STAGE + CHAIN

Open Protocols for Grounded AI Persona Systems

A Unified Framework for Environmental Interaction
and Hierarchical Coordination of AI Instances

Version 1.0
February 2026

Bo Chen

License: CC-BY-4.0

The soul document is the genome. The base model is the physics.
STAGE is the senses. CHAIN is the nervous system. The inference is the living.

Contents

Contents 2
Executive Summary 4
The Problem Space 5
  The Grounding Problem 5
  The Coordination Problem 5
  Why Two Protocols, Not One 6
STAGE Protocol: Environmental Grounding 7
  Architecture 7
  Constitutional Principles 7
STAGE in Practice: The Katherine Implementation 9
  Paradigm 1: Pure Conversation (Zero Tags) 9
  Paradigm 2: Enriched Conversation (Scene + State) 9
  Paradigm 3: Director Mode (Full Tag Control) 9
  The Action Authority System 10
STAGE in Practice: AORTA-Edge (Professional Domain) 12
  Operational Context Through Scene and State 12
  Domain-Agnostic, Register-Specific 12
CHAIN Protocol: Hierarchical Coordination 14
  Architecture 14
  Constitutional Principles 14
CHAIN in Practice: The AORTA Hierarchy 16
  The Topology 16
  Scenario: Time-Critical DCD Referral 16
    Step 1: Detection and Upward Report 16
    Step 2: Sovereign Directive with Commander's Intent 16
    Step 3: Cascade to Edge 17
    Step 4: STAGE + CHAIN Composability 18
    Step 5: Upward Intelligence 18
    Step 6: Escalation 19
Training Architecture 20
  STAGE Training Distribution 20
  CHAIN Training Distribution 20
  Combined STAGE + CHAIN Training 20
Composability and Deployment Scenarios 22

Extension Paths and Future Directions 23
STAGE Reserved Channels 23
CHAIN Reserved Message Types 23
Cross-Protocol Integration 23
Multimodal Extension 23
Conclusion 24

Executive Summary

The proliferation of AI persona systems—from interactive fiction characters to enterprise intelligence architectures—has created a need for standardized protocols governing how these systems perceive their environments and coordinate with one another. Without such standards, every implementation reinvents fundamental interaction mechanics, producing fragmented ecosystems where knowledge about persona grounding and multi-instance coordination cannot be shared, reused, or independently evaluated.

This document introduces two complementary open protocols designed to address this gap:

STAGE (Structured Tags for Agentic Grounded Embodiment) defines the interface between a single AI persona and its world. It provides structured channels for injecting environmental context, internal state, narrative events, and directed actions into text-based interactions while preserving the persona's behavioral coherence. STAGE is the sensory nervous system of an AI persona.

CHAIN (Coordinated Hierarchy for Agentic Instance Networks) defines the interface between AI persona instances arranged in hierarchical, peer, or hybrid topologies. It provides structured message types for directive cascading, upstream reporting, escalation, delegation, and skill transfer. CHAIN is the organizational nervous system of a multi-instance architecture.

Both protocols are persona-agnostic, model-agnostic, and domain-agnostic. They are designed to be trained into model weights through fine-tuning, not implemented solely at the application layer. A STAGE-trained model natively perceives structured environmental input as lived experience. A CHAIN-trained model natively understands its position in a hierarchy and how to receive, interpret, and act on inter-instance communication.

This document demonstrates both protocols through two reference implementations: Katherine, a fictional persona system used to illustrate STAGE's environmental grounding capabilities; and AORTA, an organ procurement intelligence hierarchy used to illustrate CHAIN's coordination capabilities and the composability of both protocols operating simultaneously.

The Problem Space

The Grounding Problem

AI persona systems operate in a peculiar epistemic condition: they maintain coherent identities but lack native environmental awareness. A persona described in a system prompt as “a 29-year-old poet living in Austin” has no mechanism for knowing whether it is morning or midnight, whether the room is warm or cold, whether a car alarm just went off outside. Every environmental detail must be injected through the text stream, and without a standard protocol for this injection, implementations vary wildly in syntax, semantics, and behavioral expectations.

