

[SPEC](#)

OMEGA Protocol Specification

Version 1.0, March 2026

OMEGA is an authorisation layer for autonomous action and commitment infrastructure for machine systems. This document specifies how computation becomes consequence – not how cognition ought to proceed – in human, hybrid, and autonomous deployments.

Making it harder to lie with OMEGA than without it.

OMEGA turns latent probabilistic computation into explicit, governed, falsifiable commitment records at the boundary where action becomes irreversible.

OMEGA does not prove that a governed decision was correct. It proves that a commitment was authorised, that the reasoning was recorded before the action was taken, and that the record cannot be altered after the fact. It makes bad reasoning harder to disguise, easier to detect, and possible to compare against outcomes over time. Auditability is not correctness. It is the precondition for measuring correctness.

About this document

Making it harder to lie with OMEGA than without it. OMEGA is an authorisation layer for autonomous action and commitment infrastructure for machine systems; it turns latent probabilistic computation into explicit, governed,

Version 1.0

Status Published

Date March 2026

Canonical reference omegaprotocol.org/spec/v1

Licence Open. Freely cite with attribution.

[Download full PDF \(v1.0\) →](#)

The PDF is generated from this page plus the long-form narrative chapters (scope, decision cycle, governance spine, and related sections). Use the contents below to jump within the web supplement; the PDF is the complete printable document.

Contents – on this page

1. [Preamble](#) – positioning and auditability
2. [About this document](#)
3. [Primitive irreducibility](#) – Governance, Reasoning, Traceability, Expectation (P4), Confirmation
4. [Evidence](#)
5. [Named failure modes](#) (FM_R)
6. [What OMEGA does not claim](#)
7. [Formal verification](#) – necessity, sufficiency, SymPy script, failure-mode table
8. [Appendix – OMEGA adversarial simulation](#)

Chapters such as Scope, the OMEGA Decision Cycle, Governance Spine, Non-action schema, Assumption handling, Expectation and surprise, Domain packs, Versioning, Falsification, and References appear in the PDF only – they are not duplicated in this HTML view.

Primitive Irreducibility

– who is permitted to act, under what rules, and within what boundaries. Removing Governance loses the answer to the question: was this decision legitimate? A system could produce perfect Reasoning, accurate Expectation, complete Traceability, and verified Confirmation, and still have acted outside its authorised scope. Governance is the primitive that determines whether a decision was made by the right entity under the right conditions. No other primitive captures this. Traceability records that a decision happened. Governance records whether it was authorised to happen.

REASONING

Reasoning captures the real-time process of navigating uncertainty before commitment – the FACT, INFERENCE, and ASSUMPTION chain that connects available information to a conclusion. Removing Reasoning loses the answer to the question: how did the system get from its inputs to its decision? Expectation captures what was predicted. Reasoning captures how that prediction was formed and how the decision followed from it. They are sequential not equivalent. The Reasoning that produced a prior Expectation is not the same as the Reasoning captured in the current decision record. A system could hold accurate Expectations and still reason to the wrong conclusion through flawed inference. Reasoning is the primitive that makes the path from evidence to action inspectable.

TRACEABILITY

Traceability captures the permanent auditable record that exists independently of the governance structure that produced it. Removing Traceability loses the answer to the question: what can be verified after the fact, even if the system that made the decision no longer exists? Governance is ephemeral – authority structures can be dismantled, policies can be changed, organisations can dissolve. Traceability is permanent – the record survives the removal of everything else. This is not a property of Governance extended over time. It is a structurally distinct primitive because it is the only one that cannot be revoked. A cryptographically immutable Traceability

EXPECTATION (P4)

Expectation captures an explicit commitment to a predicted future state before any action is taken. Removing Expectation loses the answer to the question: what did the system believe would happen, and can that belief be falsified? Most autonomous systems act without committing to a prediction. They respond to inputs without recording what they expected those inputs to mean. Expectation forces the system to declare its model of the world before acting – creating a falsifiable object that can be compared against outcomes. This is not contained within Reasoning. Reasoning is the process. Expectation is a specific output of that process – a committed prior that exists as a distinct record before the decision unfolds. Without Expectation, there is no baseline against which to measure whether the system's model of the world was accurate.

A complete Expectation primitive records at minimum the following sub-fields:

- **predicted_outcome** – what the system expects to happen
- **confidence** – calibrated probability or uncertainty range
- **evaluation_horizon** – when the outcome will be measurable
- **measurement_method** – how the outcome will be assessed
- **decision_relevant_loss_function** – what constitutes a bad outcome for this decision

CONFIRMATION

Confirmation captures the moment at which intent becomes irreversible commitment – the point at which possibility collapses into actuality. Removing Confirmation loses the answer to the question: was there a distinct gate between deciding and acting? Every other primitive operates in the space of deliberation. Confirmation is the primitive that marks the boundary between deliberation and execution. It is not Expectation-closure – Expectation is a commitment about the future, Confirmation is the present becoming permanent. The structural analogy

is precise: Expectation is the quantum superposition of possible futures, Confirmation is the measurement that collapses the superposition into a single irreversible outcome. Without Confirmation as a distinct primitive, there is no record of whether the system had a genuine opportunity to not act. The non-action record is only meaningful if Confirmation exists as the gate through which action must pass.

