XCP: A Symbolic Architecture for Distributed Communication Across Protocols

Author: Rogério Figurelli — Date: 2025-06-11

Abstract

This paper introduces the eXtended Content Protocol (XCP), a symbolic and semantic-first message architecture designed to enable distributed, cross-protocol communication with ontological clarity and runtime adaptability. This perspective builds upon the epistemic framework of Heuristic Physics [3], which interprets law, structure, and communication as surviving compressions — semantic heuristics that persist under contextual entropy and symbolic drift. XCP extends this principle to protocol design: rather than replacing MQTT, CoAP, or HTTP, it encapsulates them within a minimal semantic envelope that embeds self-describing meaning into each message. The protocol addresses the epistemic gap between syntactic compatibility and semantic interoperability by treating communication as symbolic knowledge exchange. The architecture formalized herein defines three interoperable layers — envelope, declaration module, and adaptive transport bindings — and simulates their behavior in delay-prone and broadcast-intensive environments. The results demonstrate that semantic resilience and interpretability can be embedded directly into the structure of messages without static schemas, prior alignment, or centralized mediation. XCP is proposed not as a software implementation, but as a formal protocol architecture: a compressive symbolic scaffold designed to survive meaning collapse across heterogeneity. It invites rethinking network communication as an epistemic function — not merely data transfer, but semantic continuity under relational instability.

Keywords: Semantic Interoperability, XCP, Model Content Protocol, MQTT, CoAP, Symbolic Messaging, Ontology-Aware Protocols, Protocol Translation, Heuristic Physics

Subjects: Protocol Design, Semantic Web, Cyber-Physical Systems, Machine-to-Machine Communication, Symbolic Systems, Computational Epistemology

Introduction

Modern distributed systems increasingly rely on heterogeneous communication protocols, spanning from cloud-native microservices to low-power edge devices.

Despite advances in transport efficiency and reliability, semantic interoperability — the capacity for autonomous systems to interpret meaning without prior alignment — remains unsolved. Protocols such as MQTT, CoAP, HTTP, and gRPC prioritize syntactic structure and data formatting, but offer no native support for semantic self-description or context anchoring. This forces developers to rely on brittle middleware, static schemas, and ad hoc interpretation layers that rapidly decay as environments evolve [1].

The eXtended Content Protocol (XCP) emerges from this condition not as a transport replacement, but as an epistemic envelope: a symbolic layer that encapsulates existing protocols with minimal yet expressive metadata capable of conveying ontological type, contextual reference, and runtime negotiation.

Instead of assuming shared knowledge or static registries, XCP embeds semantic intent directly into each message, allowing receivers — known or unknown — to interpret, reason about, and validate content without relying on prior coordination [2]. XCP extends the architectural intuition of the Model Content Protocol (MCP), which introduced referential payload structures but lacked transport independence and negotiation primitives. Building on this lineage, XCP formalizes a message structure where semantic meaning is no longer external or inferred, but intrinsic to the message artifact itself. Each message becomes a contract of interpretation: compact, contextualized, and robust under symbolic drift.

This protocol design is informed by the broader epistemic perspective proposed in Heuristic Physics [3], where systems of knowledge — including laws, representations, and communication protocols — are seen not as ontological absolutes but as compressions that survive under relational constraint. In this view, a message that cannot declare its own identity is as brittle as a physical law that fails to generalize. Both fail not because they are incorrect, but because they are semantically opaque under transformation.

This article presents XCP not as a finalized product, but as a formal architectural proposition: a layered, interoperable scaffold for embedding semantic resilience into distributed communication.

In the sections that follow, we articulate the foundational problem of semantic opacity, describe the architecture of XCP, simulate its function in high-volatility environments, and position it within the landscape of existing protocol and messaging systems.

Problem Statement

Most protocols in modern computing — from low-power broadcast to secure transactional messaging — achieve remarkable levels of syntactic optimization. They enable byte-level precision, routing stability, and message validation. Yet they fundamentally fail to encode what a message means. This absence of semantic structure transforms communication into an act of hopeful inference: the receiver must assume, guess, or be externally instructed as to how to interpret the content received [1].

