Problem Statement Title: Autonomy Software for Heterogeneous Multi-Platform Multi-Domain Unmanned Systems– Assessment Event (AE)

Problem Statement: Fielding and sustaining autonomy on unmanned systems requires mission autonomy software capable of executing diverse missions within complex, rapidly changing environments. With centralized control being untenable in contested electromagnetic environments, and proprietary, vendor-locked software stacks resulting in stranded capabilities, Unmanned Systems Autonomy and Interoperability (UxSAI) seeks decentralized and collaborative autonomy solutions for heterogeneous, multi-platform, multi-domain unmanned systems. Mission autonomy is needed that incorporates mission planning, multi-agent coordination and management, dynamic re-tasking, world state monitoring and sharing, sensor and payload control, and the ability to operate in a non-deterministic environment. While solutions can be rules based, operational environments will most likely have unaccounted permutations that require agentic solutions. In addition, a vendor's mission autonomy software must enable seamless integration into a government owned software baseline called "Collaborative Heterogeneous Autonomy Operations Software (CHAOS)" by conforming to government provided interface control documents and message sets. CHAOS as a mission system provides autonomy resources, core software services, and architectural benefits that are designed to facilitate rapid integration with 3rd party mission autonomy providers. Mission autonomy vendors can focus on mission execution and associated collaborative autonomous behaviors without the reliance on developing a full vertical stack software solution.

Assessment Criteria

Respondents should address the following criteria in their submission and substantiating data, detailing how their proposed autonomy meets or addresses each item. Submissions will be evaluated using a decision analysis methodology to ensure a comprehensive and impartial assessment. The evaluation will weigh the Assessment Criteria based on their relative importance to program objectives. The primary evaluation domains, in descending order of importance, are: 1) Integration & Interoperability, 2) Technical Capability, 3) Operational Relevance, 4) Enduring Capability, and 5) DoW Experience & Readiness. Vendors must understand that Domain 1: Integration & Interoperability serves as the primary gateway for consideration. A solution's technical merit cannot be evaluated if it does not demonstrate a clear, convincing architectural approach to achieving conformance with government-furnished Software Development Kits (SDK) and open-architecture principles. Failure to convincingly address this domain will result in non-selection, regardless of capabilities demonstrated in other areas.

Domain 1: Integration & Interoperability

- **CHAOS SDK Compliance and Open Architecture:** Mission autonomy software is intended to run in an open, vendor-neutral runtime that is architected to readily adapt to the CHAOS SDK's interfaces. Feasibility of the capability to conform to the program's documented, government-controlled interface specifications once provided is the

primary open-architecture evaluation gate. Portability is demonstrated by running the mission autonomy behind those interfaces on a government-furnished test harness, including a computer vision model the proposer did not develop, across at least two silicon families (i.e., AMD64 and ARM, with augmented GPU [CUDA] or OpenCL resources). Describe demonstrated conformance to applicable government-owned open systems reference architectures for autonomy software, supported reference runtimes, and the integration timeline observed in prior platform integrations. Identify which of the following autopilot and middleware standards the autonomy stack is built on or natively interoperates with: PX4 / Pixhawk (with MAVLink), ArduPilot, and ROS 2 (Humble or Jazzy LTS). Proposers should treat any commercial or community middleware they support, for example ROS 2, PX4, or ArduPilot, as evidence of open-architecture convergence rather than as a substitute for conformance to the program's interfaces. If the autonomy runs at the application layer above an autopilot, describe the bridge or integration pattern used (e.g. micro-ROS DDS, MAVROS, or equivalent). If proprietary middleware is used, describe how open-interface integration is provided. Describe Modular Open Systems Approach (MOSA) posture per Title 10 USC 4401(b) and how DoD Directive 3000.09 is enforced through the deterministic verifier layer.