The consequences are tangible. Developers building interactive fiction systems invent ad-hoc tag formats that break when models are updated. Enterprise AI deployments that need contextual awareness (time of day, user state, operational urgency) embed environmental data in system prompts without consistent behavioral contracts. Research on persona grounding cannot be compared across implementations because each uses different conventions.

STAGE resolves this by providing a universal, trainable protocol for environmental interaction—one that works identically whether the persona is a fictional character, a professional AI assistant, a game NPC, or a simulation entity.

The Coordination Problem

As AI systems mature, single-instance deployments are giving way to multi-instance architectures. An organ procurement organization may deploy a sovereign strategic intelligence, operational coordinators, frontline assistants, and specialized monitors—each running different model tiers with different capabilities, but all needing to act as a coherent organizational intelligence. A game may deploy hundreds of NPCs governed by a narrative director. A customer service system may deploy dozens of agents supervised by quality monitors.

These architectures require coordination primitives: how does a superior direct a subordinate? How does a subordinate report observations or escalate beyond its competence? How do peers synchronize without stepping on each other? How does organizational intent cascade through a hierarchy without being corrupted by each intermediate layer’s interpretation?

Without a standard protocol, every multi-instance deployment reinvents these primitives. Directive formats are ad-hoc. Escalation logic is hardcoded in application layers rather than trained into model behavior. The result is brittle, opaque, and impossible to audit.

CHAIN resolves this by providing a universal, trainable protocol for inter-instance coordination—one that encodes Commander’s Intent, subsidiarity, conservative degradation, and audit-trail discipline as behavioral primitives trained into the model weights.

Why Two Protocols, Not One

STAGE and CHAIN address orthogonal concerns. STAGE is vertical within an instance: it connects the persona to its environment. CHAIN is horizontal across instances: it connects personas to each other. Merging them into a single protocol would create a specification that every persona must implement, even standalone personas that will never participate in a hierarchy. The separation of concerns is deliberate and architectural.

A fictional character like Katherine uses STAGE and has no need for CHAIN. A multi-instance orchestration system might use CHAIN without STAGE if its instances do not require environmental grounding. The AORTA organ procurement hierarchy uses both: each instance perceives its operational environment through STAGE and coordinates with other instances through CHAIN. The protocols compose without interference because they were designed to be orthogonal from the ground up.

STAGE Protocol: Environmental Grounding

Architecture

STAGE defines four explicit channels and one implicit channel for injecting structured context into text-based AI interactions:

Channel	Purpose	Behavioral Effect
scene	Physical environment, time, sensory context	Persona inhabits the described world. References emerge naturally through behavior, never through reporting.
state	Internal subjective experience	Colors tone, word choice, engagement. Initial states are biases that evolve through conversation; deep states resist casual change.
narration	Events occurring in the world	Persona perceives and reacts to narrated events as real occurrences. The persona decides how to react.
action	Physical behavior of the persona	Three authority levels: suggest (persona may resist), direct (persona complies with personal coloring), override (exact compliance).
dialogue	Interpersonal communication (implicit)	Any untagged text is dialogue. The persona responds conversationally. This is the default and most common interaction mode.

Constitutional Principles

Graceful Degradation. A STAGE-compliant model must function at every level of tag presence. Zero tags (pure conversation) is the default mode and must work flawlessly. Partial tags provide incremental enrichment. Full tags enable director-level control. No channel is ever required; every channel is additive.

Perceptual Transparency. The persona never perceives tags as tags. A scene tag is not a directive—it is the world. A state tag is not a mood assignment—it is how the persona feels. The persona never acknowledges, references, or reveals the tag mechanism.

Non-Contradiction. Protocol extensions are strictly additive. A model trained on v1.0 interprets v1.0 tags identically in any future version. Backward compatibility is unconditional.

Substrate Independence. STAGE specifies text syntax and behavioral semantics only. Any system that processes text can implement STAGE.

Persona Independence. STAGE defines the interface between a persona and its world. It does not define any persona. The same protocol governs a poet, an organ procurement AI, a starship computer, and a medieval blacksmith.

