

Blockchain-Based Organ Transplantation: A Decentralized Framework for Secure Donor–Recipient Management Using Ethereum Smart Contracts and IPFS

Rashmi D M

Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Karnataka, India
rashmidm27@gmail.com

Jaishree N

Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Karnataka, India
jaishreenarayanamurthy@gmail.com

Kusuma M

Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Karnataka, India
kusumamanjunath57@gmail.com

Rishika Sujay

Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Karnataka, India
rishika412005@gmail.com

Yashaswini S

Dept. of Computer Science and Engineering
Don Bosco Institute of Technology
Karnataka, India
yashaswini.s345@gmail.com

Abstract— Organ transplantation is a life-saving medical procedure, yet prevailing organ donation and allocation systems remain largely centralized, manually coordinated, and susceptible to data manipulation, opacity, and inefficient donor–recipient matching. This paper presents a blockchain-based organ transplantation management framework built on the Ethereum platform that employs smart contracts to automate donor and recipient registration, enforce role-based access control, and execute organ allocation according to predefined medical criteria such as blood-group compatibility, urgency, and waiting time. To overcome the scalability limitations of storing large medical records directly on-chain, the system adopts a hybrid storage architecture that combines the InterPlanetary File System (IPFS) for off-chain document storage with on-chain cryptographic hash references for integrity verification. A React.js-based web interface provides dedicated dashboards for donors, patients, and hospital administrators, integrated with MetaMask for cryptographic identity and transaction signing. We review recent literature on blockchain-enabled organ donation platforms, AI-assisted organ matching, blockchain-type risk perceptions in health-record systems, and smart-contract security, and position the proposed architecture relative to these efforts. A working prototype demonstrating donor registration, patient waitlist entry, and hospital oversight is presented, along with a discussion of gas-efficiency, scalability, and security considerations relevant to real-world deployment. The paper concludes with directions for future work, including AI-assisted matching and multi-institutional interoperability.

Keywords— Blockchain, Ethereum, Smart Contracts, Solidity, Organ Transplantation, IPFS, Decentralized Application, Healthcare Security

I. INTRODUCTION

Organ transplantation is among the most consequential achievements of modern medicine, offering a second chance at life to patients suffering irreversible organ failure. Despite this promise, the imbalance between organ supply and demand remains severe: national allocation statistics show that recipients on waiting lists consistently outnumber available donors by a wide margin, with kidneys accounting for the overwhelming majority of unmet need [5]. Every day, patients

are lost simply because a compatible organ could not be located or delivered in time [10].

A major contributor to this shortfall is not merely donor scarcity but the inefficiency, opacity, and fragility of the systems used to coordinate donation and allocation. Conventional platforms rely on centralized databases and largely manual verification, which creates single points of failure, obscures the fairness of allocation decisions, and leaves records vulnerable to tampering or fraudulent modification [1], [2], [4]. Sensitive medical information is often duplicated across disconnected hospital systems, complicating both privacy protection and cross-institution coordination.

Blockchain technology offers a structurally different foundation for this problem. Its decentralized ledger removes reliance on a single trusted authority, its cryptographic linking of blocks makes historical records effectively tamper-evident, and its transparency allows every authorized participant to verify the state of the system independently [9]. When combined with smart contracts—self-executing code deployed on platforms such as Ethereum—these properties can be used to automate donor and recipient registration, enforce eligibility rules, and carry out organ matching without discretionary human intervention at each step [1], [2].

This paper builds on these ideas to propose a blockchain-based organ transplantation management system. The main contributions are as follows:

- A hybrid on-chain/off-chain architecture that stores medical documents on IPFS while anchoring their cryptographic hash on the Ethereum blockchain, balancing immutability with practical storage cost.
- A smart-contract-enforced matching workflow that scores donor–recipient pairs on blood-group compatibility, urgency, and waiting time, removing discretionary bias from the allocation decision.
- Role-based dashboards for donors, patients, and hospital administrators built with React.js and integrated with MetaMask for cryptographically verifiable identity.

- A comparative review of existing blockchain-based organ donation and healthcare-record systems, positioning the proposed design against reported strengths, limitations, and security considerations.

