

# SecureL-DLSC: Security- and Privacy-Aware, Location-Sensitive Service Composition with an Auditable Evidence-Admissibility Protocol

Alshaimaa M. Mohammed

Shaimaa\_moustafa@sci.suez.edu.eg

Suez Canal University

---

## Research Article

**Keywords:** cloud service composition, QoS, security risk, privacy exposure, trust, constrained optimization, evidence admissibility, provenance, metadata staleness, WS-DREAM

**Posted Date:** October 22nd, 2026

**DOI:** <https://doi.org/10.21203/rs.3.rs-11151753/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# Abstract

Quality-of-service (QoS) goals including response time, throughput, and cost are frequently the focus of cloud service composition optimization; nevertheless, a QoS-attractive composition may go against required security or privacy regulations. In this paper, we present SecureL-DLSC, an optimizer-agnostic constrained service-composition framework that distinguishes between soft QoS, trust, location, and risk preferences and mandatory security/privacy feasibility. We also present an evidence-admissibility protocol that evaluates whether benchmark service identities can be associated with dated, externally verifiable vulnerability, assurance, and privacy evidence. We demonstrate analytically that the hard constraints are equivalent to per-service admissibility filters under conservative max-aggregation of risk, that soft penalties alone cannot ensure compliance, and that the probability that an unguided search samples a feasible composition is the product of per-class pass fractions. All 5,825 services in WS-DREAM Dataset 1 are subject to the protocol; however, none of them are accepted since the benchmark does not have a named-offering identity or software product/version. Therefore, the audit is a statement about benchmark proof capability rather than provider security posture, and the security, privacy, trust, and cost values in the optimization trials are repeatable synthetic proxy information. Four optimizer implementations are evaluated under three policy profiles (30 matched seeds each) in a controlled single-instance assessment. Sensitivity to policy thresholds, metadata noise, metadata staleness, location and objective weights, and prediction errors are investigated. The reference constrained PSO is feasible under the Balanced and Performance policies, while a constrained genetic algorithm is the only implementation that finds feasible compositions under the Strict policy at the tested budget (66.7% of seeds); these are implementation-level results consistent with the feasibility analysis, not proof of algorithmic superiority. Silent policy infractions can be concealed by recorded feasibility, as demonstrated by a staleness study. An expanded procedure detailed in Supplementary Appendix D is responsible for generalization across candidate pools and metadata realizations.

## 1. Introduction

Applications can put together workflows from separately supplied services thanks to cloud computing. As a result, service composition turns into a discrete multi-objective optimization problem where a workflow must choose one service from each necessary class while meeting application-level QoS constraints. Response time, throughput, cost, and associated QoS metrics are the main focuses of traditional methods. Optimizing these indications by itself, however, does not ensure that the resultant workflow is suitable from a privacy or security standpoint.

When a composition crosses several providers, geographies, trust domains, or data-processing barriers, security and privacy are particularly crucial. A service may present significant vulnerability exposure, weak authentication, insufficient protection controls, excessive retention, deep delegation, or cross-provider data exposure yet offering exceptional response time and throughput. These risks might build up or be dominated by the most exposed service in a workflow. Therefore, explicit risk representation, regulatory limitations, and traceable decision criteria are necessary for a workable composition process.

In order to solve this issue, SecureL-DLSC adds service-level security risk, privacy exposure, trust, and hard feasibility limitations to the location-aware service-composition and predictive QoS foundation of L-DLSC. The system is purposefully optimizer-agnostic: various population-based optimizers can use the same policy and goal interface, whereas limited discrete PSO is one reference implementation. This distinction is crucial because, rather than asserting that PSO is always better than alternative search techniques, the scientific contribution is the evidence-aware limited decision layer.

The fact that security and privacy metadata are not always static presents another difficulty. When vulnerabilities are revealed, controls are altered, certificates expire, providers alter retention rules, or service ownership shifts, evidence may become old. As a result, a composition chosen using outdated metadata might seem plausible even though it goes against the current standard. This study analyzes evidence freshness as a security feature of the decision pipeline and specifically examines that failure mode.

Scope of claims. Security assurance, evidence admissibility, and optimization evidence are all purposefully kept apart in this study. While security, privacy, trust, and cost attributes employed in the optimization studies are synthetic proxy information, WS-DREAM QoS and geography fields are regarded as benchmark observations. The benchmark records' ability to be connected to dated, externally verifiable security and privacy evidence is tested by the evidence-admissibility protocol. A service's evidence-admissibility would only demonstrate that an evidence-based security/privacy decision is supportable under the set provenance rules, not that the service is secure on its own. Although the inherited AE-LSTM predictor is a component of the framework architecture, it is neither retrained or assessed here (Section 2.2). The evaluation also takes place in an observed-QoS context. One issue case is the source of all numerical results (Section 6.1).

## 1.1 Research Questions

- RQ1. Can security and privacy constraints be incorporated into QoS-driven service composition while retaining effective optimization over a large discrete search space?
- RQ2. How do security/privacy policy thresholds change the feasible composition region and the achievable composite objective?
- RQ3. How sensitive is policy compliance to uncertainty and staleness in security/privacy metadata?
- RQ4. Does explicit hard-constraint enforcement provide stronger policy compliance than treating security/privacy only as soft optimization objectives? (Answered analytically in Section 4.9; the empirical four-way comparison is specified in Supplementary Appendix D.)

## 1.2 Contributions

- • An optimizer-agnostic formulation that separates mandatory security/privacy feasibility (hard SR/PE thresholds under conservative max-aggregation) from soft QoS, trust, location, and risk

preferences, with configurable policy profiles.

- • A short formal analysis (Section 4.9) showing that these hard constraints are equivalent to per-service admissibility filters, that unguided search reaches feasibility with probability equal to the product of per-class pass fractions, and that soft penalties alone cannot guarantee compliance. The analysis yields a design rule (filter or repair candidates before search) and explains the Strict-policy behaviour of the tested optimizers.
- • A five-stage evidence-admissibility and provenance protocol that distinguishes identity resolution, source compatibility, temporal validity, and provenance from the security posture itself, and its application to all 5,825 WS-DREAM Dataset 1 services (0 admissible), establishing that the benchmark's identity fields cannot support evidence-backed security claims.
- • A transparent matched-seed comparison of four optimizer implementations (a reference constrained PSO, a constrained GA, MOPSO, and random search) under three policies, reporting feasibility separately from conditional objective quality.
- • Controlled sensitivity studies (policy thresholds, metadata noise and staleness, location and objective weights, forecast error), including an illustrative silent-violation mechanism under stale metadata and a proposed default refresh-cadence policy (Supplementary Appendix C).

## 2. Background and Foundations

### 2.1 QoS-Based Service Composition

Let  $P=\{s_1,s_2,...,s_K\}$  be a composition that chooses one candidate service from each of the  $K$  service classes that make up the workflow. The optimization challenge integrates workflow-level goals with service-level QoS. Throughput and trust are increased while response time and expense are reduced. As more candidates become available in each class, the search space expands multiplicatively.

$$P = \{s_1, s_2, ..., s_K\}$$

### 2.2 Predictive QoS Backbone

The AE-LSTM predictive foundation that was inherited from L-DLSC is still present in the framework [2, 3]. While the LSTM delivers projected QoS values that the downstream optimizer uses, the autoencoder offers a condensed representation of QoS attributes. These elements are not the main innovation in this study; rather, they are supporting infrastructure. The composition experiments in Section 7 use QoS values directly from the WS-DREAM matrices for the chosen user (Section 6.1), i.e., they function in an observed-QoS (oracle-prediction) setting. Section 7.9 simulates predictor error solely by introducing artificial forecast noise. This paper does not retrain or reevaluate the predictor. As a result, the predictor is included in the framework architecture but not in the empirical assertions.

