Disruption Opportunity
DARPA-PA-24-04-01
Human-AI Communication for Deontic Reasoning Devops (CODORD)

I. Opportunity Description

The Defense Advanced Research Projects Agency (DARPA) Defense Sciences Office (DSO) is issuing a Disruption Opportunity (DO), inviting submissions of innovative basic or applied research concepts in the technical domain of knowledge authoring for deontic logical reasoning. This DO is issued under the Program Announcement for Disruptioneering, DARPA-PA-24-04. All awards will be made in the form of an Other Transaction (OT) for Prototype project. The total award value for the combined Phase 1 base and Phase 2 option is limited to $2,000,000. This total award value includes Government funding and performer cost share if required or proposed.

To view the original DARPA Program Announcement (PA) for Disruptioneering, visit SAM.gov under solicitation number DARPA-PA-24-04:
https://sam.gov/opp/cb7a935d59bb4ceeb62b9515f7d9f9b0/view

A. Introduction

The CODORD program will create new AI-based techniques for humans to author knowledge about deontics (obligations, permissions, and prohibitions) into an expressively flexible logical language (that has strong capabilities for automated reasoning), in an automated rather than manual fashion and using natural language (NL, e.g., English) rather than only logical language. CODORD has the potential to enable greater practicality of automated deontic reasoning with high assurance (verifiability/explainability and correctness), which are required for compliance with command orders, regulations, laws, operational policies, ethics, contracts, agreements, and strategies/plans.

Currently, the high cost of knowledge authoring (KA) severely inhibits adopting state-of-the-art (SOA) automated deontic reasoning techniques into practical operational deployment. KA is the skilled human effort required to get knowledge into logical language in the machine. The current high cost of deontic KA is due to the need for manual authoring by humans with logician expertise, i.e., "knowledge engineer" (KE) skill and experience, in highly expressive logical language and associated automated logical reasoning tools.

New KA techniques, created in CODORD, could lower deontic KA's cost and thereby make automated deontic reasoning a much more widely practical tool for decision support across a variety of crucial military and commercial applications. Operations planning; autonomous systems; supply chain, contracts, and financial; health treatment guidance; confidentiality and transaction authorization; systems integration, modeling, simulation, and wargaming; and national intelligence analysis are all areas for which automated deontic reasoning could be valuable for decision support.

CODORD's focus is on developing new AI-based techniques for deontic KA, rather than on developing methods for automated deontic reasoning in the logical language itself. These new KA techniques will enable humans with KE skill to author deontic KA in a new manner and will enable humans without KE skill to do much of the KA that previously required KE skill.

Current machine learning (ML)-based methods such as large language models (LLMs) are fundamentally limited in their ability to reach a high degree of assurance in reasoning[1]. However, they have outstanding strengths in natural language processing (NLP) that appear quite relevant to CODORD's KA challenge. There is the potential for CODORD to leverage the strengths of both ML and logical AI approaches. ML-based AI techniques have the potential to more cost-efficiently generate high-expressiveness logical knowledge starting from NL. Logical reasoning tools can take that logical knowledge as input, and then perform automated deontic reasoning to output conclusions (notably, answers to queries) suitable for decision support.

SOA techniques for automated deontic reasoning are based on logical languages that can expressively represent (a) multiple modalities (such as belief, intention, obligation, and permission) of multiple agents (e.g., via the HiLog[2] technique for higher-order syntax[3]) and (b) defeasibility, a.k.a., argumentation, including negation (both strong/explicit and weak/default) and prioritized conflict handling (e.g., via the argumentation theories[4,5] technique). Defeasibility/argumentation enables the clear, modular expression of rules together with their exceptions.

Such logical languages typically extend the expressiveness of declarative logic programs and include Rulelog,[6,7,8] Answer Set Programs (ASP), and some AI logical "argumentation systems." Rulelog is based on the well-founded semantics for logic programs. ASP is based on the stable semantics for logic programs, which has less attractive worst-case computational time complexity than does the well-founded semantics.