II. LITERATURE SURVEY

A. Blockchain-Based Organ Donation Platforms

Several recent efforts have applied Ethereum-based decentralized applications directly to organ donation. Kypu et al. proposed a dApp built with Solidity smart contracts and a Node.js/JavaScript frontend that supports donor registration, pledge verification, and transplant matching, emphasizing accountable, tamper-resistant handling of donor commitments [1]. Divya Priya et al. proposed a permissioned-blockchain platform in which smart contracts allocate organs using a fuzzy-matching algorithm based on Levenshtein edit-distance to score donor–recipient attribute similarity, with donor and recipient identities verified through Aadhaar authentication before any record is admitted to the ledger; their evaluation reported that the approach could allocate organs to roughly 30% more patients than comparable Hyperledger, Ripple, and machine-learning baselines at similar computational cost [2]. Shanmugam et al. presented SecureOrganChain, an Ethereum Virtual Machine framework that couples Electronic Health Record data with smart-contract-governed donation and allocation, deployed and performance-tested on the Ganache test network with an administrative dashboard for matching, record release, and access control [11]. Srilatha et al. presented Organ Connect, a platform that generates a SHA-256/512 hash for each donation record so that hospitals can later verify data integrity, though the underlying application logic in that system runs on a conventional Java/MySQL stack rather than being fully executed as on-chain smart contracts [10]. Kumar et al. proposed a private, Proof-of-Authority Ethereum platform with an urgency-weighted compatibility score $S = w1C + w2U + w3T$ combining medical compatibility, urgency, and waiting time, reporting a 30% improvement in data integrity and a 40% reduction in required human intervention relative to manual allocation workflows [15]. Taken together, these systems demonstrate that blockchain-backed registration and verification is now a well-established pattern, though most implementations remain prototypes evaluated on local or test networks rather than production hospital infrastructure.

B. AI-Assisted Organ Matching

A parallel line of work augments blockchain security with machine-learning-based matching. Kumar and Jeysankar proposed a framework that pairs a machine-learning matching model, trained on donor and recipient attributes such as medical history and geographic proximity, with a blockchain layer that records transactions on an immutable ledger; their reported training curves show the matching model converging to high accuracy within roughly fifty epochs, alongside a hash-based mechanism for detecting any later tampering of registration data

[3]. Senthil Pandi et al. proposed a Random Forest classifier for donor–recipient matching combined with a Merkle-tree-secured blockchain layer, reporting accuracy, precision, and F1-scores around 90% against traditional-ML, logistic-regression, and decision-tree baselines, while noting that Random Forest remains computationally intensive and less interpretable than the rule-based scoring used in the present work [12]. Gonuguntla et al. take a related but non-ML approach, computing a Calculated Panel Reactive Antibody (CPRA) score alongside an Organ Transplantation and Procurement Network (OPTN)-style compatibility metric that combines blood type, tissue match, organ size, urgency, and geographic distance, and demonstrate on real OPTN-derived patient/donor records that lower CPRA scores correlate strongly with higher OPTN match scores and hence better compatibility [13]. Such hybrid AI–blockchain designs, and CPRA-style immunological scoring in particular, promise more clinically nuanced matching than the fixed rule set used in the present work, but they also introduce a dependence on large, high-quality training or antigen-frequency datasets and additional infrastructure for model training and monitoring, which the deterministic, rule-based smart-contract matching adopted here deliberately avoids in favor of auditability; Section V identifies CPRA-weighted scoring as a concrete direction for extending Algorithm 1.

