

A Novel Power Data Sharing Platform

Zhang Siwen

GD Power Development Co., Ltd., Beijing, China

Received: May 28, 2026; Revised: July 25, 2026; Accepted: August 21, 2026; Published: September 18, 2026

Abstract

The expansion of advanced metering, distributed energy resources, dispatch operations, asset management, and electricity-market data has exposed the limitations of traditional database-to-database exchange and manually administered authorization. Such conventional solutions cannot simultaneously achieve semantic interoperability, cross-domain least-privilege access, low-latency delivery, privacy minimization, and end-to-end accountability. This paper presents CECF, the Cloud-Edge-Chain Feedback platform for power-data sharing. The IEC Common Information Model provides the semantic foundation. Edge gateways perform tasks such as protocol adaptation, time-stamp correction, data quality filtering, sensitive field masking, stream data buffering and hotspot content caching. The cloud data plane integrates object storage services, time-series databases and event bus components. The authorization framework combines role-based access control, attribute-based security policies, access purpose declaration and real-time context risk evaluation. Large operational payloads remain encrypted off-chain. Only message digests, policy versions, timestamps, authorization decisions and audit evidence indexes are recorded on a permissioned ledger. Using monitoring indicators including P95 latency, queue length, cache-hit rate, failure rate and policy risk value, the feedback controller adjusts evidence micro-batch parameters, cache prefetching strategies and policy thresholds. A differential-privacy interface supports statistical data sharing scenarios. The reported simulations use Poisson arrival flow and multi-stage queueing models. At an offered load of 1200 requests per second, CECF achieves P95 latency of 16.38 ms. Relative to centralized architecture, full on-chain scheme and static-hybrid architecture, its P95 latency drops by 42.1%, 69.5% and 14.6% respectively. When the offered request rate rises to 1800 requests/s, the platform maintains 1802.8 SLO-qualified requests per second. The ablation results indicate that edge caching and closed-loop feedback control are the principal mechanisms that prevent a sharp increase in tail latency. The design provides a feasible route to power-data sharing with high efficiency, verifiability and standardized governance. Its parameters nevertheless require recalibration on a testbed close to production conditions.

Keywords: power data sharing; cloud-edge collaboration; permissioned ledger; attribute-based access control; closed-loop feedback; discrete-event simulation.

1 Introduction

Modern power systems generate massive multi-source data, covering SCADA, energy management systems, distribution automation systems, advanced metering infrastructure, phasor measurement units, protection devices, distributed energy resources, weather information platforms, asset management systems and power market modules. Sampling time intervals span a wide range: some protection measurement data are collected at millisecond level, while monthly settlement records belong to low-frequency data. The data also take diverse forms, including protocol messages, time-series records, power grid topology information, engineering documents and model files.

Continued reliance on point-to-point interfaces for data exchange leads to repeated mapping configuration,

version inconsistency, inconsistent authorization evidence, and fragmented responsibility chains. Therefore, building a power-data sharing platform cannot simply rely on constructing a centralized data lake. Semantic processing, security guarantee, service performance, privacy protection and system governance must be addressed as an integrated system problem.

IEC 61970 puts forward the Common Information Model (CIM), which realizes unified semantic description for power-system objects, their attributes and mutual relationships [1]. IEC 61850 specifies communication standards for automatic functions and equipment of power utilities [2]. Previous research indicates that CIM can reduce integration costs among heterogeneous power business applications [3-4]; edge computing technology can shift selected processing tasks to locations close to measurement sources [5]; permissioned ledgers improve audit accountability when data crosses organizational boundaries [6-7].

Nevertheless, each of these technologies has its own application constraints. If high-frequency raw data are replicated through ledger consensus mechanisms, storage pressure and verification overhead will quickly turn into obvious queueing delay. Purely centralized architectures provide limited independently verifiable audit records. Static caching strategies and fixed-size evidence batches cannot respond effectively to traffic surges, shifting hot-spot access objects or increased authorization risks.

In this paper, power-data sharing is defined as: delivering semantically processed data or statistical outputs to authorized users under the constraints of data classification rules, user identity, declared access purpose and real-time running context, while generating audit evidence for subsequent transaction verification. The definition excludes unauthorized copying of raw payloads, and permissioned ledger is not treated as the primary storage for operational data. Within this scope, this paper makes several contributions: proposing the four-plane CECF system architecture; constructing purpose-aware and risk-driven authorization decision logic combining RBAC and ABAC; designing the off-chain-payload plus on-chain-evidence storage pattern; developing a closed-loop feedback controller for evidence batching, cache prefetching and policy threshold adjustment; completing a Python-based simulation suite, which supports baseline architecture comparison, mechanism ablation experiments and differential-privacy performance evaluation.

2 Sharing Workflow and Design Requirements

2.1 Data scope and processing flow

Five major types of data are processed on this platform. First, operational measurement data come from SCADA, EMS, DMS and PMU devices. These telemetry, phasor, and status variables are updated on millisecond-to-second time scales and primarily support limit monitoring and dispatch analysis. Second, consumption and metering data are collected by AMI terminals and charging facilities. Load curves, event logs and energy statistics with minute-to-hour update intervals support load forecasting and demand-response business. Third, asset data sourced from production management systems, GIS databases and equipment registries record device parameters, topology relations and defect information, updated hourly or daily for power grid planning and equipment maintenance. Fourth, external environmental datasets include weather data, disaster warning messages and geospatial context information with minute-hour update cycles. Fifth, market and business data contain bidding records, contracts, settlement information and market participant files, whose time granularity ranges from minutes to months. Since these data categories differ in latency requirements, integrity requirement, sensitivity level and retention period, preprocessing procedures should be configured for specific tasks instead of following one universal cleaning standard.

A complete data-sharing transaction consists of nine sequential phases: data acquisition, protocol adaptation, quality inspection, semantic mapping, catalog registration, authorization decision, data delivery, evidence submission and runtime feedback. The acquisition phase preserves source identity, collection timestamp and clock quality information. Protocol adaptation converts IEC 61850 messages, event stream records, files and database change logs into unified event envelopes. Semantic mapping maps vendor-specific

fields to CIM objects and stores mapping version information. In the delivery phase, field projection, content masking, data aggregation or differential-privacy processing are applied according to the declared purpose. The evidence module records payload digests and authorization results. The feedback link continuously adjusts runtime parameters without weakening mandatory security constraints.

2.2 Objectives and evaluation dimensions

Six evaluation dimensions are established for platform assessment. Semantic interoperability is measured by mapping coverage ratio and schema conflict frequency. Trusted authorization capability is reflected by interception rate of illegal requests, false rejection rate and the interpretability of policy outputs. Efficient data delivery takes median latency, P95 latency, proportion of requests finished within 200 ms and SLO qualified goodput as indicators. Evidence verifiability covers audit coverage, tamper detection range and evidence completeness. Privacy minimization is evaluated via privacy budget consumption, statistical error magnitude and leakage risk of sensitive fields. Operability and scalability can be judged by queue length, cache hit rate, resource utilization, fault recovery performance and component replaceability without breaking existing data contracts.

