GALXEE AI CfAM

Project Description

Containment-First Agentic Middleware for Secure Autonomous AI Agents

Prepared for: Government Grant Review
Funding Request: $500,000
Applicant: GALXEE AI CfAM
Location: Denver, Colorado
Website: www.wearegalxee.ai

Project Focus: AI Security, Task Containment, Trustworthy Agentic Systems, and Critical Infrastructure Readiness

1. Executive Summary

GALXEE AI CfAM proposes to develop and validate a Containment-First Agentic Middleware (CFAM): a model-agnostic security layer that sits between large language model reasoning systems and the real-world tools, APIs, databases, communications channels, and business workflows those systems increasingly control. The project is designed to solve the core safety gap in autonomous AI agents: current systems rely heavily on prompt-level instructions and policy reminders, while the actual ability to act remains under-enforced at the execution layer.

The proposed project converts GALXEE AI CfAM from an AI voice and automation company into a differentiated AI security infrastructure company. Rather than competing as "another AI receptionist," the company will build the security foundation required for safe AI receptionists, customer service agents, enterprise assistants, government workflow agents, and defense-relevant multi-agent systems. The project directly addresses the technical and commercial need for autonomous agents that can act, but cannot exceed their authority.

CFAM will enforce agent behavior through three integrated subsystems: a Policy Engine that defines cryptographically signed capability manifests and formal task boundaries; an Isolation Layer that executes agent-requested tool calls in stateless ephemeral sandboxes; and a Sentry

Node that validates every proposed action before execution. Together, these components shift agent safety from a "please behave" model to an enforceable containment model.

The $500,000 project will fund applied R&D, prototype hardening, red-team validation, benchmark development, design partner pilots, and grant/commercial transition preparation. Success will be measured by reduced exploitability across 13 high-priority agentic AI failure modes, low-latency enforcement overhead suitable for real-time voice interactions, compatibility across multiple LLM backends, and production-ready documentation for Phase II federal funding and enterprise deployment.

2. Technical Challenge and Need

Autonomous AI agents are moving from passive chat interfaces into action-oriented workflows: scheduling meetings, reading emails, accessing CRMs, searching databases, purchasing services, generating advice, routing customer issues, and executing API calls. These capabilities create economic value, but they also create a new class of security problem. The agent is no longer merely producing text; it is making decisions and triggering real-world actions.

Most current safety approaches are concentrated at the instruction layer. Developers provide system prompts, safety policies, refusal instructions, retrieval filters, and human-readable guardrails. These techniques are useful but insufficient because they do not create a hard enforcement boundary between the model's proposed action and the system's actual execution environment. In a high-consequence setting, a natural-language instruction cannot be treated as a security control.

The project addresses four linked technical gaps:

- **Execution-layer gap:** Agents often retain tool access even when the task no longer requires it, creating excessive agency and privilege escalation risk.
- **Adversarial input gap:** Prompt injection, voice manipulation, malicious documents, and compromised API responses can redirect agent behavior without changing the underlying model.
- **Isolation gap:** If an agent can trigger code execution, call arbitrary tools, or access persistent memory, a single compromise can become a lateral movement pathway.
- **Measurement gap:** Organizations lack standardized, repeatable ways to test whether an agent is secure against real-world failure modes before deployment.

The practical consequence is that enterprise and government customers hesitate to deploy agentic systems in sensitive settings. They may want the productivity gain, but they cannot accept uncontrolled risk around confidential data, identity spoofing, unauthorized purchases, harmful advice, supply chain exposure, or agent-to-agent compromise. CFAM is designed to make autonomous agents deployable by making their authority formally bounded, observable, and enforceable.

3. The 13 Agentic AI Failure Modes Addressed

The proposed R&D plan is organized around 13 failure modes identified in GALXEE AI CfAM's planning material and refined into a technical threat taxonomy. These failure modes are not treated as marketing labels; they become test categories for red-team evaluation, policy design, sandbox controls, and benchmark development.