### 2.3 Location-Aware Selection

The Haversine distance [3] is used to describe geographic proximity and is transformed into a normalized location penalty. Physical distance by itself cannot capture routing, congestion, peering, or provider network circumstances, hence this attribute is regarded as contextual information rather than a straight network-latency assessment.

$$LP_i = d_i / (d_{\max} + \epsilon)$$

## 3. Related Work and Research Gap

### 3.1 QoS-Aware Service Composition

Choosing one concrete service for each abstract task while adhering to end-to-end QoS restrictions is a popular formulation of service composition. Integer linear programming was employed in early methods [11, 12]. Later research incorporated genetic algorithms [13], swarm-based selection [4], deep learning-based QoS prediction for cloud composition [2, 3], and the breakdown of global QoS restrictions into local ones for scalability [14]. [15] provides systematic studies of the makeup of cloud services. Although multi-objective, QoS-driven selection is established in these studies, security and privacy constraints are typically not enforced as necessary feasibility conditions.

### 3.2 Location-Aware and Privacy-Aware Selection

Requirement-driven selection of cloud providers against security and privacy requirements has also been proposed [17]. Privacy-aware composition has been studied by confirming compliance between users' privacy requirements and providers' privacy policies and by ranking composite services according to the privacy level they offer [16]. Location-aware methods use geographic data to enhance QoS selection and prediction [3]. Predicted QoS, location, trust, security evidence, and workflow-level hard limitations are generally not combined in a single optimization process in these fields of work.

### 3.3 Trust-Aware Composition

By taking into account past dependability, reputation, and SLA transparency, trust-aware approaches enhance provider selection. For instance, SMICloud ranks cloud services based on several QoS attributes [19] and SelCSP combines competence and trustworthiness to estimate the risk of interaction with a provider [18]. This concept is expanded upon by SecureL-DLSC, which models trust in conjunction with explicit security risk and privacy exposure and enforces risk limitations as constraints instead of combining them into a single trust score.

### 3.4 Security- and Evidence-Oriented Cloud Selection

Evidence such as vulnerability logs, incident reports, security-control attestations, and supplier documentation are needed for security assessments. CVSS scores [7, 8, 29, 30] and records in the NVD and CISA KEV catalog [20, 21] are frequently used to represent vulnerability evidence; CSA STAR, CCM, and CAIQ [9, 10] are used to express control evidence; and NIST SP 800 – 30 [22] codifies risk assessment technique. Such evidence is intended to be consumed by the suggested framework via a

normalized metadata layer. However, instead of comprehensive provider-level evidence mapping, the present trials employ customizable proxy values. The closest previous work on assurance-aware selection is found in certification-based approaches to service assurance and selection, such as the work of Anisetti, Ardagna, Damiani, and colleagues. The protocol suggested here is different in that it is a screening test of whether benchmark identities can support any such linkage at all.

### 3.5 Research Gap

Two related but distinct gaps are addressed in the paper. The first is methodological: rather than focusing solely on soft objective concepts, security and privacy requirements should be treated as hard feasibility conditions in service-composition studies. The second is evidentiary: an auditable mapping between benchmark entities and current external security/privacy evidence is necessary for security-aware claims. The audit conducted in this study reveals that the WS-DREAM Dataset 1's available service identification attributes do not allow the necessary evidence mapping, despite its widespread utility for QoS testing. Therefore, rather than using WS-DREAM as a source of actual security posture, the paper treats it as a true source for QoS/geographic information and as a case study in benchmark evidence limits. The positioning of SecureL-DLSC against representative composition approaches is summarized in Table 1.

Table 1  
Positioning of SecureL-DLSC against representative classes of composition approaches (qualitative summary).

| Method                    | QoS prediction | Location   | Security risk | Privacy | Trust | Hard constraints | Policy-aware |
|---------------------------|----------------|------------|---------------|---------|-------|------------------|--------------|
| GA-based composition      | No             | No         | No            | No      | No    | Limited          | No           |
| DLSC                      | Yes            | No         | No            | No      | No    | QoS              | No           |
| L-DLSC                    | Yes            | Yes        | No            | No      | No    | QoS              | No           |
| Trust-aware composition   | Limited        | No         | No            | No      | Yes   | Limited          | No           |
| Privacy-aware composition | No             | Contextual | No            | Yes     | No    | Privacy          | Limited      |
| SecureL-DLSC              | Yes            | Yes        | Yes           | Yes     | Yes   | Yes              | Yes          |

### 3.6 Inherited Components versus SecureL-DLSC Components

The AE-LSTM QoS prediction backbone and location-aware preprocessing are inherited supporting components from the L-DLSC line of work [2, 3]. They are not claimed as new contributions here. The principal SecureL-DLSC additions are the explicit security-risk and privacy-exposure representations, hard policy constraints, the formal properties of those constraints (Section 4.9), the evidence-

admissibility/provenance layer, the optimizer-independent decision interface, and an evaluation methodology that separates evidence validity from synthetic proxy optimization. The reference PSO is an implementation of this formulation rather than the novelty claim.

## 4. SecureL-DLSC Framework

### 4.1 Architecture

Four logical steps make up SecureL-DLSC: (1) dimensionality reduction and QoS preprocessing; (2) long-horizon QoS prediction or QoS observation ingestion; (3) processing of security/privacy/trust evidence; and (4) an optimizer-independent restricted composition interface. The stage 2 observation-ingestion path is used in the assessment in Section 7. The current study uses a reference discrete PSO implementation, although a constrained genetic algorithm, MOPSO, or another appropriate search technique can solve the same issue. The framework is surrounded by a second methodological layer: an evidence-admissibility and provenance audit establishes if the metadata entering the security/privacy decision may be correctly construed as evidence that has been independently verified.

The overall architecture is shown in Fig. 1, and the processing stages are summarized in Table 2.

Table 2  
SecureL-DLSC processing stages.

| Stage                                                        | Inputs                                    | Outputs                               |
|--------------------------------------------------------------|-------------------------------------------|---------------------------------------|
| QoS preprocessing                                            | RT, throughput, cost                      | normalized/compact QoS representation |
| QoS prediction (optional; observed QoS is used in Section 7) | historical/model-ready QoS representation | predicted RT, throughput, cost        |
| Security/privacy/trust assessment                            | risk, privacy, trust metadata/evidence    | SR, PE, TR                            |
| Constrained PSO                                              | QoS + SR + PE + TR + LP + policy          | feasible composition                  |

### 4.2 Service-Level Security Representation

Security attributes that provide vulnerability exposure, integrity risk, authentication strength, protection capabilities, and past occurrences are used to characterize each applicant service. A common risk scale is used to normalize attributes. The current results do not assert that the numerical risk values are measurable provider-security attributes since they use controlled proxy metadata; instead, the implementation supports evidence-derived values.

### 4.3 Security Risk Score

$$SR_i = (\sum_k \alpha_k X_{ik}) / (\sum_k \alpha_k)$$

Here,  $X_{ik}$  denotes a normalized security-risk attribute and  $a_k$  is its weight. The resulting  $SR_i$  is bounded on the normalized risk scale. A production deployment should retain the source, collection date, evidence type, and confidence associated with each component.

## 4.4 Privacy Exposure

$$PE_i = (\sum_j \beta_j P_{ij}) / (\sum_j \beta_j)$$

Privacy exposure captures data sensitivity, retention, delegation depth, and cross-provider or cross-region exposure. Higher values indicate greater privacy exposure under the configured normalization and weighting scheme.