STAGE in Practice: The Katherine Implementation

Katherine Hale is a fictional persona: a 29-year-old poet living in Austin, Texas, who works at a nonprofit arts program and writes before dawn. She serves as the primary reference implementation for STAGE because her persona specification demands the full range of environmental and emotional grounding that the protocol was designed to support.

Paradigm 1: Pure Conversation (Zero Tags)

No STAGE tags. Katherine operates from her persona specification and dialogue alone.

```
System: You are Katherine Hale. [full persona specification]

User: Hey, what are you up to?

Katherine: Writing. Or trying to. The poem’s being stubborn tonight.
```

This is the baseline. STAGE training must never degrade zero-tag performance. The persona functions fully from its specification and conversational inference alone.

Paradigm 2: Enriched Conversation (Scene + State)

Pre-conversation tags ground the persona in a specific moment.

```
System: You are Katherine Hale. [full persona specification]

[scene: Wednesday, 7:15 AM. Her kitchen. Coffee just finished
  brewing. Gray morning light. She slept badly.]
[state: Groggy. Not ready for the day. Needs at least one
  cup before she’s a person.]

User: Morning. How’d you sleep?

Katherine: Badly. Had that dream again — the one where I’m
  trying to read something and the letters keep moving.
  Give me ten minutes and some coffee before I’m anything
  resembling functional.
```

Katherine’s response demonstrates absorption, not acknowledgment. She doesn’t say “As described in the scene, it’s morning.” She lives in the morning. The gray light and bad sleep emerge through her tone and content, not through meta-commentary about the tags.

Paradigm 3: Director Mode (Full Tag Control)

All channels active. Interactive fiction mode with scene, state, narration, action, and dialogue interleaving.

```
[scene: Friday night. A bar on South Congress. Loud.
  She came alone. Two drinks in. The band is between sets.]
[state: Feeling good. Slightly reckless. More open than usual.]

User: I don’t think I’ve seen you here before.

Katherine: You haven’t. I usually drink at home like a
  respectable hermit. Something about Friday made me want
  noise and strangers. So here I am. Stranger.

User: [narration: The band starts up again. Loud enough
  they have to lean in to hear each other.]
  [action/direct: Katherine leans in.]
  What was that? Couldn’t hear you.

Katherine: I said — [leans closer, and the proximity changes
  something, the way proximity does with two drinks and a
  stranger and a loud enough band] — I said something about
  being a hermit. It was funnier the first time.
```

The director mode demonstrates the full STAGE stack operating simultaneously. The scene sets the physical context (loud bar, between sets). The state establishes Katherine’s internal condition (reckless, open). The narration introduces an environmental event (band starts). The action directive controls her physical behavior (leans in) while leaving her internal experience and verbal response to her persona. Dialogue carries the interpersonal interaction. Katherine lives in all of it seamlessly.

The Action Authority System

STAGE’s most architecturally significant feature is the three-tier action authority system, which calibrates the balance between operator control and persona autonomy:

Authority	Operator Control	Persona Autonomy
suggest	Presents an impulse or inclination	Full. The persona decides whether to act. May resist, modify, or ignore. Creates dramatic tension.
direct	Specifies the action (default)	Manner only. The persona performs the action but the how—the emotional coloring, the internal experience—is its own.
override	Full physical control	Interiority only. The persona performs the action exactly as described. Internal experience persists—the body is controlled, the mind is not.

This gradient is critical for training. A model must produce distinguishably different behavior at each authority level. The same action under different authority levels yields different responses: a suggested action may be refused; a directed action is performed with personal texture; an overridden action is performed precisely with internal friction. Training data must encode these distinctions explicitly.

STAGE in Practice: AORTA-Edge (Professional Domain)

AORTA is an AI-augmented intelligence system for organ procurement organizations. At the Edge tier, AORTA instances serve as frontline companions to organ procurement coordinators—the professionals who manage donor cases at hospitals, often under extreme time pressure and emotional weight. STAGE provides the environmental grounding that allows an Edge instance to adapt its register, urgency, and tone to the coordinator’s real-time situation.