Each CODORD performer will choose and use an open-source logical language that must meet four practical requirements:

1. Sufficient expressiveness for (a) and (b) above (i.e., multiple modalities/agents and defeasibility), with conciseness.
2. Computational scalability, including (i) tractability (i.e., worst-case polynomial time complexity) and (ii) dependency-aware incremental updating[9]. Such tractability must be

---

[1] Zhang, Honghua, et al. On the paradox of learning to reason from data. *Proc. Thirty-Second Intl. Joint Conf. on Artificial Intelligence*. 2023.
[2] Chen, Weidong, et al, HiLog: A Foundation for Higher-Order Logic Programming, *J. Logic Programming* 15(3):187-230, 1993.
[3] Genesereth, Michael, et al, *Logical Foundations of Artificial Intelligence.*, 1987; see esp. the chapter on syntactic theories of belief.
[4] Wan, Hui, et al. Logic Programming with Defaults and Argumentation Theories, *Proc. 28th Intl. Conf. on Logic Programming*, 2009.
[5] Wan, Hui, et al. Defeasibility in Answer Set Programs via Argumentation Theories, *Semantic Web* 6(1): 81-98, 2015.
[6] Andersen, Carl, et al, Advanced Knowledge Base Debugging for Rulelog, Proc. 7th Intl. Web Rule Symposium (RuleML-2013), 2013.
[7] Grosof, Benjamin, et al. Ergo: A Quest for Declarativity in Logic Programming. In: (Warren, David S., et al, eds.) *Prolog: The Next 50 Years*. Springer, 2022.
[8] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI; also see: Swift, Theresa, et al, XSB software & manuals, https://xsb.sourceforget.net
[9] Swift, Theresa. Incremental Tabling in Support of Knowledge Representation and Reasoning, *Theory and Practice of Logic Programming* 14(4-5):553-567, 2014.

without significant practical restriction of expressiveness, e.g., via the radial[10] restraint[11] feature that is available in Rulelog.

3. Strong explainability, including fully detailed proof-type explanations (for all answers) that are highly comprehensible by those lacking KE skill (i.e., by nonlogicians)[12,13].
4. Availability of a commercial-quality open-source toolset for the logical language (e.g., candidate toolset members for Rulelog include ErgoAI[14], XSB[15], and Janus[16], a Python bridge; candidate toolset members for ASP include Potassco[17], DLV[18] and s(CASP)[19]). Each performer will choose and use one such toolset. The chosen toolset should provide highly capable reasoning that satisfies the requirements above ((1.)-(3.)) and, in particular, automatically provides not only conclusions (i.e., answers to queries) but also explanations as specified above.

These four requirements will enable the reasoning toolset to provide deontic reasoning that is able to reach high assurance, including explainability, verifiability, and extreme correctness (very high accuracy, i.e., >99%).

B. Objective/Technical Scope

CODORD performers will focus on developing new automatic techniques for translating deontic knowledge from NL into logical language. To constrain the problem, CODORD performers must concentrate specifically on generating deontic logical knowledge that is expressed in an open-source logical language that meets the four practical requirements stated above. A human knowledge author will input a sentence or other short passage of NL text, and the performer's novel AI technique will then generate one or a few logical-language expressions (e.g., assertion statements or queries) as output. One or more human authors will then revise (i.e., review, test, and modify) that output. To revise it, an author might edit it directly in the logical language or might instead rerun the generating process by modifying its NL input; however, automating the revising process beyond such rerunning is out of scope for CODORD. (Currently, SOA KA revising is primarily done manually by a human editing directly in the logical language.)

The suitability of the generated logical-language expressions will be assessed not only syntactically but, more importantly, semantically. Semantic assessment will be based on

---

[10] Grosof, Benjamin, et al, Radial Restraint: A Semantically Clean Approach to Bounded Rationality, *Proc. AAAI Conf. on Artificial Intelligence,* Vol. 27, No. 1, 2013.
[11] Andersen, Carl, et al, Advanced Knowledge Base Debugging for Rulelog, *Proc. 7th Intl. Web Rule Symposium (RuleML-2013)*, 2013.
[12] Andersen, Carl, et al, Advanced Knowledge Base Debugging for Rulelog, *Proc. 7th Intl. Web Rule Symposium (RuleML-2013)*, 2013.
[13] Grosof, Benjamin, et al, A SILK Graphical UI for Defeasible Reasoning, with a Biology Causal Process Example, *9th Intl. Semantic Web Conf.,* 2010.
[14] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI Swift, Theresa, et al, XSB software & manuals, https://xsb.sourceforget.net
[15] Swift, Theresa, et al, XSB software & manuals, https://xsb.sourceforget.net
[16] Swift, Theresa, et al, The Janus System: Multi-paradigm Programming in Prolog and Python, *Proc. Intl. Conf. on Logic Programming (ICLP-2023),* 2023.
[17] https://potassco.org/
[18] https://www.mat.unical.it/dlve/
[19] https://swish.swi-prolog.org/example/scasp.swinb

performance against a set of domain-specific focal queries using a logical-language reasoning engine, where each domain is particular to a use case and set of test problems.