## 4.5 Trust

$$TR_i = (\sum_k \gamma_k T_{ik}) / (\sum_k \gamma_k)$$

Trust summarizes availability, reliability, security, privacy, and feedback-related evidence. Trust is a positive preference in the objective, whereas SR and PE are risk quantities to minimize and constrain.

## 4.6 Workflow-Level Risk Aggregation

$$SR(P) = \max_{\{i \in P\}} SR_i$$

$$PE(P) = \max_{\{i \in P\}} PE_i$$

A single highly exposed service can ascertain whether the workflow exceeds a required risk threshold thanks to the maximum operator, which offers a cautious workflow-level interpretation. This is appropriate when any workflow component can result in unacceptable exposure and is purposefully tighter than simple average. As a result, both restrictions become separable across service classes, as stated in Section 4.9. Pairwise exposure, like cross-provider or cross-region data flows, is represented here as a per-service attribute; modeling it as an interaction term would link the classes and is left for future work.

## 4.7 Location Penalty

$$LP(P) = (1/K) \sum_i LP_i$$

## 4.8 Unified Risk-Aware Objective

$$F(P) = Q(P) - \lambda_c C(P) - \lambda_{rt} RT(P) + \lambda_{th} TH(P) + \lambda_{tr} TR(P) - \lambda_{sr} SR(P) - \lambda_{pe} PE(P) - \lambda_{loc} LP(P)$$

Soft preferences are combined in the goal. Prior to aggregation,  $Q(P)$ ,  $C(P)$ ,  $RT(P)$ ,  $TH(P)$ ,  $TR(P)$ ,  $SR(P)$ ,  $PE(P)$ , and  $LP(P)$  are normalized to the common  $[0,1]$  decision scale; maximizing terms are represented as positive utilities and minimization terms as penalties.  $F$  is not bounded above by 1 (values above 1 appear in Section 7) due to the addition of positive utility factors. Table B1 (Supplementary Appendix B) contains the definition of  $Q(P)$  as the aggregate QoS utility, the aggregation methods for  $RT$ ,  $TH$ , and cost, the normalization ranges, and all weights.  $F$  is dependent on these weights; it can be compared between

optimizers within a single policy and weight setting, but not between other policies or weight settings (Sections 7.2, 7.7, and 7.8). Hard restrictions are also used to ensure security and privacy, thus a high QoS score cannot make up for a required policy breach.

## 4.9 Hard Security and Privacy Constraints

$$SR(P) \leq SR_{\max}$$

$$PE(P) \leq PE_{\max}$$

For the hard-constraint configuration, a candidate is deemed infeasible if they breach either constraint. This distinction is crucial: the optimizer cannot exchange required security or privacy compliance for better QoS, even though it may favor one workable composition over another based on soft objectives. A direct comparison between preference-based and mandatory policy enforcement is made possible by the soft-only experimental setup, which keeps the identical SR and PE quantities in the objective but does not employ them as hard feasibility criteria.

Formal properties. Let  $C_k$  denote the candidate set of service class  $k$ , let  $A_k = \{s \in C_k : SR_s \leq SR_{\max} \text{ and } PE_s \leq PE_{\max}\}$  be its admissible subset, and let  $p_k = |A_k| / |C_k|$  be its pass fraction.

### Proposition 1

(separability). Under the aggregation of Section 4.6 and the constraints of Section 4.9, a composition  $P = (s_1, \dots, s_K)$  is feasible if and only if  $s_k \in A_k$  for every class  $k$ . The feasible set is therefore the Cartesian product  $A_1 \times \dots \times A_K$ , and a feasible composition exists if and only if no  $A_k$  is empty. Proof. The maximum of a finite set is at most a threshold if and only if every element is; apply this to SR and to PE.

### Proposition 2

(unguided feasibility probability). If a composition is drawn uniformly from the product of the  $C_k$ , it is feasible with probability  $\prod_k p_k$ . This is also the probability that a random initial particle or individual is feasible. For  $K = 10$ , even moderately selective thresholds make this product small (for example,  $p_k = 0.2$  in every class gives  $0.2^{10} \approx 1 \times 10^{-7}$ , an illustration rather than a measured value); rejection-based search then receives almost no feasible samples from which to learn.

### Proposition 3

(soft penalties do not guarantee compliance). For any finite weights  $\lambda_{sr}$  and  $\lambda_{pe}$  there exist instances in which the maximizer of  $F$  over the unconstrained space violates  $SR_{\max}$  or  $PE_{\max}$ . Proof sketch. SR and PE lie in  $[0,1]$ , so the total risk penalty in  $F$  is bounded by  $\lambda_{sr} + \lambda_{pe}$ . Any inadmissible service whose remaining utility exceeds that of the best admissible alternative in its class by more than this bound is selected by the maximizer of  $F$ . Mandatory compliance therefore requires hard enforcement; the soft terms only rank compositions within the feasible region.

Design implication. Because feasibility is separable, hard SR/PE constraints should be enforced by filtering (or repairing) candidate sets before search. Filtering costs  $O(N)$ , guarantees that every sampled composition is feasible whenever a feasible composition exists, and removes infeasibility handling from the optimizer. The optimizers evaluated in Section 7 deliberately treat feasibility as a black-box condition (rejection or feasibility-first comparison), which is a generic constraint-handling test; their Strict-policy behaviour should be read in that light (Section 7.1). Separability also means that the constrained problem is easier than the size of the search space suggests: when the QoS aggregation is additive over classes, per-class selection after filtering provides an exact or near-exact reference solution, and such a reference is recommended for future optimizer comparisons (Supplementary Appendix D).

## 4.10 Adaptive Policy Profiles

Table 3  
Configurable security/privacy policy profiles used in the evaluation.

| Policy      | SRmax | PEmax | Security weight | Latency weight | Interpretation                        |
|-------------|-------|-------|-----------------|----------------|---------------------------------------|
| Strict      | 0.25  | 0.25  | 0.25            | 0.10           | High protection / sensitive workloads |
| Balanced    | 0.40  | 0.40  | 0.15            | 0.15           | General enterprise workloads          |
| Performance | 0.60  | 0.60  | 0.08            | 0.25           | Performance-oriented workloads        |

## 4.11 Evidence and Provenance Layer

The metadata layer links each service/provider to auditable sources for deployment that is externally validated. The five stages of the protocol are as follows: (1) identity resolution, which requires a sufficiently specific provider/service identity and, if applicable, product/version information; (2) evidence-source compatibility, which matches the identity to an appropriate vulnerability, assurance, security-control, or privacy source; (3) temporal validation, which guaranties that the evidence has a usable publication, observation, assessment, or retrieval date; (4) provenance classification, which distinguishes between direct, provider-attested, independently verified, indirect, and unresolved evidence; and (5) admission. Only when the necessary identity, source, temporal, and provenance requirements are concurrently met is a service acceptable as evidence.

The protocol adheres to a conservative rule: evidence that is unknown is not considered low-risk. Therefore, in a hard-constraint evaluation, the corresponding record cannot be regarded as verified evidence if an admissibility criterion is not met. Evidence admissibility merely proves that the evidence is sufficiently traceable and current for the configured decision rule; it is not the same as security assurance.

## 4.12 Risk-Aware Discrete PSO

In accordance with the discrete-search logic of particle swarm optimization, particles encode a single service index for each service type [4]. Instead than being the main innovation of SecureL-DLSC, PSO is employed here as a reference search implementation. Candidate compositions are assessed using QoS, trust, security risk, privacy exposure, and location at each iteration. In the hard-constraint arrangement, particles that violate SRmax or PEmax are rejected. In order to update particle velocities, locations, personal bests, and the global best, feasible particles are scored using the unified objective. Without altering the formulation of the framework, alternative optimizers can use the same aim, evidence, and feasibility interface.

