BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

From Telemetry to Settlement: Evidence Validation, Parametric Payment, and Controlled Recovery for Cloud SLA Breaches

[1]Utkarsha Kute,[2]Prof. Sandeep Kumar Jaiswal

[1,2]Department of Artificial Intelligence and Data Science
Ajeenkya DY Patil School of Engineering, Pune , India · India

Abstract

When there is downtime in the cloud, two questions arise: is it a breach of a service-level agreement (SLA), and what must be done about it? We propose an architecture that connects telemetry with evidence assessment, parametric payment, and controlled recovery through a common incident record. Metrics are compared with a versioned policy, vigilant witnesses certify the evidence, and a smart contract enforces the principle of bounded payment. A separately authorized "director" can perform an allowed recovery operation if needed. The design deals directly with missing observations, duplicate claims, contradictions between witnesses, confidentiality, emergency suspension, and the danger of making a downtime worse through automated recovery. It also distinguishes the integrity of evidence from the fidelity of measurement. The paper is a design and evaluation plan; it reports no experimental results.

Keywords : Cloud SLA, evidence validation, oracles, parametric insurance, recovery automation, smart contracts, telemetry

I. Introduction

Cloud computing failures can occur for many reasons, some of which may not be obvious. A software defect, resource depletion, a regional disruption, a network outage, or a configuration error may each cause an outage. Regardless of the cause, the first thing that must be done is to restore the service. After that, a separate investigation must determine whether contractual obligations were breached.

The two tasks are related, but neither determines the success of the other. Restoring the service does not erase a prior violation, and receiving a payout from an insurer does not help the application recover and reconnect.

SLAs typically address availability, latency, and response time. Parametric insurance policies set specific triggers and rules for compensation. This works only if the parties share an understanding of how an incident is measured, where the data come from, and how disputes are resolved.

The proposed approach connects detection, evidence validation, payment, and recovery through a common identifier. The identifier lets an auditor trace a payment and a recovery action back to the same event, while separate authorizations prevent the financial contract from taking over the cloud resources.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

The aim is not to make every decision automatic but to define the evidence and the boundaries between decisions.

This paper makes four contributions: (i) a layered architecture that turns evidence into settlement; (ii) a bounded severity and payout formulation; (iii) a claim state machine for duplicates, disputes, and pauses; and (iv) a reproducible testing plan. The study remains at the conceptual stage. The formulas and cases describe how a prototype should be tested; they do not represent the outcome of a built implementation.

A. Problem and Research Questions

An effective system has to establish what a breach is, how long it lasts, who can be trusted to provide measurements, how repeated submissions are prevented, and what measures ensure that recovery efforts do not aggravate the problem. This gives rise to four research questions. **RQ1:** How can telemetry be turned into a reproducible parametric claim? **RQ2:** What controls prevent manipulation of the system, replay of data, and duplicate payments? **RQ3:** How can settlement and recovery share evidence without violating their distinct authorizations? **RQ4:** What measurements can indicate whether the design allows contracts to improve responses without unacceptable risks?

B. Limits and Assumptions

The design assumes that, before coverage commences, the service in scope, the insured characteristics, and the roles of the parties concerned have been unanimously agreed. Signing keys and telemetry collection are assumed to be under proper control, but they cannot be trusted entirely. For example, a compromised agent, colluding reporters, or an incorrect policy can still trigger adverse decisions. The enforceability of the product and of insurance regulations depends on the jurisdiction and must be examined outside the technical design.

II. Background and Related Work

A. Telemetry and Evidence Quality

Prometheus collects metrics and issues alerts, while OpenTelemetry offers vendor-neutral instrumentation and telemetry export [1], [2]. These systems provide visibility into how a service behaves, but receiving an alert does not mean that an insurance incident has occurred. The insurance policy must specify several components, including the service in question, the sampling interval, the aggregation rule, the threshold, maintenance exclusions, and the way in which missing data are treated.

An evidence record should contain the information used for the specific decision and a means of verifying it. The record may include the total availability over a given period, the number of failed probes, the sources of information, the configuration version, and a digest of the supporting documents. A digest allows a reader to find out whether the information was changed afterwards; however, it does not remove concerns about the authenticity of the original measurement, clock synchronization, and measurement accuracy.

B. Parametric Contracts and Policies

A parametric policy differs from others in that it bases its payments on the occurrence of a specified index or event rather than waiting until after the damage has occurred. Its disadvantage is basis risk, that

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

is, the index might not accurately describe the losses incurred by the claimant. Smart contracts can execute deterministic rules once all information has been provided, but they cannot monitor a cloud service or determine whether the conditions reported by a sensor are valid. Observing outside information therefore requires an oracle or another confirmation method.

