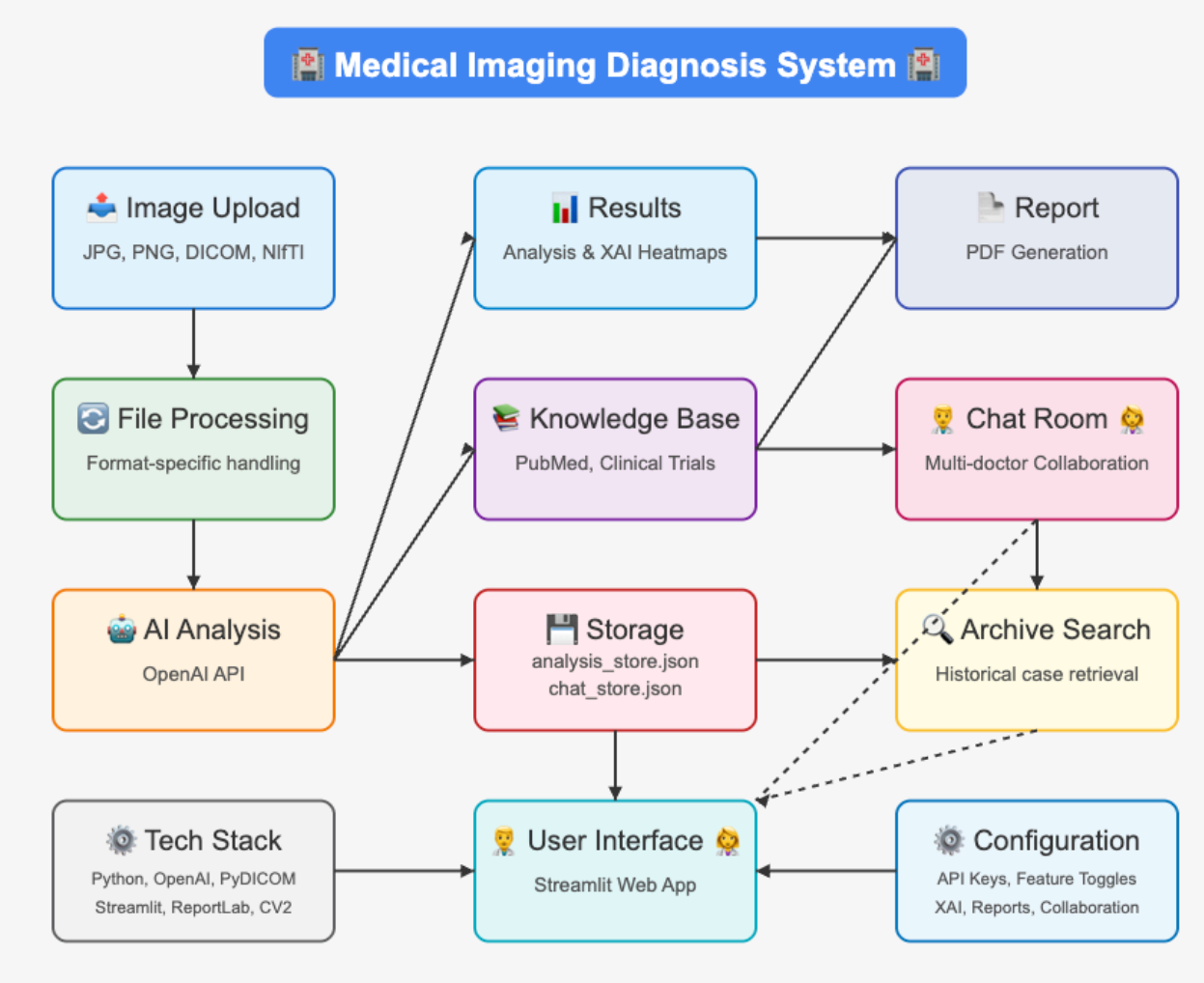
