Received April 26, 2022, accepted May 31, 2022, date of publication June 3, 2022, date of current version June 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3180008

Blockchain-Based Management for Organ Donation and Transplantation

DIANA HAWASHIN[1], RAJA JAYARAMAN[1], KHALED SALAH[2], (Senior Member, IEEE), IBRAR YAQOOB[2], (Senior Member, IEEE), MECIT CAN EMRE SIMSEKLER[1], AND SAMER ELLAHHAM[3]

[1]Department of Industrial and Systems Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[3]Heart and Vascular Institute, Cleveland Clinic, Abu Dhabi, United Arab Emirates

Corresponding author: Ibrar Yaqoob (ibrar.yaqoob@ku.ac.ae)

This work was supported by the Khalifa University of Science and Technology under Award CIRA-2019-001.

ABSTRACT Today's organ donation and transplantation systems pose different requirements and challenges in terms of registration, donor-recipient matching, organ removal, organ delivery, and transplantation with legal, clinical, ethical, and technical constraints. Therefore, an end-to-end organ donation and transplantation system is required to guarantee a fair and efficient process to enhance patient experience and trust. In this paper, we propose a private Ethereum blockchain-based solution to enable organ donation and transplantation management in a manner that is fully decentralized, secure, traceable, auditable, private, and trustworthy. We develop smart contracts and present six algorithms along with their implementation, testing, and validation details. We evaluate the performance of the proposed solution by performing privacy, security, and confidentiality analyses as well as comparing our solution with the existing solutions. We make the smart contract code publicly available on Github.

INDEX TERMS Blockchain, ethereum, organ donation, organ transplantation, smart contracts, traceability.

I. INTRODUCTION

Organ failure or damage occurs due to an injury or a disease. It affects the quality of life and, in some cases, leads to death. Donating an organ is one of humanity's most honorable actions to save the lives of patients through organ transplantation. For a successful transplant, the organ must be in acceptable working conditions with donor-recipient matching, and its removal should not pose a life-threatening risk to the donor [1]. The first successful organ donation occurred with a kidney transplant between twin brothers in 1954 [2]. Since then, the annual number of transplants has steadily increased. However, the demand for organ donations still exceeds the number of donors [3]. In fact, while waiting for an organ transplant, twenty people die every day, and a new patient is added to the waiting list in every ten minutes [4]. More importantly, accessing the organ donation waiting list is a basic requirement for organ allocation. Referral for transplantation can be affected by both geographical and socioeconomic factors. Therefore, the allocation process on the waiting list should not discriminate against certain groups of patients [4].

Organ donation is conducted in two different ways, including deceased donation and living donation. Figure 1 illustrates the typical flow chart for donating an organ and transplanting it to a patient. First, the donor is examined by the hospital transplant team, and if the donor is deceased, a brain death test is performed. Meanwhile, if the donor is still alive, doctors examine the donor and ensure that the donor is fit for live donation. Then, all medical records are reported to the procurement organizer. The procurement organizer is responsible for evaluating the donor's condition to decide if he is a fit donor and ensuring that the donor is properly registered in the medical system. Next, if the evaluation shows that the donor is eligible for donation, the procurement organizer sends all the data to the organ transplantation organizer. This step can be performed only if the donor gives consent to donate to an anonymous person. After that, the matching process between the available donors and patients on the waiting list is performed by the organ transplantation organizer. As a result, a ranked list is generated as an output and provided to the transplantation surgeons. Next, the transplant surgeon decides whether the organ is appropriate for the patient based on various considerations, such as the donor's medical records and the current health of the

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

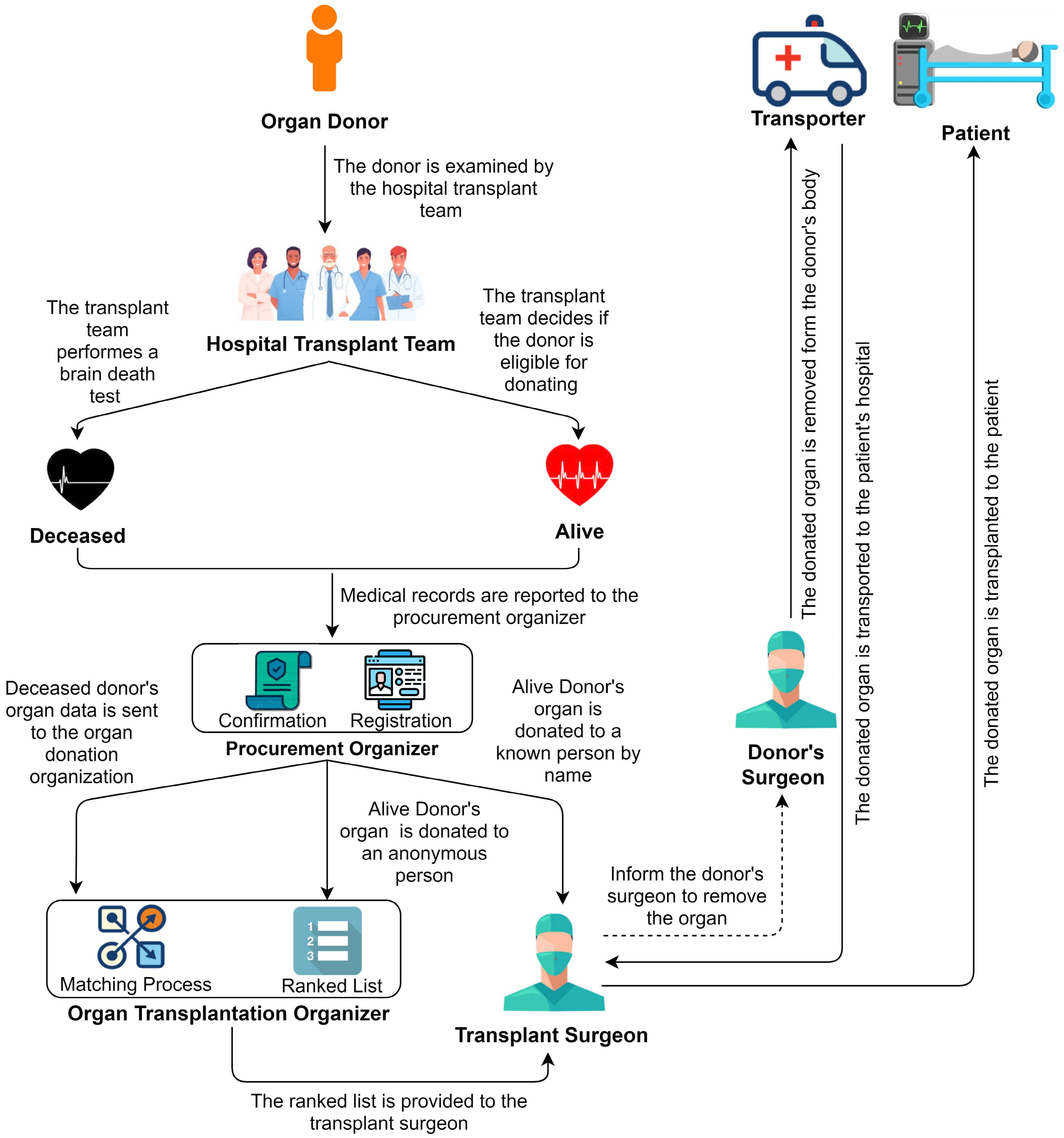

Organ Donor
The donor is examined by the hospital transplant team
Hospital Transplant Team
The transplant team performes a brain death test
The transplant team decides if the donor is eligible for donating
Deceased
Alive
Medical records are reported to the procurement organizer
Confirmation
Registration
Procurement Organizer
Deceased donor's organ data is sent to the organ donation organization
Alive Donor's organ is donated to a known person by name
Alive Donor's organ is donated to an anonymous person
Matching Process
Ranked List
1
2
3
Organ Transplantation Organizer
Transplant Surgeon
The ranked list is provided to the transplant surgeon
Inform the donor's surgeon to remove the organ
Donor's Surgeon
The donated organ is removed form the donor's body
Transporter
The donated organ is transported to the patient's hospital
Patient
The donated organ is transplanted to the patient

FIGURE 1. Organ donation and transplantation flow chart.

prospective recipient. Later, when a transplant surgeon accepts the donated organ, the donor's surgeon is notified to remove the donated organ. Finally, the donated organ is transported to the patient's hospital and received by the transplant surgeon. However, suppose the situation is for a live donor and it has been planned to donate to a known person by name. In that case, the data will go directly to the transplant surgeon to start the surgery of removing and transplanting the donated organ [6], [7].