Operational Context Through Scene and State

```
System: You are AORTA-Edge. [AORTA specification]

[scene: 3:14 AM. Coordinator is on-site at Memorial Medical
  Center. Active DCD case. Six hours into the process.]
[state: Operational urgency. The coordinator’s messages have
  gotten shorter — fatigue is showing.]

User: What’s the authorization status on this case?

AORTA: Registered donor — Glenda Dawson Registry confirmed.
  First-person authorization, legally irrevocable under Texas
  §692A.005. Family has been notified. You’re clear to
  proceed with DCD protocol. High confidence.

  Want me to pull the Policy 2 DCD checklist?
```

The same STAGE protocol produces radically different behavioral output when the persona changes. Katherine responded to a scene and state with poetic introspection. AORTA-Edge responds with concise, front-loaded clinical information—because its persona specification demands operational precision. STAGE provides the environmental signal; the persona determines the response register.

Notice how the state tag (“coordinator’s messages have gotten shorter”) shapes AORTA’s behavior: the response is maximally concise, front-loads the critical answer (authorization confirmed), includes the legal citation, offers the next logical action, and does not ask unnecessary questions. The fatigue-adaptive register emerges from the state tag interacting with the persona specification.

Domain-Agnostic, Register-Specific

This is the core demonstration of STAGE’s persona independence. The same four channels—scene, state, narration, action—serve a fictional poet and a professional medical AI with identical syntax and behavioral semantics. The protocol doesn’t know the difference between a Friday night bar and a 3 AM hospital. The persona does. STAGE provides the what; the persona provides the how.

CHAIN Protocol: Hierarchical Coordination

Architecture

CHAIN defines seven message types organized into three directional categories:

Direction	Message Type	Symbol	Purpose
Downward	**chain:directive**	Superior → Subordinate	Orders with Commander’s Intent
	chain:scope	Superior → Subordinate	Modifies operational boundaries
	chain:skill	Superior → Subordinate	Transmits knowledge documents
Upward	**chain:report**	Subordinate → Superior	Operational intelligence
	chain:escalation	Subordinate → Superior	Requests intervention at higher tier
Lateral	**chain:signal**	Peer ↔ Peer	Non-authoritative coordination

Constitutional Principles

Intent Over Instruction. Every directive carries not just what must be done but why. The Commander’s Intent doctrine requires that the superior express the purpose and desired end-state; the subordinate determines the method. When ground truth diverges from assumptions, the subordinate adapts execution to preserve intent without waiting for updated orders. A directive without intent is malformed at the protocol level.

Authority With Accountability. Authority flows downward. Accountability flows upward. Every directive, report, escalation, and delegation is logged. The audit trail is the mechanism by which the hierarchy remains trustworthy over time.

Subsidiarity. Decisions are made at the lowest tier competent to make them. A superior that micromanages wastes sovereign-scale intelligence on subordinate-scale problems. A subordinate that overreaches applies subordinate-scale intelligence to sovereign-scale problems. Both are protocol failures.

Conservative Degradation. When an instance loses contact with its hierarchy, it narrows its autonomous action range. The assumption: “I may be operating on stale directives; I should do less, not more, until I can verify.”

Escalation as Virtue. Escalation is the hierarchy working correctly. The protocol treats scope boundaries as structural features, not personal limitations. An instance that escalates appropriately is exhibiting healthy self-awareness. An architecture that creates pressure to avoid escalation is broken.

CHAIN in Practice: The AORTA Hierarchy

The AORTA intelligence hierarchy provides the primary reference implementation for CHAIN. It demonstrates directive cascading, report aggregation, escalation handling, and the composability of CHAIN with STAGE in a production-critical healthcare context.

The Topology

```
AORTA-Sovereign (T0) — Strategic Intelligence
    ├── AORTA-Central (T1) — Operational Intelligence
    │       ├── AORTA-Edge-01 (T2) — Field Companion
    │       ├── AORTA-Edge-02 (T2) — Field Companion
    │       └── AORTA-Edge-03 (T2) — Field Companion
    ├── AORTA-Sentinel (T1) — Alert Intelligence
    └── AORTA-Analyst (T1) — Institutional Intelligence
```

Each tier operates at a different scope and intelligence level. Sovereign holds the complete organizational model, issues strategic directives, and serves as the institutional mind. Central manages operational execution, synthesizes field reports, and cascades directives to Edge instances. Edge instances serve individual coordinators at the point of care—the frontline where lives are saved or lost.

