

RSP-M/CNX: A Governed Loop-Context Compiler for Delta-Propagated Agentic Workflows

A Machine-Readable Reference Scaffold for Structural Validation, Custody, and
Authority Separation

Ivan Silva

July 11, 2026

Publication record

Author	Ivan Silva
Affiliation	Carlonoscopen, LLC
ORCID	https://orcid.org/0009-0005-2284-8891
Publication context	Carlonoscopen Journal of Coherence Intelligence (CJCI)
Issue	Volume 1, Issue 18
CJCI ISSN	Digital 3069-874X; Print 3071-0022
Archive target	Zenodo archival deposit
Reserved DOI	10.5281/zenodo.21310639
Version	1.2 final author-review candidate
Date	July 11, 2026
License	Creative Commons Attribution 4.0 International (CC BY 4.0), paper text only
Document status	Controlled manuscript - final rendering-corrected author-review candidate

© 2026 Ivan Silva / Carlonoscopen, LLC. The paper text is licensed under CC BY 4.0. Software, schemas, procedure artifacts, sealed packages, and proprietary implementation lineage are not licensed by implication and remain subject to separate explicit terms.

Contents

Abstract	3
1. Introduction	3
2. Related work and positioning	4
2.1 Long-context execution and serving efficiency	4
2.2 Agent loops, memory, and interfaces	4
2.3 Energy reporting	4
3. Design requirements	4
4. RSP-M/CNX method	5
4.1 Semantic bindings	5
4.2 Literal islands	5
4.3 Loop macros	5
4.4 Delta packets	6
4.5 SCL-style run-in-mind validation	6
4.6 CNX authority separation	6
4.7 Sealing and custody	6
5. Receiver coupling and interface metabolization	7
6. Implementation and lineage	7
6.1 v0.3 procedure kernel	7
6.2 Phase 0A	7
6.3 Phase 0B	7
6.4 Phase 0B.1 and 0B.2	7
7. Evaluation	8
7.1 Artifact identity and environment	8
7.2 Functional and refusal tests	8
7.3 Custody-idempotence result	8
7.4 Context-size analysis	8
8. Discussion	9
9. Threats to validity and limitations	9
10. Phase 0C: Receiver Coupling Profile and pilot measurement	10
11. Conclusion	10
Artifact and publication availability	10
Author contribution	11
Funding	11
Conflict of interest	11
Ethical and governance statement	11
Licensing	11
Recommended citation	11
References	11

Abstract

Repeated language-model agent workflows commonly retransmit stable project doctrine, constraints, validation rules, and prior reports even when the next task is only a small change. This repeated context propagation consumes tokens and increases the opportunity for semantic drift, stale-state reuse, and authority confusion. This paper introduces **RSP-M/CNX**, a governed loop-context compiler and local-first reference scaffold for transforming dense Request Specification Protocols into reusable machine-readable structures.

RSP-M, Regenerable Structural Prompt Metabolization, represents stable meaning as semantic bindings, preserves exact obligations as literal islands, encodes repeated procedures as loop macros, and communicates current work through bounded delta packets. Structural Calculus Language (SCL)-style run-in-mind validation checks packet admissibility before execution. Coherence Nexus (CNX)-style policy separates context optimization from authority: a valid or sealed packet may describe an action but does not authorize it.

The Phase 0B.2 artifact implements JSON/YAML schemas, local command-line validation, packet sealing, strict manifests, compatibility adapters, machine-checked literal-island hashing, release profiles, pilot templates, and custody-idempotent tests. Independent local validation reproduced 85/85 scaffold tests and 28/28 hardened-procedure tests, with the hardened manifest passing both before and after the test suite. In one documented development workflow, the full source context contained 43,728 characters, the metabolized handoff 2,927 characters, and the example delta packet 1,761 characters, corresponding to a 9.3x full-context-to-handoff-plus-delta ratio and a 14.9x full-context-to-handoff-only ratio. These are artifact-size results, not provider token, billing, latency, or energy measurements.

The contribution is a methods and infrastructure scaffold for moving repeated agent work from prose-context retransmission toward sealed, auditable, receiver-coupled delta propagation while preserving source precedence and authority separation.