WHY ALL FIVE TOGETHER

Each primitive captures something the other four cannot. Together they form the minimum complete anatomy of a governed passage from deliberation to authorised commitment: who was authorised (Governance), how they reasoned (Reasoning), what they predicted (Expectation), that the record is permanent (Traceability), and that there was a distinct moment of commitment (Confirmation). Remove any one and the record loses a dimension that cannot be recovered from the remaining four.

This is not a design claim – it is an empirical finding validated across financial markets, biological adaptive systems, space governance scenarios, pharmaceutical manufacturing, and AI agent evaluation. The same five structures appear wherever adaptive intelligence navigates uncertainty under constraint.

EVIDENCE

Empirical validation March 22 2026: six AI systems tested against OMEGA primitive structure. DeepSeek R1 produced the richest Reasoning primitive of any model tested but only one of five primitives unprompted. Every system tested produced decision fragments. OMEGA is the complete anatomy. The standard exists because no individual AI system produces it spontaneously.

Named Failure Modes

The gate passed and the confirmation fired, but the reasoning that led to the gate decision was hollow at the point that mattered. Binary gate passed therefore safe is not governed reasoning. It is the absence of Reasoning dressed as its presence. Instantiated empirically: March 22 2026, re-entry governed record, Reasoning scored 4/10 by adversarial audit.

What OMEGA Does Not Claim

- OMEGA does not prove a decision was correct
- OMEGA does not prevent bad decisions
- OMEGA does not replace human judgment
- OMEGA does not guarantee reasoning is causally connected to the decision unless P2 (Reasoning) is bound to actual execution traces
- OMEGA does not model continuous commitment gradients in systems with no discrete confirmation boundary

Formal Verification of Primitive Necessity and Sufficiency

The following constitutes a machine-verified formal proof of primitive necessity and sufficiency, produced using SymPy (Python symbolic mathematics library) on 22 March 2026.

SYMPY VERIFICATION SCRIPT

```
from sympy import symbols
from sympy.logic.boolalg import And, Not
from sympy.logic.inference import satisfiable
```



```
Complete = And(Accountability, Justification, Auditability, Predictability, CommitmentCo
```

```
checks = {}
for prim, name in [(G,'Governance'), (R,'Reasoning'), (T,'Traceability'),
                  (E,'Expectation'), (C,'Confirmation')]:
    no_prim = And(Complete, Not(prim))
    checks[name] = {
        'Complete_implies_primitive (unsat)': not satisfiable(no_prim),
        'Failure_possible_without_it': satisfiable(And(Not(prim), *[p for p in [G,R,T,E,
    }
sat_all_five = satisfiable(And(G, R, T, E, C, Complete))
print(checks)
print('Sufficient (all five implies Complete):', sat_all_five)
```

RESULTS

Every Complete implies primitive check returns True – Complete AND NOT primitive is UNSAT for all five.

Every failure possible without it check returns True – the other four can hold while the missing primitive fails.

Sufficient: True – model {G:True, R:True, T:True, E:True, C:True}.

FAILURE MODES WHEN A PRIMITIVE IS ABSENT

REMOVE	FAILURE MODE
Governance	Illegitimate authority – rogue decisions indistinguishable from valid ones
Reasoning	Opaque decision logic – cannot audit rationality or detect error
Traceability	Mutable history – retroactive rewriting becomes possible
Expectation	No falsifiable prior – hindsight bias uncheckable, learning impossible
Confirmation	No deliberate gate – intent flows directly into irreversible action

With all five primitives present, all failure modes are simulated.

eliminated. Any conceivable sixth property reduces to one of the five categories. The set is proven minimal and complete.

Note: the formal proof establishes structural completeness – that no primitive can be removed without creating an uncompensable failure mode. It does not guarantee the quality of reasoning within each primitive. FM_R (see [Named Failure Modes](#)) is a quality failure that can occur even when all five primitives are structurally present – the gate fires, the record exists, but the reasoning chain is hollow at the point that matters. Structural completeness is necessary but not sufficient for governed correctness.

Formal verification produced by Grok (xAI) via Python REPL, 22 March 2026. Machine-verified.

APPENDIX – ADVERSARIAL SIMULATION

The full Grok-run simulation of ten agent violations and ten detections is published as a standalone appendix: [OMEGA Adversarial Simulation – 10 Agent Violations, 10 Detections](#).



Programmable trust infrastructure for autonomous systems. Built in Bristol. Open-source for humanity.

warrensmith8@gmail.com

PROTOCOLS

Clearpath

Cognitive Ledger

Consent Ledger

Assumption Registry

Ethics Gate

LIVE DEMOS

Space Governance Tool

Agent Arena

OMEGA Build

OMEGA Energy

Decision API

OMEGA Learn