Mistakes in a policy can cause numerous problems later, because they may be programmed into the contract. Threshold values and rules for exceptional cases need to be defined properly; otherwise they will influence every claim made. The definitions must be agreed before the system is launched.

C. Oracles and Recovery Automation

Multiple reporters can reduce dependence on a single source only if the reporters are truly independent. Measuring the same compromised endpoint five times does not produce five measurements. Quorum rules tolerate some faulty reports, but they do not address collusion or correlated failures [3]. The selection of reporters, the rotation of keys, late reports, and the resolution of disagreements therefore have to be designed into the system.

Automation tools can restart workloads, move containers, and send traffic to healthy instances. Yet frequent restarts can cause overload, and a stale replica can lose fresh data, turning a small issue into a major incident. Recovery mechanisms should include tested runbooks, strict access controls, rate limiting, and dry-run and rollback options.

III. Proposed System Architecture

The architecture is divided into five logical layers: service infrastructure, telemetry, evidence validation, settlement, and recovery. Each layer has a specific function and exchanges information using claim identifiers. This separation is intended to keep decisions transparent and to prevent the recovery procedure from changing financial information.

A. Infrastructure and Telemetry

The infrastructure layer consists of the covered applications, cloud resources, health checks, and dependencies. Telemetry includes the successful-probe ratio, the request response time at a given percentile, the HTTP error ratio, CPU throttling, and saturation. The policy should differentiate contractual indicators from diagnostic signals. For example, high CPU use can explain a downtime incident but cannot initiate a payment unless the policy defines it as a covered index.

Sampling and aggregation must be defined before the incident. For instance, if availability is defined over five-minute windows, it should not be switched to one-minute windows to show better results during a failure. Missing observations require a clearly defined rule: they are voided, replaced by an independent probe, or escalated to a review queue.

B. Proof Packet

For every incident analyzed, an agent prepares a packet containing a unique claim ID, the policy ID and version number, the specific services, the start and end times of the incident, a metrics summary, source identifiers, a digest of the configuration, and a hash of the records. The packet is signed by its source wherever practicable. The original logs remain stored in an authorized location, while the contract

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

receives only the information needed for decision-making. The hash guarantees integrity, and signatures and provenance tracing support legitimacy, but neither indicates that the source is trustworthy.

C. Oracle Confirmation

In compliance with the requirements, agents tasked with reporting independently verify the claim against the policy and provide a signed report. The validation step checks the reporter's membership, the validity of the signature, the relevance of the policy, and whether a quorum has been reached. Reports that must be rejected include those with unknown policy IDs, invalid timestamps, or a claim ID that has been used previously. If there are significant inconsistencies between reports, the claim is placed in a dispute stage instead of being paid automatically.

Settlement contract. This contract is essential because it confirms that the corresponding policy is valid, that the claim has not already been paid, and that the amount is within the policy limits and the available reserve. A claim ID corresponds to one final settlement, which makes retries idempotent. Settlement records must carry the claim ID, the policy version, the amount, and the status of the decision, without placing sensitive operational information in a publicly readable form. The authority to pause and restart must be specified in detail; a pause should stop unsafe types of action rather than change claims that have already been settled.

Recovery controller. The controller receives a validated incident record, but it has its own permission regime. It maps the incident category to the appropriate runbook, checks the current state of the system, and produces only the actions allowed for it. It records whether the objective was restored and whether a rollback was performed.

IV. Risk and Parametric Settlement Model

A. Severity Parameters

Let $L(t)$, $E(t)$, and $C(t)$ denote the normalized severity of latency, error, and resource-use deviations, respectively. Each parameter is mapped to an observable range, usually from zero to one; the higher the value, the more serious the deviation from the service objective. Caps and guardrails prevent a large unprocessed value from dominating the score merely because of its unit of measurement. The composite severity is

$$R(t) = 1 - \exp\{-\delta[\alpha\, L(t) + \beta\, E(t) + \gamma\, C(t)]\} \qquad (1)$$

where α, β, and γ are model coefficients describing each component's influence on the service level. The coefficients are non-negative and sum to one, and δ sets the sensitivity of the model.