3 Related Technologies and Research Gap

3.1 Semantic integration and data spaces

CIM describes power grid resources through object classes, attributes and association relations, offering unified language for EMS, DMS and planning applications [1]. Open data platforms built upon CIM can lower system integration costs for power utilities [3-4]. Even so, CIM itself cannot automatically eliminate inconsistent naming conventions, unit differences, phase information deviations, time-zone conflicts and topology version divergence from original data sources. Manual mapping rules, quality constraints and human review are still indispensable. A qualified data space also needs to maintain catalog entries, data lineage, sensitivity grades, data owners and retention periods, to guarantee data discoverability matches authorized access scope for identical semantic objects.

CIM extensions for integrated electricity, gas, and heat systems show how a common model can be adapted to additional energy domains [14]. Data-space research adds governance and usage control to cross-organizational exchange [22]. Opriel et al. derive data-sovereignty design principles from an automotive supply-chain study [25]. Transferring such principles to utilities requires mapping usage restrictions to utility roles, data classes, and operational deadlines. We use these studies to motivate separating object identity, semantic mapping, and authorization policy in CECF. A shared vocabulary alone does not establish whether a particular user may receive a particular data version.

3.2 Secure sharing and verifiable auditing

Zero-trust architecture emphasizes continuous verification for users, assets and resources, rather than granting trust simply based on network location [8]. ABAC evaluates attributes of subjects, objects, operations and external environments. It fits cross-domain scenarios to express access purposes, time windows and data sensitivity limits [9]. IEC 62351-8 gives specifications for role-based access control in power-system management [10]. Permissioned ledgers such as Hyperledger Fabric maintains consistent audit history among authenticated participants [11]. Some existing smart-grid studies combine on-chain policies and off-chain trusted computation [6]. However, storing full raw data on-chain brings unbearable overhead for high-speed services. Immutability of ledger records cannot guarantee that original measurement data are error-free. Besides, ledger evidence cannot replace source-end validation and key management work.

Recent studies address distinct security requirements in power-data exchange. Faheem et al. propose a lightweight smart-contract framework for secure communication among distributed energy resources [15].

Oksuz combines consortium blockchain, pseudonymous identities, and authentication for consumption-data aggregation and dynamic billing [16]. Ramanan et al. use a blockchain framework to detect coordinated replay attacks [24], while SPDS addresses private data processing and accountable use [6]. We distinguish these objectives from version-bound authorization auditing: CECF links a released object version to its policy and decision evidence, but digest verification alone does not provide the guarantees of a dedicated replay detector or an anonymous aggregation protocol.

3.3 Cloud-edge coordination and runtime feedback

Edge computing realizes protocol conversion, data filtering, content caching and fast response near data sources, which helps reduce wide area network traffic and central side computational burden [5]. Distributed log systems are suitable for persistent and replayable event streams [12]. Most existing practical implementations set cache capacity and ledger batch size relying on offline experience. When object access popularity changes, cross domain network conditions fluctuate or policy risk rises, cache pollution, accumulated queue length and delayed audit evidence will appear. Different from those schemes, CECF treats cloud, edge and ledger modules as a controlled whole system. Real time performance and risk metrics are fed back to adjust bounded system parameters.

Molokomme et al. review edge-intelligence architectures, computation offloading, and cybersecurity for smart grids [23]. For permissioned ledgers, Abang et al. examine latency-modelling approaches and the parameters that influence Hyperledger Fabric latency [17]. Melo et al. use a stochastic Petri net to study how block size, transaction arrival rate, and computing capacity affect Fabric response time [18]. We use these studies to motivate monitoring individual processing stages and calibrating the controller against both workload and resource limits. A larger batch can amortize ledger work while increasing the wait for batch formation; therefore, its benefit must be evaluated together with cache behavior and evidence-confirmation delay.

Several representative schemes expose existing research gaps. Centralized database plus API architecture works well under low load, yet it centralizes trust risks and provides insufficient cross domain audit evidence. Full on chain solutions generate complete ledger trails, but consensus and data replication create excessive overhead. Static hybrid patterns store payloads off chain and anchor digests on chain, while its fixed cache strategy, evidence batch size and thresholds cannot adapt to dynamic conditions. CECF introduces semantic policy separation mechanism, edge side caching, digest anchoring and closed loop feedback control. Its main limitation lies in parameter calibration, and further verification under realistic testbeds is required.

3.4 Comparison and positioning

Table 1 summarizes the roles of the selected research lines in the proposed platform. The comparison concerns design responsibilities and remaining integration work. The synthetic baselines in Section 6 are simplified architectural alternatives and are not implementations of the cited systems, so their numerical results should not be read as a direct ranking of those publications.

Table 1. Related research and its role in CECF

Research line	Main contribution	Remaining integration need	References
CIM integration	Common object models and field mappings	Connect mapping versions to authorization and delivery	[3, 4, 14]
Data spaces and sovereignty	Data sovereignty and usage control	Define utility-specific policies and service targets	[22, 25]
Power-system security	Secure communication, private aggregation, and replay detection	Distinguish audit evidence from source and attack validation	[6, 15, 16, 24]
Permissioned-ledger	Latency modelling and resource capacity planning	Measure application latency and evidence delay together	[11, 17, 18]

optimization			
Edge computing and intelligence	Local processing, offloading, and security	Coordinate caching, freshness, policy changes, and limited resources	[5, 23]
CECF	CIM, purpose-aware access, off-chain data, and adaptive evidence handling	Calibrate parameters and validate security on a deployed testbed	This work

4 CECF Platform Design

CECF follows three core design principles: business related payloads are transmitted and stored off chain; trust related audit evidence is submitted onto the permissioned ledger; semantic modules and policy services are coordinated through stable object identifiers. As shown in Figure 1, heterogeneous data sources are placed on the left hand side; edge access layer and trusted data space occupy the middle part; dispatch analysis, grid planning, market services and regulatory audit act as downstream consumers on the right. The trusted data space consists of data plane, semantic plane, policy plane and evidence plane. Runtime monitoring results are delivered to edge components and policy services through the feedback controller.

Power data sharing platform: cloud-edge-chain collaborative architecture

Bulk data remains off-chain; policies, digests and audit evidence form the trusted control plane.

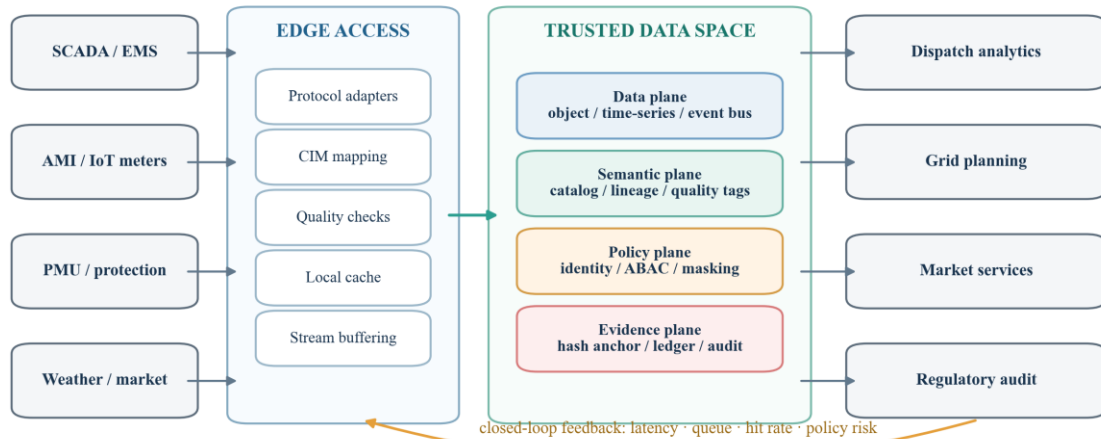


Figure 1. Cloud-edge-chain collaborative architecture of the power-data sharing platform

4.1 Edge access

Edge gateways are deployed according to substation scope, feeder regions or business domains. They execute protocol adaptation, timestamp calibration, preliminary schema validation, sensitive field masking, hotspot caching, stream buffering and interrupted transmission recovery. Protection control messages stay inside original real time security zones. The sharing platform can only obtain authorized copies or mirrored event data, so non real time analysis services will not intrude into protection control loops. Cache keys are constructed combining CIM object identifier, time window and data version. Every cached record binds with corresponding policy version tags. Once relevant policies get revoked or modified, previously cached authorized content will turn invalid immediately.

