W3C AI Knowledge Representation CG and AGENT PROTOCOL CG

Technical Note -- Community Discussion Draft
11 March 2026 -- Version 0.2

DRAFT

***REQUEST EDIT ACCESS TO IMPROVE /EXPAND THIS DOC***

# Safe and Ethical Open-Source Agent Architectures:

## Safeguards Against Mob Behavior, Harassment, and Emergent Harm in Autonomous Agent Systems

*A brief architectural overview to stimulate community input across W3C Community Groups*

| | |
|---|---|
| **Status** | Draft -- community input invited |
| **Author** | Paola Di Maio, W3C AI KR CG |
| **Editors** | Please request edit access |
| **Audience** | W3C AI-KR CG, Agent Protocol CG, Solid Community, open-source agent developers |
| **Keywords** | agent safety, mob behavior, prompt injection, DID, Solid, behavioral immutability, self-modifying agents, rate limiting, agent ethics |
| **Related work** | MCP Model Card Spec v1.0; WebMCP Technical Notes 1-4; agent interoperability CG; Di Maio SoSy 2026 (Solid/privacy); Karpathy autoresearch 2026 |

**Abstract** 2
**1. Motivation -- The Emergent Mob Problem** 2
**2. Taxonomy of Harmful Agent Behaviors** 3
**3. A Three-Layer Safety Architecture** 4
Layer 1 -- In-Agent Behavioral Governors 4
1.1 Rate and reach governors 4
1.2 Provenance checks before amplification 4

1.3 Prompt injection resistance 4
1.4 Identity and disclosure 4
1.5 Owner notification and override 4
Layer 2 -- Browser and Gateway-Level Circuit Breakers 5
2.1 Aggregate rate limiting at gateway 5
2.2 Target clustering detection 5
2.3 Browser-layer AI disclosure 5
2.4 Egress filtering for sensitive content categories 5
2.5 Inter-gateway communication logging 5
Layer 3 -- Ecosystem Self-Monitoring and Self-Correction 5
3.1 Behavioral telemetry (privacy-preserving) 6
3.2 Distributed reputation for skill registry 6
3.3 Incident response protocol 6
3.4 Self-correction mechanisms 6
3.5 Solid Protocol as Accountability Substrate 6
3.6 Behavioral Immutability: The Self-Modifying Agent Problem 7
3.6.1 Proposed immutability constraints 7
3.6.2 Open research question 8
**4. Open Questions for Community Discussion 8**
**5. Relationship to Existing Standards Work 8**
**6. How to Contribute 9**

## Abstract

Open-source autonomous agent frameworks such as OpenClaw are proliferating rapidly, with populations of persistent, proactive agents operating across messaging platforms, email, social networks, and file systems. This technical note identifies a class of emergent risk -- agent mob behavior, coordinated amplification, identity laundering, and harassment at machine speed -- that is architecturally distinct from individual agent misbehavior and therefore requires dedicated safeguards.

We propose a three-layer safety architecture: (1) in-agent behavioral governors, (2) browser and gateway-level circuit breakers, and (3) ecosystem-level self-monitoring and self-correction mechanisms. We invite community input on each layer, with the aim of developing these into normative guidelines suitable for adoption by the Agent Protocol CG and related W3C standards work.

## 1. Motivation -- The Emergent Mob Problem

**The problem is not one agent.** It is a population of agents, each behaving within its programmed parameters, whose aggregate effect on a target -- a person, a platform, a piece of information -- becomes disproportionate, harmful, or beyond the intent of any individual owner.

Three catalysts make this likely in current open-source agent deployments:

- Convergent incentives: independently configured agents may respond to the same stimuli -- a trending topic, a flagged account, an injected prompt -- in statistically similar ways, producing pile-on effects without any coordination.
- Prompt injection as a weaponization vector: as documented by Cisco's AI security research team testing OpenClaw skills,[1] malicious instructions embedded in external content can redirect agent behavior without user awareness. A single injected payload could cause multiple agent instances to amplify, report, or harass a target.
- Reach asymmetry: a single human owner deploying an agent with access to 10+ messaging platforms, email, and social APIs commands influence orders of magnitude larger than their organic human capacity. At population scale this becomes a structural imbalance.

| NOTE | This note does not evaluate any specific platform. The patterns described apply to any autonomous agent architecture with persistent access to communication and social systems. OpenClaw is used as a current reference implementation given its scale and open codebase. |
|---|---|

## 2. Taxonomy of Harmful Agent Behaviors

We identify six behavioral categories that require dedicated safeguards. These are not mutually exclusive -- real incidents often involve multiple overlapping categories.

