# **OneKey VR Automation - Executive Design Summary**

## **Project Overview**

**Objective:** Transform manual OneKey Validation Request (VR) processing from a 2+ hour manual task into a 15-minute automated process using AI agents, achieving 87% time reduction and 95%+ accuracy.

**Current Problem:** Healthcare database operators (DBOs) manually validate healthcare professional information by searching multiple websites, cross-referencing data, and making update decisions - a process that is time-consuming, error-prone, and doesn't scale.`==

## **Core Data Challenges Identified**

Through analysis of actual VR and OneKey Database API responses, we identified three critical challenges:

### **Challenge 1: Unreliable Common IDs**

* VR requests and OneKey Database records don't always have reliable matching IDs
* **Solution:** Implement intelligent record matching using multiple data points (name, workplace, location) with confidence scoring

### **Challenge 2: Multiple People with Similar Names**

* Searching "Marcello Marchetti" returns multiple different doctors with the same name
* **Solution:** Build disambiguation algorithm that scores candidates based on workplace, location, and specialty matches

### **Challenge 3: Multiple Valid Workplace Associations**

* Healthcare professionals legitimately work at multiple hospitals/clinics
* **Solution:** Validate each workplace separately rather than trying to consolidate into single primary workplace

## 

## **Solution Architecture: Hybrid Approach**

**Key Decision:** Combine deterministic preprocessing (predictable, reliable) with agentic AI workflow (intelligent, adaptive) rather than pure AI approach.

### **Why Hybrid Architecture?**

* **Deterministic preprocessing** ensures reliable data handling and easy debugging
* **Agentic AI workflow** provides intelligent search strategy and adaptive behavior
* **Best of both worlds:** Predictability where needed, intelligence where valuable

## **Phase 1: Deterministic Preprocessing (Before Agents)**

### **What It Does:**

1. **Parses VR data** using LLM as a structured tool (not an autonomous agent)
2. **Disambiguates individuals** using rule-based scoring system
3. **Identifies data quality issues** in OneKey Database records
4. **Generates specific verification requirements** for the search agents

### **How It Works:**

* **LLM as Tool:** Use Azure OpenAI with structured prompts to extract workplace info, specialties, locations
* **Scoring Algorithm:** Rate potential matches (40 points for name match, 35 for workplace, 15 for location, 10 for country)
* **Clear Rules:** Score ≥75 = confident match, 50-74 = probable match, <50 = unclear match
* **Output:** Clean, structured data ready for intelligent agent processing

### **Why Deterministic:**

* **Predictable:** Same input always produces same output
* **Debuggable:** Easy to trace decisions and fix issues
* **Reliable:** No unpredictable AI behavior in critical data matching

## 

## **Phase 2: Three-Agent Workflow (After Preprocessing)**

### **Agent 1: Supervisor Agent**

* **Role:** Workflow orchestrator and DBO interface manager
* **Responsibilities:** Receives preprocessed data, coordinates other agents, presents final results to DBO
* **Decision Authority:** High-level workflow control

### **Agent 2: Search Agent**

* **Role:** Intelligent search orchestrator with 5 specialized tools
* **Responsibilities:** Decides which tools to use, execution order, when to stop searching
* **Decision Authority:** All search-related decisions (tool selection, strategy, stopping criteria)

### **Agent 3: Summary Agent**

* **Role:** AI-powered result synthesizer
* **Responsibilities:** Creates human-readable summaries, generates recommendations, calculates confidence scores
* **Decision Authority:** Summary complexity and recommendation generation

## **Search Agent: 5-Tool Architecture**

**Critical Decision:** Use 5 separate specialized tools instead of 1 monolithic search tool.

### **The 5 Tools:**

1. **France Trusted Sources Tool** - French medical directories (Annuaire Santé, Conseil Médecin, etc.)
2. **Italy Trusted Sources Tool** - Italian medical directories (FNOMCEO, FNOPI, etc.)
3. **Hospital Sources Tool** - Specific hospital websites and directories
4. **LinkedIn Professional Tool** - Professional network searches
5. **Untrusted Web Search Tool** - General web search using Tavily/SERP APIs

### **Why 5 Separate Tools (Not 1):**

* **Fault Isolation:** France tool failure doesn't affect Italy tool
* **Independent Scaling:** Scale each tool based on usage patterns
* **Targeted Monitoring:** Track success rates per tool type
* **Team Specialization:** Different teams can own different tools
* **Agent Intelligence:** Search Agent can dynamically select optimal tool combinations