Keywords: agentic AI; context compilation; delta propagation; prompt infrastructure; authority separation; structural validation; reproducibility; AI governance; software agents

1. Introduction

Language-model agents increasingly interact with repositories, command shells, tests, APIs, and other tools in iterative workflows. ReAct interleaves reasoning and action [4]; Reflexion carries linguistic feedback across attempts [5]; AutoGen supports configurable multi-agent conversations [6]; and SWE-agent shows that an agent-computer interface can materially shape software-agent behavior [7]. These systems make repeated execution loops practical, but they also expose a less studied infrastructure problem: stable instructions are often retransmitted and reinterpreted on every loop.

A software-development agent may repeatedly receive the same project purpose, repository boundaries, acceptance criteria, forbidden actions, test commands, reporting format, and governance rules. The actual new task may be only a failing test, a bounded patch, or a release decision. Nevertheless, the executor must recover the structure of the full instruction field again.

This problem matters for both correctness and efficiency. Transformer attention creates substantial sequence-length and memory pressure [1], motivating IO-aware attention algorithms [2] and serving systems that improve key-value-cache management [3]. These advances optimize model execution and serving. RSP-M/CNX addresses a different layer: **what stable workflow context should be propagated to the model at all.**

The central hypothesis is that a dense, program-like instruction should be compiled once into

reusable structures, after which subsequent loops should carry only the stable references and the current verified delta. The method must preserve exact obligations and remain auditable. It must also prevent a dangerous category error: compression cannot become authorization.

This paper makes four contributions:

1. a machine-readable representation of stable meaning, exact obligations, repeated procedures, and bounded deltas;
2. a structural pre-execution validation and hash-bound custody procedure;
3. an explicit CNX-style separation between efficiency and authority;
4. a validated local reference scaffold and an artifact-level context-size analysis.

Receiver Coupling Profile (RCP) is introduced as a design extension for Phase 0C. It formalizes the idea that the handoff should match the receiving executor's capacity and failure profile. RCP is not claimed as an implemented Phase 0B.2 component.

2. Related work and positioning

2.1 Long-context execution and serving efficiency

The Transformer made attention central to modern language models [1]. Standard self-attention has quadratic time and memory complexity in sequence length, and long sequences impose significant memory traffic. FlashAttention improves exact-attention efficiency by reducing high-bandwidth-memory reads and writes [2]. PagedAttention and vLLM address serving-side key-value-cache fragmentation and sharing [3].

RSP-M/CNX is complementary to these techniques. It neither changes attention kernels nor manages GPU memory. It attempts to reduce unnecessary repeated input at the orchestration boundary by separating stable workflow state from the active delta.

2.2 Agent loops, memory, and interfaces

ReAct combines reasoning traces with external actions [4]. Reflexion stores linguistic feedback in episodic memory to improve subsequent trials [5]. AutoGen treats agent interaction patterns as programmable infrastructure [6]. SWE-agent demonstrates that specialized interfaces can improve an agent's ability to operate on software repositories [7].

These works support two premises of the present method: iterative agents depend on retained state, and the interface presented to the agent can materially affect execution. RSP-M/CNX focuses on compiling and governing the persistent instruction layer rather than proposing a new reasoning algorithm.

2.3 Energy reporting

Energy and carbon effects in machine learning require explicit measurement boundaries and standardized reporting [8]. Inference energy can vary substantially by task, model, and deployment configuration [9]. For that reason, this paper does not infer energy savings directly from character reduction. It defines a pilot-ready measurement hypothesis and reserves energy claims for runs with named telemetry.

3. Design requirements

RSP-M/CNX was developed under six requirements.

Source precedence. A compact representation must not supersede the authoritative source RSP. Conflicts resolve toward the higher-precedence source, not toward model convenience.

Exact-obligation preservation. Formulas, acceptance criteria, prohibited actions, and other load-bearing text require stronger preservation than ordinary summarization.

Bounded delta propagation. Each new loop should identify the current issue, allowed scope, forbidden actions, escalation conditions, and required output.

Structural admissibility. A packet should be rejected before execution when references are unresolved, authority is invalid, scope is absent, or source hashes cannot support the claimed status.

Authority separation. Context optimization may reduce repetition, but it may not grant operational authority or permit self-certification.

Reproducible custody. Manifests, seals, and tests should detect missing, changed, extra, or post-test-mutated files.

4. RSP-M/CNX method

4.1 Semantic bindings

A semantic binding is a stable, named representation of a high-density instruction. It includes a canonical meaning, source references, scope, authority status, required expansion terms, and the risk created by omission.

For example, @NO_OVERCLAIM can bind the rule that cost and energy claims remain engineering estimates until measured with appropriate telemetry. A later packet can reference this binding rather than retransmit the entire doctrine.