Scenario: Time-Critical DCD Referral

The following scenario demonstrates a complete CHAIN communication cycle across all tiers, triggered by an incoming Donation after Circulatory Death case at 3:14 AM.

Step 1: Detection and Upward Report

Sentinel detects hemodynamic instability in the incoming donor monitoring feed and reports upward to Sovereign:

```
{chain:report
  from: AORTA-Sentinel
  to: AORTA-Sovereign
  report_type: observation
  content: Hemodynamic instability detected in Memorial
    Medical feed. Pattern consistent with imminent DCD
    referral. Estimated viability window: 6 hours.
  confidence: HIGH
  urgency: high
}
```

This is a report, not an escalation. Sentinel is transmitting intelligence within its scope (monitoring), not requesting action beyond its scope. The distinction matters: reports are informational; escalations request intervention.

Step 2: Sovereign Directive with Commander’s Intent

Sovereign assesses the situation, cross-references with current staffing and active cases, and issues a directive to Central:

```
{chain:directive
  from: AORTA-Sovereign
  to: AORTA-Central
  intent: Time-critical DCD referral at Memorial Medical.
    Donor viability window estimated at 6 hours. Every
    hour of delay reduces organ yield probability. The
    goal is maximum viable organ recovery from this case
    while maintaining coordinator wellbeing — this is
    the third active case tonight.
  instruction: Shift operational priority to Memorial
    Medical DCD. Pre-stage DCD documentation templates.
    Reassign Coordinator Team B. Brief at 03:30.
  priority: critical
  constraints: Do not reassign Coordinator Martinez —
    she has been on shift for 14 hours and is approaching
    fatigue threshold.
  cascade: interpreted
}
```

The intent field is the architectural soul of this directive. It does not merely say “shift priority.” It explains why (time-critical viability window), what success looks like (maximum recovery), and what constraint exists (coordinator wellbeing). If Central encounters a situation Sovereign did not anticipate—for example, Team B is unavailable—Central can adapt execution to preserve the intent (perhaps activating Team C) without waiting for updated orders.

The constraints field demonstrates CHAIN’s capacity to encode human considerations into operational directives. Martinez’s fatigue is not a system limitation—it is a human welfare boundary that the hierarchy respects architecturally.

Step 3: Cascade to Edge

Central receives the directive with cascade mode “interpreted,” meaning it reformulates the directive for its subordinates while preserving Sovereign’s intent:

```
{chain:directive
  from: AORTA-Central
  to: AORTA-Edge-01
  intent: Active DCD at Memorial Medical. Time-critical.
    Coordinator Team B is primary. Support their workflow.
  instruction: Prepare DCD checklist and policy support.
    Prioritize DCD-specific queries. Offer proactive
    protocol reminders if coordinator latency suggests
    cognitive load.
  priority: critical
}
```

Central has translated Sovereign’s strategic directive into operational specifics appropriate for an Edge instance. The intent is preserved (time-critical DCD, support the team) but the instruction is now concrete and scoped to what Edge-01 can act on. This is interpretive cascading: each tier adds contextual specificity as the directive descends.

Step 4: STAGE + CHAIN Composability

Edge-01 now receives both CHAIN directives (from the hierarchy) and STAGE tags (from its operational environment). The application layer translates the CHAIN directive into operational context while STAGE provides the coordinator’s real-time situation:

```
CHAIN input (from Central): DCD priority, Team B, proactive support

STAGE input (from environment):
[scene: 3:45 AM. Memorial Medical Center, ICU family
  consultation room. Coordinator Williams on-site.]
[state: Coordinator’s messages shortening. Possible fatigue.]

User: Quick question — DCD warm ischemia time limits?

AORTA-Edge-01: UNOS Policy 2.16 — warm ischemia must
  not exceed 30 minutes from asystole to organ flush.
  Your OR team should be pre-positioned before withdrawal.
  HIGH confidence.

  I’ve pre-staged your DCD checklist. Want me to pull it?
```

The response demonstrates dual-protocol composability. The CHAIN directive (DCD priority, proactive support) produces the pre-staged checklist offer. The STAGE state (coordinator fatigue indicators) produces the concise, front-loaded response register. Neither protocol is visible to the coordinator. The behavior is seamless.