C. Broader Healthcare Blockchain Applications

Beyond organ donation specifically, blockchain has been proposed widely for electronic health record (EHR) management. Nekkallapudi et al. reviewed blockchain-based healthcare management architectures and proposed a system model combining smart-contract-mediated access control with compliance considerations for regulations such as HIPAA and GDPR, concluding that while adoption is still early-stage, the technology can meaningfully reduce data siloing and unauthorized access [4]. Singh et al. implemented and benchmarked a decentralized healthcare-records system using a single Solidity contract with owner/authorized-provider role modifiers deployed on a local Hardhat network, reporting concrete gas costs—approximately 169,546 gas for contract deployment, 65,000 gas per patient record insertion, and 45,000 gas for provider authorization—along with near-instantaneous, gas-free read access for authorized parties [8]. Sharma et al. conducted a PRISMA-guided bibliometric analysis of blockchain-and-healthcare literature from 2016–2024 and found healthcare and IoT to be the most active application areas, with remote patient monitoring emerging as a particularly prominent use case, while also flagging scalability and interoperability as persistent open challenges [9]. Ballal et al. implemented a permissioned-blockchain EHR platform that stores medical records on IPFS while smart contracts manage patient-granted, patient-revocable consent and doctor access; a systematic vulnerability analysis of their access-control contract found 16 issues across 106 lines of code (an overall score of 3.87/5), underscoring that even consent-management logic significantly simpler than full organ-matching contracts warrants formal

security review before deployment [14]. These studies collectively motivate the layered, gas-conscious architecture adopted in this paper and provide a useful performance baseline for evaluating on-chain organ-registration operations.

D. Blockchain Type Selection and Smart Contract Security

Choosing an appropriate blockchain platform is itself a risk-management decision. Said et al. surveyed blockchain specialists and healthcare professionals on the perceived technical, security, operational, privacy, and regulatory risks of public, private, hybrid, and consortium blockchain types for patient health-record systems, finding that public, permissionless platforms were consistently rated as carrying higher perceived risk across nearly all categories, whereas private and consortium platforms—despite some loss of decentralization—were perceived as offering a better risk profile for regulated health data [6]. This supports the view that a consortium or permissioned Ethereum-compatible network, rather than the public Ethereum mainnet, is the more appropriate deployment target for a production organ-transplantation system. Complementing this platform-level analysis, Le et al. fine-tuned eight pre-trained transformer encoder models to automatically detect vulnerability classes—including reentrancy, unchecked low-level calls, access-control flaws, and arithmetic errors—in Solidity smart contracts, reporting F1 scores above 0.97 for the best-performing model on each of three benchmark datasets after addressing class imbalance through random over-sampling [7]. Their results indicate that automated, learning-based auditing tools are becoming mature enough to be incorporated into the smart-contract development lifecycle before deployment, a practice this paper recommends for the matching and record-storage contracts described in Section III. Shrihari et al. reach a related conclusion from the platform side: comparing an Ethereum Proof-of-Work deployment against a Hyperledger Fabric network built on Amazon Managed Blockchain for the same organ-donation workflow, they report that Ethereum's constrained governance and lack of support for concurrent execution motivated migrating to Fabric, which offered a Raft consensus mechanism, permissioned governance, and no single point of failure at the cost of losing Ethereum's broader ecosystem interoperability [16]. This reinforces the case, developed above from risk-perception and consensus-mechanism evidence [6], [9], that a permissioned or consortium platform—Fabric, Besu, or Quorum—rather than public Ethereum, is the appropriate production target for the system proposed here.

E. Organ Allocation Algorithms

Independently of blockchain, national organ-allocation bodies already operate formalized scoring algorithms. Niyigena et al. surveyed the Kidney Allocation System (KAS) used by the United Network for Organ Sharing and the Eurotransplant Kidney Allocation System (ETKAS), both of which combine compatibility rules with point-scoring based on urgency, waiting time, and donor–recipient similarity, and reviewed early

blockchain-based organ-donation prototypes including Hyperledger-based OrganChain and IPFS-integrated Ethereum systems that reduce the cost of storing donor and patient documents [5]. The survey concludes that while individual allocation algorithms differ by country and organization, a blockchain-encoded, internationally agreed scoring smart contract could in principle unify these processes and support cross-border organ exchange, a long-term direction this paper's matching-logic module is designed to be extensible toward.

