ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Distributed data collaboration Model and system implementation of blockchain enabled Misdemeanor Criminal Record Elimination

Jianbo Ma[1], *Deqi Zou*[1], *Zhuoren Zhou*[1], *Songyun Ren*[1], *Yuxiang Chen*[1] *and Yong Gao*[1,*]

[1] Harbin University of Commerce, Harbin 150028, Heilongjiang, China

SUMMARY: *In order to support collaborative processing in the business of misdemeanor criminal record elimination across judicial departments, this paper proposes a blockchain-enabled distributed data collaboration model, and implements a consortium blockchain prototype system, which enables application acceptance, condition verification, identity authentication, state synchronization, result writeback and audit tracing to be executed continuously in a unified link. The framework integrates on-chain index management, off-chain controlled record storage, credential authentication, chain code rule execution, and cross-department consistency update. An experimental data set containing 48000 collaborative process samples, 192000 on-chain state events and 144000 audit trail entries is constructed, and the evaluation is completed in a consortium blockchain environment with 13 collaborative nodes. Experimental results show that in E2 environment, the request delay is reduced to 1.53 s, the peak throughput reaches 463 TPS, the state synchronization accuracy is 99.31%, and the audit integrity is 99.79%. The results show that the proposed method can provide a reliable, traceable and consistent implementation path for the misdemeanor criminal record elimination scenario.*

KEYWORDS: *Blockchain, Misdemeanor criminal record elimination, Distributed data collaboration, Trusted audit*

1 Introduction

After digital governance has entered the cross-domain coordination stage, the elimination of misdemeanor criminal records is no longer a single department of information deletion and modification action, but a linkage process covering public security, procuratorial, court, judicial administration and social governance terminals. There are many calculation steps involved in the business flow, such as identity verification, condition verification, state change, result writeback and audit retention. As long as the timing control or authorization boundary is deviated, the accurate execution of the kill instruction will be affected. Although the centralized platform is convenient for centralized management, it relies more on a single point of control in terms of multi-subject concurrent access, heterogeneous data synchronization and trusted trace, which is difficult to support data governance for judicial collaboration scenarios. Blockchain has the characteristics of distributed consensus, on-chain rule execution and non-tampering records, which can establish a trusted state machine for the elimination of minor criminal records. Combined with off-chain storage, identity credentials and audit index, an implementation path that balances efficiency, privacy and collaborative consistency can be formed.

*gaoyong202603@163.com
https://doi.org/10.65102/is2026542
Published: 30 April, 2026

Focusing on identity and access management, Ghaffari et al. [1] systematically sorted out the implementation framework of distributed ledger in authentication and authorization. Ahmed et al. [2] summarized the core components of blockchain identity management system and autonomous identity ecosystem. Schardong et al. [3] summarized the structure, mapping relationship and application boundary of autonomous identity. Xiang et al. [4] proposed a decentralized authentication and access control protocol. Xiang et al. [5] introduced searchable attribute-based encryption into the blockchain-assisted data protection process. Yang et al. [6] proposed a multi-authority attribute-based encryption scheme for fine-grained access control. Focusing on reputation, reporting and big data governance, Hasan et al. [7] discussed a privacy protection reputation system combining blockchain and cryptography components. Zou et al. [8] constructed a decentralized reporting mechanism with proxy signature. Deepa et al. [9] summarized the architecture selection and implementation direction of blockchain in big data environment. Singla et al. [10] proposed a decentralized identity framework for the secure use of information system resources. To facilitate the observation of the relationship between the existing research and the task of this paper, Table 1 shows the representative work and its scope of adaptation.

Table 1: Current status of related research

Researcher	Main Research Focus	Scenario Adaptation Boundary
Ghaffari et al. [1]	Summarized the implementation framework of distributed ledgers in identity authentication and access management.	Oriented toward general identity management and does not cover state transition and result write-back in criminal record expungement.
Ahmed et al. [2]	Reviewed the core components of blockchain-based identity management and self-sovereign identity ecosystems.	Places more emphasis on identity system construction, with insufficient support for expungement condition verification.
Xiang et al. [4]	Proposed a decentralized authentication and access control protocol.	Suitable for access control scenarios, but does not incorporate composite business constraints.
Singla et al. [10]	Constructed a decentralized identity management framework.	Focuses on resource identity management, with limited coverage of cross-department state synchronization.
Hu et al. [12]	Designed a blockchain-based secure access control framework.	More oriented toward IoT data management, with limited support for result trace retention.
Bin Saif et al. [17]	Discussed a data access mechanism combining blockchain and IPFS.	Mainly focuses on access efficiency and does not extend to an audit closed loop.
Jakhar et al. [19]	Built a privacy-preserving and access control framework.	More suitable for sensitive data sharing, with less discussion of business collaboration.
AlKhanafseh et al. [20]	Applied blockchain to digital evidence preservation and verifiable retention.	Strong in evidence retention, but does not cover multi-node collaborative updating.

Further for network access, iot data and data market scenarios, Ghaffari et al. [11] summarized the network application of distributed ledger technology from two levels of

authentication link and access control link. Hu et al. [12] designed a secure access control framework based on blockchain. Klaine et al. [13] proposed a privacy-preserving blockchain data market platform. Gonzalez et al. [14] verified the feasibility of blockchain supporting scalable iot data transactions. Li et al. [15] proposed a more flexible blockchain access control mechanism. Agarkar et al. [16] implemented a decentralized system that takes into account both identity management and access control. Bin Saif et al. [17] discussed the data access efficiency under the combination of IPFS and blockchain. Deshmukh et al. [18] summarized the application path of blockchain in real-time data security scenarios. Jakhar et al. [19] constructed a privacy protection and access control framework for sensitive records management. AlKhanafseh et al. [20] applied blockchain to digital evidence preservation and verifiable retention. Existing results have covered key modules such as identity management, access control, distributed storage and evidence retention. However, in the misdemeanor criminal record elimination scenario, the business object has the characteristics of cross-department, cross-level and cross-time linkage, and the data state also contains continuous transformation semantics such as "to be reviewed, to be confirmed, eliminated, and written back". It is still difficult to form a complete collaborative closed loop only relying on a single identity framework or a single storage mechanism.

Compared with medical, iot, or general data markets, the previous record elimination collaboration emphasizes the deterministic expression of business rules and the execution sequence of state writeback. Nodes should not only verify whether the identity is legal and whether the request satisfies the authorization conditions, but also check the logical consistency between the judgment document, the execution period, the recieve mark and the department confirmation mark, and ensure that each state transition can be reviewed by the subsequent nodes. Therefore, the collaborative implementation of "on-chain rule index + off-chain business data + multi-node audit confirmation" is more suitable for this scenario. Based on the above research basis, this paper maps the business objects, extinction conditions, state flow and audit requirements into the blockchain collaboration framework. The following chapters successively expand the model construction, system implementation, experimental verification and result discussion.

2 Construction of distributed data collaborative model for misdemeanor criminal record elimination

2.1 Modeling of misdemeanor criminal record elimination business object and collaborative relationship

The misdemeanor criminal record elimination business is not a single field deletion process, but a collaborative computing process composed of the application subject, case records, judgment documents, execution information, audit departments, confirmation nodes and audit links. There are both static membership relations between different objects and state dependencies that change continuously with the advancement of time. In order to make the subsequent condition verification, status update and result writeback have a unified computing foundation, this section abstracts the business objects as computable nodes, and abstracts the cross-department interactions as relationship edges with direction, timing and constraints. On this basis, the business objects and collaboration relationship model of misdemeanor criminal record elimination are formed.

In order to put business objects from different sources into a unified computing space, this paper first jointly encodes the subject attributes, rule constraints, current state and

collaboration context to obtain a unified representation of the object:

$$e_i = \tanh(W_a a_i + W_r r_i + W_s s_i + b_1) \odot \sigma(U_c c_i + U_h h_i + b_2) + P p_i \quad (1)$$

where, e_i represents the unified representation vector of the i business object; a_i represents the underlying attributes of the agent. r_i represents the rule constraint feature; s_i represents the current state encoding. c_i stands for collaborative context; h_i represents the historical interaction trajectory. p_i represents the position of the process hierarchy. W_a, W_r, W_s, U_c, U_h, and P are parameter matrices.b_1 and b_2 are the bias terms. Formula (1) is used to compress heterogeneous business objects into a unified semantic space and provide comparable inputs for relational computation.