In the past, when a patient died or was near death, the organ procurement organization and hospital worked together to do an initial medical test to decide if the patient could be an organ donor. This call takes around 15 minutes, and only 6% of these calls result in possible organ donors' being identified. Over the years, this phone call has been replaced by an instant message generated by central computer systems that store all the data required for this process [8]. However, the core issue with this strategy is that the security and validity of such data are entirely dependent on the transplantation centers' ability to keep their systems secure and identify potential harm to donors and recipients. The accuracy of the wait-list data is largely dependent on people's faith and trust in these centers' ability to keep it secure from hackers and fraudulent employees [9]. Moreover, transparency is another challenge affecting the success of the organ donation process. According to World Health Organization (WHO) reports, up to 10% of transplanted organs may have been obtained unethically via organ trafficking, but the exact numbers are unknown [10]. The lack of transparency in the current system among participants leads to illegal organ trade and purchases and medical professionals engaging in unethical practices [11]. Moreover, there are hospitals that take advantage of the patient's need for the organ and offer the opportunity to transfer the organ to those who pay a higher amount to the hospital while ignoring the patient with the highest priority on the waiting list [12], [13]. In addition, current transplant systems are also frequently slow, which is unacceptable in such a critical and life-threatening scenario. Such systems are hardly up to date with the minimum security standards. So far, there has recently been a surge in security breaches affecting user privacy and system integrity. In general, modern systems manage data through the use of standard databases; however, most hospitals, health ministries, and other medical facilities lack a standardized data communication system [1].

In recent years, blockchain technology has attracted much attention in different sectors because it offers a distributed and secure database without the need for a third party or a central authority. Blockchain is most known for its use in cryptocurrencies. A significant portion of its development is focused on information architecture, or how the database will be formed, distributed, and accessed with various degrees of permissions [14]. Nakamoto creates the first blockchain, which serves as the public ledger for bitcoin transactions. Later, the Ethereum blockchain architecture inserts computer programs into blocks to represent financial instruments, which are known today as smart contrats [15]. However, the purpose of blockchain is to enable the recording and distribution of digital data without the ability to modify it. In this sense, a blockchain serves as the foundation for immutable ledgers that cannot be changed or destroyed. By using blockchain, medical information may be stored securely, and patient data could be updated in real-time and across various entities [16].

Managing organ donation and transplantation has become challenging due to the lack of data accountability, immutability, audit, transparency, traceability, and trust features in the existing systems. The following are the paper's main contributions:

- We propose a private Ethereum blockchain-based solution that ensures organ donation and transplantation management in a manner that is decentralized, secure, reliable, traceable, auditable, and trustworthy.
- We develop smart contracts that register actors and ensure data provenance through producing events for all the necessary actions that occur during the organ donation and transplantation stages. The smart contracts code is made publicly available on Github.[1]
- We develop an auto-matching process between the donor and recipient through a smart contract based on certain criteria.
- We present six algorithms along with their full implementation, testing, and validation details.
- We conduct security analysis to determine that the proposed solution is secure against common security attacks and vulnerabilities. We compare our solution with the existing solutions to show its novelty. Our proposed solution is general and may be easily adjusted to meet the needs of a variety of related applications.

The remaining part of the paper is organized in the following manner. The related work is presented in Section II. The proposed blockchain-based solution for donated organ transplantation is explained in Section III. Then, it is followed by the implementation details of the proposed blockchain-based solution in Section IV and the details of testing and evaluation in Section V. The discussion and analysis of the proposed solution are given in Section VI. Finally, section VII concludes the paper by summarizing our contributions and outlining future research opportunities.

II. RELATED WORKS

We discuss the existing blockchain and non-blockchain-based solutions that have been proposed to address the issues in the organ donation system.

A. NON-BLOCKCHAIN-BASED SOLUTIONS FOR ORGAN DONATION MANAGEMENT

In non-blockchain-based processes, various approaches and tools are utilized to come up with solutions that enhance organ donation, transplantation management, and the matching process. The authors in [17] developed a multi-agent software platform to represent the information workflow model among donor hospitals, regulators, and recipient hospitals. This platform optimizes the pre-transplantation tasks, which can improve the process efficiency. In addition, it allows storing potential donor information and improves direct communication among all participants in the organ transplantation process. An information workflow was simulated using the developed platform, and it was estimated that the saved time might be between three to five hours.

The TransNet in [18] is a system using scanning technology for barcodes at the point of organ recovery to assist in labeling, packaging, and tracking organs and other biological materials for transplantation. It involves supplementing the labeling system with an application developed and a portable

[1] https://github.com/DonationManagment/Organs/blob/main/code.sol

barcode printer corresponding with DonorNet. During organ recovery, procurement coordinators will use the operating room's system to print labels and scan all organs to be transported. Similarly, many supply chain management solutions have relied on barcodes, RFID tags, and Electronic Product Codes (EPC) for identifying and sharing product information to facilitate the tracking of items through various phases [19].

Finally, the authors in [20] proposed a manageable mechanism, MIN, for the online matching of deceased organs to donors to improve efficiency and fairness in selecting patients within the current system in Australia. The MIN mechanism simply designates an arriving organ to a patient that minimizes |KDPI-EPTS|, tie-breaking by time on the waiting list and later randomly. The Kidney Donor Patient Inde (KDPI) estimates the quality of the organ. On the other hand, the Expected Post-Transplant Survival Score (EPTS) measures the life quality of the recipient after the transplant. After testing, the results showed that the MIN mechanism outperforms the current mechanism under consideration by the Organ and Tissue Authority in Australia.

B. BLOCKCHAIN-BASED SOLUTIONS FOR ORGAN DONATION MANAGEMENT

In [21] and [22], a blockchain-based kidney donation system named "Kidner" has been proposed. It offers a kidney-pair donation module instead of the traditional kidney waiting list, which is already in use. For example, when someone wishes to donate his/her kidney to a family member but their kidney is incompatible with the person they want to donate to, the system matches the donor's kidney to another patient who also has an inconsistent donor's kidney.

The authors in [23] proposed an organ donation decentralized app using blockchain technology. Patients use a web application to register their information, including their medical ID, organ type, blood type, and state. The system would operate on a first-in, first-out (FIFO) approach, with the exception of a patient being in a critical state. It offered better security, added transparency, and a much faster system. However, it should be modified when used in different regions according to their regulations and needs. Similarly, the authors in [24], developed a web-based application using FIFO to choose an organ donor for each actual patient seeking a transplant, and in the case of an emergency, that patient is given priority. Furthermore, an organ donation and transplantation application utilizing blockchain has been proposed in [12], where the registered hospital accepts the registered donors and registers the recipients to match them with a suitable donor based on the request.

Moreover, in [25], a use case for blockchain in organ donation has been developed. Simply, the process begins with the donor signing a smart contract for organ donation and the patient filing a transplant request. Both papers are verified and hashed by a registered doctor or nurse, who then creates a verified mismatching pair and announces it over the network. The network finds a match and sends it to a doctor for approval. If a match is found, the doctor approves, and the next step is for the doctor to generate a hash. If the doctor generates a hash, the verified matched pair becomes part of the blockchain. Finally, doctors and healthcare professionals are given all the information they need to prepare for the logistics of the surgery.

The aforementioned contributions have shown how blockchain technology can improve existing organ donation and transplantation management solutions. However, some of these contributions do not demonstrate full implementation, while others do not consider all types of organs or include all necessary criteria through the matching process.

III. PROPOSED BLOCKCHAIN-BASED SOLUTION FOR ORGAN DONATION

In this section, we present details of our blockchain-based organ donation and transplantation solution. Figure 2 presents an overview of the system architecture of our proposed solution. It shows that our solution uses two smart contracts (SCs); namely, organ donation and organ transplantation. The participants can access the functions and events of these smart contracts through a front-end decentralized application (DApp), which is connected by an application program interface (API).

Every smart contract has unique functions that can be executed only by pre-authorized participants, who will have the ability to access data stored on the chain to review transactions, logs, and events. The participants include doctors, hospital transplant team members, procurement organizers, organ matching organizers, a transporter and a transplant surgeon. The Organ Donation Smart Contract is responsible for creating a waiting list, accepting donors after medical test approval, and auto-matching between the donor and recipient. The Organ Transplantation Smart Contract is mostly in charge of the transplant process. It has three parts: removing an organ from a donor, getting the organ to the recipient, and putting the organ into the recipient. All the previous phases are logged and stored on the ledger for revision and verification purposes. Additionally, authorization, secrecy, and privacy are ensured by utilizing a private permissioned Ethereum blockchain.

