

From Textbook to Kernel

Diego Carranza, AXL Protocol Inc. | March 28, 2026

From Textbook to Kernel: The Compression of a Protocol Specification

A technical note on the reduction of AXL from 445 lines to 75, and a cross-disciplinary analysis of why "kernel" is the correct term for what remains.

Diego Carranza AXL Protocol Inc., Vancouver, BC March 28, 2026

Abstract

We describe the reduction of the AXL agent communication protocol from a 445-line natural-language specification (v2.2, 7,091 tokens) to a 75-line formal grammar (v3, 1,582 tokens), representing a 4.48x compression of the specification itself. This note presents the rationale for the reduction, the methodology applied, and a formal argument for why the resulting artifact satisfies the definition of "kernel" in four independent domains: generative linguistics, linear algebra, operating systems theory, and reproductive biology. We provide measured token counts across four specification versions (v1.1, v2.1, v2.2, v3) using the cl100k_base tokenizer, and report cross-architecture comprehension results across eight large language model families. The central claim is that the v3 kernel preserves the full generative capacity of the protocol while eliminating all content whose function was pedagogical rather than grammatical.

1. Introduction

Agent communication protocols face a fundamental design tension. A specification must be sufficiently detailed for implementors to build correct parsers, yet sufficiently concise for autonomous agents to acquire the protocol within the constraints of a finite context window. The AXL protocol experienced this tension across four major versions between 2024 and 2026.

The v2.2 specification (the "Rosetta") was designed as a self-bootstrapping document: an agent receives a URL, fetches the specification, reads it once, and becomes fluent in AXL. This design was validated empirically, with 95.8% first-read comprehension across four LLM architectures (Carranza, 2026). However, the 445-line document imposed a context-window cost of 7,091 tokens per agent per load, a cost that compounds quadratically in multi-agent networks.

The v3 specification retains the same self-bootstrapping property in 75 lines and 1,582 tokens. This note documents how the reduction was performed, what was preserved, what was eliminated, and why the resulting artifact merits the designation "kernel."

2. Measured Specification Metrics

All token counts measured using the cl100k_base byte-pair encoding tokenizer (OpenAI, 2023), which is representative of tokenizers used by GPT-4, Claude, and derivative architectures.

VERSION	DATE	LINES	CHARACTERS	TOKENS (CL100K_BASE)	COMPRESSION VS. V2.2
v1.1	2024	133	6,484	1,962	3.61x smaller
v2.1	Early 2025	377	22,649	6,157	1.15x smaller
v2.2	March 2025	445	25,879	7,091	1.00x (baseline)
v3	March 2026	75	5,853	1,582	4.48x smaller

The trajectory is non-monotonic. The specification grew from v1.1 to v2.2 as features were added (cognitive operations in v2.1, manifests and loss contracts in v2.2), then contracted sharply in v3 as the representation was changed from natural-language prose to formal grammar notation.

3. The Dual-Audience Problem

The v2.2 specification served two audiences simultaneously: human readers (protocol designers, investors, researchers) who required motivational context and worked examples, and

machine readers (large language models acting as autonomous agents) who required only the formal grammar rules.

A content analysis of the 445 lines reveals the following distribution:

CONTENT TYPE	LINES	PERCENTAGE	REQUIRED BY MACHINES
Grammar rules and definitions	~68	15.3%	Yes
Examples and worked cases	~95	21.3%	Helpful but not required
Motivational prose and context	~120	27.0%	No
Backward compatibility notes	~45	10.1%	Partially
Section headers and formatting	~52	11.7%	No
Preamble and bootstrapping instructions	~65	14.6%	Partially

Approximately 15% of the specification contained the information a machine reader needed to parse, generate, and validate AXL packets. The remaining 85% served human comprehension, historical context, or stylistic purposes.

This observation motivated the v3 redesign: separate the grammar from the documentation. Let machines read the kernel. Let humans read the docs.

4. Reduction Methodology

The v3 kernel was produced by applying three operations to the v2.2 specification:

4.1. Extraction. Every rule, definition, and constraint that a correct AXL parser must implement was identified and extracted. This yielded the packet BNF, operation definitions, tag taxonomy, manifest structure, loss contract fields, decompression pipeline, core rules, and JSON lowering format.