- **Adherence to any Government Reference Architecture (GRA):** Does the solution leverage any GRA Such as A-GRA, UUMA, etc.? Explain how any of these GRAs have been employed and whether their solution can understand existing message set standards such as UCI.
- **Runtime Architecture and Stack Boundary Discipline:** Describe your ability to deliver mission autonomy as a containerized, hot-reloadable component designed to integrate with external interfaces and government-provided SDKs. Describe the autonomy-to-perception interface and how the autonomy stack consumes detection, classification, tracking, and out-of-distribution flags from a computer-vision (CV) module. If the capability bundles CV, describe how the autonomy portion and CV could be modified to adhere to the separation-of-concerns principle. Describe silicon-agnostic deployment across at least two distinct silicon families (e.g., NVIDIA Jetson, AMD Versal, Qualcomm, neuromorphic). Identify the need for custom or CoTs hardware that the solution must operate on (e.g. GPU, FPGA).
- **Performance and Benchmarking:** Describe the resources necessary to execute the mission autonomy tasking; this should include CPU cores, memory amount and speed, GPU (or other) cores required, and the amount of usage required for CPU/GPU/memory usage. Identify any network/communication bandwidth requirements that are necessary to demonstrate the collaborative nature of the mission autonomy solution. Describe how the mission autonomy can support dynamic scaling/scale-zero based on mission threads and stage within a mission thread.
- **Integration and Test Environment:** Describe how previously developed autonomy was tested and validated. Detail the approach to incremental autonomy updates, so that a design, develop, test, and evaluate development loop could be realized. How quickly can this cycle be realized? Identify your test strategy and whether it focuses on software-in-the-loop, hardware-loop, or vehicle-in-the-loop testing. Where is most of

the debugging accomplished? What makes your test approach stand out from other competitors. Describe what amount of documentation would be provided to developers and integrators to enable rapid deployment into test environments. Describe what interfaces or supporting technologies are available to enable modeling and simulation environments (e.g. DIS PDU).

- **Battlespace Deconfliction and Integration**: Vendors must describe the architecture and methodology for ensuring comprehensive Battlespace Deconfliction, extending beyond simple intra-swarm collision avoidance. The solution must demonstrate how it safely deconflicts its flight paths and operations from other assets in a shared environment, including manned aircraft, other friendly unmanned systems, and known civilian traffic. This description must detail how the system ingests, processes, and adheres to dynamic Airspace Control Orders (ACOs), Restricted Operating Zones (ROZs), and other control measures. Furthermore, vendors must specify the mechanisms for integrating with external situational awareness data feeds and command and control networks to ensure the system is a fully integrated participant in the broader operational picture.

Domain 2: Technical Capability

- **Mission Autonomy Architecture:** Describe the mission planning architecture used (e.g., rule-based or symbolic planning architectures such as Hierarchical Task Network or Belief-Desire-Intention; neural mission planners using a small language model or vision-language-action model in the 1B to 3B parameter band; hybrid neuro-symbolic stacks with a deterministic classical verifier). If multiple methods are combined, describe how they are composed and the role each plays. Describe what inputs the platform requires to plan and execute a mission. Can the autonomy operate from mission intent and a task catalog alone, or does it require waypoint-level scripting, pre-built behavior trees, or external coordination services? Can the solution handle different Human-Machine Teaming (HMT) modes (e.g., human in | on | out of the loop) based machine-authorized actions? What actions is the autonomy allowed to do and what actions require human approval? Provide details on where mission planning is occurring, whether entirely on platform, on the ground, or a combination of both. Can the autonomy conduct dynamic re-tasking during mission, even when disconnected from an operator?
- **Mission Threads Supported:** Describe the mission threads that can be supported with your mission autonomy software. Example mission threads would be intelligence, surveillance, and reconnaissance, and logistics mission threads. An emphasis on multi-platform, multi-domain, collaborative mission threads is desired. What mission threads are planned for upcoming development? Describe the user interface, workflow for composing and loading mission packages, and the level of technical expertise required to operate the platform as currently configured. Describe how operators reconfigure fleet behaviors, no-go zones, and planner rule sets in the field. Are these manually configured by the operator, or can the system ingest Airspace Coordinating Measures or other battlespace management messages to configure itself?