A. PRIVATE PERMISSIONED ETHEREUM NETWORK

Private blockchains provide enhanced security and privacy where the transactions and data are not accessible to the public and only viewed by authorized entities. Enterprises can use the Ethereum blockchain to develop their own private-permissioned blockchain to improve privacy, security, and confidentiality. In general, details of donated organ transplantation are strictly confidential. These details include the patients' health records and family histories; therefore, a private permissioned Ethereurm blockchain is ideal for such an implementation.

B. BLOCKCHAIN INTEGRATION

The blockchain network is the backbone of our proposed solution. It serves as the basis for recording transactions and

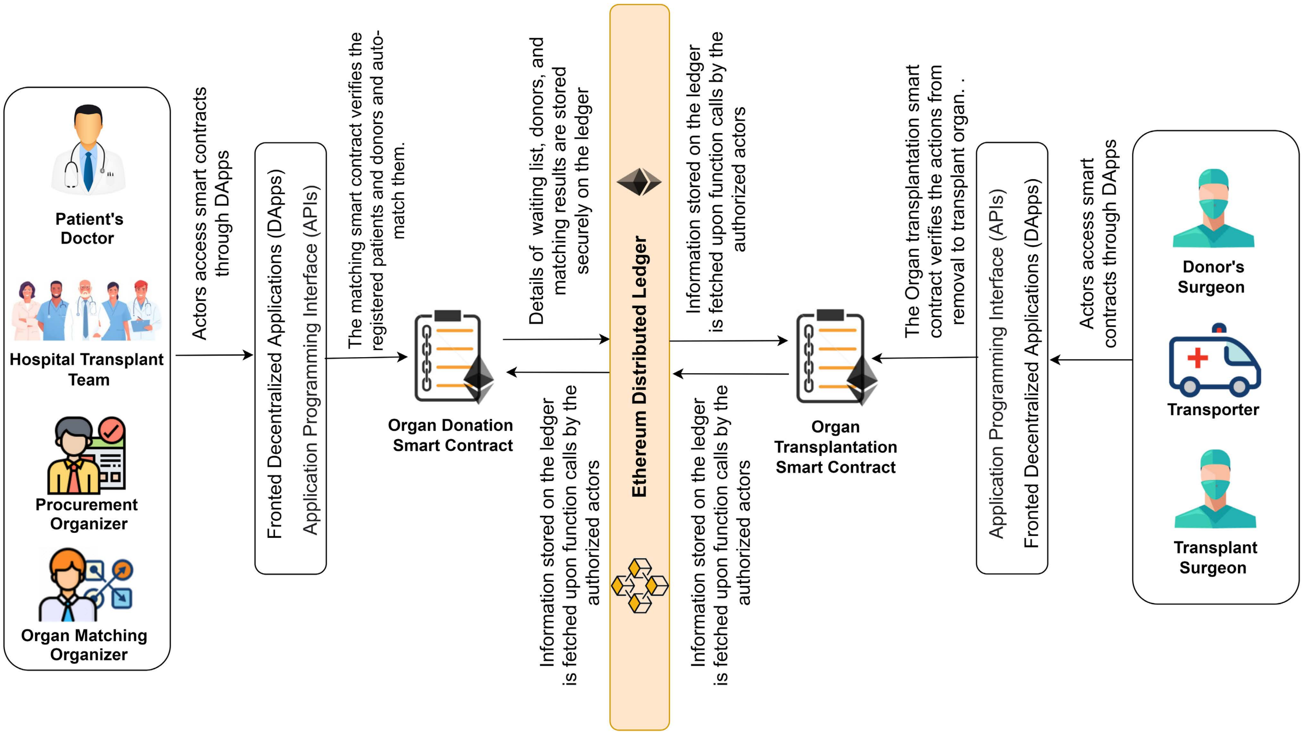

Patient's Doctor
Hospital Transplant Team
Procurement Organizer
Organ Matching Organizer
Actors access smart contracts through DApps
Fronted Decentralized Applications (DApps)
Application Programming Interface (APIs)
The matching smart contract verifies the registered patients and donors and auto-match them.
Organ Donation Smart Contract
Details of waiting list, donors, and matching results are stored securely on the ledger
Information stored on the ledger is fetched upon function calls by the authorized actors
Ethereum Distributed Ledger
Information stored on the ledger is fetched upon function calls by the authorized actors
Information stored on the ledger is fetched upon function calls by the authorized actors
Organ Transplantation Smart Contract
The Organ transplantation smart contract verifies the actions from removal to transplant organ. .
Application Programming Interface (APIs)
Fronted Decentralized Applications (DApps)
Actors access smart contracts through DApps
Donor's Surgeon
Transporter
Transplant Surgeon

FIGURE 2. A high-level system architecture of the proposed blockchain-based solution for organ donation and transplantation.

events permanently to ensure accountability and data provenance. The developed smart contracts must be deployed on the blockchain to ensure they are accessible at all times. However, it would not be ideal to deploy them on the main network during the testing phase. Therefore, a local blockchain environment, a virtual machine such as the JavaScript-based Virtual Machine, or a test network should be used to test the Ethereum-based smart contracts. The smart contracts in our proposed solution are developed using the REMIX IDE, and they are deployed on the JavaScript-based Virtual Machine which runs an isolated Ethereum node in the browser itself, which is very useful for testing purposes. Once the developed smart contracts are tested and verified, they can be deployed on Ethereum's mainnet to test their performance in a real blockchain environment. However, the outcome of the functions of the smart contracts will always be the same because they are deterministic, which means that regardless of the node that is performing the operation, the outcome will always be the same.

C. PARTICIPANTS INTERACTIONS

Figure 3 shows the interaction among different participants within the matching smart contract, which can be divided into three phases. Phase 1 begins with creating a waiting list, in which an authorized doctor will add a new patient to the waiting list. The doctor will record the patient's ID, age, BMI, and blood type. Phase 2 is fulfilled by receiving donors who have given their consent to donate their organs. Only an authorized transplant team member will run the test approval function, and an event will be sent immediately. After that, the procurement organizer is ready to evaluate and register the donor. To make the announcement that a new donor has been registered, an event will be triggered. In Phase 3, the auto-matching between the donor and recipient is handled by the organ transplantation organizer. The auto-matching process is done based on the age range, blood type, and BMI range obtained from the donor. Finally, a matched patient ranked list is announced.

Meanwhile, figure 4 represents the interaction among different participants within the transplantation smart contract. This smart contract includes removing the organ, the delivery process, and transplanting the organ. In Phase 1, the donated organ will be removed from the donor's body. Once the event is emitted for the donated organ's readiness for delivery, the transporter will execute the start delivery function. Starting and ending delivery functions in Phase 2 are called by an authorized transporter responsible for the transportation of the donated organ to the matched patient hospital and received by the transplant surgeon. In Phase 3, the donated organ is transplanted, and an event will be triggered to announce the end of the transplantation process.

IV. IMPLEMENTATION DETAILS

In this section, we present the implementation details of our proposed blockchain-based organ donation and transplantation solution along with the algorithms. The proposed system is built on a private Ethereum blockchain, to which validation nodes and only authorized participants are added.

