**Document Summarization Tool - Problem Statement & Solution Architecture**

An advanced document processing platform that combines multiple AI technologies to automatically extract, analyze, and summarize content from diverse document formats while providing users with flexible deployment options for both cloud and on-premises environments.

### **Component Architecture**

#### **1. User Interface Layer (Streamlit)**

Streamlit Document Processing Interface

├── File Upload & Management

│ ├── Multi-format Support

│ │ ├── PDF Upload Handler

│ │ ├── Word Document Handler

│ │ ├── Text File Handler

│ │ └── Batch Upload Support

│ ├── File Validation

│ │ ├── Format Verification

│ │ ├── Size Validation

│ │ ├── Security Scanning

│ │ └── Metadata Extraction

│ └── Direct Input Options

│ ├── Text Paste Interface

│ ├── URL Import

│ └── Clipboard Integration

├── AI Backend Configuration

│ ├── Backend Selection

│ │ ├── Local AI (Ollama)

│ │ ├── Cloud AI (Hugging Face)

│ │ └── Hybrid Processing

│ ├── Model Management

│ │ ├── Available Models List

│ │ ├── Model Information

│ │ ├── Performance Metrics

│ │ └── Model Switching

│ └── Processing Parameters

│ ├── Summary Length Control

│ ├── Summary Type Selection

│ ├── Quality vs Speed Trade-off

│ └── Custom Prompts

├── Results & Export

│ ├── Summary Display

│ │ ├── Multi-format Results

│ │ ├── Comparison Views

│ │ ├── Quality Metrics

│ │ └── Processing Statistics

│ ├── Export Options

│ │ ├── Text Export

│ │ ├── PDF Generation

│ │ ├── Word Document

│ │ └── API Integration

│ └── History Management

│ ├── Session History

│ ├── Favorite Summaries

│ └── Batch Processing Results

└── User Experience Features

├── Progress Indicators

├── Error Handling

├── Help & Documentation

└── Accessibility Features

#### **2. Document Processing Layer**

Document Analysis Engine

├── Format Detection System

│ ├── MIME Type Analysis

│ │ ├── File Header Inspection

│ │ ├── Extension Validation

│ │ ├── Content Analysis

│ │ └── Binary vs Text Detection

│ ├── Structure Recognition

│ │ ├── Document Layout Analysis

│ │ ├── Section Identification

│ │ ├── Hierarchy Detection

│ │ └── Metadata Extraction

│ └── Quality Assessment

│ ├── Readability Scoring

│ ├── Corruption Detection

│ ├── Language Identification

│ └── Encoding Validation

├── Text Extraction Engine

│ ├── PDF Processing

│ │ ├── PyPDF2 Integration

│ │ ├── Text Layer Extraction

│ │ ├── OCR Fallback (Future)

│ │ ├── Table Extraction

│ │ └── Image Description

│ ├── Word Document Processing

│ │ ├── python-docx Integration

│ │ ├── Paragraph Extraction

│ │ ├── Style Preservation

│ │ ├── Header/Footer Handling

│ │ └── Embedded Objects

│ ├── Text File Processing

│ │ ├── Encoding Detection

│ │ ├── Character Set Conversion

│ │ ├── Line Ending Normalization

│ │ └── Format Preservation

│ └── Rich Text Processing

│ ├── RTF Parser

│ ├── Markdown Support

│ ├── HTML Content

│ └── XML Processing

├── Content Analysis

│ ├── Linguistic Analysis

│ │ ├── Sentence Tokenization

│ │ ├── Paragraph Segmentation

│ │ ├── Topic Modeling

│ │ └── Sentiment Analysis

│ ├── Structural Analysis

│ │ ├── Document Sections

│ │ ├── Key Points Identification

│ │ ├── Citation Recognition

│ │ └── Reference Extraction

│ └── Content Classification

│ ├── Document Type Detection

│ ├── Domain Classification

│ ├── Language Detection

│ └── Complexity Assessment

└── Quality Control

├── Content Validation

├── Error Detection

├── Completeness Check

└── Processing Metrics

#### **3. AI Processing Layer**

Ollama Local Processing

├── Model Management

│ ├── Available Models

│ │ ├── Llama2 (General Purpose)

│ │ ├── Mistral (High Performance)

│ │ ├── CodeLlama (Technical Documents)

│ │ └── Specialized Models

│ ├── Model Operations

│ │ ├── Dynamic Loading

│ │ ├── Memory Management

│ │ ├── Performance Monitoring

│ │ └── Resource Optimization

│ └── Custom Configuration

│ ├── Temperature Settings

│ ├── Token Limits

│ ├── Prompt Engineering

│ └── Response Formatting

├── Summarization Engine

│ ├── Prompt Engineering

│ │ ├── Context-Aware Prompts

│ │ ├── Length-Specific Templates

│ │ ├── Domain Adaptation

│ │ └── Quality Instructions

│ ├── Processing Pipeline

│ │ ├── Text Chunking

│ │ ├── Context Assembly

│ │ ├── Inference Execution

│ │ └── Response Aggregation

│ └── Quality Control

│ ├── Output Validation

│ ├── Coherence Checking

│ ├── Length Verification

│ └── Format Compliance

└── Performance Optimization

├── Batch Processing

├── Parallel Execution

├── Memory Optimization

└── Response Caching

##### **Cloud AI Engine (Hugging Face)**

Hugging Face Transformers

├── Model Selection

│ ├── BART (facebook/bart-large-cnn)