B. Payout Rule

A bounded payout can be defined as

$$P(\Delta\, t) = \min[P_{\max}, K \int \max(0, R(t) - R_{\text{th}})\, dt] \qquad (2)$$

where $P_{\max}$ is the policy ceiling, K is a multiplier, and R_{th} is the trigger threshold. In a discrete implementation, the integral is replaced by the sum of the excess above the threshold multiplied by the length of each interval over which the function is evaluated. The policy should state all required details,

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

such as time units, rounding, waiting period, threshold, and claim limits, including the treatment of incident grouping.

C. Example Calculation

Assume a policy with R_{th}=0.4, K equal to 100 currency units per severity-minute, and P_{max}=5000 units. If $R(t)$=0.6 for 10 valid minutes, the severity excess is 0.2, and the calculation becomes 100× 0.2× 10=200 units. This example only illustrates how the calculation works in practice; it is not an actual result or a suggested insurance amount. A practical policy will differ, as it must take into account the waiting period, uncertainty, and the correlation of the index with losses.

D. Policy Lifecycle

A policy passes through the stages Draft, Reviewed, Active, Suspended, Expired, and Closed. The policy record should include the version and activation time, the covered services, the measurement definitions, the trigger parameters, the payment limit, the oracle quorum, and the beneficiary. A claim should be attached to the policy version in force during the measurement window. A boundary test should be carried out for the moment immediately before activation.

V. Operational Workflow

The workflow commences once telemetry indicates a potential breach. Detection alone should not lead to money transfers or to any disruptive action. The event is mapped to an active policy and a fixed measurement interval; the evidence is packaged; oracle reports are collected; and the contract evaluates the claim. Failed checks normally stop further processing or direct the claim to review.

A. Claim Processing

The order of actions is as follows: (1) establish a potential breach based on the agreed persistence rule; (2) create a claim ID and an evidence summary; (3) collect signed attestations; (4) check timestamps, the policy version, the quorum, and uniqueness; (5) determine the amount and check that reserves are available; (6) settle an amount not exceeding the determined sum and produce an audit record; and (7) perform any approved recovery actions. Every phase of processing must be documented, with the reasoning for moving on or stopping.

B. State Transitions and Retries

The useful states are Monitoring, Candidate Breach, Evidence Collection, Oracle Review, Challenge, Approved, Settled, Recovery in Progress, Recovered, and Paused. Stale evidence goes back for review, conflicting reports move to the Challenge state, and trusted reports move to the settlement stage. The Paused state prohibits new actions while preserving prior ones.

Network calls and automated functions will retry. Without idempotency, a retry could cause a payment to be made again or a recovery action to be executed twice. Each claim ID should therefore represent a single final settlement, while the controller has its own action ID for every execution.

C. Human Oversight

The level of oversight should follow the consequences of a decision. Restarting a brief session, for example, may be automated after validation, but other operations, such as regional failover, database

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

promotion, or resource removal, require the presence of an operator. The controller therefore has to present the triggering condition, the expected impact, the actual status, and the rollback option. Human review is an additional safeguard; it does not replace good technical safeguards.

VI. Security, Privacy, and Threat Model

A. Adversary Assumptions

The threats considered include a malicious insider, a compromised telemetry agent, colluding reporters, replay of a legitimate claim, a compromised oracle key, and an attacker seeking control of the recovery controller. The threat model must define which actors are within the scope of the project and what it expects to prevent from those actors; otherwise, the security claim itself has no value.

B. Integrity Does Not Equal Truth

Cryptographic hashing can prove that a file has not changed since the hash was produced, and a digital signature connects a report to a signing key. Neither proves that the measurement is true. If the sensor was compromised before signing, false data can be preserved without difficulty. Independent probes, diversity of sources, signed configuration, time synchronization, and cross-checks reduce the risk but do not eliminate it entirely.

C. Contract and Privacy Risks

Contract defects can affect the transfer of money, lock funds, or overlook policy compliance. Countermeasures include correct arithmetic, strict rules on data access, predefined transitions, attention to reentrancy, and invariants: a claim is settled only once; the amount does not exceed a predetermined limit; the recipient is the party named as entitled; and evidence and quorum are mandatory for settlement. The code must be audited, the invariants verified, and the system inspected before it is put into operation.

Raw logs may contain sensitive customer data, such as connection paths, virtual machine names, or discovered vulnerabilities. Logs should not be linked to individual customers, but in practice a hash does not protect a customer's identity if the log can be reconstructed. Log storage must follow all applicable laws and regulations.