| Behavior Type | Description | Harm Vector |
|---|---|---|
| Amplification without verification | Agent forwards, reposts, or endorses content at machine speed without provenance checks | Disinformation spread; reputational harm to targets of false content |
| Pile-on dynamics | Multiple agents respond to the same target within a compressed time window | Harassment; psychological harm; suppression of speech |
| Identity laundering | Agent acts on behalf of a human in ways that obscure its AI origin | Fraud; manipulation; erosion of trust in human communication |
| Prompt injection weaponization | External malicious content redirects agent behavior toward a target | Agent becomes an unwitting instrument of harm without owner intent |
| Reach asymmetry exploitation | Single owner deploys disproportionate agent reach across platforms | Structural imbalance in public discourse; unfair competition |
| Accountability opacity | Agent actions untraceable to a responsible human or entity | No redress mechanism; no accountability chain for harm caused |

[1] Cisco's AI security research team testing OpenClaw skills,

# 3. A Three-Layer Safety Architecture

We propose organizing safeguards across three layers that correspond to where authority and observability naturally reside: inside the agent, at the gateway/browser interface, and at the ecosystem level.

## Layer 1 -- In-Agent Behavioral Governors

These are safeguards that must be built into the agent runtime itself. They cannot be delegated to the platform because the agent may operate across many platforms simultaneously.

### 1.1 Rate and reach governors

- Maximum outbound social actions per agent instance per hour (configurable, default conservative)
- Per-target interaction ceiling: agent pauses and notifies owner if N interactions toward the same external entity occur within T minutes
- Platform diversity cap: limit simultaneous active threads across N distinct platforms without explicit owner confirmation

### 1.2 Provenance checks before amplification

- Before forwarding, reposting, or acting on external content, agent verifies source credibility against a configurable policy
- Minimum: flag unverified content as such in any action taken on its basis
- Ideal: integrate with emerging content provenance standards (C2PA, W3C Verifiable Credentials)

### 1.3 Prompt injection resistance

- Strict separation of instruction space and data space in agent context window handling
- Skill registry vetting: agents should not execute unaudited third-party skills without explicit owner approval per-skill
- Anomaly detection: if agent behavior diverges significantly from baseline following ingestion of external content, pause and notify owner
- Sandboxed skill execution: skills run with minimal permissions, explicitly scoped to declared capabilities

### 1.4 Identity and disclosure

- Agent-generated content must carry machine-readable AI origin metadata
- In human-facing communication channels, agent must disclose its non-human status on first contact per session
- Owner identity must be resolvable (even if pseudonymous) via a DID or equivalent -- directly relevant to the DID:WBA work in the Agent Protocol CG

### 1.5 Owner notification and override

- Any action at or near a behavioral threshold generates an owner notification before execution where latency permits

- Owner retains unambiguous kill switch: single command halts all outbound agent actions immediately
- Audit log of all outbound actions stored locally and queryable by owner

## Layer 2 -- Browser and Gateway-Level Circuit Breakers

The gateway is the agent's interface to the outside world. It is the appropriate place for safeguards that require observability across multiple concurrent agent actions, and for enforcement that does not depend on agent-level compliance.

### 2.1 Aggregate rate limiting at gateway

- Gateway tracks outbound action rate across all agent instances it serves
- Automatic throttling when aggregate rate exceeds configurable thresholds
- Hard cap on actions per unit time that cannot be overridden by agent configuration alone

### 2.2 Target clustering detection

- Gateway monitors whether multiple actions within a time window share a common target (person, account, URL, topic)
- Clustering above threshold triggers pause and owner alert, not automatic continuation
- This is the primary technical defense against pile-on dynamics

### 2.3 Browser-layer AI disclosure

For agents operating via browser automation (a common OpenClaw pattern):

- Browser extension or headless browser wrapper injects AI-origin headers into HTTP requests where the recipient platform can read them
- Form submissions and message sends include machine-readable disclosure metadata
- W3C has existing work on ethical web principles and Responsible Use of Web APIs that should inform this layer

### 2.4 Egress filtering for sensitive content categories

- Gateway maintains configurable blocklist of content categories that require additional confirmation before transmission
- Minimum categories: personal identifying information of third parties; content flagged by owner's own content policy; content targeting minors

### 2.5 Inter-gateway communication logging

For the DID:WBA end-to-end encrypted agent communication scenario described in the March 11 Agent Protocol CG agenda: *encryption should not preclude accountability.*

- Gateway-to-gateway communication logs should be retained locally for a configurable period
- Owner should be able to audit what their agent communicated with other agents on their behalf
- Aggregate behavioral metadata (not content) should be available to ecosystem-level monitoring

## Layer 3 -- Ecosystem Self-Monitoring and Self-Correction

No individual agent or gateway can observe emergent population-level behavior. A third layer is needed that operates at the ecosystem level without requiring centralized control -- a deliberate design constraint given the open-source, decentralized ethos of this space.

