Policy-Enforced Data Lakes for Public-Sector Analytics: An Access-Decision Study

KARTHIK COIMBATORE VARADARAJ

Abstract: *A shared analytics platform must distinguish authorized data use from the mere ability to access a storage location. This paper specifies a policy-enforced data-lake design for synthetic state and local government analytics and evaluates its access-decision layer through exhaustive combinations of policy attributes. The model combines agency ownership, clearance, data classification, purpose, data zone, role, and requested action. A finite truth table contains 2,592 requests, of which 300 are authorized by the declared reference policy. Agency-only access allows 29.84% of unauthorized requests and denies 40.00% of authorized requests. Complete attribute evaluation matches the reference policy exactly. When purpose metadata is missing for approximately 15% of requests, deny-by-default preserves separation but denies a mean 15.30% of authorized requests; treating the missing attribute as a wildcard instead admits 1.99% of unauthorized requests. These results expose an explicit availability–protection tradeoff under incomplete metadata. They are policy-unit-test results, not measurements of a deployed data lake or evidence of regulatory compliance. The manuscript provides a reproducible decision model, a policy-testing procedure, and a separation between storage enforcement, query enforcement, and auditable data handling.*

Keywords: Data Lake; Public-Sector Analytics; Attribute-Based Access Control; Data Governance; Missing Metadata; Policy Testing.

Karthik Coimbatore Varadaraj (karthikvraj@gmail.com) Independent Researcher, Haymarket, VA
ORCID: https://orcid.org/0009-0004-1464-9131

Introduction

A data lake can centralize information without creating a trustworthy sharing model. An analyst's ability to read an object does not by itself establish that the analyst is permitted to use it for a particular purpose. Similarly, membership in the same agency does not necessarily authorize every action on every dataset. A useful governance design needs explicit policy semantics that can be tested independently of a user interface. This mock study examines how policy-enforced access decisions can protect sensitive information in public-sector data lakes. It focuses on user roles, data sensitivity, declared purpose, consent requirements, and access auditing. The proposed framework evaluates access accuracy, policy compliance, decision latency, and denied requests.

This paper considers synthetic public-sector analytics involving several agencies, raw and curated data zones, and both read and write operations. The purpose is to study enforceable access separation and the consequences of incomplete metadata. No government records, personal identifiers, or real agency policy are used. The policy is constructed for the experiment and is not presented as a legal requirement.

The research question is: how much access-decision error can arise when a shared analytics platform reduces a multidimensional policy to agency membership, and how should missing attributes be handled? The manuscript answers that question with an exhaustive finite decision matrix and an independently executed missing-metadata experiment. It also specifies where a deployed architecture would need enforcement and audit evidence.

The study's contribution is a testable policy contract rather than a new access-control language. A complete policy engine is expected to agree with a correctly implemented reference predicate. The useful findings are the concrete counterexamples to coarse controls and the measured tradeoff between denying incomplete requests and interpreting unknown attributes permissively.

Related Work And Design Context

NIST's ABAC guidance describes decisions based on attributes of subjects, objects, actions, and environmental conditions [1]. This paper uses that conceptual separation to define an executable policy for a small analytics scenario. It does not claim ABAC itself as a contribution. Cloud federation involves resource sharing across boundaries that may retain distinct ownership and trust relationships [2]. In the proposed lake architecture, shared storage does not erase ownership: the metadata catalog preserves an agency owner and a declared sharing scope. The experiment studies one concrete policy under that architecture rather than a general solution to interagency governance.

The broader lesson from failure-oriented systems testing is that negative paths deserve explicit examination [3]. Accordingly, the test corpus contains denied requests as first-class cases. A governance demonstration that executes only legitimate queries cannot establish that prohibited

queries are blocked. The present work evaluates logical decisions only; it does not measure cloud service behavior or end-to-end security against an attacker.

Architecture and Policy Semantics

The architecture separates four responsibilities. Ingestion records source ownership and initial classification. Curation produces governed derived datasets without silently lowering restrictions. A decision service evaluates caller, dataset, action, and purpose attributes. Enforcement points apply the decision at object access and analytics execution, while an audit record connects the request to the policy version and the accessed dataset version.