Bindings are not free-form aliases. They are registry objects with controlled names, machine validation, and source links.

4.2 Literal islands

A literal island is an exact obligation anchor. Its content may be a verbatim excerpt or a normalized exact-value representation whose normalization preserves value, scope, and obligation. Each island has a machine-readable identifier and a content hash.

Typical literal islands include:

- acceptance criteria;
- forbidden shortcuts;
- formulas and threshold values;
- required file trees;
- validation commands;
- final reporting contracts.

Delta packets reference literal islands by identifier rather than by informal prose label.

4.3 Loop macros

A loop macro names a repeated procedure. Primitive loops represent operations such as reference loading, structural projection, policy checking, action, validation, fallback, and reporting. Composite macros bind an ordered sequence of primitives for tasks such as fixing tests, running a matrix, reviewing failure surfaces, or preparing a release.

Every macro declares allowed actions, blocked actions, and required outputs. Unknown macro references fail closed.

4.4 Delta packets

The delta packet is the bounded unit of subsequent work. A product-schema packet contains, at minimum:

- packet identifier and creator class;
- authority level;
- source, binding, literal-island, macro, prior-report, and packet hashes;
- active macro;
- stable binding references;
- selected literal-island references;
- current issue and rationale;
- allowed files and forbidden actions;
- escalation conditions;
- required outputs.

A model-created packet cannot raise its own authority. `bounded_action` requires an explicit scope and a valid packet seal. Missing source artifacts are reported as `hash_unverifiable`, never as `hash_valid`.

4.5 SCL-style run-in-mind validation

Structural Calculus Language is used here as a pre-execution admissibility discipline. The Phase 0B.2 validator checks that the active macro exists, all bindings and literal islands resolve, authority is allowed, the packet does not self-elevate, bounded action identifies allowed files, forbidden actions and escalation conditions are present, hashes have valid statuses, and source precedence is preserved.

This layer is intentionally narrower than semantic proof. It checks declared structure and refuses structurally unsafe packets before an agent acts.

4.6 CNX authority separation

Coherence Nexus supplies the authority doctrine:

RSP-M optimizes context handling. CNX governs action.

A sealed packet establishes integrity of a declared packet; it does not establish permission, correctness, or production acceptance. CNX policy determines whether an output remains review-only, advisory, bounded action, or blocked. Human approval remains required for production action, registry mutation, scope expansion, and claim upgrades.

4.7 Sealing and custody

Phase 0B.2 uses canonical serialization for packet sealing, strict manifests that detect missing, changed, extra, and malformed entries, and deterministic literal-island hashing. The product adapter preserves the invariant that conversion may narrow authority but may not widen it.

5. Receiver coupling and interface metabolization

The development process produced a further design insight: source intent should be transformed to the operational capacity of the receiver rather than written for a named model alone.

A **Receiver Coupling Profile** is proposed as a substrate-agnostic description of:

- context capacity;
- preferred execution format;
- ambiguity tolerance;
- schema and tool aptitude;
- validation capability;
- authority sensitivity;
- likely failure modes.

The same source intent may therefore produce different handoffs for a frontier coding agent, a small local model, a human contractor, or a deterministic CLI, while preserving the same source precedence and authority boundary.

Formative executor feedback supported this interface-level interpretation. The receiving model reported that the metabolized RSP declared stable bindings and execution order up front, enumerated escalation and forbidden moves, produced near-zero clarification pressure, and reduced doctrine re-derivation. This is qualitative feedback from one development loop, not a controlled performance result. Phase 0C will formalize RCP and compare receiver-matched handoffs experimentally.

6. Implementation and lineage

6.1 v0.3 procedure kernel

The frozen RSP-M v0.3 lineage established the procedure kernel: bindings, literal islands, macros, packets, state, handoff, round-trip artifacts, aliases, tools, tests, manifests, and custody records. The canonical implementation remains proprietary to Carlonoscope, LLC and is referenced publicly only under the approved IP posture.

6.2 Phase 0A

Phase 0A productized the procedure as a local Python scaffold with JSON/YAML schemas, CLI validation, examples, tests, documentation, reports, and an artifact manifest.

6.3 Phase 0B

Phase 0B added packet sealing, strict manifest verification, a conservative v0.3 compatibility adapter, machine-checked literal-island hashing, a public-release profile, and pilot-measurement templates.