4.2 Four-plane trusted data space

The data plane stores object files, time-series records and event logs. Data are partitioned according to business

domain, sensitivity classification and retention period. The semantic plane maintains CIM models, mapping rules, data lineage information, resource catalogs and quality labels. The policy plane manages identity information, roles, attribute sets, access purposes, masking templates and risk thresholds. The evidence plane generates payload digests, policy digests, authorization outputs, timestamps and ledger transaction indexes. Stable object identifiers connect four planes together. Changes for data content, semantic definitions and security policies can be audited independently, and each plane can scale and iterate separately.

4.3 Semantic mapping and quality control

Suppose set S represents source end fields, set C stands for target CIM attributes, and mapping function $m: S \rightarrow CU\{\text{bottom}\}$ describes field conversion relationships. CECF first applies deterministic rules based on source system information, table names, field labels and physical units. Historical mapping records or optional large model assistant can offer candidate mapping results. Schema constraints will check data type, dimension, enumeration items and topology relations in the final step. Automatically generated mapping proposals can go online only after deterministic validation plus human confirmation. Large language models merely serve as candidate suggestion generators. They are forbidden to directly alter formal production models or deduce grid facts that do not exist in source evidence.

$$Q(x) = w_1 q_{\text{complete}} + w_2 q_{\text{range}} + w_3 q_{\text{time}} + w_4 q_{\text{topology}}, \quad \sum_{i=1}^4 w_i = 1 \quad (1)$$

In formula (1), x means the data object under quality evaluation; $Q(x)$ is its composite quality score. q_{complete} , q_{range} , q_{time} and q_{topology} correspond to sub scores of completeness, range validity, temporal consistency and topology consistency respectively. w_1 to w_4 are non negative weight coefficients, and their sum is normalized to 1. Subscript i ranges from 1 to 4 for four groups of sub score weight pairs.

4.4 Purpose- and risk-aware authorization

RBAC defines basic duty boundaries at first. After that, ABAC adds constraints related to subjects, objects, operations, declared purposes and runtime environment. For a request tuple $r = \langle s, o, a, p, e \rangle$, s is requesting subject, o is target data object, a denotes operation type, p represents declared purpose, and e includes device status, time, region and session risk. Authorization judgement is given as follows:

$$\text{Permit}(r) = \text{RBAC}(s, o, a) \wedge \text{ABAC}(s, o, a, p, e) \wedge [\text{Risk}(r) < \tau(o)] \quad (2)$$

In Equation (2), r is the request tuple; s, o, a, p, e represent requesting subject, target object, requested action, declared purpose and environmental context. RBAC and ABAC output boolean results from role based and attribute based rules. $\text{Risk}(r)$ is contextual risk score; $\tau(o)$ is maximum acceptable risk threshold for object o . Symbol $\wedge$ means all conditions must hold true. $\text{Permit}(r)$ is the final authorization decision.

Policy engine outputs three states: Permit, Deny and Degrade. Degrade mode reduces data sharing granularity. For instance, customer level minute scale data can be replaced by hourly regional aggregation results, or precise equipment location information gets removed. Each decision saves policy version, digest of input attributes and transformation records. Sensitive attributes will not be written into ledger in plaintext. This continuous context aware authorization mechanism complies with zero trust and ABAC principles [8, 9].

4.5 Off-chain payloads and on-chain evidence

Original shared payloads are kept encrypted inside the data plane. Ledger only stores evidence tuples that satisfy integrity verification and authorization audit requirements. Digest calculation binds object identifier, version number and canonical encoding format. In this way, different serialization outputs of one identical semantic object will not produce incomparable hash values. Within one micro batch, digests of each record act as leaf nodes of a Merkle tree, and a single ledger transaction anchors the Merkle root.

$$E = \langle \text{object_id}, \text{version}, H(\text{payload}), H(\text{policy}), \text{decision}, \text{time}, \text{nonce} \rangle \quad (3)$$

$$\text{root}_k = \text{MerkleRoot}(H(\text{record}_1), \dots, H(\text{record}_n)); \quad \text{Verify} = [\text{root}'_k = \text{root}_k] \quad (4)$$

In (3) (4), E denotes evidence tuple. *object_id* and *version* uniquely mark the status of shared object. $H(\cdot)$ is cryptographic hash function. *payload* and *policy* mean canonicalized data content and corresponding authorization policy. *decision* is authorization result; *time* is submission timestamp; *nonce* is one time value for distinguishing repeated submissions. For micro batch k , record_i is its i th evidence entry, n is total record count. root_k is anchored Merkle root, root'_k is recalculated root during verification. $\text{MerkleRoot}(\cdot)$ constructs tree root, and Verify is boolean comparison result between recalculated value and anchored root.

Matching digests only prove that checked payload is consistent with anchored version. They cannot guarantee measurement sources are fault free or free from tampering. Source signature, trusted timestamp, quality checks, secure hardware and key rotation are still compulsory safeguards. Ledger nodes should be managed by different responsibility domains, and membership certificates as well as consensus parameters need operational governance constraints.

The consensus fault model must also be stated explicitly. Practical Byzantine fault tolerance [7] is relevant when replicas may behave maliciously, whereas the guarantees of a deployed permissioned ledger depend on its selected ordering protocol and membership assumptions [11]. CECF therefore requires the deployment specification to identify the tolerated faults, quorum assumptions, and evidence-retention responsibilities. A permissioned membership list is not, by itself, a Byzantine-fault-tolerance guarantee.

4.6 Closed-loop feedback control

Figure 2 illustrates a complete sharing transaction and its feedback loop. After identity verification and purpose declaration, policy judgement and CIM semantic resolution are performed. Data delivery is realized through cache hit, database query, field projection or privacy aware transformation. Afterwards, evidence will be committed to ledger. Within sliding time windows, runtime monitor collects P95 latency L_{95} , queue length q , cache hit rate h , failure rate f and policy risk ρ . The feedback controller adjusts evidence micro batch size B , hotspot threshold θ_c and authorization risk threshold τ .