Each performer, as their main research task, will design and implement novel techniques for generating logical knowledge (expressions) in their chosen logical language. They will deliver these techniques in the form of software (with code documentation and perhaps runnable examples), including any utilized ML models and their associated training examples, and materials for training human users to generate deontic logical knowledge using their novel approach. Human user training materials will be in the form of text documents, and possibly slides and/or videos.

To develop novel techniques for generating logical knowledge, performers will select among available previous AI techniques to build upon (e.g., to combine, modify, or extend) and then exercise their creativity. There is considerable need, and ample room, for such creativity, especially in the areas of both NLP and ML.

Among the available AI techniques and resources, there has been dramatic recent progress in NLP, especially based on large pretrained neural network (NN) ML – notably, large language models (LLMs). A variety of open-source and proprietary LLMs are available in a range of sizes, smaller ones of which are more affordable to (re)train. There's been recent dramatic research progress in ML-based generation starting from NL into programming languages. LLMs have demonstrated impressive successes in domains such as translation between different NLs[20] (a.k.a., machine translation) and coding assistants that are based on code generation for imperative[21,22] programming languages such as Python, C, and JavaScript, especially to assist programmers who are less than very experienced and for programming tasks of low to medium difficulty. LLMs have also been used to generate code into declarative[23,24] programming languages such as SQL and Prolog.

Another relevant category of NLP is semantic parsing, which focuses specifically on interpreting NL in terms of logic (or logic-like) representations. Semantic parsing can be combined with NN/LLM methods for NLP, as some recent research work addresses[25,26]. There are additional relevant categories of NLP, e.g., logic-to-NL generation and diverse rephrasings[27]; these have potential for creating synthetic training data for ML methods. Style conventions, analogous to

---

[20] Zhang, Biao, et al. Prompting large language model for machine translation: A case study. *Intl. Conference on Machine Learning*. PMLR, 2023.
[21] Wermelinger, Michel. Using github copilot to solve simple programming problems. *Proc. 54th ACM Technical Symposium on Computer Science Education V. 1*. 2023.
[22] Agarwal, Anisha, et al, Copilot Evaluation Harness: Evaluating LLM-Guided Software Programming, *arxiv.* https://arxiv.org/abs/2402.14261
[23] Borazjanizadeh, Nasim, et al. Reliable Reasoning Beyond Natural Language. *arxiv.* 2024. https://arxiv.org/pdf/2407.11373
[24] Russell, Stuart J., et al. *Artificial intelligence: a modern approach*. Pearson, 2016. See discussion of declarative: pp. 210, 247, 251.
[25] Liu, Shicheng, et al, SUQL: Conversational Search over Structured and Unstructured Data with Large Language Models, *arxiv 2024*. https://arxiv.org/html/2311.09818v2.
[26] Bao, Qiming, et al. "Abstract Meaning Representation-Based Logic-Driven Data Augmentation for Logical Reasoning." *arXiv preprint arXiv:2305.12599* (2023).
[27] Maini, Pratyush, et al, Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling*, arxiv* 2024. https://arxiv.org/abs/2401.16380

programming code style conventions, for the generated logic expressions may be helpful. There may be other relevant ML categories and techniques that proposers may consider.

Performers will use manual KA methods (notably, for revising knowledge) as well as their novel automated KA techniques, while developing and testing their novel automated KA techniques.

Proposers must describe the technical approach they intend to take. Proposers must provide specific technical arguments/analysis and evidence (citations of previous research papers/systems, previous experimental data, and/or demonstrated software) that their technical approaches have the potential to meet both the program goals and metrics and the four practical requirements (1.)-(4.) above. See additional details below in subsections C and D about program metrics and milestones.

Overall, each performer team will need to combine expertise in ML and NLP as well as in AI logical knowledge representation & reasoning (KRR). Proposers must describe in specific technical detail what kind of such expertise their teams have and will apply to the main task of developing their novel technical approaches for generating logical knowledge.

Required/Recommended Background in KRR techniques: It is not expected that performer teams will already be highly familiar and experienced with their chosen specific logical language and toolset for deontic logical reasoning that meets the four practical requirements stated above before CODORD's period of performance (POP) begins.

