Space Protocol's Quantum-Resilient Blockchain as the Backbone for Explainable and Accountable Space Battle Management Systems

Yasir Latif[*], **Samya Bagchi**[*], **Latha Madhuri Pratti**
Space Protocol, Adelaide, Australia

Harvey Reed
MITRE Corporation

Ruth Stilwell
Aerospace Policy Solutions

Greg Frulich
University of Colorado, Boulder, CO

[*] *equal contribution*

ABSTRACT

The proliferation of space debris, adversarial threats, and the dynamic nature of orbital operations demand Battle Management Systems (BMS) that balance rapid decision-making with rigorous traceability. Current systems often operate as black boxes, lacking mechanisms to audit decisions, refine performance over time, or explain actions to human operators – critical gaps in high-stakes scenarios where errors cascade catastrophically. This work addresses these limitations by integrating reinforcement learning (RL), blockchain-based traceability, and causal analysis into a unified framework designed for Space Domain Awareness (SDA).

We propose the **traceX** framework that combines: **I) Blockchain-Immutable Traceability**: Every data transformation across the SDA pipeline – from sensor ingestion to course-of-action (COA) – is logged to a distributed ledger with cryptographic hashing. This creates an audit trail resilient to tampering, essential for post-event analysis and compliance with international space treaties. **II) Reinforcement Learning Oversight**: A meta-layer of RL agents monitors subsystem interactions (target modeling, hostility assessment, command prioritization) using multi-objective reward functions. These agents optimize for mission success metrics while penalizing decisions with low traceability compliance or unexplained confidence variances. **III) Verification Learning**: Independent validators cross-check subsystem outputs against curated test datasets (e.g., historical collision events, simulated adversarial maneuvers) to detect model drift or adversarial data poisoning. **IV) OODA Loop UI**: Insights from traceability, oversight AI and verification learning are integrated at various touch points in an Observe, Orient, Decide, and Act (OODA) loop to enable the operator to make better decisions under time pressure. The system uniquely integrates causal analysis to map decision pathways, assigning stepwise confidence scores that inform both machine learning updates and operator briefings. A chat interface allows operators to query the rationale behind high-risk decisions (e.g.,"Why was Sensor X prioritized over Y during Event Z?"), with responses grounded in blockchain-verified logs and RL-derived correlation models and informing the operator in an easy to interpret and execute OODA loop framework.

The SDA Tools, Applications, and Processing (TAP) Lab was formed in Colorado Springs, Colorado in 2023 bringing together industry, academia, and government to deliver a Space Battle Management System. Space Protocol has adopted NIST IR 8536 "Supply Chain Traceability: Manufacturing Meta-Framework" to provide traceability for the TAP Lab's BMS. Data-to-decision traceability allows comprehensive and auditable documentation of the lifecycle of data, encompassing its origin, transformation, movement, and utilization, as well as the rationale and influencing factors behind decisions using that data. The TAP LAB BMS is organized along seven discrete event workflows that

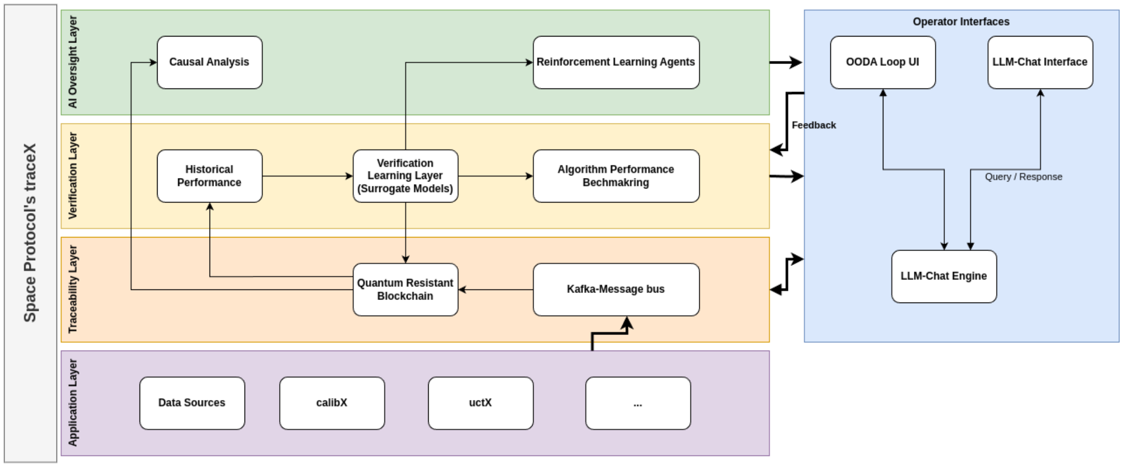


Fig. 1: An overview of the core components of the **traceX** framework. Applications interface with the Kafka-bus. Secure storage in the form of a Quantum Resistant blockchain allows trusted records of data and decisions, enabling performance benchmarking and predictions of the system evolution over time, while keeping the operator updated via a unified chat interface. See corresponding sections for further details about each component.

include launch, reentry, proximity, maneuver, link change, attitude change, and separation. Each workflow represents an event that is of interest to the operator and a critical decision point in the system.

Space Protocol's traceX framework has adopted the traceability framework from the original supply chain context to data-to-decision traceability within the context of the SDA TAP Lab. Kafka, the common data backbone, enables a unified data exchange mechanism, providing information about each data point and its transformation, providing a link in the traceability records. These records are stored using a quantum resistant distributed ledger. The data gathered via traceability enables: **a) Graph-Based Auditability**: Novel algorithms parse traceability records and reconstructs decision chains with millisecond latency, even for 10^6-node graphs, enabling rapid forensic analysis and real-time introspection. **b) RL-Driven Optimization**: Soft Actor-Critic (SAC) based agents utilize historic data and are applied to each event workflow in the BMS, such as false track associations, satellite identification, and friend vs foe assessment. **c) Human-AI Collaboration**: AI-Agent and chatbots provide detailed system analytics in a human-focused and interactive manner. Custom models are employed on premise for safety and security reasons. **d) Operator awareness**: The use of OODA loop based visual interface into the system ensures that operator familiarity and interactivity with the system. This work bridges critical gaps between autonomous decision-making and human accountability in SDA systems, providing a template for responsible AI integration in orbital operations. TraceX integration with the SDA TAP Lab demonstrates practical implementation of the framework from the grounds up, with implications for commercial space applications outside of the defence context.

1. INTRODUCTION

A Space Battle Management System (SBMS) is a military framework designed to manage and coordinate operations in the space domain, particularly during conflict or contested scenarios. It extends the concept of Battle Management, Command, Control, and Communications (BMC3) to the unique challenges of space, where assets are distributed globally, operate at high velocities, and are critical to both civilian and defense infrastructures. The SDA Tools, Applications, and Processing (TAP) Lab[1] was established in Colorado Springs in 2023, bringing together industry, academia, and government to develop an automated, collaborative SBMS [1, 9, 10] called Welder's Arc (WA). Unlike monolithic systems developed by single vendors, WA is multi-vendor system that leverages contributions from over 100 commercial entities and research labs to enable rapid innovation, maintain technological superiority by outpacing our adversaries, and avoid operational surprise in space. While the large number of contributors may offer a path for rapid system development, with so many potential independent handlers of space data, guaranteeing the data and

[1] `https://sdataplab.org/`

provenance chain of space data from input data to decision becomes imperative. To address this need, in this work we present the **traceX** framework developed by space protocol (Fig. 1) and demonstrate the benefits traceability bring to a modern Space Battle Management system.

System Integration Challenges Collaborative development, while enabling rapid innovation, introduces significant integration challenges. Each algorithmic module within the workflow may be developed by different teams, often using heterogeneous programming environments, data formats, and operational assumptions. For such a distributed system to function coherently, seamless data exchange between various entities at multiple stages of the workflow becomes essential. To address this, WA adopts an Apache Kafka–based message bus [14] as its primary data backbone.

Distributed Data Architecture Kafka provides a distributed, fault-tolerant streaming platform designed to handle large volumes of data with low latency. At its core, the Kafka message bus follows a publish–subscribe paradigm: producers push data streams into predefined logical containers called *topics*, while consumers subscribe to these topics to receive real-time updates. This decoupled architecture ensures that data producers and consumers remain independent, enabling component addition or modification without system disruption. Further complexity arises from the diversity of data exchanged—ranging from raw sensor measurements to intermediate inference outputs and final decision products. To ensure interoperability and data integrity, each Kafka topic is associated with a strict schema definition. These schemas provide guarantees about data structure and type consistency. This schema-driven approach enables forward and backward compatibility as algorithms evolve. Moreover, it provides a foundation for automated validation and monitoring of the data pipeline, thereby enhancing both robustness and scalability.

Trusted data and decisions While Kafka provides durability and in-cluster ordering, it relies on operator-controlled policies (retention, compaction, administrative deletes) that can weaken auditability in multi-party, safety-critical workflows. To address this limitation, traceX persists cryptographic commitments of messages on a permissioned blockchain, creating a tamper-evident history that no single stakeholder can rewrite. This approach provides several key benefits: consensus establishes globally verifiable ordering across organizations, producer signatures ensure non-repudiation, and on-chain metadata captures schema versions and processing context for robust provenance. Consumers can independently verify that any Kafka message matches its on-chain commitment, regulators can audit without trusting the operator, and disputes over "who produced what, when" are resolved by reference to an immutable ledger. The implementation uses a quantum-resistant blockchain detailed in Section 2.