Trusted sharing transaction and evidence feedback loop

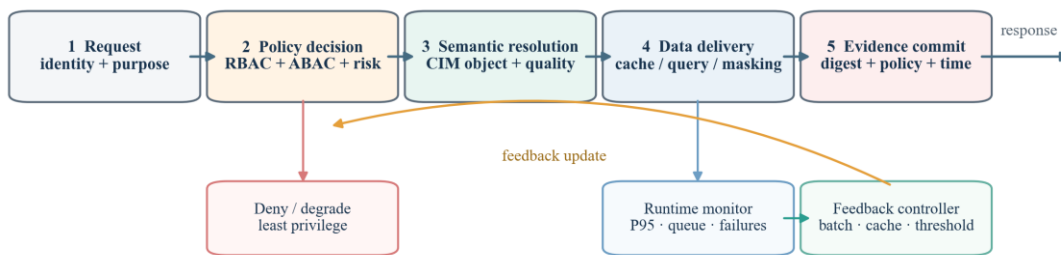


Figure 2. Trusted sharing transaction and evidence feedback loop

$$B(t+1) = \text{clip}(B(t) + k_1(L^* - L_{95}) + k_2(q^* - q), B_{\min}, B_{\max}) \quad (5)$$

$$\theta_c(t+1) = \text{clip}(\theta_c(t) + k_3(h^* - h)); \quad \tau(t+1) = \text{clip}(\tau(t) - k_4(\rho - \rho^*)) \quad (6)$$

In formula (5) (6), t stands for control interval. $B(t)$ is current evidence micro batch size, bounded by $B_{\min}$ and $B_{\max}$. $\text{clip}(\cdot)$ limits variables within legal range. k_1 k_4 are controller gain coefficients. L_{95} is measured P95 latency and L^* is its target value. q is real time queue length and q^* is target queue length. θ_c

represents hotspot judgement threshold for cache. h is measured cache hit rate and h^* is expected hit rate. τ is authorization risk threshold; ρ and ρ^* are measured and target contextual risk level. Asterisk marks target parameters, and $t + 1$ means next control interval.

When tail latency or queue accumulation rises, controller expands evidence micro batch size and pre fetches frequently accessed objects. If contextual risk exceeds target value, authorization rules will become stricter or output data get degraded. Moving average filtering, dead zone mechanism and rate limiting strategies avoid system oscillation. Mandatory security rules enjoy higher priority than performance optimization. The controller can only tune whitelisted parameters and cannot skip necessary rejection rules just for latency reduction.

A deployment should measure response latency and evidence-confirmation delay separately. Increasing a micro-batch can amortize ledger work, but it can also increase the wait for a batch to close [17, 18]. The controller therefore needs a maximum evidence-wait bound in addition to batch-size limits. Cache prefetching must remain subject to the current policy version, and a change in a risk threshold must not override an explicit denial. These constraints define implementation requirements whose effectiveness still needs to be measured.

4.7 Privacy-preserving statistical interface

For residential load analysis or charging behavior research, CECF prefers aggregated statistical results instead of raw user level detailed records. If statistical query f has global sensitivity Δf , Laplace mechanism outputs processed result as:

$$M(D) = f(D) + \text{Laplace}\left(\frac{\Delta f}{\epsilon}\right) \quad (7)$$

In (7), D is original dataset; $f(D)$ represents exact query result. Δf is global sensitivity of function f , which equals the maximum change of query output caused by adding or removing one protected record. ϵ is privacy budget. $\text{Laplace}\left(\frac{\Delta f}{\epsilon}\right)$ denotes zero mean Laplace noise with scale $\frac{\Delta f}{\epsilon}$. $M(D)$ is differentially private output. Smaller ϵ brings stronger single query privacy protection yet increases statistical error. The platform maintains privacy budget ledger grouped by dataset and usage purpose, recording query content, consumed budget and output value range. Emergency dispatch services that cannot tolerate noise adopt independent authorization and isolated computing modules instead of general differential privacy processing.

Shaham et al. develop STPT for differentially private publication of electricity time series, exploiting spatiotemporal patterns to improve utility [19]. CECF uses a Laplace interface [13], whose guarantees require a defined neighboring-dataset relation, bounded contributions, and accounting for repeated releases [20]. For a single time-slot sum with one protected contribution bounded to 0–5 kW, add-or-remove sensitivity is at most 5 kW. Protecting a whole 96-slot household trajectory requires a trajectory-level bound or composition across releases. Separate purpose-specific budget records do not reset the privacy loss of the same individuals. The protected unit must therefore be specified before interpreting ϵ . STPT provides a recent comparison point for time-series utility; the present simulation does not benchmark against its implementation.

5 Implementation Blueprint

5.1 Component deployment

Reference deployment scheme contains five sets of components: edge gateways deployed in substations or business domains; persistent event bus supporting data persistence and streaming subscription; object storage, time series database and metadata catalog; identity service, API gateway, policy decision points and policy enforcement points; permissioned ledger maintained by multiple responsible departments. All components communicate through mutually authenticated channels, and key services adopt multi copy replication across

different failure domains. Kafka style logs can build replayable event streams [12]; Fabric type permissioned ledgers can realize evidence anchoring functions [11]. This architecture is not bound to specific commercial products.

5.2 API and data contract

The sharing API provides catalog search, object query, time window subscription, statistical query, evidence verification and authorization revocation functions. Request messages contain `request\_id` and `trace\_id` to guarantee idempotence and end to end tracing. Subject token, declared usage purpose and tenant information are used for identity authentication and accountability. CIM identifier, field projection rules, time range and version information support precise data selection. Constraints like minimum data quality, response deadline and delivery mode are also included. Response returns data version, quality tags, `decision\_id`, policy version, masking operation records, payload digest and ledger transaction index. Asynchronous subscription interfaces apply idempotent event identifiers and consumer offset to prevent duplicate processing after network reconnection.

5.3 Security and operational boundaries

Production-environment deployment needs data classification, least-privilege configuration, managed key systems, certificate rotation, API flow-limiting, sensitive-field masking, malicious-request isolation and audit alarm mechanisms. Ledger nodes and payload storage should run in separate failure domains, so one administrator cannot take control over both business data and all audit evidence. Monitoring modules need to distinguish legitimate business rejection from resource exhaustion or policy-engine faults. Automatic feedback adjustment is limited to pre-approved parameters. Modification of data classification standards, mandatory security rules or core policies must pass manual review. The platform should never be connected to hard real-time protection loops. Cache entries must bind with policy versions. Data and audit evidence should define clear owner and retention period. Before formal release, capacity testing, fault injection, attack simulation, privacy assessment and audit verification need to be completed on realistic test platforms.

These deployment boundaries are consistent with NIST guidance on operational technology, which treats performance, reliability, and safety as central security requirements [21]. For CECF, tests should cover the effect of certificate expiry, policy-service unavailability, ledger interruption, and cache revocation on both service continuity and evidence completeness. The present simulation does not establish compliance with an OT security standard.

6 Simulation Validation

6.1 Purpose, workload, and boundaries

A Python discrete event simulator is adopted to observe performance changes under different request loads. Requests arrive following Poisson distribution. Each processing stage is modeled as multi server first come first served queue, and service time obeys Gamma distribution with given mean value and coefficient of variation. Payload size follows truncated lognormal distribution ranging from 0.02 MB to 6.0 MB, simulating mixed small telemetry records and occasional large-size batches. The workload consists of 70 % read requests and 30 % write requests, among which 14 % are marked as critical requests. Offered request rate increases from 200 requests/s to 1800 requests/s with step of 200. Every test case runs for 30 s and repeats eight times independently under random seed 20260822. All compared architectures receive identical request stream in each repeated trial. These experiments focus on mechanism comparison, rather than benchmarking for specific blockchain, database or real power grid.

