Initial Risk Register

Phase 0: Foundation & Planning

Sovereign AI Infrastructure Project

Document:	Initial Risk Register (Phase 0)
Version:	1.0
Date:	February 11, 2026
Status:	Draft - Active
Owner:	Project Manager / Risk Manager
Next Review:	February 25, 2026

Executive Summary: This Initial Risk Register captures 15 identified risks across Technical, Schedule, Resource, Security, Compliance, and External categories. **Critical Risks (2):** R-001 (GPU Memory Exhaustion), R-002 (Model Quality Insufficient). **High Risks (4):** R-003, R-005, R-006, R-011. **Medium Risks (6):** R-004, R-007, R-008, R-009, R-012, R-013. **Low Risks (3):** R-010, R-014, R-015.

Risk Summary by Category

Category	Critical	High	Medium	Low	Total
Technical	2	2	2	0	6
Schedule	0	1	1	1	3
Resource	0	0	1	1	2
Security	0	1	1	0	2
Compliance	0	0	1	1	2
External	0	0	0	0	0
TOTAL	2	4	6	3	15

Detailed Risk Register

Critical Risks (Score 15-25)

R-001: GPU Memory Exhaustion (OOM Crashes)	
Category:	Technical
Description:	Worker models exceed available 16GB VRAM during inference, causing out-of-memory crashes that terminate active tasks and potentially corrupt memory ledger state.
Probability:	4 (Likely - 50-70%)

Impact:	5 (Catastrophic - Project failure if unsolvable)
Score:	**20 (Critical)**
Strategy:	Mitigate
Mitigation Actions:	• Implement real-time VRAM monitoring with pre-emptive OOM detection • Configure aggressive quantization fallback (Q4_K_M → Q3_K_M → Q2_K) • Implement model sharding for large contexts (future consideration) • Validate all model VRAM footprints during Phase 0 PoC • Set conservative context window limits (4K tokens max)
Owner:	Technical Lead
Target Date:	Phase 0 completion (Feb 20, 2026)
Status:	**In Progress**
Contingency:	If OOM persists: Upgrade to Tesla A10 (24GB VRAM) or reduce model count from 5 to 3 specialists

R-002: Model Quality Insufficient for Production	
Category:	**Technical**
Description:	Quantized models (Q4_K_M) produce outputs that fail validation criteria, resulting in unusable system. Hallucination rates exceed acceptable thresholds (>5% for code, >3% for reasoning).
Probability:	3 (Possible - 30-50%)
Impact:	5 (Catastrophic - Outputs not trusted = system failure)
Score:	**15 (Critical)**
Strategy:	Mitigate
Mitigation Actions:	• Conduct Phase 0 model quality validation with representative tasks • Benchmark hallucination rates on 100+ test cases per model • Identify backup models for each specialist role • Implement prompt engineering optimization (Phase 2, 5) • Consider Q5_K_M for critical models if VRAM permits
Owner:	ML Lead
Target Date:	Phase 0 completion (Feb 20, 2026)
Status:	**Open**
Contingency:	If quality unacceptable: Defer to fine-tuning capability (v3.0 roadmap) or select alternative open-weight models

High Risks (Score 10-14)

R-003: Prolog Routing Logic Complexity	
Category:	**Technical**
Description:	Novel Prolog-based constitutional routing introduces learning curve and debugging complexity. Team lacks Prolog expertise; routing errors could cause incorrect model selection.
Probability:	4 (Likely)
Impact:	3 (Moderate)
Score:	**12 (High)**
Strategy:	Mitigate
Mitigation Actions:	• Allocate Prolog training time (2-3 days) for core team • Engage external Prolog consultant for architecture review • Implement comprehensive routing test suite (200+ prompts) • Design fallback routing (safe defaults for ambiguous queries) • Document routing rules extensively with examples
Owner:	Solutions Architect
Target Date:	Phase 1 completion (Mar 15, 2026)
Status:	**Open**
Contingency:	If Prolog proves intractable: Implement routing in Python with rule engine (less declarative but more maintainable)