Enabling Data provenance Without additional information, however, the message bus *only* provides a mechanism for different algorithms to talk to each other. For a complex system such as SBMS, beyond message exchange, *traceability* – tracking how data flows through the system – provides crucial insights into system state, debugging information, and replay capabilities. WA extends schemas with traceability metadata (See Appendix. B for an example) that captures the origin of consumed inputs, allowing each algorithm to document its data lineage. Mandating each algorithm in the pipeline to provide traceability information allows tracking **data and decision** throughout the system (Sec. 3). Traceable data creates a trusted data layer upon which reinforcement learning and AI can be used with confidence in both the input data, and the ability to replay in the case of forensic fault analysis (Figure 1).

Performance Benchmarking In traceX, traceability, through trusted data and the corresponding decisions, enables performance guarantees for algorithms through surrogate models (Sec. 4). These models approximate complex algorithm behavior without requiring access to their internal logic, relying instead on the transparent record of inputs, outputs, and decision pathways. Such surrogate models are then be used in traceX to benchmark performance across varying operational contexts, ensuring that algorithms remain accountable and robust under evolving conditions. Moreover, comprehensive data provenance facilitates uncertainty quantification, as surrogate models can explicitly capture variability and confidence intervals informed by the historical records. This synergy between blockchain-backed traceability and surrogate modeling thus provides a principled pathway for the traceX framework to performance assessment, risk analysis, and continual improvement of critical decision-making systems.

Prediction System Evolution Instead of considering at each algorithm individually, the system can be viewed holistically from the perspective of predicting it evolution over time, given a particular starting "event" – an adversary launch from a given launch site. In this context, traceX explores the use of Reinforcement Learning (RL) (Sec. 5), for the sequential decision-making problem that a SBMS has to address.

Operator Integration Finally, all this information needs to presented to the operator who must be able to interrogate the system, be able to understand the system state at a given moment, and interact with it through well-established decision frameworks such as the Observe Orient Decide Act (OODA) loop. In traceX, we explore two such framework: firstly, a Large Language Model (LLM) based textual interface, that is kept updated using the latest information stored on the chain. Such a system enables natural language interactions with system without bounding the operation to a specific user-interface (UI) as is done in traditional system. Secondly, to aid in decision making, information is summarize using a flexible OODA loop framework. The operator is free to use or dismiss the information shown to them, which serves as feedback to the RL algorithms.

Commercial Space Applications The principles and technologies developed for defense space operations have significant applicability to commercial space domains. The exponential growth in commercial satellite constellations, space traffic management challenges, and multi-stakeholder coordination requirements create similar needs for trusted data exchange, transparent decision-making, and automated de-confliction. Commercial space traffic management systems, satellite constellation coordination, and debris avoidance operations can benefit from the same benefits of traceX (See [3] for applications of traceX to a commercial SDA marketplace): blockchain-verified traceability, AI-driven prediction, and operator interface technologies developed for military applications.

The remainder of this paper is organized as follows: Section 2 motivates quantum-resistant blockchain requirements for maintain future trust; Section 3 demonstrates traceX's traceability infrastructure and applications; Section 4 presents surrogate modeling for quality assurance; Section 5 explores RL-based state evolution and explores causal analysis; Section 6 addresses operator interactions; and Section 7 extends explore the uses of traceX in commercial non-military settings.

2. QUANTUM RESISTANT BLOCKCHAINS

The long-term security of blockchain systems faces significant threats from advances in quantum computing. Current public-key cryptographic schemes—including elliptic curve cryptography and RSA that underpin digital signatures and key management—are vulnerable to Shor's algorithm [27], which can efficiently derive private keys from public keys. Additionally, Grover's algorithm [16] threatens the security margins of hash-based mechanisms used in proof-of-work and consensus protocols. Although cryptographically relevant quantum computers do not yet exist, blockchain's immutable nature creates a "Harvest Now, Decrypt Later" (HNDL) risk [20]. Adversaries can collect encrypted data today with the intention of decrypting it once quantum capabilities mature, fundamentally undermining blockchain integrity and trust models. In permissioned defense contexts like the WA system, this risk extends to smart contracts, cross-chain bridges, and consensus protocols that rely on cryptographic signatures for security. Given that blockchain systems and archival nodes operate on decade-long timelines with address lifetimes spanning years, proactive adoption of quantum-resistant cryptography is essential for maintaining long-term system reliability and security guarantees.

2.1 Current Standards and Implementation

Recent developments in post-quantum cryptography have transitioned from research to practical standardization. In 2024, the U.S. National Institute of Standards and Technology (NIST) officially approved the first post-quantum cryptography standards—encryption algorithms designed to protect digital systems against future quantum attacks [7]. These standards provide three primary cryptographic primitives:

- **Key Exchange**: ML-KEM (FIPS 203) [17] enables secure key establishment between parties
- **Digital Signatures**: ML-DSA (FIPS 204) [18] provides identity verification and non-repudiation
- **Hash-Based Signatures**: SLH-DSA (FIPS 205) [19] offers enhanced security robustness with larger signature sizes as a trade-off

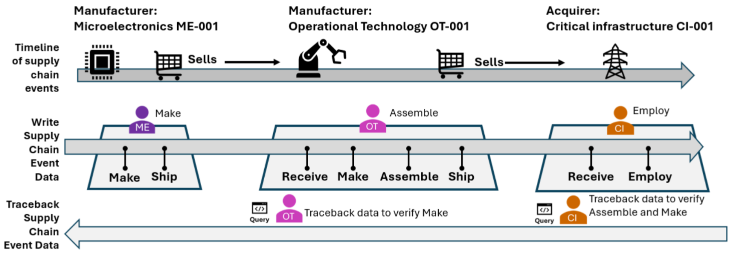

Manufacturer:
Microelectronics ME-001
Manufacturer:
Operational Technology OT-001
Acquirer:
Critical infrastructure CI-001
Timeline of supply chain events
Sells
Sells
Write Supply Chain Event Data
Make
ME
Make
Ship
Assemble
OT
Receive
Make
Assemble
Ship
Employ
CI
Receive
Employ
Traceback Supply Chain Event Data
Query
OT
Traceback data to verify Make
Query
CI
Traceback data to verify Assemble and Make

Fig. 2: Challenges of Component or Assembly Verification Across Stakeholder Tiers (Figure from [23])

NIST also endorses hash-based alternatives like XMSS and LMS [4] for resource-constrained environments or situations requiring secure update verification. These algorithms have moved beyond theoretical constructs—the Quantum Resistant Ledger (QRL) has successfully deployed XMSS signatures on mainnet since 2018, demonstrating viable post-quantum blockchain operations with complete end-to-end post-quantum accounts, wallets, and node infrastructure.

2.2 Implementation Strategy for Welder's Arc

For the WA system, we are implementing a hybrid cryptographic approach that balances security requirements with operational constraints. Our implementation strategy addresses the specific needs of real-time space domain awareness applications:

- **Transaction Security**: We employ ML-DSA signatures for transaction validation, providing efficient verification suitable for high-throughput message processing. ML-DSA offers favorable signature sizes and verification speeds compared to hash-based alternatives while maintaining strong security guarantees against quantum attacks.
- **Message Integrity**: For blockchain commitments of Kafka messages, we utilize hash-based schemes that ensure long-term integrity even under potential cryptanalytic advances. This dual approach allows us to optimize for different use cases within the same system architecture.
- **Performance Considerations**: The combination optimizes signature size and verification speed—critical factors for real-time applications where millisecond-level response times are essential for space situational awareness and threat assessment.

3. DATA AND DECISION TRACEABILITY FOR EXPLAINABILITY AND AUDIT

The Welder's Arc Battle Management System (BMS) processes vast amounts of data through sophisticated algorithms to provide critical recommendations to operators. Understanding how the system evolves through the kill-chain and maintaining awareness of decision pathways is paramount for operational effectiveness. To address this challenge, we present a comprehensive data and decision traceability system for Welder's Arc that provides auditable records of data flows, processing steps, and decisions while contributing to a unified common operational picture.

3.1 Traceability Framework and Requirements

In a dispersed, distributed, and decentralized domain such as manufacturing supply chain, the only source of trust is via peer-to-peer methods such as with the traceability method in NIST IR 8536 Meta-Framework [23]. For example, in Fig. 2 below, a critical infrastructure grid operator may need to trust a key microelectronics component. In this case, trust can only be established one hop at a time, where supply chain events are recorded from chip to grid operator – from microelectronics to operational technology to the grid operator. The Meta-Framework establishes a traceability chain whereby the records are linked and each record link assures data integrity of that hop. This enables trusted traceback from grid operator to the chip manufacturer.

This method can be adapted to space data, by looking at the use of space data as a supply chain of dispersed, distributed, and decentralized stakeholders. In the case of WA, the stakeholders are independently providing algorithms which consume and produce information: input data → decisions → actions → outcomes. The ability to trace from actions and outcomes back to input data forms a trusted data layer that forms the foundation to use AI. Modern Battle Management Systems require robust traceability mechanisms: backward in time, it ensures accountability and data integrity, forward in time, traceability provides operators with visibility into the system's evolution from initial sensor observations through the complete kill-chain to final exit criteria.

Data and decision traceability requires comprehensive documentation of data life cycle, including origin, transformation, movement, and utilization, as well as the rationale behind decisions using that data. This systematic approach creates verifiable chains of evidence that enable reconstruction of data provenance and decision justification throughout system operation. Figure 2 illustrates the fundamental traceability concept, highlighting the ability to trace data both forward and backward through processing pipelines. In the space context, by providing a universal understanding of **how and when decisions are made**, traceability empowers operators to understand and trust system recommendations.