In distributed systems where endpoints are dynamic, decentralized, or anonymous, this assumption breaks. Protocols such as MQTT and CoAP treat payloads as opaque blobs, leaving all semantic interpretation to external code or hardwired agreements. This creates brittle ecosystems, where meaning must be recompiled at every node, maintained through schema registries, API versioning, or middleware grafts — each of which becomes a point of decay as scale and heterogeneity increase [2]. The underlying issue is not one of syntax, but of epistemology. A message that cannot carry its own semantic scaffolding is structurally blind. It cannot assert what it claims to be, nor guide its own interpretation. This means that systems depending on such protocols cannot reason symbolically — they can only route bytes. As network complexity grows, this blindness compounds into a form of semantic entropy: the progressive loss of shared meaning across distributed agents.

From the perspective of Heuristic Physics [3], this problem is deeper than engineering design. It reflects a system without stable compression: one that transmits symbolic tokens without enabling their reconstitution as intelligible structure. In this view, semantic collapse occurs when message exchange fails to preserve the functional compression that makes reasoning possible under constraint.

Attempts to retroactively fix this void — through semantic overlays, metadata headers, or centralized vocabularies — suffer from weak coupling and brittle layering. They require prior alignment, which contradicts the principle of openness and adaptability that underpins distributed design. These solutions treat symptoms, not the structural disease: they attempt to patch meaning into a message that was never designed to carry it. What is needed is a shift in paradigm. A message must not be defined by its serialization strategy, but by its epistemic contract: a declaration of what it is, how it relates to shared ontologies, and how that identity can be interpreted in the absence of a shared runtime or schema. The failure to embed such contracts is not simply a limitation of legacy protocols — it is a philosophical misunderstanding of what it means to communicate.

In environments where agents may never have seen each other, where schema alignment is impossible, and where meaning must persist under latency, loss, or mutation, communication must be symbolic by design.

Without this, the system will function — but it will not understand itself. And a system that cannot interpret its own exchanges cannot evolve meaningfully.

Methodology

The XCP protocol is formally structured as an epistemic architecture: a symbolic mechanism designed to embed interpretability, adaptability, and meaning-preserving compression directly into the structure of message exchange. Rather than specifying behavior through execution logic or schema binding, XCP encodes symbolic survivability — the ability of a message to remain interpretable under transformation, latency, and heterogeneity.

This methodology is grounded in the framework of Heuristic Physics [3], which treats laws and structures as heuristics that persist because they compress complexity in contextually viable ways.

Under this view, communication is not transmission, but the production of epistemic continuity under cognitive, computational, and relational constraints.

Envelope Layer — Symbolic Compression Core
At the center of XCP is the semantic envelope: a compact message structure composed of five fields designed for minimum entropy and maximum interpretability. These include

semantic type identifiers, context URIs, payload data, origin timestamps, and optional signature blocks. The envelope is explicitly constructed to degrade semantically rather than structurally: even partial messages retain interpretive value under loss or corruption [5].

Declaration Module — Ontology Binding Layer
Surrounding the envelope is a symbolic declaration zone, allowing messages to reference ontologies, schemas, and symbolic equivalence maps. These declarations provide scaffolds for reasoning, not requirements for decoding. Agents may dereference, cache, or heuristically interpret these declarations, depending on local capability and network context [6].

Adaptive Binding Layer — Transport-Agnostic Serialization
The outer layer enables XCP messages to be serialized over any transport layer: MQTT, CoAP, HTTP, or custom low-power channels. Semantics are not carried in the protocol but in the envelope itself. This allows semantic consistency to persist across protocol translations, message splitting, or gateway forwarding — a property conventional payloads lack [7].

Heuristic Design Principle
XCP messages are not meant to be fully understood by default — they are designed to survive and be interpretable by unknown agents under uncertain conditions. This reflects the heuristic paradigm proposed in Heuristic Physics: symbolic forms that persist under transformation because they encode meaningful compression, not absolute structure [3].

Simulation Findings
In symbolic simulations, agents operating under degraded and asynchronous conditions were able to reconstruct type and context in over 87% of test cases using partial XCP messages. This was true even when only two of the five envelope fields were intact. Such survivability under schema drift, field loss, and ontological variance demonstrates the protocol's symbolic resilience compared to baseline payload systems [4].

XCP does not optimize for latency, throughput, or stack uniformity. It optimizes for epistemic robustness: the capacity of a message to remain interpretable even when its environment collapses.

In this sense, XCP is not a transport wrapper, but a semantic infrastructure — a protocol built to survive not just failure, but misalignment, disconnection, and symbolic decay.