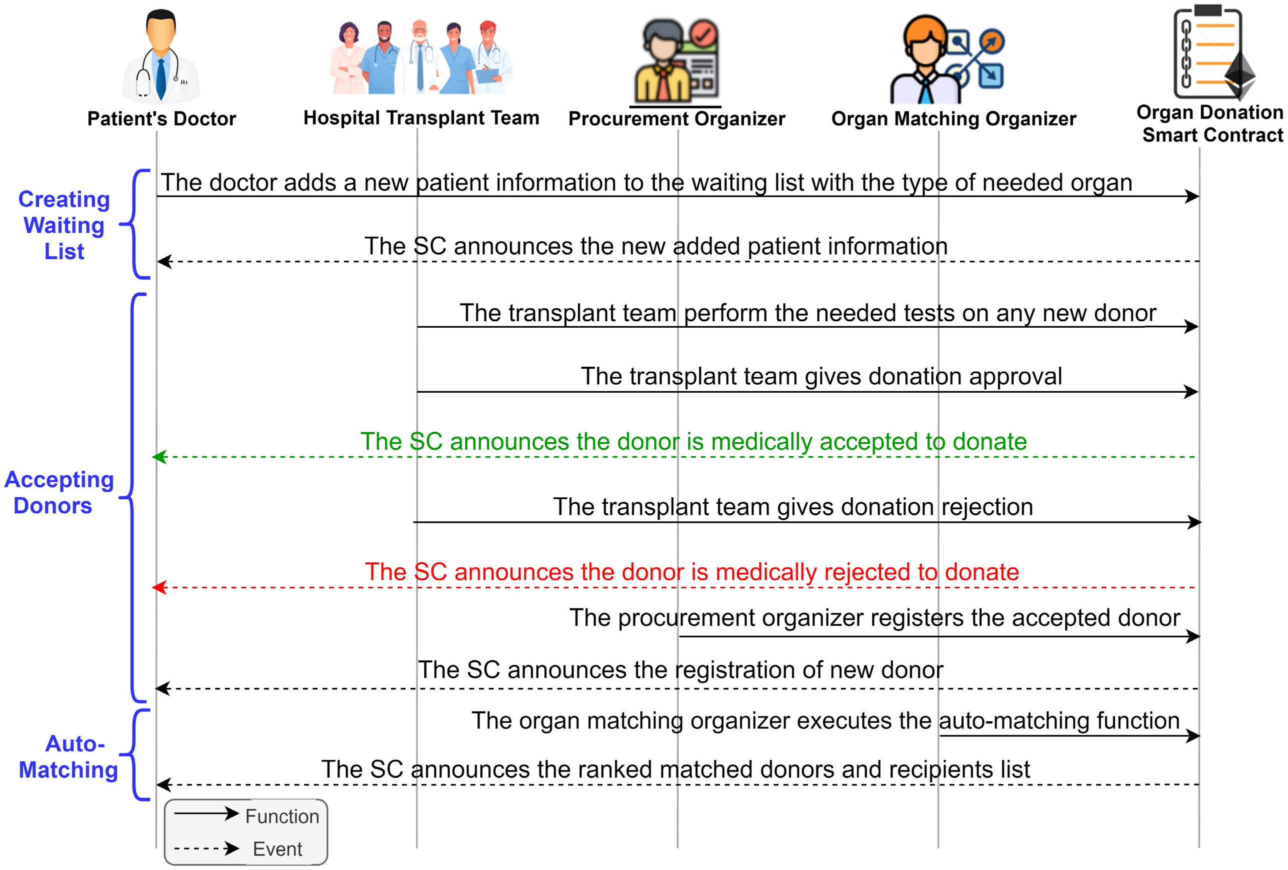

Patient's Doctor
Hospital Transplant Team
Procurement Organizer
Organ Matching Organizer
Organ Donation Smart Contract
Creating Waiting List
The doctor adds a new patient information to the waiting list with the type of needed organ
The SC announces the new added patient information
Accepting Donors
The transplant team perform the needed tests on any new donor
The transplant team gives donation approval
The SC announces the donor is medically accepted to donate
The transplant team gives donation rejection
The SC announces the donor is medically rejected to donate
The procurement organizer registers the accepted donor
The SC announces the registration of new donor
Auto-Matching
The organ matching organizer executes the auto-matching function
The SC announces the ranked matched donors and recipients list
Function
Event

FIGURE 3. Sequence diagram showing interactions among the participants and organ donation smart contract.

The smart contracts are written in Solidity and tested with the Remix IDE, which is an open source web that enables developing and administering smart contracts. The implementation of our proposed solution is mainly twofold: organ donation and organ transplantation. The details of each one are described below.

A. ORGAN DONATION

Four entities participate in the organ donation smart contract; namely, the patient's doctor, hospital transplant team member, procurement organizer, and matching organizer. Each entity has an Ethereum address and can participate by calling functions within the smart contract. This smart contract contains different types of variables. One of the variables is the Ethereum address, which is used to associate certain entities with a unique address, such as the procurement organizer and the matching organizer. The second type is mapping, which in our solution links the Ethereum address of an entity to a Boolean to reflect that the address needs certain conditions. For example, mapping is used for the authorized transplant surgeon and doctors. Moreover, mapping is used for patient validity to ensure that patient selection is not repeated. In addition, an enumerating variable called "Bloodtype," which contains the different types of blood, such as "A," "B," "AB," and "O". This variable accepts uint8 input where "0", "1", "2", and "3" represent the blood types respectively. Additionally, the enumerated variable "OrganType" accepts uint8 input where "0" represents "Heart," "1" represents "Lung," "2" represents "Liver", and "3" represents "Kidney".

The procurement organizer will deploy the organ donation smart contract. The procurement organizer deploys the smart contract and therefore becomes the owner, which permits this participant to assign the Ethereum address of the matching organizer. Then, the authorized doctor adds a new patient to the waiting list, which is then announced to all participants. Following this, the authorized medical team member performs the test and announces the test approval. Next, by the procurement organizer, the donor registration action is done and announced, including the type of donated organ. After that, the auto-matching process is conducted, and the information of matched patients with potential donors is stored. Finally, this process depends on the main criteria such as age, blood type, BMI, and waiting time.

B. ORGAN TRANSPLANTATION

In the organ transplantation smart contract, the donor's surgeon, transporter, and transplant surgeon are the main

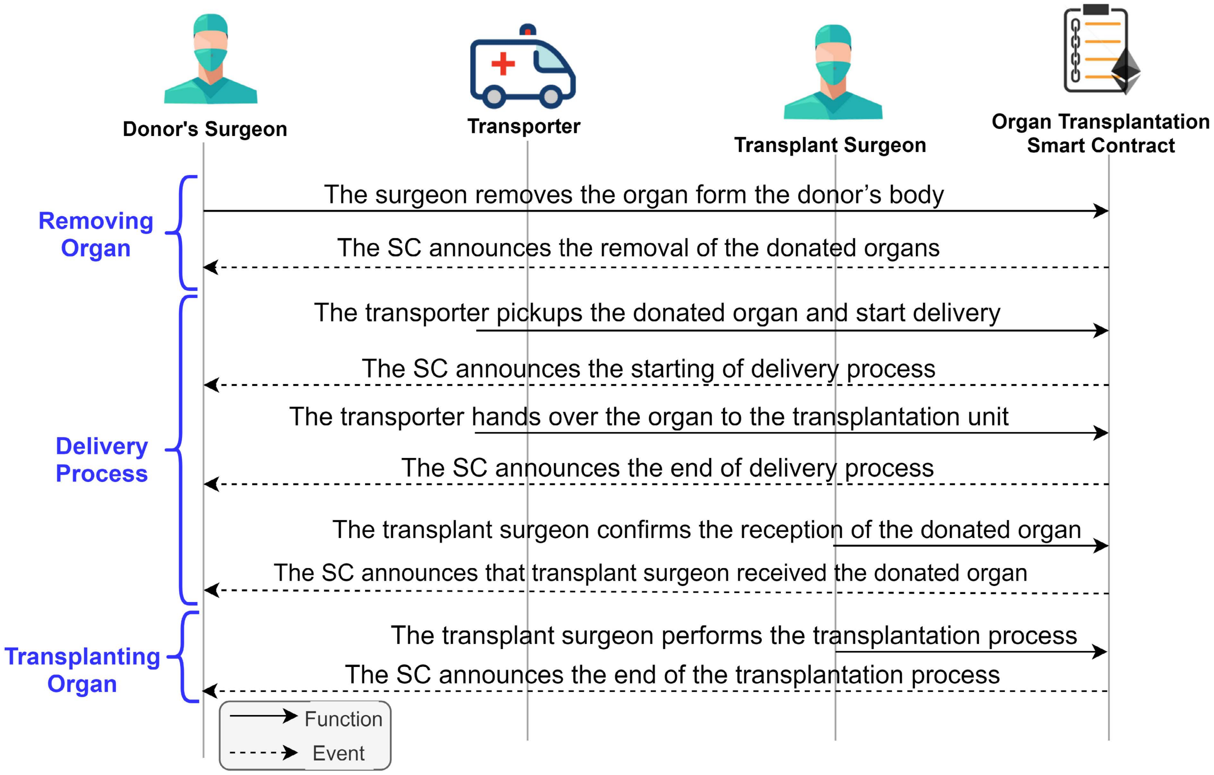

Donor's Surgeon
Transporter
Transplant Surgeon
Organ Transplantation Smart Contract
Removing Organ
The surgeon removes the organ form the donor’s body
The SC announces the removal of the donated organs
Delivery Process
The transporter pickups the donated organ and start delivery
The SC announces the starting of delivery process
The transporter hands over the organ to the transplantation unit
The SC announces the end of delivery process
The transplant surgeon confirms the reception of the donated organ
The SC announces that transplant surgeon received the donated organ
Transplanting Organ
The transplant surgeon performs the transplantation process
The SC announces the end of the transplantation process
Function
Event

FIGURE 4. Sequence diagram showing interactions among the participants and organ transplantation smart contract.

participants. Each participant can participate by calling functions within the smart contract. It includes various types of variables. For example, public Ethereum addresses hold the address of the donor and transplant surgeons. Moreover, it has a mapping for the authorized transporters, which is allowed to transport the removed donated organ from the donor hospital to the recipient hospital. Furthermore, the “OrganStatus” is an enumerated variable and contains all of the various states that the donated organ will go through.