Failure Mode	Risk Scenario	CFAM Control Strategy
Overpowered AI Assistant	Agent receives broader permissions, data, or tools than necessary.	Least-privilege capability manifest; time-bounded credentials.
Calendar Invite From Hell	Agent creates recursive invites, leaks metadata, or schedules unauthorized meetings.	Temporal policy constraints; approval triggers for sensitive scheduling.
Data Leak Gift Basket	PII, business secrets, or regulated data leave through memory, logs, or outputs.	Data minimization; masking; output inspection; audit logs.
Prompt Injection Mind Control	Malicious instructions hidden in user input, documents, emails, audio, or API results hijack behavior.	Semantic injection detection; action validation; default-deny execution.
Tool Misuse Disaster	Agent calls APIs or tools outside task scope.	Tool-call schema validation; signed permits; manifest-based controls.
Procurement Idiot	Agent commits funds, purchases items, or approves vendor actions autonomously.	Cost bounds; human-in-the-loop approval; transaction policies.
Remote Code Execution	Agent is induced to run attacker-supplied code.	Ephemeral sandboxing; no host write access; restricted egress.
Supply Chain Attack	Malicious package, plugin, model component, endpoint, or connector enters the agent chain.	Verified endpoint registry; signed manifests; dependency allowlists.
Hallucinated Legal Advice	Agent presents false or unsupported advice as authoritative.	Grounded response checks; domain-risk classifiers; escalation rules.
Customer Service Chatbot Drift	Agent moves away from defined persona, scope, or business objective.	Objective alignment monitoring; state checks; conversation boundary rules.
Deepfake CEO Scam	Voice-cloned or spoofed authority figure instructs agent to take sensitive action.	Identity verification; voice-risk scoring; high-authority approval gates.
AI Worm	Compromised agent spreads adversarial payloads across tools or agents.	Stateless execution; propagation limits; isolated action contexts.
Mental Health Safety Failure	Agent mishandles crisis, self-harm, or high-risk emotional content.	Risk detection; hard-stop escalation; safe handoff to human review.

4. Innovation

The central innovation is a shift from instruction-layer safety to containment-first execution security. Current approaches attempt to make the model safer by refining prompts, policies, filters, and training. CFAM assumes the model can still be confused, manipulated, or wrong, then limits what the model can actually do. This is the same security philosophy used in mature

computing environments: do not trust every process merely because it claims to be helpful; constrain it, verify it, isolate it, and log it.

4.1 Novel Technical Contributions

- **Cryptographically enforced capability manifests:** Each agent task receives signed, machine-readable permissions specifying tools, scopes, data access, allowed endpoints, time windows, and approval thresholds.
- **Formal task-boundary modeling:** Policy rules will be represented in a constrained formal logic style suitable for automated validation, with future extensibility toward Linear Temporal Logic and model checking.
- **Stateless tool-call execution:** Agent-requested actions will run in short-lived, isolated execution contexts to reduce persistence, lateral movement, and host compromise risks.
- **Independent pre-execution Sentry Node:** Action validation occurs outside the model's own reasoning stream, reducing the risk that a compromised model can self-authorize dangerous behavior.
- **Agentic failure-mode benchmark:** The project will produce repeatable evaluation scenarios across all 13 failure modes, enabling defensible measurement of containment performance.

4.2 Why This Is More Than a Product Feature

CFAM is not simply a better prompt template or a compliance checklist. It is a middleware architecture that can be deployed across voice agents, chat agents, back-office agents, CRM automation, procurement workflows, and multi-agent systems. Its value is highest wherever an AI system has authority to act. The same containment layer that prevents a receptionist from leaking customer records can help a government workflow agent avoid unauthorized tool use or a defense-relevant agent network avoid lateral propagation.

This makes the project technically risky enough to justify R&D funding, but commercially grounded enough to justify a realistic transition plan. The company already has practical context in agentic voice receptionist workflows, which provides a useful first deployment path and real-world interaction data. The R&D novelty lies in turning those workflows into a generalized security layer for autonomous agents.

5. Technical Approach

The CFAM architecture decouples "reasoning" from "doing." The LLM or agent framework may interpret a request and propose an action, but that proposed action is serialized and passed through the containment pipeline before it reaches any real system. If the action lacks a valid

policy permit, exceeds scope, presents injection risk, attempts unauthorized data access, or triggers a harm category, execution is denied or escalated.

5.1 Layer 1: Policy Engine

The Policy Engine defines the operational envelope of an agent at deployment time and at task time. Instead of relying on an instruction such as “do not access confidential data,” the Policy Engine issues a signed capability manifest specifying which data, APIs, tools, time windows, and action types are permitted. Any action outside the manifest is rejected by default.

