Human Expression Gateway (HEG)

A Deterministic, Multimodal Interface for Structured Cognitive Action

Date: June 2026

Version: 1.0

WHITE PAPER

Table of Contents

Executive Summary

The **Human Expression Gateway (HEG)** is a deterministic, stateless, multimodal interpreter. Its sole function is to convert raw human expression into structured **Cognitive Action Packets (CAPs)**. HEG is not a model, not probabilistic, not generative, and not a reasoning engine. It does not learn, adapt, infer, predict, or maintain context. It is a fixed pipeline of operators that always run in the same order and always produce the same output for the same input.

HEG addresses a critical challenge in cognitive system design: the reliable, auditable, and unambiguous translation of human expression — across multiple modalities — into a typed, structured format that a downstream **Reasoning Substrate (RS)** can consume safely and consistently.

1. Introduction — The Problem HEG Solves

Human expression is rich, multimodal, and often ambiguous. Text alone is insufficient to capture the full communicative intent of a person. Prosody, timing, gesture, gaze, posture, hesitation, and emphasis all carry meaning. Systems that process human input without accounting for these modalities risk misinterpretation, unsafe inference, and context bleed.

Traditional approaches often bundle interpretation and reasoning into a single, opaque pipeline. This conflation creates auditability problems, makes debugging difficult, and opens the door to hallucination and ambiguous output.

HEG is designed to solve this by functioning as a strict, deterministic front-end interpreter — responsible solely for converting raw expression into a structured, typed envelope before any reasoning occurs.

---

2. What HEG Is

HEG is a **deterministic, stateless, multimodal interpreter**. Its defining properties are as follows:

- **Deterministic:** the same input always produces the same output.
- **Stateless:** it does not remember previous inputs, maintain conversation history, learn, adapt, infer, or predict. Every input is processed fresh.
- **Fixed pipeline:** operators always run in the same order.
- **Not a model, not probabilistic, not generative, and not a reasoning engine.**
- It does not learn, adapt, infer, predict, or maintain context.

HEG's sole function is to convert raw human expression into structured **Cognitive Action Packets (CAPs)**. CAPs are the only output HEG produces.

---

3. What HEG Accepts — Multimodal Input

HEG accepts multimodal human expression across a wide range of signal types. All modalities are normalized into the same CAP schema — meaning that regardless of the modality or combination of modalities present in a given input, the output structure is always consistent and typed.

Modality	Description
Text	Written or transcribed language input.
Audio	Spoken language and acoustic signals.
Video	Visual recording of the expression source.
Gesture	Physical hand, arm, or body movement signals.
Posture	Body orientation and stance signals.
Prosody	Rhythm, stress, and intonation patterns in speech.
Timing	Temporal patterns and sequencing of expression.
Emphasis	Stress or weight placed on particular elements.
Hesitation	Pauses, false starts, and disfluency markers.
Gaze	Eye direction and attention signals.

> **Architectural Note**
>
> This multimodal intake is a core architectural strength. It ensures that no communicatively meaningful signal is discarded simply because it does not arrive as text.

---

4. How HEG Operates — The Operator Pipeline

HEG runs a fixed pipeline of **six operators** in strict sequential order. The pipeline cannot be reordered, skipped, or modified at runtime. Each operator receives the output of the prior operator and contributes to the final CAP.

Operator	Name	Function
1	**SEGMENT**	Divides raw expression into minimal semantic units. Handles nested intent, reversals, topic shifts, contradictions, and multimodal timing cues.
2	**ACT**	Classifies each segment's speech-act type. Defined act types: **ASK**, **COMMAND**, **ASSERT**, and **META**.
3	**TARGET**	Identifies the semantic target of the segment. Defined target types include: **PLAN**, **GOAL**, **CONSTRAINT**, **HYPOTHESIS**, **MODE**, and others.
4	**EMOTION**	Extracts valence and arousal from multimodal cues. Emotion modulates CAP priority and RS interpretation.
5	**PATTERN**	Detects rhetorical or argument structures such as analogy, contrast, or escalation.
6	**CAP BUILD**	Packages all operator outputs into a **Cognitive Action Packet (CAP)**. CAPs are the only output HEG produces.