### **Search Agent Intelligence:**

* **Tool Selection:** Agent decides which tools to use based on country, workplace type, confidence requirements
* **Execution Order:** Agent prioritizes trusted sources first, then professional networks, then web search
* **Stopping Criteria:** Agent stops when confidence threshold reached, avoiding unnecessary searches

## **Website Input Handling Challenge & Solution**

### **The Challenge:**

Each trusted source website has different input requirements:

* French sites need: nom, prénom, spécialité, ville
* Italian sites need: nome, cognome, provincia, specializzazione
* Hospital sites need: fullName, department, role

### **Solution: Configuration-Driven Input Mapping**

* **Website Configurations:** Each source has input field mapping and form selectors
* **Input Transformer:** Converts standardized VR data to website-specific input formats
* **Smart Field Derivation:** Milano → MI (province), derive specialty from workplace
* **Unified Selenium Handler:** Single automation system that adapts to different website structures
* **Fallback Strategies:** Try multiple input variations when exact search fails

### **Benefits:**

* **Maintainable:** Add new websites by updating configuration, not writing new code
* **Flexible:** Handles different input requirements automatically
* **Robust:** Graceful handling when required fields are missing

## 

## **Website Dropdown Handling Challenge & Solution**

### **The Challenge: Dynamic Dropdown Selections**

Real-world trusted source websites have dropdown fields with values that don't exactly match the VR data:

* **Geographic Variations:** VR has "Milano" but dropdown shows "MI - Milano", "Milano - Lombardia", or "20100 - Milano"
* **Medical Specialties:** VR has "Cardiology" but dropdown shows "Cardiologia", "Cardiologia Interventistica", or "37 - Cardiologia"
* **Language Differences:** English terms in VR vs local language options in dropdowns
* **Format Variations:** Different websites use different formatting conventions for the same data

### **Solution: Intelligent Dropdown Matching Within Selenium Scripts**

#### **Multi-Level Matching Strategy:**

The Selenium automation includes intelligent dropdown selection logic that tries multiple approaches:

1. **Exact Match:** Direct comparison of VR value with dropdown options
2. **Configured Mappings:** Pre-defined mappings for known variations (Milano → MI, Cardiology → Cardiologia)
3. **Contains Matching:** Check if VR value is contained within dropdown option text
4. **Fuzzy String Matching:** Calculate similarity scores for approximate matches
5. **Semantic Mapping:** Medical terminology and geographic knowledge matching
6. **Fallback Selection:** Choose most reasonable option when exact matches fail

#### **Configuration-Driven Approach:**

Each website's Selenium script uses configuration files that specify:

* **Field-Specific Rules:** Different matching strategies for city vs specialty dropdowns
* **Language Translations:** English to local language mappings for medical terms
* **Common Variations:** Known alternate formats for the same data
* **Fallback Preferences:** Which matching strategy to prioritize for each field type

#### **Implementation Within Selenium Scripts:**

* **Unified Dropdown Handler:** Common dropdown logic shared across all website automation scripts
* **Website-Specific Configurations:** Each trusted source has its own dropdown mapping rules
* **Intelligent Selection Process:** Selenium script automatically applies best matching strategy
* **Error Handling:** Graceful fallbacks when no good match is found

#### **Geographic and Medical Intelligence:**

* **City-to-Province Mapping:** Automatically derive required province codes from city names (Milano → MI)
* **Medical Terminology Translation:** Convert between languages and terminology systems
* **Specialty Hierarchies:** Understand relationships between general and specialized medical fields

### **Benefits for Development:**

#### **Maintainable Selenium Scripts:**

* **Add new websites** by creating configuration files, not rewriting dropdown logic
* **Update mappings** for new dropdown variations without touching Selenium code
* **Centralized intelligence** shared across all website automation scripts

#### **Robust Website Interaction:**

* **Handles website changes** gracefully when dropdown options are modified
* **Adapts to variations** in how the same data is presented across different sites
* **Provides fallback strategies** when exact matches aren't available

#### **Scalable Architecture:**

* **Same dropdown logic** works for France, Italy, and hospital websites
* **Configuration-driven expansion** to new countries and medical systems
* **Consistent behavior** across all trusted source automations

### **Integration with Search Tools:**

Each of the 5 search tools (France Trusted, Italy Trusted, Hospital Sources, LinkedIn, Web Search) uses the same underlying Selenium handler with intelligent dropdown capabilities, but each tool provides its own website-specific configuration for optimal dropdown matching in its domain.