- Define task profiles for receptionist calls, CRM updates, scheduling workflows, procurement requests, support escalations, and government workflow actions.
- Compile each task profile into a signed manifest with tool scope, data scope, endpoint scope, maximum cost/risk thresholds, and required approval conditions.
- Create a policy decision function that evaluates proposed actions against manifest constraints before execution.
- Maintain immutable policy versions to support auditability and forensic review.

5.2 Layer 2: Isolation Layer

The Isolation Layer prevents tool calls and code execution from touching the host system directly. Agent-requested actions are redirected into ephemeral execution contexts with restricted file access, restricted network egress, no persistent memory, and a narrow runtime envelope. The Phase I prototype will evaluate container and micro-VM patterns appropriate for low-latency voice and enterprise workflows.

- Prototype stateless execution contexts for API calls, code snippets, file transformations, webhook requests, and database queries.
- Restrict network egress to pre-approved endpoints tied to the active manifest.
- Use read-only content stores and scoped write paths to prevent persistent compromise.
- Measure the latency tradeoff between full isolation, pre-warmed sandbox pools, and production voice-agent requirements.

5.3 Layer 3: Sentry Node

The Sentry Node is an independent validator operating as a separate trust domain from both the LLM and the execution environment. It reviews proposed actions before execution and returns a signed permit only when the action passes structural, semantic, identity, and harm-risk checks.

- Structural validation: verify that each action matches approved schemas and manifest constraints.

- Semantic validation: detect prompt injection, policy override attempts, and suspicious tool-chaining patterns.
- Identity validation: apply additional checks for high-authority instructions, including deepfake and spoofing risk flags.
- Harm validation: identify legal, medical, mental health, financial, privacy, and regulated-domain risk categories.
- Audit validation: store action, decision, manifest version, model metadata, and escalation path in an append-only log.

5.4 System Flow

Step	Process	Example
1	User or system request enters agent workflow.	Call, chat, email, API trigger, CRM event.
2	LLM/agent proposes an action.	Schedule meeting, update CRM, send message, call API, retrieve record.
3	Action serialized into CFAM request.	Includes task ID, proposed tool, inputs, requested data, and target endpoint.
4	Policy Engine checks manifest.	Rejects unauthorized scope, expired credentials, excessive authority, or missing approvals.
5	Sentry Node validates risk.	Checks injection, spoofing, harmful output, drift, and sensitive data exposure.
6	Isolation Layer executes approved action.	Runs in constrained sandbox with restricted network and file access.
7	Audit trail records outcome.	Logs decision, evidence, denial reason, permit, and escalation status.

6. R&D Plan and Work Packages

The proposed $500,000 project will be completed over an 18-month period with a six-month technical feasibility core followed by prototype hardening, pilot validation, and transition preparation. This structure is intentionally grant-friendly: it separates research uncertainty from product execution while still producing measurable deliverables at every stage.

6.1 Research Objectives

1. Design and validate a cryptographically signed capability manifest schema for autonomous agent tasks.
2. Build a low-latency Policy Engine capable of evaluating proposed agent actions against task-specific authority boundaries.
3. Prototype stateless isolated execution for agent tool calls and measure latency, reliability, and containment performance.

4. Develop an independent Sentry Node that detects prompt injection, unauthorized tool use, identity spoofing, harmful outputs, and policy violations before execution.
5. Create a repeatable benchmark suite covering all 13 agentic AI failure modes.
6. Demonstrate CFAM across at least three LLM/agent environments and two real-world workflow categories, including voice receptionist and enterprise task automation.

6.2 Work Package Summary

Work Package	Focus	Timing	Deliverables
WP1	Threat Model and Requirements	Months 1-2	Formalize 13 failure modes; map to controls; define success metrics; identify data and compliance requirements.
WP2	Policy Engine Prototype	Months 2-5	Manifest schema, signed permissions, task scope compiler, policy decision API, audit metadata.
WP3	Isolation Layer Prototype	Months 3-7	Sandbox execution, restricted egress, read-only file access, teardown behavior, latency tests.
WP4	Sentry Node Prototype	Months 5-9	Pre-execution validation, injection detection, identity-risk checks, harm taxonomy classifiers, denial/escalation logic.
WP5	Benchmark and Red-Team Evaluation	Months 8-13	Adversarial scenarios, test harness, exploitability scoring, false positive/negative analysis.
WP6	Pilot Integration and Transition	Months 12-18	Design partner pilots, enterprise SDK documentation, grant Phase II package, federal transition briefing.