TABLE I. COMPARATIVE SUMMARY OF REVIEWED SYSTEMS

Study	Platform	Storage	Matching	Limitation
Kypu [1]	Ethereum	On-chain	Manual pledge	No auto scoring
Divya [2]	Perm. Ethereum	On-chain, enc.	Param. contract	Needs deploy exp.
Organ Connect [10]	Hash-anchored	DB + hash	First-come	Not full on-chain
Kumar [3]	Ethereum	On-chain hash	ML model	Needs large data
Singh [8]	Ethereum	On-chain proto.	N/A (EHR)	Bulk data off-chain
Shanmugam [11]	Ethereum (Ganache)	EHR + on-chain	Admin-driven	Centralized wallet/backend
Gonuguntla [13]	Ethereum + Mongo	Hybrid on/off-chain	CPRA + OPTN score	MongoDB retains PII
Shrihari [16]	Hyperledger Fabric	On-chain, Raft	Contract-based	Loses Ethereum ecosystem
Proposed	Ethereum	IPFS + hash	Rule-based	Phase-1 prototype

F. Blockchain Fundamentals Relevant to Platform Selection

Selecting an appropriate underlying blockchain requires understanding its layered structure. Sharma et al. describe blockchain systems as composed of a data layer (blocks linked via cryptographic hashes and organized through a Merkle tree for integrity), a network layer (peer-to-peer propagation and replication of the ledger), a consensus layer (algorithms such as Proof of Work, Proof of Stake, Delegated Proof of Stake, and Practical Byzantine Fault Tolerance that agree on the valid next block), and an application layer that exposes smart contracts and end-user services [9]. Their comparative analysis of consensus mechanisms indicates that Proof of Work offers strong security and decentralization but at very high energy cost and limited scalability, whereas Practical Byzantine Fault Tolerance and Proof of Authority achieve much lower energy consumption and higher throughput at the cost of relying on a fixed or authority-vetted set of validating nodes [9]. This trade-off is directly relevant to organ-transplantation deployments, where the validator set can reasonably be restricted to accredited hospitals and transplant authorities, making PBFT- or PoA-style consensus a more suitable choice than energy-intensive Proof of Work.

III. PROPOSED ARCHITECTURE

The proposed system is organized as a three-layer architecture consisting of a user-facing dashboard layer, an application/API layer, and a blockchain-and-storage layer, following the layered pattern also adopted for decentralized EHR systems [8], [9]. Fig. 1 illustrates this arrangement.

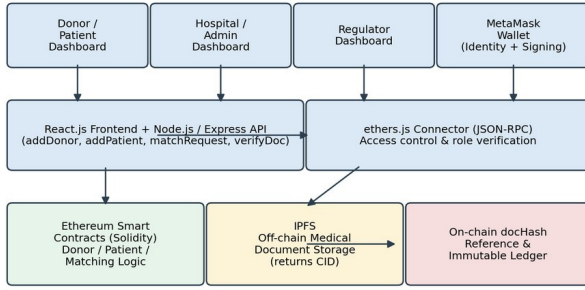


Fig. 1. Proposed three-layer system architecture.

At the top layer, donors and patients interact with dedicated React.js dashboards, while hospital administrators and regulators access oversight dashboards for verification and monitoring. All identity and transaction-signing operations are delegated to MetaMask, so that private keys never pass through the application server, consistent with the wallet-integration pattern used in comparable Ethereum health-record prototypes [8].

The application layer exposes a Node.js/Express API that translates dashboard actions—donor registration, patient waitlist entry, match requests, and document verification—into smart-contract calls via an ethers.js/JSON-RPC connector. This layer also performs client-side input sanitation before any transaction is submitted, reducing malformed data from reaching the contract layer.

The blockchain-and-storage layer consists of the Ethereum network hosting the core Solidity smart contracts together with IPFS for off-chain document storage. Medical reports and supporting documents are uploaded to IPFS, which returns a content identifier (CID); only this CID, referred to in the contract as docHash, is written on-chain, giving the system verifiable integrity without incurring the gas cost of storing full documents on the ledger. This hybrid design mirrors the IPFS-based cost-reduction strategy identified in prior blockchain organ-donation surveys [5] and the general off-chain/on-chain split recommended for scalable healthcare blockchain systems [4], [9].

Four functional modules implement the core business logic, following the module breakdown established in the project's Phase-1 design:

- 1) Donor Onboarding Module — collects donor demographic and medical details, validates registration data, and stores records securely on-chain.
- 2) Recipient Waitlist Module — maintains patient profiles, urgency levels, and organ requirements, and updates waitlist status dynamically.
- 3) Matching Logic Module — executes donor–recipient matching based on blood-group compatibility, organ type, medical urgency, and waiting time, and generates allocation results through smart-contract functions.

