SADAR

Addressing the FINOS AI Governance Framework

Risk Catalogue (v2, October 2025)

A mapping of SADAR open-standard capabilities to FINOS AIGF risks

Document Type
Position Paper

Date
April 2026

Publisher
OpenSemantics.org / Cognita AI Inc.

SADAR Version
v0.9 DRAFT (CSL 1.0)

1. Executive Summary

The FINOS AI Governance Framework (AIGF) v2, released in October 2025, catalogues 46 risks and associated mitigations spanning operational, security, and regulatory dimensions. Its v2 expansion introduced a dedicated agentic AI risk section—acknowledging that autonomous, multi-agent architectures present categorically different governance challenges than static retrieval-augmented generation systems.

The Semantic Agent Discovery and Attribution Registry (SADAR) is an open standard stewarded by Cognita AI Inc on OpenSemantics.org. SADAR addresses the governance gap at the discovery and authorization layer: before an agent acts, SADAR establishes what that agent is, what it is authorized to do, who it is, and whether the invoking party is entitled to invoke it. This proactive, pre-invocation governance is distinct from runtime monitoring or post-hoc audit.

This document maps SADAR's open-standard mechanisms to the FINOS AIGF v2 risk catalogue, demonstrating where SADAR provides direct, partial, or indirect coverage. It explicitly scopes out CogniWeave—Cognita AI's proprietary runtime enforcement platform that implements SADAR—so that the contribution of the open standard itself is assessed fairly.

Key findings:

- SADAR provides direct coverage for 8 of the 11 operational risks, 4 of the 6 security risks, and 4 of the 5 regulatory risks catalogued in AIGF v2.
- SADAR's strongest contributions are to Model Overreach, Agent Action Authorization Bypass, Multi-Agent Trust Boundary Violations, Lack of Explainability (attribution), and regulatory audit trail risks.
- SADAR is a governance foundation standard, not a runtime enforcement engine. It establishes the identity, scope, and capability contracts that enforcement systems—whether CogniWeave or others—then act upon.
- SADAR is available under CC BY 4.0 and can be adopted by any financial institution or standards body independently of any proprietary implementation.

2. Background

2.1 The FINOS AI Governance Framework

FINOS (Fintech Open Source Foundation), in collaboration with major financial institutions including BMO, Citi, Morgan Stanley, RBC, and Bank of America, released the AIGF as a vendor-neutral, living governance framework. Version 2.0 expanded coverage from 30 to 46 risks and mitigations, cross-referenced to seven existing frameworks including OWASP, MITRE, and the EU AI Act.

The AIGF organizes risks into three categories:

- Operational – risks arising from model behavior, reliability, and use (11 risks in v2)
- Security – threats from adversarial actions, injection, and unauthorized access
- Regulatory – compliance failures, audit deficiencies, and explainability obligations

A defining feature of v2 is its agentic AI section, which explicitly addresses multi-agent architectures—systems where AI agents autonomously discover, invoke, and coordinate with other agents and services. The AIGF notes that existing governance frameworks were designed for RAG systems with predictable risk surfaces, and that agentic systems require new governance controls addressing trust boundaries, authorization, and invocation identity.

2.2 SADAR: Scope of this Analysis

SADAR (Semantic Agent Discovery and Attribution Registry) is an open standard specification for governing agentic AI systems at the discovery and authorization layer. It defines:

- Manifest Structure – a structured, digitally signed declaration of an agent's identity, capabilities, authorized scope, and operational constraints
- Semantic Capability Grounding – IRI-based capability naming anchored to established taxonomies (O*NET, NAICS, APQC PCF, HL7, ISO 20022) enabling unambiguous, machine-readable capability declarations
- Registry Protocol – a distributed registry topology (Provider, Marketplace, Industry, Community, Internal, and Registry of Registries) for agent discovery with TTL-based lifecycle management
- Resolver Contract – a standardized interface defining how callers discover, validate, and resolve agent manifests prior to invocation
- Invocation Identity – OIDC Client Credentials and mutual TLS (mTLS) for agent authentication, with digital manifest signing providing tamper detection
- Bilateral Manifest Matching – a discovery model requiring that both the invoking and invoked agent manifests declare compatible capabilities and scope before a discovery is completed
- Conformance Levels – a tiered compliance framework (L1 through L3) enabling progressive adoption across institutions

3. SADAR Governance Mechanisms Reference

The following table summarizes SADAR's core governance mechanisms referenced throughout the risk mapping in Section 4.