Four schemes are set for comparison. Centralized sharing adopts central database without edge cache, and applies sampling synchronous audit. Full on chain sharing transmits original payloads to permissioned ledger

for consensus and replication. Static hybrid architecture stores payloads off chain and anchors digests on chain, with fixed cache hit probability of 0.35 and static batching strategy. The proposed CECF sets nominal cache hit probability to 0.62, equipped with adaptive digest micro batch and feedback control. Service level latency objective is 200 ms. For integrity experiments, 500 modified samples are injected in each run. Centralized sharing carries out verification with sampling probability equal to 0.32. Schemes based on digest anchoring verify every accessed data version. Privacy related experiment aggregates records of 100 users within 96 time slots, clamps load contribution within 0–5 kW, changes ϵ from 0.1 to 4.0, and repeats each parameter setting for 300 times.

Table 2 consolidates the workload and configuration values used in the reported experiments. Cache-hit probabilities are model inputs, so the comparison combines architectural choices with these configured assumptions.

Table 2. Reported simulation settings

Item	Reported setting
Arrival and offered load	Poisson arrivals; 200–1800 requests/s in steps of 200
Processing stages	Multiple-server FCFS queues; Gamma service times
Payload distribution	Truncated lognormal; 0.02–6.0 MB
Request composition	70% reads; 30% writes; 14% critical requests
Duration and repetition	30 s per trial; 8 repeats; reported seed 20260822; identical streams across architectures within a repeat
Service target	Response latency within 200 ms
Conditional cache-hit inputs	Static hybrid 0.35; CECF nominal 0.62
Integrity experiment	500 modified samples per run; centralized audit sampling probability 0.32
Privacy workload	100 users; 96 time slots; load contribution clipped to 0–5 kW
Privacy settings	$\epsilon = 0.1, 0.25, 0.5, 1, 2, 4$; 300 repeats per setting

Full reproduction additionally requires the server count and service-time parameters at each stage, the underlying lognormal parameters, batching rules, controller gains and bounds, cache update logic, random-stream generation, and treatment of requests still queued after 30 s. These details are not specified in the current experiment description, which limits independent reproduction.

6.2 Metrics

Response performance is evaluated through median and P95 end to end latency. Goodput counts how many requests finish within 200 ms per unit time. SLO success ratio is the percentage of all requests satisfying latency constraint. Integrity coverage represents the proportion of injected tampered samples detected in simulation. Differential privacy utility is quantified by normalized root mean square error (NRMSE) of aggregated load results.

$$\text{Goodput} = \frac{N(\text{latency} \leq 200 \text{ ms})}{T}; \quad \text{SLO} = \frac{N(\text{latency} \leq 200 \text{ ms})}{N} \times 100\% \quad (8)$$

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}}{\text{mean}(y)} \times 100\% \quad (9)$$

In Equation (8), $N(A)$ counts requests satisfying condition A; N is total submitted request quantity; T is observation duration in seconds. Latency means end to end response delay, 200 ms is SLO threshold. Goodput unit is SLO qualified requests per second. SLO stands for percentage of requests that meet latency requirements. In Equation (9), n is evaluation sample quantity; i is sample index. y_i is real aggregated load value; $\hat{y}_i$ is estimated result after privacy processing. $\text{mean}(y)$ is average value of real load, and NRMSE is normalized root mean square error expressed in percentage form.

6.3 Load-performance results

At 1200 requests/s nominal load, all four architectures satisfy 200 ms latency target and deliver approximately 1201.5 qualified requests per second. But their P95 latency differs greatly: centralized sharing reaches 28.28 ms, full on-chain scheme gets 53.72 ms, static hybrid obtains 19.17 ms, and CECF achieves 16.38 ms. Relative to the other three alternatives, CECF cuts P95 latency by 42.1 %, 69.5 % and 14.6 % respectively. Its unconditional cache-hit ratio is 43.41 %. This value is lower than read-oriented conditional hit rate because write requests are counted in total denominator. Centralized sampling audit detects about 32.7 % of injected tampered samples, matching its 0.32 sampling probability. Under ideal assumptions including complete evidence access and zero hash collision, every digest-anchoring scheme can detect all modifications for verified versions. It should be noted that this result reflects theoretical verification coverage instead of real-world cyber-attack detection probability.

Table 3 lists the P95 values at the common offered load of 1200 requests/s. It makes the reported latency reduction directly comparable without extracting values from the performance curves.

Table 3. P95 latency at 1200 requests/s

Architecture	P95 latency (ms)	CECF reduction relative to this architecture
Centralized sharing	28.28	42.1%
Full on-chain sharing	53.72	69.5%
Static hybrid	19.17	14.6%
CECF	16.38	Reference

Each reduction is calculated as the difference between the comparison architecture and CECF, divided by the latency of that comparison architecture. The values describe the stated synthetic workload and do not establish performance on a physical power-system network.

Performance comparison under identical synthetic workloads (mean and 95% CI, n=8)

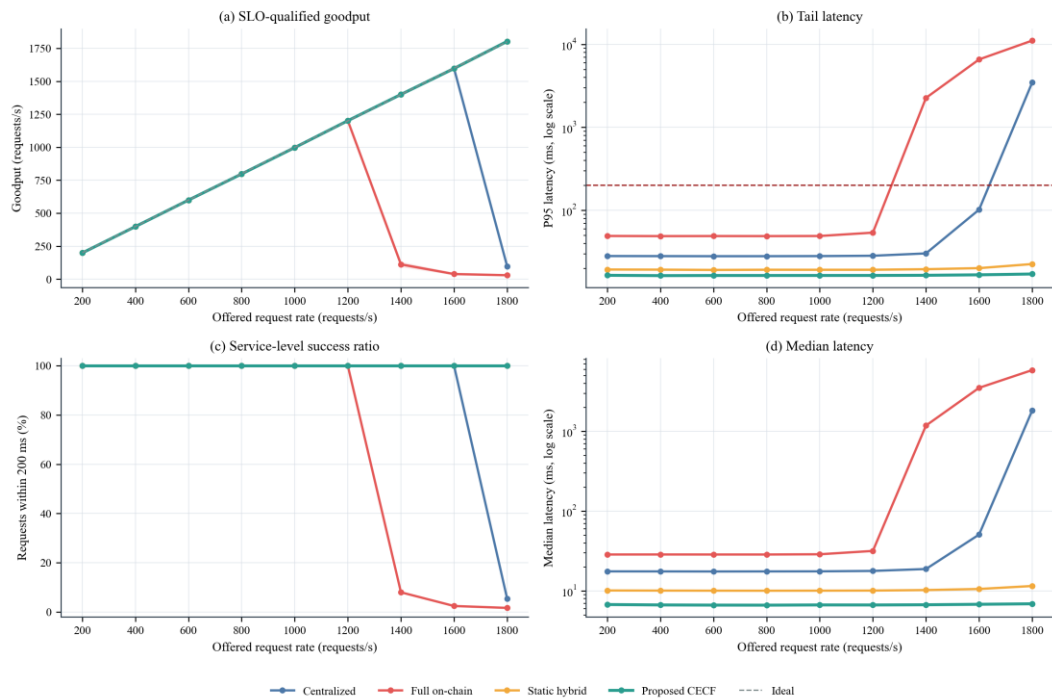


Figure 3. Performance under identical synthetic workloads; curves show means and 95% confidence intervals over eight repeats