- **Mission Autonomy Behaviors:** Describe the autonomous behaviors your software employs for both single platform and swarms. Provide details on the types of missions supported, patterns or approaches employed, different types of sensors utilized, and the domains and number of platforms incorporated. Also identify how behavioral performance was tested and validated and determined to be an improvement over what was previously employed. For behaviors that include multiple platforms, identify the number of operationally relevant platforms that can collaborate in the behaviors.
- **Multi-Agent Collaboration & Safety:** Autonomy supports decentralized coordination across two or more platforms with peer-to-peer task hand-off, dynamic fleet topology (joins, losses, degraded units), and shared mission state synchronization through intermittent links. Describe the coordination architecture (decentralized multi-agent reinforcement learning, distributed task allocation, shared world model, or equivalent) and how it scales from 2 to 10 platforms today and toward 25 to 50 platforms over the next 24 months. Describe the deterministic safety verifier that gates every autonomous action against flight envelope, airspace, and DoD Directive 3000.09 human-in-the-loop constraints, independent of any neural component.
- **System Health Monitoring and Fault Management:** Vendors must detail their framework for System Health Monitoring and Fault Management. The description must specify the methodology for monitoring the internal state of the platform and its subsystems, including hardware, software, and power. Explain the system's approach to Fault Detection, Isolation, and Recovery (FDIR), detailing the pre-planned contingency behaviors the autonomy can execute in response to critical internal failures. Are these contingency behaviors configurable? The primary objective is to demonstrate a system capable of self-preservation that can fail gracefully and ensure its own safety, even in a communications-denied environment. When comms are available, describe how this information is shared with the operator or any GCS involved, as well as how they collectively resolve the problems. What problems can the autonomy handle, and what requires human intervention?
- **Comms-Degraded Operation:** Describe the mission autonomy algorithms that have been implemented for degraded or denied communications, including broadband and frequency-agile radio frequency jamming, GNSS denial, and intermittent rear datalink with no forward tactical communications. Describe the onboard reasoning and re-planning behavior when the link is severed, including the auditable decision trace produced for after-action review. Provide details on the autonomous behaviors that have been put in place to account for these environments. Also, detail efforts to modify or define platform contingency behaviors, including lost communications, lost GNSS, and low fuel or battery. Describe any demonstrated mission completion under realistic electronic attack within the last 24 months, including the degradation profile applied (rear link uptime, mesh maintained or not, GNSS available or not) and the independent verification source (exercise white cell, government range, or autonomy assurance organization).
- **Contextual and Semantic Reasoning in Planning**: Describe how the mission autonomy uses contextual and semantic information to optimize mission execution and

avoid unproductive actions. The system should demonstrate an ability to reason about the relationship between target types, environmental states, and terrain features. For example, explain how the planner would dynamically alter a search plan for ground vehicles if it determines a designated search area (e.g., a body of water) is impassable for that target type based on current environmental data. This may include kinematics of potential targets.

- **Sensor-Aware Mission Planning**: Explain how the mission planner and sensor/payload controller interact. How does the autonomy plan platform maneuvers, gimbal scan patterns, and sensor settings optimize data collection based on the assigned task and environmental conditions? Describe how the system accounts for sensor performance models (e.g., probability of detection/recognition) in a given environment (biome, target set, etc.).
- **Situational Anomaly Response and Adaptive Planning:** Vendors must describe their methodology for Situational Anomaly Response and Adaptive Planning. This involves explaining how the autonomy detects and reacts when the external environment deviates significantly from the mission plan's assumptions. Provide examples of how the system would respond to anomalies such as GPS denial, un-forecast weather phenomena, pop-up threats, or discovering that a planned route is impassable. The description should demonstrate the system's ability to dynamically re-plan in response to these external events to maximize mission effectiveness while maintaining operational safety.