6.4 Phase 0B.1 and 0B.2

Phase 0B.1 corrected publication wording. Independent review then found that one hardened test executed the round-trip generator in place, rewriting four manifest-bound files because the generated drift score included a timestamp. Phase 0B.2 moved generator tests to temporary copies, added a session-scoped manifest snapshot guard, and added an explicit custody-idempotence test.

The resulting requirement is stronger than “tests pass”: tests must pass without dirtying the custody state.

7. Evaluation

7.1 Artifact identity and environment

The validated internal artifact was:

- **File:** RSPM_CNX_Infrastructure_Phase0B_2.zip
- **Release manifest:** 125 bound files
- **Local environment:** Windows PowerShell 5.1 with Python 3.12 dependencies installed in an isolated environment

SHA-256 (one continuous 64-character value; the print line break is not part of the hash):

```
cd8f36ecd734ab1faf3784eef5f007fe
39515182cb3edb3d486000eaa5734a35
```

The public release of implementation files remains governed by the public-safe profile described in the accompanying IP statement.

7.2 Functional and refusal tests

The Phase 0B scaffold passed 85/85 tests. The hardened procedure copy passed 28/28 tests. The suites cover schema validation, reference resolution, authority-level validation, model self-elevation refusal, bounded-action sealing, strict manifest behavior, v0.3 compatibility, literal-island hashing, hash-status distinctions, public-release profile behavior, pilot-schema validation, and custody idempotence.

The local CLI successfully validated bindings, literal islands, macros, delta packets, and authority policy; ran the structural pre-execution check; estimated context reduction; sealed a packet; verified hash statuses; computed the literal-island aggregate hash; and performed strict manifest verification.

7.3 Custody-idempotence result

The decisive Phase 0B.2 sequence was:

1. verify the hardened manifest - PASS;
2. run 28 hardened tests - PASS;
3. verify the hardened manifest again - PASS.

This sequence directly addresses the mutation defect found in Phase 0B.1.

7.4 Context-size analysis

Let:

- $F = 43,728$ characters in the full RSP context;
- $H = 2,927$ characters in the metabolized handoff;
- $D = 1,761$ characters in the example delta packet.

The immediate handoff-plus-delta ratio is:

$$R_{HD} = \frac{F}{H + D} = \frac{43,728}{2,927 + 1,761} \approx 9.33.$$

The handoff-only ratio is:

$$R_H = \frac{F}{H} = \frac{43,728}{2,927} \approx 14.94.$$

These ratios are reproducible character-count results. A rough characters-per-token heuristic produces a similar ratio but does not constitute tokenizer-specific measurement. The previously reported 15-25x cached steady-state range is retained only as a future hypothesis dependent on provider caching and workload structure.

8. Discussion

RSP-M/CNX is not ordinary prompt summarization. A summary usually optimizes brevity. RSP-M optimizes regenerability, exact-obligation preservation, bounded execution, and auditability.

The method also occupies a different layer from attention-kernel and serving optimizations. FlashAttention and PagedAttention reduce the cost of processing a sequence [2,3]. RSP-M asks whether stable parts of that sequence need to be resent and reinterpreted at all.

For small and medium-sized businesses, the near-term value proposition is controlled reduction of repeated agent-loop context, fewer ambiguous handoffs, and improved auditability. For enterprises, the same scaffold can support project registries, authority policies, hash-bound packets, and pilot instrumentation. For providers, the long-term hypothesis is front-end compilation and caching of stable context with only selected literal islands and active deltas routed to inference.

No economic claim should precede pilot evidence. The correct next step is a controlled comparison between a full-context arm and an RSP-M delta arm using the same task, receiver, model configuration, and acceptance criteria.

9. Threats to validity and limitations

Single-workflow measurement. The context-size result comes from one documented development lineage. Different RSPs may compress more or less effectively.

Character counts. The reported ratios are based on characters, not provider token accounting. Tokenizers, hidden prompts, tool transcripts, and cache behavior vary.

No controlled receiver comparison. Qualitative executor feedback is formative evidence only. There was no randomized or repeated A/B study.

No provider telemetry. Billing, latency, GPU utilization, power, and facility energy were not measured.

Reference scaffold, not deployment. Phase 0B.2 is local-first and pilot-ready, not a production SaaS or provider-integrated service.

No formal semantic equivalence. Round-trip checks, hashes, and manifests improve custody and drift detection but do not prove semantic equivalence for arbitrary source instructions.

