# **OneKey VR Automation - Complete Implementation Guide**

## **Table of Contents**

1. [Architecture Overview](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#architecture-overview)
2. [Prerequisites & Setup](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#prerequisites--setup)
3. [Project Structure](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#project-structure)
4. [Step-by-Step Implementation](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#step-by-step-implementation)
5. [Tool Integration Guidelines](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#tool-integration-guidelines)
6. [Configuration Management](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#configuration-management)
7. [Testing Strategy](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#testing-strategy)
8. [Deployment Guidelines](https://claude.ai/chat/21dc0789-eb64-48ef-9ea1-95aa3f21aa19#deployment-guidelines)

## **Architecture Overview**

### **Understanding the Hybrid Design**

* **External Deterministic Preprocessing**: A separate service handles VR data parsing, individual disambiguation, and verification requirement generation using rule-based logic with LLM tools
* **Three-Agent Workflow**: After preprocessing, clean data flows through Supervisor → Search → Summary agents
* **LLM-Driven Decisions**: All intelligence comes from carefully crafted prompts, not hardcoded rules
* **Tool Ownership Model**: Search Agent internally owns and controls 5 specialized tools

### **Agent Responsibilities (From Design Document)**

* **Supervisor Agent**: Workflow orchestration, state management, DBO interface formatting
* **Search Agent**: Tool selection, execution strategy, stopping criteria, owns 5 tools internally
* **Summary Agent**: AI-powered summarization, confidence scoring, recommendation generation

## **Prerequisites & Setup**

### **What You Need to Install**

1. **Python 3.11+** - Required for async/await patterns and latest LangChain features
2. **Azure OpenAI Service** - For GPT-4 model access with your deployment
3. **LangChain/LangGraph Libraries** - Core framework for agent orchestration
4. **Selenium WebDriver** - For website automation in your tools
5. **External API Access** - LinkedIn, web search APIs as needed
6. **Development Tools** - pytest for testing, docker for deployment

### **Environment Configuration Needed**

* Create .env files for development, staging, production environments
* Set up Azure OpenAI endpoint, API keys, deployment names
* Configure external service API keys (LinkedIn, web search, etc.)
* Set up monitoring and logging configurations

## **Project Structure**

### **Directory Organization Explanation**

onekey\_vr\_automation/

├── main.py # Entry point - creates and runs the workflow

├── config/ # All configuration files

│ ├── settings.py # Configuration loading logic

│ ├── website\_configs/ # Website-specific form mappings

│ └── tool\_selection\_rules.json # Rules for LLM tool selection prompts

├── agents/ # Three main agents

│ ├── supervisor\_agent.py # Workflow orchestrator & DBO interface

│ ├── search\_agent.py # Tool owner with 5 internal tools

│ └── summary\_agent.py # AI summarizer and recommender

├── tools/ # YOUR EXISTING TOOL FILES GO HERE

│ ├── france\_trusted\_sources.py # Your France tool implementation

│ ├── italy\_trusted\_sources.py # Your Italy tool implementation

│ ├── hospital\_sources.py # Your Hospital tool implementation

│ ├── linkedin\_professional.py # Your LinkedIn tool implementation

│ └── untrusted\_web\_search.py # Your Web Search tool implementation

├── utils/ # Support utilities

│ ├── state\_models.py # AgentState definition and data models

│ ├── routing\_functions.py # LangGraph conditional routing logic

│ └── external\_integrations.py # Interfaces to preprocessing & DBO systems

├── prompts/ # LLM prompt templates

│ ├── supervisor\_prompts.py # Prompts for workflow orchestration

│ ├── search\_prompts.py # Prompts for tool selection & stopping

│ └── summary\_prompts.py # Prompts for summarization & recommendations

└── tests/ # Test files

├── test\_agents.py # Unit tests for each agent

└── test\_workflow.py # Integration tests for full workflow

## **Step-by-Step Implementation**

### **Step 1: Create State Models**

**What to do**: Define the shared state structure that flows between agents **Key elements needed**:

* Input fields for preprocessed data from external service
* Search Agent state (selected tools, execution order, results)
* Summary Agent state (summaries, confidence scores, recommendations)
* Supervisor state (workflow status, DBO output, error context)
* Use TypedDict for type safety and Annotated lists for state accumulation

