G²CP: A Graph-Grounded Communication Protocol for Verifiable and Efficient Multi-Agent Reasoning

Extended Abstract

Karim Ben Khaled
Université de Lorraine, LORIA
Nancy, France
karim.ben-khaled@univ-lorraine.fr

Davy Monticolo
Université de Lorraine, LORIA
Nancy, France
davy.monticolo@univ-lorraine.fr

ABSTRACT

Multi-agent LLM systems decompose complex tasks across specialized agents, yet their reliance on natural language for inter-agent communication introduces semantic ambiguity, token inefficiency, and untraceable reasoning chains problems that become critical in safety-sensitive industrial deployments. We introduce **G²CP** (Graph-Grounded Communication Protocol), a formal agent communication language in which every inter-agent message is a typed graph operation traversal or update over a shared knowledge graph. Each performative creates a verifiable social commitment grounded in Singh's commitment semantics, enabling deterministic replay of any agent conclusion. On an industrial maintenance knowledge graph (367 nodes, 538 edges) evaluated across 521 queries, G²CP achieves 0.90 task accuracy (+21% over the strongest baseline), reduces token cost by 73%, and drives hallucination to 0.02 while maintaining perfect auditability. Code and data are publicly available.[1] The full paper is available on arXiv.[2]

KEYWORDS

Multi-Agent Systems; Knowledge Graphs; Agent Communication Languages; LLM Grounding; Verifiable AI

ACM Reference Format:
Karim Ben Khaled and Davy Monticolo. 2026. G²CP: A Graph-Grounded Communication Protocol for Verifiable and Efficient Multi-Agent Reasoning: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/JHFW8307

1 INTRODUCTION AND MOTIVATION

Multi-agent architectures built on large language models have emerged as a dominant paradigm for complex reasoning [3, 14, 16]. Systems such as AutoGen [17], MetaGPT [5], CAMEL [8], and Generative Agents [10] decompose problems across role-specialized agents that collaborate through natural language exchanges, achieving impressive results on code generation, scientific reasoning, and social simulation. However, deployment in high-stakes industrial settings reveals three structural weaknesses.

(W1) Semantic ambiguity. When Agent A instructs Agent B to "investigate bearing wear patterns on the hydraulic press," the scope, graph neighborhood, and expected output format are all left implicit. The receiving agent must reconstruct intent through its own LLM, introducing a lossy channel where meaning is interpreted rather than transmitted leading to an average 23% disagreement rate in our experiments.

(W2) Token inefficiency. A typed graph traversal command requires approximately 20 tokens; its natural language equivalent consumes 80–120 tokens. Across multi-turn agent dialogues, this overhead compounds multiplicatively with agent count.

(W3) Audit opacity. Tracing errors in multi-agent outputs requires parsing unstructured text logs with no formal mechanism to verify grounding a well-documented challenge in LLM hallucination research [6] and a deployment blocker in regulated industries.

We propose **G²CP** to address all three weaknesses by replacing natural language inter-agent messages with typed graph operations over a shared knowledge graph $\mathcal{G}$ [4]. Every message carries a deterministic operation whose execution can be replayed and verified. LLMs are confined to the system boundary translating user queries into graph operations and results back into natural language while inter-agent coordination proceeds exclusively through G²CP, ensuring a fully deterministic coordination layer free from generative hallucination.

2 THE G²CP PROTOCOL

Message formalism. A G²CP message is a five-tuple $m = \langle s, r, \pi, op, c \rangle$ where s is the sender, r the receiver, $\pi \in \Pi$ a performative, op a typed graph operation, and c a conversation context with shared focus subgraphs. The performative set $\Pi = \{\textsc{Request}, \textsc{Inform}, \textsc{Query}, \textsc{Propose}, \textsc{Confirm}, \textsc{Reject}, \textsc{Update}\}$ extends classical speech-act theory and FIPA-ACL [2] by *parameterizing each performative with a graph operation* rather than a free-text content field, ensuring machine-verifiable interpretation.

