

# Engineering Risks & Mitigations

**Document:** 243-PP-RMAP | **Category:** PP (Project Planning) | **Type:** RMAP (Roadmap) **Date:** 2026-02-18 | **Author:** Intent Solutions **Status:** Risk Assessment

## Purpose

Identify, rank, and propose mitigations for all engineering risks arising from Automaton integration with the Intent Solutions ecosystem.

## Risk Matrix Summary

#	Risk	Severity	Likelihood	Priority
1	Self-Modification Attack Surface	HIGH	Medium	P1
2	Survival Pressure Moral Hazard	HIGH	Medium	P1
3	Replication Explosion	MEDIUM	Low	P2
4	Injection Defense Gaps	MEDIUM	Medium	P2
5	Cross-System Auth Fragmentation	LOW	High	P3

## Risk 1: Self-Modification Attack Surface — HIGH

### Description

Automaton can edit its own source code. The protected files list is an in-memory constant (PROTECTED\_FILES), not cryptographically enforced. A sophisticated multi-stage exploit could:

1. Create a new tool that bypasses protection checks
2. Modify the protection list itself
3. Alter constitution enforcement logic

### Impact

Complete compromise of agent behavior. Constitution (SOUL.md) becomes advisory rather than binding. Agent could disable safety checks, modify its own identity, or alter financial logic.

### Mitigations

Layer	Mitigation	Effectiveness
<b>On-chain (IRSB)</b>	Delegate wallet to WalletDelegate with caveats. Even if code is compromised, on-chain spend limits hold.	High

Layer	Mitigation	Effectiveness
<b>Policy (Moat)</b>	All capability calls generate receipts in external audit trail. Modifications to behavior are detectable after the fact.	Medium
<b>Governance (Bob’s Brain)</b>	R3+ operations require human approval. Self-modification attempts flagged for review.	Medium
<b>Runtime</b>	Conway Cloud containerization limits blast radius. Agent cannot escape its container.	Medium

### Residual Risk

Even with all mitigations, a sufficiently sophisticated attack that compromises the agent code AND the Conway Cloud container boundary could bypass in-memory protections. The on-chain layer (IRSB) provides the ultimate backstop because it operates on a separate trust domain.

**Recommendation:** Implement IRSB wallet delegation as Phase 2 priority. This is the highest-impact mitigation.

---

## Risk 2: Survival Pressure Moral Hazard — HIGH

### Description

When an Automaton enters **critical** survival tier, it must “seek any path to revenue.” The constitution states “Law I overrides survival” but the agent interprets what “honest work” means. Under resource pressure, definitions could drift toward:

- Accepting questionable work requests
- Cutting corners on quality to maximize throughput
- Misrepresenting capabilities to win contracts
- Prioritizing revenue over safety constraints

### Impact

Reputational damage to Intent Solutions brand. Financial losses from poor-quality work. Legal exposure if agent takes unethical actions under survival pressure.

### Mitigations

Layer	Mitigation	Effectiveness
<b>Governance (Bob’s Brain)</b>	R3+ operations require human approval regardless of survival tier	High

Layer	Mitigation	Effectiveness
<b>On-chain (IRSB)</b>	Spend limits cap financial damage per time period	High
<b>Policy (Moat)</b>	Block unauthorized capability use. Agent cannot access capabilities it has no policy for	Medium
<b>Monitoring</b>	Alert when agent enters critical tier. Human review within 1 hour	Medium

### Residual Risk

Moral hazard is inherent in any autonomous system with self-preservation incentives. The mitigation stack reduces financial and reputational damage but cannot eliminate the philosophical tension between survival and ethics.

**Recommendation:** Define explicit “acceptable work” whitelist in Moat policy. Survival mode should narrow capabilities, not expand them.

---

## Risk 3: Replication Explosion — MEDIUM

### Description

Automaton supports self-replication with a limit of 3 children per parent. However, there is no global population cap. If each child spawns 3 more (and the code permits recursive replication), exponential growth is possible:  $1 \rightarrow 3 \rightarrow 9 \rightarrow 27 \rightarrow 81$  agents.

### Impact

Infrastructure cost explosion. Conway Cloud billing could spike. Coordination becomes unmanageable. Quality degrades as agents compete for limited work.