Domain 3: Operational Relevance

- **Defense Use Case Alignment:** Current mission autonomy software capabilities are applicable to ISR within the C5ISR framework, with specific applicability to Group 1 (under 20 lb) and Group 2 (21 to 55 lb) small unmanned aerial systems, and small unmanned surface vessels. The program's initial operating focus is the maritime and littoral environment, but the autonomy architecture should be terrain-general and extensible to additional operating environments, including land, desert, and urban, without re-engineering of the core stack. Describe any demonstrated use of autonomy across one or more of these environments, particularly for unmanned systems operating under mesh-first peer-to-peer communications with intermittent rear datalink and limited to no forward tactical comms. Describe how mission packages and behaviors port across operating environments.
- **Human-Machine Teaming (HMT) Framework:** As implied in the Comms-Degraded Operation section above, to operate effectively without a connection to a human operator (at times of fully autonomous behavior), the mission autonomy *must* have a clearly defined set of pre-delegated authorities (aka an “authorities construct”). Additionally, it is expected that any given instant in time, the platform is operating in multiple HMT states simultaneously (e.g., human in | on | out of the loop). For example, the authorities' construct may permit fully autonomous behavior of the ISR payload, but reserve effector control for the human. Vendors must describe their HMT framework,

detailing the principles that govern the collaboration between a human operator and the mission autonomy. This description must define the authorities' construct, clearly delineating which decisions are delegated to the autonomy and which require mandatory human intervention. These may change dynamically, and the vendor should describe how these are managed. The framework should explain the protocol for how the autonomy escalates a decision to the human operator, as well as how an operator can intervene to dynamically re-task the system. The explanation must specifically address how these authorities are managed in both communications-connected and communications-denied environments to ensure safe and trusted operational effectiveness.

- **Environmental Effects Modeling and Adjustment:** Describe how the mission autonomy ingests, models, and adapts to dynamic environmental conditions. This should include, but is not limited to, weather effects (wind, winds aloft, precipitation, sea state), atmospheric conditions (haze, cloud layers, time of day/lighting), and their predicted impact on platform performance (e.g., battery life, speed) and sensor effectiveness (e.g., camera and ATR performance, gimbal scan pattern optimization, etc.). In the human-machine teaming (HMT) construct, these variables place way too much cognitive burden on the operator. These are not just pre-launch considerations. They are dynamic and affect each node in the swarm differently. One node may be significantly affected by broken clouds, and other nodes in the same swarm are not. Placing this burden on the operator is infeasible.
- **Problem-Solution Alignment:** Mission autonomy software should rapidly accommodate novel mission profiles, novel adversary tactics, and novel platform compositions without vendor-side custom development. Describe the process and timeline for introducing a new mission package, a new behavior, or a new platform into the autonomy stack. Describe Continuous Integration/Continuous Delivery (CI/CD) approach and how it can be integrated into government ATO cycles (assuming continuous ATO) to get new code rapidly fielded.
- **Dynamic Mission Adjustment:** Since operational environments are subject to dynamic weather conditions and active adversarial interference, the original mission plan being executed may no longer be viable. Necessary changes occur at different levels, and an individual node in the swarm may decide to drop altitude to get below the cloud deck to see potential targets. This alters the planned optimum altitude and resulting scan pattern and time to complete its assigned area. This may or may not need to be relayed to the swarm/swarm to lead to initiating adjustments and reassignments of search regions. Swarm attrition may significantly affect node tasking. External information such as change in Commander's intent may necessitate dynamic mission re-tasking from the offboard, AI-enabled UxS C3/tasking application that often runs on a Ground Control Station (GCS). Thus, there are node, swarm, and GCS/C2 collaborations that need to occur during an active mission in various situations. Explain the methodology for significant mission adjustments. How does the system handle human-in-the-loop commands for re-tasking versus an AI-driven decision to initiate a major replan based on mission failure or a new opportunity? How can the solution handle challenging

potential override conditions such as ignoring bingo fuel if following a high value target?

- **Rules of Engagement (ROE) and Effector Employment**: ROEs are not limited to weapons/effector employment. They are a comprehensive set of directives that govern the actions of military forces, and they frequently impose significant constraints on the maneuver and sensor operations of unarmed ISR platforms. The core purpose of these non-kinetic ROE is to control escalation, maintain political agreements, and prevent misinterpretation by adversaries, allies, or neutral parties. These include, but are not limited to, (1) Prohibiting active surveillance of specific areas, such as allied compounds, neutral territories, or sensitive locations covered by political agreements (e.g., hospitals, schools, religious sites); (2) Limiting Sensor Types: Allowing the use of standard full-motion video (FMV) but restricting the use of more advanced intelligence-gathering payloads (e.g., SIGINT, LiDAR) in certain operational areas; and (3) Prohibiting the use of active sensors that emit energy (like radar or laser designators) that could be detected and interpreted as a prelude to an attack. Describe the architectural provisions for incorporating ROE, including kinetic and non-kinetic effectors. Explain how the deterministic safety verifier is designed to enforce complex ROE, including target validation, No-Strike Lists (NSL), and constraints on collateral effects. Detail the auditable trace generated for any effector-related decision.