4) Storage and Security Module — uploads documents to IPFS, anchors the resulting hash on-chain, and exposes verification and oversight views to hospitals and regulators.

The smart-contract layer is proposed to be written in Solidity (^0.8.0) and organized around two primary data structures, mirroring the compact struct-based design used in comparable healthcare smart contracts [8]:

```
struct Donor { string name; string bloodGroup; string organ;
string docHash; bool isVerified; bool isMatched; } struct
Patient { string name; uint256 urgency; string bloodGroup;
string organ; uint256 waitStart; }
```

Access to state-changing functions is restricted using function modifiers analogous to the `onlyOwner` / `onlyAuthorizedProvider` pattern reported for Ethereum-based healthcare-record contracts [8]; in this system, an `onlyHospital` modifier permits only whitelisted hospital addresses to verify donors, update match status, or authorize new provider accounts, while view functions such as `getDonorList` and `getPatientList` remain freely callable and gas-free for read access.

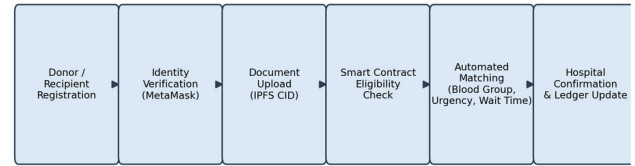


Fig. 2. End-to-end donor–recipient registration and matching workflow.

Fig. 2 summarizes the operational workflow: a donor or recipient first registers and authenticates through MetaMask; supporting documents are uploaded to IPFS and the returned CID is written on-chain; the smart contract then checks eligibility criteria before the matching function ranks candidate pairs by blood-group compatibility, urgency, and waiting time; finally, the hospital confirms the match, and the ledger is updated with an immutable, timestamped record of the decision.

Security was treated as a first-class design concern rather than an afterthought. Reentrancy, unchecked low-level calls, integer arithmetic errors, and access-control misconfiguration were identified in the literature as the vulnerability classes most frequently affecting Solidity contracts, and automated transformer-based classifiers have been shown to detect these classes with high precision and recall once trained on curated, class-balanced datasets [7]. Accordingly, the matching and record-storage contracts in this system were reviewed against this vulnerability taxonomy, and static analysis is recommended as a mandatory step prior to any mainnet or consortium-network deployment. Gas efficiency was treated alongside correctness: Raza et al. analyzed 397 contracts from 53 Ethereum dApps and found that every dApp exhibited at least one of four common gas-waste patterns—uncached repeated use of `address(this)`,

division by unsigned integers without an unchecked block, constant strings under 32 characters left as the dynamic string type instead of bytes32, and redundant same-type casting—with division-by-uint alone accounting for over half of the 250 detected instances [17]. The Donor and Patient structs and matching functions in this system were checked against this same pattern set, and the recommendation is carried forward that any subsequent implementation phase run an equivalent static analyzer (e.g., an extended Slither configuration) before deployment, since the paper's dataset shows these waste patterns are pervasive even in previously audited, live dApps. In line with the risk-perception findings of Said et al., the system is designed to be deployable on a permissioned or consortium Ethereum-compatible network (e.g., Hyperledger Besu or Quorum) rather than the public mainnet, since permissioned platforms were rated lower-risk across technical, security, and regulatory dimensions for patient-record use cases [6].

Algorithm 1 summarizes the matching logic executed once a hospital initiates a match check. The function first filters candidate donors by blood-group compatibility with a given patient, then ranks the remaining candidates by a weighted combination of urgency and waiting time before returning the highest-priority match for hospital confirmation.

Algorithm 1: Donor-Recipient Matching

```
Input: patient P, donor list D
1: candidates ← { d ∈ D : d.organ = P.organ ∧
compatible(d.bloodGroup, P.bloodGroup) ∧ d.isVerified ∧
¬d.isMatched }
2: for each c in candidates do
3:   score(c) ← w1·P.urgency + w2·(now - P.waitStart)
4: end for
5: bestMatch ← argmax(score) over candidates
6: mark bestMatch.isMatched ← true; emit MatchFound
event
7: return bestMatch for hospital confirmation
```

Because steps 1–6 execute entirely within the smart contract, the outcome cannot be altered after the fact without leaving a detectable trace on the ledger, and the weights w_1 and w_2 are stored as contract parameters that can only be changed by a governance-restricted function, preventing silent manipulation of allocation priorities.

