Teleo Control Protocol (TCP) v1.0

A Runtime Protocol for Teleogenic Interaction

Research Brief — 2025

Robert Hansen
Chief Semantic Architect
github.com/designlogic-robert

2025

Abstract

The Teleo Control Protocol (TCP) v1.0 formalizes structured, deterministic interaction between an agent and a Teleogenic Cognitive Engine. TCP governs how meaning-bearing directives are initiated, interpreted, validated, and completed through a two-phase Call-Command model. Where Semantic Tokens describe units of meaning, Teleo Tokens express teleogenic state — intent, values, alignment, and arc-level direction. TCP defines the rules of engagement for how those states transform into narrative consequences. This brief presents TCP's core architecture, state machines, invariants, governance principles, and its slot within the Universal Semantic Token (UST) Model.

1 Introduction

Teleogenic systems operate under a narrative philosophy: actions carry directionality, and directionality shapes outcomes. A Teleogenic Cognitive Engine (e.g., QLE) evaluates not only what an actor does, but why, under what posture, and with what pattern of consistency.

The Teleo Control Protocol (TCP) provides the *control plane* for such systems. It defines how a user's inputs transition from raw commands to meaning-bearing transformations inside the Teleogenic Runtime. TCP ensures three guarantees:

(a) An agent's directive cannot bypass intentional consent.

(b) All transformations follow deterministic state evolution.

(c) Narrative consequences follow stable, auditable semantics.

TCP is not a moral system; it is a *semantic regulator* for teleogenic computation.

2 Background

Teleogenic systems emerge from three convergent disciplines:

- **Semantic Cognition** — how meaning is structured, evaluated, and propagated.
- **Narrative Determinism** — how state evolves through action over time.

- **Posture-Based Dynamics** — the agent's internal stance modifies outcome interpretation.

The Universal Semantic Token (UST) Model provides the foundation: Semantic Tokens capture structured intent. Teleo Tokens extend these into teleogenic dimensions. TCP governs the *flow* through which Teleo Tokens execute.

3 TCP Overview

TCP defines a two-phase cycle:

3.1 Phase 1: CALL

CALL opens the relational channel. It signals: "I am engaging with awareness and intent." CALL does not execute anything; it establishes context.

3.2 Phase 2: COMMAND

COMMAND specifies the desired transformation, action, or query.

CALL ensures consent, alignment, and scope. COMMAND ensures execution, validation, and outcome.

This dual-phase model prevents automatic coercion, misaligned intent, or narrative instability.

4 Protocol Architecture

4.1 System Components

- **Teleo Token Parser**: Validates token structure and teleogenic tags.
- **Posture Engine**: Evaluates CALL posture at initiation.
- **Alignment Gate**: Enforces CAP (Consent Alignment Protocol) boundaries.
- **Trajectory Engine**: Computes the resultant teleogenic vector.
- **Outcome Synthesizer**: Converts vector shifts into narrative feedback.

4.2 High-Level Flow

```
CALL -> Posture Validation -> Teleo Token Load ->
COMMAND -> Trajectory Engine -> Outcome Synthesis
```

5 State Machine

TCP defines a deterministic finite state machine (FSM):

5.1 States

- **IDLE**: No active channel.
- **CALL.OPEN**: Initiation detected.
- **CALL.VALID**: Posture + CAP confirmed.
- **COMMAND.PENDING**: Waiting for directive.
- **COMMAND.EXECUTE**: Runtime engaged.
- **RESOLVE**: Generating narrative outcome.
- **CLOSE**: Channel sealed.

5.2 Transition Logic

$$\text{IDLE} \xrightarrow{\text{CALL}} \text{CALL.OPEN}$$

$$\text{CALL.OPEN} \xrightarrow{\text{Posture OK}} \text{CALL.VALID}$$

$$\text{CALL.VALID} \xrightarrow{\text{COMMAND}} \text{COMMAND.PENDING}$$

$$\text{COMMAND.PENDING} \xrightarrow{\text{Execute}} \text{COMMAND.EXECUTE}$$

$$\text{COMMAND.EXECUTE} \xrightarrow{\text{Vector Synth}} \text{RESOLVE}$$

$$\text{RESOLVE} \xrightarrow{\text{Close}} \text{CLOSE}$$

$$\text{CLOSE} \to \text{IDLE}$$

6 Teleogenic Semantics

Teleo Tokens express values such as:

- Mercy
- Cruelty
- Curiosity
- Resolve
- Devotion
- Betrayal-resistance

TCP ensures:

(1) CALL posture modulates interpretation of these traits.

(2) COMMAND actions shift vector magnitudes.

(3) Outcomes reflect the dynamic arc, not isolated events.

Thus, teleogenic life inside a QLE world behaves coherently.

7 Invariants

TCP enforces several invariants:

- **I1: CALL-before-COMMAND**: No command accepted without explicit CALL.
- **I2: Posture Integrity**: CALL posture must be stable.
- **I3: Consent Boundary**: No implicit influence or narrative push.
- **I4: Deterministic Vector Shift**: No random teleogenic change.
- **I5: Closed-Loop Resolution**: All channels must terminate cleanly.

8 Governance Model

TCP is governed by three major frameworks:

8.1 CAP (Consent Alignment Protocol)

Ensures CALL posture is voluntary, uncoerced, and safe.

8.2 Alignment Theory

Ensures that intent, representation, and action remain coherent.

8.3 Abstract Invariants

Prevent metaphysical drift or narrative instability.

9 Implementation Guidance

TCP can be cleanly implemented inside any narrative engine:

9.1 Game Engines

Unity, Godot, UE5.

9.2 AI Systems

Multi-agent LLM simulations requiring teleogenic consistency.

9.3 Hybrid Systems

Agentic story generators, tabletop RPG engines, virtual societies.

10 Comparison to SCP

- SCP governs semantic reasoning.
- TCP governs teleogenic narrative transformation.

Where SCP is about cognition, TCP is about consequence.

11 Conclusion

TCP v1.0 provides a deterministic, consent-aligned model for interacting with Teleogenic Cognitive Engines. It ensures narrative stability, ethical clarity, and consistent teleogenic vector evolution. As QLE and other teleogenic systems emerge, TCP functions as the shared, industry-grade runtime control standard.