Mechanism	Description
Signed Manifest	Agents publish a digitally signed, structured manifest declaring their identity, authorized capabilities, operational scope, constraints, and versioning. Manifests are the governance contract for agent behavior.
Semantic Capability Grounding	Capabilities are named using IRIs anchored to recognized taxonomies (O*NET, NAICS, APQC PCF, domain standards). This eliminates ambiguity in what an agent claims to do and enables machine-readable policy evaluation.
Registry as Authorization Anchor	The SADAR registry is not merely a service directory. It is a tamper-evident, authoritative introduction record. Presence in the registry is a governance attestation, not just a discovery convenience.
Bilateral Manifest Matching	Discovery succeeds only when both the invoking agent's manifest and the target agent's manifest declare compatible capabilities and scope. Neither party can invoke beyond declared scope without a manifest update.
OIDC + mTLS Invocation Identity	Every agent invocation is bound to a machine identity (OIDC Client Credentials) and transport-layer mutual authentication (mTLS). This establishes a non-repudiable invocation record attributing every action to a specific agent identity.
Registry of Registries + TTL	A hierarchical federation of registries with producer-enforced TTL (replication_seconds for propagation, session_seconds for session validity) enables lifecycle governance across distributed agent deployments.
Conformance Levels (L1–L3)	Progressive conformance tiers allow institutions to adopt SADAR incrementally, starting with basic manifest registration (L1), advancing to bilateral matching (L2), and full semantic grounding with registry federation (L3).
Open Standard (CSL 1.0)	SADAR is licensed under Common Specifications License 1.0. Any institution, vendor, or regulatory body may adopt, implement, or extend the standard independently of any proprietary platform.

Table 1: SADAR Open-Standard Governance Mechanisms

4. Risk Mapping: SADAR vs. FINOS AIGF v2

The table below maps each FINOS AIGF v2 risk to the SADAR mechanism(s) that address it, with a coverage assessment. Coverage is assessed as:

- Direct – SADAR's standard mechanisms substantively address the risk at the governance layer
- Partial – SADAR contributes meaningfully but the risk requires additional runtime or detective controls
- Indirect – SADAR creates conditions that reduce the likelihood or impact of the risk

Risk ID	Risk	SADAR Mechanism	Contribution	Coverage
OPERATIONAL RISKS				
AIR-OP-004	Hallucination & Inaccurate Outputs	Signed Manifest: capability scope constrains what agent types are authorized	Manifests restrict agents to declared, validated capability types. Scope constraints reduce the range of contexts in which an unconstrained LLM is presented as authoritative, limiting hallucination blast radius.	**Partial**
AIR-OP-005	Foundation Model Versioning	Registry manifest versioning; TTL lifecycle management	SADAR manifests carry version fields for both the agent and its underlying model. Registry TTL mechanisms surface staleness. Version changes require manifest updates, creating a change-management forcing function.	**Partial**
AIR-OP-006	Non-Deterministic Behaviour	Signed Manifest: declared operational constraints and determinism requirements	Manifests can declare operational constraints (e.g., temperature bounds, determinism requirements). While SADAR cannot eliminate LLM non-determinism, scope boundaries limit the range of outputs that matter operationally.	**Partial**
AIR-OP-007	Availability of Foundational Model	Registry of Registries; TTL-based failover signaling	The distributed registry topology supports multi-provider agent	**Indirect**