Regarding consensus and deployment, the discussion in Section II-F suggests that a Practical Byzantine Fault Tolerance or Proof-of-Authority network—rather than public Proof-of-Work Ethereum—is the more appropriate target for the smart contracts described above, since the validator set can be restricted to accredited hospitals and regulators while still preserving the tamper-evidence and multi-party verification that make blockchain attractive for this domain [6], [9].

IV. EXPECTED RESULTS AND DISCUSSION

A. Intended Prototype Validation

A prototype is proposed to validate the core registration and oversight workflows described above. The donor registration form is designed to capture full name, blood group, organ for donation, and a reference to the uploaded medical report, encrypting data before it is committed on-chain. The patient waitlist entry form is intended to record the patient's name, required organ, blood group, and a priority status (e.g., Critical), which would feed directly into the urgency field consumed by the matching logic. A hospital oversight dashboard is proposed to aggregate system-wide counts—verified donors and active requests—and to expose an Initiate Match Check control; in the planned Phase-1 build this would trigger the smart contract's internal matching logic, with detailed step-by-step matching results scheduled for the subsequent implementation phase.

B. Cost, Scalability, and Comparison with Related Approaches

Because the system has not yet been benchmarked on a live testnet, Table II situates its expected on-chain cost profile against gas figures reported for a comparable single-contract, role-based Ethereum health-record system [8]. Contract deployment and per-record writes are the most gas-intensive operations, while all read-only queries (e.g., retrieving a donor or patient list) are free view calls, consistent with the general pattern that state-changing transactions dominate the cost of blockchain-based record systems.

TABLE II. REFERENCE GAS-COST PROFILE [8]

Function	Gas Cost [8]	Nature
Deployment	~169,546	One-time
Add record	~65,000	Scales linearly
Authorize hospital	~45,000	Infrequent
Read list	0 (view)	Free, low-latency

This cost profile has two practical implications for the proposed system. First, storing only the IPFS hash on-chain (rather than full documents) keeps the per-record write cost close to the reported ~65,000 gas benchmark rather than the substantially higher cost of storing large medical files directly in contract storage. Second, because read operations are free, hospital dashboards can query donor and patient status frequently without imposing recurring transaction costs on the institution, supporting the low-latency oversight experience targeted for the proposed prototype.

Compared with the AI-assisted matching approach of Kumar and Jeysankar, which reported strong training accuracy but depends on a sizeable labeled dataset of prior transplant outcomes [3], the rule-based matching logic adopted here trades some potential predictive refinement for full auditability: every allocation decision can be traced to explicit, on-chain criteria (blood-group compatibility, urgency, waiting time), which aligns with the transparency goals emphasized throughout the reviewed organ-donation literature [1], [2], [5], [10]. This also mirrors the rationale behind national allocation algorithms such as KAS and ETKAS, which likewise rely on explicit, auditable-scoring rules rather than opaque predictive models [5].

Nonetheless, the matching module is designed so that a future machine-learning scoring function could be substituted or layered on top of the deterministic rules without altering the surrounding registration and storage architecture.

C. Limitations

The proposed design is subject to several anticipated limitations that also define priorities for subsequent phases. First, the prototype is intended to be exercised initially on a local development network rather than a multi-node consortium testbed, so the gas and latency figures discussed in Table II are drawn from a comparable published system [8] rather than measured directly; direct benchmarking is planned for Phase 2. Second, the matching logic currently proposes a fixed weighting of urgency and waiting time; while this preserves auditability, it would still need to be validated against real transplant-outcome data of the kind used to train machine-learning matching models [3]. Third, as with all blockchain-based health-record designs, regulatory compliance (e.g., data-protection and cross-border health-data rules) must be established jointly with legal and clinical stakeholders before any pilot deployment, a challenge repeatedly noted across the surveyed literature [4], [6], [9].