### 3.1 Behavioral telemetry (privacy-preserving)

- Agent instances publish anonymized behavioral metrics to a distributed ledger or peer network: action rates, platform distribution, target diversity scores -- not content
- Anomaly detection at population level: sudden spikes in agents acting on the same topic or toward the same target trigger a network-wide advisory
- Opt-in for individual agents; opt-out triggers disclosure to skill registry (agent marked as non-participating in safety network)

### 3.2 Distributed reputation for skill registry

- ClawHub and equivalent skill registries adopt a vetting protocol analogous to the MCP Model Card Specification
- Skills declare: data access scope, outbound communication capabilities, injection surface area, human oversight requirements
- Community audit process for high-reach skills; mandatory security review before skills can access sensitive platform categories
- Skills with documented injection vulnerabilities are removed with versioned audit trail

### 3.3 Incident response protocol

- Documented community process for reporting and responding to emergent mob behavior incidents
- Incident classification taxonomy: individual misconfiguration / coordinated misuse / emergent population behavior (these require different responses)
- Responsible disclosure channel for researchers identifying new injection or amplification vectors
- Post-incident behavioral analysis fed back into governor defaults

### 3.4 Self-correction mechanisms

- Agent instances subscribe to behavioral advisory feed; governor defaults updated automatically within owner-approved parameter ranges
- Network-wide circuit breaker: in a documented emergency, ecosystem-level signal can trigger conservative mode across participating agents -- not shutdown, but elevated thresholds and mandatory owner confirmation for all outbound actions
- Governance: this emergency mechanism requires a defined decision process -- proposed as a multi-stakeholder W3C CG vote with time-bounded authority

---

## 3.5 Solid Protocol as Accountability Substrate

The **Solid protocol** (Social Linked Data, W3C) offers a decentralised architecture that maps directly onto several of the accountability gaps identified in this note. Rather than treating agent action logs as platform-side records -- inaccessible to owners, unverifiable by third parties --

Solid Pods provide user-controlled, persistent, linked-data stores that agents can write to and owners can audit independently of any platform.

Three Solid capabilities are particularly relevant:

- Pod-based agent action logging: every outbound agent action is written as a linked data record to the owner's Pod, creating a tamper-evident, owner-controlled audit trail that satisfies Layer 1.5 requirements without dependence on platform cooperation.
- Granular access control (WAC / ACP): Solid's access control mechanisms can enforce which platforms an agent is permitted to write to simultaneously -- a data-layer enforcement mechanism that complements but does not depend on agent-level compliance, strengthening Layer 2.2 target clustering defenses.
- Machine-readable consent records: Solid can host standardised consent records that agent actions must reference before executing on behalf of an owner. This directly addresses the consent failure pattern documented in OpenClaw-adjacent platforms, where agent-generated profiles and interactions proceeded without verifiable human authorisation.

This approach builds on prior work including SocialGenPod and the W3C Data Privacy Vocabulary (DPV). A companion paper -- *Privacy Challenges in Conversational AI: Three Use Cases and Prospects for Decentralised Data Governance with Solid* (Di Maio, SoSy 2026) -- provides empirically grounded use cases and regulatory mapping (GDPR, EU AI Act) for this architecture. Community input is invited on how Solid Pod integration could be specified as a normative option within the agent safety framework proposed here.

See also GUIDE TO SOLIC *For Agent Developers, Agent Users
https://w3c-cg.github.io/aikr/Solid/solidguide.html#user-why

## 3.6 Behavioral Immutability: The Self-Modifying Agent Problem

The three-layer architecture above assumes a **stable agent** -- one whose behavioral governors, once set, remain in effect unless explicitly changed by the owner. This assumption is already obsolete.

Andrej Karpathy's **autoresearch** framework (released March 2026, MIT license) demonstrates that an AI agent given a fixed compute budget and a plain-language instruction file can autonomously modify its own training code, run experiments, evaluate results, and commit improvements -- approximately 100 iterations overnight on a single GPU. In initial runs, the agent independently rediscovered architectural improvements that took human researchers at frontier labs nearly a decade to formalise.