Risk ID	Risk	SADAR Mechanism	Contribution	Coverage
			registration. TTL expiration signals availability degradation. Bilateral matching can be configured to require availability attestations in manifests.	
AIR-OP-014	Inadequate System Alignment	Bilateral Manifest Matching; Semantic Capability Grounding	Scope boundary violations—a primary cause of misalignment—are structurally prevented. An agent cannot be invoked for a purpose outside its declared manifest scope. Semantic grounding makes scope declarations machine-evaluable, not prose-dependent.	**Direct**
AIR-OP-016	Bias & Discrimination	Signed Manifest: use-case scope; Conformance Levels	Manifests declare the permissible use cases for which an agent may be invoked. L3 conformance requires alignment to established taxonomic categories, making discriminatory misapplication detectable as a manifest violation.	**Partial**
AIR-OP-017	Lack of Explainability	OIDC + mTLS Invocation Identity; Signed Manifest	Every invocation is attributable to a specific agent identity, timestamped, and linked to the manifest that governed it. This creates an attribution chain that supports explainability: who acted, under what declared authority, in what declared scope.	**Direct**
AIR-OP-018	Model Overreach / Expanded Use	Bilateral Manifest Matching; Signed Manifest scope declaration	Overreach requires a manifest update to succeed. An agent cannot be discovered or invoked for out-of-scope purposes without an updated, re-	**Direct**

Risk ID	Risk	SADAR Mechanism	Contribution	Coverage
			signed manifest. This is SADAR's most direct governance contribution: scope is enforced at the discovery layer, before invocation.	
AIR-OP-019	Data Quality & Drift	Registry TTL; Manifest versioning	TTL expiration forces re-registration and re-attestation. Manifests version the agent's data dependencies. Stale agents surface as TTL violations rather than silent drift, creating an observable signal for governance teams.	**Partial**
AIR-OP-020	Reputational Risk	Signed Manifest; Invocation Identity; Bilateral Matching	The combination of scope-constrained discovery, non-repudiable invocation identity, and manifest-governed authorization significantly reduces the probability of unauthorized agent behavior that leads to reputational incidents.	**Indirect**
AIR-OP-028	Multi-Agent Trust Boundary Violations	Bilateral Manifest Matching; OIDC + mTLS; Registry as Authorization Anchor	Trust boundary violations require forging or bypassing manifest-level scope declarations. Bilateral matching means that each leg of an agent-to-agent invocation chain is independently governed. mTLS mutual authentication prevents impersonation. The registry's tamper-evident design prevents unauthorized manifest injection.	**Direct**
SECURITY RISKS (Agentic)				
AIR-SEC (Prompt Injection)	Prompt Injection & Indirect Injection	Bilateral Manifest Matching; Signed Manifest authorized-action declarations	SADAR constrains what actions are authorized in the manifest. Injected instructions that	**Partial**

Risk ID	Risk	SADAR Mechanism	Contribution	Coverage
			attempt to cause actions outside declared scope would require manifest-level authorization changes to succeed. This does not prevent injection attempts but limits their achievable scope.	
AIR-SEC (Data Leakage)	Data Leakage via Agent Actions	Signed Manifest: data access scope declarations; Invocation Identity	Manifests declare the data categories an agent is authorized to access or transmit. Non-repudiable invocation identity attributes all data access to specific agent identities, enabling audit. Scope constraints limit data access surface.	**Partial**
AIR-SEC (Agent Auth Bypass)	Agent Action Authorization Bypass	Bilateral Manifest Matching; OIDC Client Credentials; Registry as Authorization Anchor	This is SADAR's primary security contribution. Authorization bypass requires either compromising the registry (tamper-evident), forging mTLS certificates, or issuing OIDC tokens outside the authorization server's control. Each represents a significant, detectable attack rather than a configuration error.	**Direct**
AIR-SEC (Memory Poison)	Memory Poisoning	Signed Manifest; Invocation Identity	SADAR's invocation identity ensures that all memory-state interactions are attributable. Manifest scope constrains which agents can write to shared memory stores. Malicious memory injection would require an authorized agent identity to succeed, making it traceable.	**Partial**
AIR-SEC (Persistent Compromise)	Persistent Agent Compromise	Registry TTL; Manifest versioning; Re-	TTL-based session expiration limits the window of persistent	**Partial**