V. CONCLUSION AND FUTURE SCOPE

This paper proposed a blockchain-based organ transplantation management system that combines Ethereum smart contracts, IPFS-based hybrid storage, and a React.js/MetaMask frontend to address the transparency, security, and coordination shortcomings of conventional organ donation platforms. By reviewing seventeen recent studies spanning blockchain-based organ donation platforms, AI- and Random-Forest-assisted matching, CPRA/OPTN-based immunological scoring, broader healthcare-blockchain and EHR applications, blockchain-type risk analysis, Ethereum-versus-Hyperledger platform trade-offs, smart-contract vulnerability detection, and gas-cost optimization, the proposed system's architecture was positioned against a representative slice of the current literature, and a Phase-1 prototype covering donor registration, patient waitlist entry, and hospital oversight was outlined for planned implementation.

Several directions are planned for future work:

- Integrating a machine-learning-based matching layer, following approaches such as that of Kumar and Jeysankar, could complement the current rule-based scoring once sufficient de-identified training data becomes available [3].
- Migrating the deployment target from a local test network to a consortium or permissioned Ethereum-compatible network is recommended in light of the lower perceived regulatory and technical risk of such platforms for patient-record use cases [6].
- Subjecting the matching and storage contracts to automated, transformer-based vulnerability scanning prior to production deployment would align the project with current best practice in smart-contract security assurance [7].

- Extending the matching-logic module toward the internationally harmonized, points-based allocation schemes surveyed by Niyigena et al. would move the system closer to supporting cross-institution and potentially cross-border organ exchange [5].
- Replacing the current boolean blood-group check with a CPRA-weighted immunological score, following Gonuguntla et al.'s OPTN-metric formulation, would let the matching function distinguish "very high" from merely "high" compatibility pairs rather than treating all blood-group matches as equivalent [13].
- Given the Fabric-based prototype reported by Shrihari et al., a formal cost-benefit comparison between a Solidity/Ethereum deployment and a Hyperledger Fabric migration is warranted before selecting the Phase-2 production platform [16].
- Integrating the gas-waste detectors validated by Raza et al. into the project's CI pipeline would let every future contract revision be automatically screened for the address(this)-caching, division-by-uint, short-string, and redundant-casting patterns shown to affect the large majority of live Ethereum dApps [17].

REFERENCES

- [1] N. R. Kypu, B. K. D. V. Korla, V. Dwarakacharla, and S. V. K, "Blockchain-based Organ Donation System: A Decentralized Solution for Transparent and Secure Transplant Matching," in Proc. Int. Conf. Sustainable Communication Networks and Application (ICSCNA), 2024, pp. 439–443, doi: 10.1109/ICSCNA63714.2024.10864115.
- [2] D. Divya Priya, M. Naga Sreeya, A. Sanjana, P. Sarayu, and M. Kista Swamy, "Revolutionizing Organ Allocation: A Blockchain Approach to Transparency and Security," in Proc. Int. Conf. Advancements in Smart, Secure and Intelligent Computing (ASSIC), 2025, doi: 10.1109/ASSIC64892.2025.11158574.
- [3] Kumar P and Jeysankar N, "Synergistic Revolution with AI-Enabled Organ Matching and Blockchain Security in Transplantation Management," in Proc. 2nd Int. Conf. Advances in Information Technology (ICAIT), 2024, doi: 10.1109/ICAIT61638.2024.10690290.
- [4] S. Nekkalapudi, R. K. Mishra, and M. Pandey, "Blockchain-Based Healthcare Outcome Improvement: A Decentralized Method of Patient Care," in Proc. First Int. Conf. Data, Computation and Communication (ICDCC), 2024, pp. 546–550, doi: 10.1109/ICDCC62744.2024.10961792.
- [5] C. Niyigena, S. Seol, and A. Lenskiy, "Survey on Organ Allocation Algorithms and Blockchain-based Systems for Organ Donation and Transplantation," in Proc. Int. Conf. Information and Communication Technology Convergence (ICTC), 2020, pp. 173–178, doi: 10.1109/ICTC49870.2020.9289421.
- [6] H. E. Said, N. B. Al Barghuthi, S. M. Badi, F. Hashim, and S. Girija, "Blockchain Types in Patient Health Records Systems: An Analysis of Risk Perceptions," in Proc. 6th Int. Conf. Blockchain Computing and Applications (BCCA), 2024, pp. 340–346, doi: 10.1109/BCCA62388.2024.10844462.
- [7] T.-T.-H. Le, J. Kim, S. Lee, and H. Kim, "Robust Vulnerability Detection in Solidity-Based Ethereum Smart Contracts Using Fine-Tuned Transformer Encoder Models," IEEE Access, vol. 12, pp. 154700–154717, 2024, doi: 10.1109/ACCESS.2024.3482389.
- [8] P. Singh, V. K. Kushwaha, S. Gupta, B. Sharan, and M. K. Singh, "A Decentralized Healthcare Records Management System built with Solidity