IP boundary. The internal validation ZIP contains proprietary-lineage material. A public-safe derivative or hash-only reference is required for open deposition.

10. Phase 0C: Receiver Coupling Profile and pilot measurement

Phase 0C should implement a Receiver Coupling Profile schema and a repeatable pilot protocol. At minimum, the experiment should record:

- receiver identity and configuration;
- model and context limits;
- source context size;
- compiled handoff size;
- delta size;
- number of loops;
- provider input and output tokens where available;
- wall-clock latency;
- API or infrastructure cost where available;
- acceptance quality and rework count;
- hardware energy only when a named telemetry source is available.

The primary comparison should hold task, source doctrine, acceptance criteria, and receiver configuration constant while varying only the handoff method.

11. Conclusion

RSP-M/CNX provides a local-first reference scaffold for governed loop-context compilation. Stable meaning becomes semantic bindings; exact obligations become literal islands; repeated procedures become macros; current work becomes sealed delta packets; SCL-style checks reject structurally inadmissible packets; and CNX-style policy keeps authority separate from model output.

Phase 0B.2 demonstrates that this method can be made machine-readable, testable, hash-bound, compatibility-aware, and custody-idempotent. In one development workflow it reduced the transmitted character footprint by approximately 9.3x for handoff plus delta and 14.9x for handoff alone. These results justify controlled pilots, not production energy claims.

The broader contribution is a shift from repeated prose-context retransmission toward receiver-coupled, auditable delta propagation.

Artifact and publication availability

This article is prepared for publication in the *Carlonosopen Journal of Coherence Intelligence*, Volume 1, Issue 18, and subsequent Zenodo archival deposit under the reserved DOI [10.5281/zenodo.21310639](https://doi.org/10.5281/zenodo.21310639). Public distribution of implementation files is governed by the accompanying public/IP posture and must exclude or hash-reference proprietary canonical material unless separately approved.

The validated internal Phase 0B.2 artifact is identified as follows:

```
File: RSPM_CNX_Infrastructure_Phase0B_2.zip  
SHA-256: cd8f36ecd734ab1faf3784eef5f007fe  
39515182cb3edb3d486000eaa5734a35
```

The two displayed SHA-256 lines form one continuous value; no spaces or line break are part of the hash.

Author contribution

Ivan Silva conceived the RSP-M/CNX method, directed the iterative development and validation program, performed the local acceptance runs, defined the authority and no-overclaim boundaries, and authored the publication. Frontier-model assistants were used as bounded drafting, implementation, and review tools under human direction; they are not listed as authors.

Funding

No external funding was declared for this publication.

Conflict of interest

The author is associated with Carlonosopen, LLC, which may develop products or services based on the methods described. This commercial interest is disclosed so that readers can evaluate the work accordingly.

Ethical and governance statement

The method is designed to preserve human and policy authority over agent actions. It does not authorize autonomous production decisions, personnel decisions, access denial, or other consequential actions without an appropriate external authority process.

Licensing

The paper text is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). Software, schemas, procedure artifacts, sealed packages, and proprietary implementation lineage are excluded from that grant unless a separate artifact license expressly states otherwise.

Recommended citation

Silva, I. (2026). *RSP-M/CNX: A Governed Loop-Context Compiler for Delta-Propagated Agentic Workflows*. *Carlonosopen Journal of Coherence Intelligence*, 1(18), Version 1.2. Zenodo archival deposit: [10.5281/zenodo.21310639](https://zenodo.org/record/21310639).

References

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30. arXiv:1706.03762.
- [2] Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Re, C. (2022). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *Advances in Neural Information Processing Systems*, 35. arXiv:2205.14135.
- [3] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. (2023). Efficient Memory Management for Large Language Model Serving

with PagedAttention. *Proceedings of the 29th Symposium on Operating Systems Principles*. <https://doi.org/10.1145/3600006.3613165>

[4] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *International Conference on Learning Representations*. arXiv:2210.03629.

[5] Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. *Advances in Neural Information Processing Systems*, 36. arXiv:2303.11366.

[6] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv:2308.08155.

[7] Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. (2024). SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. *Advances in Neural Information Processing Systems*, 37. arXiv:2405.15793.

[8] Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. (2020). Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *Journal of Machine Learning Research*, 21(248), 1-43.

[9] Luccioni, A. S., Jernite, Y., and Strubell, E. (2024). Power Hungry Processing: Watts Driving the Cost of AI Deployment? *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. <https://doi.org/10.1145/3630106.3658542>