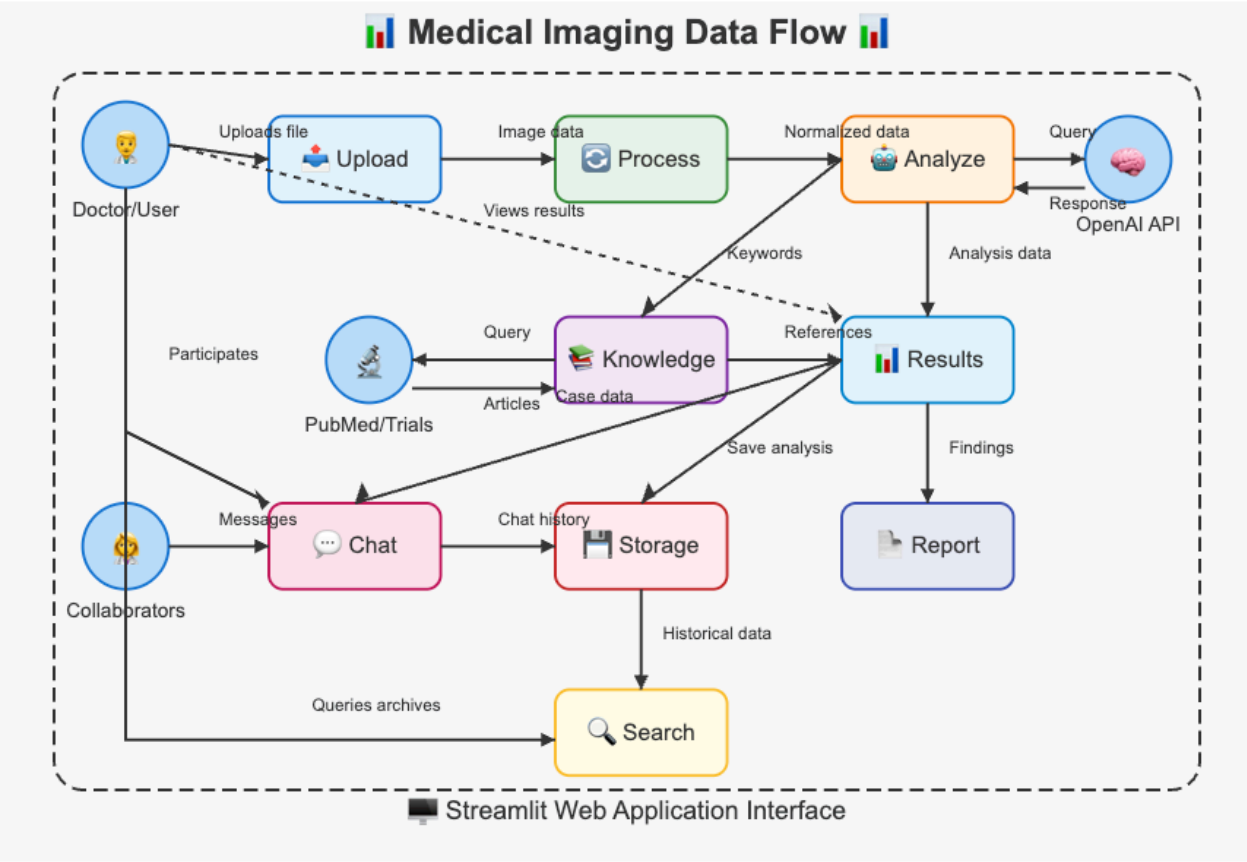