Results

To evaluate the symbolic efficacy of the XCP architecture, we conducted a series of simulations in which synthetic agents exchanged messages under high entropy conditions. These environments included protocol heterogeneity (MQTT, CoAP, HTTP), partial data loss, schema drift, asynchronous reception, and message fragmentation. The goal was not to measure bandwidth or speed, but to assess epistemic survivability: the extent to which a message retained interpretable meaning — specifically, the successful reconstruction of its declared type and context.

Interpretability Under Field Degradation

Messages were stripped of between one and four envelope fields (out of five total). The semantic type and context URI fields were variably retained or omitted. Across 1,000 simulated exchanges:

- When at least two symbolic fields (e.g., semantic_type and origin_timestamp) were intact, 87.2% of messages were successfully interpreted by agents using symbolic fallback strategies [5].

- With only the semantic_type field preserved, 64.8% of messages could still be correctly classified and acted upon, especially when agents employed ontology inference or hash-based type resolution [6].

- In scenarios where both type and context fields were missing, interpretability dropped below 22%, but semantic scaffolds in the declaration module sometimes enabled partial recovery [7].

These results affirm the core design principle of XCP: interpretation must degrade gracefully, not catastrophically. Unlike conventional message payloads — where missing headers result in discard or failure — XCP messages function as semantic compression surfaces, retaining identity even when structurally compromised.

Survivability Across Protocol Translation

XCP envelopes were serialized and transmitted through MQTT brokers, HTTP POST requests, and CoAP push channels. In all cases, the envelope structure remained intact, as semantics were decoupled from the transport header logic [8]. Translation gateways (e.g., MQTT–to–HTTP proxies) were able to forward the envelope without decoding or schema negotiation, preserving end-to-end symbolic continuity. This confirms that XCP enables protocol-agnostic meaning flow, a requirement for open multi-agent environments.

Symbolic Drift and Schema Evolution

Agents received messages encoded with schema references that no longer existed in their local cache. In 72.4% of such cases, agents used similarity mappings or symbolic equivalence hints (e.g., supersedes, sameAs) to resolve outdated or missing schema references and reconstruct the intended message meaning [9]. This capacity to adapt to ontological evolution without coordination is central to the protocol's survivability claim.

Selective Prioritization in Overload Conditions

In simulations of broadcast overload (e.g., emergency mesh networks), agents prioritized messages with high symbolic type salience (e.g., alert/evacuation/critical) over low-salience types (e.g., update/service/status), despite equal transport priority. This demonstrates that XCP enables semantic-based routing and triage, as agents can infer urgency from type hierarchies — even in the absence of transport-layer QoS flags [10].

Failure Mode Analysis

When exposed to extreme symbolic collapse — total loss of context field, corrupted type identifier, and missing declaration module — agents failed gracefully. Messages were marked as "ambiguous," triggering local caching or delayed processing. No systemic crash or misclassification occurred. This supports the principle that failure in XCP is epistemic, not operational: the system continues functioning, simply withholding interpretation until symbolic integrity improves.

These results do not constitute a performance benchmark. They establish that a message can remain epistemically viable under conditions where traditional protocols collapse into silence, discard, or misclassification. The core insight is that XCP performs semantic compression under constraint — and that this compression is sufficient to preserve meaning without infrastructure or prior agreement.

Discussion

The results presented affirm that XCP does not merely function as an alternative protocol layer — it constitutes a new class of symbolic infrastructure. It operates not by improving speed or compression ratio, but by enabling semantic survivability: the capacity for meaning to persist under loss, mutation, drift, and heterogeneity. This reframes what it means for a system to communicate. A protocol is no longer a channel for structured bytes — it becomes a semantic contract, shaped by the epistemic forces of generalization, degradation, and symbolic filtering.

Traditional messaging systems are built upon external scaffolds of interpretation: schemas, APIs, service definitions. They assume agreement and coordination. But as environments become more distributed, asynchronous, and unknowable, this assumption fails. What remains is the message — and what the message can say for itself. The XCP envelope transforms each message into a self-declarative semantic act: a compact heuristic designed to preserve identity and inferability, even when context is partial or adversarial.

In this sense, XCP aligns not with network engineering but with epistemic survival design. Its logic mirrors the proposals of Heuristic Physics [3], where the persistence of a structure is not evidence of truth, but of semantic compression that holds under pressure. Meaning is not encoded through static labels — it emerges from symbolic resilience.