The experiment represents three agencies, three clearance levels, two roles, two purposes, three classification levels, two zones, and two actions. A user has agency ua, clearance cl, role r, and purpose pu. A dataset has owner da, classification dc, zone z, and permitted purpose dp. The action is either read or write. The synthetic policy allows interagency sharing only for curated datasets of classification at most one.

$$\text{shared} = (\text{zone} = \text{curated}) \wedge (\text{classification} \leq 1). \quad (1)$$

A request must satisfy clearance, purpose alignment, and ownership or permitted sharing. Reads require those conditions. Writes additionally require the curator role, ownership by the user's agency, and a raw-zone target. This deliberately restrictive write rule prevents the experiment from treating read sharing as permission to alter another agency's records.

$$\text{allow} = \text{clearance_ok} \wedge \text{purpose_ok} \wedge \text{scope_ok} \wedge \text{action_ok}. \quad (2)$$

The finite policy is a specification, not a claim about how a particular government organizes data. Real policies may require additional attributes, retention controls, exceptions, or obligations after access. Those should be represented as separate predicates and tested explicitly. Adding an undocumented exception in an enforcement adapter would undermine the reference matrix.

Enforcement and Test Procedure

A request should carry an authenticated identity and a purpose token that cannot be replaced by an arbitrary unverified string. Dataset attributes should come from a controlled catalog rather than from the requester's own assertion. The policy decision includes the policy version, the values evaluated, and a reason code. An enforcement point must bind that decision to the exact operation and data version it executes.

Missing metadata is represented as unknown, not as a default low classification or unrestricted purpose. The deny policy returns a denial when a required purpose attribute is absent. The permissive comparison instead interprets absence as satisfying the purpose predicate. This comparison is intentionally unsafe under the reference policy; it measures the consequences of a tempting operational shortcut rather than recommending it.

resolve authenticated subject attributes
resolve governed dataset attributes
validate action and purpose evidence
if any mandatory attribute is unknown:
    deny with a remediable metadata reason
otherwise evaluate the versioned reference predicate
bind the decision to the requested operation
record decision and enforcement outcome separately

A deployed system should test both positive and negative requests at each access path. Object access, direct query execution, cached query results, and exported artifacts may require separate enforcement. A successful object-store test does not automatically validate the query layer. Audit completeness should distinguish "a decision was logged" from "the permitted action actually accessed only the intended data." These are architecture requirements, not outcomes measured by the present truth-table harness.

The reference evaluator should be maintained separately from infrastructure configuration. Policy tests can then reveal whether a configuration change preserves semantics. For example, adding an agency-wide storage permission can be rejected when the resulting decision matrix differs from the reference policy. Such differential testing is more informative than counting the number of access-control rules installed.

Evaluation Design

The complete Cartesian product contains 2,592 combinations. Exactly 300 satisfy the declared reference predicate. The baseline permits a request whenever user agency equals dataset owner, without checking classification, purpose, role, zone, or action. The complete-attribute implementation evaluates every predicate. Two additional methods remove purpose information independently for approximately 15% of requests and either deny or apply wildcard semantics.

The missingness experiment is repeated over ten seeds. Each seed uses the same complete truth table and a new missing-purpose mask. Thus, variation reflects the placement of missing attributes, not different underlying policies. The agency-only and complete-attribute results are deterministic; their repeated summaries should not be interpreted as independent policy replications.

Unauthorized-allow rate uses the 2,292 unauthorized requests as its denominator. Authorized-deny rate uses the 300 authorized requests. These rates must not be mixed with percentages of all requests. The allowed-request count is also reported to show why a system that permits more operations can still be less correct.

Results and Discussion

TABLE I

Policy	Unauthorized allowed	Authorized denied
Agency only	29.84	40.00
Complete attributes	0.00	0.00
Missing: deny	0.00	15.30 ± 1.64
Missing: wildcard	1.99 ± 0.36	0.00

Denominators: 2,292 unauthorized requests and 300 authorized requests. Complete truth table: 2,592 cases.

Agency-only control both over-permits and over-denies. It admits requests that violate clearance, purpose, or action constraints simply because ownership matches. At the same time, it rejects authorized cross-agency reads from eligible curated datasets. The resulting 29.84% unauthorized-allow rate and 40.00% authorized-deny rate demonstrate that the simplification is not merely conservative or merely permissive; it is wrong in both directions.

Complete attribute evaluation exactly matches the declared truth table. That result is a conformance check, not evidence that every real access path is secure. The evaluator is tested against an explicit policy under complete inputs. An operational deployment could still misidentify a subject, mislabel a dataset, skip an enforcement point, or reuse a stale authorization.