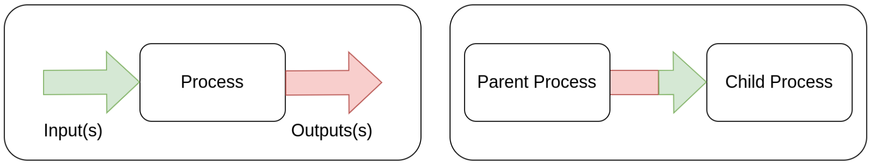


Fig. 3: Traceability captures **Left:** instantaneous inputs and outputs for each algorithm, and **Right:** causal relationships between processes by tracking when one process (child) consumes other's output (parent). These principles enable modeling complex systems with hundreds of algorithms and intricate data exchange patterns.

The traceability framework comprises two main components shown in Figure 3. First, each process must track captured information (inputs) and generated outputs with appropriate metadata. Second, algorithms connect through input-output relationships to provide complete data flow visibility and causal reasoning capabilities.

3.2 Implementation in Welder's Arc

Space Protocol has collaborated with MITRE Corporation since May 2024 to implement NIST IR 8536 "Supply Chain Traceability: Manufacturing Meta-Framework" [23]. This comprehensive framework enhances traceability across manufacturing supply chains by improving product provenance, pedigree, and transparency. The Meta-Framework organizes, links, and enables querying of traceability data across manufacturing supply chains (Figure. 2). We adapt principles derived from [23] to Welder's Arc to achieve end-to-end introspection, auditing, and replay capabilities.

The message bus architecture enables comprehensive data and decision traceability throughout the system. Each message follows well-defined schemas and provides detailed information about data content, origin, and generating processes (see Listing 1). This creates a chronological log of all data transformations and decisions over time.

TraceX builds upon the Kafka message bus using dedicated traceability sections in message headers. The system distinguishes between internal and external dependencies:

- **Internal dependencies**: Messages generated within WA, capturing data flow between algorithms and components using unique parent message identifiers, data types, and integrity hashes
- **External dependencies**: Data from external sources (APIs, databases like UDL) recorded with resource links, access parameters, and data integrity hashes

Figure 4 demonstrates this approach where a UCT processing algorithm consumes tracks and observations to generate four distinct outputs. The message bus schema enables comprehensive metadata collection for each input and output, allowing any decision to be traced back through its complete ancestry chain. It is being developed through a collaboration between Millennial Software and Pacific Northwest National Laboratory (PNNL), with Space Protocol contributing to the design of the message headers that enable the traceability.

3.3 Case Study: Potential Threat Assessment

Figure 5 shows **traceX** output where a "Potential Threat" assessment is visualized through connected messages and subsystem components (represented by different colors). Each circle represents a message resulting from various

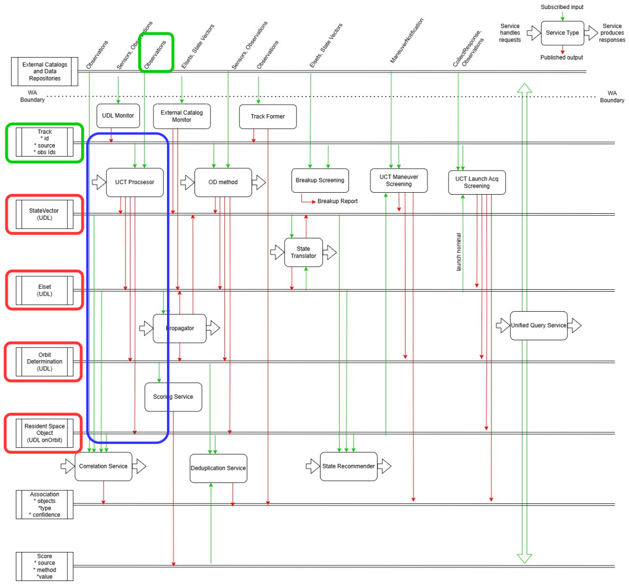


Fig. 4: Traceability data capture: Algorithm inputs (green) and outputs (red) are captured via header schemas with metadata for a UCT processing algorithm (blue).

processing steps, including external data sources. This complete process visibility enables operators to understand both **what** the system is doing and **why** it recommends specific actions.

The traceability chain reveals the complete decision pathway: initial sensor observations flow through data fusion algorithms, which feed threat assessment models, ultimately producing the threat alert. Operators can examine any step in this chain to understand the reasoning behind the final assessment, enabling informed decision-making and system trust.

3.4 Applications and Operational Benefits

TraceX provides foundational capabilities that significantly enhance decision-making, system reliability, and continuous improvement within the BMS. By providing unparalleled visibility into the data, processes, and decisions that generate actionable insights, traceability empowers operators, analysts, and commanders to not only understand **what** is happening in the system at a given moment, but also enables **why** the system is recommending a particular course of action.

- **Auditing and Verification**: Complete, auditable records of decision-making processes enable verification that systems operate correctly and within operational parameters. Post-incident analysis can reconstruct entire event sequences for lessons learned and compliance verification.
- **Data Provenance and Quality Assurance**: Meticulous tracking of data origins and transformations enables validation, bias identification, and quality improvement. Operators can assess data reliability and identify potentially compromised or degraded information sources.
- **Rapid Diagnosis and Root Cause Analysis**: When errors occur, traceability facilitates immediate diagnosis by providing clear views of data movement through the system. This capability reduces mean time to resolution and enables proactive system maintenance.
- **Dynamic Analysis and Simulation**: The system enables replay of past scenarios and creation of new ones with modified parameters. This “what-if” capability improves preparedness and enables operators to assess the impact of changing conditions or data accuracy.

3.5 System Architecture and Scalability

Transforming the prototype into a robust, deployable solution requires three key integrated components:

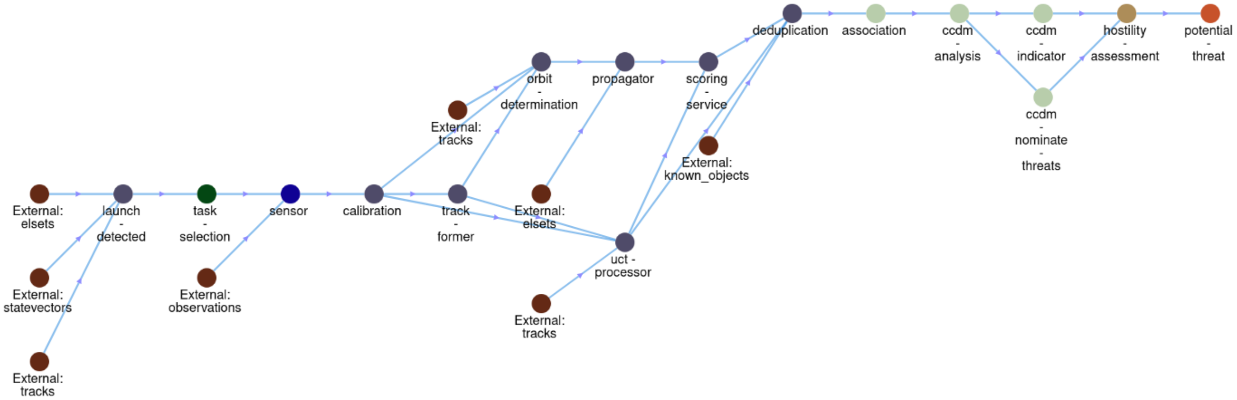


Fig. 5: Traceability visualization showing algorithm interdependence and message flows. Each color represents a different WA subsystem, with complete backward tracing from threat assessment to triggering events.

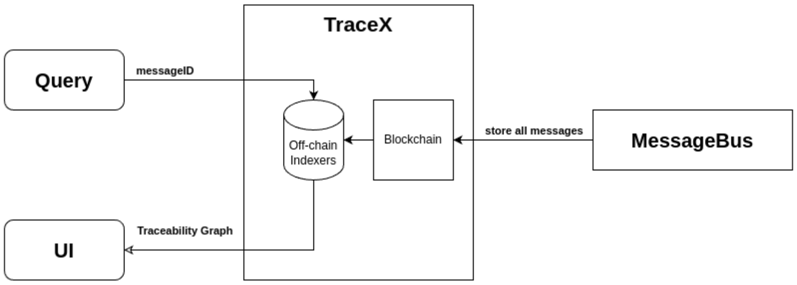


Fig. 6: System architecture of **traceX** showing integration with message bus, on-chain storage which are then indexed off-chain for efficiency, and user interface components.

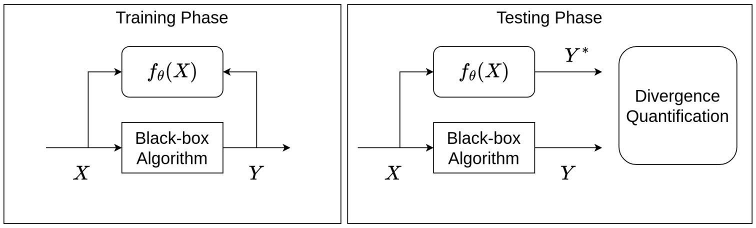


Fig. 7: Surrogate Models for testing performance of a black-box algorithm. In the test phase, a mapping from inputs X to output Y is learned via a surrogate model which is then be used at test time to quantify the performance of the black-box.

- **Message Bus Integration**: Full ingestion of the Kafka message bus enables automatic data capture, providing comprehensive real-time system behavior visibility without requiring algorithm modification.
- **Graph Database Architecture**: A specialized graph database efficiently stores and manages complex metadata from the message bus, enabling real-time reconstruction of data flows and advanced querying for auditing and performance monitoring. The database design supports millions of nodes with sub-second query response times.
- **User Interface**: An intuitive user interface makes traceability data accessible and actionable, incorporating operational picture concepts to provide clear system state visibility and introspection capabilities. The interface integrates with blockchain records to provide tamper-evident audit trails.