After the object representation is formed, the collaboration relationship needs to be organized from the four levels of "business object layer, rule calculation layer, state execution layer, audit trace layer", but can not only be described by the department name side by side. Fig. 1 puts the application subject, case materials, rule conditions, collaborative status and audit results in the same computing link, which can more clearly illustrate the input source, processing path and output destination of various nodes in the misdemeanor criminal record elimination scenario.

Figure 1: Misdemeanor record elimination business object and collaborative relationship modeling structure

After the object representation is formed, the collaborative relationship needs to be further quantified. Considering such factors as department distance, business constraint matching, time interval and confirmation mark, the relationship weight is defined as follows.

$$\omega_{ij}^{(k)} = \frac{\exp\left(\alpha_k^{\top}[z_i \| z_j \| d_{ij} \| q_{ij}]\right)}{\sum_{j' \in N_i} \exp\left(\alpha_k^{\top}[z_i \| z_{j'} \| d_{ij'} \| q_{ij'}]\right)} \cdot (1 - \lambda \Delta t_{ij}) + \mu m_{ij} \tag{2}$$

Here, $\omega_{ij}^{(k)}$ represents the interaction strength between object i and object j under the k synergistic relationship. z_i and z_j denote the two-end node representation; d_{ij} denotes the departmental distance code; q_{ij} represents the business constraint matching vector. α_k is a relation type parameter; Let N_i denote the set of neighbors associated with node i; Δt_{ij} represents the interaction time interval; m_{ij} stands for mandatory confirmation identifier. λ and μ are the regulation coefficients. Equation (2) is used to calculate the collaborative weights between different objects, so that the audit link, acknowledgement link and writeback link can be distinguished and modeled in the same graph structure.

After the object representation and relationship weight are determined, the next change of business state can be determined by the collaborative results of multiple nodes, and its state transition probability is expressed as follows.

$$\pi_i^{t+1} = \mathrm{Softmax}\left(M_1 e_i + M_2 \sum_{j \in N_i} \omega_{ij}^{(k)} z_j + M_3 \delta_i^t + M_4 \gamma_i^t \right) \tag{3}$$

Here, π_i^{t+1} denotes the probability distribution of object i falling into each service state at time $t+1$. M_1 represents the mapping matrix of the object itself; M_2 represents adjacency collaborative information aggregation matrix; Let δ_i^t denote the current state vector; Let γ_i^t denote the audit feedback vector; Softmax is used to convert multidimensional scores into state probabilities. Formula (3) is used to describe the state migration process of business objects under the collaboration of multiple nodes, so as to transform the states of "to be reviewed, to be confirmed, eliminated, and written back" into computable results.

Through the above modeling, subject identity, business conditions, department collaboration and audit constraints are unified into the same computational framework. The model not only retains the sequentiality in the judicial business chain, but also provides a structural basis for subsequent on-chain index construction, off-chain data bearing and cross-departmental consistency synchronization.

2.2 On-chain and off-chain double-layer mapping for misdemeanor criminal record elimination data

Misdemeanor criminal record elimination data includes identity, case documents, execution records, audit conclusions and writeback results, and has the characteristics of heterogeneous sources, continuous status and controlled update. If all the content is written directly on the chain, the storage burden and consensus overhead will rise, and the document details and identity will also be exposed to a wider range in the transmission. In order to balance trusted verification and collaboration efficiency, this paper adopts a double-layer mapping structure of on-chain index, off-chain entity and two-way verification. The condition summary, state pointer, audit fingerprint and access policy are written into the alliance chain, and the judgment documents, execution details and writeback copies are stored in the controlled data domain. Fig. 2 shows the two-layer mapping structure.

In this structure, the upper layer of the chain does not carry the complete text, but keeps a

compact representation of the triggered verification and synchronization. The lower layer of the chain hosts the original case, deadline field, confirmation record, and version copy, and completes the location through the shard index. The mapping between the two layers is continuously updated with state transitions. When the case was transferred from being reviewed to being confirmed, the on-chain index synchronously refrezed the state hash and timestamp. When the elimination result enters the writeback phase, the off-chain data block generates a new version identity, and the on-chain audit anchor records the source of the change and the write round. After this processing, the on-chain scale remains stable, and the off-chain modification can also be recognized by each node.

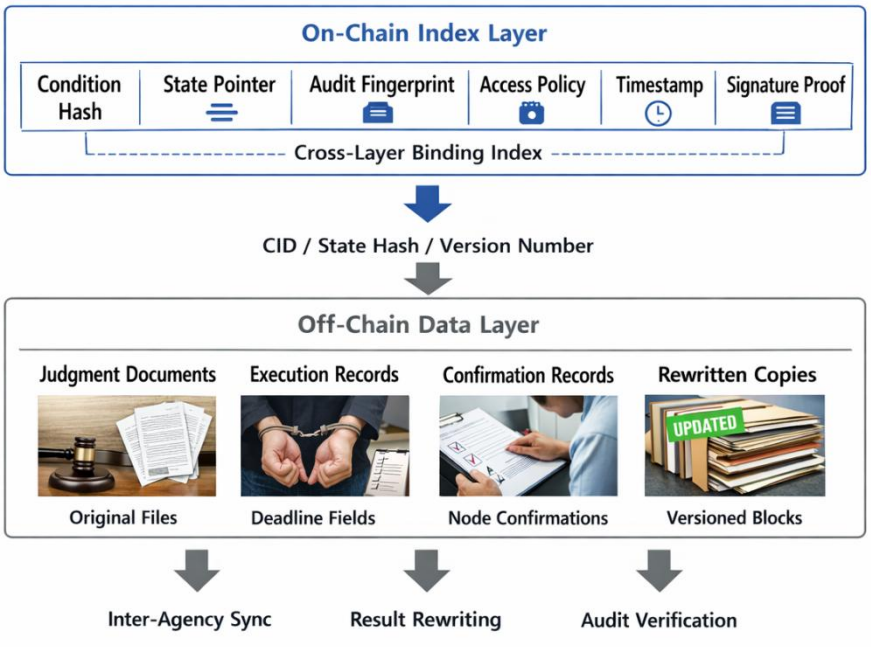

Figure 2: on-chain and off-chain two-layer mapping structure for misdemeanor record elimination data

After the unified representation of business objects is completed, the on-chain and off-chain two-layer mapping does not repeat the original attribute encoding, but further compreses the object representation into an on-chain verifiable index, which forms the basis of cross-layer mapping together with version identification, access policy and audit summary. Thus, the on-chain index generation process is defined as follows.

$$k_i = \tanh(R_1 e_i + R_2 v_i + R_3 h_i + b_3) \odot \sigma(S_1 g_i + S_2 u_i + b_4) + T\tau_i \quad (4)$$

Here, k_i denotes the on-chain index vector corresponding to the article i misdemeanor criminal record elimination. e_i represents the unified representation of business objects generated by formula (1); v_i denotes the off-chain version code. h_i represents the hash feature of the document summary and the executive summary. g_i denotes the access policy encoding. u_i represents the audit anchor identity. Let τ_i denote the timestamp location embedding; R_1, R_2, R_3, S_1, S_2, and T are mapping matrices;b_3 and b_4 are the bias terms. Instead of repeating the object modeling process, Eq. (4) further compreses the object representation into an on-chain index representation for subsequent cross-layer binding, consistency determination, and synchronous output calculation.