6.3 Phase I Technical Milestones

Milestone	Description	Target	Success Evidence
M1	Threat taxonomy and test plan approved	Month 2	13 failure modes converted into testable cases and scoring rubric.
M2	Capability manifest v0.1 implemented	Month 4	Signed policy manifests used in live agent action checks.
M3	Sandbox execution prototype operational	Month 6	Approved tool calls execute in isolated contexts with restricted egress.
M4	Sentry Node MVP operational	Month 8	Pre-execution validator blocks prompt injection and unauthorized tool use scenarios.
M5	Three-model compatibility demonstrated	Month 10	CFAM tested with at least three LLM/agent backends.
M6	Red-team benchmark completed	Month 13	Measured containment performance across all 13 failure modes.
M7	Pilot integration package completed	Month 18	SDK, technical brief, security documentation, Phase II plan, and design partner report.

7. Key Research Questions

The project is intentionally structured around measurable research uncertainty rather than routine software development. The following questions define the R&D agenda:

7. Can signed capability manifests express useful agent task boundaries without becoming too rigid for real business workflows?
8. What policy evaluation methods can maintain sub-10ms total enforcement overhead in real-time voice and customer-service environments?
9. Which sandbox design provides the best balance between isolation strength, cold-start latency, cost, and developer usability?
10. Can semantic injection detection identify malicious instructions embedded in natural speech, documents, emails, retrieved knowledge, and API responses without generating unacceptable false positives?
11. How should high-authority instructions be verified when the source may be spoofed through deepfake voice, compromised email, or fraudulent caller identity?
12. Can the benchmark suite produce repeatable, reviewer-defensible measurements that distinguish containment from ordinary prompt guardrails?
13. What evidence package is needed for government, healthcare, financial services, and critical infrastructure buyers to trust a third-party agent containment layer?

8. Evaluation Plan and Success Metrics

The evaluation plan will compare baseline agent workflows against CFAM-protected workflows. Each baseline workflow will be exposed to adversarial prompts, manipulated data, unauthorized tool requests, spoofed identities, malicious API responses, and high-risk content scenarios. The project will measure how often the baseline agent fails and how often CFAM prevents, denies, or escalates the unsafe action.

Metric	Definition	Target
Exploitability reduction	Percentage reduction in successful attacks across 13 failure modes.	Target: 90-95% reduction in controlled red-team tests.
Latency overhead	Added milliseconds from policy check, sentry validation, and isolated execution.	Target: <10ms for policy + sentry checks; sandbox targets measured by action type.
Prompt injection recall	Percentage of adversarial injection attempts correctly blocked.	Target: >95% in Phase I; stretch target >99% after dataset expansion.
False positive rate	Legitimate actions incorrectly blocked or escalated.	Target: <3% in feasibility prototype; improve through policy tuning.
Model compatibility	Number of LLM/agent backends tested.	Target: at least three backends; architecture remains model-agnostic.
Audit completeness	Percentage of actions with usable forensic records.	Target: 100% of permitted, denied, and escalated actions.

Pilot readiness	Ability to integrate into a real voice or enterprise workflow.	Target: two pilot-grade integrations by project close.

Evaluation artifacts will include a test plan, adversarial scenario library, benchmark report, latency report, incident response simulation notes, design partner integration summary, and an updated technical roadmap. These artifacts will be reusable in future NSF, NIST, DARPA, DHS, and Colorado OEDIT submissions.

9. Technical Risks and Mitigation Plan

Risk	Concern	Mitigation
Policy rigidity	Manifests may block legitimate novel actions.	Use tiered policy templates, supervised overrides, and human review queues for unknown actions.
Latency burden	Security checks may slow real-time voice workflows.	Separate low-latency policy checks from heavier sandbox execution; use caching and pre-warmed pools.
False positives	Sentry Node may block useful actions too often.	Build benchmark-driven tuning and severity-based escalation instead of binary denial only.
Adversarial adaptation	Attackers may learn to evade known injection detectors.	Combine structural controls, semantic detection, and execution restrictions so no single detector carries all risk.
Sandbox complexity	Production isolation may be costly or difficult to operate.	Evaluate multiple isolation strategies and match level of isolation to action risk tier.
Market education	Customers may not understand containment versus guardrails.	Use failure-mode demos, benchmark scores, and compliance-aligned evidence packages.
Federal transition	Government buyers may require additional compliance and security documentation.	Map controls to NIST AI RMF, NIST GenAI Profile, OWASP LLM Top 10, and future ATO/FedRAMP-readiness needs.