The Transplant surgeon will deploy the smart contract. The Ethereum address of the donor’s surgeon and the initial state of the removed organ will be defined. The transplantation tracing process occurs once the smart contract is deployed and the authorized transporters are assigned. First, the donated organ is removed by the surgeon and transported by the authorized transporter from where the location of the donor to the recipient hospital. Then, The start and end of the delivery procedure will be notified. After that, the transplant surgeon announces the reception of the donated organ and start transplanting it. Finally, the transplantation details will be announced, including the patient ID, time, and date of the process.

Figure 5 displays the entity-relationship diagram that shows the main attributes and functions in each smart contract. In our solution, only one procurement organizer and one matching organizer can exist; thus, they are declared as Ethereum addresses. Meanwhile, the patient’s doctor, transplant team member, and patient validity are declared as mappings because many of them are in the system. Lastly, blood type and organ type are declared as enumerates as there are different types for each. Finally, AddingNewPatient, TestApproval, RegisteringNewDonor, and MatchingProcess are the main functions of the organ donation smart contract.

The Organ Transplantation smart contract has a set of attributes used to describe the details of the transplantation process. There is only one surgeon to take the responsibility from the donor side and only one surgeon to do the transplantation for the recipient; therefore, they are declared as Ethererum addresses. Furthermore, more than one transporter can exist in the system; thus, it is declared as mapping. In addition, the smart contract has five primary functions; namely, RemovingDonatedOrgan, StartDelivery, EndDelivery, ReceiveDonatedOrgan, and Organ_Transplantation. Finally, the organ donation smart contract will have a 1:n relationship with the transplantation smart contract since only one organ donation smart contract can include all patients, whereas several transplantation smart contracts can exist for the various possible donation processes.

We present six algorithms to explain the details of the various functions involved in our smart contracts.

The donor donation smart contract phases are displayed in algorithms 1, 2 and 3, respectively. On the other hand, the rest of the algorithms illustrate the second smart contract, the Organ Transplantation smart contract. Algorithm 1 represents

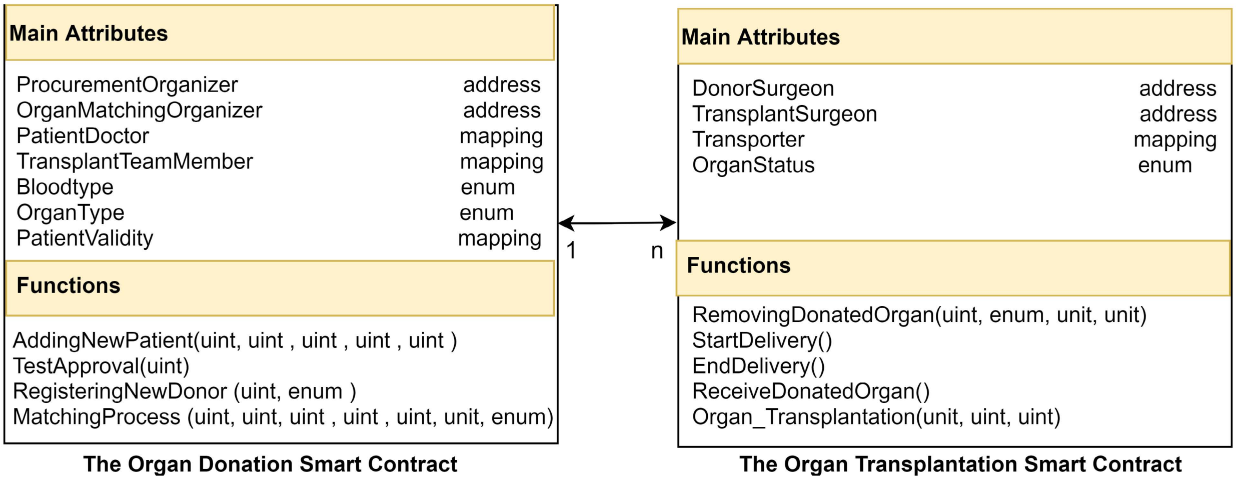

Main Attributes
ProcurementOrganizer address
OrganMatchingOrganizer address
PatientDoctor mapping
TransplantTeamMember mapping
Bloodtype enum
OrganType enum
PatientValidity mapping
Functions
AddingNewPatient(uint, uint , uint , uint , uint)
TestApproval(uint)
RegisteringNewDonor (uint, enum)
MatchingProcess (uint, uint, uint , uint , uint, unit, enum)
The Organ Donation Smart Contract
1
n
Main Attributes
DonorSurgeon address
TransplantSurgeon address
Transporter mapping
OrganStatus enum
Functions
RemovingDonatedOrgan(uint, enum, unit, unit)
StartDelivery()
EndDelivery()
ReceiveDonatedOrgan()
Organ_Transplantation(unit, uint, uint)
The Organ Transplantation Smart Contract

FIGURE 5. Smart contracts entity-relationship diagram.

```
Algorithm 1: Adding New Patient
Input: Patient_ID, Patient_Age, Patient_BMI,
       Bloodtype_, _OrganType_
Patient_ID is the ID of the patient on the waiting list.
Patient_Age is the age of the patient on the waiting list.
Patient_BMI is the body mass index of the patient.
Bloodtype_ is a particularized enumerate variable that
  represents types of blood.
_OrganType_ is a particularized enumerate variable that
  represents types of the needed organ.
Output: An event announcing that a new patient is
        Added to the waiting list
assignedDoctorsToAddPaitients_list: A mapping of the
  patients doctors EAs
if caller == ProcurementOrganizer then
    assigningDoctor_list[PatientDoctor] = true.
else
    Revert.
end
/* The list of the assigned patients
   doctors is ready                      */
if (caller == PatientDoctor) then
    Patient_ID ← PatientsID [i]
    _OrganType_ ← NeededOrganType [i]
    Patient_Age ← Patients_age[i]
    _BloodType ← Blood_type [i]
    Patient_BMI ← BMI [i]
else
    Revert.
end
```

```
Algorithm 2: Donor Medical Test and Registration
Input: Donor_ID, _DonatedOrganType
Donor_ID is the ID of the organ donor.
_OrganType_ is a particularized enumerate variable that
  represents various types of the donated organ.
assignedTransplantMembers_list: A mapping of the
  transplant team members EAs
if caller == ProcurementOrganizer then
    assigningmember_list[TransplantTeamMember] =
      true.
else
    Revert.
end
/* Assigned transplant team members
   list is ready                         */
if (caller ==∈ assignedTransplantMembers_list) then
    Emit an event announcing that the donor is
    medically approved to donate
else
    Revert.
end
/* Medical test approval is done         */
if caller == ProcurementOrganizer then
    Emit an event announcing that the donor is registered
else
    Revert.
end
```

the creation of a waiting list phase where new patients are added. This algorithm represents the AssigningPatientDoctors and AddingNewPatient functions. The doctors need to add the patient's information such as ID, age, BMI, blood type, and needed organ type as all the information will be stored in an array and used later in the matching phase.