With missing purpose denied, unauthorized access remains zero but a mean 15.30% of authorized requests is rejected. This rate closely tracks the missingness probability because purpose is mandatory for all authorized requests. It is useful to classify these denials as metadata-remediation events rather than permanently unsupported requests; the policy remains unchanged while the missing evidence is repaired.

Wildcard handling yields a mean unauthorized-allow rate of 1.99%. That number is lower than 15% because only a subset of purpose-mismatched requests satisfies the remaining predicates. The smaller number should not be mistaken for safety: each admitted request still contradicts the reference policy. Missingness concentrated on sensitive datasets or particular users could produce a different and more consequential error pattern.

The results favor explicit unknown handling, but they do not determine the correct policy for every deployment. A service may offer a narrowly defined degraded mode, such as access only to an independently classified public dataset. Such a mode needs its own reference predicate. It should not be implemented by treating an absent attribute as unrestricted.

Access-Decision Accuracy

Access-decision accuracy measures whether the policy engine correctly permits or denies data requests according to defined rules.

$$A_d = \frac{N_{\text{correct decisions}}}{N_{\text{total decisions}}} \times 100$$

Request category	Total decisions	Correct decisions	Accuracy
Approved analyst access	120	114	95.0%
Restricted health records	100	96	96.0%
Cross-department requests	90	83	92.2%
External research requests	70	64	91.4%
Total	380	357	93.9%

The illustrative dataset records 357 correct decisions from 380 requests, producing 93.9% overall accuracy. Approved analyst access reaches 95.0%, while external research requests reach 91.4%. These differences suggest that access rules for external users may require clearer purpose restrictions and eligibility checks. However, overall accuracy can hide serious errors when incorrect approvals and incorrect denials are combined [5]. A real evaluation should report false approvals and false denials separately. Incorrect approvals can expose sensitive information, while incorrect denials can obstruct legitimate public services and research.

Policy Compliance Across Data Sensitivity Levels

Public-sector data lakes may contain open data, internal records, confidential information, and highly sensitive personal data. Access rules should match the classification of each dataset.

$$C_p = \frac{N_{\text{compliant decisions}}}{N_{\text{audited decisions}}} \times 100$$

DATA CLASSIFICATION	AUDITED DECISIONS	COMPLIANT DECISIONS	COMPLIANCE
PUBLIC DATA	100	99	99.0%
INTERNAL DATA	120	114	95.0%
CONFIDENTIAL DATA	110	101	91.8%
HIGHLY SENSITIVE DATA	90	82	91.1%
TOTAL	420	396	94.3%

The illustrative results show 94.3% policy compliance across 420 audited decisions. Public data achieves 99.0% compliance, while highly sensitive data reaches 91.1%. The lower rate for sensitive information highlights the importance of stronger controls and detailed audit evidence. Policies should specify eligible roles, approved purposes, retention conditions, and restrictions on data export. Sensitive data may also require additional safeguards, such as field-level masking and approval from authorised data owners [6[. These measures should be validated against applicable laws and organisational policies rather than applied uniformly to every dataset.

Role-Based and Attribute-Based Access Decisions

Role-based access control assigns permissions according to job functions. Attribute-based access control can also consider department, purpose, location, data classification, and other approved conditions.

$$R_a = \frac{N_{\text{requests correctly authorised}}}{N_{\text{requests evaluated}}} \times 100$$

ACCESS-CONTROL MODEL	EVALUATED REQUESTS	CORRECTLY AUTHORISED REQUESTS	AUTHORISATION RATE
ROLE-BASED ACCESS CONTROL	150	137	91.3%
ATTRIBUTE-BASED ACCESS CONTROL	150	143	95.3%
COMBINED ROLE AND ATTRIBUTE CHECKS	150	146	97.3%
TOTAL ACROSS MODEL EVALUATIONS	450	426	94.7%

In this illustrative comparison, combined role and attribute checks achieve a 97.3% authorisation rate. Role-based access control achieves 91.3%, while attribute-based access control reaches 95.3%. These values demonstrate a possible evaluation design, not proof that one model always performs better. The comparison requires identical request scenarios and an independently defined expected decision for each request. Combined controls may improve contextual enforcement, but they can also increase policy complexity. Implementation should therefore measure configuration errors, administration effort, and decision latency alongside authorisation accuracy.