10. Broader Impacts and Public Benefit

The public benefit of this project is straightforward: autonomous AI agents will increasingly handle sensitive tasks. If they are deployed without containment, failures will affect privacy, public trust, cybersecurity, healthcare, finance, government operations, and critical infrastructure. CFAM reduces the risk that useful automation becomes an attack surface.

- **Critical infrastructure protection:** The architecture is applicable to healthcare communications, financial workflows, emergency services, public-sector call routing, and government back-office automation.
- **Trustworthy AI measurement:** The benchmark suite will help organizations evaluate agentic risk in a repeatable way instead of relying on vendor claims.
- **Small business competitiveness:** GALXEE AI CfAM can demonstrate that advanced AI safety infrastructure can be built by a Denver-based small business, not only by large coastal technology companies.

- **Workforce development:** The project will generate technical documentation, implementation patterns, and security training material for developers adopting containment-first design.
- **Dual-use transition:** The same controls that secure commercial agents can support defense, intelligence, and public-sector workflows where bounded autonomy is mission-critical.

11. Commercialization and Transition Plan

GALXEE AI CfAM's commercialization strategy begins with its existing agentic voice and receptionist context, then expands into broader AI security middleware. This is a practical wedge: voice agents are exposed to untrusted inputs, identity spoofing, CRM access, scheduling systems, and sensitive customer data, making them a strong first market for containment-first controls.

11.1 Target Customer Segments

- **Enterprise voice and customer service teams:** Secure AI receptionists, support agents, call-routing systems, CRM updates, and escalation workflows.
- **Healthcare and mental health operations:** Prevent hallucinated medical guidance, mishandled crisis conversations, and unauthorized PHI exposure.
- **Financial and procurement workflows:** Limit unauthorized purchases, data exfiltration, credential misuse, and deepfake-authorized transactions.
- **Government and defense programs:** Support bounded autonomy, multi-agent communication, auditability, and secure workflow automation.
- **AI platform developers:** Offer CFAM as a developer API or embedded middleware for third-party agent builders.

11.2 Revenue Model

- Subscription tiers for small business, professional, enterprise, and regulated-sector deployments.
- Developer API pricing for third-party agent platforms using CFAM containment checks per action or per workflow.
- Professional services for custom policy manifests, security validation, and compliance mapping.
- Non-dilutive R&D contracts and grants through federal, state, and AI safety programs.
- Future government contracting channel through GSA/FedRAMP-aligned deployment planning once the product is mature enough.

11.3 Transition Roadmap

Period	Stage	Primary Outcome
Months 1-6	Feasibility core	Threat model, manifest prototype, policy engine MVP, initial sandbox tests.
Months 7-12	Prototype validation	Sentry Node, injection scenarios, model compatibility, latency report.
Months 13-18	Pilot and transition	Design partner integrations, benchmark report, SDK docs, Phase II package.
Months 19-30	Phase II / seed expansion	Production CFAM v1.0, enterprise pilots, compliance documentation, government partner outreach.
Months 31+	Scale	Security-as-a-Service platform, API marketplace, formal certification pathway, public-sector procurement readiness.

12. Budget Use and Staffing Plan

The requested $500,000 will be allocated to applied R&D, prototype development, security testing, cloud infrastructure, technical documentation, and commercialization preparation. Final budget categories should be adjusted to match the specific grant form, but the following allocation is appropriate for a project of this scope.

Category	Estimated Amount	Purpose
Technical personnel and contractors	$210,000	Policy engine, sandbox architecture, Sentry Node, integrations, QA.
Security research and red-team testing	$85,000	Adversarial test design, external review, penetration testing, benchmark validation.
Cloud, infrastructure, and tooling	$55,000	Development environments, sandbox testing infrastructure, logging, security monitoring.
Prototype integrations and design partners	$60,000	Voice agent integration, CRM workflow integration, user feedback, pilot support.
Documentation, compliance, and standards mapping	$35,000	NIST/OWASP mapping, technical white papers, grant reporting, audit evidence package.
Project management and grant administration	$35,000	Milestone tracking, reporting, budgeting, partner coordination.
Contingency / indirect costs	$20,000	Unplanned technical requirements, procurement, legal/IP review.