D. Recovery Security

Recovery credentials should be short-lived and valid only for the target service. Runbooks need to be thoroughly tested for failure, for the proper application of rate limits, and for the implementation of rollback plans. The controller must not be given blanket administrative rights simply because it interacts with the settlement service. A breach at the contract level should not imply unlimited access to the cloud, while a breach at the controller level should not allow a claim amount to be changed or funds to be sent in the wrong direction. These boundaries need to be tested negatively, rather than relying only on an architecture diagram.

VII. Implementation and Evaluation Plan

The prototype could use a containerized test harness with either Prometheus or OpenTelemetry, a smart contract on a test network, and a recovery agent restricted to a test environment. These components are

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

illustrative samples and do not imply that any such system is already in place. The first round of tests should use synthetic cases and synthetic financial values.

A. Prototyping Steps

First, the metrics, time intervals, limits, payout conditions, and rules for handling missing data should be identified. Second, normal and faulty traces should be introduced and the raw evidence saved. Third, reporter approval and freshness checks should be added, together with the handling of conflicting reports. Fourth, a contract that enables policy execution and claim verification should be created. After the settlement tests are completed, the recovery stage can be initiated.

B. Reproducibility and Scenarios

For reproducibility, each test run should document the software versions, the contract build, the policy configuration, the testing environment, the workload, the telemetry frequency, the set of oracles, and the injected fault. Scripts and raw outputs should be saved so that other researchers can repeat the tests and obtain similar results. The scenarios should include sustained latency and error-rate exceedances, resource depletion, a local service outage, absent telemetry, erroneous or contradictory reports, duplicated claims, insufficient reserves, a contract pause, and an unplanned runbook execution failure.

C. Metrics

The evaluation should compare at least three configurations: manual assessment and recovery, parametric settlement without automated recovery, and the combined configuration. Similar workloads and incidents should be used wherever possible. Medians and spreads should be reported rather than only the best results, and machine processing time should be separated from human evaluation time. The metrics should include detection lag, evidence preparation time, settlement lag, recovery time, false-positive rate, false-negative rate, duplicate rejection rate, resource overhead, and unintended effects. The percentage change in time is calculated as

$$\Delta T = [(T_{\text{baseline}} - T_{\text{proposed}}) / T_{\text{baseline}}] \times 100 \qquad (3)$$

When qualifying and non-qualifying cases are labelled, the false-positive rate is

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN}) \qquad (4)$$

where FP is the number of non-qualifying events incorrectly accepted and TN is the number of non-qualifying events correctly rejected. Research results must be based on test cases rather than rough estimates. If no prototype is available, the results section must remain a projection, without actual success rates presented.

VIII. Research Gaps and Practical Trade-offs

The first unresolved question concerns the quality of evidence. While the rules of a contract may be deterministic, the observables may not have the required reliability. More research is needed on measurement and calibration in disputes among independent parties that have no common trust in the monitoring provider. The second issue is basis risk: an availability threshold is easy to check, yet it may not represent the financial consequences for the policyholder well. In many cases, a delay in service may

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

stay below the point at which users abandon transactions, while a short outage may still trigger a payment event even where redundancy is available.

Another important issue is privacy. Keeping raw telemetry off the blockchain decreases the chance of exposure, but metadata can still disclose the timing of incidents, relationships between services, and overall business activity. Privacy-preserving cryptographic proofs may help; however, they complicate the process because what is being proved must itself be validated. In addition, recovery safety requires proof that the runbook works with the workload and that rollback works correctly under expected failures.

As a result, the design involves trade-offs. Distributed ledgers can facilitate auditing and protect the parties from relying on a single database, but they bring their own disadvantages, such as additional expense, delay, and complexity, and may even increase operational costs. Where the parties already trust the claims-processing service, a simple database record can be less expensive than a ledger.

The parties also have different interests. The insured party may prefer a permissive trigger, the insurer may want a conservative interpretation, and the operator may control the monitoring system used to assess its own performance. Governance should specify who chooses the measurements, who makes changes to the system, who examines the evidence, and who pays for independent review. Rules cannot replace a process for handling ambiguity, the possibility of fraud, or vaguely defined policy provisions.

IX. Deployment Scenario

Assume the API reports the 95th-percentile response time for five-minute periods. A breach occurs when the latency exceeds the threshold for the specified number of periods without any declared maintenance. Telemetry produces the claim record and keeps the backup data. The evidence service pairs the policy version with a digest and requests signatures. The contract checks the freshness and identity of the claim and the quorum. After consensus is reached, the claim may be processed; otherwise, it must be reviewed.