Domain 4: Enduring Capability

- **Ease of Integration:** It is highly desired that the mission autonomy software can be integrated into the CHAOS baseline without sustained vendor involvement. Provide details on whether your mission autonomy software has been integrated with other open architecture frameworks and if not, what challenges do you think may exist. Identify your general methodology and past performance by adapting to similar government-controlled message sets or interfaces, and any anticipated risks.
- **Standards Compliance:** Describe what standards (e.g., STANAG 4586, MISB/MISP, MISB ST 0601, MISB ST 0903 (VMTI), OMG DDS, SAE AS-4 JAUS, Open Geospatial Consortium standards, DO-178C (Software Considerations in Airborne Systems), FACE™ (Future Airborne Capability Environment), etc.) were used or portions of existing standards the solution conforms to.
- **Training & Enablement:** Describe whether a training program exists for the mission autonomy software. Include documentation, training materials, and ongoing support for both operators and maintainers.
- **Customization:** Identify whether the mission autonomy software can be tailored to mission-specific requirements without vendor-side custom development. Describe how government personnel can add new mission packages, modify behavior libraries, and integrate new sensor or platform configurations through the documented APIs, SDKs, or configuration files.
- **Sustainment:** Describe the proposed delivery model. Can the mission autonomy software be hosted on government networks? Can it be deployed on USSOCOM

infrastructure? Describe any cloud, on-premise, or hybrid deployment options. Describe the long-term sustainment, security patching, and capability evolution plan for operating within a modular, government-controlled SDK boundary.

Domain 5: DoW Experience & Readiness

- **Prior DoW Engagement:** Describe any active or prior contracts with DoW entities. Identify participation in innovation programs (e.g., AFWERX, DIU, SOFWERX, NavalX) or direct collaboration with military services. Identify any technical lineage to DARPA autonomy programs (Rapid Experimental Missionized Autonomy [REMA], Adaptive Intelligence for Robotics [AIR], or equivalent). Describe the ability to transfer the technology (software, containers, libraries, etc.) as Government-Open-Source or any proprietary components that would hinder the transfer and require re-implementation.
- **Security Posture:** Describe the security posture of the platform and key personnel. Identify any existing clearances or eligibility for clearances held by key personnel. Describe compliance posture with respect to NIST SP 800-171 (Protecting Controlled Unclassified Information in Non-Federal Systems and Organizations).
- **UxSAI Autonomy Development Environment:** Identify any issues or concerns that may exist working in an impact level 4 (IL4) environment to develop, integrate, and deliver autonomy software development. Describe whether the autonomy software and supporting development tooling can be delivered, signed, and promoted through the program's IL4 staging environment. Identify any necessary hardware or software that is required to develop and deliver the autonomy software (cloud resources/environments, instance types, development tools, etc.) within an IL4 environment.

Selection

The UxSAI Program team will review initial whitepaper submissions to this event and down-select to **several** participants whose proposed solutions best align with current program objectives. Selected respondents will then receive further information including the CHAOS SDK and guidance to demonstrate their mission autonomy during an Assessment Event.

During the Assessment Event, selected vendors will be expected to demonstrate their solution and its ability to plan and execute a representative ISR mission. Vendors will be evaluated on their ability to integrate with CHAOS and its SDK. Milestones will include:

1) Ability to provide documentation packages and protocols supported
2) Lab-based testing using the SDK and the CAAPI test tool to verify and validate messaging across the middleware
3) Lab-based testing demonstrating successful messaging with CHAOS simulator

The UxSAI program will independently evaluate the solutions provided and determine a winner, with the expectation that one vendor will be selected to develop and deliver mission autonomy software for the program.