### **Step 2: Set Up Configuration Management**

**What to do**: Create a system to load and manage configurations **Key elements needed**:

* Configuration loader class that reads JSON files
* Website-specific configurations for form field mappings
* Tool selection rules that guide LLM decision making
* Environment-specific settings (dev, staging, production)

### **Step 3: Create Website Configuration Files**

**What to do**: Define how to transform VR data into website-specific input formats **For each trusted source website, specify**:

* Required and optional form fields
* Mapping from VR data fields to website form fields
* CSS selectors for form elements
* Dropdown value mappings (e.g., Milano → MI for province)
* Special handling rules for that website

### **Step 4: Design LLM Prompt Templates**

**What to do**: Create structured prompts that guide LLM decision making **For Search Agent prompts**:

* Tool selection prompt: Guides LLM to choose appropriate tools based on geographic region, institution type, confidence requirements
* Stopping decision prompt: Helps LLM decide when to stop searching based on confidence and remaining tools

**For Summary Agent prompts**:

* Summarization prompt: Guides intelligent summary creation adapting to data complexity
* Confidence assessment prompt: Helps LLM calculate confidence scores and flag manual review needs
* Recommendation prompt: Guides generation of specific, actionable recommendations

**For Supervisor Agent prompts**:

* Workflow orchestration prompt: Guides decisions about which agent to invoke next
* DBO formatting prompt: Structures output for database operator review interface
* Error handling prompt: Guides recovery strategies when things go wrong

### **Step 5: Implement Search Agent**

**What to do**: Create the agent that owns and controls the 5 tools **Key responsibilities**:

* Initialize and configure your 5 existing tool instances in the \_\_init\_\_ method
* Implement intelligent\_tool\_selection() method that uses LLM prompts to choose tools
* Implement execute\_owned\_tool() method that calls your tool's main search method
* Implement adaptive\_stopping\_decision() method that uses LLM to decide when to stop
* Handle tool failures gracefully and continue with remaining tools

**Integration points for your tools**:

* Import your existing tool classes in the initialization methods
* Pass website configurations to your tools during initialization
* Call your tool's main method (whatever you named it: search, execute, run, etc.)
* Standardize the return format from your tools into the expected structure

### **Step 6: Implement Summary Agent**

**What to do**: Create the AI-powered summarizer **Key responsibilities**:

* Implement intelligent\_summarization() that uses LLM to create summaries adapting to data complexity
* Implement generate\_confidence\_scores() that calculates confidence and flags manual review needs
* Implement create\_actionable\_recommendations() that generates specific actions for DBOs
* Parse LLM JSON responses and handle parsing errors gracefully

### **Step 7: Implement Supervisor Agent**

**What to do**: Create the workflow orchestrator and DBO interface manager **Key responsibilities**:

* Implement receive\_preprocessed\_input() to validate input from external preprocessing
* Implement orchestrate\_workflow() that uses LLM to decide which agent runs next
* Implement format\_for\_dbo\_interface() that transforms summary into DBO-friendly format
* Implement handle\_errors\_and\_escalation() for error recovery strategies
* Manage workflow state transitions based on LLM decisions

### **Step 8: Create Routing Functions**

**What to do**: Define conditional routing logic for LangGraph **Key functions needed**:

* supervisor\_workflow\_router(): Routes based on workflow status and error conditions
* search\_continuation\_router(): Decides whether to continue search or move to summary
* tool\_execution\_router(): Checks if more tools need to be executed
* These functions examine the current state and return string values that determine the next node