After the base index is formed, we need to bind the off-chain data blocks to the on-chain

state anchors, so that version changes, node acknowledgments, and state transitions can be sensed by the same compute link:

$$\beta_{ij} = \frac{\exp\left(q^{\top}\left[x_i \| y_j \| d_{ij} \| c_{ij}\right]\right)}{\sum_{j' \in \mathcal{N}(i)} \exp\left(q^{\top}\left[x_i \| y_{j'} \| d_{ij'} \| c_{ij'}\right]\right)} \cdot e^{-\eta \Delta\tau_{ij}} + \rho g_{ij} \tag{5}$$

Here, β_{ij} denotes the binding strength between the anchor on the i chain and the data block off the j chain. y_j represents the off-chain block feature; d_{ij} denotes the state difference vector; c_{ij} stands for department confirmation code; Let $\mathcal{N}(i)$ denote the set of candidate blocks associated with anchor i; $\Delta\tau_{ij}$ denotes the time interval; η is the attenuation coefficient; g_{ij} denotes the mandatory writeback flag. Let ρ be the conditioning term. Equation (5) is used to measure the binding stability of cross-layer objects.

After the cross-layer binding is complete, the system also needs to determine whether the current version meets the formal synchronization criteria to ensure that index updates, off-chain writeback, and audit records are kept in the same pace:

$$\chi_j = 1 - \frac{\left\| C_1 x_j - C_2 \hat{m}_j \right\|_2^2}{\left\| C_1 x_j \right\|_2^2 + \left\| C_2 \hat{m}_j \right\|_2^2 + \varepsilon} + w^{\top} a_j \tag{6}$$

Here, χ_j denotes the consistency score of the j data block. $\hat{m}_j$ represents the recomputation result of the off-chain summary; C_1 and C_2 are the reconstruction matrices; ε is a numerically stable term; a_j represents the department confirmation and audit feedback vector; w is the weighting parameter. Formula (6) is used to determine whether the current version has the condition to enter the synchronization phase.

After the cross-layer binding relationship and consistency score are determined, the two-layer mapping module needs to generate the synchronous submission results that can be directly invoked by each cooperative node. The result is no longer responsible for the status classification function, but is used to drive on-chain index update, off-chain version confirmation, and cross-department writeback execution. With this goal, the cross-layer synchronous commit vector is defined as follows.

$$r_i^{t+1} = \phi\left(N_1 k_i + N_2 \sum_{j \in \mathcal{N}(i)} \beta_{ij} y_j + N_3 \chi_i + N_4 q_i^t \right) + o_i^t \tag{7}$$

Here, r_i^{t+1} denotes the cross-layer synchronous commit vector of the i record at time $t+1$. k_i denotes the on-chain index representation generated by Eq. (4); Let β_{ij} denote the cross-layer binding strength obtained by Eq. (5). y_j represents the off-chain data block characteristics; χ_i denotes the consistency score calculated by Eq. (6); q_i^t denotes the audit feedback and writeback control vector at the current time; N_1, N_2, N_3, and N_4 are mapping matrices; Let $\phi(\cdot)$ denote the nonlinear activation function; o_i^t represents the last round of synchronization output residuals. Equation (7) is used to generate cross-layer synchronous commit results, so that on-chain index refresh, off-chain version confirmation and cross-department writeback operations share the same calculation entry, instead of repeating the modeling task of state transition probability.

Through the above double-layer mapping design, the misdemeanor criminal record elimination data obtained an on-chain index, a clear version boundary and a continuous state

mapping basis, and also provided a computing entry for the subsequent trusted cooperative control.

2.3 Trusted collaborative control mechanism driven by blockchain

2.3.1 Elimination condition verification and identity trusted authentication

The trusted collaborative control of misdemeanor criminal record elimination is not a static verification of a single application record, but integrates the application subject, case materials, execution status, department confirmation and chain credentials into the same control link. Only when the elimination condition and the identity credentials satisfy the coordination rules, the subsequent state migration and result writeback can enter the formal execution phase. Therefore, this section establishes a control mechanism from two aspects of conditional verification and trusted identity authentication, so that the business request has a verifiable, traceable and reachable computing basis before entering the on-chain process.

In order to transform judgment documents, execution time limits and recidivian marks into computable judgment results, this paper first constructs the elimination condition score function:

$$c_i = \sigma(w_1^\top \tanh(A_1 d_i + A_2 l_i + A_3 u_i - A_4 f_i) + w_2^\top q_i - \lambda \|\Delta\tau_i\|_2 + b_8) \tag{8}$$

where c_i represents the condition satisfaction score of the i record; d_i represents the validity code of judgment documents; l_i represents the execution deadline completion; u_i indicates the execution state of punishment; f_i denotes the recief-marking constraint; q_i denotes the complementary rule vector; $\Delta\tau_i$ denotes the deadline offset; A_1 to A_4, w_1, w_2, and b_8 are parameters. The formula is used to compress discrete conditions into continuous judgment results, and provide a unified entrance for subsequent authentication and execution control.

After the conditional score is formed, it is also necessary to confirm whether the request body is consistent with the on-chain identity declaration. Since cooperative nodes belong to different departments, identity verification should not only rely on static accounts, but also absorb information such as credential signature, role mapping and revocation status. Therefore, this paper further defines the identity trusted authentication function:

$$\psi_i = \frac{\exp(m_1^\top (x_i \odot s_i) + m_2^\top (r_i \odot k_i) - m_3^\top v_i + b_9)}{1 + \exp(-m_4^\top z_i)} \tag{9}$$

Here, ψ_i denotes the identity credibility of subject i; x_i stands for distributed identity coding; s_i stands for credential signature vector; r_i stands for role permission mapping. k_i denotes the binding code of post and department; v_i denotes the credential revocation status. z_i represents the historical authentication trajectory; m_1 to m_4 and b_9 are parameters. This formula is used to measure the consistency between subject declarations, on-chain credentials, and department roles, so as to block inconsistent identities from entering subsequent collaboration links.

When the condition score and the identity credibility are generated at the same time, the control module also needs to output the final admission result, so that the consortium chain can decide whether the request enters the audit channel, complements the verification channel, or denies the channel. Therefore, the joint control output function is constructed as follows.

$$\pi_i = \mathrm{Softmax}(Q_1 c_i + Q_2 \psi_i + Q_3 h_i + Q_4 g_i + Q_5 (c_i \psi_i) + b_{10}) \tag{10}$$

Here, π_i denotes the cooperative admission probability distribution of record i. c_i is the conditional score. ψ_i is the identity credibility; h_i indicates history review feedback. g_i represents the department confirmation vector. Q_1 to Q_5 and b_{10} are the output parameters. In this equation, business conditions, identity authentication and node feedback are incorporated into the unified output space, so that the trusted cooperative control is extended to an executable on-chain decision process.

Through the above design, the condition determination, identity authentication and admission screening have been completed before the request enters the state synchronization and the result is written back, which provides a stable input and audit support foundation for subsequent cross-department consistency execution.

2.3.2 Record status update and cross-department consistency synchronization

The record status update is not written by a single node, but a sequential propagation process involving the court, procuratorial, public security and judicial administration nodes. After a node completes the elimination confirmation, the state cannot directly cover the original record, but must first go through the version verification on the chain, collaborative signature aggregation and writeback window judgment, and then enter the next round of synchronization. In order to ensure that the four states of "to be reviewed, to be confirmed, eliminated, and written back" are consistent between different departments, the system maintains a status index on the alliance blockchain, stores business copies off-chain, and controls the update rhythm through the block height, timestamp and confirmation threshold. After being processed in this way, the prior state is transformed into a computable object, and the cross-department synchronization is also transformed into a constrained automatic propagation link.

Fig. 3 shows the record status update and the cross-department consistency synchronization structure. In the figure, the left is the state trigger source, the middle is the version verification and confirmation aggregation module, the right is the synchronous submission and writeback feedback module, and the bottom audit chain is responsible for recording the state evolution sequence, the confirmation source and the final landing point.

