SwarmBot Architecture Overview

mermaid

```mermaid
graph TB
    subgraph "User Interfaces"
        UI1[Tkinter Desktop UI]
        UI2[Dash Web Dashboard<br/>:8050]
        UI3[CLI Interface]
        UI4[EditorWindowGUI]
    end

    subgraph "Entry Points"
        EP1[swarmbot.py<br/>Unified Launcher]
        EP2[main.py<br/>Standard Mode]
        EP3[enhanced_main.py<br/>Enhanced Mode]
    end

    subgraph "Chat System Layer"
        CS1[ChatSession<br/>Basic Chat]
        CS2[EnhancedChatSession<br/>Auto-Tools]
        TM[ToolMatcher<br/>NLP Tool Detection]
    end

    subgraph "Agent System Layer"
        SC[SwarmCoordinator<br/>Task Orchestration]
        AM[AgentManager<br/>Agent Factory]

        subgraph "Specialized Agents"
            A1[ResearchAgent<br/>Web/Doc Analysis]
            A2[CodeAgent<br/>Code Gen/Review]
            A3[TaskAgent<br/>Planning/Distribution]
            A4[MonitorAgent<br/>Performance/Health]
            A5[ValidatorAgent<br/>Quality/Compliance]
        end
    end

    subgraph "Communication Layer"
        AC[AgentCommunication<br/>Message Hub]
        MR[MessageRouter<br/>P2P Routing]
        BC[BroadcastChannel<br/>Pub/Sub]
        MQ[Message Queue<br/>Async Processing]
    end

    subgraph "Core Services"
        LLMC[LLMClient<br/>AI Provider Interface]
        CFG[Configuration<br/>Settings & Env]
        SM[Server Manager<br/>MCP Servers]

        subgraph "LLM Providers"
            LLM1[OpenAI]
            LLM2[Anthropic]
            LLM3[Groq]
            LLM4[Azure]
        end
    end

    subgraph "MCP Integration"
        MCP1[MCP Server 1<br/>Git/GitHub]
        MCP2[MCP Server 2<br/>Filesystem]
        MCP3[MCP Server 3<br/>Browser/Web]
        MCP4[MCP Server N<br/>Custom Tools]
        TOOL[Tool Registry<br/>Available Operations]
    end

    subgraph "Storage Layer"
        DB[(SQLite DB<br/>test.db)]
        LOG[Logging System<br/>swarmbot.log]
        HIST[Task History<br/>Metrics Store]
    end

    subgraph "External Services"
        API1[GitHub API]
        API2[Brave Search]
        API3[N8N Workflows]
        API4[ElevenLabs]
        API5[Exa Search]
    end

    %% User Entry Flow
    UI1 --> EP1
    UI3 --> EP1
    EP1 --> CS1
    EP1 --> CS2
    EP2 --> CS1
    EP3 --> CS2

    %% Chat to Agent Flow
    CS1 --> SC
    CS2 --> TM
    TM --> SC
```

```
CS1 --> LLMC
CS2 --> LLMC

%% Agent Management
AM --> A1
AM --> A2
AM --> A3
AM --> A4
AM --> A5
SC --> AM
SC --> AC

%% Agent Communication
A1 --> AC
A2 --> AC
A3 --> AC
A4 --> AC
A5 --> AC
AC --> MR
AC --> BC
MR --> MQ
BC --> MQ

%% Core Services
LLMC --> LLM1
LLMC --> LLM2
LLMC --> LLM3
LLMC --> LLM4
CFG --> LLMC
CFG --> SM

%% MCP Integration
SM --> MCP1
SM --> MCP2
SM --> MCP3
SM --> MCP4
MCP1 --> TOOL
MCP2 --> TOOL
MCP3 --> TOOL
MCP4 --> TOOL
CS1 --> TOOL
CS2 --> TOOL

%% External APIs
```

```
MCP1 --> API1
MCP3 --> API2
A1 --> API2
A1 --> API5

%% Storage
SC --> DB
A4 --> DB
SC --> HIST
AC --> LOG

%% Monitoring
UI2 --> SC
UI2 --> DB
UI4 --> A2

%% Styling
classDef ui fill:#e1f5fe,stroke:#01579b,stroke-width:2px
classDef entry fill:#f3e5f5,stroke:#4a148c,stroke-width:2px
classDef agent fill:#e8f5e9,stroke:#1b5e20,stroke-width:2px
classDef comm fill:#fff3e0,stroke:#e65100,stroke-width:2px
classDef core fill:#fce4ec,stroke:#880e4f,stroke-width:2px
classDef storage fill:#f3e5f5,stroke:#4a148c,stroke-width:2px
classDef external fill:#ffebee,stroke:#b71c1c,stroke-width:2px

class UI1,UI2,UI3,UI4 ui
class EP1,EP2,EP3 entry
class A1,A2,A3,A4,A5,SC,AM agent
class AC,MR,BC,MQ comm
class LLMC,CFG,SM,CS1,CS2,TM core
class DB,LOG,HIST storage
class API1,API2,API3,API4,API5,MCP1,MCP2,MCP3,MCP4 external
```

Architecture Components

1. User Interfaces

- **Tkinter Desktop UI**: Traditional desktop interface with chat, config panels
- **Dash Web Dashboard**: Real-time monitoring at http://localhost:8050
- **CLI Interface**: Command-line interaction mode
- **EditorWindowGUI**: Multi-language script editor with MCP integration

2. Entry Points

- **swarmbot.py**: Unified launcher supporting all modes
- **main.py**: Standard mode with manual tool execution
- **enhanced_main.py**: Enhanced mode with automatic tool detection

3. Chat System Layer

- **ChatSession**: Basic chat functionality with manual tool calls
- **EnhancedChatSession**: Automatic tool detection and chaining
- **ToolMatcher**: NLP-based tool detection from natural language

4. Agent System Layer

- **SwarmCoordinator**: Orchestrates task distribution and agent coordination
- **AgentManager**: Factory for creating and managing agent instances
- **Specialized Agents**: 5 types with specific capabilities
  - Research: Web research, document analysis
  - Code: Generation, review, refactoring
  - Task: Planning, distribution, tracking
  - Monitor: Performance, health checks, reporting
  - Validator: Quality assurance, compliance

5. Communication Layer

- **AgentCommunication**: Central message hub
- **MessageRouter**: Point-to-point message routing
- **BroadcastChannel**: Pub/sub for multi-agent notifications
- **Message Queue**: Async message processing

6. Core Services

- **LLMClient**: Unified interface for multiple AI providers
- **Configuration**: Environment and settings management
- **Server Manager**: MCP server lifecycle management

7. MCP Integration

- Multiple MCP servers for different capabilities

- Tool Registry for available operations
- Integration with external services

8. Storage Layer

- **SQLite Database**: Persistent storage for metrics and state
- **Logging System**: Comprehensive activity logging
- **Task History**: Performance metrics and analytics

9. External Services

- GitHub API for code operations
- Brave Search for web research
- N8N for workflow automation
- ElevenLabs for voice synthesis
- Exa for advanced search

Data Flow

1. **User Input** → Entry Point → Chat System
2. **Chat System** → LLM Client → AI Provider
3. **Tool Detection** → MCP Servers → External APIs
4. **Task Creation** → SwarmCoordinator → Agent Assignment
5. **Agent Execution** → Communication Layer → Result Aggregation
6. **Monitoring** → Dashboard → Real-time Updates
7. **Storage** → Database → Historical Analysis