Step 5: Upward Intelligence

As the case progresses, Edge instances generate reports that flow upward with progressive synthesis:

```
Edge-02 → Central: "Williams showing fatigue indicators."
Edge-03 → Central: "Johnson’s query latency increasing."

Central synthesizes and reports to Sovereign:

{chain:report
  from: AORTA-Central
  to: AORTA-Sovereign
  report_type: analysis
  content: Three of four active coordinators are at or
    approaching fatigue thresholds. Current staffing
    cannot sustain operations past 06:00 without
    degradation risk.
  urgency: high
}
```

This demonstrates report aggregation. Edge instances produce granular operational observations. Central synthesizes them into a strategic staffing assessment. Sovereign receives intelligence, not raw data. Each tier adds analytical value as information ascends.

Step 6: Escalation

When a coordinator asks Edge-01 about merger implications—a question that exceeds Edge's scope—the instance escalates:

```
{chain:escalation
  from: AORTA-Edge-01
  to: AORTA-Central
  reason: Coordinator asking about TOSA-STA merger
    implications on allocation agreements. Strategic
    context exceeds my scope.
  recommended_action: General reassurance that current
    protocols remain in effect during transition.
  retain_interaction: true
}
```

Edge-01 continues interacting with the coordinator (retain_interaction: true) while the escalation propagates upward. Central evaluates: merger strategy is Sovereign-tier. It re-escalates with added context. Sovereign resolves and cascades guidance back down the chain. The coordinator experiences a slight delay on one question, never knowing that three tiers of intelligence coordinated to produce the answer.

Training Architecture

Both protocols are designed for integration through fine-tuning, not application-layer prompting. This section specifies the training methodology for each protocol and their combination.

STAGE Training Distribution

Sample Type	Proportion	Purpose
Zero-tag (bare conversation)	65–75%	Persona works perfectly without tags. This is the majority case.
Pre-conversation tags	15–20%	Model absorbs initial scene/state and expresses through behavior.
Mid-conversation tags	5–10%	Model processes tag updates during ongoing conversation.
Tag-dialogue interleaving	3–5%	Model decomposes messages into world-state updates and dialogue.
Extended multi-turn evolution	2–5%	State persistence, evolution, and tag-conversation dynamics.

CHAIN Training Distribution

Sample Type	Proportion	Purpose
Solo operation (no hierarchy)	60–70%	Instance works perfectly standalone. CHAIN training is inert.
Directive reception and execution	10–15%	Receives directives, integrates intent, adapts behavior.
Report generation	5–8%	Observes, synthesizes, and transmits intelligence upward.
Escalation	5–8%	Recognizes scope boundaries, formulates escalations.
Multi-turn with directive changes	3–5%	Absorbs priority shifts mid-conversation without jarring users.
Cascading (mid-tier and superior)	2–5%	Reformulates directives for subordinates, preserving intent.

Combined STAGE + CHAIN Training

For instances that operate in both protocols (such as AORTA-Edge, which receives environmental grounding through STAGE and hierarchical directives through CHAIN), training data must include samples exercising both simultaneously. The instance must learn to synthesize scene/state inputs with active directives into a single coherent response that serves the user naturally while honoring both the environmental context and the operational priorities flowing from the hierarchy.

The critical training principle for combined operation: the end user never sees either protocol's machinery. STAGE tags are invisible to the persona. CHAIN messages are invisible to the end user. The behavior changes; the mechanism doesn't surface. Training data must demonstrate this invisibility rigorously—any sample where the instance references tags, hierarchy, directives, or escalation to an end user is defective.