│ │ ├── News Summarization

│ │ ├── General Documents

│ │ ├── High Quality Output

│ │ └── Robust Performance

│ ├── T5 Models (t5-small, t5-base)

│ │ ├── Versatile Applications

│ │ ├── Customizable Length

│ │ ├── Good Performance

│ │ └── Resource Efficient

│ ├── Pegasus (google/pegasus-xsum)

│ │ ├── News Optimization

│ │ ├── Extractive-Style

│ │ ├── Factual Accuracy

│ │ └── Concise Output

│ └── Custom Models

│ ├── Domain-Specific

│ ├── Fine-tuned

│ ├── Specialized Tasks

│ └── Performance Optimized

├── Processing Pipeline

│ ├── Model Loading

│ │ ├── Dynamic Model Selection

│ │ ├── GPU Utilization

│ │ ├── Memory Management

│ │ └── Caching Strategy

│ ├── Text Processing

│ │ ├── Tokenization

│ │ ├── Chunking Strategy

│ │ ├── Context Window Management

│ │ └── Overlap Handling

│ ├── Inference Engine

│ │ ├── Batch Processing

│ │ ├── Progress Tracking

│ │ ├── Error Handling

│ │ └── Quality Validation

│ └── Post-processing

│ ├── Result Aggregation

│ ├── Format Standardization

│ ├── Quality Assessment

│ └── Output Validation

└── Performance Features

├── GPU Acceleration

├── Model Caching

├── Parallel Processing

└── Memory Optimization

##### **Extractive Summarization Engine**

Extractive Summary System

├── Sentence Analysis

│ ├── Embedding Generation

│ │ ├── Sentence Transformers

│ │ ├── Vector Representation

│ │ ├── Semantic Analysis

│ │ └── Context Awareness

│ ├── Similarity Computation

│ │ ├── Cosine Similarity

│ │ ├── Euclidean Distance

│ │ ├── Manhattan Distance

│ │ └── Custom Metrics

│ └── Clustering Algorithm

│ ├── K-Means Clustering

│ ├── Hierarchical Clustering

│ ├── DBSCAN

│ └── Custom Clustering

├── Selection Strategy

│ ├── Centrality-Based Selection

│ │ ├── Cluster Centers

│ │ ├── Representative Sentences

│ │ ├── Diversity Consideration

│ │ └── Redundancy Removal

│ ├── Importance Scoring

│ │ ├── TF-IDF Weighting

│ │ ├── Position-Based Scoring

│ │ ├── Length Normalization

│ │ └── Keyword Density

│ ├── Quality Filtering

│ │ ├── Sentence Completeness

│ │ ├── Grammar Validation

│ │ ├── Coherence Checking

│ │ └── Relevance Assessment

│ └── Final Selection

│ ├── Top-K Selection

│ ├── Threshold-Based

│ ├── Proportional Selection

│ └── Balanced Representation

└── Output Generation

├── Sentence Ordering

├── Transition Smoothing

├── Format Standardization

└── Quality Validation

## **Data Flow Architecture**

### **Document Processing Workflow**

1. Document Upload

├── File Validation

├── Format Detection

└── Security Scanning

↓

2. Content Extraction

├── Format-Specific Processing

├── Text Extraction

├── Metadata Collection

└── Quality Assessment

↓

3. Content Analysis

├── Language Detection

├── Structure Analysis

├── Topic Identification

└── Complexity Assessment

↓

4. Preprocessing

├── Text Cleaning

├── Normalization

├── Chunking Strategy

└── Context Preparation

↓

5. AI Processing Selection

├── Backend Choice (Local/Cloud)

├── Model Selection

├── Parameter Configuration

└── Processing Strategy

↓

6. Summarization Execution

├── Abstractive Processing

├── Extractive Processing

├── Hybrid Approach

└── Quality Control

↓

7. Result Processing

├── Output Validation

├── Format Standardization

├── Quality Metrics

└── Performance Statistics

↓

8. Presentation & Export

├── Summary Display

├── Comparison Views

├── Export Options

└── History Storage

### **AI Backend Integration Flow**

Local AI (Ollama) Flow:

Document → Chunking → Local Model → Summary Generation → Quality Check → Result

Cloud AI (Hugging Face) Flow:

Document → Preprocessing → Model Loading → Tokenization → Inference → Post-processing → Result

Extractive Flow:

Document → Sentence Extraction → Embedding → Clustering → Selection → Ordering → Result

Hybrid Flow:

Document → Multi-path Processing → Result Combination → Quality Assessment → Final Output