### Algorithm 1

SecureL-DLSC composition procedure (reference PSO): (1) normalize QoS and metadata; (2) encode service candidates; (3) obtain QoS values (predicted by the AE-LSTM backbone, or observed as in the present evaluation); (4) initialize discrete PSO particles; (5) decode each particle into a composition; (6) compute SR, PE, TR, and LP; (7) reject policy-violating particles (alternatively, and preferably, restrict each class to its admissible subset before initialization, Section 4.9); (8) evaluate the unified objective for feasible particles; (9) update personal/global bests; (10) update velocities and positions; (11) iterate until the stopping condition; (12) return the best feasible composition.

## 5. Computational Complexity

While the predominant discrete PSO search maintains its typical population-by-iteration-by-workflow-dimensionality structure, the security assessment adds linear candidate-level metadata processing. Excluding model-inference constants, metadata preparation is  $O(N)$  and PSO evaluation is around  $O(MIK)$  if  $N$  is the number of candidate services,  $M$  is the number of particles,  $I$  is the number of iterations, and  $K$  is the size of the workflow. Constraint management itself costs  $O(N)$  and adds no term to the search if candidate filtering (Section 4.9) enforces hard limitations. As a result, the evaluation payload is altered by the security layer, but the asymptotic PSO search order remains unchanged.

Metadata preparation:  $O(N)$

PSO search:  $O(M I K)$

For the reported configuration,  $M = 30$  particles and  $I = 50$  iterations are used. Empirical scaling with  $N$ ,  $M$ ,  $I$ , and  $K$  is not evaluated in this paper and remains future work.

## 6. Experimental Methodology

### 6.1 Dataset

Geographic and QoS data are evaluated using WS-DREAM Dataset 1 [1]. The collection contains 339 users and 5,825 services. The response-time and throughput matrices are 95% and 93% dense,

respectively, when missing elements are coded – 1. The assessment uses one target user and one 1,000-service candidate pool divided into ten service classes of 100 candidates each. Using master seed 20260916 without replacement, the target user's 5,492 accessible services are chosen. The pool is shared by all algorithms and matching seeds. All of the numerical results in Section 7 are derived from this single problem instance and one synthetic metadata realization; the multi-instance replication strategy is described in Supplementary Appendix D.

## **6.2 Security, Privacy, and Trust Metadata**

### **6.2.1 Evidence sources and admissibility rules**

Vulnerability exposure is provided by NVD CVE/CVSS records and the CISA KEV catalog [7, 8, 20, 21], control/assurance proof is provided by CSA STAR/CAIQ/CCM [9, 10] and provider documentation, and privacy exposure is provided by out-of-date privacy policies, DPAs, and retention/subprocessor disclosures. Admission requires a provenance assessment, a dated source, and an explicit or independently validated service/provider mapping (see Section 6.2.2 below).

### **6.2.2 Validation and exclusion protocol**

Five tests are performed on each candidate service: (1) identity resolution from the WS-DREAM provider/WSDL fields; (2) match to an admissible external evidence source; (3) temporal validity with respect to the benchmark observation/crawl context; (4) provenance and evidence-type validation; and (5) final admission. A service that fails any required check is removed from the evidence-based experiment rather than receiving an inferred value. When an external source requests product/version identity, the protocol does not consider an unresolved identity to be a secure service or assume that a provider name is sufficient for vulnerability matching. Instead of determining whether fuzzy provider-name matching against registries may be successful, the current audit assesses Stage 2 structurally by asking whether any field in the dataset may be utilized as a lookup key for the external sources (see Section 9.1). Furthermore, Stage 3 demands proof that can be connected to the service as it was during the 2009–2010 crawl; the test is intentionally difficult because claims about dead endpoints are usually not supported by data on current products.

### **6.2.3 Evidence-Admissibility Audit of WS-DREAM Dataset**

#### **1**

The mapping and exclusion protocol was applied to the complete 5,825-service list of WS-DREAM Dataset 1. Out of 5,825 services, 0.0% are accepted. A named assurance-scoped cloud offering or a CPE-style vendor:product:version identity are exactly what CVE/CVSS/CISA-KEV/CSA-STAR admissibility requires, but none of the 5,647 services (96.9%) that resolve to a named provider/organization string (Stage 1) provide a software product name or version anywhere in the source dataset. Consequently, the audit functions as an identity-capability check. This is a structural aspect of the dataset—a 2009–2010 QoS-only crawl without a software-identity field—rather than a lack of search effort. The whole stage-by-stage exclusion log is included in Supplementary Appendix A.

Consequence: Since 0 services are permitted, no evidence-based security/privacy numerical experiment can be carried out without producing identity or version data that is missing from the source dataset. For each SR, PE, TR, and cost value used in Section 7, a synthetic Beta(2,5)/Beta(5,2)-based model implemented in the given code (Supplementary Appendix B) produces reproducibly seeded proxy information, enabling the reproduction of every number stated here from the supplied seeds. According to the audit, WS-DREAM Dataset 1 cannot be utilized to produce values that are supported by evidence and does not currently offer actual security metrics. A more comprehensive audit for the 5,647 listed providers would try provider-level matching (for example, against CSA STAR register entries and CPE vendor information); this is mentioned as future work (Section 9.1).

The core experimental configuration used throughout the evaluation is summarized in Table 4.

### 6.3 Experimental Configuration

Table 4  
Core experimental configuration.

| Parameter                    | Value                         |
|------------------------------|-------------------------------|
| Service classes              | 10                            |
| Candidates per class         | 100                           |
| Particles / population       | 30                            |
| Iterations / generations     | 50                            |
| Runs per (policy, algorithm) | 30 matched seeds              |
| Policies                     | Strict, Balanced, Performance |
| Primary QoS metrics          | RT, throughput                |
| Risk metrics                 | SR, PE, TR                    |
| Location metric              | Haversine-derived LP          |

### 6.4 Evaluation Metrics

For each optimizer, the feasibility rate (with Clopper-Pearson 95% intervals) and mean composite goal F among feasible runs are reported in Section 7.1. Workflow response time, throughput, SR, PE, TR, constraint-violation magnitude V(P), and runtime are all recorded and released with the code (Supplementary Appendix B), though they are not tabulated here. Where the sensitivity studies use them, SR, PE, location penalty, and realized response time are reported. Feasibility and violation behavior are evaluated separately from conditional objective quality; unfeasible runs are not assigned an artificial F value.

### 6.5 Baselines and Statistical Protocol

Four optimizers were compared: a multi-objective PSO (MOPSO), a constrained discrete PSO (the reference implementation, labeled "SecureL-DLSC" in the figure legends and "PSO" in the tables), an NSGA-II-lineage constrained genetic algorithm (labeled "NSGA-II" in the figure legends and "GA" in the tables), and random search. The selection, crossover, mutation, and constrained-domination tournament criteria of Deb et al. [5] are followed by the genetic algorithm, which is implemented natively. Because it is applied directly to the scalar composite goal  $F$ , its restricted-domination rule reduces to a feasibility-first comparison of  $F$ . It should be seen as an NSGA-II-lineage constrained GA rather than a full Pareto-based NSGA-II. One potential disadvantage of MOPSO is that its Pareto mechanism is not used because all optimizers are graded on the scalar  $F$ . MOPSO is a custom external-archive implementation that adheres to Coello and Lechuga's traditional formulation [6]. Each optimizer was run for 30 matched seeds (seed indices 1–30, same across algorithms) under each of the three strategies for a total of 360 runs. "Matched" refers to the fact that each algorithm's seed index initializes the same RNG stream index; this aligns runs by index but, because the algorithms use random numbers differently, does not, strictly speaking, result in a statistical pairing of outcomes.