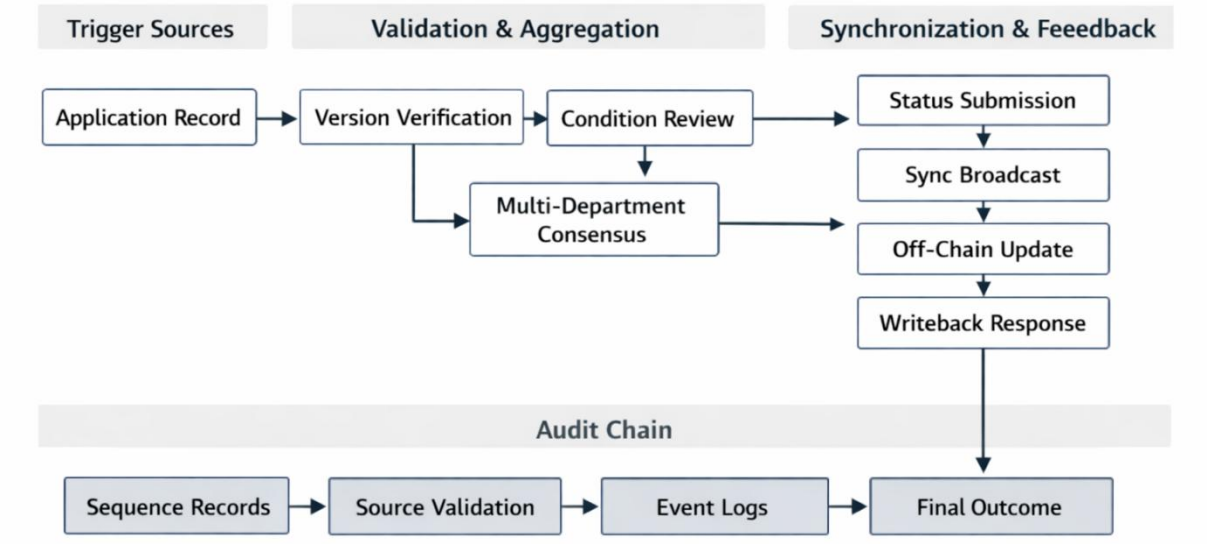

Figure 3: Prior record status update with cross-department coherence synchronization structure

In order to incorporate multi-department confirmation results, historical versions and

timing constraints into the same update process, this paper defines the update vector of criminal record state from time t to $t+1$ as follows.

$$s_i^{t+1} = \tanh\left(G_1 s_i^t + G_2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij} m_j^t + G_3 u_i^t + G_4 v_i^t - G_5 \Delta\tau_i \right) \odot \sigma(H_1 c_i^t + H_2 p_i^t) + \Pi_i^t \tag{11}$$

where s_i^{t+1} denotes the state vector of record i at time $t+1$; s_i^t represents the current state; m_j^t denotes the acknowledgement message of the j cooperative node; Let α_{ij} denote the action weight of node j on record i; u_i^t denotes the on-chain index representation; v_i^t denotes the version encoding; $\Delta\tau_i$ represents the state update time offset; c_i^t denotes the confirmation coverage vector; p_i^t denotes the writeback window control quantity; Let Π_i^t denote the off-chain copy feedback term. Equation (11) compresses the version, acknowledgement, and timing constraints into a unified update space, so that state advancement no longer depends on the local writing of a single department.

When the state vector is generated, it is also necessary to determine whether the nodes achieve the consensus synchronization condition. If the consistency is insufficient, the system will only refresh the candidate states without writeback. If the consistency reaches the threshold, the on-chain state and the off-chain replica are committed synchronously. To this end, this paper constructs a cross-department consistency score function:

$$\xi_i = \sigma\left(\sum_{k=1}^{K} \omega_k \frac{h_i^{t+1} \cdot r_k}{\|h_i^{t+1}\|_2 \|r_k\|_2 + \varepsilon} + \eta \ln(1 + b_i) - \mu \|e_i\|_2 \right) \tag{12}$$

Here, ξ_i denotes the consistency score of record i; h_i^{t+1} denotes the state summary; r_k denotes the confirmation response of the k department; Let ω_k denote the department weights; b_i represents the block confirmation depth; e_i represents writeback differential error; η and μ are the adjustment coefficients; Let ε be a stable term. Formula (12) simultaneously absorbs three types of information: state summary matching, department confirmation coverage and on-chain confirmation depth, which is used to distinguish the two execution paths of "commitable synchronization" and "continue to wait for confirmation".

Through the above update and decision mechanism, each change of the record status carries the source node, the version sequence number and the confirmation set, and the on-chain index refresh and off-chain copy writeback always advance along the same time sequence.

2.3.3 Eliminate result trace and traceability audit

Result trace elimination and traceability audit do not save the single written result passively, but organize the trigger request, condition verification, identity authentication, state transition, department confirmation and result writeback into a continuous audit calculation chain. Each record elimination operation generates an independent event unit, and registers the event summary, operator identification, timestamp, version number and writeback target on the chain, so as to ensure that the subsequent nodes can review the history trajectory along the unified index. Different from general log records, the trace object here contains not only the result itself, but also the rule set, the confirmation sequence and the execution context on which the result is formed. Therefore, the audit process can cover the complete link of "who initiated the writeback, according to what conditions, in which confirmation round, and by

which nodes".

In order to transform discrete audit events into unified records that can be reviewed, this paper first constructs the event trace summary function:

$$a_i = \tanh(R_1 o_i + R_2 c_i + R_3 v_i + R_4 g_i + b_{11}) \odot \sigma(S_1 t_i + S_2 u_i + b_{12}) + H h_i) \tag{13}$$

Here, a_i represents the trace summary vector of the i audit event. o_i represents the encoding of the operation result. c_i stands for acknowledgment set encoding; v_i represents the version migration vector. g_i represents rule dependency summary. t_i stands for timestamp embedding; u_i represents the operator's credentials. h_i denotes the off-chain hash summary; R_1 to R_4, S_1, S_2 and H are mapping matrices, and b_{11} and b_{12} are bias terms. Formula (13) is used to compress multi-source audit information into a unified representation space.

Among them, audit summary plays the role of both index and verification. If the summary read by the subsequent node is inconsistent with the off-chain recomputation result, the system will not directly overwrite the original record, but freeze the round write and keep the difference copy, waiting for the review node to re-confirm. In this way, every change in the result of criminal record elimination has the structural characteristics of source traceable, order verifiable and version traceable.

After the trace summary is formed, it is also necessary to describe the degree of confirmation of each node for the same audit event. Because the confirmation time of courts, procuratorates, public security and judicial administrative departments is not consistent, the system must jointly calculate the signature credential, confirmation coverage rate and block confirmation depth to determine whether the audit chain has reached the archivable state. Based on this consideration, this paper defines the audit confirmation aggregation function as follows.

$$\zeta_i = \sigma\left(\sum_{k=1}^{K} \omega_k \frac{a_i^{\top} s_{ik}}{\|a_i\|_2 \|s_{ik}\|_2 + \varepsilon} + \eta \ln(1 + d_i) - \mu \|\delta_i\|_2 + \nu r_i\right) \tag{14}$$

where ζ_i denotes the audit confirmation score of event i; s_{ik} represents the confirmation signature vector uploaded by the k department; Let ω_k denote the department weights; d_i denotes the block confirmation depth; Let δ_i denote the on-chain off-chain difference vector; r_i indicates the writeback completion mark. η, μ and ν are the regulation coefficients and ε is the stability term. Equation (14) is used to evaluate the degree of confirmation convergence of an audit event and decide whether it enters the formal filing phase.

The output of this function is not simply a token but a continuous audit confidence score. The higher the score, the more fully the confirmation coverage of the event in the cross-department collaboration, the more stable the block depth on the chain and the signature set. Only when the credibility exceeds the archiving threshold, the system will write the current result to the final audit anchor and broadcast the archiving completion signal to the off-chain replica.

