AGENTIC SYSTEM IMPLEMENTATION BLUEPRINT

**Tri-Semantic AI Agent System**

*Version 1.0 | August 31, 2025*

# Table of Contents

1. Project Context  
2. R-Agent Requirements  
3. A-Agent Requirements  
4. B-Agent Requirements  
5. Bridging Agent Requirements  
6. Inter-Agent Communication Protocol  
7. Success Criteria & Metrics  
8. Deployment Strategy  
9. References to Existing Codebase  
10. Key Innovation Points

# 1. PROJECT CONTEXT

|  |  |
| --- | --- |
| PROJECT | Tri-Semantic AI Agent System |
| GOAL | Create 4 autonomous agents that collaborate to provide integrated semantic understanding |
| FOUNDATION | Based on existing conceptual\_space pipeline system with R, A, B pipelines |
| INNOVATION | Implements R4X cross-pipeline semantic integration for PhD research |
| ARCHITECTURE | 4 collaborative agents: R-Agent, A-Agent, B-Agent, Bridging Agent |

# 2. R-AGENT REQUIREMENTS

## R-Agent (BIZBOK Authority)

PRIMARY ROLE: Business ontology expert and concept validation authority

### 2.1 Knowledge Base Setup

* Load and maintain 97 BIZBOK concepts from JSON file  
  Store concept definitions, relationships, and domain mappings  
  Build in-memory graph structure for fast concept traversal  
  Cache frequently accessed concepts for performance

### 2.2 Core Functions

**Function 1: validate\_concept(concept\_string)**• Input: String representing a business concept  
• Process: Exact match → Fuzzy match → Semantic similarity  
• Output: {valid: boolean, confidence: float, canonical\_form: string}

**Function 2: get\_related\_concepts(concept, depth=2)**• Input: Concept name and relationship depth  
• Process: Traverse graph, weight relationships, filter by threshold  
• Output: List of related concepts with relationship types and strengths

**Function 3: validate\_concept\_set(concept\_list)**• Input: List of concepts to validate  
• Process: Batch validation, coverage metrics, identify gaps  
• Output: {coverage: float, validated: list, missing: list, suggestions: list}

### 2.3 Decision Logic Rules

|  |  |
| --- | --- |
| Confidence Range | Action |
| > 0.9 | Exact match, authoritative |
| 0.7 - 0.9 | Strong match, likely valid |
| 0.5 - 0.7 | Weak match, needs verification |
| < 0.5 | Reject, suggest alternatives |

### 2.4 Performance Requirements

* Load time: < 2 seconds for full BIZBOK  
  Query response: < 100ms  
  Concurrent requests: Handle 100+ simultaneous validations

# 3. A-AGENT REQUIREMENTS

## A-Agent (Document Intelligence)

PRIMARY ROLE: Document processing and concept extraction specialist

### 3.1 Document Processing Pipeline

* Accept FinQA financial documents with embedded tables  
  Parse complex financial data structures  
  Handle multiple document formats (text, JSON, parquet)  
  Preserve financial precision (don't fragment numbers)

### 3.2 Core Functions

**Function 1: process\_document(document)**• Input: Raw document text or structured data  
• Process: Clean text, extract sentences, identify tables  
• Output: {cleaned\_text: string, sentences: list, tables: list}

**Function 2: extract\_concepts(processed\_doc)**• Input: Processed document from Function 1  
• Process: TF-IDF → Theme grouping → Core concepts → R-Agent validation  
• Output: {concepts: list, confidence\_scores: dict, themes: list}

**Function 3: build\_concept\_network(concepts)**• Input: List of extracted concepts  
• Process: Co-occurrence analysis, semantic distances, graph creation  
• Output: {nodes: list, edges: list, centrality\_scores: dict}

### 3.3 Specialized Processing Rules

|  |  |
| --- | --- |
| Data Type | Processing Rule |
| Financial numbers | Preserve "$2.2 billion" as single entity |
| Percentages | Keep "18.2%" intact |
| Tables | Extract as structured data, not text |
| Domains | Auto-classify (finance/manufacturing/tech) |

# 4. B-AGENT REQUIREMENTS

## B-Agent (Question Intelligence)

PRIMARY ROLE: Question understanding and answer synthesis

### 4.1 Question Analysis Pipeline

* Parse financial questions (e.g., "What was the change in deferred income?")  
  Extract calculation requirements  
  Identify required data points  
  Determine answer format needed

### 4.2 Core Functions

**Function 1: analyze\_question(question\_text)**• Input: Natural language question  
• Process: Intent detection → Semantic analysis → Requirement analysis  
• Output: {intent: string, concepts: list, data\_requirements: list}

**Function 2: retrieve\_relevant\_knowledge(analysis)**• Input: Question analysis from Function 1  
• Process: Query A-Agent and R-Agent, rank by relevance  
• Output: {documents: list, concepts: list, relevance\_scores: dict}

**Function 3: synthesize\_answer(question, knowledge)**• Input: Question and retrieved knowledge  
• Process: Extract data, perform calculations, format answer  
• Output: {answer: string, confidence: float, evidence: list}

### 4.3 Question Pattern Types

|  |  |
| --- | --- |
| Pattern Type | Example |
| Factual | "What was X in year Y?" |
| Computational | "Calculate the percentage change..." |
| Comparative | "How does X compare to Y?" |
| Definitional | "What is meant by Z?" |