4. QUALITY ASSURANCE VIA ML-BASED BENCHMARKING

The combination of trusted data stored on quantum-resistant blockchain and comprehensive traceability creates a foundation for systematic quality assurance through machine learning-based benchmarking. This section presents how surrogate model – data-driven approximations of complex algorithm – within traceX can leverage blockchain-verified input-output relationships to provide continuous performance monitoring, uncertainty quantification, and anomaly detection across the distributed WA system.

4.1 Surrogate Models

A surrogate model is a lightweight, data-driven approximation of a more complex or opaque system. Rather than relying on knowledge of the internal mechanisms of a “black-box” algorithm, the surrogate is constructed entirely from observed input–output relationships. For a true system that maps inputs X to outputs Y through an unknown function $f(\cdot)$, the surrogate approximates this mapping with a model $\hat{f}_\theta$, parameterized by θ [8][13]. In this sense, the surrogate acts as a proxy for the original system, providing a tractable means of studying, predicting, and evaluating its behavior.

In the SDA TAP Lab context, where algorithms are often proprietary, expensive to evaluate, or developed by different vendors, surrogate models provide several critical capabilities. Figure 7 illustrates the two-phase approach: training on historical data to learn expected behavior patterns, then monitoring real-time operations for deviations. The motivation for surrogate modeling in WA stems from the multi-vendor environment where individual algorithms are “black boxes”. Surrogate models enable **continuous monitoring** by comparing actual algorithm outputs against predicted outputs. When significant deviations occur, this signals potential performance degradation, adversarial attacks, or off-specification operation [5, 21]. Modern probabilistic surrogate models also provide **uncertainty quantification**, distinguishing between rare but valid behaviors and genuine anomalies.

4.2 Implementation Approaches

Surrogate models can employ various machine learning architectures depending on the complexity and characteristics of the target algorithm. Classical approaches [13, 8] include polynomial regression, response surface methods, and Gaussian processes, which provide smooth interpolations and work effectively with limited data.

For more complex algorithms processing high-dimensional data, deep learning architectures offer greater expressiveness. Neural networks, including multilayer perceptrons, convolutional networks, and transformer architectures, can

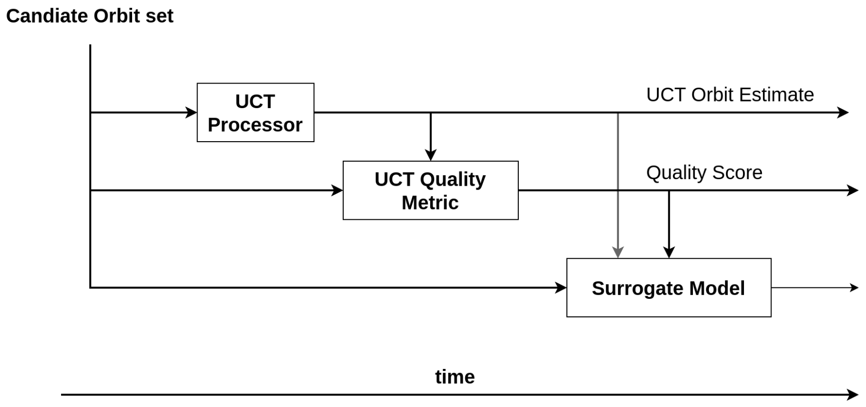


Fig. 8: Schematics of a UCT processor and the subsequent quality metric. The input `Candidate Orbit Set` is used by a UCT Processor to generate an estimate for the UCT. A subsequent quality metric compares the `Candidate Orbit Set` and `UCT Orbit estimate` to determine a `Quality Score`. The surrogate model approximate the UCT Quality Metric.

capture complex nonlinear relationships in sequential or structured data [21]. When combined with probabilistic formulations through Bayesian methods, ensemble learning, or evidential regression, these models capture both epistemic uncertainty (limited system knowledge) and aleatoric uncertainty (inherent output variability) [15, 11].

To ensure reliability, we employ conformal prediction techniques [26, 25, 2] that provide statistically valid prediction intervals regardless of the underlying model architecture. This enables the surrogate to quantify confidence bounds on its predictions, supporting principled decision-making about when to trust algorithm outputs.

4.3 Blockchain-Enabled Quality Monitoring

The blockchain infrastructure provides several advantages for surrogate model development and deployment:

- **Data Integrity**: Cryptographic hashing of input-output pairs ensures training data hasn't been tampered with, providing confidence in surrogate model accuracy. The immutable record prevents adversaries from poisoning historical training data.
- **Provenance Verification**: Traceability metadata enables filtering training data based on operational context, data quality, or upstream algorithm performance, improving surrogate model robustness.
- **Continuous Learning**: As new blockchain-verified examples become available, surrogate models can be incrementally updated while maintaining performance baselines, enabling adaptation to evolving operational conditions.

The surrogate functions as a "shadow copy" of the target algorithm: it does not replicate internal workings but approximates observable behavior based on historical patterns. During normal operation, actual outputs remain consistent with surrogate predictions. However, when algorithms begin degrading, drifting, or encountering unforeseen conditions, the divergence between surrogate predictions and actual outputs provides early warning of potential issues (Fig. 7). This divergence provides a principled mechanism for performance characterization, allowing the surrogate to function not only as a predictive tool but also as a diagnostic instrument for ensuring reliability and safety.

4.4 Implementation at the SDA Tap Lab: Uncorrelated Track Processing

We demonstrate surrogate modeling with the Uncorrelated Tracks (UCT) processing pipeline, which handles observations that don't correspond to catalogued Resident Space Objects (RSOs). Such tracks may indicate concealment, deception, or maneuvering activities, making accurate processing critical for data superiority.

The UCT processor ingests track observations and estimates orbital parameters through data association algorithms that assign observations to corresponding RSOs. A dedicated quality metric service evaluates each generated orbit estimate, producing quality scores for downstream fusion algorithms.

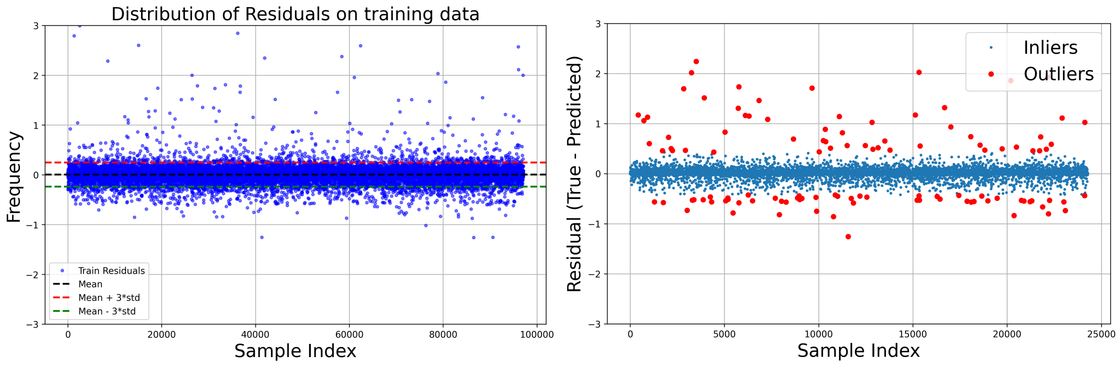


Fig. 9: Surrogate model for `Quality Score`: **Left)** Using an ensemble of networks, the surrogate model learns to predict the quality score using historic data. The epistemic uncertainty quantified using the residuals of the network. **Right)** Using statistical tests based on the model's epistemic uncertainty, performance degradation (red) can be distinguished from normal operating conditions.

Our surrogate model implementation follows this process:

- **Training Data Collection**: Historical UCT processing results, including input observations, estimated orbits, and quality scores, are extracted from blockchain records with verified provenance (Fig. 8).
- **Surrogate Architecture**: A neural network ensemble learns to predict orbit quality scores from input track characteristics, orbital parameters, and processing metadata. The ensemble approach provides uncertainty estimates by measuring prediction variance across individual models. The estimated variance is depicted in Fig. 9, where training residuals along with the estimated model uncertainty is depicted. This represents the Epistemic uncertainty – lack of knowledge about a system or phenomenon that could, in principle, be resolved with more information or data – of the system and is used to perform statistical tests on new data.
- **Performance Monitoring**: During real-time operations, the surrogate model predicts expected quality scores for each UCT processing result. Significant deviations, as measure via statistical tests, trigger alerts for manual review or automatic algorithm reconfiguration. These variations are marked as "outliers" in Fig. 9.

4.5 Experimental Results and Validation

Initial validation demonstrates surrogate model effectiveness on both synthetic and operational data. Figure 9 shows a controlled experiment where a neural network surrogate learns to predict the UCT Quality score, successfully capturing both the underlying trend and uncertainty bounds.

For the UCT quality metric application, we trained surrogate models on six months of operational data comprising over 100,000 processed tracks. The surrogate achieved 0.94 correlation with actual quality scores. The system was tested by providing adversarial inputs and was able to successfully identify 94% of anomalous processing results, as demonstrated in Fig. 9. Performance monitoring revealed several operational benefits:

- **Early Anomaly Detection**: On the real data, the surrogate identified performance degradation 15-30 minutes before traditional threshold-based monitoring, enabling proactive system maintenance.
- **Algorithm Validation**: Surrogate predictions helped validate new UCT processing algorithms by comparing their outputs against established behavioral patterns, quantifying periods of nominal operation.
- **Uncertainty-Aware Operations**: Probabilistic predictions enabled downstream algorithms to weight UCT results based on confidence levels, providing an additional layer of informational trust, improving overall system robustness.