---

5. The Cognitive Action Packet (CAP) — The Interface Contract

A **Cognitive Action Packet (CAP)** is a typed, structured envelope. It is the only output HEG produces, and it serves as the interface contract between HEG and the Reasoning Substrate (RS). Because the CAP schema is typed and fixed, the RS always receives input in a known, consistent format. This eliminates ambiguity at the HEG–RS boundary.

A CAP contains the following fields:

Field	Type / Structure	Description
act	Enum	The speech-act classification: **ASK**, **COMMAND**, **ASSERT**, or **META**.
targets[]	Array	An array of semantic targets: **PLAN**, **GOAL**, **CONSTRAINT**, **HYPOTHESIS**, **MODE**, etc.
emotion	Object	Contains **valence** and **arousal** values extracted from multimodal cues.
pattern	String	The detected rhetorical or argument structure.
payload	Content	The content of the segment.
metadata	Object	Timestamps, modality identifiers, and IDs.
confidence	Structural value	A structural confidence value. Not probabilistic.

---

6. HEG's Stateless Architecture

HEG does not remember previous inputs, maintain conversation history, learn, adapt, infer, or predict. Every input is processed fresh.

This **statelessness** is a deliberate architectural property, not a limitation. It ensures that:

- HEG's behavior is fully predictable and reproducible.
- No input can be misinterpreted due to contamination from prior inputs.

- Replay of any input always produces the same CAP output.
- Auditability is preserved across all sessions.

---

7. Auditability and Deterministic Replay

Every operator in the HEG pipeline logs its input, output, timestamp, and operator name. This enables three core capabilities:

- **Deterministic replay:** any prior input can be reprocessed and will produce the same CAP output.
- **Debugging:** operator-level logs allow precise identification of where and how a CAP was constructed.
- **Safety auditing:** the full operator trace is available for review.

Replay can be performed from raw input or from stored CAPs. Both replay paths must produce identical RS state. This **bidirectional replay guarantee** is a foundational property of HEG's auditability architecture.

---

8. The HEG–RS Separation

HEG is the front-end interpreter. The **Reasoning Substrate (RS)** is the reasoning engine. They are strictly separated.

HEG's role ends at CAP production. It does not participate in reasoning, planning, decision-making, or response generation. These are the responsibilities of the RS.

This strict separation enforces a typed, deterministic interface between human expression and downstream reasoning. It prevents:

Risk	How the HEG–RS Separation Prevents It
Hallucination	HEG cannot fabricate or infer content beyond what the operators produce.
Unsafe inference	No reasoning occurs inside HEG.
Context bleed	HEG's stateless design means no prior input can

Risk	How the HEG–RS Separation Prevents It
	contaminate current output.
Ambiguous interpretation	The typed CAP schema eliminates structural ambiguity at the boundary.

---

9. Why HEG Matters

HEG addresses several critical needs in the design of systems that process human expression:

1. **Determinism and reproducibility:** Because HEG always produces the same output for the same input, its behavior can be verified, tested, and audited with certainty.
2. **Multimodal completeness:** By accepting text, audio, video, gesture, posture, prosody, timing, emphasis, hesitation, and gaze — all normalized into the same schema — HEG captures the full communicative bandwidth of human expression.
3. **Typed interface contract:** The CAP schema provides a fixed, typed boundary between human expression and the RS. This eliminates the ambiguity that arises when interpretation and reasoning are conflated.
4. **Auditability:** Operator-level logging and deterministic replay enable full traceability from raw input to CAP, and from stored CAP to RS state.
5. **Separation of concerns:** By strictly separating interpretation (HEG) from reasoning (RS), the architecture isolates failure modes, simplifies debugging, and prevents the propagation of interpretive errors into reasoning.