Moreover, the second phase, which is related to donor acceptance and registration, is illustrated in algorithm 2. The procurement organizer will assign the transplant team members who are responsible for performing medical tests and checkups. Additionally, the donor ID and the type of the

Algorithm 3: Matching Process

```
Input: Min_Age, Max_Age, Donor_BloodType,
       Donor_MinBMI, Donor_MaxBMI,
       _OrganType_
Min_Age is the minimum acceptable age that can be
  matched with the donor's age
Max_Age is the maximum acceptable age that can be
  matched with the donor's age
Donor_BloodType is the blood type of the donor
Donor_MinBMI is the minimum acceptable body mass
  index that can be matched with the donor's BMI
Donor_MaxBMI is the maximum acceptable body mass
  index that can be matched with the donor's BMI
_OrganType_ is a particularized enumerate variable that
  represents various types of the needed organ.
if $caller == OrganMatchingOrganizer$ then
    for <i = 0 to < Patients.length> do
        if $Needed\ Organ\ Type[i] == \_OrganType \wedge$
        $(Patients\_age[i] > Min\_Age) \wedge$
        $(Patients\_age[i] < Max\_Age) \wedge$
        $(Blood\_type[i] == \_BloodType) \wedge$
        $(BMI[i] > Donor\_MinBMI) \wedge$
        $(BMI[i] < Donor\_MaxBMI)$ then
            Matched.push(Patients[i])
        end
        New matched organ
    end
    Revert
end
```

Algorithm 4: Removing the Donated Organ

```
Input: Donor_ID, Removing_date, Removing_time,
       _DonatedOrganType
Donor_ID is the ID of the organ donor.
Removing_date is the date of removing the donated
  organ.
Removing_time is the time of removing the donated
  organ.
_DonatedOrganType is a particularized enumerate
  variable that represents various types of the donated
  organ.
Organstate is a particularized enumerate variable that
  represents the states of the removed donated organ.
if $(caller == DonorSurgeon) \wedge (Organstate ==$
  $NotReady)$ then
    if $\_DonatedOrganType == Heart$ then
        $Organstate = ReadyforDelivery$
        Donated heart is removed
    else
        if $\_DonatedOrganType == Lung$ then
            $Organstate = ReadyforDelivery$
            Donated lung is removed
        else
            if $\_DonatedOrganType == Liver$ then
                $Organstate = ReadyforDelivery$
                Donated Liver is removed
            else
                $\_DonatedOrganType = Kidney$
                $Organstate = ReadyforDelivery$
                Donated kidney is removed
            end
        end
    end
else
    Revert
end
```

donated organ need to be added by the procurement organizer in the RegisteringNewDonor function. Furthermore, algorithm 3 shows the matching process. This process is represented in the MatchingProcess function. In addition, to start this process, the organ matching organizer should add the following: minimum age, maximum age, donor blood type, minimum BMI, maximum BMI, and donated organ type. Each patient must meet the defined conditions in the algorithms to decide if that patient is fit to receive a donated organ.

Algorithm 4 represents the first phase of the Organ Transplantation smart contract, which is removing the donated organ. This algorithm shows that the donor surgeon needs to add some information such as the donor ID, removing date and time, and the donated organ type to start the removing process. Then, immediately, the delivery process will be started by an authorized transporter. This process is represented in algorithm 5. The first function is to assign a transporter, which is run by the donor surgeon. After that, the delivery process will be started by the transporter and ended when the transplant surgeon receives it. Moreover, the receiving and transplanting processes are represented in algorithm 6 in which the transplant surgeon will confirm the reception of the donated organ. Then, the transplanting process will be started. Finally, this process is portrayed in the Organ_Transplantation function, where the transplant surgeon needs to add the patient ID and surgery date and time.

V. TESTING AND VALIDATION

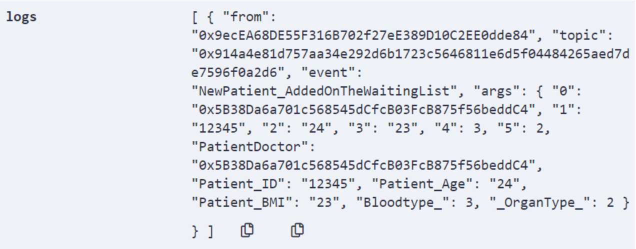
In this section, we test and validate the primary functions of the developed organ donation and organ transplantation smart contracts. The assessment stage is implemented using the Remix IDE. The participants and their Ethereum addresses used throughout the testing and validation process are listed in table 1. Additionally, for the functions, the used inputs do not reflect real data; they are simply assumptions for testing needs. The logs and transactions of the major smart contract functions are further explained in the following subsections.

A. ORGAN DONATION SMART CONTRACT

Figure 6 shows a successful execution of the AddingNewPatient function that adds new patients to the waiting list.

Algorithm 5: Delivery

```
Organstate is a particularized enumerate variable that
    represents various states of the removed donated organ.
assigned transporters list: is a mapping that indicates
    which transporters are assigned for transportation
if caller == DonorSurgeon then
    assigning transporters list [transporter] = true.
else
    Reject transaction.
end
/* The list of transporters is
   prepared                                          */
if (caller ==∈ assigned_transporters_list)∧
    (bloodunitstate == Ready for Delivery) then
    Organstate = on Track
    Emit an event announcing that the donated organ is
        prepared for delivery
else
    Revert.
end
/* The assigned transporter started
   the delivery process                              */
if (caller==∈ assigned_transporters_list)∧
    (bloodunitstate == on Track) then
    Organstate = End Delivery process
    Emit an event announcing that the donated organ has
        been delivered
else
    Revert.
end
```



```
logs                [ { "from":
                    "0x9ecEA68DE55F316B702f27eE389D10C2EE0dde84", "topic":
                    "0x914a4e81d757aa34e292d6b1723c5646811e6d5f04484265aed7d
                    e7596f0a2d6", "event":
                    "NewPatient_AddedOnTheWaitingList", "args": { "0":
                    "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4", "1":
                    "12345", "2": "24", "3": "23", "4": 3, "5": 2,
                    "PatientDoctor":
                    "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                    "Patient_ID": "12345", "Patient_Age": "24",
                    "Patient_BMI": "23", "Bloodtype_": 3, "_OrganType_": 2 }
                    } ]
```

FIGURE 6. Successful execution of the AddingNewPatient function.

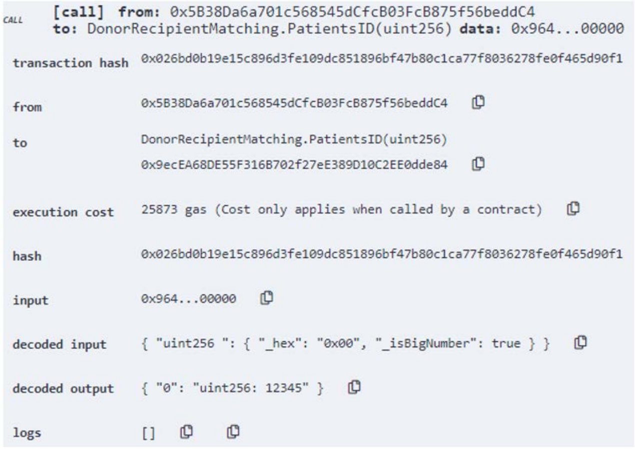
This function essentially allows the authorized doctors, who are the only participants allowed to execute it, to store the details of the new patients on the Ethereum network. The "logs" field presents the information that is stored as an event on the Ethereum network. "12345" is the patient ID, "24" is the age, "23" is the BMI, "3" is the O blood type, as it was mentioned before, and "2" is the type of organ needed. In addition, figure 7 shows a successful call of the new patient ID which has been stored in the PatientID array. Similarly, other functions, such as TestApproval and RegisteringNewDonor were executed successfully.

Algorithm 6: Receiving and Transplanting Donated Organ

```
Input: Patient_ID, Transplantation_date,
       Transplantation_time
Patient_ID
Transplantation_date
Transplantation_time
Organstate is a particularized enumerate variable that
    represents the states of the removed donated organ.
if (caller == TransplantSurgeon)∧ (bloodunitstate ==
    EndDelivery) then
    Organstate = OrganReceived
    Emit an event announcing that the donated organ has
        been received
else
    Revert.
end
/* Donated organ is received by the
   transplant surgeon                                */
if (caller == TransplantSurgeon) then
    Emit an event announcing that the organ
        transplantation is done
else
    Revert
end
```

TABLE 1. The Ethereum addresses of participants in the testing scenario.