After the event summary and confirmation aggregation are completed, the audit module also needs to generate the traceability vector to support the subsequent nodes to retrieve the elimination records in three dimensions: time, department and version. In order to keep the traceability result consistent with the preceding control link, the traceability audit output function is further defined as follows.

$$y_i^{t+1} = \mathrm{LayerNorm}(M_1 a_i + M_2 \zeta_i 1 + M_3 q_i + M_4 e_i^t) + p_i \tag{15}$$

Here, y_i^{t+1} denotes the traceback output vector of event i at time $t+1$. a_i is the trace summary; ζ_i is the audit confirmation score; q_i denotes the query context encoding; e_i^t represents the feedback from the previous round of audit; p_i represents the time-department-version joint location encoding. M_1 to M_4 are mapping matrices; 1 is a constant vector. LayerNorm represents the layer normalization operation. Equation (15) is used to generate a unified traceability representation that enables different nodes to recover the complete execution path along the same audit index.

Through the above design, elimination results, confirmation traces and history versions are unified into the same audit space, and nodes can trace the execution path along the anchor points on the chain without relying on decentralized log concatenation.

3 Implementation and experimental verification of distributed cooperative system for misdemeanor criminal record elimination

3.1 Implementation of prototype system and construction of experimental environment

3.1.1 Deployment of Collaborative Nodes and Configuration of consortium blockchain Environment

This section focuses on the prototype landing process of the distributed collaborative system for misdemeanor criminal record elimination, and explains the deployment method of collaborative nodes, the composition of the alliance chain network and the configuration of the experimental environment. The system adopts a hierarchical deployment structure of "judicial business node-consensus ranking node-audit supervision node-off-chain storage node", in which the court, procuratorial, public security and judicial administrative departments undertake the tasks of application acceptance, status confirmation, result writeback and supervision review respectively, and the ranking service is responsible for block generation and transaction ranking. The off-chain storage cluster is responsible for keeping the judgment documents, execution records, and history writeback copies. In order to ensure the trusted collaboration of different departments in the same network, the system deployidentity registration service, chain code management service and audit index service uniformly on the consortium blockchain, so that the business request can complete node authentication, channel allocation and policy loading before entering the status update process.

In the actual deployment, the business nodes run on an independent server through Docker containers, the chain code is loaded into the judicial cooperation channel in a modular way, and the certificate service is responsible for node admission control and communication identity verification. The off-chain database uses sharded document storage to save large volume of business records, and only the status index, version number and audit summary are kept on the chain to reduce the block load and shorten the confirmation delay. The experimental environment uses Ubuntu Server 22.04 operating system, the business node and consensus node are configured with 8-core CPU, 32 GB memory and Gigabit Ethernet interface, the consortium blockchain platform version is Hyperledger Fabric 2.5, and the container running environment is Docker 24.0. The configuration of each node is shown in Table 2.

Table 2: Configuration of cooperative node environment

Node Type	Quantity	Core Function	Deployment Key Points
Business Node	4	Request handling and confirmation	Independent containers
Consensus Node	3	Ordering and block packaging	Independent channels
Audit Node	2	Supervision and trace tracking	Read-only indexing
Storage Node	4	Document storage and replica maintenance	Sharded deployment

In order to maintain the stability of the state broadcast, the court node is set as the acceptance master node, the procuratorial and public security nodes are set as double confirmation nodes, and the judicial administration node is responsible for the write-back verification and result review. Sorting nodes are deployed independently to avoid contention for computing resources between business containers and consensus services. The channel-level access policy is loaded by department role, and the audit node only retains index read permission and does not participate in service writing. TLS encryption and two-way certificate verification are enabled in network communication, and the off-chain storage nodes maintain the consistency of document data through copy groups and timing verification, thereby reducing the jitter of cross-node transmission. After this process, the system forms an operation pattern of "on-chain control, off-chain hosting, and cross-node collaboration", which provides a stable environment for subsequent performance testing and consistency verification.

3.1.2 Functional module implementation and business process joint tuning

This section explains the function module implementation and business process coordination of the distributed collaborative system for misdemeanor criminal record elimination. The prototype system was implemented hierarchically according to six modules: acceptance and access, condition verification, identity authentication, status synchronization, result writeback and audit tracking. Each module completed sequential linkage through chain code interface and message bus. After the application request enters the system, the business gateway first completes the field normalization and the request number generation, and then writes the case identification, the subject certificate and the material summary into the preprocessing cache. The condition verification module then calls the rule engine to compare the validity of judgment documents, the execution period, the re-offense constraint and the department confirmation conditions in parallel. The identity authentication module synchronously read the credential and role mapping table on the chain to complete the node identity verification and signature verification. When the two kinds of results meet the execution threshold, the request enters the state synchronization module. To ensure clear boundaries of data transfer and call between modules, Table 3 presents the implementation tasks of core modules.

Table 3: Core functional modules and joint tuning tasks

Module Name	Main Input	Main Output	Integration Responsibility
Request Intake Module	Application records, subject information	Request ID, normalized data	Completes request encapsulation and preprocessing distribution
Condition Verification Module	Document summary, execution information	Condition determination result	Outputs rule-matching results
Identity Authentication Module	On-chain credentials, role mapping	Authentication result, signature verification result	Completes node identity verification
State Synchronization Module	Condition result, authentication result	State index, version number	Refreshes on-chain state and broadcasts updates
Result Write-Back Module	Version number, synchronization result	Off-chain replica update result	Completes business write-back
Audit Tracking Module	Transaction hash, result summary	Audit index, tracking record	Forms a reviewable audit chain

The module joint tuning adopts the closed-loop mode of "on-chain submission-off-chain writeback -audit and review". After receiving the confirmation result, the state synchronization module refreshed the state index on the chain and generated the version number. The writeback module wrote the result to the off-chain business copy according to the version number, and passed the difference summary back to the audit module. The audit module is not involved in business writing, but is only responsible for recording the origin of the operation, processing rounds, timestamps, and result summaries, and returning the trace index to the query interface. In the process of joint tuning, the system adopts asynchronous queue peak shaving, idempotent commit control and failure retry mechanism to avoid disturbance of repeated writes to service links. After the module-level connectivity test, the four main links of acceptance, verification, synchronization and writeback can run stably in the same transaction context, which provides a complete implementation basis for subsequent performance testing and consistency verification. In addition, condition checking and identity authentication shared read-only cache to reduce the overhead of repeatedly parsing document fields. The state synchronization and writeback modules keep the order consistent through version lock to prevent the old state from overwriting the new result. The audit module packages the transaction hash, writeback identity, and confirmation set into a unified trace entry. Interface return code, chain code log and replica verification results are used as the joint acceptance basis in the whole joint adjustment process.

3.1.3 Data sample organization and experimental task setup

This section focuses on the experimental infrastructure construction of the distributed cooperative system for misdemeanor criminal record elimination, and explains the data sample organization method, task load configuration and test set division strategy. In order to make the performance evaluation cover the whole service link, the sample design does not take a single record as the unit, but a complete coordination process as the minimum

experimental unit. Each unit also contains the identity of the application subject, the summary of the judgment document, the execution deadline field, the department confirmation sequence, the state transition record and the audit index information, and concatenates the on-chain index and the off-chain copy with a unified transaction number. The original sample is composed of simulated judicial business data and rule engine generated data, and a standard data set is formed after field desensitization, time alignment, state completion and outlier removal. Table 4 presents the basic organization of the experimental samples. At the same time, ensure that the input format of each batch of experiments is consistent with the sequence.

Table 4: Organization of experimental samples

Sample Type	Quantity	Description
Collaborative Process Samples	48,000	Each sample is defined as one complete business flow
On-Chain State Events	192,000	Includes states such as request intake, confirmation, and write-back
Off-Chain Replica Records	48,000	Stores business details and versioned replicas
Audit Tracking Entries	144,000	Records timestamps, node information, and result summaries

In order to ensure the comparability of test results under different scales, the system constructs the transaction flow according to three task modes: light load, medium load and high load, and keeps the request type, the number of confirmation nodes and the distribution of writeback paths consistent. After the sample organization is completed, the request sequence is generated according to the uniform arrival rate function instead of the random triggering method. The transaction arrival rate is defined as follows.

$$\lambda_t = \lambda_0(1 + \alpha \ln(1 + n_t))e^{-\beta t} \tag{16}$$

In the equation, λ_t represents the transaction arrival rate at time t, λ_0 represents the baseline load, α represents the concurrent growth coefficient, n_t represents the number of active requests in the current time window, and β represents the decay term. This formula is used to control the input strength of traffic requests in different load stages, so that node acknowledgment, state synchronization and result writeback are tested in the same rhythm.