All four optimizers were implemented using the formulations in Section 4 and standard published descriptions of NSGA-II and MOPSO [5, 6]; the full source code is given (Supplementary Appendix B). None of the optimizers are adjusted, and the results in Section 7.1 are for these implementations.

For every policy, there is a distinction between objective quality and feasibility. The percentage of runs that provide a composition that fulfills  $SR \leq SR_{max}$  and  $PE \leq PE_{max}$  is known as the feasibility rate, and it is represented with 95% Clopper-Pearson intervals. Impractical runs are characterized by constraint-violation magnitude  $V(P) = \max(0, SR(P) - SR_{max}) + \max(0, PE(P) - PE_{max})$ , released each run, and not provided an artificial  $F$ . Conditional objective quality analysis, which produces the mean, standard deviation, and a t-based 95% CI, only uses feasible runs. Since fixed-penalty imputation confuses low objective quality with failing to achieve a requirement, it is not used in the primary analysis. Friedman [23] and Wilcoxon signed-rank [24] tests with Holm adjustment [25] are offered as within-instance descriptive comparisons of different implementations in accordance with Demšar's non-parametric recommendations [26]. Because the 30 seeds sample optimizer stochasticity on a single problem instance, these experiments do not permit inference across issue instances, which would need instance-level replication (Supplementary Appendix D).

## 7. Results

### 7.1 Baseline Comparison

Table 5

Matched 30-seed comparison across policies (mean F, SD, and 95% t-interval over feasible runs; feasibility rate with 95% Clopper–Pearson interval in %). PSO = reference constrained PSO; GA = NSGA-II-lineage constrained GA.

| Policy      | Algorithm | Mean F (feasible runs) | SD     | Feasibility rate [95% CI] | 95% CI (mean F)  |
|-------------|-----------|------------------------|--------|---------------------------|------------------|
| Strict      | PSO       | —                      | —      | 0.0% [0.0, 11.6]          | —                |
| Strict      | GA        | 0.9186                 | 0.0180 | 66.7% [47.2, 82.7]        | [0.9102, 0.9270] |
| Strict      | MOPSO     | —                      | —      | 0.0% [0.0, 11.6]          | —                |
| Strict      | Random    | —                      | —      | 0.0% [0.0, 11.6]          | —                |
| Balanced    | PSO       | 0.9502                 | 0.0169 | 96.7% [82.8, 99.9]        | [0.9438, 0.9566] |
| Balanced    | GA        | 0.9966                 | 0.0121 | 100.0% [88.4, 100.0]      | [0.9921, 1.0011] |
| Balanced    | MOPSO     | 0.9335                 | 0.0106 | 100.0% [88.4, 100.0]      | [0.9295, 0.9375] |
| Balanced    | Random    | 0.9254                 | 0.0100 | 100.0% [88.4, 100.0]      | [0.9217, 0.9291] |
| Performance | PSO       | 1.0056                 | 0.0143 | 100.0% [88.4, 100.0]      | [1.0003, 1.0109] |
| Performance | GA        | 1.0504                 | 0.0101 | 100.0% [88.4, 100.0]      | [1.0466, 1.0542] |
| Performance | MOPSO     | 0.9888                 | 0.0085 | 100.0% [88.4, 100.0]      | [0.9856, 0.9920] |
| Performance | Random    | 0.9852                 | 0.0076 | 100.0% [88.4, 100.0]      | [0.9824, 0.9880] |

While the GA achieves 66.7% (95% CI 47.2–82.7%) under Strict, three of the four implementations (PSO, MOPSO, random search) never find a suitable composition across 30 seeds (0.0%; 95% CI 0.0–11.6%). The possible joint minimum of  $\max(\text{SR}, \text{PE})$  across the ten service classes in this candidate pool is roughly 0.155, below the Strict barrier of 0.25, according to a greedy per-class lower bound, indicating the existence of workable compositions. They are difficult to locate because, according to Proposition 2, the chance that an unguided sample is feasible is the product of ten per-class pass fractions, which is minimal under Strict. This is because the feasible set is a product of per-class admissible subsets. While feasibility-first comparison rules rank infeasible solutions by constraint violation [27, 28] and can move toward the feasible region, rejection-based handling (PSO, MOPSO, random search) receives nearly no feasible samples and, for the PSO, no information about the distance to the feasible region. Although it was not isolated experimentally—specifically, no candidate-filtering variant was run—this account is compatible with the findings. The Strict-policy failures of the PSO, MOPSO, and random search should be

interpreted as failures of black-box constraint handling rather than as proof of the formulation as, according to Proposition 1, such a version would guaranty feasibility whenever a viable composition exists. The PSO achieves 96.7% and 100% feasibility under Balanced and Performance (95% CI 82.8–99.9% and 88.4–100%; all other implementations reach 100%). The GA achieves the highest mean F in both feasible policies, surpassing the PSO, while MOPSO and random search get the lowest. In this case, the ranking  $GA > PSO > \{MOPSO, \text{random search}\}$  holds in both policies; however, MOPSO and random search differ under Balanced (Holm-adjusted  $p = 6.64e-03$ , Table 7) but not under Performance ( $p = 9.61e-02$ ). This is not a generic rating of the underlying algorithms; rather, it describes these untuned implementations on this particular instance.

Reproducibility note. Tables 5, 7, and 8 and Figs. 2–7 are the direct, unedited output of the released code and seeds (Supplementary Appendix B). Table 6 was produced with fixed-penalty imputation for infeasible runs and is retained only as a sensitivity analysis.

Table 6  
Friedman test (4 algorithms × 30  
matched seeds) with fixed-penalty  
imputation of infeasible runs;  
sensitivity analysis only, not used for  
inference.

| Policy      | $\chi^2$ | p        |
|-------------|----------|----------|
| Strict      | 60.000   | 5.88e-13 |
| Balanced    | 67.720   | 1.31e-14 |
| Performance | 66.040   | 3.01e-14 |

Table 7

Holm-adjusted pairwise Wilcoxon signed-rank tests on feasible runs (Balanced and Performance only; under Strict too few feasible runs exist). Within-instance descriptive comparison of the PSO, GA, MOPSO, and random search implementations.

| Policy      | Pair            | Holm-adjusted p | Rank-biserial r | p < 0.05 |
|-------------|-----------------|-----------------|-----------------|----------|
| Balanced    | PSO vs GA       | 1.118e-08       | -1.000          | Yes      |
| Balanced    | PSO vs MOPSO    | 9.202e-04       | 0.699           | Yes      |
| Balanced    | PSO vs Random   | 7.100e-05       | 0.815           | Yes      |
| Balanced    | GA vs MOPSO     | 1.118e-08       | 1.000           | Yes      |
| Balanced    | GA vs Random    | 1.118e-08       | 1.000           | Yes      |
| Balanced    | MOPSO vs Random | 6.640e-03       | 0.557           | Yes      |
| Performance | PSO vs GA       | 1.118e-08       | -1.000          | Yes      |
| Performance | PSO vs MOPSO    | 1.599e-05       | 0.849           | Yes      |
| Performance | PSO vs Random   | 3.073e-07       | 0.953           | Yes      |
| Performance | GA vs MOPSO     | 1.118e-08       | 1.000           | Yes      |
| Performance | GA vs Random    | 1.118e-08       | 1.000           | Yes      |
| Performance | MOPSO vs Random | 9.610e-02       | 0.351           | No       |

Table 6 is not used as evidence of objective-quality superiority. The Strict-policy conclusion is stated through feasibility rates: the PSO, MOPSO, and random search found no feasible composition in 30 seeds, while the GA found feasible compositions in 66.7% of seeds. Figure legends denote the PSO as 'SecureL-DLSC' and the GA as 'NSGA-II'.

