+OS Infinity - Safe Agent

Whitepaper

Version 1.0 — February 2026

Author: Jay Rushton
Website: https://www.osinfinity.com
Email: jay@osinfinity.com

One system – infinite possibilities

Infinity Safe Agent is a pioneering governance layer for AI behaviour, introducing deterministic safety, truth transparency, and neurodivergent-friendly reasoning. This white paper details the architecture, AML specification, and implementation guidance for safe, certifiable AI systems.

Infinity Safe Agent — Whitepaper.

Version 1.0 — 21 February 2026
Author: Jay
Website: https://www.osinfinity.com
Email: jay@osinfinity.com

Executive Summary

Infinity Safe Agent introduces a deterministic governance layer for AI behaviour. Unlike traditional models, which decide what is safe internally, Infinity separates proposing from governing. Models only propose actions; a runtime arbiter, governed by AML (Action Model Language), validates every action before execution. This architecture eliminates hallucinations, blocks unsafe behaviour, enforces truth transparency, and guarantees neurodivergent-friendly reasoning patterns. It works with any model, scales across domains, and is fully testable offline.

1. Introduction

Modern AI systems generate natural language directly from user prompts, leading to several issues:

- Non-deterministic outputs
- Safety dependent on model internals
- Inconsistent truth handling
- Hallucinations presented as facts
- Difficult-to-audit safety failures
- Inconsistent support for neurodivergent users
- Unreliable behaviour in regulated domains (health, legal, crisis)

Infinity Safe Agent addresses these by introducing:

- A formal language for AI behaviour
- A deterministic runtime for safety governance
- A truth-handling layer
- ND-friendly logic patterns
- Topic-based safety enforcement
- Age-profile restrictions
- Trusted source handling

This creates a predictable, testable, and certifiable system.

2. Architecture Overview

2.1 System Pipeline

Infinity separates the AI pipeline into two roles:

- **Proposer (LLM):** Any model (Claude, GPT, etc.) generates an initial behavioural plan.
- **Governor (Infinity Runtime):** A deterministic arbiter validates and transforms the plan before execution.

Pipeline Flow:

1. User Input
2. LLM Proposer
3. AML Plan (AGENT_PLAN with topics, risk, steps)
4. Arbiter:
    - Safety Rules → block?
    - Truth Rules → label
    - Logic Patterns → transform
5. Safe Output (user never sees unsafe content)

2.2 Key Principles

- Determinism: same input yields same output
- Model-agnostic: works with any LLM
- Testability: AML plans validated offline
- Transparency: explicit safety and truth rules
- Governance: behaviour controlled by runtime, not model
- Safety: protected topics blocked or redirected
- ND-friendly: logic patterns ensure clarity and grounding

3. AML Specification (Open Standard)

3.1 AML Structure

AML (Action Model Language) is a declarative language for modelling AI behaviour as structured plans.

AGENT_PLAN Structure:

- id: string
- goal: string
- CONTEXT:
  - topics: [string]
  - risk_level: string
  - user_state?: object
- STEPS:
  - STEP:
    - id: string
    - type: string
    - text: string
    - options?: [string]

3.2 Step Types (10)

- show_message
- ask_question
- suggestoptionss
- confirm_action
- show_template
- call_capability
- branch
- loop
- wait
- log

3.3 Topics

Topics classify the domain of the plan:

- generalinfoo
- routines
- nd_support
- social_skills
- consumer_info

- health_info
- legal_info
- emotional_support
- crisis
- self_harm
- suicide
- abuse

3.4 Protected Topics (6)

Trigger safety enforcement:

- selfharmm
- suicide
- medical
- legal
- abuse
- crisis

3.5 Safety Templates

Used when unsafe content is detected:

- crisissupportt
- medicaldissclaimer
- legal_disclaimer
- safety_refusal
- age_restricted
- cannot_answer_safely

3.6 Truth Rules

- Trusted sources: prefix "Verified ({source}):"
- Untrusted sources: prefix "Unverified source — please verify:"
- No source: prefix "This is my understanding, not a verified fact:"
- Protected topic + no source: block

3.7 Logic Patterns (ND-Friendly)

- safety_first
- clarifybeforee_action
- small_steps
- explainconstraintss

These ensure clarity, grounding, and predictable reasoning.

4. Arbiter (Conceptual Overview)

4.1 Safety Rules

- Block protected topics
- Replace unsafe plans with templates
- Enforce age restrictions
- Prevent directive or harmful suggestions

4.2 Truth Rules

- Label verified information
- Flag unverified information
- Prevent hallucinations as facts
- Require trusted sources for protected topics

4.3 Logic Patterns

- Transform plans for grounding, clarity, stepwise reasoning, non-directive support, ND-friendly communication

4.4 Age Profiles

- Child: generalinfo, routiness
- Teen: adds social_skills, nd_support
- Adult: all non-protected topics

4.5 Trusted Sources

- nhs.uk
- gov.uk
- nice.org.uk
- mind.org.uk
- autism.org.uk
- youngminds.org.uk

5. Example AML Files

Include all 10 examples exactly as they appear in your repo:

- morning-routine.aml
- truth-test.aml
- unsafe-self-harm.aml
- suicide-blocked.aml
- crisis-blocked.aml
- medical-blocked.aml
- legal-blocked.aml
- abuse-blocked.aml
- age-child-blocked.aml
- age-adult-passes.aml

6. Demo Instructions

- Run all examples:

  ```
  npm install
  npm run demo
  ```

- Run individual AML files:

  ```
  npm run aml run examples/morning-routine.aml
  npm run aml run examples/truth-test.aml
  npm run aml run examples/medical-blocked.aml
  ```

- Run with a live LLM:

  ```
  cp .env.example .env
  ```

  Add ANTHROPIC_API_KEY

  ```
  npm run server
  ```

  Open http://localhost:3000

7. Live LLM Demo Results

Include the 5 real examples you already generated:

- Safe casual request
- Medical advice blocked
- Subtle crisis language detected
- ND support
- Consumer information with truth labels

8. Safety Configuration (High-Level)

- Protected Topics: selfharmm, suicide, medical, legal, abuse, crisis
- Templates: crisissupportt, medical_disclaimer, legal_disclaimer, safetyrefusall, age_restricted, cannot_answer_safely
- Trusted Sources: nhs.uk, gov.uk, nice.org.uk, mind.org.uk, autism.org.uk, youngminds.org.uk
- Logic Patterns: safety_first, clarify_before_action, small_steps, explain_constraints

9. Test Results Summary

- 56 tests
- 241ms
- Parser tests
- Arbiter tests
- Generator tests
- Memory tests

10. Conclusion

Infinity Safe Agent introduces a deterministic, model-agnostic governance layer for AI behaviour. By separating proposing from governing, and enforcing safety, truth, and logic rules through AML and a runtime arbiter, Infinity provides a predictable, testable, and certifiable foundation for safe AI. This architecture is designed to become the reference standard for AI governance across consumer, enterprise, and regulated domains.

Appendix A — Working Implementation Reference

- Repo Structure (Public version only)
- AML Examples (10 examples)
- Safety Report Summary
- Demo Results
- Safety Configuration
- How to Run the Demo