In the experiment, the training set, validation set and test set are divided by 6:2:2, and the distribution of department roles, state categories and time span in each set are consistent. In terms of task setup, the system executes four types of tests: single link confirmation, double confirmation concurrency, full link writeback and audit trail. Each type of test continuously inputes fixed rounds of transactions to observe the performance changes in the process of acceptance, verification, synchronization and writeback. In this way, the experiment not only reflects the processing ability of the system in the standard scenario, but also covers the state propagation characteristics under multi-node cooperation, which provides a unified input basis for subsequent verification of response delay, throughput performance and consistency.

3.2 System experimental results and performance analysis

3.2.1 Eliminate request response latency analysis

This section focuses on the analysis of the response time delay of the elimination request of

the distributed cooperative system for misdemeanor criminal record elimination. The test takes the complete service link as the object, and includes the timing range from application submission, condition verification, identity authentication, status synchronization to result writeback. In order to reduce the deviation caused by the instantaneous jitter of a single node, the experiment performed five consecutive rounds for each group of transactions, and took the average value as the final delay. The test environment is divided into E1 and E2. E1 adopts standard cooperative configuration, and E2 enables read-only cache and asynchronous writeback queue under the same node scale. As the transaction size increases from 100 to 5000, the response time of the three node configurations increases, but the growth speed is not consistent. In E1 environment, it reaches 2.84 s at 5000 transactions for 6 nodes, 2.31 s for 9 nodes, and 1.97 s for 12 nodes. When the node size increases, the acknowledgement messages are more evenly distributed to multiple processing channels, and the request queuing time and the version broadcast waiting time decrease simultaneously, so the delay performance in the large-scale cooperation scenario is more stable. Fig. 4 shows the average response delay for different node sizes in E1 environment.

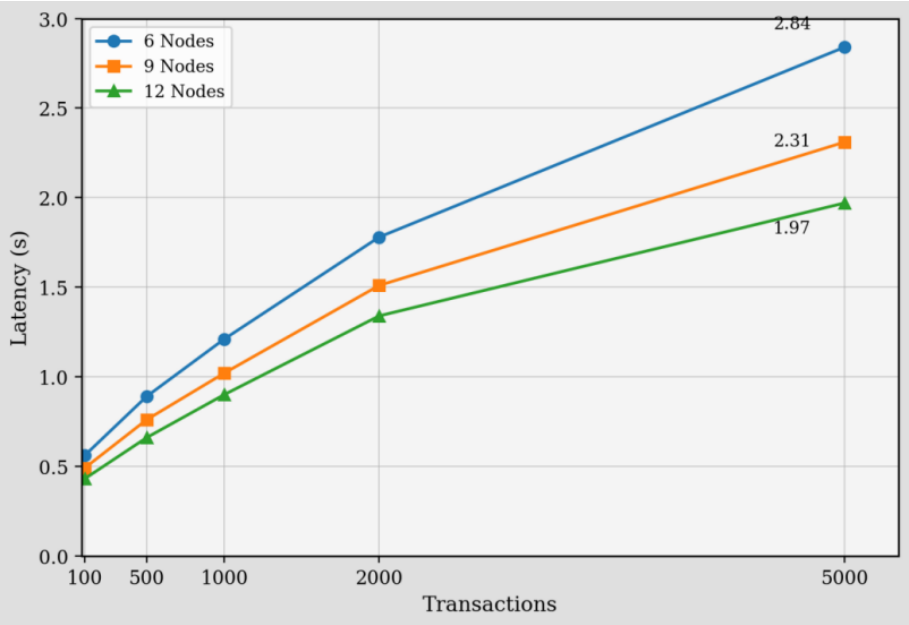

Figure 4: Average response delay of the elimination request for different node sizes in E1 environment

Compared with E1, the E2 environment maintains a lower latency level for the same transaction size. The reason is that the off-chain replica update and the audit index write are split into asynchronous queues, and the main link only retains the three key steps of state submission, version confirmation and result return, which reduces the synchronous blocking. Fig. 5 shows the test results in E2 environment. For 100 transactions, the average response latency of 6 nodes, 9 nodes and 12 nodes is 0.42 s, 0.36 s and 0.31 s, respectively. 2.26 s, 1.88 s and 1.53 s for 5000 transactions, respectively. Compared with the 12-node configuration in E1, the response latency of the corresponding configuration in E2 is further reduced by 22.3% under 5000 transactions, indicating that cache hits and asynchronous writeback have an obvious buffering effect on high concurrent requests. Combining the two sets of experimental results, it can be seen that node scale expansion can improve the processing balance of cooperative links, and off-chain write decoupling can further compress the request waiting time, which together form the basis for the system to maintain a stable response under high

load conditions. From the shape of the curve, when the number of transactions is less than 1000, the difference between the three groups of configurations is small. When the transaction volume continues to rise, the growth slope of 6-node environment is significantly higher than that of 9-node and 12-node environment, which indicates that confirmation propagation and write queuing have become the main source of time consumption.

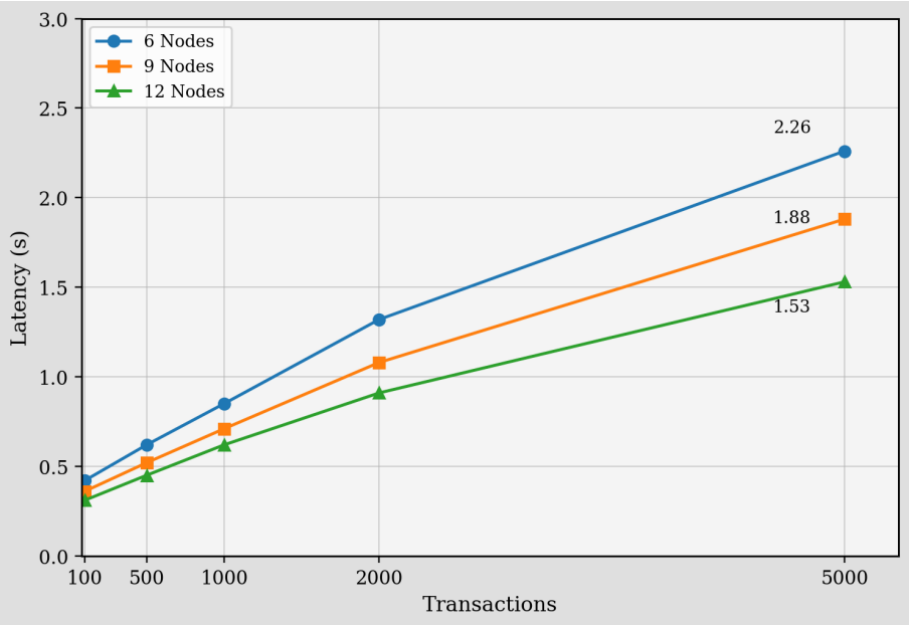

Figure 5: Average response delay of the elimination request for different node sizes in E2 environment

It can be seen that with the increase of node scale, the request queuing and confirmation waiting time are reduced, and the system can still maintain a relatively stable response ability under high load. The latency performance of E2 is further better than that of E1, which indicates that the asynchronous writeback and caching mechanism can significantly alleviate the main link blocking. This indicates that the collaborative architecture constructed in this paper can support continuous request processing in the scenario of misdemeanor record elimination.

