹 1. User Interface**

* **Chat UI** (web or embedded) natural language interaction.
* It could be powered by OpenAI-style chat or a fine-tuned model.

**🔹 2. Actus AI Agent (Core Orchestrator)**

This is the **Autonomous AI Agent** or **MCP-capable Chatbot Agent**, which:

* Parses user queries
* Plans tasks
* Gathers data
* Triggers other agents or tools
* Responds in natural language

It uses:

* **LLM (Large Language Model)** for understanding and reasoning
* **Planner/Orchestrator module** (tool selection, memory usage, etc.)
* **Memory module** (short-term for current chat, long-term for logs/history)

**🔹 3. Toolset/Skillset Modules**

These are like "tools" the Actus Agent can use via APIs or local modules:

| **Tool** | **Description** |
| --- | --- |
| **Data Query Tool** | Searches Actus DBs: clips, alerts, logs |
| **Transcription Tool** | Invokes transcription pipeline if transcripts are missing |
| **Face Detection Tool** | Triggers image analysis |
| **Alert Checking Tool** | Queries if alerts were generated |
| **Log Inspection Tool** | Parses logs for system behavior |
| **Email Agent Interface** | Talks to Gmail/Outlook agent to check/confirm sent alerts |

**🔹 4. External Agents**

The Actus Agent can interface with:

* **Gmail Agent** (to confirm notifications)
* **Slack/Teams Agent** (optional)
* **Other enterprise AI agents**

This makes it a **multi-agent system**, enabling richer reasoning.

**🔹 5. APIs & Data Access Layer**

To support this, Actus needs to expose:

| **API Name** | **Function** |
| --- | --- |
| GET /transcripts/{channel}/{date} | For content queries |
| POST /transcribe | To trigger transcription |
| GET /alerts | For alert history |
| GET /logs | For system audit/search |
| POST /invoke-face-detection | Trigger face analysis |
| GET /clip/{id} | Retrieve clip metadata or video |
| POST /send-email | (Optional) Let Actus send direct notifications |

**🔹 6. Knowledge Graph / Metadata Store (Optional but recommended)**

To represent relationships like:

* Clip ↔ Transcript ↔ Alerts ↔ Events ↔ Email Sent

This enables semantic reasoning and easier traceability.

**🖼️ Architecture Diagram**

Here's a conceptual diagram:

[ User (Chat UI) ]

|

v

+----------------------+

| Actus AI Agent | ← Natural Language Interface (LLM)

|----------------------|

| • Planner/Tool-User |

| • Memory |

+----------+-----------+

|

+----------+----------+-------------+--------------+----------------+

| Data Query Tool | Transcriber Tool | Face Detection | Log Search Tool |

| (clips/transcripts)| (if missing) | Trigger | Tool |

+---------------------+-----------------+-----------------+------------------+

|

[Internal APIs / Data Layer]

|

+-----------------+ +-----------------+ +-----------------+

| Transcripts DB | | Alerts / Logs | | Media Store |

+-----------------+ +-----------------+ +-----------------+

|

[External Agents]

|

+-------------------+

| Gmail Agent |

| (check notifications)|

+-------------------+

**🧩 Technologies and Concepts Used**

| **Concept/Tech** | **Role** |
| --- | --- |
| **AI Agent** | Intelligent reasoning and planning |
| **LLM (GPT, Azure OpenAI)** | Natural language understanding |
| **MCP / Toolformer / ReAct** | Agent planning framework |
| **Chatbot UI** | End-user interface |
| **APIs** | Accessing/transmitting internal & external data |
| **Multi-Agent Framework** | For external integrations like Gmail |
| **Triggerable Pipelines** | For on-demand AI data generation (e.g., face detection) |

**🧠 Example Prompt Flow**

**User:**

"Was there any discussion of violence yesterday across our channels?"

**Agent Flow:**

1. **Understands intent** → topic = violence, timeframe = yesterday.
2. **Checks transcripts for keywords/semantics**.
3. If missing transcripts → calls TranscriptionTool.
4. Filters results → returns answer.
5. Optionally, checks logs or alerts if alarms were raised.

**📢 Conclusion: APIs, Orchestration, and Multi-Agent Integration**

We will need to:

* **Expose internal APIs** for data access and AI operation triggers.
* **Build a central Actus AI Agent** with tool-using capability.
* **Integrate external agents** (Gmail, etc.) using agent communication (via REST, email APIs, or agent protocols).
* **Use an LLM-based interface** for user-friendly interaction (GPT-4, Azure OpenAI, or local LLM).