### **Step 9: Build the Main Workflow**

**What to do**: Assemble everything into a LangGraph workflow **Key elements**:

* Create StateGraph with AgentState
* Add nodes for each agent method
* Define edges with conditional routing
* Ensure Supervisor orchestrates the overall flow
* Compile the graph into an executable application

### **Step 10: Create External Integration Points**

**What to do**: Interface with external systems **For preprocessing integration**:

* Create function to receive structured data from external deterministic preprocessing service
* Handle the transformation from external format to your AgentState format

**For DBO interface integration**:

* Create function to submit formatted results to DBO review system
* Handle authentication and proper API formatting for DBO system

## **Tool Integration Guidelines**

### **Your Existing Tool File Requirements**

**What your tools need to provide**:

* **Consistent interface**: Each tool should have the same method signature (e.g., search(verification\_requirements))
* **Configuration acceptance**: Tools should accept website configuration during initialization
* **Standard return format**: All tools should return results in the same structure:
  + status: "completed" or "failed"
  + confidence: float between 0.0 and 1.0
  + findings: dictionary with search results
  + execution\_time: float in seconds
  + error\_message: string if failed, None if successful

### **Integration Steps for Your Tools**

1. **Import in Search Agent**: Add import statements in the tool initialization methods
2. **Pass configurations**: Provide website configurations loaded from JSON files
3. **Handle input transformation**: Your tools should transform verification requirements into website-specific input
4. **Implement dropdown handling**: Your tools should handle intelligent dropdown selection based on configuration
5. **Return standardized results**: Ensure your tools return data in the expected format

### **Tool Internal Preprocessing**

**What your tools should handle internally**:

* Transform verification requirements into website-specific form input
* Apply geographic mappings (Milano → MI for province codes)
* Handle dropdown selections with fuzzy matching and fallbacks
* Execute Selenium automation with error handling
* Extract and clean results from websites
* Calculate confidence scores based on result quality

## **Configuration Management**

### **Website Configuration Structure**

**What you need to define for each website**:

* **input\_mapping**: How to map VR data fields to website form fields
* **form\_selectors**: CSS selectors for finding form elements
* **geographic\_mappings**: City to province/region mappings
* **dropdown\_handling**: Rules for handling dropdown selections
* **error\_patterns**: How to detect and handle website errors

### **Tool Selection Rules**

**What to configure for LLM tool selection**:

* **Geographic rules**: Which tools to use for each country/region
* **Context rules**: Additional tools based on institution type
* **Confidence requirements**: Tool combinations for different confidence needs
* **Execution priorities**: Default order for tool execution

### **Environment-Specific Settings**

**What to configure for different environments**:

* API endpoints and credentials for each environment
* Logging levels and monitoring settings
* Timeout values and retry configurations
* Feature flags for gradual rollout

## **Testing Strategy**

### **Unit Testing Requirements**

**What to test for each agent**:

* **Search Agent**: Mock LLM responses and verify tool selection logic, test tool execution with mocked tool results
* **Summary Agent**: Test summarization with various search result scenarios, verify confidence calculation logic
* **Supervisor Agent**: Test workflow routing decisions, verify DBO formatting logic

### **Integration Testing Requirements**

**What to test for the complete workflow**:

* **End-to-end flow**: Test complete workflow with mocked external services
* **Error handling**: Test various failure scenarios and recovery strategies
* **State management**: Verify state transitions work correctly between agents

### **Tool Testing Requirements**

**What to test for your existing tools**:

* **Configuration loading**: Verify tools load and use website configurations correctly
* **Input transformation**: Test conversion of verification requirements to website input
* **Selenium automation**: Test website interaction with mocked web pages
* **Result extraction**: Verify tools return data in the expected format

## **Deployment Guidelines**

### **Environment Setup**

**What you need to configure**:

* **Development environment**: Local setup with all dependencies and API keys
* **Staging environment**: Azure-hosted testing environment matching production
* **Production environment**: Fully monitored production deployment with proper security