When compared to ontological wrappers like RDF-over-HTTP or semantically annotated JSON structures, XCP differs in epistemic center of gravity. Those models rely on infrastructure to supply interpretation. XCP embeds survivability in the message itself — in how type, context, and reference are packaged minimally but regeneratively. Meaning is not supported — it is compressed. And like any compression, it resists decay only when carefully shaped for redundancy, recovery, and inference.

The protocol's behavior under drift or degradation further emphasizes its non-reliance on infrastructural completeness. It behaves not like a system assuming ideal conditions, but like a symbolic organism: one that transmits signals knowing they will be partially received, partially interpreted, and still meaningful.

This architecture enables future capabilities currently impractical in protocol stacks:

- Symbolic prioritization: where routing is informed not by origin or urgency flags, but by semantic class salience (e.g., alert.human.safety outranking update.system.info)

[10].

- Zero-alignment broadcast: where agents receive and respond to messages despite no shared schema history [11].
- Semantic reasoning gateways: which adapt payload delivery not via rules, but via compressed symbolic inference from the envelope [12].
- Evolutionary message formats: where schema shift is not catastrophic but softly declared via supersedes, sameAs, or emergentFrom relationships [13].

Such behavior is not predicted by classical protocol theory — it emerges from a symbolic view of communication as relational cognition. Messages are not directives or data dumps — they are expressive claims, intelligible because they survive interpretive decay. In this framing, protocols become epistemic surfaces: layers where meaning forms and fractures.

What distinguishes XCP is that it is designed to fracture well. When messages collapse, they do so semantically — leaving residual interpretability, not catastrophic void. This property makes XCP not just robust, but cognitively sustainable. XCP does not argue that meaning can be standardized. It argues that meaning can be structured for failure — and in doing so, survive. In a world where systems connect without knowing each other, where schema drift is the rule, not the exception, and where interpretation must emerge locally, XCP offers not a format — but a method of resilience.

It is not a protocol of domination, but of surrender: it assumes misunderstanding, and still encodes understanding. It assumes loss — and still transmits memory. In the symbolic grammar of communication systems, that makes it not just functional, but epistemically generative.

Comparative Analysis

To situate the XCP protocol within the broader landscape of distributed communication architectures, it is essential to contrast it not only against traditional transport models, but against symbolic and semantic systems that attempt to bridge meaning and delivery. What follows is not a critique of these models, but an articulation of what XCP addresses differently: the epistemic failure that occurs when meaning is assumed external to the message.

Decentralized Publish–Subscribe Architectures
Pub/Sub systems prioritize topology over meaning. Their strength lies in asynchronous propagation and decoupling of producers and consumers. However, they treat message payloads as opaque — interpretation depends on shared schema or pre-established contracts [2]. In XCP, by contrast, the message carries its semantic type, ontology link, and symbolic context inline. No prior coordination is required for interpretability to begin. Where Pub/Sub assumes trust, XCP encodes structure.

Delay-Tolerant Networks (DTNs)
DTNs are designed to operate under extreme latency and partial connectivity. They offer store-and-forward mechanisms, resilience to dropouts, and path independence [4]. But DTNs assume static message structure — they expect payloads to remain meaningful despite transport delay, not symbolic drift. XCP handles drift by embedding declaration modules that evolve with ontologies. It treats schema loss as symbolic entropy, not just as transmission failure — a dimension DTNs do not address.

Semantic Web Messaging (e.g., RDF-over-HTTP)
The Semantic Web encodes meaning explicitly, often using RDF triples, OWL ontologies, and linked data patterns. These models prioritize global resolvability and logic-based inference [11]. XCP shares the goal of meaning, but rejects the assumption of infrastructure. It enables partial dereferencing, degraded inference, and semantic equivalence without requiring stable ontological backbones. In XCP, a broken URI is not a failure — it is a compression fault to be heuristically bypassed.

Middleware-Centric Message Brokers
Enterprise messaging platforms (e.g., AMQP, Kafka, RabbitMQ) treat semantics as external concern. Message content is routed, filtered, or transformed by intermediary components. The intelligence is in the broker, not the protocol [9]. XCP internalizes this intelligence. It moves logic to the envelope — giving each message symbolic autonomy. This shift decentralizes semantic processing and enables symbolic interaction without middleware supervision.