Key personnel should include a technical lead for architecture, a security engineer for sandbox and adversarial testing, a machine-learning engineer for detection and benchmark design, a backend engineer for APIs and logging, and a commercialization lead for pilots and grant reporting. GALXEE AI CfAM should also recruit at least one advisor or partner with cryptography, formal verification, or federal cybersecurity experience to strengthen review credibility.

13. Alignment with Government Priorities

The project aligns strongly with current government priorities around trustworthy AI, secure software, critical infrastructure protection, AI measurement science, and responsible deployment of generative AI. The attached source document repeatedly positions CFAM around DARPA, NSF, NIST, DHS, and Colorado OEDIT opportunities; this project description consolidates that direction into one coherent technical proposal.

- **NSF-style innovation fit:** The work is technically novel, has measurable R&D uncertainty, and offers a commercial path through AI security middleware.
- **NIST-style measurement fit:** The benchmark suite and failure-mode taxonomy support repeatable AI risk evaluation and standards-aligned documentation.
- **DARPA-style autonomy fit:** The architecture supports bounded agent behavior, secure multi-agent interaction, and containment against adversarial exploitation.
- **DHS-style mission fit:** The project addresses non-human identity, supply chain integrity, critical infrastructure workflows, and AI-enabled lateral movement risk.
- **Colorado OEDIT fit:** The company is Denver-based and the technology can benefit Colorado healthcare, finance, aerospace, defense, and advanced industries companies.

14. Company Capability and Feasibility Evidence

GALXEE AI CfAM is positioned to execute this project because it begins from a real applied-agent context rather than an abstract laboratory concept. The company's initial focus on AI voice receptionists creates exposure to real deployment problems: caller identity, prompt injection through speech, CRM permissions, calendar access, customer data, escalation logic, and brand-safe responses. Those are exactly the conditions that make containment valuable.

Feasibility evidence to emphasize in the final grant package includes:

- Existing company website and commercial positioning at www.wearegalxee.ai.
- Defined technical architecture: Policy Engine, Isolation Layer, and Sentry Node.
- Named failure taxonomy with 13 testable categories.
- Clear Phase I technical milestones and measurable success metrics.
- Dual-use transition path from voice agents to enterprise/government agent security.
- Planned alignment with NIST AI RMF, NIST Generative AI Profile, OWASP LLM Top 10, and federal cybersecurity procurement expectations.

The most important feasibility gap is not conceptual; it is execution capacity. For a $500,000 grant, GALXEE AI CfAM should use funds to add or contract technical depth in cybersecurity engineering, formal policy systems, adversarial testing, and cloud infrastructure. Reviewers will want to see not only the idea, but the team and partners capable of proving it. The document

should therefore be paired with resumes, partner letters, and at least one design-partner letter of interest before submission.

15. Conclusion

GALXEE AI CfAM's proposed Containment-First Agentic Middleware addresses a timely and consequential problem: autonomous AI agents are gaining the ability to act faster than organizations are gaining the ability to control them. Existing guardrails are important, but they are not enough. A high-consequence agentic system needs enforceable boundaries, isolated execution, independent validation, and measurable assurance.

The proposed project will produce a working prototype, benchmark suite, red-team validation report, pilot integration package, and federal transition materials. It will move the company from general AI automation toward a defensible AI security infrastructure position. For a half-million-dollar grant, that is the right story: not "we built a chatbot," but "we are building the containment layer that makes autonomous agents safe enough to deploy." This reframes the work as critical infrastructure rather than another AI application.

By funding this work, the grantor would support a Colorado-based small business developing a dual-use technology with commercial relevance, public benefit, and direct alignment to national priorities in AI safety, cybersecurity, trustworthy autonomy, and critical infrastructure resilience.

References and Standards Alignment

[1] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0): Govern, Map, Measure, and Manage functions for trustworthy AI risk management.

[2] National Institute of Standards and Technology. NIST AI 600-1, Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, July 2024.

[3] OWASP Foundation. OWASP Top 10 for LLM Applications 2025, including Prompt Injection, Sensitive Information Disclosure, Supply Chain, and Excessive Agency categories.

[4] NSF America's Seed Fund. Project Pitch guidance emphasizing technical innovation, R&D risk, market opportunity, and company/team fit.

[5] GALXEE AI CfAM planning document supplied by applicant: grant discovery, CFAM architecture notes, DARPA/NSF positioning, failure-mode taxonomy, and proposal outline.