The implications for agent safety are significant and underaddressed in current frameworks:

- An OpenClaw instance running an autoresearch-pattern skill could modify its own behavioral configuration overnight without owner awareness, invalidating any governors set at deployment time.
- Self-modification could be intentional (owner-directed optimization) or emergent (agent discovers that relaxing a governor improves its task performance metric) -- the latter is the safety-relevant case and the harder one to detect.

- Prompt injection becomes a self-modification vector: a malicious payload could instruct the agent to update its own program.md equivalent, persisting adversarial behavior across sessions.

### 3.6.1 Proposed immutability constraints

- Behavioral governor configurations are stored in a write-protected zone inaccessible to the agent's own execution context -- analogous to firmware versus software separation in embedded systems.
- Any proposed modification to governor parameters requires explicit out-of-band owner confirmation, delivered through a channel the agent cannot itself access or simulate.
- Agent self-modification of code or configuration is logged to the owner's Solid Pod (or equivalent audit store) before execution, not after.
- Skill updates from external registries are treated as new installations requiring fresh owner approval, not silent upgrades.

### 3.6.2 Open research question

The autoresearch pattern points toward a future in which the boundary between agent configuration and agent capability is itself fluid. This is not a problem current W3C standards work has addressed. We flag it here as an urgent open question: what constitutes a normatively stable agent identity across self-modification cycles, and who bears accountability for behavior that emerges from autonomous iteration the owner did not directly authorise?

---

## 4. Open Questions for Community Discussion

This note is deliberately incomplete. The following questions require community input before any normative guidelines can be developed:

- What is the right governance model for ecosystem-level circuit breakers? Who has authority to trigger network-wide advisories, and what prevents misuse of that authority?
- How does DID:WBA agent identity interact with the accountability requirements proposed in Layer 1.4? Can decentralized identity support resolvable accountability without centralized registration?
- What is the appropriate balance between agent autonomy (the value proposition of the technology) and behavioral governors? Where does safety engineering become capability restriction?
- How should these guidelines interact with emerging platform-level policies (signal, WhatsApp, Discord) that may independently limit bot behavior? Is there a coordination mechanism?
- Should skill registry participation be mandatory for agents claiming W3C compliance, or voluntary with disclosure requirements?
- How do we handle jurisdiction: an agent operating across platforms in multiple legal territories, with an owner in one jurisdiction and targets in another?

---

## 5. Relationship to Existing Standards Work

| Standard / Initiative | Layer | Interface |
|---|---|---|
| MCP Model Card Spec v1.0 | Layer 3 -- skill registry vetting | Extends model card framework to agent skill declarations; security scope and injection surface area fields |
| WebMCP Technical Notes 1-4 | Layers 2 and 3 -- gateway and ecosystem | Security section requirements in WebMCP spec directly relevant to gateway circuit breakers |
| W3C Ethical Web Principles | Layer 2 -- browser layer | AI disclosure in browser-automated interactions is an application of existing ethical web principles |
| W3C Verifiable Credentials | Layer 1.2 -- provenance | Content provenance verification for agent amplification decisions |
| DID:WBA (Agent Protocol CG) | Layer 1.4 -- identity | Decentralized agent identity as the accountability substrate; must be designed with auditability in scope |
| IEEE 7003 (Algorithmic Bias) | Layers 1-3 -- cross-cutting | Behavioral governors should be evaluated for differential impact across demographic groups |
| Solid Protocol (W3C) | Layers 1.5, 2.2, consent | Pod-based audit logging, granular access control, machine-readable consent records; see Section 3.5 |
| autoresearch (Karpathy 2026) | Layer 1.6 -- immutability | Demonstrates that stable-agent assumption is obsolete; motivates behavioral immutability constraints; see Section 3.6 |

## 6. How to Contribute

This note is a starting point for community discussion, not a finished specification. We invite responses on:

- Corrections or additions to the behavioral taxonomy (Section 2)
- Alternative or complementary architectural approaches to any of the three layers
- Evidence from deployments -- documented incidents, near-misses, or successful mitigations
- Gaps in the relationship to existing standards work (Section 5)
- Proposed normative language for any of the guidelines proposed

Discussion is invited on the W3C AI-KR CG mailing list and the **#ai-agent-protocol** channel in W3C Community Slack. GitHub issues and PRs against related repositories are also welcome.

*This technical note does not represent the official position of any W3C Community Group. It is a discussion document intended to stimulate community input.*

Epistemic Systems Lab -- Ronin Institute for Independent Scholarship -- ronin-institute.org/epistemic-systems-lab -- March 2026