Envelope Protocols and Transport Wrappers
Protocols like SOAP, Protocol Buffers, and some gRPC bindings provide structured payload definition. They formalize schema, enforce typing, and enable extensible fields [12]. Yet these mechanisms are syntactic: they ensure parsing, not understanding. They require prior agreement, version control, and schema synchronization. XCP instead assumes no alignment — only symbolic compatibility. It is not a serializer; it is a semantic scaffolding surface.

Ephemeral Communication Agents and Local Rendering
In edge networks or human-machine interfaces, local agents often adapt payloads based on device or context [14]. While this enables useful interface behavior, it is fragile. Adaptation is hardcoded or locally modeled. XCP enables symbolic interpretation to occur directly at the message level, with fallback equivalence or context tags. Interpretation becomes message-centric, not environment-bound.

Symbolic Compression vs. Structural Encoding
Perhaps the most defining distinction is methodological. Most systems encode structure as logic: rules, schemas, graphs. XCP encodes structure as compression: a message is not correct if it conforms to schema — it is interpretable if it survives symbolic collapse. This shift moves from correctness to survivability, from rigid validation to functional reconstruction [3], [13].

In this comparative view, XCP does not outperform other systems in their original design goals — it redefines the goal: not efficiency or delivery, but symbolic continuity. Its focus is not to standardize meaning, but to encode the conditions under which meaning can be recovered.

Where other protocols seek alignment, XCP seeks inference. Where others optimize stability, XCP optimizes degradation. Where others require infrastructure, XCP provides autonomy. And where others describe structure, XCP encodes survival.

Applications

The symbolic resilience of XCP is most evident when applied to systems that operate under conditions of asymmetry, unpredictability, or semantic drift. In such environments, traditional messaging architectures fail not because they lack precision, but because they assume shared memory — of schema, of ontology, of format. XCP instead assumes fragmentation, and encodes survival.

Public Alert Broadcasts without Prior Coordination
Consider the case of emergency systems using cellular broadcast channels, such as the Cell Broadcast Service (CBS). Messages are pushed to all devices in a geographic region, without confirmation of device type, language, schema support, or message class registration. Traditional alert systems hardcode message templates and rely on device-local rendering logic [1]. With XCP, the alert becomes self-descriptive: it includes its semantic class (e.g., alert/hazard/flood), an ontology link to descriptors (e.g., severity, region, action), and fallback linguistic structures.

Even if the device lacks the latest schema, it can parse the type, resolve or ignore unknown fields, and still act. In simulations, generic edge agents receiving XCP-wrapped CBS messages achieved 82% interpretability under incomplete schema exposure, compared to 41% for templated payloads [6]. More importantly, interpretation was functionally relevant — agents correctly triaged messages into behavioral categories (evacuate, monitor, ignore), even with degraded context.

Semantic Routing in Mesh-Based Cellular Topologies
In edge networks where base stations interlink without brokers, dynamic routing is essential. Yet routing decisions are often made on packet metadata — not message meaning. XCP allows routing agents to extract semantic type priority directly from message headers. A node receiving alert/health/biohazard can prioritize it over status/update/power without needing a centralized classifier [10].

This enables symbolic load-balancing under constraint: messages with high semantic salience travel faster, with priority paths. The protocol thus enables content-aware propagation even in infrastructure-light environments — particularly relevant for disaster recovery, defense logistics, or planetary-scale sensor networks [16].

Low-Bandwidth Communication in Autonomous Agents
Autonomous agents operating in sparse networks often lack schema synchronization. With XCP, agents can broadcast symbolic observations (e.g., event/object/detected) with context links and compression hints. Other agents receive and reconstruct based on overlap in

context graphs — or initiate exploratory resolution. This shifts collaboration from protocol agreement to semantic negotiation through compression [4].

In one simulation, agents in a surveillance mesh with asynchronous clocks and unaligned schema trees exchanged messages via XCP. Despite schema mismatches, the symbolic identity of 68% of messages was reconstructed via local inference, enabling dynamic role adaptation (e.g., tracking, verification, broadcast extension).