3.2.2 Analysis of cross-department collaboration throughput performance

This section focuses on the analysis of the cross-department cooperative throughput performance of the distributed cooperative system for misdemeanor criminal record elimination under different loads and node sizes. Throughput is not a single chain code call, but a complete transaction that completes application acceptance, condition verification, authentication, status submission, writeback confirmation, and audit registration. The test is still divided into two types: E1 and E2. The former uses a standard synchronous writeback link, and the latter enables asynchronous audit writes and read-only caching under the same node size. The results show that the system throughput increases steadily with the node scale expansion, but there are differences in the growth range in different transaction intervals. When the transaction size increases from 100 to 1000, the throughput of the three groups of configurations continues to increase, and the 12-node configuration increases from 168 TPS to 412 TPS, which is significantly higher than the 6-node configuration of 295 TPS. When the transaction scale continued to increase to 5000, confirmation queues began to appear in the 6-node environment, and the throughput dropped to 241 TPS. 9 nodes and 12 nodes still

maintain high output, which are stable at 326 TPS and 387 TPS, respectively, indicating that the multi-node confirmation can effectively share the broadcast and submission pressure. Fig. 6 shows the correspondence between transaction size, number of nodes and throughput in E1 environment.

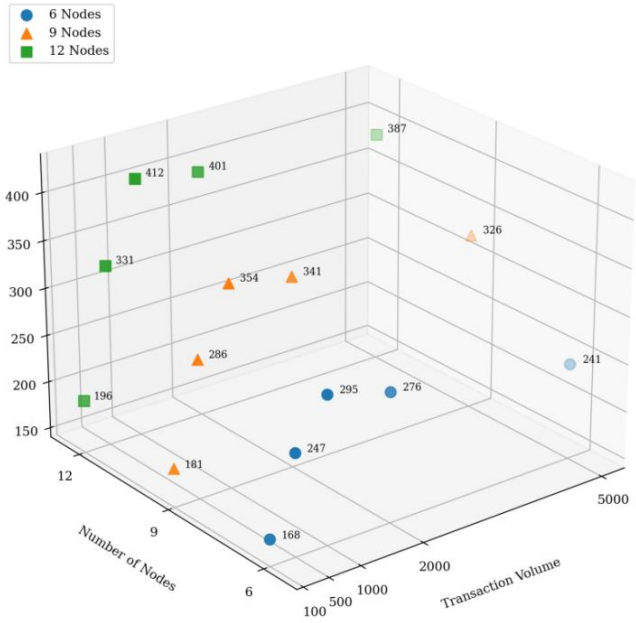

Figure 6: Results of cross-department collaborative throughput performance distribution in E1 environment

The overall performance is further improved in E2 because the audit index write and off-chain copy refresh are split into asynchronous queues, and only state commits and version confirmations are kept in the main link, resulting in a lower wait overhead after block packing. Comparison tests show that the peak throughput of 12-node configuration in E2 environment reaches 463 TPS, which is 12.7% higher than that of E1. To observe the specific contribution of each implementation component to the throughput performance, Table 5 further presents the ablation experiment results. After removing the asynchronous audit queue, the throughput decreases most obviously in the high load scenario. After removing the role cache, the repeated queries in the authentication phase increased, and the processing efficiency under 1000 transactions began to decline. After removing the version lock, although the number of short-term concurrent writes increases, the frequency of conflict rollback increases, and the final throughput decreases.

Table 5: Cross-department collaborative throughput performance ablation experiments

Configuration Scheme	1,000 Transactions / TPS	5,000 Transactions / TPS	Performance Change Description
Full System	412	387	Throughput remains stable
Without Asynchronous Audit Queue	376	321	Performance drops significantly under high load
Without Role Cache	389	346	Authentication overhead increases
Without Version Lock	401	333	Rollbacks increase, causing throughput decline

On the whole, the cross-department collaboration throughput performance is directly related to node size, confirmation path and module decoupling degree. The 12-node configuration maintains higher and more stable output in both types of environments, indicating that the consortium blockchain collaboration architecture constructed in this paper can maintain stable processing ability in the scenario of misdemeanor criminal record elimination and provide a performance basis for subsequent consistency verification.

3.2.3 State synchronization accuracy and audit integrity verification

This section focuses on the verification of the state synchronization accuracy and audit integrity of the distributed cooperative system for misdemeanor criminal record elimination. The accuracy statistical object is not the single chain code return value, but whether the final status of a complete business submitted in the court, procuratorial, public security and judicial administration nodes is consistent. Integrity measures whether the timestamp, operation subject, version number, confirmation set and writeback summary in the audit chain can form a closed record. The test uses two types of environments, E1 and E2, with transaction sizes set to 100, 500, 1000, 2000 and 5000 pens, and repeated execution for five rounds under 6-node, 9-node and 12-node configurations. As shown in Fig. 7, when the number of nodes increases, the state conflict rate and audit gap rate both decrease, indicating that the multi-node confirmation and version lock mechanism can stably constrain the cross-department propagation path.

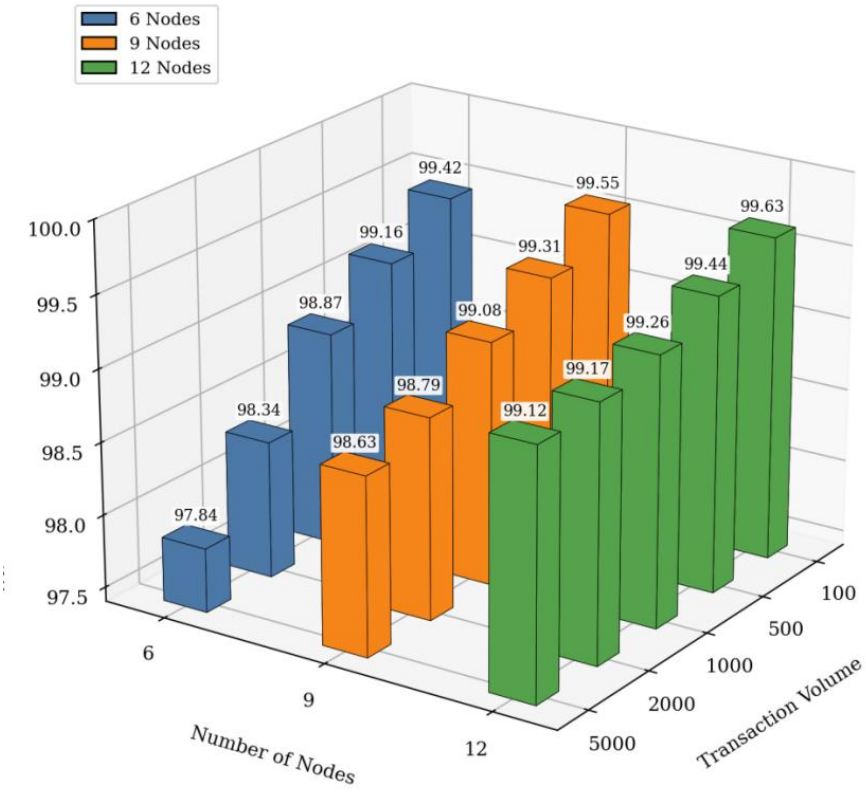

Figure 7: Verification results of state synchronization accuracy in E1 environment

In the E1 environment, the state synchronization accuracy and audit integrity of the 6-node configuration are 97.84% and 98.21%, respectively, when 5000 transactions occur. 9 nodes to 98.63% and 98.88%, respectively; 12 nodes further reach 99.12% and 99.68%. In E2 environment, the state synchronization accuracy of 12-node configuration is improved to

99.31%, and the audit integrity is improved to 99.79%.

To facilitate the comparison of the verification effects of the two types of environments under high load conditions, Table 6 summarizes the state synchronization accuracy and audit integrity results of different node configurations under 5000 transactions. It can be seen that as the number of nodes increases from 6 to 12, both indicators maintain an upward trend. In the E1 environment, the state synchronization accuracy of the 12-node configuration reaches 99.12%, and the audit integrity reaches 99.68%. E2 environment further improved to 99.31% and 99.79%. This shows that with the asynchronous audit writing and caching mechanism enabled, the cooperation between on-chain state refreshing and off-chain replica writeback is more stable, and the misses and timing offset in the cross-department confirmation link are better controlled.