# 5. BRIDGING AGENT REQUIREMENTS

## Bridging Agent (R4X Semantic Integrator)

PRIMARY ROLE: Orchestrate tri-semantic integration across all agents

### 5.1 Orchestration Capabilities

* Coordinate simultaneous requests to all three agents  
  Manage agent dependencies and sequencing  
  Handle partial failures gracefully  
  Optimize request routing

### 5.2 Fusion Strategies

|  |  |
| --- | --- |
| Strategy | Description |
| Consensus | All agents agree (highest confidence) |
| Authority | R-Agent ontology prioritized |
| Evidence | A-Agent document evidence prioritized |
| Context | B-Agent user context prioritized |
| Meta-Strategy | Intelligent selection of above strategies |

# 6. INTER-AGENT COMMUNICATION PROTOCOL

### 6.1 Standard Message Format

{  
 "message\_id": "uuid",  
 "timestamp": "ISO-8601",  
 "sender": "agent\_id",  
 "receiver": "agent\_id",  
 "message\_type": "request|response|broadcast",  
 "priority": "high|medium|low",  
 "payload": {  
 "action": "validate\_concept|extract\_concepts|synthesize\_answer",  
 "data": {},  
 "context": {}  
 },  
 "requires\_response": true,  
 "timeout\_ms": 5000,  
 "trace\_id": "conversation\_uuid"  
}

### 6.2 Agent Endpoints

|  |  |
| --- | --- |
| Agent | Endpoint |
| R-Agent | http://localhost:8001/r-agent/api |
| A-Agent | http://localhost:8002/a-agent/api |
| B-Agent | http://localhost:8003/b-agent/api |
| Bridging Agent | http://localhost:8004/bridge/api |

# 7. SUCCESS CRITERIA & METRICS

### 7.1 Functional Requirements

* All agents can operate independently  
  Agents can collaborate through Bridging Agent  
  System handles FinQA dataset questions  
  Tri-semantic integration produces measurable improvement

### 7.2 Performance Requirements

|  |  |
| --- | --- |
| Metric | Requirement |
| Single question response | < 3 seconds |
| Document processing | < 5 seconds |
| Concept validation | < 100ms |
| Concurrent users | 10+ supported |

### 7.3 Quality Metrics

|  |  |
| --- | --- |
| Metric | Target |
| Answer accuracy | > 80% |
| Concept extraction precision | > 75% |
| Tri-semantic coverage | > 60% of queries |
| User satisfaction | > 4.0/5.0 |

# 8. DEPLOYMENT STRATEGY

### 8.1 Development Phases

|  |  |
| --- | --- |
| Phase | Timeline & Activities |
| Phase 1: Individual Agents |  |
| Week 1-2 | R-Agent (Concept validation) |
| Week 2-3 | A-Agent (Document processing) |
| Week 3-4 | B-Agent (Question handling) |
| Week 4-5 | Bridging Agent (Integration) |
| Phase 2: Integration |  |
| Week 5-6 | Agent communication protocols |
| Week 6-7 | End-to-end testing with FinQA |
| Week 7-8 | Performance optimization |
| Phase 3: Production |  |
| Week 8-9 | Containerization (Docker) |
| Week 9-10 | API documentation & Deployment |

# 9. KEY INNOVATION POINTS

This agentic system represents several PhD-level innovations:

1. Tri-Semantic Integration: First system to unify ontology, document, and question understanding  
   Dynamic Fusion Strategies: Intelligent selection of integration approaches  
   Semantic Bridging: Novel concept connection across knowledge spaces  
   Autonomous Collaboration: Agents that independently coordinate for complex tasks

# 10. REFERENCES TO EXISTING CODEBASE

### 10.1 R-Pipeline Components

* R1\_bizbok\_resource\_loader.py - BIZBOK concept loading  
  R2\_concept\_validator.py - Concept validation logic  
  R3\_reference\_alignment.py - Concept alignment  
  R4L\_lexical\_ontology\_builder.py - Relationship building  
  R4X\_cross\_pipeline\_semantic\_integrator.py - Integration core

### 10.2 A-Pipeline Components

* A1.1\_document\_reader.py - Document ingestion  
  A2.1\_preprocess\_document\_analysis.py - Text preprocessing  
  A2.2\_keyword\_phrase\_extraction.py - Keyword extraction  
  A2.4\_synthesize\_core\_concepts.py - Core concept identification  
  A2.9\_r4x\_semantic\_enhancement.py - R4X enhancement

### 10.3 B-Pipeline Components

* B2.1\_intent\_layer.py - Intent analysis  
  B2.2\_semantic\_layer.py - Semantic analysis  
  B3.3\_hybrid\_retrieval.py - Information retrieval  
  B4.1\_r4x\_answer\_synthesis.py - Answer generation  
  B5.1\_r4x\_question\_understanding.py - Comprehensive understanding

### 10.4 R4X Integration Components

* R4X\_semantic\_fusion\_engine.py - Fusion strategies  
  R5X\_tri\_semantic\_visualizer.py - Visualization  
  R4X\_system\_validation.py - Testing framework

──────────────────────────────────────────────────  
Document Version: 1.0  
Generated: August 31, 2025  
Author: PhD Research System  
Status: Ready for Implementation