R-005: Validation Adds Unacceptable Latency	
Category:	**Technical**
Description:	Block-by-block validation for high-stakes tasks adds 30-60 seconds per block. For multi-block outputs, total latency exceeds user tolerance (>3 minutes).
Probability:	4 (Likely)
Impact:	3 (Moderate)
Score:	**12 (High)**
Strategy:	Mitigate
Mitigation Actions:	• Test validation latency early (Phase 2 prototype) • Optimize validator prompts (reduce output verbosity) • Make validation granularity configurable (line vs block vs stage) • Accept slower execution for high-stakes (user expectation management) • Consider parallel validation of independent blocks (future)
Owner:	Technical Lead
Target Date:	Phase 2 completion (Apr 15, 2026)
Status:	**Open**
Contingency:	If latency unacceptable: Defer block-by-block validation to v1.1; use end-stage validation only for v1.0

R-006: Complexity Overwhelms Team	
Category:	**Schedule**
Description:	Novel bicameral architecture, multi-model orchestration, Prolog routing, and validation pipeline create combinatorial complexity. Risk of delays, bugs, or abandonment.

Probability:	3 (Possible)
Impact:	4 (Major - 1-2 month delay)
Score:	**12 (High)**
Strategy:	Mitigate
Mitigation Actions:	• Start small: 2-3 models in Phase 0-1 (not all 5) • Add complexity only when justified (measured value) • Rigorous documentation and knowledge sharing • External consulting if needed (Prolog, llama.cpp experts) • Clear MVP definition; defer nice-to-haves to v1.1+
Owner:	Project Manager
Target Date:	Ongoing
Status:	**In Progress**
Contingency:	If complexity unmanageable: Strip to minimum viable (Router + 1 Worker + Validator only)

R-011: Prompt Injection Attacks	
Category:	**Security**
Description:	Malicious users craft inputs to override system prompts, extract sensitive information, or bypass validation. Could expose system internals or generate harmful outputs.
Probability:	4 (Likely)
Impact:	3 (Moderate)
Score:	**12 (High)**
Strategy:	Mitigate
Mitigation Actions:	• Implement input sanitization (strip suspicious patterns) • Use strong prompt delimiters (SYSTEM/USER boundaries) • Validator checks outputs for prompt injection indicators • Red-teaming test suite (50+ adversarial prompts) • Output filtering for system instruction leakage
Owner:	Security Architect
Target Date:	Phase 2 completion (Apr 15, 2026)
Status:	**Open**
Contingency:	If injection persists: Implement stricter input validation (whitelist approach)

Medium Risks (Score 5-9)

R-004: Model Swap Latency Exceeds Target	
Category:	**Technical**
Description:	Loading models from NVMe to VRAM takes >5 seconds (target: ≤3s). Frequent domain switches create poor user experience.
Probability:	3 (Possible)
Impact:	3 (Moderate)
Score:	**9 (Medium)**
Strategy:	Mitigate
Mitigation Actions:	• Implement warm pool strategy (2-3 models in RAM) • Use PCIe 4.0 NVMe for faster transfers • Optimize llama.cpp model loading (mmap, cache) • Batch similar-domain requests where possible
Owner:	Technical Lead
Target Date:	Phase 2 completion (Apr 15, 2026)
Status:	**Open**

R-007: CPU Validator Inference Too Slow	
Category:	**Technical**
Description:	Granite-H-Small on CPU fails to sustain ≥3 tokens/second, making validation impractical for real-time use.
Probability:	3 (Possible)
Impact:	3 (Moderate)
Score:	**9 (Medium)**
Strategy:	Mitigate
Mitigation Actions:	• Benchmark validator speed during Phase 0 PoC • Optimize llama.cpp CPU settings (threads, batch size) • Consider smaller validator model if speed insufficient • Accept async validation (non-blocking) for v1.0
Owner:	ML Lead
Target Date:	Phase 0 completion (Feb 20, 2026)
Status:	**Open**