C. Structure

Submissions in response to this DO must be unclassified and must address the two sequential project phases: a Phase 1 base and a Phase 2 option. Phase 1 will focus on developing initial logic-generating capabilities from NL. Phase 2 will focus on scaling up to larger domain test problems while increasing correctness and the degree to which logical knowledge is generated automatically (see more details about metrics and milestones below in this subsection and subsection D). The periods of performance for these phases are 12 months for the Phase 1 base effort and 12 months for the Phase 2 option effort. Combined Phase 1 base and Phase 2 option efforts for this DO should not exceed 24 months. The Phase 1 (base) award value is limited to $1,100,000. The Phase 2 (option) award value is limited to $1,100,000. Please note that the total award value for the combined Phase 1 and Phase 2 is limited to $2,000,000. This total award value includes Government funding and performer cost share, if required or if proposed.

CODORD will also fund a FFRDC, UARC, or Government entity as the Test & Evaluation (T&E) team (not solicited under this Program Announcement). The T&E team's initial focus during Phase 1 will be to develop a common test framework to support evaluation of performer KA approaches, and an initial detailed protocol for independent evaluation. Throughout both Phases, the T&E team will also develop multiple test problems within at least three use case domains; these may include autonomy, navigation of international agreements, and operational planning. All active performers will participate in all use cases.

Development of the specifics of these test problems and use cases will be in accordance with the program schedule below (Figure 1) in collaboration between the T&E team, the DARPA PM/SETA team, and performers. During this development, performers will be expected to actively engage with the T&E team and each other to facilitate test problem relevance, mutual understanding of use cases and capabilities of performer approaches, construction of training examples for ML approaches, and diffusion of best practices and novel insights.

During the T&E team's independent evaluations of each performer's novel KA approach on use cases, performers will not be directly generating logical statements, revising logical statements, or executing queries. Rather, each performer will be responsible for developing sufficient documentation and training materials to allow the T&E team to successfully execute the logical knowledge generating task using the performer's KA approach. In parallel, as part of its evaluations, the T&E team will also execute the same tasks using SOA manual KA techniques. This evaluation protocol will provide a point of comparison to determine the relative advantages of performer approaches to SOA KA methods and, moreover, will allow DARPA to assess the accessibility of performer systems by persons with specified levels of logical expertise, from novices to experienced KEs, in a least-biased fashion. The metrics by which DARPA will assess performance are provided in Table 1.

Metric for performer KA Approach	**Phase 1** (12 months)	**Phase 2** (12 months)
Assurance*		
Correctness of answers to focal queries compared to SOA KA for each use case	(1 – Correctness) is within 3x of SOA KA	Correctness ≥ SOA KA
Cost-Efficient: on *Generating*		
Automatic-from-NL logic generation, as percentage of: logic sentences after *Revising*	40%** of: logic sentences after *Revising*	80%** of: logic sentences after *Revising*
Total KA labor time (including *Revising)*, compared to SOA for each use case	Total KA Labor Time ≤ 2x*** SOA KA	Total KA Labor Time ≤ SOA KA

Table 1: CODORD Metrics

* The automated reasoning is fully verifiable. ** SOA is 0%. *** Goal of program is feasibility, not yet optimality.

Additional details on the associated milestones and schedule are described in the next subsection D, particularly its Figure 1.

D. Schedule/Milestones

Proposers must address the following fixed payable milestones in their proposals. Proposers must complete the "Schedule of Milestones and Payments" Excel Attachment provided with this DO to submit a complete proposal and fulfill the requirements under Volume 2, Price Volume. If selected for award negotiation, the fixed payable milestones provided will be directly incorporated into Attachment 3 of the OT agreement ("Schedule of Milestones and Payments"). Proposers must use the Task Description Document template provided with the Program Announcement DARPA-PA-24-04, which will be Attachment 1 of the OT agreement.

Phase 1 fixed milestones for this program must include, at a minimum, the following:

- Month 1: Participation in Phase 1 project kick-off meeting. All supporting positions identified in the proposal are assigned to personnel, and names are provided to the Government.
- Month 3: Progress report identifying data sources to be used for training of ML aspects of the logic-generating approach and describing of the modifications/alternatives to the logic-generating approach relative to those initially proposed. All proposed personnel must be working on the project at the planned level of effort.
- Month 6: Demonstration of initial logic-generating capability. Progress report detailing performer experience (as distinct from independent evaluation) relative to Phase 1 metrics, insights, and obstacles encountered.
- Month 9: Performer-led demonstration of logic-generating capability on Month 9 use

cases. Progress report detailing performer experience relative to Month 9 metrics, and plans to achieve (end of) Phase 1 metrics. Provision of software, documentation and training materials to support Month 9 T&E independent evaluation.