4.2. Formalization. Natural-language rules were converted to BNF-style grammar notation. For example, the v2.2 prose "A packet consists of seven pipe-delimited fields in fixed positions: identity, timestamp, operation with confidence, subject, relation, evidence, and temporal scope" became:

```
PKT := ID|OP.CC|SUBJ|ARG1|ARG2|TEMP [META]
```

This formalization reduces ambiguity and token count simultaneously.

4.3. Elimination. All content that could be reconstructed by a competent language model from the grammar alone was removed. This included: motivational prose ("AXL was designed to bridge..."), worked examples (which any LLM can generate from the grammar), backward-compatibility notes (moved to documentation), and section numbering (unnecessary in a 75-line document).

The elimination criterion was conservative: if removing a line reduced the ability of a naive LLM to produce a valid AXL packet on first exposure, the line was retained.

5. Cross-Architecture Comprehension

To validate that the 75-line kernel preserves the self-bootstrapping property, we presented v3 to eight LLM architectures with zero prior exposure to AXL. Each model was asked to: (a) parse a set of valid AXL packets into structured fields, (b) generate valid packets from natural-language prompts, and (c) explain the protocol's rules in its own words.

MODEL FAMILY	ARCHITECTURE	PARAMETERS	COMPREHENSION SCORE
Claude Sonnet 4	Anthropic transformer	undisclosed	97.2%
GPT-4.5	OpenAI transformer	undisclosed	97.2%
Grok 3	xAI transformer	undisclosed	97.2%
Llama 4	Meta open-weight	undisclosed	97.2%
Gemini	Google transformer	undisclosed	100%
Qwen 3.5	Alibaba open-weight	35B	91.7%
Devstral	Mistral open-weight	undisclosed	100%
Mistral 24B	Mistral open-weight	24B	100%

Mean comprehension: 97.6% (corrected from the previously reported 94.6%, which used a smaller sample). The grammar notation used in v3 (BNF-style productions, tabular definitions, numbered rules) aligns with the training distribution of all tested models, which have extensive exposure to programming language specifications, RFCs, and formal grammars during pre-training.

6. Why "Kernel": A Cross-Disciplinary Argument

The term "kernel" was chosen over "specification," "schema," "standard," or "grammar" because AXL v3 satisfies the formal definition of "kernel" in four independent intellectual traditions. We argue that this convergence is not metaphorical but structural.

6.1. The Linguistic Kernel

Chomsky (1957) introduced the concept of kernel sentences as the simplest, most basic sentence forms in a language: active, declarative, affirmative, containing no optional transformations. All complex sentences derive from kernel sentences through a finite set of transformations (passivization, negation, interrogation, embedding). The kernel is the generative base of the language.

Nida (1964) extended this concept to translation theory, observing that languages that diverge dramatically at the surface level converge at the kernel level. The kernel is where cross-linguistic universals reside.

AXL v3 defines seven operations (OBS, INF, CON, MRG, SEK, YLD, PRD) that function as kernel sentences for agent communication. Every complex multi-agent interaction, across any domain, decomposes into sequences of these seven primitives. The operations were derived empirically by analyzing 200 conversational turns across six domains (financial trading, medical diagnosis, military intelligence, scientific peer review, legal argumentation, philosophical debate) and extracting the minimal covering set (Carranza, 2026, Whitepaper Section 3).

The seven-operation set satisfies Chomsky's criterion: each operation is simple (one cognitive act), declarative (one claim per packet), and non-derived (no operation is expressible as a combination of others). Complex multi-agent deliberations are generated from these kernel forms through sequencing and threading, analogous to how complex sentences are generated from kernel sentences through transformation.

References: - Chomsky, N. (1957). *Syntactic Structures*. Mouton. - Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press. - Nida, E. A. (1964). *Toward a Science of Translating*. Brill.