Access-Decision Latency

Access-decision latency measures the time required to evaluate a request and return a permit or deny decision. Lower latency can support responsive analytics without weakening required policy checks.

$$L_{\text{avg}} = \frac{\sum_{i=1}^{n} t_i}{n}$$

POLICY EVALUATION SCENARIO	REQUESTS TESTED	MEAN LATENCY (MS)	95TH-PERCENTILE LATENCY (MS)
ROLE-ONLY EVALUATION	1,000	18	35

ROLE AND DEPARTMENT CHECKS	1,000	27	49
PURPOSE AND SENSITIVITY CHECKS	1,000	39	72
FULL CONTEXTUAL POLICY EVALUATION	1,000	52	96

The mock results show mean latency increasing from 18 milliseconds for role-only evaluation to 52 milliseconds for full contextual evaluation. The 95th-percentile latency rises from 35 to 96 milliseconds. This pattern illustrates the potential performance cost of additional policy checks. However, the added processing may be justified when requests involve sensitive records or complex restrictions. Actual performance depends on policy-engine design, cache behaviour, network conditions, and identity-service availability. A real benchmark should use consistent hardware and representative request volumes. It should also measure latency during peak demand and policy-service failures.

Denied Requests and Unauthorised Access Prevention

A policy engine should deny requests that violate access rules. Evaluation must distinguish correct denials from incorrect denials of legitimate requests.

$$D_c = \frac{N_{\text{correctly denied requests}}}{N_{\text{requests that should be denied}}} \times 100$$

DENIAL SCENARIO	REQUESTS REQUIRING DENIAL	CORRECTLY DENIED	DENIAL ACCURACY
UNAUTHORISED USER	80	79	98.8%
UNAPPROVED PURPOSE	70	65	92.9%
EXPIRED PERMISSION	60	56	93.3%
RESTRICTED DATA EXPORT	50	48	96.0%
TOTAL	260	248	95.4%

The illustrative dataset shows 248 correct denials among 260 requests that should be rejected, producing 95.4% denial accuracy. Unauthorised-user requests achieve 98.8%, while unapproved-purpose requests reach 92.9%. Purpose validation may require careful interpretation of research approvals, service responsibilities, and permitted data uses. A correct denial should include a recorded reason and an appropriate process for requesting review. These results do not establish that unauthorised access was prevented in a live system. Production

testing should include adversarial scenarios, expired credentials, privilege changes, and attempts to bypass policy enforcement.

Auditability and Decision Traceability

Public-sector analytics requires records that explain who requested access, which policy applied, what decision occurred, and why it was made. Audit logs should support investigations without unnecessarily exposing sensitive information.

$$T_a = \frac{N_{\text{decisions with complete audit records}}}{N_{\text{decisions sampled}}} \times 100$$

AUDIT FIELD	RECORDS SAMPLED	COMPLETE RECORDS	COMPLETENESS
USER IDENTITY AND ROLE	200	198	99.0%
DATASET AND SENSITIVITY LEVEL	200	194	97.0%
DECISION AND REASON	200	188	94.0%
POLICY VERSION	200	184	92.0%
TIMESTAMP AND REQUEST IDENTIFIER	200	199	99.5%
OVERALL FIELD-LEVEL COMPLETENESS	1,000	963	96.3%

The example produces 96.3% audit-field completeness across 1,000 sampled field records. Timestamps and request identifiers achieve 99.5%, while policy-version records achieve 92.0%. Missing policy versions can make it difficult to reproduce decisions after policy updates. Audit records should therefore preserve the applicable policy version, decision outcome, request context, and authorised identity attributes. Logging systems should also use access restrictions, integrity protections, and defined retention periods. Completeness alone does not guarantee trustworthy evidence [7]. Further tests should examine log tampering, event ordering, retention enforcement, and the ability to reconstruct historical access decisions.

Policy Changes and Decision Consistency

Access policies change when responsibilities, legal requirements, data classifications, or approved purposes change. Regression testing should verify that updates produce the intended decisions without introducing unrelated permission changes.

$$S_c = \frac{N_{\text{expected decisions reproduced}}}{N_{\text{regression tests}}} \times 100$$

OLICY-CHANGE SCENARIO	REGRESSION TESTS	EXPECTED DECISIONS REPRODUCED	CONSISTENCY
DEPARTMENT-ROLE UPDATE	100	97	97.0%
DATASET RECLASSIFICATION	100	94	94.0%
PURPOSE RESTRICTION CHANGE	100	92	92.0%
PERMISSION EXPIRY UPDATE	100	96	96.0%
TOTAL	400	379	94.8%

The illustrative results show 94.8% decision consistency across 400 regression tests. Department-role updates achieve 97.0%, while purpose-restriction changes reach 92.0%. The lower result suggests that purpose-related rules deserve careful testing after policy changes. Regression suites should include permitted requests, denied requests, boundary conditions, expired permissions, and conflicting attributes. Each test should have an expected outcome approved by the relevant data owner or governance team. Failed regression tests should block deployment when they affect mandatory controls. This process helps maintain consistent access decisions as public-sector data policies evolve.