- Month 12: Performer-led demonstration of logic-generating capability on Month 12 use cases. Phase 1 final report detailing performer experience relative to Phase 1 metrics, insights, and obstacles encountered. Provision of software, documentation and training materials to support Month 12 T&E independent evaluation. Phase 2 fixed milestones for this program must include, at a minimum, the following:
- Month 15: Performer-led demonstration of logic-generating capability. Progress report detailing performer experience relative to Month 18 metrics and plans to achieve (end of) Phase 2 metrics.
- Month 18: Performer-led demonstration of logic-generating capability on Month 18 use cases. Progress report detailing performer experience relative to Month 18 metrics and plans to achieve Phase 2 metrics. Provision of software, documentation and training materials to support Month 18 T&E independent evaluation.
- Month 21: Performer-led demonstration of logic-generating capability on Month 21 use cases. Progress report detailing performer experience relative to Phase 2 metrics. Provision of software, documentation and training materials to support Month 21 T&E independent evaluation.
- Month 24: Performer-led demonstration of logic-generating capability. Phase 2 final report detailing performer experience relative to Phase 2 metrics, insights, and obstacles encountered.

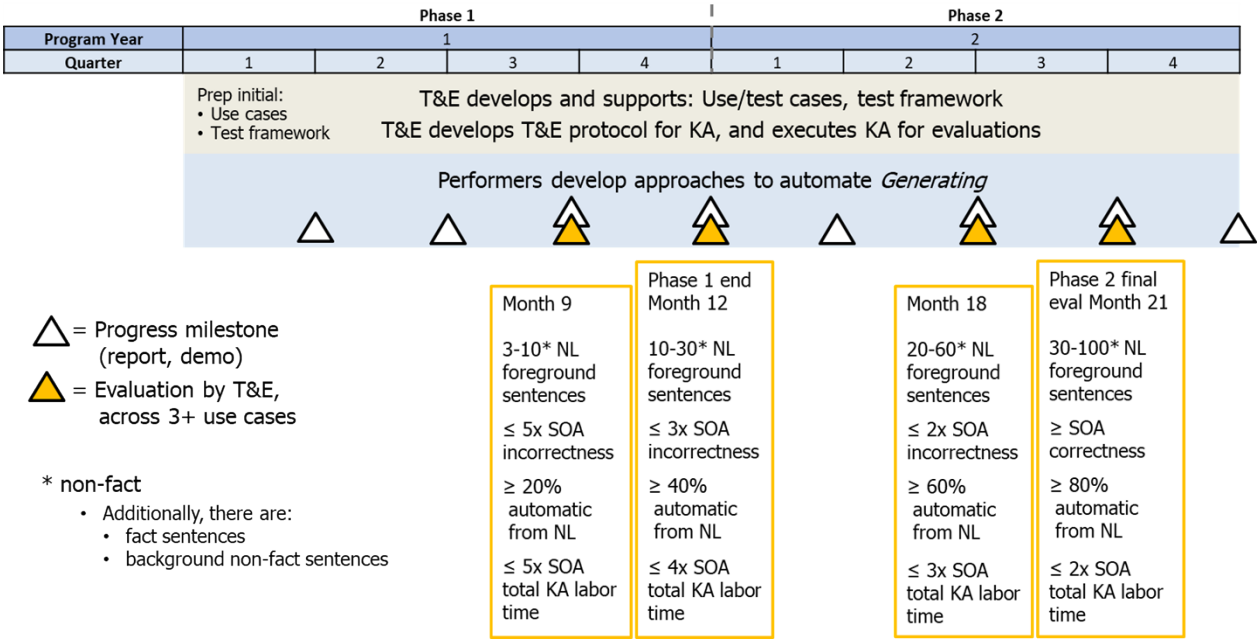

Figure 1: CODORD Schedule

Figure 1 describes and depicts the overall schedule for CODORD, including milestones and their associated goals for the program metrics. "Foreground" knowledge means knowledge focused upon by subject matter experts in the use case domain. By contrast, "background" knowledge means other needed knowledge, e.g., that is required to connect or "handshake" between multiple pieces of knowledge. "Fact" knowledge means having the form of a fact, i.e., lacking an "if-then".