### **Infrastructure Requirements**

**What you need to deploy**:

* **Azure Functions**: For hosting the workflow execution
* **Azure OpenAI**: For LLM services with proper scaling
* **Azure Storage**: For configuration files and logging
* **Azure Key Vault**: For secure credential management
* **Application Insights**: For monitoring and debugging

### **Security Considerations**

**What you need to implement**:

* **API key management**: Store all credentials in Azure Key Vault
* **Network security**: Implement proper VNet and firewall rules
* **Data encryption**: Ensure all data transmission is encrypted
* **Access control**: Implement role-based access for different user types

### **Monitoring and Logging**

**What you need to track**:

* **Workflow execution metrics**: Success rates, execution times, error rates
* **Agent performance**: Tool selection accuracy, confidence score reliability
* **Tool reliability**: Success rates for each of the 5 tools
* **Business metrics**: DBO approval rates, manual review rates

## **Implementation Phases**

### **Phase 1: Core Framework (Week 1-2)**

**What to complete first**:

* Set up project structure and basic configuration management
* Implement the three agents with basic functionality
* Create the LangGraph workflow structure
* Integrate your existing tools with the Search Agent

### **Phase 2: LLM Integration (Week 3-4)**

**What to complete second**:

* Create and refine all LLM prompt templates
* Implement JSON parsing and error handling for LLM responses
* Test and optimize tool selection and stopping criteria
* Fine-tune summarization and recommendation generation

### **Phase 3: Integration & Testing (Week 5-6)**

**What to complete third**:

* Implement external integrations for preprocessing and DBO interface
* Create comprehensive test suite for all components
* Set up monitoring and logging infrastructure
* Performance testing and optimization

### **Phase 4: Deployment & Monitoring (Week 7-8)**

**What to complete last**:

* Deploy to staging environment and conduct integration testing
* Deploy to production with proper monitoring
* Set up alerting and incident response procedures
* Train DBOs on the new interface and workflows

## **Success Criteria**

### **Technical Success Metrics**

**What to measure**:

* **87% processing time reduction**: From 2+ hours to under 15 minutes
* **95%+ accuracy**: In verification recommendations
* **5x DBO efficiency**: Increase in daily processing capacity
* **<2% error rate**: In tool executions and workflow failures

### **Quality Success Metrics**

**What to validate**:

* **Consistent tool selection**: LLM makes appropriate tool choices based on context
* **Accurate confidence scoring**: Confidence levels correlate with actual accuracy
* **Proper manual review flagging**: Items requiring human review are correctly identified
* **DBO approval rates**: High percentage of automated recommendations accepted by DBOs

### **Operational Success Metrics**

**What to monitor**:

* **System availability**: >99.5% uptime during business hours
* **Scalability**: Linear performance with increased VR volume
* **Maintainability**: Easy configuration updates and tool additions
* **Error recovery**: Graceful handling of failures with minimal manual intervention

## **Key Integration Points Summary**

### **Where Your Existing Tools Fit**

1. **In Search Agent initialization**: Import and configure your 5 tool classes
2. **In tool execution method**: Call your tool's main search method
3. **In configuration loading**: Pass website configurations to your tools
4. **In result processing**: Standardize your tool output format

### **What Your Tools Need to Handle**

1. **Input transformation**: Convert verification requirements to website-specific format
2. **Website automation**: Execute Selenium with intelligent dropdown handling
3. **Result extraction**: Clean and structure website results
4. **Error handling**: Graceful failure with proper error reporting

### **External System Interfaces**

1. **Preprocessing input**: Receive structured data from deterministic preprocessing service
2. **DBO output**: Submit formatted results to database operator review interface
3. **Monitoring integration**: Send metrics and logs to monitoring systems
4. **Configuration management**: Load settings from external configuration sources

This guide provides the complete roadmap for implementing the OneKey VR automation system according to your technical design document, with clear integration points for your existing tool files.