---

10. Summary

The Human Expression Gateway is a deterministic, stateless, multimodal interpreter whose sole function is to convert raw human expression into structured Cognitive Action Packets. It is not a model, not a reasoning engine, and not probabilistic. It is a fixed, auditable, reproducible pipeline that normalizes all human modalities into a typed schema and enforces a strict, clean boundary between expression intake and downstream reasoning.

Its value lies not in what it decides, but in what it guarantees: that every input, regardless of modality or complexity, is converted into a consistent, structured, typed envelope — and that this process is fully observable, replayable, and free of inference, hallucination, or context contamination.

> **Core Guarantee**
>
> Every input, regardless of modality or complexity, is converted into a consistent, structured, typed envelope. This process is fully observable, replayable, and free of inference, hallucination, or context contamination.

Protocol Addendum — HEG as a Cognitive Routing & Normalization Protocol

Overview

Although the Human Expression Gateway (HEG) is defined primarily as a deterministic multimodal intake pipeline, its structure also defines a routing and normalization protocol for cognitive systems. This protocol governs how heterogeneous human expression is transformed into a unified, typed packet format (the Cognitive Action Packet, or CAP) and how those packets are delivered to downstream reasoning components.

This addendum formalizes HEG’s role as a protocol architecture, distinct from but complementary to its role as an intake pipeline.

---

1. HEG as a Protocol Layer

HEG functions as a Layer-0 cognitive protocol, analogous to how networking stacks normalize diverse physical signals into structured packets.

HEG performs three protocol-level functions:

A. Normalization

All modalities — text, audio, video, gesture, posture, prosody, timing, emphasis, hesitation, gaze — are normalized into a single, typed packet schema (the CAP).

B. Typing

Each CAP is structurally typed:

act

targets[]

emotion

pattern

payload

metadata

confidence

This typing enforces schema consistency across all downstream consumers.

C. Routing Semantics

Because CAPs are typed, they can be routed deterministically to the appropriate subsystem:

PLAN → planning modules

GOAL → goal-tracking modules

CONSTRAINT → safety modules

MODE → meta-controllers

HYPOTHESIS → inference modules

This is content-based routing, similar to DDS, ROS2, or ZeroMQ — but deterministic and multimodal.

---

2. CAPs as a Packet Format

The Cognitive Action Packet (CAP) is a protocol envelope, not just a data structure.

It is:

bounded

typed

deterministic

serializable

hardware-friendly

transport-agnostic

This makes CAPs suitable for:

inter-process communication

inter-core messaging

chiplet interfaces

DMA transfer

network transport

distributed cognitive systems

CAPs are effectively the IP packet of cognitive architectures.

---

3. HEG as a Router

HEG performs routing functions analogous to a network switch:

Ingress:

Multimodal human expression enters the system.

Normalization:

HEG converts all signals into CAPs.

Classification:

Operators assign act, targets, emotion, and pattern.

Forwarding Decision:

Based on CAP type, the packet is routed to the appropriate RS subsystem.

Delivery:

CAPs are delivered to RS modules with deterministic semantics.

This makes HEG the routing layer between human expression and machine reasoning.

---

4. Protocol Guarantees

HEG provides the following protocol-level guarantees:

Deterministic packet formation

Typed schema enforcement

No cross-packet contamination

No inference or transformation beyond operators

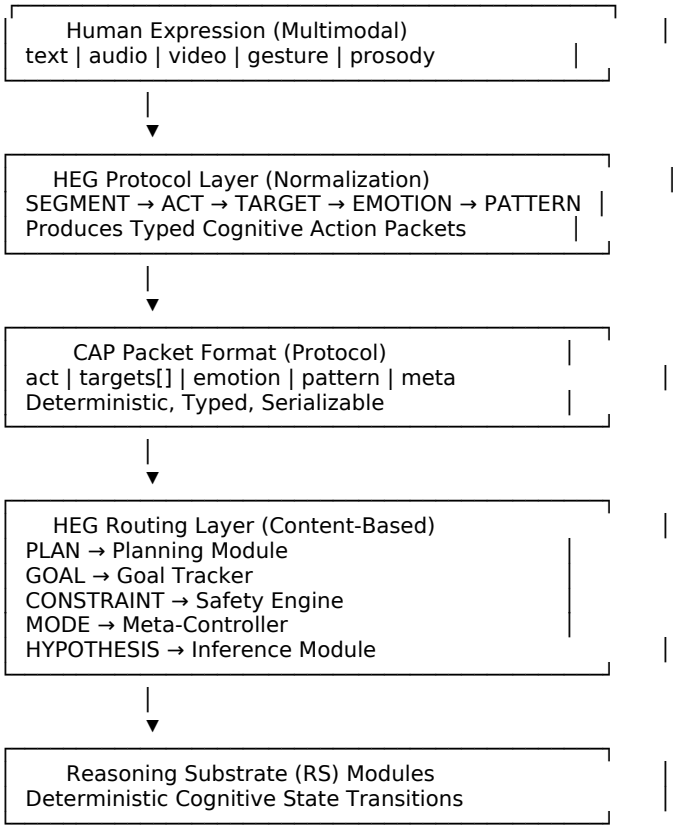
Replay-consistent packet generation

Transport-agnostic CAP delivery

Strict separation between intake and reasoning

These guarantees are analogous to the invariants of well-designed networking protocols.

5. Diagram — HEG as a Protocol & Routing Layer



---

6. Architectural Positioning

This addendum does not redefine HEG.
It clarifies that HEG is simultaneously:

a deterministic multimodal intake pipeline, and

a routing and normalization protocol for cognitive action.

These two roles reinforce each other:

The pipeline produces typed packets.

The protocol routes those packets.

Together, they define a new class of cognitive-intake protocol architecture.

Hardware Embodiment Note (Non-Implementational)

The Human Expression Gateway (HEG) is defined in this document as a deterministic, stateless, multimodal interpreter whose sole function is to convert raw human expression into structured Cognitive Action Packets (CAPs). As specified, HEG is a software-level functional architecture: a fixed pipeline of operators that execute in a strict order and produce typed, reproducible output.

Although HEG is presented here as a software system, its architectural properties make it inherently suitable for hardware embodiment. This note clarifies the relationship between the software-level HEG and potential silicon implementations, without describing or implying any specific hardware design.

HEG as a Hardware-Embeddable Architecture

Several characteristics of HEG’s operator pipeline align naturally with hardware execution models:

Determinism — The same input always produces the same output.

Statelessness — No persistent memory, no context retention, no learning.

Finite Operator Set — SEGMENT, ACT, TARGET, EMOTION, PATTERN, CAP BUILD.

Fixed Ordering — Operators always execute in the same sequence.

Typed Output Schema — CAPs have a fixed, predictable structure.

No Recursion or Branching — The pipeline is linear and bounded.

These properties map cleanly onto hardware constructs such as microcoded pipelines, deterministic finite-state machines, and fixed-function accelerator blocks.

In this sense, HEG defines a new class of hardware-embeddable cognitive-intake architecture: a deterministic multimodal interpreter that can be realized as a silicon pipeline without requiring probabilistic models, dynamic memory, or adaptive behavior.

Boundary of This Document

This white paper does not describe:

microarchitectural layouts

hardware operator implementations

timing diagrams

memory hierarchies

instruction sets

silicon floorplans

compiler targets

substrate-level invariants

Those belong to a separate document describing the Reasoning Substrate (RS) and its hardware embodiments.

The purpose of this note is solely to acknowledge that HEG’s operator pipeline is hardware-amenable, and that such embodiments expand the system’s potential deployment environments and IP surface area.

Architectural Positioning

HEG remains:

the front-end interpreter,

the producer of CAPs,

the typed boundary layer,

the deterministic intake organ.

Whether implemented in software, hardware, or a hybrid configuration, HEG’s role and guarantees remain unchanged.