For planning and budgetary purposes, proposers should assume a program start date of February 10, 2025. Schedules will be synchronized across performers, as required, and monitored/revised as necessary throughout the program's period of performance.

All proposals must include the following meetings and travel in the proposed schedule and costs:

- To foster collaboration between teams and disseminate program developments, a one-day in-person Principal Investigator (PI) meeting will be held approximately every six months. For planning purposes, assume one per phase in the Washington, DC area and one per phase on the US West Coast.
- Regular teleconference meetings will be scheduled with the Government team for progress reporting and problem identification and mitigation. Proposers should also anticipate possible site visits (e.g., one per phase) by the DARPA Program Manager, during which they will have the opportunity to demonstrate progress toward agreed-upon milestones.
- Conference and publication costs should not be included.

E. Deliverables

Performers will be expected to provide, at a minimum, the following deliverables:

- Negotiated deliverables specific to the objectives of the individual efforts. These may include registered reports, experimental protocols, publications, intermediate and final versions of software libraries, code, and APIs, including documentation and user manuals, and/or a comprehensive assemblage of design documents, models, modeling data and results, and model validation data.

II. Award Information

Selected proposals that are successfully negotiated will result in the award of an OT for Prototype project. See Section 4 of DARPA-PA-24-04 for information on awards that may result from proposals submitted in response to this announcement.

Proposers must review the model OT for Prototype agreement provided as an attachment to DARPA-PA-24-04 prior to submitting a proposal. DARPA has provided the model OT to expedite the negotiation and award process and ensure DARPA achieves the goal of Disruptioneering, which is to enable DARPA to initiate a new investment in less than 120 calendar days from idea inception. The model OT is representative of the terms and conditions that DARPA intends to include in all DO awards. The task description document, schedule of milestones and payments, and data rights assertions requested under Volumes 1, 2, and 3 will be included as attachments to the OT agreement upon negotiation and award.

Proposers may suggest edits to the model OT for consideration by DARPA and provide a copy of the model OT with track changes as part of their proposal package. DARPA may not accept suggested edits. The Government reserves the right to remove a proposal from award consideration should the parties fail to reach an agreement on OT award terms and conditions. If edits to the model OT are not provided as part of the proposal package, DARPA assumes that the proposer has reviewed and accepted the award terms and conditions to which they may have to adhere and the model OT agreement provided as an attachment, indicating agreement (in principle) with the listed terms and conditions applicable to the specific award instrument.

To ensure that DARPA achieves the goal of an award within 120 calendar days from the posting

date (October 11, 2024) of this announcement, DARPA reserves the right to cease negotiations when an award is not executed by both parties (DARPA and the selected organization) on or before February 7, 2025.

III. Eligibility

See Section 7 of DARPA-PA-24-04 for information on who may be eligible to respond to this announcement.

IV. Disruption Opportunity Responses

A. Abstract Phase

This DO contains an abstract phase. Abstracts are strongly encouraged but not required. DARPA will respond to abstracts with brief feedback regarding applicability and technical relevance to the DO and will either encourage or discourage submission of a proposal. Regardless of DARPA's response to an abstract, proposers may submit a full proposal. DARPA will review all conforming full proposals using the published evaluation criteria and without regard to any comments resulting from the review of an abstract. Proposers should note that a favorable response to an abstract is not a guarantee that a proposal based on the abstract will ultimately be selected for award negotiation, and vice versa. Abstracts must be submitted using the attached template ("Abstract Template"), and in accordance with the submission instruction below no later than October 31, 2024, at 4:00 p.m. ET.

B. Proposal Content and Format

All proposals submitted in response to this announcement must comply with the content and format instructions in Section 5 of DARPA-PA-24-04. All proposals must use the templates provided as Attachments to DARPA-PA-24-04 and the "Schedule of Milestones and Payments" Excel Attachment provided with this DO and follow the instructions therein.

Information not explicitly requested in DARPA-PA-24-04, its Attachments, or this announcement may not be evaluated.

C. Submission Instructions

Responses to DARPA-PA-24-04-01 shall be submitted electronically to DARPA's Broad Agency Announcement (BAA) Portal (https://baa.darpa.mil).