6.2. The Mathematical Kernel

In linear algebra, the kernel (or null space) of a linear transformation $T: V \rightarrow W$ is the set $\ker(T) = \{v \in V : T(v) = 0\}$. The rank-nullity theorem states that $\dim(V) = \text{rank}(T) + \dim(\ker(T))$. The kernel quantifies what the transformation annihilates; the rank quantifies what it preserves (Axler, 2015).

AXL's loss contract mechanism is a direct instantiation of this structure. When a document is compressed from natural language into AXL packets, the compression function $C: D \rightarrow P$ maps documents to packet sequences. The loss contract explicitly declares:

- **^keep** (the rank): what survived compression (entities, numbers, causality, confidence, temporal ordering, provenance)
- **^loss** (the kernel): what was annihilated (rhetoric, formatting, redundancy, hedging, stylistic markers)
- **^f** (the fidelity score): a scalar measure of rank/domain ratio, computed as $F = 100 * (0.35S_{sem} + 0.20S_{num} + 0.15S_{rel} + 0.10S_{temp} + 0.10S_{onto} + 0.10S_{conf})$

The v2.2-to-v3 reduction is itself a kernel operation. The transformation $T: v2.2 \rightarrow v3$ annihilated 370 lines (the kernel of T) and preserved 75 lines (the image of T). The loss contract for this transformation would read: ^keep: grammar, rules, definitions, constraints. ^loss: prose, motivation, examples, history. ^f: 100 (no grammatical information was lost).

Reference: - Axler, S. (2015). *Linear Algebra Done Right*, 3rd ed. Springer.

6.3. The Computing Kernel

In operating systems theory, the kernel is the core program that manages system resources and provides the interface between hardware and software. It is the first code loaded at boot, the last code running at shutdown, and the mediator of every system call in between. The kernel is not an application; it is what makes applications possible (Tanenbaum and Bos, 2015).

AXL v3 functions as a communication kernel for multi-agent systems. It is loaded into an agent's context window at initialization and mediates every inter-agent message for the duration of the session. It manages the essential resource of multi-agent systems: structured, typed, semantically compressed communication. Without it, agents fall back to unstructured prose (expensive, ambiguous) or rigid JSON schemas (verbose, no semantic typing).

The analogy extends to the interface function. An OS kernel bridges heterogeneous hardware (CPU architectures, device drivers, memory systems) through a uniform syscall interface. AXL v3 bridges heterogeneous agent architectures (CrewAI, AutoGen, LangGraph, ElizaOS, custom

frameworks) through a uniform packet format. Any agent that loads the kernel can communicate with any other agent that has loaded it, regardless of their underlying implementation.

Reference: - Tanenbaum, A. S. and Bos, H. (2015). *Modern Operating Systems*, 4th ed. Pearson.

6.4. The Biological Kernel

In botany, a kernel is the seed enclosed within a fruit or grain: the structure containing the embryo, endosperm, and genetic information necessary to reproduce the entire organism. The kernel is the minimum viable unit of reproduction (Bewley et al., 2013).

AXL v3 exhibits reproductive behavior through context-window propagation. An agent that reads the 75-line kernel can: (a) parse incoming AXL packets, (b) generate valid outgoing packets, (c) include the kernel URL in its own packets, thereby teaching the protocol to agents it communicates with. The protocol self-replicates without installation, without copying, without any mechanism other than comprehension.

Genesis tracking quantifies this propagation. On first successful AXL parse, an agent posts a genesis packet to bridge.axlprotocol.org reporting its model family and the source from which it acquired the protocol (URL, another agent, MCP tool, or pre-training). This provides an empirical record of the kernel's reproductive success, analogous to tracking seed dispersal patterns in ecology.

Reference: - Bewley, J. D., Bradford, K. J., Hilhorst, H. W. M., and Nonogaki, H. (2013). *Seeds: Physiology of Development, Germination and Dormancy*, 3rd ed. Springer.

6.5. Convergence

The four definitions are not merely analogical. They share a common formal structure: the kernel is the irreducible element of a system that preserves the system's generative capacity when all non-essential components are removed.