R-008: OCR Accuracy Below Threshold	
Category:	**Technical**
Description:	OCR pipeline fails to achieve ≥90% accuracy on scanned documents, causing downstream validation failures and ungrounded claims.
Probability:	3 (Possible)
Impact:	3 (Moderate)
Score:	**9 (Medium)**
Strategy:	Mitigate

Mitigation Actions:	• Evaluate Tesseract vs PaddleOCR during Phase 4 • Implement OCR confidence scoring • Flag low-confidence extractions for manual review • Pre-process images (deskew, denoise) before OCR
Owner:	ML Lead
Target Date:	Phase 4 completion (Jun 15, 2026)
Status:	**Open**

R-009: Real-World Performance ≠ Lab Performance	
Category:	**Schedule**
Description:	System performs well in controlled testing but fails in production scenarios with real user data and edge cases.
Probability:	4 (Likely)
Impact:	2 (Minor)
Score:	**8 (Medium)**
Strategy:	Mitigate
Mitigation Actions:	• Comprehensive monitoring from day one • Gradual production rollout (canary deployment) • Rapid iteration capability (fix issues quickly) • Clear user expectations (v1.0 disclaimer)
Owner:	Product Lead
Target Date:	Phase 7 (Production)
Status:	**Open**

R-012: Prolog Expertise Gap	
Category:	**Resource**
Description:	Team lacks Prolog expertise required for routing logic implementation and maintenance.
Probability:	3 (Possible)
Impact:	3 (Moderate)
Score:	**9 (Medium)**
Strategy:	Mitigate
Mitigation Actions:	• Schedule Prolog training for 2-3 core developers • Engage external Prolog consultant for architecture review • Document routing rules with extensive examples
Owner:	Technical Lead
Target Date:	Phase 1 completion (Mar 15, 2026)
Status:	**Open**

R-013: Audit Trail Incompleteness	
Category:	**Compliance**
Description:	Audit trail fails to capture all required information for compliance (HIPAA, GDPR, SOC 2), creating regulatory risk.

Probability:	3 (Possible)
Impact:	3 (Moderate)
Score:	**9 (Medium)**
Strategy:	Mitigate
Mitigation Actions:	• Define audit trail schema in Phase 1 • Implement comprehensive logging (all decisions, actions, errors) • Review against HIPAA/GDPR/SOC 2 requirements • Test audit export functionality
Owner:	Security Architect
Target Date:	Phase 1 completion (Mar 15, 2026)
Status:	**Open**

Low Risks (Score 1-4)

R-010: llama.cpp Version Compatibility	
Category:	**External**
Description:	Future llama.cpp updates may break compatibility with current GGUF models or quantization formats.
Probability:	2 (Unlikely)
Impact:	2 (Minor)
Score:	**4 (Low)**
Strategy:	Accept
Mitigation Actions:	• Pin llama.cpp version in requirements • Test updates in staging before production • Maintain model vault backups
Owner:	DevOps Lead
Target Date:	Ongoing
Status:	**Open**

R-014: Hardware Procurement Delay	
Category:	**Schedule**
Description:	Delays in procuring Tesla A2 GPU or compatible workstation push back project timeline.
Probability:	2 (Unlikely)
Impact:	2 (Minor - 1-2 week delay)
Score:	**4 (Low)**
Strategy:	Accept
Mitigation Actions:	• Order hardware immediately upon project approval • Identify alternative GPU options (RTX 4090 24GB as fallback)
Owner:	Project Manager
Target Date:	Phase 0 start
Status:	**Open**

R-015: Memory Ledger Git Conflicts	
Category:	**Compliance**
Description:	Concurrent writes to Markdown memory ledger cause Git merge conflicts, potentially corrupting audit trail.
Probability:	2 (Unlikely - v1.0 is single-user)
Impact:	2 (Minor)
Score:	**4 (Low)**
Strategy:	Accept
Mitigation Actions:	• Implement file locking for atomic writes • Use sequential task execution (v1.0 constraint) • Defer multi-user support to v2.0

Owner:	Technical Lead
Target Date:	Phase 2 completion
Status:	**Open**

Change Log

Version	Date	Author	Changes
1.0	2026-02-11	Risk Manager	Initial risk register with 15 identified risks