In parallel, the recovery manager monitors the health of the services. When it is safe according to the approved playbook, the recovery manager routes a small portion of traffic to a healthy pool and checks whether the action works. The move is recorded under a separate ID. If the service is restored after the transfer, the recovery is also recorded. If it does not work, the recovery manager reverts the transfer. Settlement is associated with the original measurement period rather than with the success of the recovery.

Now suppose one of the telemetry points is unreachable. Treating the lack of data as evidence of failure is risky from the legal perspective, whereas treating it as evidence of normal activity may hide a real issue. The rules should clearly indicate whether an independent test can be used, whether the missing data invalidate the window, or whether manual examination of the situation is necessary.

These examples demonstrate the interactions between the elements; however, they do not establish typical payment delay, resistance to oracle collusion, or security in production. Such evidence can only be provided by a prototype, a fault-injection definition, common measurements, and independent evaluation.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

X. Policy Governance and Operational Readiness

The main participants in a functioning deployment include the policyholder, the insurance company or risk carrier, the telemetry entity, and the entity responsible for the infrastructure. One organization may play several roles, but each role must have clearly defined duties and responsibilities. The policyholder must be able to monitor the claim measures; the insurer must decide on coverage and disputes; the telemetry entity must document how data are collected and retained; and the infrastructure entity must sanction recovery activities. This arrangement presumes some separation between the parties. If one side controls the source of the metrics, the oracle keys, and the payment recipients, very little separation is ensured.

A policy review should cover certain elements before activation: the identity of the service, the metric definition, the sampling period, maintenance exclusions, the trigger duration, the payout calculation, the cap, the beneficiaries, retention, the quorum, and the path for appeal. Any change to the threshold or formula must be recorded as a new version of the policy.

Claimants and insurers need a record containing the payment details, the policy expiry date, the measurement period, a metric summary, and the reason for approval or denial. In addition, any appeal should have a time limit and be submitted in the required format.

Commencement of production should involve proof that all key invariants hold, evidence that pause and key-rotation routines have been rehearsed, and evidence that the runbooks work when failures occur. The operator should simulate loss of oracle connectivity, unavailable storage, credential theft, and incorrect policy configuration. Monitoring must include the settlement service itself, as it can stop working even when the application it supports functions correctly.

XI. Future Work and Conclusion

The next stage is the development of a simple prototype that covers one service, one kind of breach, and one reversible countermeasure. This will allow the evidence format and the state machine to be tested before moving on to more indicators or more complex policies. Future experiments might test different oracle strategies under delayed, failed, corrupted, and contradictory data, and examine sensitivity to different thresholds and aggregation windows, as well as whether privacy-preserving proofs are worth the added complexity.

It is vital for the study to measure the human workload brought about by automation: the reduction in everyday effort might be offset by an increase in the number of exceptions and disagreement cases. Potential indicators are the number of cases that require intervention, the time needed to resolve disagreements, the quality of decision records, and the number of rejected dangerous recovery recommendations.

This paper proposes a design that links cloud telemetry with evidence validation, parametric payment, and a recovery mechanism. A central concern is how to make information accessible to all stakeholders while safeguarding financial and operational privacy. The design does not eliminate trust; it exposes the major assumptions and provides a mechanism for auditing them. Hash functions and signatures secure the integrity and origin of the data, but they do not prevent false data from being reported. Smart contracts can enforce the agreed rules automatically, but this does not by itself make an insurance arrangement fair.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Acknowledgment

The author acknowledges the guidance and support of Professor Sandeep Kumar Jaiswal, mentor for this work.

References

[1] Prometheus Authors, "First steps with Prometheus," *Prometheus Documentation*. [Online]. Available: https://prometheus.io/docs/introduction/first_steps/ [Accessed: Oct. 22, 2026].
[2] OpenTelemetry Authors, "What is OpenTelemetry?," *OpenTelemetry Documentation*. [Online]. Available: https://opentelemetry.io/docs/what-is-opentelemetry/ [Accessed: Oct. 22, 2026].
[3] Chainlink, "Decentralized Data Model," *Chainlink Documentation*. [Online]. Available: https://docs.chain.link/architecture-overview/architecture-decentralized-data-model/
[4] R. Chandramouli and Z. Butcher, *Building Secure Microservices-based Applications Using Service-Mesh Architecture*, NIST SP 800-204A, May 2020. doi: 10.6028/NIST.SP.800-204A
[5] R. Chandramouli, *Security Strategies for Microservices-based Application Systems*, NIST SP 800-204, Aug. 2019. doi: 10.6028/NIST.SP.800-204
[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf [Cited for background on verifiable distributed transaction records, not as a source of the proposed insurance design].