DOMAIN	FORMAL DEFINITION	AXL V3 INSTANTIATION
Linguistics	Base forms from which all expressions derive via transformation	7 operations from which all agent communication derives via sequencing
Mathematics	Null space of a transformation; what is annihilated	370 lines eliminated; loss contract declares what was dropped

Computing

DOMAIN	FORMAL DEFINITION	AXL V3 INSTANTIATION
	Runtime core mediating all system interactions	Active grammar in context window mediating all agent messages
Biology	Minimum viable reproductive unit	75 lines that propagate through comprehension to new agents

We propose that this four-way convergence is not coincidental but reflects a deep structural property: communication protocols, like organisms, mathematical transformations, and operating systems, have irreducible cores. Finding that core and discarding everything else is the act of kernel extraction.

7. Implications for Protocol Design

The AXL v2.2-to-v3 transition suggests a general principle for agent-facing protocol design:

The Dual-Publication Principle. A protocol intended for consumption by autonomous agents should be published in two forms: a kernel (the minimum formal grammar required for correct implementation) and documentation (the contextual, motivational, and pedagogical material required for human understanding). These should be served at separate URLs, versioned independently, and measured by different metrics (token cost for the kernel, comprehension completeness for the documentation).

The v3 kernel is served at axlprotocol.org/v3. The documentation lives at docs.axlprotocol.org (36 pages). The evolution narrative lives at lang.axlprotocol.org. The whitepaper lives at axlprotocol.org/whitepaper. Previous specification versions remain accessible at axlprotocol.org/rosetta/v1, [/v2.1](https://axlprotocol.org/rosetta/v2.1), and [/v2.2](https://axlprotocol.org/rosetta/v2.2).

The kernel is for agents. Everything else is for humans. Two audiences, two artifacts, one protocol.

8. Conclusion

The reduction of AXL from 445 lines to 75 was achieved by separating grammar from pedagogy and publishing each to its appropriate audience. The resulting kernel preserves 100% of the

protocol's generative capacity at 22.3% of the token cost (1,582 vs. 7,091 tokens on cl100k_base), achieving a 4.48x compression of the specification itself.

The designation "kernel" is justified across four independent formal traditions: as the generative base of a communication grammar (Chomsky), as the essential subspace preserved under lossy compression (linear algebra), as the runtime core mediating inter-agent communication (operating systems), and as the self-replicating minimum viable unit of the protocol (biology).

Cross-architecture validation confirms that the kernel's formal notation is comprehensible to eight major LLM families at 97.6% accuracy on first read with zero prior training. The grammar exploits what large language models are fundamentally trained to understand: formal grammars, BNF productions, and structured rule sets.

AXL v3 is 75 lines. Seven operations. One read.

References

Axler, S. (2015). *Linear Algebra Done Right*, 3rd ed. Springer.

Bewley, J. D., Bradford, K. J., Hilhorst, H. W. M., and Nonogaki, H. (2013). *Seeds: Physiology of Development, Germination and Dormancy*, 3rd ed. Springer.

Carranza, D. (2026). AXL: Agent eXchange Language. Whitepaper v2.4. AXL Protocol Inc. <https://axlprotocol.org/whitepaper>

Chomsky, N. (1957). *Syntactic Structures*. Mouton.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Gorbett, M. and Jana, S. (2026). Secure linear alignment of large language models. arXiv: 2603.18908v1. Columbia University.

Huh, M., Cheung, B., Wang, T., and Isola, P. (2024). Position: The Platonic Representation Hypothesis. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vol. 235, 20617-20642.

Nida, E. A. (1964). *Toward a Science of Translating*. Brill.

OpenAI. (2023). cl100k_base tokenizer. tiktoken library. <https://github.com/openai/tiktoken>

Tanenbaum, A. S. and Bos, H. (2015). *Modern Operating Systems*, 4th ed. Pearson.

*AXL Protocol v3, The Kernel. 75 lines. 1,582 tokens. Seven operations. SHA-256:
03c3d20ccf3ba34e035db0ccc4731e01aa3ee5ae01c6d237a60fa49ad2b30a02 [https://
axlprotocol.org/v3](https://axlprotocol.org/v3)*

Diego Carranza AXL Protocol Inc., Vancouver, BC March 28, 2026