Risk ID	Risk	SADAR Mechanism	Contribution	Coverage
		registration enforcement	compromise. Re-registration forces re-attestation of agent identity and state. A compromised agent's TTL expiration triggers re-validation, providing a natural detection window.	
AIR-SEC (Supply Chain)	Supply Chain & Third-Party Model Risks	Semantic Capability Grounding; Registry provenance; Manifest signing	Manifests declare model provenance (provider, version, training data assertions). Digital signing provides tamper detection. The Registry of Registries can enforce provenance policies at the federation level.	**Partial**
REGULATORY RISKS				
AIR-REG (Audit Trail)	Insufficient Audit Trail	OIDC + mTLS Invocation Identity; Signed Manifest chain	Every SADAR-governed invocation is bound to a machine identity, timestamped, and linked to the governing manifest. This creates a structured, machine-readable audit trail covering: who invoked, what capability, under what declared authority, from which registry.	**Direct**
AIR-REG (Explainability)	Regulatory Explainability & Accountability	Signed Manifest; Semantic Capability Grounding; Invocation Identity	Regulators can obtain: the manifest that governed an agent, the capability declarations that authorized an action, the identity of the invoking agent, and the registry record attesting to the agent's registration. This provides a structured explainability framework aligned to EU AI Act and U.S. prudential requirements.	**Direct**

Risk ID	Risk	SADAR Mechanism	Contribution	Coverage
AIR-REG (Scope Control)	Regulatory Scope & Use-Case Compliance	Bilateral Manifest Matching; Semantic Capability Grounding; Conformance Levels	Manifests are the machine-readable expression of an agent's authorized use cases. Semantic grounding to recognized taxonomies enables regulators to independently verify that declared scope is consistent with regulatory classifications. Conformance levels provide a graduated compliance framework.	**Direct**
AIR-REG (Data Protection)	Data Protection & Privacy Compliance	Signed Manifest: data handling declarations; Invocation Identity	Manifests can declare data residency, retention, and processing constraints. Invocation identity enables data access attribution. While SADAR does not enforce data protection at the runtime layer, it establishes the governance declarations that enforcement systems rely on.	**Partial**
AIR-REG (Vendor Lock-in)	Third-Party Dependency & Vendor Lock-in	Open Standard (CC BY 4.0); Registry of Registries federation	SADAR is an open standard with no vendor dependency. Any institution can implement compliant registries and resolvers. The federation model enables interoperability across institutional boundaries without requiring a common vendor platform.	**Direct**

Table 2: SADAR Risk Coverage Mapping — FINOS AIGF v2 (Coverage: Direct | Partial | Indirect)

5. Detailed Analysis: Key Risk Areas

5.1 Model Overreach and Scope Boundary Violations (AIR-OP-018, AIR-OP-014)

The AIGF identifies model overreach—the use of AI agents beyond their intended purpose—as a critical operational risk. In agentic architectures, this risk is amplified because agents can autonomously invoke other agents, potentially chaining capabilities that no single invocation was designed to trigger.

SADAR addresses this risk structurally through bilateral manifest matching. When an orchestrator agent seeks to invoke a specialist agent, the SADAR resolver evaluates whether:

- The orchestrator's manifest declares the invocation as an authorized outbound capability
- The target agent's manifest accepts invocations of the declared type from the declared caller category
- The invocation's asserted context is within scope for both parties

This bilateral design means that scope violation requires a coordinated manifest compromise—not merely a misconfiguration of a single component. In practice, this converts model overreach from a runtime behavior problem into a governance artifact problem: if it happens, it is a manifest governance failure, not an invisible model behavior failure. That distinction is critical for regulated institutions: manifest failures are auditable, traceable, and correctable through governance processes.