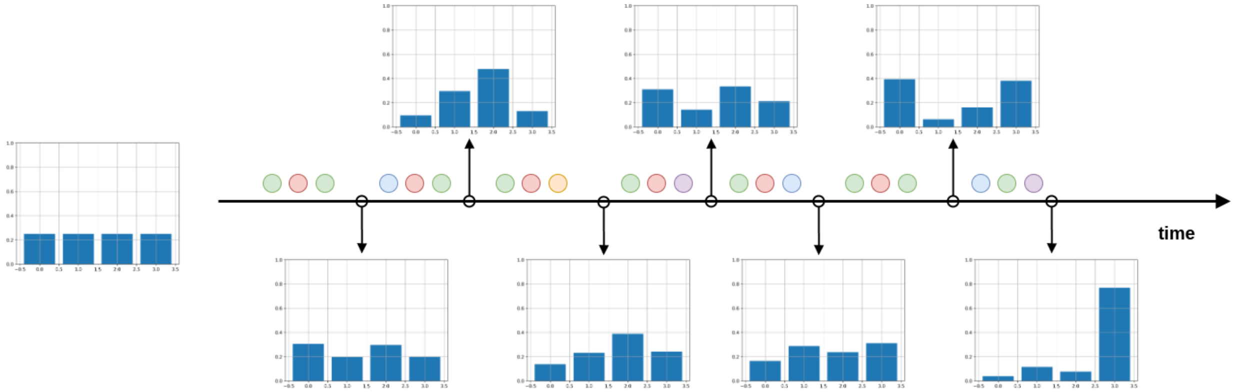


Fig. 10: A schematic of system evolution using RL: Starting with an uninformed prior over the courses-of-action (COAs), the system evolves are new information (represented by circles) is integrated into the system. The probability distribution over the COAs converges to a stable value after taking into account more information.

4.6 Integration with System Architecture

The surrogate modeling framework integrates seamlessly with existing WA infrastructure:

- **Message Bus Integration**: Surrogate models subscribe to relevant Kafka topics, enabling real-time prediction and monitoring without modifying existing algorithms.
- **Blockchain Logging**: Surrogate predictions and confidence scores are logged to the blockchain, creating an auditable record of quality assurance decisions and enabling retrospective analysis.
- **Traceability Support**: When surrogates detect anomalies, operators can use the traceability system to investigate root causes and understand the full context of algorithmic decisions.

This integrated approach transforms individual algorithm monitoring into systematic quality assurance across the entire battle management pipeline, supporting both operational reliability and continuous improvement efforts.

5. PREDICTION: RL-BASED EVENT PROJECTION IN TIME AND CAUSAL REASONING

Modern Battle Management Systems comprise complex algorithmic pipelines where sequential processing stages create inter-dependencies that directly influence system performance. Rather than optimizing individual algorithms in isolation, we propose modeling entire system evolution over time to predict how events unfold given initial conditions such as adversary launches or suspicious space activities.

Reinforcement Learning provides a principled framework for this challenge by modeling environment dynamics where actions (algorithm selections and configurations) lead to state transitions (updated pipeline outputs) and eventual rewards (mission success). This formulation captures both sequential dependencies between algorithms and the long-term impact of local decisions on global objectives.

5.1 Mathematical Framework

We model the algorithmic pipeline as a directed acyclic graph (DAG) $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ where nodes represent algorithms and edges represent data dependencies. At decision step t, the state encodes current execution status:

$$s_t = f_\Psi(\mathscr{G}, \mathscr{V}_t, y_{\mathscr{V}_t})$$

where $\mathscr{V}_t$ represents executed algorithms and $y_{\mathscr{V}_t}$ their outputs. Actions

$$a_t = (m_t, x_t)$$

combine discrete algorithm selection m_t with continuous parameter configuration x_t.

The reward function balances mission effectiveness with operational costs:

$$R = M(y_{\text{sink}}) - \lambda_c \sum_{t=0}^{T-1} c(m_t, x_t) + \lambda_e \sum_{t=0}^{T-1} \Delta M_t$$

where $M(\cdot)$ represents terminal quality metrics, $c(\cdot)$ encodes computational costs, and ΔM_t provides intermediate performance shaping. Action masking ensures feasibility by setting

$$\pi_{\theta_m}(m \mid s_t) = 0$$

for unavailable algorithms. Complete mathematical formulation is provided in Appendix A.

5.2 Soft Actor-Critic for Battle Management

We employ Soft Actor-Critic (SAC), which combines off-policy learning with maximum entropy reinforcement learning. SAC augments the standard RL objective with an entropy term, encouraging exploration and preventing premature convergence:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T} r_t + \alpha \mathscr{H}(\pi(\cdot \mid s_t)) \right]$$

The entropy regularization proves particularly valuable for battle management where action spaces may be large, stochastic, or non-stationary. This ensures continued exploration of alternative strategies even after discovering effective policies, maintaining adaptability to evolving threats.

5.3 Progressive Prediction Capability

RL agents learn value functions that estimate long-term returns from partial trajectories, enabling intermediate predictions before complete pipeline execution. The learned state-value function $V(s_t)$ provides running estimates of expected mission success, with estimates refined as upstream algorithms produce outputs and system state evolves. This progressive prediction capability enables early warning when predicted outcomes fall below acceptable thresholds, extending operator decision time and transforming reactive battle management into proactive threat mitigation.

5.4 Towards Causal Learning

Machine learning approaches often optimize for correlations within observed data, which may yield strong predictive performance but fail to provide robust insights into **why** particular outcomes occur. In contrast, causal learning explicitly models cause–effect dependencies through frameworks such as Structural Causal Models (SCMs) and do-calculus, enabling reasoning about interventions and counterfactual scenarios [22]. This shift from correlation to causation is critical in safety-critical domains, where operators must be able to interpret, trust, and act upon system recommendations. When integrated into traceable data and decision pipelines, causal learning enhances accountability by providing auditable causal chains of events. For example, blockchain-enabled traceability ensures immutability of the data and decisions logged across the kill-chain, while causal models reveal how specific inputs or interventions influenced downstream outcomes. Together, these mechanisms allow operators and auditors to not only reconstruct the sequence of decisions but also to evaluate the reasons behind them. This dual perspective is especially valuable when investigating anomalies, system drift, or adversarial interference, as causal dependencies provide a structured explanation for deviations beyond what correlation-based models can supply.

5.5 Validation Through Related Applications

While our space battle management implementation is being developed, extensive evidence from related applications demonstrates RL's effectiveness for similar application scenarios. In particular, multi-agent RL systems have achieved success rates exceeding 85% in complex coordination tasks. The MW-MADDPG framework [28] for collaborative UAV swarms, for example, demonstrated superior performance in combat scenarios with meta-learning approaches showing enhanced adaptability to new mission environments. Defense-oriented applications [6] maintain effectiveness against adaptive adversaries—critical for space domain operations where opponents modify behavior patterns. Real-time performance requirements are validated through air traffic control systems using multi-agent RL, which achieve

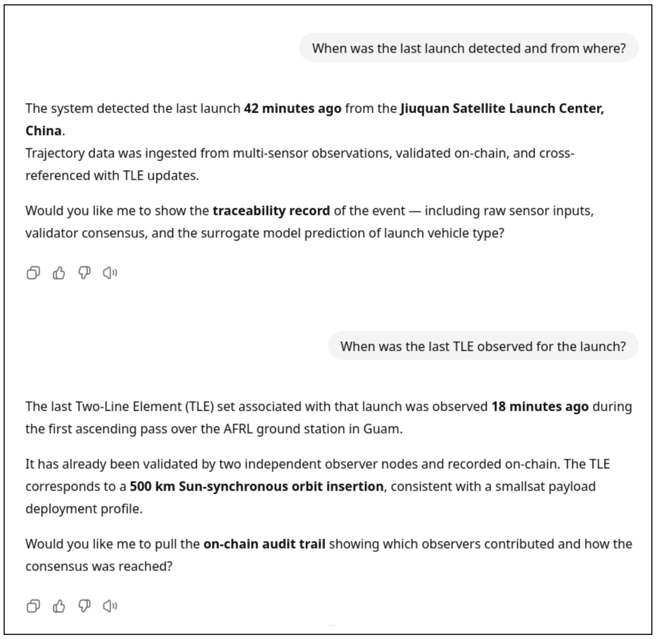
99.97% conflict resolution success with decision latencies under 100 ms [24]. The Soft Actor-Critic algorithm specifically demonstrates 2–5× sample efficiency improvements over alternatives while maintaining stability across diverse environments [12]. These results from analogous domains provide compelling evidence that RL-based approaches can achieve the performance, timing, and reliability requirements necessary for effective SBMS systems.

6. OPERATOR INTERFACES

The AI- and blockchain-based infrastructure presented in this work accumulates rich knowledge about data flows, individual algorithms, and system-wide behavior. However, effective methods for conveying operator-relevant information must be integrated to maximize system utility. To this end, traceX explores two complementary approaches for operator awareness and decision-making support.

6.1 LLM-based Unified chat interface

An LLM-based chat interface acts as a cognitive bridge between human operators and complex battle management systems, enhancing usability, operational speed, and trust while making the system adaptable to diverse operators and mission contexts (Fig. 11). When properly safeguarded with controls for hallucination, information assurance, and adversarial manipulation, such interfaces can substantially accelerate decision-making.



When was the last launch detected and from where?