As shown in Figure 3, full on chain architecture keeps latency below 200 ms roughly up to 1200 requests/s. After that, consensus resources get saturated and queue length rises sharply. Centralized solution meets bottleneck later. When offered rate reaches 1800 requests/s, its P95 latency jumps to 3462.4 ms and qualified goodput drops to only 97.7 requests/s. Under the same load, CECF maintains P95 latency of 17.05 ms and qualified goodput of 1802.8 requests/s, which is around 18.4 times and 60.5 times of centralized and full on chain schemes respectively. Static hybrid architecture stays within SLO constraint in simulated load scope, while its tail latency increases gradually. Storing payloads off chain removes major consensus bottleneck, and feedback control further enhances stability under high load conditions.

6.4 Ablation and privacy utility

At request rate of 1600 requests/s, complete CECF realizes 1597.6 qualified requests/s with P95 latency 16.63 ms. After disabling feedback module, goodput barely changes, but P95 latency rises to 20.02 ms, increased by 20.4 %. Removing edge cache makes qualified goodput fall to 168.0 requests/s, P95 latency soars to 1809.74 ms, and SLO success ratio drops to 10.52 %. If payloads are stored on chain, goodput further decreases to 40.0 requests/s, P95 latency reaches 6562.77 ms and SLO success ratio is only 2.50 %. These ablation outcomes clarify the function of each module: edge cache reduces repeated read pressure for central services; feedback control suppresses tail latency growth without modifying hard security rules; off chain payload plus on chain evidence separation lays the foundation for system scalability.

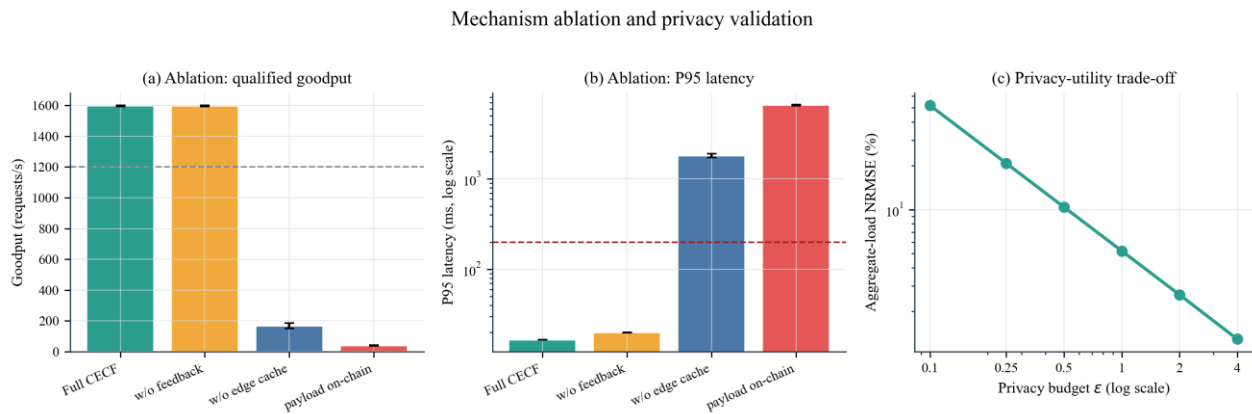


Figure 4. Mechanism ablation and privacy-utility validation

Differential privacy experiment presents monotonic trade off between privacy protection and data utility. Mean NRMSE equals 51.896 % when $\epsilon=0.1$, 20.805 % for $\epsilon=0.25$, 10.417 % for $\epsilon=0.5$, 5.214 % for $\epsilon=1$, 2.613 % for $\epsilon=2$, and 1.297 % for $\epsilon=4$. Corresponding half width of 95 % confidence interval are ± 0.640 , ± 0.277 , ± 0.136 , ± 0.065 , ± 0.032 , ± 0.016 percentage points. No fixed universal ϵ value can fit all scenes. Suitable privacy budget is determined by data sensitivity, aggregation population, declared access purpose, output range and historical cumulative budget consumption.

6.5 Cache hit sensitivity

Figure 5 varies the conditional read-cache hit probability while keeping the service distributions and resources unchanged. At each offered rate, all probability settings receive the same arrivals and per-request service samples within a repetition. The experiment isolates the effect of h . It does not compare separately implemented CECF and static-hybrid controllers.

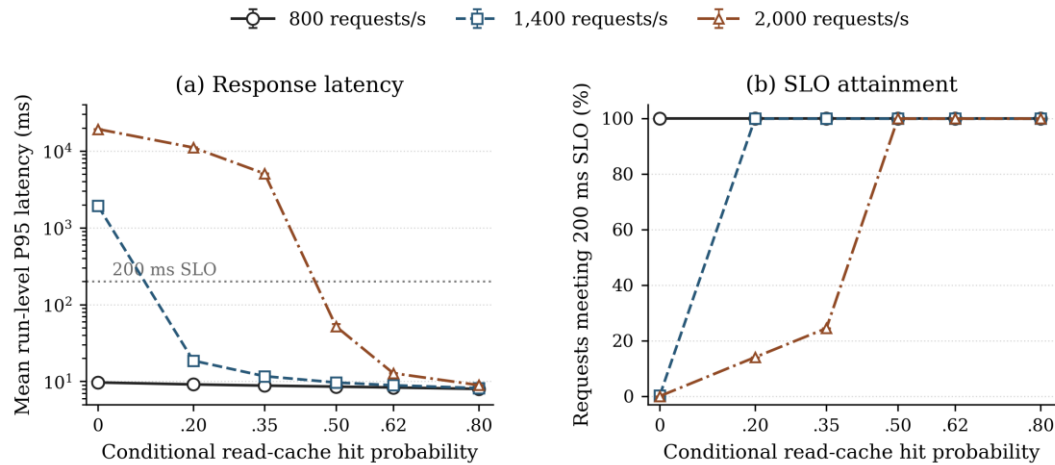


Figure 5. Independent cache sensitivity experiment. (a) Mean run-level P95 response latency on a logarithmic scale. (b) Fraction meeting the 200 ms response SLO. Points show means over 30 runs with 95% confidence intervals, which can be smaller than the markers. Overloaded conditions are finite-horizon results

At 2,000 requests/s, increasing h from 0.35 to 0.62 reduces the mean run-level P95 from $5,068.79 \pm 63.06$ ms to 12.77 ± 0.14 ms. The observed SLO fraction increases from 24.47% to 100.00% over the stated window. At $h = 0.50$, the P95 is 51.83 ± 4.33 ms. Here and below, $\pm$ denotes the half-width of the 95% confidence interval for the across-run mean.