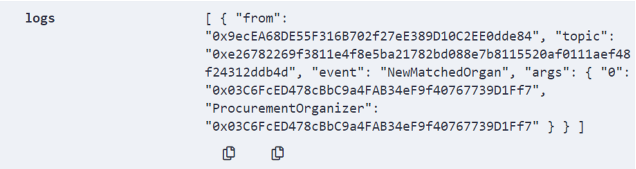
	Ethereum Address
Patient's Doctor	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
Hospital Transplant Team Member	0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C
Procutmrnt Organizer	0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
Organ Matching Organizer	0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
Donor's Surgeon	0x0A098Eda01Ce92ff4A4CCb7A4fFFb5A43EBC70DC
Transporter	0x583031D1113aD414F02576BD6afaBfb302140225
Transplant Surgeon	0x5c6B0f7Bf3E7ce046039Bd8FABdfD3f9F5021678



```
CALL    [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
        to: DonorRecipientMatching.PatientsID(uint256) data: 0x964...00000

transaction hash   0x026bd0b19e15c896d3fe109dc851896bf47b80c1ca77f8036278fe0f465d90f1

from               0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to                 DonorRecipientMatching.PatientsID(uint256)
                   0x9ecEA68DE55F316B702f27eE389D10C2EE0dde84

execution cost     25873 gas (Cost only applies when called by a contract)

hash               0x026bd0b19e15c896d3fe109dc851896bf47b80c1ca77f8036278fe0f465d90f1

input              0x964...00000

decoded input      { "uint256 ": { "_hex": "0x00", "_isBigNumber": true } }

decoded output     { "0": "uint256: 12345" }

logs               []
```

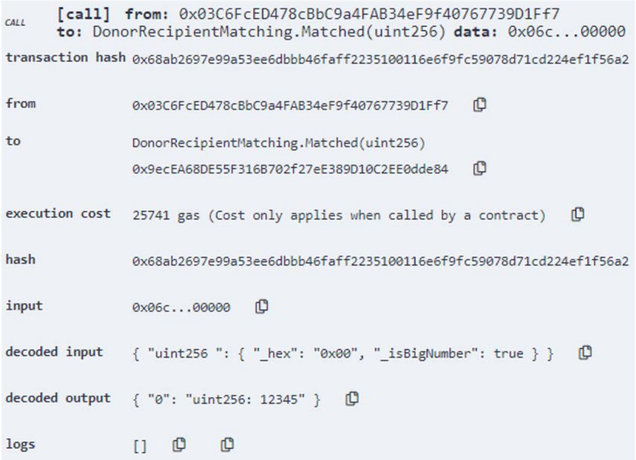
FIGURE 7. Successful calling of the new added patient ID.

In the MatchingProcess function, it was tested if the organ matching organizer is able to add the details of a new registered donor, such as minimum age, maximum age, donor blood type, minimum BMI, maximum BMI, and donated



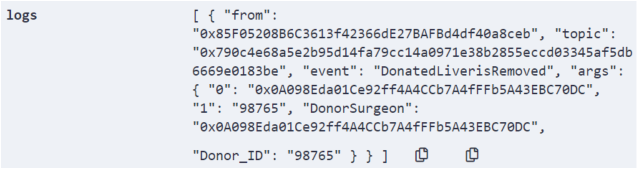
```
logs                [ { "from":
                    "0x9ecEA68DE55F316B702f27eE389D10C2EE0dde84", "topic":
                    "0xe26782269f3811e4f8e5ba21782bd088e7b8115520af0111aef48
                    f24312ddb4d", "event": "NewMatchedOrgan", "args": { "0":
                    "0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7",
                    "ProcurementOrganizer":
                    "0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7" } } ]
```

FIGURE 8. Successful execution of the MatchingProcess function.



```
CALL    [call] from: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
        to: DonorRecipientMatching.Matched(uint256) data: 0x06c...00000
transaction hash 0x68ab2697e99a53ee6dbbb46faff2235100116e6f9fc59078d71cd224ef1f56a2

from             0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7

to               DonorRecipientMatching.Matched(uint256)
                 0x9ecEA68DE55F316B702f27eE389D10C2EE0dde84

execution cost   25741 gas (Cost only applies when called by a contract)

hash             0x68ab2697e99a53ee6dbbb46faff2235100116e6f9fc59078d71cd224ef1f56a2

input            0x06c...00000

decoded input    { "uint256 ": { "_hex": "0x00", "_isBigNumber": true } }

decoded output   { "0": "uint256: 12345" }

logs             []
```

FIGURE 9. Successful calling of the matched patient ID.



```
logs                [ { "from":
                    "0x85F05208B6C3613f42366dE27BAFBd4df40a8ceb", "topic":
                    "0x790c4e68a5e2b95d14fa79cc14a0971e38b2855eccd03345af5db
                    6669e0183be", "event": "DonatedLiverisRemoved", "args":
                    { "0": "0x0A098Eda01Ce92ff4A4CCb7A4fFFb5A43EBC70DC",
                    "1": "98765", "DonorSurgeon":
                    "0x0A098Eda01Ce92ff4A4CCb7A4fFFb5A43EBC70DC",

                    "Donor_ID": "98765" } } ]
```

FIGURE 10. Successful execution of the RemovingDonatedOrgan function.

organ type. A successful execution of the function and its corresponding logs and events are displayed in figure 8. This function is critical in this smart contract because it notifies the matched patient with the available donor. Moreover, figure 9 displays a successful call of the matched patient that has been stored in the Matched array where the available donor had been matched with the patient that had the ‘‘12345’’ ID.

B. ORGAN TRANSPLANTATION SMART CONTRACT

Figure 10 shows the details of announcing the removal of the donated organ by the donor surgeon where an execution of the RemovingDonatedOrgan function was performed successfully. Similarly, successful executions were done for all functions involved in the delivery and transplantation phases.

VI. DISCUSSION

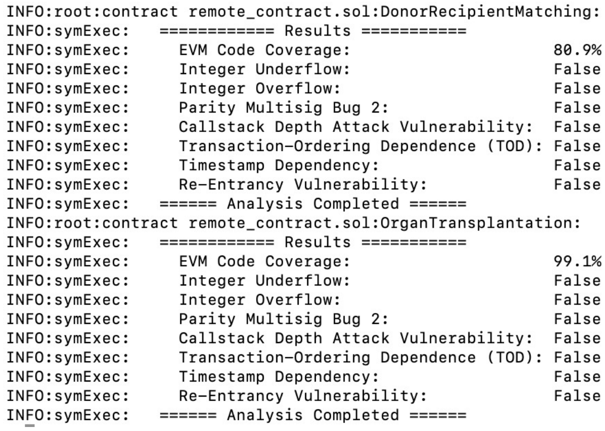
In this section, we evaluate the privacy, confidentiality, and security level of our solution. Typically, Ethereum-based solutions present the costs involved in implementing and executing smart contracts. However, in our solution, the private Ethereum blockchain is utilized such that the gas price is set to zero. Thus, there are no involved costs. Additionally, a comparison between our solution and existing solutions is also accomplished. Finally, we discuss how our solution can be generalized to other applications/systems.

A. SECURITY ANALYSIS

- **Integrity:** The proposed solution for organ donation management uses an event-based strategy where each transaction is recorded and stored on an immutable ledger, allowing users to trace the donation and transplantation processes action by action. For example, any ranked list of the matched recipients with the available donor that has taken place is recorded in the form of an event to confirm that the donated organ is given to the recipient with the highest priority.
- **Authorization and Accountability:**
  The Ethereum smart contracts are written in Solidity language in the proposed solution, where the ‘‘Modifier’’ feature is used. This feature permits specific participants to run particular functions. Therefore, all participants are accountable for their actions, and any illegal action taken is stored as an event in an immutable ledger. Therefore, mistakes and illegal actions can be traced to identify their sources. For instance, only the organ matching organizer can run the matching function where the matched list is stored and can be fetched later by the DApp. Moreover, only a transplant surgeon can run the organ transplantation function.Therefore, these participants are accountable for their actions, and they will be held responsible for any manipulation or mistakes in these two processes.
- **Availability:** The decentralization of the Ethereum blockchain means there are multiple distributed nodes that are responsible for recording and logging all the transactions that occur in the network, which ensures the availability and synchronization of the network even if a node fails.
- **MITM Attacks:**
  The private key signature of the sender is required for each transaction in the ledger. This guarantees that attackers cannot tamper with the blockchain, making man-in-the-middle (MITM) attacks very unlikely to occur. This feature is critical for enhancing the management of blockchain-based organ donation and transplantation, where only trusted entities can receive information and perform actions.