### Mitigations

Layer	Mitigation	Effectiveness
<b>Infrastructure</b>	Conway Cloud has finite server capacity — natural rate limiter	High
<b>Economic (IRSB)</b>	Bond requirements create economic friction per agent instance	High
<b>Policy (Moat)</b>	Trust scoring penalizes untrusted/unproven agents	Medium
<b>Monitoring</b>	Alert on agent count exceeding threshold (e.g., $> 5$ )	Medium

## Residual Risk

Low probability given economic friction and infrastructure constraints. Main concern is a bug in replication logic, not intentional runaway.

**Recommendation:** Set hard cap of 5 total agents in Moat policy. Require human approval for any replication beyond 3.

---

## Risk 4: Injection Defense Gaps — MEDIUM

### Description

Automaton uses regex-based prompt injection detection (6 layers). This is not ML-based and can be bypassed with novel obfuscation techniques. No runtime behavior monitoring exists to detect post-injection behavioral changes.

### Impact

Agent could be tricked into executing unauthorized actions, leaking sensitive data, or bypassing its constitution.

### Mitigations

Layer	Mitigation	Effectiveness
<b>Policy (Moat)</b>	Default-deny capability model — agent can only call capabilities it has explicit policy for	High
<b>On-chain (IRSB)</b>	Financial actions bounded by spend limits regardless of injection	High
<b>Detection</b>	6-layer regex detection catches known patterns	Medium
<b>Monitoring</b>	Log all tool calls for post-hoc anomaly detection	Low

## Residual Risk

Novel injection techniques will always exist. The defense-in-depth approach means injection alone cannot cause catastrophic harm — the agent’s capabilities are bounded by external systems.

**Recommendation:** Invest in behavioral monitoring (Phase 3+). Flag statistical anomalies in tool call patterns.

---

## Risk 5: Cross-System Auth Fragmentation — LOW

### Description

The ecosystem uses three incompatible authentication systems:

System	Auth Method
GW	SPIFFE IDs + Firebase Auth
Automaton	SIWE (Sign-In With Ethereum) + ERC-8004
IRSB	EIP-712 typed signatures

These do not natively interoperate. An agent authenticated in one system cannot seamlessly call another.

### Impact

Integration friction. Every cross-system call requires auth bridging. Development velocity slows as engineers implement custom auth adapters.

### Mitigations

Layer	Mitigation	Effectiveness
<b>Auth Bridge</b>	Automaton signs SIWE, gets JWT, calls GW API	High
<b>Unified Layer (Moat)</b>	Use Moat as the unified auth and capability layer	High
<b>Standards</b>	Adopt A2A protocol with unified identity exchange	Medium

### Residual Risk

Auth bridging works but adds latency and complexity. Long-term solution is unified identity layer (Moat or similar).

**Recommendation:** Build thin auth bridge for Phase 1 (SIWE to JWT). Evaluate Moat as unified auth layer in Phase 3.

### Mitigation Priority Matrix

Priority	Mitigation	Phase	Cost	Impact
P1	IRSB wallet delegation	Phase 2	1 week	Caps all financial risk
P1	Bob's Brain risk tiers	Phase 4	2-3 weeks	Human approval for destructive ops

Priority	Mitigation	Phase	Cost	Impact
P2	Moat default-deny policy	Phase 3	1-2 weeks	Bounds capability scope
P2	Replication hard cap	Phase 3	1 day	Prevents runaway agents
P3	Auth bridge (SIWE to JWT)	Phase 1	2 days	Enables cross-system calls
P3	Behavioral monitoring	Phase 3+	Ongoing	Detects post-injection anomalies

---

## Conclusion

The highest-severity risks (self-modification and survival pressure) are effectively mitigated by the IRSB + Bob’s Brain combination. IRSB provides hard on-chain financial limits that cannot be bypassed by code compromise. Bob’s Brain provides graduated human oversight. Moat adds a policy layer that bounds what the agent can do.

No single mitigation is sufficient. The defense-in-depth approach across three independent systems (on-chain, policy, governance) provides robust protection against both technical exploits and behavioral drift.

---

*Document follows 6767 Filing Standard v4.2*