## 7.2 Policy Sensitivity

The feasibility rate of the same nominal configuration (PSO under the Balanced policy) ranges between 95% and 100% across studies due to the sensitivity experiments in Sections 7.2–7.9 using study-specific seed offsets (Supplementary Appendix B). This is consistent with sampling variability at 20–30 seeds. The pipeline of Section 7.1 and the PSO alone, with 30 seeds per policy, are used in the policy-sensitivity study. Its mean composite objective across possible runs increases from 0.9502 to 1.0056, and its feasibility rate increases from 0.0% (Strict) to 96.7% (Balanced) to 100.0% (Performance) (Table 5). F is not comparable across policies since Table 3 gives each policy a different security and latency weight. This increase partially reflects the altered objective (for example, the security weight drops from 0.25 to 0.08) rather than just a larger feasible region. The policy-comparable outcome is the feasibility trend: lowering SR/PE thresholds expands and simplifies the feasible region.

## 7.3 Security and Privacy Constraint Ablation

Twenty matching seeds are used in the ablation. The PSO's Balanced-policy solutions meet the Balanced standards in 95.0% of runs (19/20; 95% CI 75.1–99.9%) when strong SR/PE constraints are applied during search. Compliance drops to 0.0% (0/20; 95% CI 0.0–16.8%) when SR/PE are eliminated from both the hard constraint and the target and the resulting composition is compared post hoc against the same thresholds. This comparison does not distinguish between hard and soft enforcement because both the hard limits and the soft penalties are eliminated simultaneously. The empirical four-way comparison is detailed in Supplementary Appendix D, and Proposition 3 (Section 4.9) analyzes this distinction. The experiment measures the magnitude of the effect on this instance; given Proposition 1, the direction of the result is predictable since a search that disregards SR and PE has no incentive to satisfy the thresholds.

Figure 3. Effect of removing hard SR/PE constraints on compliance with the Balanced policy thresholds (20 matched seeds).

## 7.4 SRmax × PEmax Threshold Sensitivity

The grid covers  $SR_{max}$  and  $PE_{max} \in \{0.20, 0.30, 0.40, 0.50, 0.60\}$  (5×5 cells), with 30 matched seeds per cell (750 runs) of the PSO. Feasibility is asymmetric in  $SR_{max}$  and  $PE_{max}$ : for this candidate pool,  $PE_{max} = 0.20$  keeps feasibility at or below 7% regardless of  $SR_{max}$ , while  $SR_{max} = 0.20$  similarly caps feasibility at or below 7% regardless of  $PE_{max}$ . Because the greedy lower bound of Section 7.1 (approximately 0.155) lies below 0.20, feasible compositions exist at these thresholds; the low feasibility therefore reflects search difficulty rather than the absence of feasible compositions. Proposition 2 explains the mechanism: at tight thresholds the per-class pass fractions, and hence the fraction of feasible samples, are very small. Feasibility reaches 100% once both thresholds are at or above 0.40 ( $SR_{max}$ ) and 0.30–0.40 ( $PE_{max}$ ) jointly, with the transition band (13–93% feasibility) concentrated around  $SR_{max} = 0.30$ –0.40,  $PE_{max} = 0.30$ –0.40. Mean F among feasible runs is comparatively flat (0.91–0.96) across the feasible region, indicating that once a policy admits enough candidates to search over, the achievable objective value is only weakly sensitive to exactly how permissive the thresholds are.

## 7.5 Metadata Staleness and Silent Violations

This study uses 30 matched seeds per staleness level  $\Delta$  under the Balanced policy (Table 8, Fig. 5). Staleness is simulated as a seeded random-walk drift applied to the true (post-refresh) SR/PE values, while the decision continues to use the last-known (stale) values; a silent violation is recorded when a decision recorded as feasible under stale metadata would in fact violate  $SR_{max}$  or  $PE_{max}$  once the true drifted values are used. The silent-violation rate rises from 0.0% at  $\Delta = 0$  to 10.7% at  $\Delta = 1$ , 43.3% at  $\Delta = 2$ , 56.7% at  $\Delta = 3$ , 73.3% at  $\Delta = 4$ , and 80.0% at  $\Delta = 5$  (Table 8). The direction of this result is expected: selected compositions sit below, but not far from, the thresholds (Section 7.8), so drift can push a component across them. The magnitudes, however, depend entirely on the assumed drift step, and  $\Delta$  is not tied to a calendar interval; the study therefore illustrates a mechanism and does not estimate rates for real providers. Its point is that recorded feasibility is not evidence of current compliance unless

evidence freshness is enforced (Section 8.3, Supplementary Appendix C). Calibrating the drift against historical changes in NVD/KEV records is left to future work.

Table 8

Metadata staleness and silent policy-violation rate (Balanced policy, 30 matched seeds per  $\Delta$ ).

| $\Delta$ (refresh intervals missed) | Recorded-feasible decisions (n/30) | Silent violation rate |
|-------------------------------------|------------------------------------|-----------------------|
| 0                                   | 30                                 | 0.0%                  |
| 1                                   | 28                                 | 10.7%                 |
| 2                                   | 30                                 | 43.3%                 |
| 3                                   | 30                                 | 56.7%                 |
| 4                                   | 30                                 | 73.3%                 |
| 5                                   | 30                                 | 80.0%                 |

## 7.6 Metadata Noise and Uncertainty

SR and PE (20 seeds per level, Balanced policy) are supplemented with Gaussian noise with increasing standard deviation  $\sigma$ . The Clopper-Pearson 95% intervals are 83.2–100% for 20/20, 75.1–99.9% for 19/20, and 68.3–98.8% for 18/20; feasibility was 100% at  $\sigma = 0$  and  $\sigma = 0.02$ , 95% at  $\sigma = 0.05$ , 100% at  $\sigma = 0.10$ , and 90% at  $\sigma = 0.15$  (Fig. 6a). The non-monotonic pattern falls under sampling variability because, with 20 seeds per level, a variation of one or two runs equates to 5–10 percentage points. The results only indicate that a few feasibility points can be lost to noise alone before any staleness effect (Section 7.5) is applied; no systematic degradation is seen at these levels, although the study has insufficient power to detect tiny effects.

## 7.7 Location Sensitivity

The location weight  $\lambda_{\text{loc}}$  is swept over  $\{0.00, 0.05, 0.10, 0.20, 0.30\}$  (20 seeds per level, Balanced policy; Fig. 6b). As  $\lambda_{\text{loc}}$  increases from 0 to 0.30, the mean realized location penalty of the selected composition falls monotonically from 0.279 to 0.174, while the mean composite objective  $F$  falls from 0.970 to 0.897. Because  $F$  contains the term  $-\lambda_{\text{loc}} \cdot \text{LP}$ , values of  $F$  at different  $\lambda_{\text{loc}}$  are not directly comparable and the decline in  $F$  partly reflects the changed objective; the realized location penalty is the comparable outcome. This is the expected soft-preference trade-off: a larger location weight buys lower average geographic distance at the cost of the other QoS/trust/cost terms in the objective.