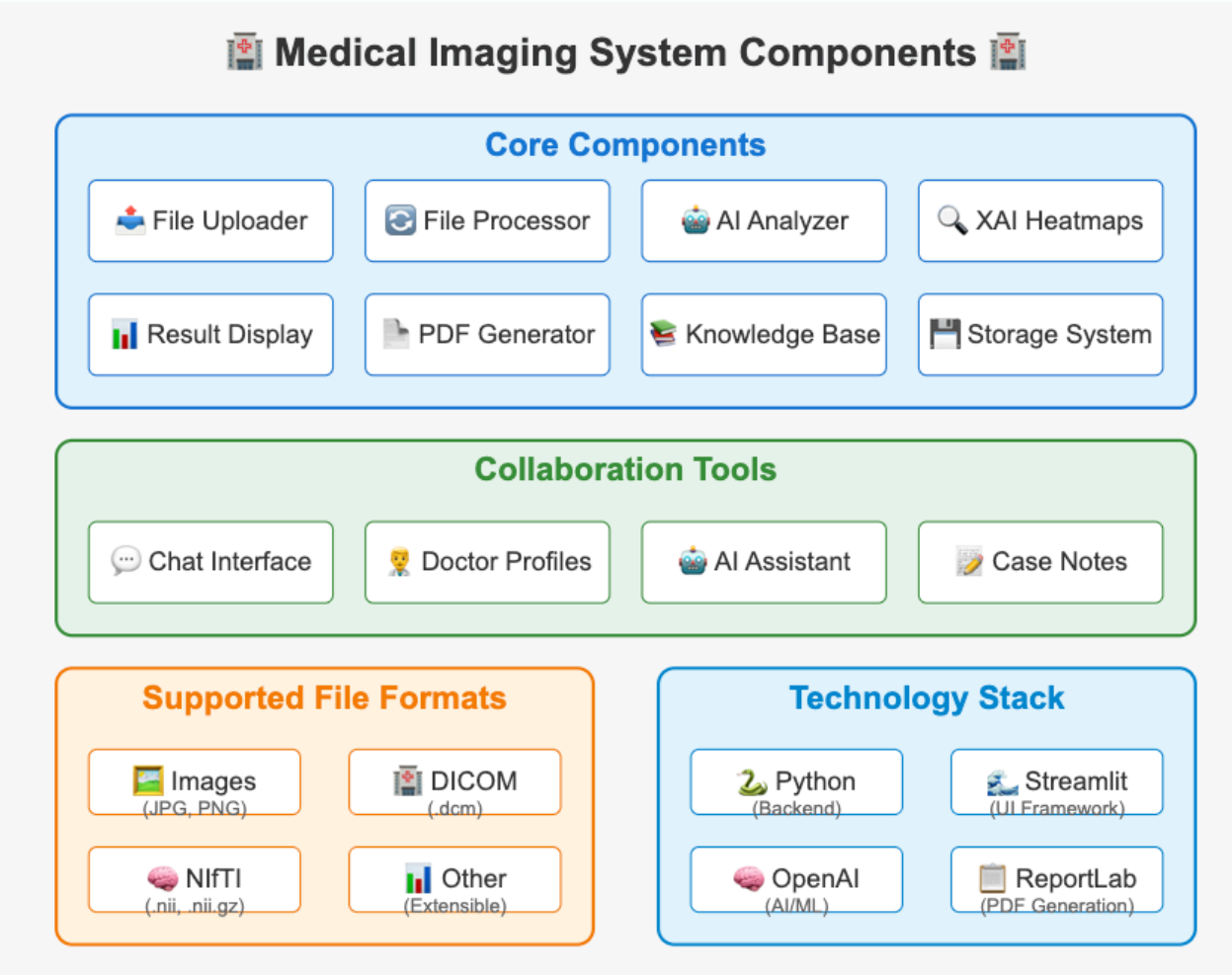
Docter Diagnosis