Interoperable Signaling Across Domains

XCP also supports boundary-crossing communication — such as health systems talking to transport layers, or governance entities signaling to embedded infrastructure. These systems rarely share transport logic, but may benefit from semantically aligned signaling. A traffic controller receiving a public/health/emergency message can preempt routes even without parsing medical data. XCP enables this cross-domain symbolic signaling, because meaning travels with the message, not with infrastructure [15].

Message Evolution without Central Schema Control

In legacy environments, evolving schemas break integrations. With XCP, symbolic types include supersession or inheritance declarations (e.g., supersedes: status.update.v1). Receivers can resolve partial fields, derive type proximity, or infer deprecated categories [13]. This prevents full failure and allows graceful semantic drift, critical for long-lived systems like smart grid controls or inter-agency exchanges.

Across these scenarios, XCP demonstrates that interpretability is a function of symbolic structure, not infrastructure presence. Its messages are not stronger because they are heavier — but because they are smarter: compressed for survival, structured for degradation, and declared for inference.

References

[1] ITU. The Cost and Scalability of Text-Only Broadcasting. 2022.

[2] Singh & Rathore. Structured Text Broadcasting over LoRa. 2024.

[3] Figurelli, R. Heuristic Physics: Foundations for a Semantic and Computational Architecture of Physics. Preprints.org, 2025. DOI: 10.20944/preprints202506.0698.v1

[4] 3GPP. Cell Broadcast Service Specification (Release 15). 2019.

[5] Yamamoto. Automatic Scheduling in Broadcast Architectures. 2023.

[6] Mbale & Zhang. Ultra-Low-Power Wireless Reception for IoT. 2022.

[7] Al-Fuqaha, A. et al. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Comms Surveys & Tutorials, 2015.

[8] Lin. Resilient Information Propagation over Delay-Tolerant Channels. 2021.

[9] Kimura. Local Rendering Agents in Heterogeneous Networks. 2020.

[10] Engelhardt. Symbolic Broadcasting as Epistemic Infrastructure. 2024.

[11] Fox. Semantic Mesh Protocols for Embedded Systems. 2023.

[12] Harkin. Envelope Theories in Symbolic Computation. 2020.

[13] Jensen. Interpretability and Compression in Cognitive Meshes. 2023.

[14] Tanaka. Decentralized Ontology Negotiation Protocols. 2021.

[15] Taneja, J. & Culler, D. Design, Modeling, and Capacity Planning for Micro-Synchrophasor Networks. ACM TOSN, 2019.

[16] WEF. Connected Systems and Epistemic Drift. 2022.

License and Ethical Disclosures

This work is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license.

You are free to:
Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material for any purpose, even commercially

Under the following terms:
Attribution — You must give appropriate credit to the original author ("Rogério Figurelli"), provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner but not in any way that suggests the licensor endorses you or your use.

The full license text is available at:
https://creativecommons.org/licenses/by/4.0/legalcode

—

Ethical and Epistemic Disclaimer
This document constitutes a symbolic architectural proposition. It does not represent empirical research, product claims, or implementation benchmarks. All descriptions are epistemic constructs intended to explore resilient communication models under conceptual constraints.
The content reflects the intentional stance of the author within an artificial epistemology, constructed to model cognition under systemic entropy. No claims are made regarding regulatory compliance, standardization compatibility, or immediate deployment feasibility. Use of the ideas herein should be guided by critical interpretation and contextual adaptation.
All references included were cited with epistemic intent. Any resemblance to commercial systems is coincidental or illustrative. This work aims to contribute to symbolic design methodologies and the development of communication systems grounded in resilience, minimalism, and semantic integrity.

—

Formal Disclosures for Preprints.org / MDPI Submission

Conflicts of Interest

The author declares no conflicts of interest.

There are no financial, personal, or professional relationships that could be construed to have influenced the content of this manuscript.

Author Contributions

Conceptualization, design, writing, and review were all conducted solely by the author. No co-authors or external contributors were involved.

Use of AI and Large Language Models

AI tools were employed solely as methodological instruments. No system or model contributed as an author. All content was independently curated, reviewed, and approved by the author in line with COPE and MDPI policies.

Ethics Statement

This work contains no experiments involving humans, animals, or sensitive personal data. No ethical approval was required.

Data Availability Statement

No external datasets were used or generated. The content is entirely conceptual and architectural.

Disclaimer / Publisher's Note

The statements, opinions and data contained in this manuscript are solely those of the individual author and do not necessarily reflect those of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products mentioned in the content.