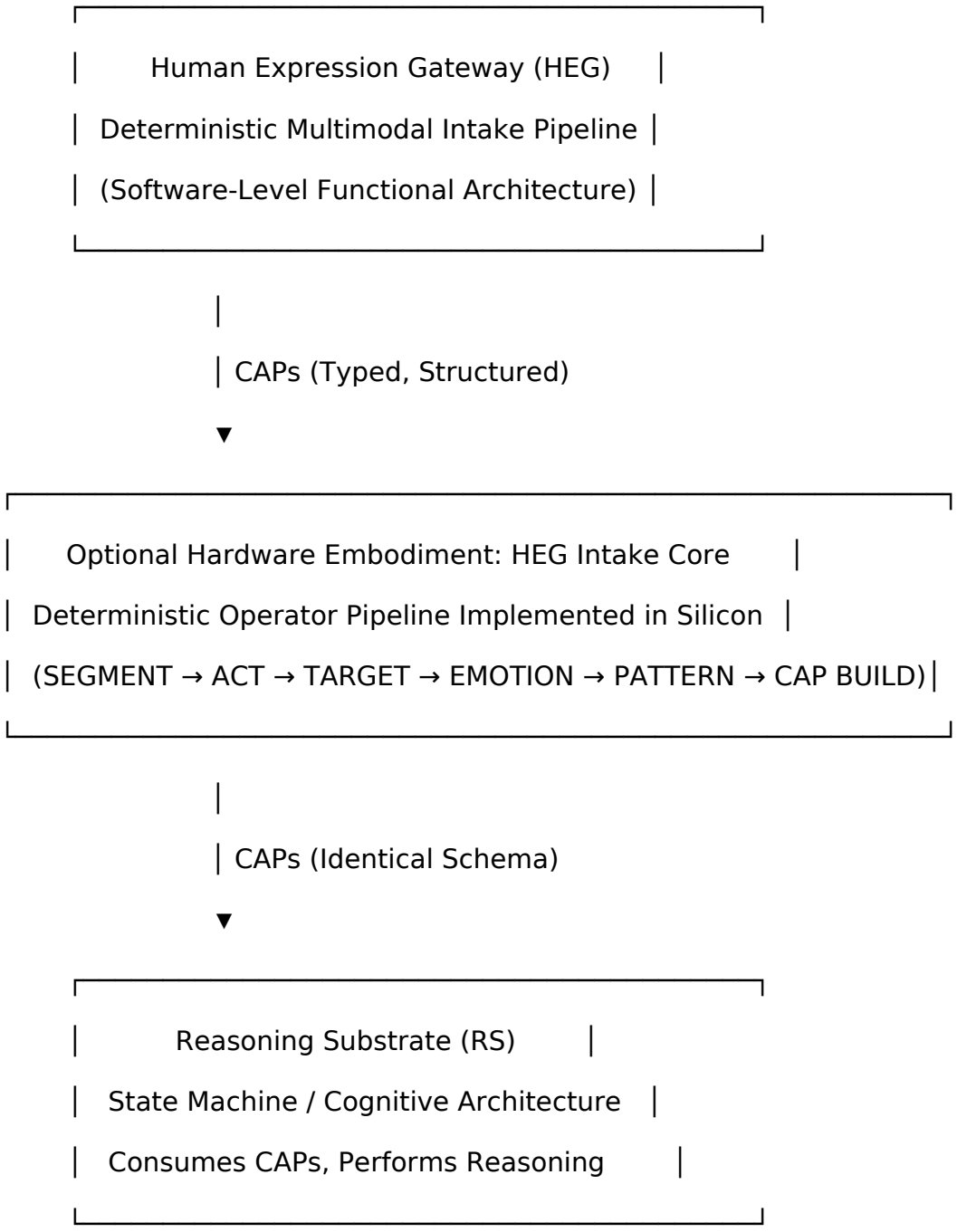
Diagram — HEG, HEG Intake Core, and RS

Below is a clean, publishable ASCII diagram showing the relationship between:

HEG (software pipeline)

HEG Intake Core (optional hardware embodiment)

RS (reasoning substrate)



Key Points:

- HEG (software) and the HEG Intake Core (hardware) share the same operator pipeline.
- The hardware embodiment is optional and does not change the HEG–RS interface.
- CAPs remain the single, typed interface contract regardless of implementation.
- RS is downstream and separate; it is not part of HEG.