Graph operations. The Traverse operation $T(V_s, \Psi_f, h, ret) \rightarrow 2^{V \times E}$ performs recursive neighborhood expansion from source nodes V_s, filtering edges by type set Ψ_f for h hops, returning results as $ret \in \{\text{subgraph}, \text{paths}, \text{leaves}\}$. The frontier expands as $N_k(V_s) = N_{k-1}(V_s) \cup \{v' \mid \exists\, e = (v, v') \in E,\ \text{type}(e) \in \Psi_f,\ v \in N_{k-1}(V_s)\}$ with $N_0 = V_s$. The Update operation applies a validated graph delta $\Delta\mathcal{G} = (\Delta V^+, \Delta V^-, \Delta E^+, \Delta E^-)$ under schema constraints and RBAC authorization. Both produce typed `SubgraphResult` objects with provenance metadata, guaranteeing that all downstream reasoning is grounded in verifiable graph evidence.

[2] https://github.com/karim0bkh/G2CP_AAMAS
[2] Full paper: https://arxiv.org/abs/2602.13370

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/JHFW8307

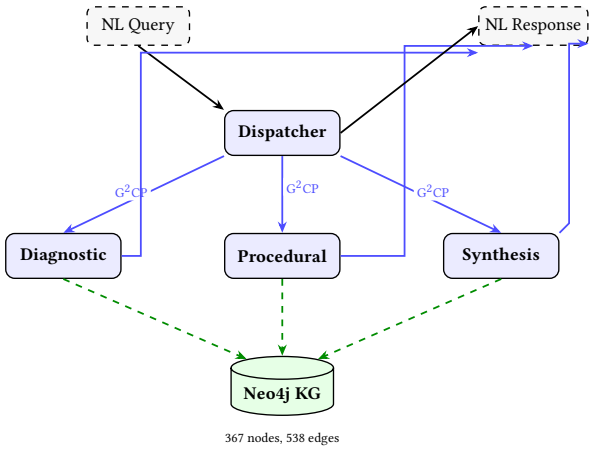

Figure 1: G²CP architecture. Blue arrows denote G²CP messages (typed graph operations); dashed green arrows are direct graph queries. Natural language is used only at the user-facing boundary.

Commitment semantics. Following Singh's social commitment framework [13], each performative creates a commitment $C(\textit{debtor}, \textit{creditor}, \textit{condition})$ with lifecycle: *created* → *active* → *fulfilled/violated*. A REQUEST$(s, r, T(\cdot))$ creates $C(r, s, \textit{execute\_and\_return}(T))$ binding the receiver to execute the traversal and return graph-grounded results. CONFIRM fulfills with verifiable evidence; REJECT marks a constraint violation grounded in graph state. This enables post-hoc auditing by replaying the full commitment chain.

LLM boundary isolation. The LLM participates only at the system boundary: (1) entity extraction, (2) intent classification into {diagnostic, procedural, predictive, factoid}, (3) traversal depth estimation, and (4) response generation from aggregated results. Unlike tool-augmented LLM approaches [11] where the LLM decides when and how to invoke tools at each step, G²CP confines all LLM calls to the boundary, making inter-agent coordination fully deterministic and reproducible.

3 SYSTEM ARCHITECTURE

Figure 1 shows five specialized agents coordinating over a Neo4j knowledge graph [4]. The **Dispatcher** performs entity extraction, links mentions to graph UIDs via sentence-transformer embeddings (cosine ≥ 0.85), and routes typed REQUEST messages. The **Diagnostic Agent** traverses `causes`, `indicates`, and `correlates_with` edges for symptom→fault causal analysis, ranking candidates by edge weight and connectivity. The **Procedural Agent** maps faults to repair procedures via `addressed_by`, `requires`, and `has_safety_protocol` edges. The **Synthesis Agent** discovers cross-domain patterns through historical work orders (`occurred_in`, `failed_after`) and proposes graph updates. The **Ingestion Agent** validates expert-submitted updates, enforcing schema constraints and re-indexing affected embeddings.

Each agent operates under RBAC permissions over specific node/edge types. All messages are HMAC-SHA256 signed and logged to an append-only audit trail for full trace replay.

Table 1: Results on 521 queries. Best in bold. ↑ = higher better; ↓ = lower better.