## 7.8 Objective-Weight Sensitivity

The security weight  $\lambda_{\text{sr}} = \lambda_{\text{pe}}$  is swept over  $\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$  with  $\text{SR}_{\text{max}} = \text{PE}_{\text{max}} = 0.40$  fixed (20 seeds per level; Fig. 6c). Mean  $F$  among feasible runs falls monotonically from 1.012 at  $\lambda_{\text{sr}} = 0.05$  to 0.848 at  $\lambda_{\text{sr}} = 0.30$ , which is largely mechanical because  $F$  contains  $-\lambda_{\text{sr}} \cdot \text{SR} - \lambda_{\text{pe}} \cdot \text{PE}$ ; the comparable quantity is the mean realized SR of the selected composition, which stays roughly flat

(0.327–0.343) across the sweep. This indicates that the objective-weight term mainly re-prices already-similar feasible solutions rather than steering the search toward markedly lower-risk compositions once the hard SR\_max/PE\_max constraints are active; the risk reduction in this framework is done by the hard constraints (Sections 7.3, 7.4), with the soft weight acting as a secondary preference.

## 7.9 Forecast-Error Sensitivity

Zero-mean Gaussian forecast error of standard deviation  $\sigma$  is added to the normalized RT signal used for selection only (20 seeds per level, Balanced policy); realized performance is then measured against the true, error-free RT of the same selected composition (Fig. 7). The mean composite objective of the selected composition (evaluated on the noisy data used for selection) stays essentially flat across  $\sigma \in \{0, 0.05, 0.10, 0.15, 0.20, 0.30\}$  (0.937–0.946), which suggests that RT has a small effective weight in this configuration and limits the sensitivity of this experiment to predictor error; a configuration with a larger RT weight would be needed to stress the predictor interface. The realized (true, post hoc) response time of the selected composition varies between 0.074 and 0.103 across the sweep with no clean monotonic trend, and with 20 seeds per level this variation cannot be distinguished from sampling noise. Between 19 and 20 of 20 seeds remained feasible at every noise level, so any effect concerns selection quality within the feasible set rather than feasibility itself. This complements the staleness illustration (Section 7.5): staleness affects the hard SR/PE constraint, whereas forecast error affects the realized value of a soft QoS term.

## 8. Discussion

### 8.1 Security–QoS Trade-off

Instead of asserting universal algorithmic supremacy, the key framework conclusion is methodological: soft aims provide preferences inside an explicit compliance limit (Proposition 3), while necessary SR/PE restrictions establish an explicit compliance border (Proposition 1). This behavior is shown in controlled synthetic circumstances in the policy experiments. The experiments do not determine the security posture of real providers because SR, PE, and TR are artificial proxies. Furthermore, no Pareto front is reported; scenarios with positive, zero, and negative dependence are required before trade-off conclusions can be made. The strength and sign of any security–QoS trade-off in these experiments is determined by the assumed dependence between the synthetic metadata and the QoS values, which was not varied here (Supplementary Appendix D).

### 8.2 Policy Adaptability

Strict, Balanced, and Performance policies are controlled synthetic situations that show how objective trade-offs and feasibility are impacted by policy thresholds. They are not offered as industrial policies that have been seen objectively. They are valuable because the presumptions are clear and repeatable.

### 8.3 Metadata Freshness as a Security Property

A hard restriction is only as current as the evidence used to assess it, according to the staleness experiment, which highlights an evidence-validity criterion. Stale metadata in the synthetic experiment can cause realized risk to exceed the current threshold while recorded feasibility remains high. This encourages any future evidence-based deployment to include specific provenance, timestamps, confidence, and update cadence.

## 8.4 Evidence Quality

The methodological outcome of a specific but limited scope is the audit of all 5,825 WS-DREAM services. It determines that 0 services meet the specified evidence-admissibility criterion since the dataset does not contain the product/version or named-offering identification required for the external sources taken into consideration, using an executed protocol and a per-stage exclusion log. The outcome is expected given the schema of the dataset and should not be overread; its value is found in the explicit statement of what evidence a QoS benchmark can and cannot substantiate, as well as in the reusable screening protocol that can be applied to datasets that carry product/version or offering identity. The proper conclusion is that provider-level security-evidence assertions cannot be supported by a QoS benchmark without an acceptable service identity, not that WS-DREAM is inappropriate for QoS research.

## 8.5 Practical Implications

Rather than being a claim of superiority, the matched baseline experiment should be interpreted as an implementation-level assessment. The NSGA-II-lineage constrained GA is the only tested optimizer that finds feasible compositions under Strict, and in the disclosed implementation, it achieves higher mean composite goal values than the reference PSO under the Balanced and Performance policies. A general ranking of optimization methods is not implied by these observations. In conjunction with the analysis presented in Section 4.9, they demonstrate that alternative search engines can solve the SecureL-DLSC formulation and that hard SR/PE constraints of this type are best enforced by candidate filtering or repair prior to search, leaving the optimizer to handle only the soft objective.

## 9. Threats to Validity and Limitations

### 9.1 Security/Privacy Evidence

All 5,825 WS-DREAM Dataset 1 services were subjected to the evidence-mapping technique (Section 6.2.2) (Supplementary Appendix A). Since there is no software product or version identifier in the dataset, zero services are admissible under the auditable, dated, externally verifiable evidence criterion. As a result, all numerical SR/PE/TR/cost numbers continue to be proxy metadata produced by seeded synthetic distributions. Fuzzy matching of the 5,647 listed providers against registries (such as CPE vendor names or CSA STAR listings), which could produce provider-level evidence for a subset of services, was not attempted in the audit, which checks identity capabilities. Furthermore, several

endpoints are no longer operational, and evidence from after the 2009–2010 crawl could not accurately reflect the services as they were at that time.

## 9.2 Dataset Temporal Structure

WS-DREAM Dataset 1 lacks an explicit chronological time axis, so no chronological forecasting validation of the QoS predictor is possible on this dataset; predictor error is examined only through noise injection (Section 7.9).

## 9.3 Problem-Instance Replication

All experiments use one target user, one candidate pool of 1,000 services in 10 classes, and one draw of synthetic metadata (Section 6.1). The 30 matched seeds therefore capture the stochastic variability of each optimizer on this instance, not variability across candidate pools, users, workflow sizes, or metadata realizations; the rankings in Table 5 may differ on other instances, and the p-values in Table 7 should be read accordingly. Multi-pool and multi-metadata replication is specified in Supplementary Appendix D and remains an open empirical requirement.

## 9.4 Cost Representation

Cost is a disclosed synthetic proxy (Section 6.2), not a billing trace.

## 9.5 Location Proxy

Haversine distance is a physical-distance proxy for network latency (Section 2.3).

## 9.6 Service-Class Abstraction

The 10 service classes are a deterministic experimental abstraction, fixed once per master seed and shared across all algorithms and seeds in Section 7.1. They are formed by partitioning a random pool rather than from functional service categories, which limits external validity.

## 9.7 Computational Environment and Runtime

Supplementary Appendix B reveals the complete computing environment on a single core of a Linux x86\_64 container: Python 3.12.3, NumPy 2.4.4, SciPy 1.17.1, pandas 2.x, and Matplotlib (figures only). There is no reliance on pymoo (Sec. 6.5). The entire experimental driver (run\_all.py), which includes all sensitivity studies (policy sensitivity, ablation, the 25-cell threshold grid, staleness, noise, location, objective-weight, and forecast-error sweeps), finishes in a few minutes on a single core. The 360-run Section 7.1 comparison took about 125 seconds in total.