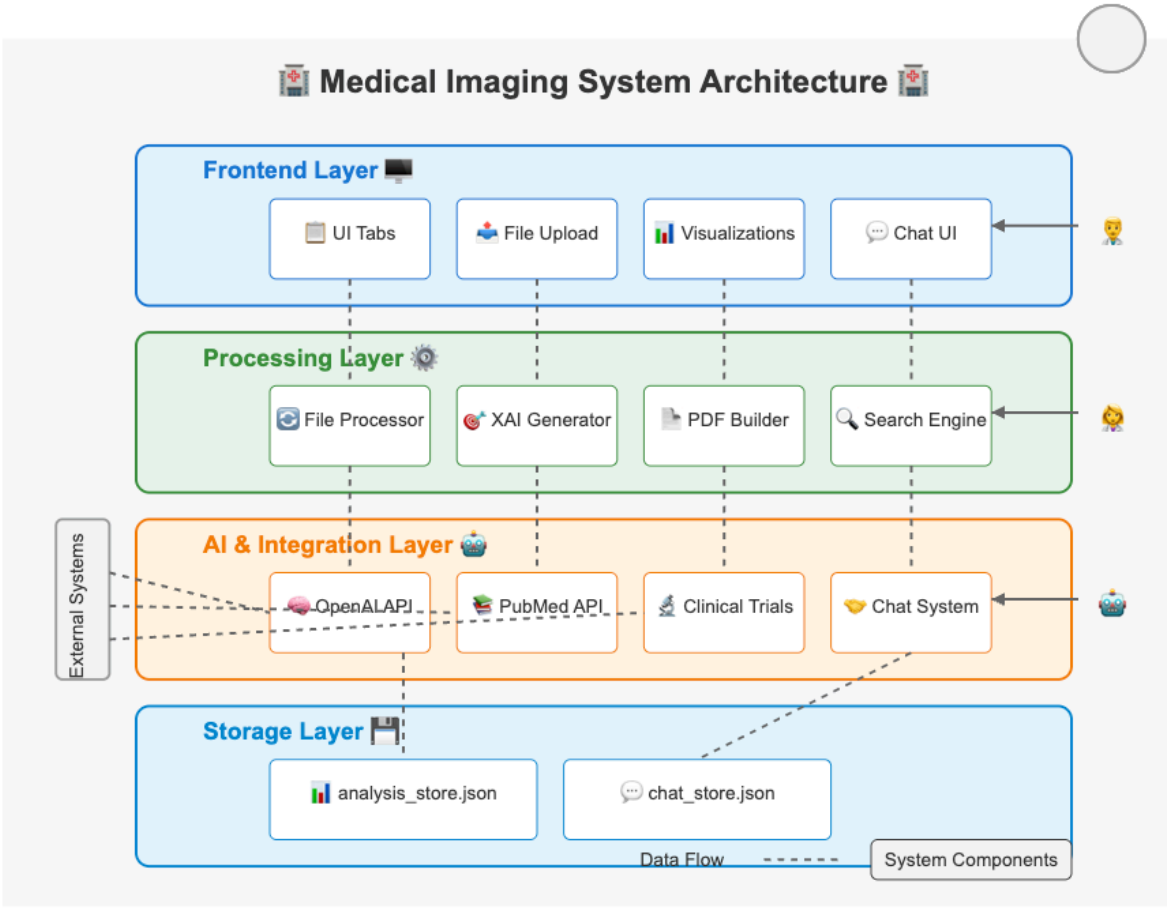
- Working Explanation
- Understanding the Flow and Architecture
- Building the Project
- Deployment