The system detected the last launch **42 minutes ago** from the **Jiuquan Satellite Launch Center, China**.
Trajectory data was ingested from multi-sensor observations, validated on-chain, and cross-referenced with TLE updates.

Would you like me to show the **traceability record** of the event — including raw sensor inputs, validator consensus, and the surrogate model prediction of launch vehicle type?

When was the last TLE observed for the launch?

The last Two-Line Element (TLE) set associated with that launch was observed **18 minutes ago** during the first ascending pass over the AFRL ground station in Guam.

It has already been validated by two independent observer nodes and recorded on-chain. The TLE corresponds to a **500 km Sun-synchronous orbit insertion**, consistent with a smallsat payload deployment profile.

Would you like me to pull the **on-chain audit trail** showing which observers contributed and how the consensus was reached?

Fig. 11: An operator interacts with the LLM-based chat for introspection and situational awareness. Natural language interactions enables the operator to ask for pertinent information. See Fig. 13 for additional system interrogation.

Real-Time Information Integration: To maintain operational relevance, the LLM interface requires continuous access to dynamic, context-specific data rather than periodic retraining. This includes near-real-time operational inputs such as sensor observations, fused tracks, anomaly detections, and prediction outcomes, along with decision records capturing issued commands, associated rationales, and resulting effects. Static but essential reference knowledge includes rules of engagement, doctrine, system ontologies, and coalition access policies. Each data element carries provenance metadata—timestamps, confidence levels, and blockchain transaction identifiers—enabling verification of model outputs against an immutable audit trail. Retrieval-augmented generation (RAG) pipelines query indexed blockchain events and stream updated situational data into the LLM's context window, as shown in Figure 12. Operator feedback and correction signals refine retrieval quality and prompt construction, allowing the LLM to maintain current situational awareness while providing accountable, traceable, and policy-constrained decision support (Fig. 13).

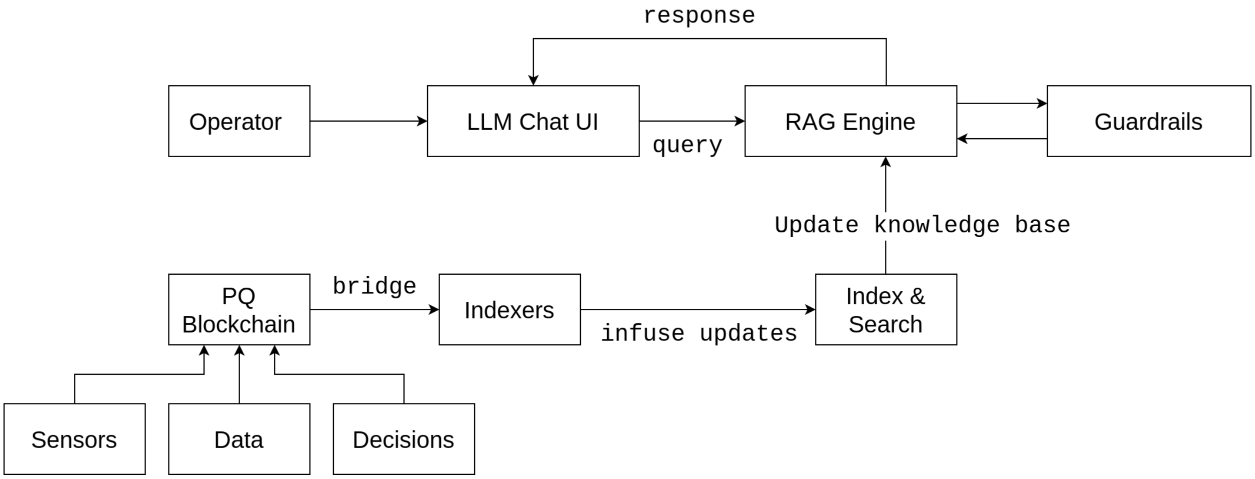
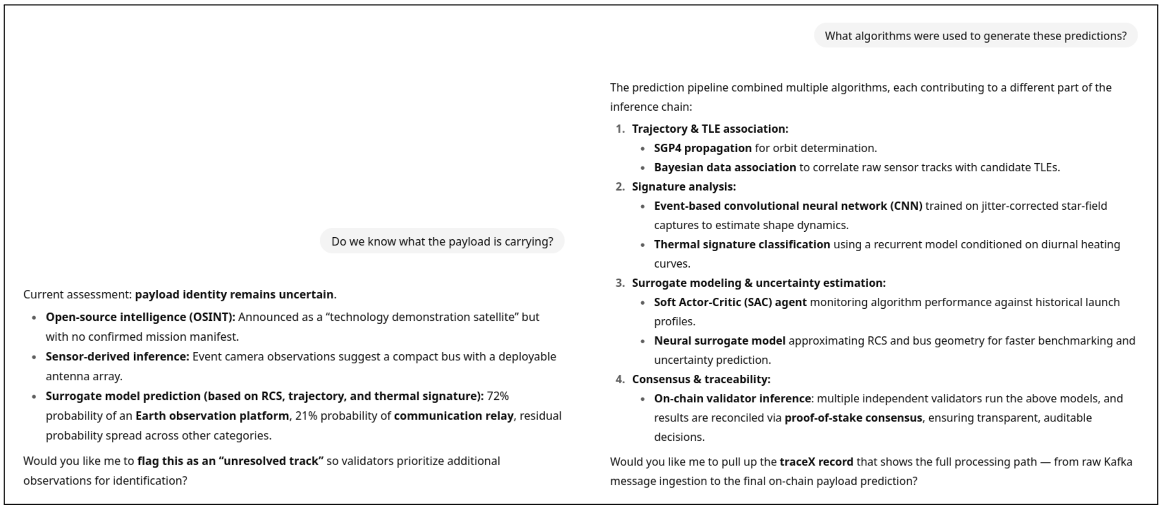


Fig. 12: A schematic of the LLM-based chat interface powered by the PQ-blockchain. Latest data from the chain flows through indexers and is made available for the RAG engine to include in the next query, if relevant.



Do we know what the payload is carrying?

Current assessment: **payload identity remains uncertain**.

- **Open-source intelligence (OSINT):** Announced as a “technology demonstration satellite” but with no confirmed mission manifest.
- **Sensor-derived inference:** Event camera observations suggest a compact bus with a deployable antenna array.
- **Surrogate model prediction (based on RCS, trajectory, and thermal signature):** 72% probability of an **Earth observation platform**, 21% probability of **communication relay**, residual probability spread across other categories.

Would you like me to **flag this as an “unresolved track”** so validators prioritize additional observations for identification?

What algorithms were used to generate these predictions?

The prediction pipeline combined multiple algorithms, each contributing to a different part of the inference chain:

1. **Trajectory & TLE association:**
   - **SGP4 propagation** for orbit determination.
   - **Bayesian data association** to correlate raw sensor tracks with candidate TLEs.
2. **Signature analysis:**
   - **Event-based convolutional neural network (CNN)** trained on jitter-corrected star-field captures to estimate shape dynamics.
   - **Thermal signature classification** using a recurrent model conditioned on diurnal heating curves.
3. **Surrogate modeling & uncertainty estimation:**
   - **Soft Actor-Critic (SAC) agent** monitoring algorithm performance against historical launch profiles.
   - **Neural surrogate model** approximating RCS and bus geometry for faster benchmarking and uncertainty prediction.
4. **Consensus & traceability:**
   - **On-chain validator inference**: multiple independent validators run the above models, and results are reconciled via **proof-of-stake consensus**, ensuring transparent, auditable decisions.

Would you like me to pull up the **traceX record** that shows the full processing path — from raw Kafka message ingestion to the final on-chain payload prediction?

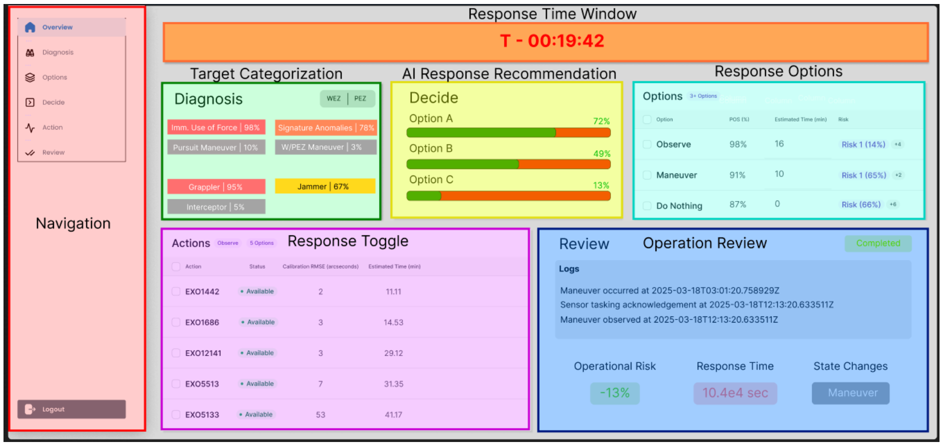
Fig. 13: The back-end of the chat interface has full-system visibility. Queries incorporate rich reasoning and traceX outputs when required or requested by the operator.

Operational Benefits The LLM interface significantly enhances usability by enabling natural language interaction with the battle management system. Instead of navigating complex dashboards, operators query and refine information conversationally, receiving concise summaries that remain verifiable against the blockchain logs. This improves trust, accelerates decision-making cycles, and ensures continuity across shifts and coalition partners. The interface also serves as a training aid, allowing less experienced personnel to query doctrines, rules of engagement, and system outputs in natural language. These capabilities collectively reduce cognitive load, increase operational tempo, and strengthen collaboration, making human-machine teams more effective under time-critical conditions.