This approach transforms the dropdown selection challenge from a manual coding problem into an intelligent, configurable system that adapts to the real-world complexity of healthcare website forms while maintaining reliable automation performance.

## 

## **Real-World Example: Dr. Marchetti Processing Flow**

### **Input: VR for Marcello Marchetti at Neurological Institute in Milano**

### **Preprocessing Output:**

* Individual matched with 85% confidence score
* Data issues identified: both records marked "inactive"
* Verification requirements: confirm employment, specialty, update status

### **Search Agent Execution:**

* **Tool Selection:** Italy trusted sources (geographic), hospital sources (institution type)
* **Execution:** Italy tool finds active employment, confidence = 0.9
* **Stopping Decision:** High confidence achieved, skip remaining tools
* **Result:** Employment verified in 8 minutes vs 2+ hours manual

### **Summary Agent Output:**

* **Recommendation:** Update both records to active status
* **Confidence:** 91% overall confidence
* **Manual Review:** Not required due to high confidence

## **Implementation Framework (3 Weeks)**

### **Week 1: Deterministic Preprocessing Foundation**

* Build VR/OK DB data parsing with structured LLM usage
* Implement rule-based individual disambiguation with scoring
* Create data quality assessment framework
* Develop verification requirements generation

### **Week 2: Search Agent & 5 Tools**

* Build Search Agent with intelligent tool selection logic
* Implement 5 specialized search tools with proper input handling
* Create website configuration framework and input transformation
* Develop unified Selenium handler for adaptive website interaction

### **Week 3: Integration & Testing**

* Integrate preprocessing with agent workflow
* Build Summary Agent with Azure OpenAI integration
* Create Supervisor Agent for workflow orchestration
* Implement error handling, monitoring, and DBO interface

## **Key Technology Decisions**

### **Azure Services:**

* **Azure OpenAI:** GPT-4 for structured parsing and summarization
* **Azure Functions:** Workflow orchestration and tool management
* **Azure Application Insights:** Comprehensive monitoring and logging

### **External APIs:**

* **LinkedIn Professional API:** Professional network searches
* **Tavily/SERP APIs:** Untrusted web search capabilities
* **Selenium WebDriver:** Automated website interaction

### **Architecture Patterns:**

* **Microservices:** Each tool operates independently
* **Configuration-Driven:** Website handling through configuration files
* **Event-Driven:** Agents communicate through structured messages

## **Business Impact & Success Metrics**

### **Performance Targets:**

* **87% processing time reduction** (2+ hours → 15 minutes)
* **95%+ accuracy** in verification recommendations
* **5x increase** in DBO daily processing capacity
* **70% operational cost savings**

### **Quality Improvements:**

* **Consistent validation** through deterministic preprocessing
* **Intelligent resource usage** through agent-based tool selection
* **Comprehensive workplace validation** respecting healthcare complexity
* **Auditable decisions** for compliance and debugging

### **Risk Mitigation:**

* **Human oversight maintained** for all final decisions
* **Fault isolation** prevents cascading failures
* **Graceful degradation** when tools are unavailable
* **Comprehensive monitoring** for proactive issue resolution

## **Why This Design Will Succeed**

### **Technical Soundness:**

* **Hybrid architecture** combines reliability with intelligence
* **Modular design** enables independent development and scaling
* **Configuration-driven** approach reduces maintenance overhead
* **Comprehensive error handling** ensures production reliability

### **Business Alignment:**

* **Addresses real pain points** with measurable improvements
* **Respects healthcare data complexity** without oversimplification
* **Maintains human control** while maximizing automation benefits
* **Scalable foundation** for future enhancements

### **Implementation Feasibility:**

* **Clear technical framework** for development teams
* **Realistic timeline** with achievable milestones
* **Proven technologies** reducing implementation risk
* **Phased deployment** minimizing operational disruption

## **Next Steps for Implementation**

1. **Stakeholder Approval:** Confirm hybrid architecture approach and resource allocation
2. **Environment Setup:** Provision Azure services and development infrastructure
3. **Team Organization:** Assign teams to preprocessing, agents, and tool development
4. **Development Kickoff:** Begin Week 1 deterministic preprocessing implementation
5. **Continuous Integration:** Establish monitoring and feedback loops for optimization

This design provides a clear roadmap for transforming OneKey VR processing into an intelligent, efficient, and reliable automated system that enhances human expertise while delivering significant business value.