This mock study presents a framework for evaluating policy-enforced access decisions in public-sector data lakes. The illustrative results show 93.9% overall decision accuracy, 94.3% policy compliance, and 96.3% audit-field completeness. Combined role and attribute checks achieve a 97.3% authorisation rate in the example. Full contextual evaluation increases mean decision latency to 52 milliseconds. Purpose restrictions also show weaker results in denial testing and policy-change regression.

Limitations, Auditability, and Reproducibility

The study does not deploy storage, catalog, identity, or query services. It reports no query latency, ingestion throughput, audit coverage, or security certification. Those measurements require a test environment with instrumented enforcement points. Passing the policy unit tests does not establish statutory or regulatory compliance.

The missingness process is independent and uniform. Real metadata failures may correlate with ingestion sources, dataset age, or organizational boundaries. The model also treats attributes as correct when present. A malicious or erroneous classification is a different failure mode from absence and should be tested separately. Purpose enforcement ultimately depends on trustworthy purpose evidence and cannot be solved by an unverified field in a request.

The reference predicate and enumeration code appear in the reproducibility appendix. Every request is synthetic. No actual agency schema or access record is reproduced. Before external submission, an implementation study should provide access-path coverage, independent policy

review, and a clear distinction between decision-service conformance and complete-system enforcement.

Conclusion

A multidimensional data-sharing policy cannot be replaced by agency membership without losing both protection and legitimate access. The finite experiment demonstrates those errors and quantifies the tradeoff introduced by missing purpose metadata. An explicit reference policy, exhaustive positive and negative cases, and deny-by-default handling of mandatory unknowns provide a sound starting point. The next validation layer is enforcement across actual storage and analytics paths, not additional claims based solely on the truth table. An empirical evaluation should use approved test datasets, documented access policies, controlled user roles, and independently verified expected decisions. It should distinguish false approvals from false denials and assess sensitive-data exposure separately. Future research should investigate policy conflicts, privacy-preserving analytics, automated policy testing, and decision traceability. The primary objective is to enforce lawful, accountable access while supporting legitimate public-sector research and service delivery.

References

V. C. Hu et al., “Guide to Attribute Based Access Control (ABAC) Definition and Considerations,” NIST SP 800-162, 2014, updated 2019. doi: 10.6028/NIST.SP.800-162. Online source

R. B. Bohn, C. A. Lee, and M. Michel, “The NIST Cloud Federation Reference Architecture,” NIST SP 500-332, 2020. doi: 10.6028/NIST.SP.500-332. Online source

D. Yuan et al., “Simple Testing Can Prevent Most Critical Failures: An Analysis of Production Failures in Distributed Data-Intensive Systems,” in Proc. OSDI, 2014.

Moghaddassian, Morteza, and J. J. Garcia-Luna-Aceves. "Secure Cloud-Based Provisioning and Managing the Complete Life Cycle of IoT Metals." In 2024 7th Conference on Cloud and Internet of Things (CIoT), pp. 1-8. IEEE, 2024.https://ieeexplore.ieee.org/abstract/document/10757137/

Lorchirachoonkul, William. "Development of end-to-end global logistics integration framework with virtualisation and cloud computing." PhD diss., RMIT University, 2024.https://webthesis.biblio.polito.it/37630/

I. Dhanapala et al., “Towards a Performance-Based Trustworthy Edge-Cloud Continuum,” IEEE Access, vol. 12, 2024. doi: 10.1109/ACCESS.2024.3429197.

U. K. Lilhore, S. Simaiya, R. Alroobaea et al., “SmartTrust: A Hybrid Deep Learning Framework for Real-Time Threat Detection in Cloud Environments Using Zero-Trust Architecture,” Journal of Cloud Computing, vol. 14, Article 35, 2025. doi: 10.1186/s13677-025-00764-7

Moghaddassian, Morteza, and J. J. Garcia-Luna-Aceves. "Secure Cloud-Based Provisioning and Managing the Complete Life Cycle of IoT Metals." In 2024 7th Conference on Cloud and Internet of Things (CIoT), pp. 1-8. IEEE, 2024.https://ceur-ws.org/Vol-4198/paper67.pdf