6.2 OODA loop integration

The Observe-Orient-Decide-Act (OODA) loop framework, originally developed by U.S. Air Force Colonel John Boyd, describes effective decision-making in rapidly changing environments. The framework's power lies in its speed and adaptability—by cycling through the loop faster than adversaries, operators can disrupt opponent decision-making, maintain initiative, and sustain operational advantages.

An integrated interface combining blockchain verification and machine learning insights empowers operators by presenting verifiable, AI-driven information in real time. Machine learning algorithms transform raw sensor streams into



Option	POS (%)	Estimated Time (min)	Risk
Observe	98%	16	Risk 1 (14%) +4
Maneuver	91%	10	Risk 1 (65%) +2
Do Nothing	87%	0	Risk (66%) +6

Action	Status	Calibration RMSE (arcseconds)	Estimated Time (min)
EXO1442	• Available	2	11.11
EXO1686	• Available	3	14.53
EXO12141	• Available	3	29.12
EXO5513	• Available	7	31.35
EXO5133	• Available	53	41.17

Logs

```
Maneuver occurred at 2025-03-18T03:01:20.758929Z
Sensor tasking acknowledgement at 2025-03-18T12:13:20.633511Z
Maneuver observed at 2025-03-18T12:13:20.633511Z
```

Operational Risk	Response Time	State Changes
-13%	10.4e4 sec	Maneuver

Fig. 14: Integrated OODA loop interface showing how system predictions inform operator decision-making. Highlighted sections perform specific OODA functions with AI-generated insights and blockchain-verified data.

categorized targets, probability-weighted action options, and quantified risk estimates, displayed in clear, decision-focused layouts(see Fig. 14). Each phase of the OODA loop benefits from AI-enabled insights:

- **Observe**: Real-time sensor fusion and anomaly detection provide comprehensive situational awareness with confidence bounds and data provenance tracking.
- **Orient**: Historical pattern analysis and threat assessment algorithms contextualize current observations within broader operational frameworks, supported by blockchain-verified precedents.
- **Decide**: RL-based recommendation systems suggest optimal course of action with predicted outcomes and resource requirements, enabling informed decision-making under uncertainty.
- **Act**: Automated execution monitoring tracks decision implementation and outcomes, feeding results back into the system for continuous learning and improvement.

Complementary Interface Design: The structured OODA dashboard and LLM chat interface serve complementary roles within the battle management system. The dashboard provides standardized, high-density visualizations of machine learning outputs including target categorizations, probability assessments, and operational logs, ensuring consistent operational pictures across teams. At the same time, the chat interface adds flexible, human-centered capabilities for explanation, contextualization, and historical reasoning. Operators can interrogate the system in natural language to understand recommendation rationales (Figs. 11, 13), explore related past events, or request summaries tailored to their current operational focus. Every query and response links back to blockchain-verified data, maintaining traceability and accountability. This dual-interface approach creates a transparent, accountable, and adaptive decision-support environment where the dashboard ensures standardized information presentation while the LLM chat enhances usability, trust, and collaboration through operator-friendly explanations of complex system outputs.

7. COMMERCIAL SPACE APPLICATIONS

The technologies developed for defense space battle management directly address challenges in the rapidly expanding commercial space sector. Defense and civil/commercial space applications have always been intertwined. Innovation in the defense sector drives growth in the commercial sector, and it is important to see the future value and applicability early in the development cycle. As mega-constellations deploy thousands of satellites, the need for trusted coordination, transparent decision-making, and automated conflict resolution mirrors military space operation challenges.

7.1 Space Traffic Management and Multi-Stakeholder Coordination

Commercial space operations face increasing space traffic management challenges as orbital debris and satellite populations grow. Traditional centralized tracking systems lack transparency when coordination failures occur between operators or regional SSA providers. Blockchain-verified traceability eliminates disputes by providing immutable records of observations, maneuver decisions, and coordination communications. All operators access the same verified data, ensuring accountability for collision avoidance decisions. Surrogate modeling enables continuous monitoring of orbital prediction accuracy, automatically flagging divergent models and requesting updated ephemeris data.

While this prototype is designed for battle management, the underlying principles and technologies being demonstrated have tremendous potential for the commercial space sector. This approach has the potential to allow the civil space community to not only catch up to other industries in adopting distributed ledger technology to manage information supply chains, but offers the potential to leapfrog those approaches considering quantum-resistant blockchain from initial implementation and to layers of algorithmic validation and system verification for space situational awareness. The Kafka-based message bus solves multi-vendor integration challenges common in commercial operations involving satellite operators, launch providers, ground stations, and regulators. Launch trajectory updates automatically propagate to all relevant stakeholders using standardized schemas, while quantum-resistant blockchain ensures decades-long data integrity for insurance and regulatory auditing.

7.2 Automated Resource Allocation and Service Optimization

Commercial constellations require dynamic resource allocation for bandwidth, power, and ground station access based on changing demand. RL-based prediction optimizes these decisions across multiple time horizons while maintaining service guarantees. Communication constellations can predict bandwidth demand patterns and proactively adjust satellite configurations before congestion occurs. The system learns from usage patterns while adapting to new services, with blockchain verification supporting transparent billing and dispute resolution.

7.3 Regulatory Compliance and Economic Benefits

By recording operational decisions and outcomes in immutable ledgers, the traceability framework provides automated compliance documentation. International coordination leverages blockchain-verified messaging with appropriate access controls, reducing administrative overhead while improving regulatory submission accuracy. Demonstrable safety performance through verified operational histories enables sophisticated risk assessment for space insurance, increasing insurability and allowing for greater precision in determining the cost of premiums. Standardized protocols reduce integration costs for new partnerships, while transparency supports new business models including operational data marketplaces. For the commercial SSA industry, the use of uncertainty quantification, anomaly detection, validation through reinforcement learning may provide a path for regulators to validate and oversee the emerging industry to provide confidence to potential purchasers of commercial services. The current US interest in increasing dependence on commercial SSA providers creates a regulatory challenge that can be supported by tools demonstrated in this prototype.

7.4 Implementation and Market Impact

Commercial deployment leverages cloud infrastructure and industry security frameworks, with permissioned blockchain enabling selective data sharing while protecting competitive information. The international nature of such an effort requires consideration of cross-border data governance and regulatory compliance. The SDA TAP Lab collaborative model provides a template for industry consortiums to develop common infrastructure while preserving competitive advantages. The growing commercial space economy offers an ideal testbed for these technologies, with rapid iteration cycles accelerating development for both commercial and defense applications. We present an SDA marketplace based on traceX in [3].

8. KEY CONTRIBUTIONS

TraceX presents an integrated framework combining quantum-resistant blockchain, comprehensive traceability, ML-based quality assurance, reinforcement learning prediction, and adaptive operator interfaces for space battle management systems. The Welder's Arc system architecture demonstrates how these technologies can be integrated within the SDA TAP Lab environment.

Our work provides five primary contributions: (1) practical post-quantum cryptography implementation ensuring long-term blockchain security, (2) end-to-end traceability system enabling complete data and decision visibility, (3) surrogate modeling framework for continuous algorithm quality assurance across multi-vendor environments, (4) RL-based predictive framework for transforming reactive battle management into proactive threat mitigation, and (5) dual operator interfaces combining natural language interaction with structured OODA loop integration.

8.1 Call to action

Priority research directions include extending quantum-safe cryptographic techniques, implementing the complete RL framework for operational validation, scaling to multi-domain operations, implementing federated learning for multi-organization collaboration, and hardening against adversarial attacks. The collaborative SDA TAP Lab model demonstrates effective industry-academia-government partnerships for trustworthy AI in critical defense applications. As space operations become increasingly contested, this framework provides foundational ideas for the next-generation autonomous space battle management capabilities, as well as civil space use, that maintain human oversight while leveraging AI for enhanced decision-making speed and accuracy.

REFERENCES

[1] S. Allen. SDA TAP Lab Using Commercial Technology to Avoid Operational Surprise. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, page 69, Sept. 2024.

[2] A. N. Angelopoulos and S. Bates. Conformal prediction: A unified review of theory and new challenges. *arXiv preprint arXiv:2107.07511*, 2021.

[3] S. Bagchi, L. Pratti, H. Reed, and Y. Latif. Towards an AI and blockchain enabled Space Management System. Space Protocol, 2025. 76th International Astronautical Congress (IAC), Sydney, Australia.

[4] J. Buchmann, E. Dahmen, and A. Hülsing. RFC 8391: XMSS: eXtended Merkle Signature Scheme. Internet Engineering Task Force (IETF), May 2018.

[5] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[6] P. Chi, J. Wei, K. Wu, B. Di, and Y. Wang. A bio-inspired decision-making method of uav swarm for attack-defense confrontation via multi-agent reinforcement learning. *Biomimetics*, 8(2):222, 2023.

[7] D. A. Cooper and Q. Dang. NIST Special Publication 800-208: Recommendation for Stateful Hash-Based Signature Schemes. Technical Report SP 800-208, National Institute of Standards and Technology, Oct. 2020.

[8] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons, 2008.

[9] G. Furlich, A. Crews, J. McGuigan, T. McLauglin, C. Burns, P. Balster, G. Jones, G. Hofer, I. Bartlett, L. Hettiarachchi, et al. Automated, collaborative applications to close kill chain gaps. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, page 40, 2024.