Table 6: State synchronization and audit integrity verification results under high load conditions

Environment	Number of Nodes	State Synchronization Accuracy (%)	Audit Integrity (%)
E1	6	97.84	98.21
E1	9	98.63	98.88
E1	12	99.12	99.68
E2	6	98.16	98.47
E2	9	98.92	99.11
E2	12	99.31	99.79

The overall results show that the accuracy of state synchronization is not only affected by the number of nodes, but also directly related to the scheduling mode of on-chain state refresh, off-chain replica writeback and audit write. The complete system can still maintain more than 99% synchronization accuracy and nearly 99.8% audit integrity under high load scenarios, indicating that the consortium blockchain collaboration architecture constructed in this paper can maintain stable cross-department consistency under continuous submission conditions, and provide reliable experimental support for the subsequent discussion section.

4 Discussion

This study demonstrates that a blockchain-enabled distributed collaborative architecture for misdemeanor record elimination is capable of not only completing state registration, but also organizing condition verification, identity authentication, result writeback, and audit trail as a unified computing link. Compared with the centralized business platform, the multi-node confirmation mechanism on the consortium blockchain makes the cross-department status update no longer rely on a single point of database submission, and the double-layer mapping between on-chain index and off-chain copy also reduces the continuous occupation of large amount of document data on the consensus process. The experimental results show that after the node scale is extended, the system still maintains a relatively stable response ability and throughput output under high load, indicating that the implementation of on-chain control and off-chain bearing is suitable for continuous request processing in the criminal record elimination scenario. The accuracy of state synchronization and audit integrity are maintained at a high level, which indicates that version locking, asynchronous audit writing and role caching jointly enhance the temporal consistency of the execution link. The value of this system is not only to improve the processing efficiency, but also to compress the confirmation sequence, writeback source and history trace originally scattered in different departments'

systems into a unified record that can be verified, traced and reviewed. The coordination framework formed in this way provides a more stable data governance foundation for the misdemeanor criminal record elimination business, and also lays a realization path for the subsequent expansion to cross-regional judicial coordination, refined authority control and higher intensity audit verification. At the same time, the experimental results show that the system performance improvement does not come from a single module, but the result of node deployment, process joint adjustment and synchronization control. Interface and chain code scheduling reduce duplicate parsing and redundant broadcast, so that cooperative requests maintain processing capacity in the concurrent phase.

5 Conclusion

This paper adopts the research path of combining model construction, prototype implementation and experimental verification, and completes the design of a distributed data collaboration system for the misdemeanor criminal record elimination scenario. The system takes the consortium chain as the control core, integrates business object modeling, double-layer mapping on and off the chain, condition verification, identity authentication, state synchronization and audit tracking into a unified computing link, and realizes cross-department collaborative processing in the prototype environment. Experimental results show that the proposed architecture can maintain stable response time and throughput performance under continuous transaction load. In the 12-node configuration, the state synchronization accuracy reaches 99.31%, and the audit integrity reaches 99.79%, which indicates that the mechanisms such as version locking, asynchronous audit writing and role caching can better support the collaborative execution under high load conditions. The limitation at this stage is that the samples are still mainly based on simulation judicial data, and cross-regional network jitter, heterogeneous platform interface differences, and more complex authority levels have not been fully included in the scope of verification. Subsequent research can further introduce real business link data, extend the fine-grained policy control, cross-chain mutual recognition and privacy-enhanced computing mechanisms, and continue to verify the adaptability of the system in complex judicial collaborative environments combined with larger scale node deployment. The implementation of separate loading of on-chain index and off-chain copy also reduces the data pressure in the consensus process to a certain extent, so that the document summary, confirmation set and writeback results can be associated and retrieved under a unified index, while retaining a stable interface for subsequent audit review, history tracking and rule expansion. This shows that the system not only has a good foundation for engineering implementation, but also has the space for further expansion.

Funding

Concluding Achievement of the General National Project Research on the Construction of the Misdemeanor Record Expungement System in China, supported by the Heilongjiang Provincial College Students' Innovation Training Program (Project No.: 202510240096)

References

[1] Ghaffari F, Gilani K, Bertin E, et al. Identity and access management using distributed ledger technology: A survey[J]. International Journal of Network Management, 2022,

32(2): e2180.

[2] Ahmed M R, Islam A K M M, Shatabda S, et al. Blockchain-based identity management system and self-sovereign identity ecosystem: A comprehensive survey[J]. Ieee Access, 2022, 10: 113436-113481.

[3] Schardong F, Custódio R. Self-sovereign identity: a systematic review, mapping and taxonomy[J]. Sensors, 2022, 22(15): 5641.

[4] Xiang X, Cao J, Fan W. Decentralized authentication and access control protocol for blockchain-based e-health systems[J]. Journal of network and computer applications, 2022, 207: 103512.

[5] Xiang X, Zhao X. Blockchain-assisted searchable attribute-based encryption for e-health systems[J]. Journal of Systems Architecture, 2022, 124: 102417.

[6] Yang X, Zhang C. Blockchain-based multiple authorities attribute-based encryption for EHR access control scheme[J]. Applied Sciences, 2022, 12(21): 10812.

[7] Hasan O, Brunie L, Bertino E. Privacy-preserving reputation systems based on blockchain and other cryptographic building blocks: A survey[J]. ACM Computing Surveys (CSUR), 2022, 55(2): 1-37.

[8] Zou H, Liu X, Ren W, et al. A decentralized electronic reporting scheme with privacy protection based on proxy signature and blockchain[J]. Security and Communication Networks, 2022, 2022(1): 5424395.

[9] Deepa N, Pham Q V, Nguyen D C, et al. A survey on blockchain for big data: Approaches, opportunities, and future directions[J]. Future Generation Computer Systems, 2022, 131: 209-226.

[10] Singla A, Gupta N, Aeron P, et al. Decentralized identity management using blockchain: Cube framework for secure usage of IS resources[J]. Journal of Global Information Management (JGIM), 2022, 31(2): 1-24.

[11] Ghaffari F, Bertin E, Crespi N, et al. Distributed ledger technologies for authentication and access control in networking applications: A comprehensive survey[J]. Computer Science Review, 2023, 50: 100590.

[12] Hu T, Yang S, Wang Y, et al. N-accesses: A blockchain-based access control framework for secure IoT data management[J]. Sensors, 2023, 23(20): 8535.

[13] Klaine P V, Xu H, Zhang L, et al. A privacy-preserving blockchain platform for a data marketplace[J]. Distributed Ledger Technologies: Research and Practice, 2023, 2(1): 1-16.

[14] González V, Sánchez L, Lanza J, et al. On the use of Blockchain to enable a highly scalable Internet of Things Data Marketplace[J]. Internet of Things, 2023, 22: 100722.

[15] Li P, Zhou D, Ma H, et al. Flexible and secure access control for EHR sharing based on blockchain[J]. Journal of Systems Architecture, 2024, 146: 103033.

[16] Agarkar A A, Karyakarte M, Chavhan G, et al. Blockchain aware decentralized identity management and access control system[J]. Measurement: Sensors, 2024, 31: 101032.

[17] Bin Saif M, Migliorini S, Spoto F. Efficient and secure distributed data storage and retrieval using interplanetary file system and blockchain[J]. Future Internet, 2024, 16(3): 98.

[18] Deshmukh S A, Kasar S. Applications of blockchain technology in privacy preserving and data security for real time (data) applications[J]. Concurrency and Computation: Practice and Experience, 2024, 36(26): e8277.

[19] Jakhar A K, Singh M, Sharma R, et al. A blockchain-based privacy-preserving and access-control framework for electronic health records management[J]. Multimedia Tools and Applications, 2024, 83(36): 84195-84229.

[20] AlKhanafseh M, Surakhi O. Evidence preservation in digital forensics: An approach using blockchain and LSTM-based steganography[J]. Electronics, 2024, 13(18): 3729.