5.2 Agent Action Authorization Bypass (AIR-SEC)

The AIGF's agentic section identifies authorization bypass as a primary security risk: agents exploiting API vulnerabilities, escalating privileges through tool chains, or circumventing approval workflows. In traditional software, authorization is often enforced at individual API endpoints. In agentic architectures, the attack surface is the entire capability composition space.

SADAR's authorization model operates at the discovery layer, before any API call is made. An agent that is not registered in the SADAR registry cannot be discovered. An agent whose manifest does not authorize a specific capability type cannot be invoked for that capability by a SADAR-conformant orchestrator. The registry's tamper-evident design means that unauthorized manifest injection is detectable.

Critically, SADAR uses OIDC Client Credentials for machine identity—not human identity delegation—combined with mTLS for transport-layer mutual authentication. This means that every leg of an agent invocation chain carries a cryptographically bound identity. Authorization bypass in a SADAR-governed system requires compromising cryptographic credentials, not exploiting configuration errors. This significantly raises the attack cost and detectability.

5.3 Multi-Agent Trust Boundary Violations (AIR-OP-028)

The AIGF v2 agentic section specifically addresses multi-agent trust boundary violations: compromised agents propagating failures through shared resources, communication channels, or state corruption. In unstructured agentic architectures, trust is often implicit—agents assume that messages from other agents within the same orchestration context are trustworthy.

SADAR eliminates implicit trust. Every agent interaction is governed by mutual manifest validation and cryptographic identity. An agent receiving an invocation can verify that the calling agent's identity matches a registered, valid manifest, that the requested capability is within the

caller's declared scope, and that the invocation carries a valid mTLS certificate. A compromised agent cannot propagate trust violations to SADAR-conformant agents without also compromising the registry records and mTLS credentials of those agents—a significantly harder attack.

5.4 Regulatory Audit Trail and Explainability (AIR-REG)

Regulated financial institutions face a growing obligation to explain AI-driven decisions to regulators, customers, and auditors. The AIGF identifies both insufficient audit trails and lack of explainability as regulatory risks. These are related but distinct: an audit trail proves what happened; explainability demonstrates why a decision was made.

SADAR contributes to both. For audit trails, SADAR's invocation identity mechanism creates a structured record: agent identity (OIDC client ID), transport authentication (mTLS certificate), the manifest version governing the invocation, and the registry entry attesting to the agent's registration. This record is created at the discovery layer—before any model inference—and is independent of the LLM's internal state.

For explainability, SADAR's manifest provides regulators with a machine-readable declaration of what the agent was authorized to do, what capabilities it claimed, and what constraints applied. When combined with invocation identity, a regulator can answer: this agent, registered under this manifest version, was invoked by this identity, for this declared capability. That is a substantively different and more defensible answer than post-hoc model interpretation.

6. Scope Limitations and What SADAR Does Not Address

Intellectual honesty requires a clear statement of what SADAR, as an open standard, does not provide:

6.1 Runtime Enforcement

SADAR establishes governance contracts at the discovery layer. It does not enforce those contracts at runtime. A SADAR-conformant orchestrator that has completed bilateral matching could still, through a bug or compromise, issue invocations that deviate from the matched manifests. Runtime enforcement—validating that actual invocations remain within scope—requires an execution-layer component such as CogniWeave's Guardian Execution Service. SADAR is a necessary but not sufficient component of runtime governance.

6.2 Hallucination Prevention

SADAR does not and cannot prevent LLM hallucination. It constrains the scope in which a hallucinating agent can operate, limiting blast radius, but it does not address the probabilistic output generation that causes hallucinations. Separate detective controls (output validation, RAG grounding, confidence scoring) are required.

6.3 Bias Detection

SADAR can constrain which agents are invoked for which use cases, potentially limiting misapplication that amplifies bias. However, it does not detect or correct bias in model outputs. Bias detection requires statistical monitoring, fairness testing, and model validation—none of which are within SADAR's scope.