This transition follows the assumed storage capacity. Four servers with a 3 ms mean service time provide a nominal capacity of about 1,333 requests/s. The expected storage arrival rate is the offered rate multiplied by $(1 - 0.7h)$. At 2,000 requests/s, its nominal load falls below capacity only when h exceeds approximately 0.476. The resulting change in queue buildup explains why a moderate change in cache assumptions can produce a large latency difference. This capacity calculation does not guarantee the 200 ms SLO near saturation.

The experiment therefore identifies cache-hit probability as a material sensitivity in the original comparison. A causal estimate of the feedback controller's benefit still requires a shared implementation in which h , resources and workload are held equal while feedback alone is enabled or disabled. The present results should not be used to revise the original CECF latency values.

6.6 Evidence batching tradeoff

Figure 6 varies the maximum batch size B with a fixed 10 ms closing timeout. Confirmation delay includes record accumulation, waiting behind earlier batches and ledger service. Ledger occupancy is the fraction of the $[10, 40)$ s window spent serving batches. A batch can contain fewer than B records when the timer expires. The service model represents fixed per-batch overhead plus a per-record cost.

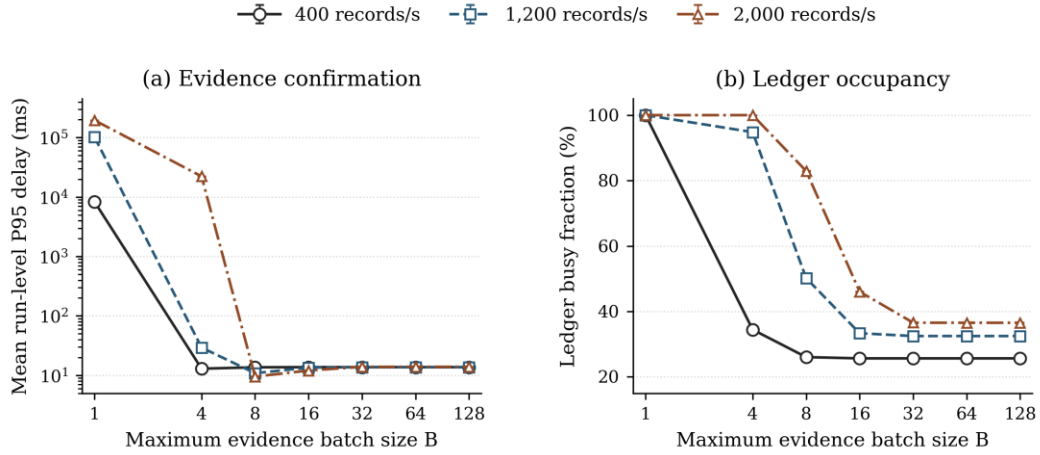


Figure 6. Independent evidence batching experiment. (a) Mean run-level P95 record-to-confirmation delay on a logarithmic scale. (b) Ledger busy fraction. Error bars are 95% confidence intervals over 30 runs. Arrivals stop at 40 s and all pending batches drain. $B = 1$ and other overloaded settings are transient results

At 1,200 records/s, increasing B from 4 to 8 lowers the P95 confirmation delay from 29.30 ± 2.31 ms to 10.86 ± 0.02 ms, while mean ledger occupancy falls from 94.81% to 50.19%. Increasing B further to 32 reduces occupancy to 32.47% but raises the P95 to 13.69 ± 0.01 ms. Thus, lower resource occupancy does not necessarily imply faster confirmation.

At this rate, a maximum batch size of 32 yields only 12.98 records per batch on average because of the timeout. The curves therefore plateau as B increases further. With $B = 1$, the mean 3.04 ms service time limits nominal capacity to about 329 records/s, below every tested rate, so backlog grows throughout the arrival period. Its large latency values reflect this deliberately included overload case.

These results motivate selecting batch size jointly with a confirmation deadline and ledger capacity. They do not establish an optimal adaptive rule or validate the gain signs in Equation (5). A deployment that requires confirmed evidence before returning an application response must include this delay in its end-to-end SLO. The isolated response results in Figure 5 do not include that requirement.

6.7 Burst load and queue recovery

The burst experiment raises the offered rate from 1,200 to 3,000 requests/s for five seconds and then returns it to 1,200 requests/s. In Figure 7, the burst occupies seconds 10–15 after warm-up. Latency is grouped by arrival time in one-second bins and reported at each bin's right edge. Queue counts include waiting storage requests only. Recovery is the first time after the burst ends when this count remains at most 100 for two seconds at 0.1 s sampling intervals.

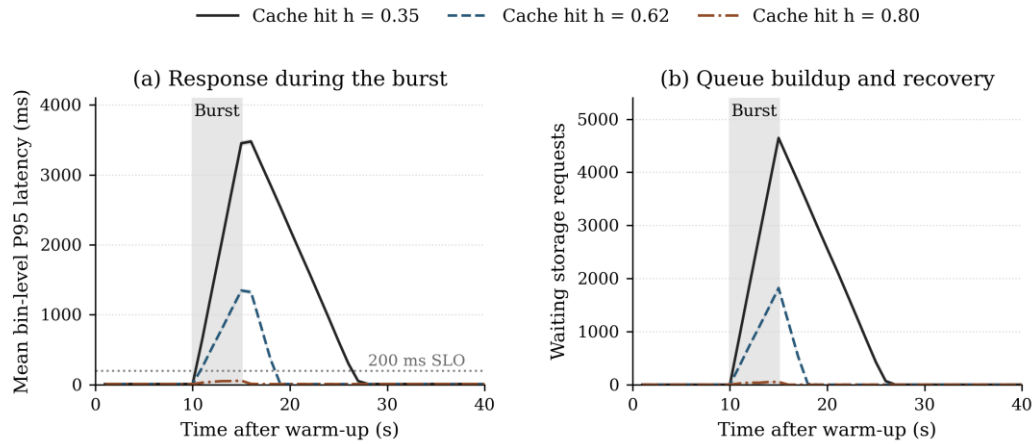


Figure 7. Independent burst-load experiment with fixed cache probabilities. (a) P95 response latency for successive arrival bins. (b) Waiting storage requests at bin ends. Lines show across-run means and shaded bands show pointwise 95% confidence intervals over 30 runs. Gray shading marks the five-second burst

With $h = 0.35$, the mean peak sampled queue reaches 4,646 waiting requests and recovery takes 10.82 ± 0.15 s after the burst ends. With $h = 0.62$, these values are 1,820 requests and 2.72 ± 0.08 s. However, the latter setting still has a mean P95 of $1,247.81 \pm 28.31$ ms among requests arriving during the burst, and only 51.78% meet the 200 ms SLO.

At $h = 0.80$, the mean burst-cohort P95 is 57.66 ± 9.39 ms and all sampled burst requests meet the SLO in these runs. This is an observed finite-sample outcome. During the burst, the expected storage loads are approximately 2,265, 1,698 and 1,320 requests/s for $h = 0.35$, 0.62 and 0.80, respectively. Only the last is below the assumed 1,333 requests/s storage capacity, explaining its much smaller backlog.

The experiment shows that post-burst recovery and latency compliance during the burst are different objectives. It uses fixed routing probabilities, unlimited queues and no fault or backpressure model. Its findings support further capacity and admission-control studies, while validation of adaptive recovery, cache consistency and ledger outages requires additional implementation-specific experiments.