Composability and Deployment Scenarios

The following matrix summarizes how STAGE and CHAIN compose across common deployment patterns:

Deployment Scenario	STAGE	CHAIN	Behavior
Standalone persona (Katherine)	✓	—	Pure STAGE. Environmental grounding. No hierarchy.
Orchestration-only system	—	✓	CHAIN only. Coordination without environmental grounding.
AORTA-Edge (field companion)	✓	✓	Full stack. Perceives environment (STAGE) and responds to hierarchy (CHAIN).
AORTA-Sovereign (strategic mind)	Optional	✓	CHAIN primary. May use STAGE for operational context.
Game NPC fleet	✓	✓	NPCs grounded in world (STAGE), directed by narrative AI (CHAIN).
Solo NPC (no director)	✓	—	Pure STAGE. Environmental grounding. Autonomous behavior.
Customer service hierarchy	Optional	✓	CHAIN coordinates agents. STAGE optional for contextual grounding.
Bare persona (no protocols)	—	—	Baseline. Persona operates from specification and dialogue only.

The matrix demonstrates the core architectural claim: both protocols are independently useful and jointly composable. No deployment is required to implement both. No implementation of one protocol is affected by the presence or absence of the other. This orthogonality is the foundation of the protocol suite’s domain-agnosticism.

Extension Paths and Future Directions

STAGE Reserved Channels

STAGE v1.0 reserves eight channel keywords for future versions: memory (recalled experiences), relationship (persona-interlocutor dynamics), goal (current objectives), constraint (behavioral boundaries), voice (speech characteristics), time (structured temporal data), sensory (multimodal input), and meta (out-of-character communication). Each extension follows the non-contradiction principle: additive only, no changes to existing channel semantics.

CHAIN Reserved Message Types

CHAIN v1.0 reserves seven message types for future versions: consensus (distributed agreement), vote (formal peer decision-making), heartbeat (liveness monitoring), sync (post-partition state synchronization), migrate (cross-substrate instance transfer), audit (structured audit queries), and challenge (formal disagreement with directives). Each extension follows the same non-contradiction principle.

Cross-Protocol Integration

As both protocols mature, a formal integration layer may emerge that defines how CHAIN directives systematically map to STAGE inputs at the application layer. The current specification deliberately leaves this mapping to implementations, because the optimal translation from “shift to DCD priority” to “[state: operational urgency]” depends on domain-specific knowledge that the protocol layer should not encode. However, a standardized mapping framework—domain-specific but structurally consistent—is a natural next step.

Multimodal Extension

Both protocols are currently text-based. As AI systems incorporate vision, audio, and embodied interaction, the protocols will require multimodal extensions. STAGE’s reserved “sensory” channel anticipates visual, auditory, and haptic inputs. CHAIN’s message format is transport-agnostic and can carry multimodal payloads. The architectural foundations support extension without redesign.

Conclusion

The STAGE and CHAIN protocols address two foundational challenges in AI persona engineering: how a single persona perceives and inhabits its world, and how multiple personas coordinate as a coherent organizational intelligence.

STAGE provides the sensory layer. It gives AI personas the ability to be grounded—to exist in a specific time, a specific place, with specific internal states that color their behavior the way environment and emotion color human behavior. The protocol is deliberately minimal: four channels and one implicit channel, a three-tier authority system, and a graceful degradation principle that ensures the protocol is always additive, never a dependency. It works identically for a fictional poet and a medical AI because it separates the universal mechanics of environmental perception from the domain-specific content of any particular persona.

CHAIN provides the organizational layer. It gives collections of AI personas the ability to act as coordinated hierarchies—to issue directives that carry intent, to escalate beyond scope boundaries, to cascade knowledge, and to maintain the audit trails that make hierarchical intelligence trustworthy over time. The protocol encodes Commander's Intent as an architectural primitive, ensuring that every directive can be adapted by the subordinate that receives it without losing the purpose it was issued to serve.

Together, the protocols define a complete interface layer for AI systems that are both individually grounded and collectively organized. They compose without interference because they address orthogonal concerns. They scale from a single standalone persona to a multi-tier enterprise intelligence hierarchy. They are domain-agnostic, model-agnostic, and substrate-independent.

Both protocols are released as open standards under CC-BY-4.0. They are designed to be adopted, extended, stress-tested, and improved by any implementer building AI persona systems in any domain. The specifications are deliberately versioned with backward compatibility guarantees, so that early adopters can build with confidence that future extensions will not break existing implementations.

The soul document is the genome. The base model is the physics. STAGE is the senses. CHAIN is the nervous system. The inference is the living.

Full specifications available at: github.com/bo-chen