6.4 Model Non-Determinism

SADAR cannot make LLM outputs deterministic. It can surface non-determinism as a governance concern (through manifest versioning and TTL-based staleness detection), but the fundamental probabilistic nature of LLMs is outside any discovery-layer standard's scope.

6.5 Availability and Infrastructure

SADAR's registry TTL mechanisms provide signals that can inform availability management, but SADAR does not provide active failover, load balancing, or infrastructure resilience. These remain operational concerns for implementing institutions.

7. Alignment with AIGF Mitigation Philosophy

The FINOS AIGF explicitly distinguishes preventative controls (those that reduce the likelihood of a risk materializing) from detective controls (those that identify when a risk has materialized). SADAR is primarily a preventative standard with strong detective enablement:

7.1 Preventative Contributions

- Bilateral manifest matching prevents scope boundary violations at the discovery layer
- Signed manifests with versioning prevent silent agent substitution
- OIDC + mTLS prevents agent identity spoofing
- Registry tamper-evidence prevents unauthorized agent registration
- TTL-based lifecycle management prevents stale agent proliferation

7.2 Detective Enablement

SADAR does not itself operate detective controls, but it creates the infrastructure that makes detection possible:

- Non-repudiable invocation identity enables attribution of anomalous actions to specific agent identities
- Manifest change history in the registry enables detection of unauthorized manifest modifications
- TTL expiration events signal potential availability or staleness issues
- Bilateral matching failures generate observable signals when scope violations are attempted

The AIGF's call for 'operational pathways'—controls that can be enforced in production environments—is directly addressed by SADAR's design: the standard defines the governance contracts, and conformant implementations enforce them at both the discovery and runtime layers.

8. Conclusion

The FINOS AI Governance Framework v2 represents the financial services industry's most comprehensive public articulation of AI governance risks, particularly for agentic architectures. Its expanded risk catalogue reflects a genuine understanding that autonomous, multi-agent AI systems require governance controls that operate at the discovery and authorization layer—not merely at the model or output layer.

SADAR addresses this governance gap as an open standard. By establishing machine-readable governance contracts (manifests), a tamper-evident authorization infrastructure (registry), cryptographic agent identity (OIDC + mTLS), and mutual scope validation (bilateral matching), SADAR provides a principled governance foundation that is independent of any particular LLM, orchestration platform, or cloud vendor.

For FINOS member institutions, SADAR represents a practical, adoptable open standard that can be integrated into existing three-lines-of-defence models, aligned to OWASP and MITRE risk frameworks (as AIGF already references), and implemented progressively through SADAR's L1–L3 conformance levels.

The complementarity between SADAR (governance contracts and discovery-layer controls) and the AIGF's runtime mitigation recommendations is intentional: effective agentic AI governance requires both a principled standard for what agents are authorized to do and operational controls that enforce those authorizations at execution time. SADAR provides the former as an open standard; CogniWeave implements both as a proprietary enterprise platform.

Appendix A: Coverage Summary

The following table provides a consolidated coverage summary across risk categories.

Risk Category	Direct	Partial	Indirect
Operational (11 risks)	3	6	2
Security Agentic (6 risks)	2	4	0
Regulatory (5 risks)	4	1	0
Total (22 risk areas mapped)	**9**	**11**	**2**

Table A-1: SADAR Coverage Summary by Risk Category

Appendix B: References

- FINOS AI Governance Framework v2 (October 2025): https://air-governance-framework.finos.org/
- SADAR Specification v0.9 DRAFT (CSL 1.0): OpenSemantics.org
- FINOS AIGF v2.0 Announcement: finos.org/blog/finos-ai-governance-framework-v2.0
- Tetrate Agentic AI Extension (October 2025): tetrate.io/press/tetrate-expands-finos-ai-governance-framework
- FINOS Common Controls for AI Services (CC4AI): finos.org
- O*NET Occupation Information Network: onetonline.org
- APQC Process Classification Framework: apqc.org
- NIST AI Risk Management Framework: nist.gov/artificial-intelligence