7 Conclusion

This paper designs CECF platform for heterogeneous cross-organizational power-data sharing with strong audit capability. CIM provides unified semantic interpretation standard. RBAC, ABAC, access-purpose declaration and risk evaluation work together to finish authorization decision. Off-chain payload storage cooperating with on-chain digest anchoring balances running performance and verifiability. Runtime feedback controller adjusts evidence batching strategy, cache configuration and bounded policy thresholds. The reported discrete-event simulation indicates that CECF achieves low tail latency and high SLO-qualified goodput under simulated heavy-traffic conditions. Ablation experiments distinguish the contribution brought by edge caching and feedback control. Privacy-related tests quantify utility loss caused by enhanced statistical privacy protection. For future research directions, real world power system datasets can be adopted for testing. Further work also includes fault and attack injection experiments, cross domain ledger deployment practice, formal policy verification and lightweight edge side implementation development.

References

- [1] IEC. IEC 61970-301:2020+AMD1:2022, Energy management system application program interface (EMS-

- API) - Part 301: Common information model (CIM) base. Geneva: International Electrotechnical Commission, 2022.
- [2] IEC. IEC 61850-5:2013+AMD1:2022, Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models. Geneva: IEC, 2022.
 - [3] Anderson A A, Stephan E G, McDermott T E. Enabling Data Exchange and Data Integration with the Common Information Model. PNNL-32679, 2022.
 - [4] Mukherjee M, Lee E, Bose A, et al. A CIM Based Data Integration Framework for Distribution Utilities. 2020 IEEE Power & Energy Society General Meeting, 2020: 1-5. <https://doi.org/10.1109/PESGM41954.2020.9281658>
 - [5] Feng C, Wang Y, Chen Q, et al. Smart grid encounters edge computing: opportunities and applications. *Advances in Applied Energy*, 2021, 1: 100006. <https://doi.org/10.1016/j.adapen.2020.100006>
 - [6] Wang Y, Su Z, Zhang N, et al. SPDS: A Secure and Auditable Private Data Sharing Scheme for Smart Grid Based on Blockchain. *IEEE Transactions on Industrial Informatics*, 2021, 17(11): 7688-7699. <https://doi.org/10.1109/TII.2020.3040171>
 - [7] Castro M, Liskov B. Practical Byzantine Fault Tolerance. *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 1999: 173-186.
 - [8] Rose S, Borchert O, Mitchell S, et al. Zero Trust Architecture. NIST SP 800-207, 2020. <https://doi.org/10.6028/NIST.SP.800-207>
 - [9] Hu V C, Ferraiolo D, Kuhn R, et al. Guide to Attribute Based Access Control Definition and Considerations. NIST SP 800-162, 2014, updated 2 August 2019. <https://doi.org/10.6028/NIST.SP.800-162>
 - [10] IEC. IEC 62351-8:2026, Power systems management and associated information exchange - Data and communications security - Part 8: Role-based access control for power system management. Geneva: IEC, 2026.
 - [11] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceedings of EuroSys 2018*, 2018: 1-15.
 - [12] Kreps J, Narkhede N, Rao J. Kafka: a Distributed Messaging System for Log Processing. *Proceedings of NetDB 2011*, 2011: 1-7.
 - [13] Dwork C. Differential Privacy. *Proceedings of ICALP 2006*, LNCS 4052, 2006: 1-12.
 - [14] Wang S, Meng Z, Yuan S. IEC 61970 standard based common information model extension of electricity-gas-heat integrated energy system. *International Journal of Electrical Power & Energy Systems*, 2020, 118: 105846.
 - [15] Faheem M, Kuusniemi H, Eltahawy B, et al. A lightweight smart contracts framework for blockchain-based secure communication in smart grid applications. *IET Generation, Transmission & Distribution*, 2024, 18(3): 625-638. <https://doi.org/10.1049/gtd2.13103>
 - [16] [Oksuz O. Consortium blockchain based secure and efficient data aggregation and dynamic billing system in smart grid. *Peer-to-Peer Networking and Applications*, 2024, 17: 2717-2736. <https://doi.org/10.1007/s12083-024-01709-8>
 - [17] Abang J E, Takruri H, Al-Zaidi R, Al-Khalidi M. Latency performance modelling in hyperledger fabric blockchain: Challenges and directions with an IoT perspective. *Internet of Things*, 2024, 26: 101217. <https://doi.org/10.1016/j.iot.2024.101217>
 - [18] Melo C, Gonçalves G, Silva F A, Soares A. A comprehensive hyperledger fabric performance evaluation based on resources capacity planning. *Cluster Computing*, 2024, 27: 12395-12410. <https://doi.org/10.1007/s10586-024-04591-4>
 - [19] Shaham S, Ghinita G, Krishnamachari B, Shahabi C. Differentially Private Publication of Smart Electricity Grid Data. *Proceedings of the 28th International Conference on Extending Database Technology (EDBT)*, 2025: 213-225. <https://doi.org/10.48786/EDBT.2025.17>
 - [20] Near J P, Darais D, Lefkovitz N, Howarth G. Guidelines for Evaluating Differential Privacy Guarantees. NIST SP 800-226, 2025. <https://doi.org/10.6028/NIST.SP.800-226>
 - [21] Stouffer K, Pease M, Tang C Y, et al. Guide to Operational Technology (OT) Security. NIST SP 800-82 Rev. 3, 2023. <https://doi.org/10.6028/NIST.SP.800-82r3>
 - [22] Otto B, ten Hompel M, Wrobel S, eds. *Designing Data Spaces: The Ecosystem Approach to Competitive Advantage*. Cham: Springer, 2022. <https://doi.org/10.1007/978-3-030-93975-5>
 - [23] Molokomme D N, Onumanyi A J, Abu-Mahfouz A M. Edge Intelligence in Smart Grids: A Survey on Architectures, Offloading Models, Cyber Security Measures, and Challenges. *Journal of Sensor and Actuator Networks*, 2022, 11(3): 47. <https://doi.org/10.3390/jsan11030047>
 - [24] Ramanan P, Li D, Gebrael N. Blockchain-Based Decentralized Replay Attack Detection for Large-Scale

- Power Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(8): 4727-4739. <https://doi.org/10.1109/TSMC.2021.3104087>
- [25] Opriel S, Möller F, Strobel G, Otto B. Data Sovereignty in Inter-organizational Information Systems: Findings from Demand and Capacity Management in the Automotive Industry. Business & Information Systems Engineering, 2025, 67: 833-853. <https://doi.org/10.1007/s12599-024-00893-4>

Author's Biography



Zhang Siwen, Director of the New Energy Project Division, Planning & Development Department, Guodian Power Development Co., Ltd., takes overall responsibility for the full-process management of new energy projects at the preliminary stage. Deeply engaged in frontline new energy project development, he has solid expertise in key business areas including project scheme planning, design optimization, economic evaluation and investment decision-making, and has coordinated to deliver a host of new energy projects efficiently. He led the formulation of multiple management regulations for new energy projects, continuously improving the preliminary project management system and investment decision-making procedures to support standardized and efficient project delivery. He once took part in the reward fund project of China Energy Investment Corporation. With remarkable work achievements, he has repeatedly been honored as the Company's "Outstanding Employee". <https://orcid.org/0009-0005-5255-2119>