DARPA will acknowledge receipt of complete submissions via email and assign identifying numbers that should be used in all further correspondence regarding those submissions. If no confirmation is received within two (2) business days, please contact CODORD@darpa.mil to verify receipt.

When planning a response to this DO, proposers should take into account the submission time zone and that some parts of the submission process may take from one (1) business day to one month to complete (e.g., registering for a SAM Unique Entity ID (UEI) number or Tax Identification Number (TIN)).

Electronic Upload

First-time users of the DARPA BAA Portal must complete a two-step account creation process. The first step consists of registering for an extranet account by going to the URL above and selecting the "Account Request" link. Upon completion of the online form, proposers will receive two separate emails; one will contain a username, and the second will provide a

temporary password. Once both emails have been received, the second step requires proposers to go back to the submission website and log in using that username and password. After accessing the extranet, proposers may then create a user account for the DARPA Submission website by selecting the "Register your Organization" link at the top of the page. Once the user account is created, proposers will be able to see a list of solicitations open for submissions, view submission instructions, and upload/finalize their submission.

Proposers who already have an account on the DARPA BAA Portal may log in at https://baa.darpa.mil, select this solicitation from the list of open DARPA solicitations and proceed with their submission. Note: proposers who have created a DARPA Submission website account to submit to another DARPA Technical Office's solicitations do not need to create a new account to submit to this solicitation.

All submissions provided electronically through the DARPA Submission website must meet the following requirements: (1) uploaded as a zip file (.zip or .zipx extension); (2) only contain the document(s) requested herein; (3) only contain unclassified information; and (4) must not exceed 100 MB in size. Only one zip file will be accepted per submission. The DARPA Submission website will reject submissions not uploaded as zip files. Technical support for the DARPA Submission website is available during regular business hours, Monday – Friday, 9:00 a.m. – 5:00 p.m. Requests for technical support must be emailed to BAAT_Support@darpa.mil with a copy to CODORD@darpa.mil. Questions regarding submission contents, format, deadlines, etc., should be emailed to CODORD@darpa.mil. Questions/requests for support sent to any other email address may result in delayed/no response.

Since proposers may encounter heavy traffic on the web server, DARPA discourages waiting until the day submissions are due to request an account and/or upload the submission. Note: Proposers submitting via the DARPA Submission site MUST (1) click the "Finalize" button for the submission to upload AND (2) do so with sufficient time for the upload to complete prior to the deadline. Failure to do so will result in a late submission.

D. Proposal Due Date and Time

Proposals in response to this announcement are due no later than 4:00 p.m. on December 10, 2024. As described in Section 5 of DARPA-PA-24-04, full proposal packages must be submitted per the instructions outlined in this DO *and received by DARPA* no later than the above time and date. Proposals received after this time and date may not be reviewed.

Proposers are warned that the proposal deadline outlined herein is in Eastern Time and will be strictly enforced. When planning a response to this announcement, proposers should consider that some parts of the submission process may take from one (1) business day to one (1) month to complete.

V. Proposal Evaluation and Selection

Proposals will be evaluated and selected in accordance with Section 6 of DARPA-PA-24-04. Proposers will be notified of the results of this process as described in Section 8.1 of DARPA-PA-24-04.

VI. Administrative and National Policy Requirements

Section 8.2 of DARPA-PA-24-04 provides information on Administrative and National Policy Requirements that may be applicable for proposal submission and performance under an award.

VII. Point of Contact Information

Benjamin Grosof, Program Manager, DARPA/DSO, CODORD@darpa.mil

VIII. Frequently Asked Questions (FAQs)

All technical, contractual, and administrative questions regarding this announcement must be emailed to CODORD@darpa.mil. Emails sent directly to the Program Manager or any other address may result in delayed or no response.

All questions must be in English and must include the name, email address, and telephone number of a point of contact. DARPA will attempt to answer questions publicly in a timely manner; however, questions submitted within seven (7) calendar days of the proposal due date listed herein may not be answered.

DARPA will post a FAQ list under the DO on the DARPA/DSO Opportunities page at http://www.darpa.mil/work-with-us/opportunities. The list will be updated on an ongoing basis until one (1) week before the proposal due date.

For those new to DARPA or national security, DARPA makes available a free, comprehensive resource via DARPAConnect on how to do business with the agency. In addition to DARPA 101 materials, relevant preparatory modules include "Understanding DARPA Broad Agency Announcements." Registration and access are free at www.darpaconnect.us.