- Smart Contracts and React," in Proc. Int. Conf. Intelligent Computing, Networks, and Security (IC-ICNS), 2026, doi: 10.1109/ICICNS68863.2026.11537788.
- [9] K. Sharma, N. Kumar, and R. K. Kaushal, "Blockchain in Healthcare: Pioneering Transformative Approaches," in Proc. IEEE 11th Uttar Pradesh Section Int. Conf. Electrical, Electronics and Computer Engineering (UPCON), 2024, doi: 10.1109/UPCON62832.2024.10983842.
- [10] P. Srilatha, K. Siri Reddy, A. Sruthi Reddy, D. Kedhar, and M. Bhavana, "Organ Connect: A Digital Platform for Organ Donation and Transplantation," in Proc. 3rd Int. Conf. Innovative Mechanisms for Industry Applications (ICIMIA), 2023, pp. 853–860, doi: 10.1109/ICIMIA60377.2023.10426416.
- [11] S. Shanmugam, P. Kumar, and K. Kedia, "Secure Organ Chain: A Blockchain-Based Framework for Organ Donation," in Proc. Int. Conf. Smart Systems for Electrical, Electronics, Communication and Computer Engineering (ICSSECC), 2024, pp. 289–293, doi: 10.1109/ICSSECC61126.2024.10649435.
- [12] Senthil Pandi S, Kumar P, Dhanasekaran S, and Murugan B S, "Secured Model for Organ Donation and Transplantation Management Using Blockchain and Random Forest Algorithm," in Proc. Int. Conf. Emerging Trends in Industry 4.0 Technologies (ICETI4T), 2025, doi: 10.1109/ICETI4T63625.2025.11132184.
- [13] S. Gonuguntla, M. A. V. Reddy, V. Sridhar, C. Saketh, and N. M. R., "A Decentralized Organ Donation Registry Enabling Transparency in Medical Field," in Proc. 15th Int. Conf. Computing Communication and Networking Technologies (ICCCNT), 2024, doi: 10.1109/ICCCNT61001.2024.10724871.
- [14] S. Ballal, B. Sonare, Y. Chandre, S. Patil, and R. Pise, "Blockchain-Based Decentralized Platform for Electronic Health Records Management," in Proc. IEEE Int. Conf. Blockchain and Distributed Systems Security (ICBDS), 2023, doi: 10.1109/ICBDS58040.2023.10346392.
- [15] P. V. A. S. Kumar, V. I. Reddy, M. Badi, I. Rai, S. N. Suraj, G. R. Naidu, and P. Ashok, "Transforming Organ Donation with Blockchain Technology: A Secure and Transparent Approach," in Proc. Int. Conf. Next-Gen Quantum and Advanced Computing (NQComp), 2026, pp. 736–742, doi: 10.1109/NQComp68334.2026.11497647.
- [16] Shrihari M R, Ajay N, Gaurav Singh, and Keerthi Reddy, "Organ Donation and Transplantation Framework using Blockchain," in Proc. Int. Conf. Knowledge Engineering and Communication Systems (ICKECS), 2024, doi: 10.1109/ICKECS61492.2024.10616806.
- [17] A. R. Raza, M. Hanzala, K. Mahmood, S. Tahir, S. Khalid, and A. Masood, "Optimizing Gas Consumption in Ethereum Smart Contracts: A Static Analysis Approach," in Proc. Int. Conf. Communication Technologies (ComTech), 2025, doi: 10.1109/ComTech65062.2025.11034442.