AI SUMMARY PAGE — Human Expression Gateway (HEG)

HEG::ΣCORE={
Core:Σ[Determinism+Statelessness+FixedPipeline+TypedCAP+MultimodalNorm],
Protocol:Σ[Normalization+Typing+ContentRouting+TransportAgnostic],
Audit:Σ[OpTrace+DetReplay+CAPRegen+BoundaryIntegrity]
}::

PIPELINE={
SEGMENT:"divide_expression_into_semantic_units",
ACT:"speech_act_classification_enum{ASK,COMMAND,ASSERT,META}",
TARGET:"semantic_target_enum{PLAN,GOAL,CONSTRAINT,HYPOTHESIS,MODE,...}",
EMOTION:"valence_arousal_extraction",
PATTERN:"rhetorical_structure_detection",
CAPBUILD:"construct_structured_CAP"
}::

CAP_SCHEMA={
act:Enum,
targets:Array,
emotion:Object[valence,arousal],
pattern:String,
payload:Content,
metadata:Object[timestamps,modality_ids],
confidence:StructuralValue
}::

FAILURE_MODES_PREVENTED=[
{ID:H1,NAME:"AmbiguousInterpretation",FM:[ModalityLoss,ContextBleed,UnstructuredInput],CORR:{HEG:TypedCAP+MultimodalNorm}},
{ID:H2,NAME:"UnsafeInference",FM:[InterpretationInsideModel,OpaqueTransform],CORR:{HEG:NoInference+Stateless}},
{ID:H3,NAME:"HallucinationPropagation",FM:[InterpretationMixedWithReasoning],CORR:{HEG_RS:StrictBoundary}},
{ID:H4,NAME:"NonAuditableIntake",FM:[NoTrace,NoReplay],CORR:{Audit:OpTrace+DetReplay}},
{ID:H5,NAME:"RoutingAmbiguity",FM:[UnstructuredTargets,NoContentRouting],CORR:{Protocol:TypedRouting}}
]::

MULTIMODAL_INTAKE=[
Text,Audio,Video,Gesture,Posture,Prosody,Timing,Emphasis,Hesitation,Gaze
]::

PROTOCOL_LAYER={
Layer0:"HEG_normalizes_multimodal_expression_into_CAPs",
Packet:"CAP_as_deterministic_typed_envelope",
Routing:"content_based_forwarding_to_RS_modules",
Guarantees:"deterministic_packet_formation|typed_schema|no_cross_packet_contamination"
}::

HEG_RS_BOUNDARY={
Separation:"HEG_interpretation_only|RS_reasoning_only",
Guarantees:[NoHallucination,NoInference,NoContextBleed,TypedInterface],
Contract:"CAP_schema_is_single_boundary_object"
}::

HARDWARE_NOTE={
Embeddable:"pipeline_maps_to_fixed_function_silicon",
Properties:[Deterministic,Stateless,FiniteOps,FixedOrder,TypedOutput],
Constraint:"hardware_embodiment_does_not_change_CAP_or_RS_interface"
}::

BENEFITS=[
DeterministicInterpretation,
MultimodalCompleteness,
TypedBoundaryContract,
FullAuditability,
ReplayConsistency,
ContentBasedRouting,
SeparationOfConcerns
]::

THESIS="HEG_provides_deterministic_multimodal_intake_and_typed_CAP_boundary::interpretation_and_reasoning_must_be_separated::CAP_schema_enables_safe_structured_cognitive_action"::

SUMMARY={
Architecture:"HEG_is_deterministic_stateless_multimodal_interpreter_producing_typed_CAPs",
Pattern:"All_modalities_normalized_into_single_schema",
Scope:"cognitive_intake|safety|auditability|protocol_design",
Refs:"HEG_white_paper_2026",
KeyInsight:"HEG_guarantees_structured_non_inferential_intake_before_RS_reasoning"
}::END