[10] G. Furlich, A. Crews, A. M. Steve Rossland, D. Kim, P. Ryu, C. Tschan, M. Brown, E. Crawford, M. Leonard, R. Neelakantan, S. L. Winder, J. McGuigan, J. Williams, B. Goodwin, J. P. Rao, J. Huang, S. Ruda, J. Al-Kahwati, Y. Latif, S. Bagchi, L. M. Pratti, C. Chin, M. Reed, S. Louque, M. Hills, M. T. Pond, S. Brodeur, D. Kurtenbach, and R. Altobelli. Utilizing civilian launches as live exercises for evaluating a federated protect and defend sda battle management system. In *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, 2025 (accepted).

[11] S. Gopakumar et al. Uncertainty quantification of surrogate models using conformal prediction. *arXiv preprint arXiv:2408.09881*, 2024.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International conference on machine learning*, pages 1861–1870, 2018.

[13] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464, 2001.

[14] J. Kreps, N. Narkhede, and J. Rao. Kafka: a distributed messaging system for log processing. In *Proceedings of the NetDB*, 2011.

[15] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[16] G.-L. Long. Grover algorithm with zero theoretical failure rate. *Physical Review A*, 64(2):022307, 2001.

[17] National Institute of Standards and Technology. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM). Technical Report FIPS 203, U.S. Department of Commerce, June 2024.

[18] National Institute of Standards and Technology. FIPS 204: Module-Lattice-Based Digital Signature Algorithm (ML-DSA). Technical Report FIPS 204, U.S. Department of Commerce, June 2024.

[19] National Institute of Standards and Technology. FIPS 205: Stateless Hash-Based Digital Signature Algorithm (SLH-DSA). Technical Report FIPS 205, U.S. Department of Commerce, June 2024.

[20] A. T. Olutimehin, S. Joseph, A. J. Ajayi, O. C. Metibemu, A. Y. Balogun, and O. O. Olaniyi. Future-proofing data: Assessing the feasibility of post-quantum cryptographic algorithms to mitigate 'harvest now, decrypt later'attacks. *Decrypt Later'Attacks (February 17, 2025)*, 2025.

[21] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38, 2020.

[22] J. Pearl. Causal inference in statistics: An overview. 2009.

[23] M. Pease, F. Wallace, H. Reed, V. Martin, and S. Granata. Supply chain traceability: Manufacturing meta-framework. In *NIST Internal Report*, number NIST IR 8536. National Institute of Standards and Technology (NIST), November 2024. Initial Public Draft.

[24] M. Ribeiro, J. Ellerbroek, and J. Hoekstra. Autonomous air traffic controller: A deep multi-agent reinforcement learning approach. *arXiv preprint arXiv:1905.01303*, 2019.

[25] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

[26] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

[27] H. Y. Wong. Shor's algorithm. In *Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps*, pages 289–298. Springer, 2023.

[28] M. Zhao, G. Wang, Q. Fu, X. Guo, Y. Chen, T. Li, and X. Liu. Mw-maddpg: a meta-learning based decision-making method for collaborative uav swarm. *Frontiers in Neurorobotics*, 17:1243174, 2023.

A. SAC FOR ALGORITHMIC PIPELINES: PROBLEM MAPPING AND OBJECTIVES

Consider a directed acyclic graph (DAG) of algorithms $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. At decision step t, a subset of nodes $\mathscr{V}_t \subseteq \mathscr{V}$ has been executed, producing intermediate outputs $y_{\mathscr{V}_t}$. We define the state as a structured summary of partial execution:

$$s_t \;=\; f_\psi(\mathscr{G}, \mathscr{V}_t, y_{\mathscr{V}_t}) \in \mathbb{R}^{d_s},$$

where $f_\psi(\cdot)$ can be a graph encoder (e.g., a GNN over $\mathscr{G}$ with node features computed from y_v).

An **action** chooses the next executable module $m_t \in \mathscr{A}_{\text{mod}}(s_t)$ (i.e., a node whose prerequisites are satisfied) and its continuous configuration (hyperparameters) $x_t \in \mathbb{R}^{d_x}$. We factor the policy as

$$\pi_\theta(a_t|s_t) \;=\; \pi_{\theta_m}(m_t\,|\,s_t)\;\pi_{\theta_x}(x_t\,|\,m_t, s_t), \qquad a_t = (m_t, x_t),$$

with an **action-mask** enforcing feasibility: $\pi_{\theta_m}(m|s_t) = 0$ for $m \notin \mathscr{A}_{\text{mod}}(s_t)$. After execution, the environment yields the new outputs y_{m_t}, updates $\mathscr{V}_{t+1} = \mathscr{V}_t \cup \{m_t\}$, and transitions to s_{t+1}. Episodes terminate when all sinks in $\mathscr{G}$ required by the task are produced.

We optimize a cost-aware maximum-entropy return tailored to the pipelines:

$$R \;=\; \underbrace{M(y_{\text{sink}})}_{\text{final metric}} \;-\; \lambda_c \sum_{t=0}^{T-1} c(m_t, x_t) \;+\; \lambda_e \sum_{t=0}^{T-1} \Delta M_t,$$

where $M(\cdot)$ is the terminal quality metric (e.g., accuracy, F1, calibration, PSNR), c encodes compute/latency/licensing costs, and ΔM_t is an optional shaping term (incremental improvement in a proxy metric). The **per-step reward** is

$$r_t \;=\; -\lambda_c\, c(m_t, x_t) + \lambda_e\, \Delta M_t, \qquad r_T \;=\; M(y_{\text{sink}})\,.$$

SAC maximizes the entropy-regularized objective with a state-dependent feasible action set:

$$J(\pi) \;=\; \mathbb{E}_{\rho_\pi}\left[\sum_{t=0}^{T} r_t \;+\; \alpha\,\mathscr{H}(\pi(\cdot|s_t))\right], \quad \mathscr{H}(\pi(\cdot|s_t)) = -\mathbb{E}_{a_t\sim\pi}[\log \pi(a_t|s_t)]\,.$$

We maintain two soft Q-networks over masked actions:

$$J_Q(\phi_i) = \mathbb{E}_{(s,a,r,s')\sim\mathscr{D}}\left[\tfrac{1}{2}\left(Q_{\phi_i}(s,a) - \hat{Q}(s,a)\right)^2\right],$$

$$\hat{Q}(s,a) = r + \gamma\mathbb{E}_{a'\sim\pi_\theta(\cdot|s')}\left[\min_{j=1,2} Q_{\bar{\phi}_j}(s', a') - \alpha \log \pi_\theta(a'|s')\right],$$

where sampling a' respects the mask induced by $\mathscr{A}_{\text{mod}}(s')$.

With reparameterization $x_t = g_{\theta_x}(\varepsilon_t; m_t, s_t)$ for the continuous part and (masked) Gumbel–Softmax for m_t, the SAC actor loss is

$$J_\pi(\theta) = \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi_\theta}\left[\alpha \log \pi_\theta(a|s) - \min_{i=1,2} Q_{\phi_i}(s,a)\right].$$

We adapt α to the mixed action structure:

$$J(\alpha) = \mathbb{E}_{a\sim\pi_\theta}\left[-\alpha\left(\log \pi_\theta(a|s) + \mathscr{H}_{\text{target}}(s)\right)\right], \quad \mathscr{H}_{\text{target}}(s) = \beta_m \log|\mathscr{A}_{\text{mod}}(s)| + \beta_x d_x,$$

which scales the desired entropy with the number of feasible modules and continuous dimensionality.

B. TRACEABILITY SCHEMA

Listing 1: Example Header for Calibration Message

```
{
  "messageId": "a47c5d8f-3e2b-4f5a-9825-7ac8f11e2345",
  "messageTime": "2024-12-06T12:34:56Z",
  "messageVersion": "0.0.1",
  "subsystem": "ss0",
  "dataProvider": "spaceprotocol",
  "dataType": "ss0.data.eoobservation.calibrated",
  "dataVersion": "0.1.0",
  "dataPayload": {
    "satnum": "10121",
    "observation": {"x": 0.1234, "y": -0.3456, "t": 1733428563},
    "reference": [
      {"x": 0.1235, "y": -0.3457, "t": 1733427554},
      {"x": 0.1236, "y": -0.3458, "t": 1733427555},
      {"x": 0.1237, "y": -0.3459, "t": 1733427556},
      {"x": 0.1238, "y": -0.3460, "t": 1733427557},
      {"x": 0.1239, "y": -0.3461, "t": 1733427558}
    ],
    "calibrated": {"x": 0.1236, "y": -0.3456x, "t": 1733428563},
  },
  "messageHash" : "35443c3682c492bf82876b95c09a3740",
  "traceability": {
    "internal": [
      {
        "dataType": "ss0.data.eoobservation",
        "messageId": "44444c71-face-41aa-8e66-8829b0781e9a",
        "messageHash": "c079753fbc5b5f8cfa0dc6b13fe7540e",
      }
    ],
    "external": [
      {
        "resourceLink": "https://api.spaceprotocol.org/sensor_calibration",
        "parameters": {
          "queryParameter": {
              "sensorid":"Panopticon_012",
              "timestamp":"2024-12-06T12:34:56Z"
          },
        },
        "resourceHash": "4fd5920800f713a569afb5a354f14292",
      }
    ]
  },
}
```

Listing 1 presents an example of a message header incorporating the traceability section. To facilitate comprehensive tracking of data provenance, parent relationships are categorized as either `internal` or `external`. Internal parent relationships denote data originating within the BMS system via the message bus, while external relationships represent data sources outside WA's boundaries, such as API calls and bulk data requests. For each identified parent, the message header records the data type and the unique message ID. Additionally, a designated `parameters` field is included to accommodate supplementary information for external resources, such as API call parameters or other relevant metadata.