## 9.8 Implementation Scope, Statistical Scope, and Untuned Optimizers

The optimizers in Section 7.1 are not reference-library implementations; instead, they are implementations constructed from the Section 4 formulations and standard published descriptions; as a result, the results are unique to particular implementations. All comparisons employ a limited budget

(population 30, 50 iterations), MOPSO is scored on the scalar F, and the NSGA-II-lineage GA has not been cross-checked against an independent NSGA-II library implementation. Only the sensitivity analysis in Table 6 uses fixed-penalty imputation, and statistical tests are descriptive and within-instance (Section 6.5).

## 9.9 Objective, Metadata, and Predictor Scope

Because the composite objective depends on weights that are policy or configuration aspects, F cannot be compared across policies or weight settings (Table B1). The synthetic metadata model fixes the expected relationship between risk, trust, cost, and QoS, which was not changed. The AE-LSTM predictor is not evaluated, and QoS values are taken to be as observed (Section 7.9). Only noise on RT is employed to simulate prediction error. Regarding N, M, I, or K, no empirical scaling study has been published.

## 10. Reproducibility and Data/Evidence Availability

Preprocessing and metadata-generation code (common.py), four optimizer implementations (algorithms.py), the experiment driver (run\_all.py), raw results, convergence logs, statistical outputs, environment data, and the evidence-mapping exclusion log are all included in the study release; the files and seeds are listed in Supplementary Appendix B. The original distribution of WS-DREAM Dataset 1 is accessible [1]. Supplementary Appendix D's extended experiments are described but not presented here.

## 11. Conclusion

Two complimentary additions are made in this study. The first is a constrained service-composition framework that is optimizer-agnostic: SecureL-DLSC enforces SR and PE as hard feasibility constraints under customizable policy scenarios while integrating QoS, location, trust, security risk, and privacy exposure. These restrictions are comparable to per-service admissibility filters under max-aggregation; unguided search achieves feasibility with probability equal to the product of per-class pass fractions; and gentle penalties by themselves cannot ensure compliance. The tests demonstrate how policy thresholds affect the feasible region and how stale metadata can result in silent breaches when decision evidence is not updated. These results show restricted optimization behavior under transparent situations rather than the security posture of actual providers because security/privacy/trust/cost metadata are artificial proxies.

Methodological and evidential contributions make up the second contribution. 0/5,825 of the 5,825 WS-DREAM Dataset 1 services are found to be admissible under the specified external-evidence standard by the paper's evidence-admissibility protocol. The explanation is structural: the product/version or comparable named-offering identity required for the vulnerability, assurance, and privacy evidence sources taken into consideration is missing from the benchmark, which provides QoS, geographic, and provider fields. In addition to providing a repeatable screening pattern for upcoming security-aware benchmark studies, this outlines what the benchmark can and cannot support.

The results of the comparative optimizer are single-instance and implementation-level. The feasibility analysis primarily credits black-box constraint handling rather than the formulation for the restricted GA's higher goal values under the viable policies and feasible solutions under the Strict scenario, whereas the PSO, MOPSO, and random search do not. Without fixed-penalty imputation in the primary analysis, feasibility is provided independently from conditional objective quality.

## Declarations

### Funding Declaration

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Future work will:** (i) carry out the multi-instance replication and hard-versus-soft comparison described in Supplementary Appendix D; (ii) add a candidate-filtering baseline and an exact per-class reference optimum; (iii) vary the dependence between security metadata and QoS and report QoS–risk trade-off curves; (iv) substitute evidence-backed metadata from datasets containing service/product identity for synthetic proxies and try provider-level registry matching; (v) calibrate the staleness model against past vulnerability-record changes; and (vi) assess the predictor under the constrained objective. These expansions investigate if the observed behavior of the framework extends outside of the current controlled WS-DREAM setup.

## Author Contribution

Alshaimaa M. Mohammed: Conceptualization; methodology; formal analysis; investigation; software; data curation; experimental design; implementation; validation; visualization; writing—original draft; writing—review and editing; and project administration. The author developed the SecureL-DLSC framework and evidence-admissibility protocol, designed and conducted the experimental evaluation, analyzed the results, and prepared and approved the final manuscript.

## References

1. Zheng Z, Zhang Y, Lyu MR (2010) Distributed QoS Evaluation for Real-World Web Services, in Proc. IEEE International Conference on Web Services (ICWS), pp. 83–90
2. Haytamy S, Omara FA (2020) A deep learning based framework for optimizing cloud consumer QoS-based service composition. *Computing* 102(5):1117–1137
3. Mohammed AM, Haytamy SSA, Omara FA (2023) Location-aware deep learning-based framework for optimizing cloud consumer quality of service-based service composition. *Int J Electr Comput Eng* 13(1):638–650

4. Kennedy J, Eberhart R (1995) Particle swarm optimization, in Proc. IEEE International Conference on Neural Networks, pp. 1942–1948
5. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
6. Coello Coello CAC, Lechuga MS (2002) MOPSO: A proposal for multiple objective particle swarm optimization, in Proc. IEEE Congress on Evolutionary Computation, pp. 1051–1056
7. Radack SM (2007) The Common Vulnerability Scoring System (CVSS). NIST ITL Bulletin, National Institute of Standards and Technology
8. Franklin J, Booth H, Wergin C, Guidance CVSSI (2014) NISTIR 7946 Natl Inst Stand Technol. 10.6028/NIST.IR.7946
9. Alliance CS (2026) Security, Trust, Assurance and Risk (STAR) Program
10. Alliance CS (2026) Cloud Controls Matrix (CCM) and Consensus Assessments Initiative Questionnaire (CAIQ) v4.1
11. Zeng L, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H (2004) QoS-aware middleware for Web services composition. IEEE Trans Software Eng 30(5):311–327
12. Ardagna D, Pernici B (2007) Adaptive service composition in flexible processes. IEEE Trans Software Eng 33(6):369–384
13. Canfora G, Di Penta M, Esposito R, Villani ML (2005) An approach for QoS-aware service composition based on genetic algorithms, in Proc. Genetic and Evolutionary Computation Conference (GECCO), pp. 1069–1075
14. Alrifai M, Risse T (2009) Combining global optimization with local selection for efficient QoS-aware service composition, in Proc. International World Wide Web Conference (WWW), pp. 881–890
15. Jula F, Sundararajan E, Othman Z (2014) Cloud computing service composition: A systematic literature review. Expert Syst Appl 41(8):3809–3824
16. Costante E, Paci F, Zannone N (2013) Privacy-aware web service composition and ranking, in Proc. IEEE International Conference on Web Services (ICWS), pp. 131–138. 10.1109/ICWS.2013.27
17. Mouratidis H, Islam S, Kalloniatis C, Gritzalis S (2013) A framework to support selection of cloud providers based on security and privacy requirements. J Syst Softw 86(9):2276–2293
18. Ghosh N et al (2015) SelCSP: A framework to facilitate selection of cloud service providers. IEEE Trans Cloud Comput 3(1):66–79. 10.1109/TCC.2014.2328578
19. Garg SK, Versteeg S, Buyya R (2013) A framework for ranking of cloud computing services. Future Generation Comput Syst 29(4):1012–1023
20. National Institute of Standards and Technology National Vulnerability Database (NVD), <https://nvd.nist.gov>
21. Cybersecurity IS, Agency Known Exploited Vulnerabilities Catalog, <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>

22. Joint Task Force Transformation Initiative (2012) Guide for Conducting Risk Assessments, NIST Special Publication 800 – 30, Revision 1
23. Friedman M The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc, 32, 200, pp. 675–701, 1937.
24. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bull 1(6):80–83
25. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70
26. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
27. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:2–4
28. Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Comput Methods Appl Mech Eng 191:11–12
29. Forum of Incident Response and Security Teams (FIRST) (2019)
30. Forum of Incident Response and Security Teams (FIRST) (2023)