🏥 Advanced Medical Imaging Diagnosis Agent


Medical Imaging Diagnosis System
Image Upload
JPG, PNG, DICOM, NIfTI
File Processing
Format-specific handling
AI Analysis
OpenAI API
Results
Analysis & XAI Heatmaps
Knowledge Base
PubMed, Clinical Trials
Storage
analysis_store.json
chat_store.json
Report
PDF Generation
Chat Room
Multi-doctor Collaboration
Archive Search
Historical case retrieval
Tech Stack
Python, OpenAI, PyDICOM
Streamlit, ReportLab, CV2
User Interface
Streamlit Web App
Configuration
API Keys, Feature Toggles
XAI, Reports, Collaboration


Medical Imaging Data Flow
Doctor/User
Uploads file
Upload
Image data
Process
Normalized data
Analyze
Query
Response
OpenAI API
Views results
Keywords
Analysis data
Participates
PubMed/Trials
Query
Articles
Knowledge
Case data
References
Results
Save analysis
Findings
Collaborators
Messages
Chat
Chat history
Storage
Report
Historical data
Queries archives
Search
Streamlit Web Application Interface


Medical Imaging System Components
Core Components
File Uploader
File Processor
AI Analyzer
XAI Heatmaps
Result Display
PDF Generator
Knowledge Base
Storage System
Collaboration Tools
Chat Interface
Doctor Profiles
AI Assistant
Case Notes
Supported File Formats
Images
(JPG, PNG)
DICOM
(.dcm)
NIfTI
(.nii, .nii.gz)
Other
(Extensible)
Technology Stack
Python
(Backend)
Streamlit
(UI Framework)
OpenAI
(AI/ML)
ReportLab
(PDF Generation)



🔍 System Overview

The Medical Imaging System is an AI-powered Streamlit web application designed to assist healthcare professionals with medical image analysis. This comprehensive platform analyzes diverse medical images, provides AI-generated diagnoses with explanatory visualizations, connects to medical knowledge bases, and enables multi-doctor collaboration.

🧩 Main Components

- 📸 **Image Processing**: Handles multiple formats (DICOM, NIfTI, JPG/PNG)
- 🤖 **AI Analysis**: Leverages OpenAI API for intelligent image interpretation

- 🔆 **XAI Visualization**: Creates heatmaps highlighting influential diagnostic regions
- 📚 **Knowledge Integration**: Connects to PubMed and clinical trials databases
- 👨‍⚕️👩 **Multi-doctor Collaboration**: Real-time chat system for case discussions
- 📄 **Report Generation**: Creates downloadable PDF reports with findings
- 🗃️ **Archive & Search**: Stores and retrieves historical analyses

⚙️ Technology Stack

- 🖥️ **Frontend**: Streamlit for intuitive web interface
- 🔧 **Image Processing**: PyDICOM, Nibabel, OpenCV, PIL
- 🧠 **AI Integration**: OpenAI API for analysis
- 📊 **Data Handling**: Pandas, NumPy for data manipulation
- 📈 **Visualization**: Matplotlib, custom heatmap generation
- 📝 **Report Generation**: ReportLab for PDF creation
- 💾 **Storage**: JSON-based file storage system
- 🔍 **Knowledge Base**: PubMed API (via Biopython)

🔄 Data Flow

1. 👨‍⚕️ **Doctor uploads** medical image file
2. 🔄 **System processes** the file based on format
3. 🧠 **AI analyzes** the image using OpenAI API
4. 📊 **Results display** with visualizations and heatmaps
5. 📚 **Literature fetched** from medical databases
6. 💬 **Collaborative discussions** initiated with other doctors
7. 📄 **Reports generated** for documentation
8. 💾 **Analysis stored** for future reference

👨‍💼 User Journey

1. ⚙️ **Setup**: Configure API keys and enable desired features
2. 📤 **Upload**: Select and upload medical image files
3. 🔍 **Analyze**: Trigger AI analysis of the image
4. 📊 **Review**: Examine findings, heatmaps, and literature
5. 👥 **Collaborate**: Discuss with colleagues through the chat system
6. 📋 **Document**: Generate reports and save to the archive

💪 Benefits for Medical Professionals

- 🩺 **Diagnostic Support**: AI-assisted analysis provides second opinion
- ⏱️ **Efficiency**: Automates literature search and report generation
- 🔍 **Explainability**: Heatmaps show how AI reached conclusions
- 👥 **Collaboration**: Multiple specialists can discuss complex cases
- 📚 **Knowledge Access**: Integration with medical literature databases
- 📝 **Documentation**: Structured reports for medical records

🛠️ Implementation Details

- 🗄️ **Storage System**: Uses two JSON files
  - `analysis_store.json`: Archives all case analyses with metadata
  - `chat_store.json`: Maintains collaboration rooms and messages
- 🔐 **Configuration**: OpenAI API key setup in sidebar
  - Toggle-based feature enablement for XAI, reports, knowledge base
- 🔄 **Mock Services**: Simulated data for development
  - PubMed/clinical trials responses
  - AI-generated analyses

📱 UI Organization

- 🗂️ **Tabs-based interface**:
  - "Image Analysis" for uploads and results
  - "Collaboration" for multi-doctor chat
  - "Search Archives" for historical case lookup
- 📊 **Analysis Display**:
  - Split view with image preview and results
  - Expandable sections for findings and literature
  - XAI visualization with multiple viewing options