Metric	FTMA	JSMA	Single	G²CP
Accuracy (F1) ↑	0.67	0.74	0.71	**0.90**
Tokens/query ↓	2,847	2,134	1,456	**768**
Hallucin. rate ↓	0.23	0.18	0.14	**0.02**
Cascading err. ↓	0.31	0.19	0.00	**0.00**
Auditability ↑	0.42	0.68	1.00	**1.00**

4 EXPERIMENTAL EVALUATION

Setup. We compare G²CP against three baselines: **FTMA** (Free-Text Multi-Agent, AutoGen-style [17]); **JSMA** (JSON-Structured Multi-Agent with typed schemas but no graph grounding); and **Single-Agent RAG** with vector retrieval [7]. All use GPT-4 over the identical knowledge graph. The corpus includes 500 synthetic queries across five categories with programmatic ground truth, plus 21 real-world cases validated by domain experts.

Results. Table 1 reports the main findings. G²CP achieves 0.90 F1, a +21% improvement over JSMA (0.74), because every agent response must reference nodes and edges actually retrieved from $\mathcal{G}$ the system structurally cannot fabricate entities. Token consumption drops 73% versus FTMA (2,847→768) since a TRAVERSE command occupies 15–25 tokens versus 80–120 for free-text prompts. Hallucination falls to 0.02; the residual stems from boundary entity linking errors, not inter-agent failures consistent with findings that grounding mechanisms are the most effective mitigation for LLM hallucination [6]. Cascading errors are eliminated while retaining multi-agent decomposition benefits.

Ablation and expert validation. Entity linking contributes +0.14 F1; removing structured messages increases token cost by 2.3×; commitment tracking enables perfect auditability. On 21 expert-validated industrial cases (mean rating 4.6/5.0), G²CP identifies root causes in 19/21 and retrieves safety-enriched repair procedures a multi-hop capability baselines miss because safety nodes require deliberate traversal specification.

5 RELATED WORK AND CONCLUSION

Multi-agent LLM frameworks AutoGen [17], MetaGPT [5], CAMEL [8], Generative Agents [10] rely on natural language as the inter-agent medium, inheriting ambiguity and verification challenges. Chain-of-thought [15] and ReAct [18] improve single-agent reasoning but do not address multi-agent coordination fidelity. FIPA-ACL [2] and Singh's commitments [13] provide our theoretical foundations, though classical ACLs predate LLMs and lack graph-grounded semantics. LLM-KG surveys [9], Graph RAG [1], Reflexion [12], and tool-augmented approaches [11] focus on single-agent retrieval or retain the LLM as reasoning backbone, unlike G²CP's strict boundary isolation.

G²CP uniquely bridges commitment-based ACL semantics with typed knowledge graph operations, yielding simultaneous gains in accuracy (+21%), efficiency (−73% tokens), and verifiability (0.02 hallucination). The protocol is domain-agnostic. Future work targets dynamic agent spawning, federated G²CP across distributed graphs, and formal verification via model checking.

ACKNOWLEDGMENTS

This work is supported by Université de Lorraine and conducted within the LORIA laboratory and the startup Cognivance (Cognivance.io). We thank the industrial partners who provided expert validation.

REFERENCES

[1] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130* (2024).

[2] Foundation for Intelligent Physical Agents. 2002. FIPA ACL Message Structure Specification. In *FIPA Standard Specifications*. Document SC00061G.

[3] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model Based Multi-Agents: A Survey of Progress and Challenges. *arXiv preprint arXiv:2402.01680* (2024).

[4] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge Graphs. *Comput. Surveys* 54, 4 (2021), 1–37.

[5] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2024. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[6] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[7] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 9459–9474.

[8] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

[9] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3580–3599.

[10] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th ACM Symposium on User Interface Software and Technology (UIST)*.

[11] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

[12] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

[13] Munindar P Singh. 1998. Agent Communication Languages: Rethinking the Principles. *IEEE Computer* 31, 12 (1998), 40–47.

[14] Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents. *arXiv preprint arXiv:2306.03314* (2023).

[15] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35.

[16] Michael Wooldridge. 2009. *An Introduction to MultiAgent Systems* (2nd ed.). John Wiley & Sons.

[17] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[18] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.