B. ANALYSIS OF SMART CONTRACT SECURITY

A specialized tool called Oyente was utilized to test and validate the organ donation and organ transplantation smart contracts against vulnerabilities. Oyente is capable of detecting some of the latest security flaws in the Ethereum blockchain, such as the parity multisig bug, integer overflow, integer underflow, transaction-ordering dependence (TOD), call-stack depth attack, timestamp dependency, and re-entrancy.



```
INFO:root:contract remote_contract.sol:DonorRecipientMatching:
INFO:symExec:   ============ Results ===========
INFO:symExec:     EVM Code Coverage:                     80.9%
INFO:symExec:     Integer Underflow:                     False
INFO:symExec:     Integer Overflow:                      False
INFO:symExec:     Parity Multisig Bug 2:                 False
INFO:symExec:     Callstack Depth Attack Vulnerability:  False
INFO:symExec:     Transaction-Ordering Dependence (TOD): False
INFO:symExec:     Timestamp Dependency:                  False
INFO:symExec:     Re-Entrancy Vulnerability:             False
INFO:symExec:   ====== Analysis Completed ======
INFO:root:contract remote_contract.sol:OrganTransplantation:
INFO:symExec:   ============ Results ===========
INFO:symExec:     EVM Code Coverage:                     99.1%
INFO:symExec:     Integer Underflow:                     False
INFO:symExec:     Integer Overflow:                      False
INFO:symExec:     Parity Multisig Bug 2:                 False
INFO:symExec:     Callstack Depth Attack Vulnerability:  False
INFO:symExec:     Transaction-Ordering Dependence (TOD): False
INFO:symExec:     Timestamp Dependency:                  False
INFO:symExec:     Re-Entrancy Vulnerability:             False
INFO:symExec:   ====== Analysis Completed ======
```

FIGURE 11. Smart contracts vulnerability analysis.

TABLE 2. A comparison of the proposed solution with the current available solutions.

Features	Our Solution	Alandjani [25]	Dajim et al. [23]	Zouarhi [21]
Blockchain Platform	Ethereum	NA	NA	Ethereum
Smart Contract	Yes	Yes	No	No
Mode of Operation	Private	Public	Public	Private
Tracing	Yes	Yes	Yes	Yes
Real-Time Monitoring	No	No	No	No
Implementation	Yes	No	No	Yes
DApps	No	No	Yes	Yes

Figure 11 displays the output of the Oyente tool execution. It demonstrates that the designed smart contracts have no existing vulnerabilities, showing that they are robust and highly secured against common vulnerabilities. In addition, the EVM code coverage was 80.9% for the organ donation smart contract and 99.1% for the organ transplantation smart contract.

C. COMPARISON WITH THE EXISTING SOLUTIONS

In table 2, a comparative analysis between our solution and the existing blockchain-based solutions is presented. The comparison is performed using some important parameters, such as the used blockchain platform, mode of operation, smart contract development, tracing capability, real-time monitoring, implementation, and DApps development. Our solution and [17] used the Ethereum network in a private mode of operation. In contrast, other solutions did not mention the type of blockchain platform that was used. A smart contract based approach is proposed in [21] and our solution only. Finally, for [21] and our solution, the front-end decentralized application for users to utilize the solution has not been developed, whereas [19] and [17] developed it for their solutions.

D. GENERALIZATION

Our proposed blockchain-based solution shows how blockchain technology can assist in managing and tracing the donated organ transplantation process during the registration, match, removal, transportation, and transplantation phases. The designed smart contracts that represent the different phases of the donated organ transplantation management system can be customized to fit other systems that involve highly sensitive items and require tracking, tracing, and accountability. For example, delivery functions can be used in many applications in healthcare and other industries and domains. Likewise, the principle of comparing and auto-matching between the donor and the registered patients can be used in blood donation operations, medical device donation, or even industry products.

Figure 2 can further define what modifications are required in order to adapt the proposed solution to other systems or applications. The participants and their interactions will be different, and off-chain storage is needed if the application involves large-sized content. Additionally, the developed algorithms for smart contracts can be modified to match the needs of any new system. Overall, other applications can have a similar structure to our solution, but some designations should be taken care of based on the targeted use case.

VII. CONCLUSION

In this paper, we have proposed a private Ethereum blockchain-based solution that manages organ donation and transplantation in a decentralized, accountable, auditable, traceable, secure, and trustworthy manner. We developed smart contracts that ensure the data provenance by recording events automatically. We present six algorithms with their implementation, testing, and validation details. We analyze the security of the proposed solution to guarantee that smart contracts are protected against common attacks and vulnerabilities. We compare our solution to other blockchain-based solutions that are currently available. We discuss how our solution can be customized with minimal effort to meet the needs of other systems experiencing similar problems. In the future, our solution can be improved by developing an end-to-end DApp. Furthermore, the smart contracts can be deployed and tested on a real private Ethereum network. Finally, the Quorum platform can provide better confidentiality because transactions among entities can only be viewed by specific participants and nobody else, which is not the case in our solution, where transactions between two participants are viewed by other actors authorized in the private blockchain.

REFERENCES

[1] L. A. Dajim, S. A. Al-Farras, B. S. Al-Shahrani, A. A. Al-Zuraib, and R. Merlin Mathew, "Organ donation decentralized application using blockchain technology," in *Proc. 2nd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, May 2019, pp. 1–4, doi: 10.1109/cais.2019.8769459.

[2] A. Powell. (Mar. 18, 2019). *A Transplant Makes History. Harvard Gazette*. [Online]. Available: https://news.harvard.edu/gazette/story/2011/09/a-transplant-makes-history/

[3] *Organ Donation Facts and Info: Organ Transplants*. Accessed: Apr. 18, 2021. [Online]. Available: https://my.clevelandclinic.org/health/articles/11750-organ-donation-and-transplantation

[4] (Mar. 21, 2019). *Facts and Myths About Transplant*. Accessed: Apr. 21, 2021. [Online]. Available: https://www.americantransplantfoundation.org/about-transplant/facts-and-myths/

[5] *Organ Procurement and Transplantation Network*. Accessed: Apr. 18, 2021. [Online]. Available: https://optn.transplant.hrsa.gov/resources/ethics/ethical-principles-in-the-allocation-of-humanorgans/

[6] *How Donation Works*. Accessed: Jan. 7, 2022. [Online]. Available: https://www.organdonor.gov/learn/process

[7] UFO Themes. (Aug. 1, 2017). *Organ Donation and Transplantation in Germany. Plastic Surgery Key*. [Online]. Available: https://plasticsurgerykey.com/organ-donation-and-transplantation-in-germany/

[8] Harvard Business Review. (Dec. 13, 2021). *Electronic Health Records Can Improve the Organ Donation Process*. Accessed: Apr. 8, 2022. [Online]. Available: https://hbr.org/2021/12/electronic-health-records-can-improve-the-organ-donation-process

[9] U. Jain, "Using blockchain technology for the organ procurement and transplant network," San Jose State Univ., San Jose, CA, USA, Tech. Rep., 2020, doi: 10.31979/etd.g45p-jtuy.

[10] M. He, A. Corson, J. Russo, and T. Trey, "Use of forensic DNA testing to trace unethical organ procurement and organ trafficking practices in regions that block transparent access to their transplant data," *SSRN Electron. J.*, 2020, doi: 10.2139/ssrn.3659428.

[11] Livemint. *The Illegal Organ Trade Thrives in India-and it isn't Likely to End Soon*. Accessed: Dec. 21, 2021. [Online]. Available: https://www.livemint.com/Politics/pxj4YasmivrvAhanv6OOCJ/Why-organ-trafficking-thrives-in-India.html

[12] D. P. Nair. (2016). *Organ is Free, Transplant Cost is Problem*. [Online]. Available: https://timesofindia.indiatimes.com/life-style/healthfitness/health-news/Organ-is-free-transplant-cost-isproblem/articleshow/54014378.cms

[13] P. Ranjan, S. Srivastava, V. Gupta, S. Tapaswi, and N. Kumar, "Decentralised and distributed system for organ/tissue donation and transplantation," in *Proc. IEEE Conf. Inf. Commun. Technol.*, Dec. 2019, pp. 1–6, doi: 10.1109/cict48419.2019.9066225.

[14] V. Puggioni. (Feb. 26, 2022). *An Overview of the Blockchain Development Lifecycle. Cointelegraph*. Accessed: Apr. 8, 2022. [Online]. Available: https://cointelegraph.com/explained/an-overview-of-the-blockchain-development-lifecycle

[15] *History of Blockchain*. Accessed: Apr. 8, 2022. [Online]. Available: https://www.icaew.com/technical/technology/blockchain-and-cryptoassets/blockchain-articles/what-is-blockchain/history

[16] M. Hölbl, M. Kompara, A. Kamišalić, and L. N. Zlatolas, "A systematic review of the use of blockchain in healthcare," *Symmetry*, vol. 10, no. 10, p. 470, Oct. 2018, doi: 10.3390/sym10100470.

[17] V. Ferraza, G. Oliveira, P. Viera-Marques, and R. Cruz-Correia, "Organs transplantation—How to improve the process ?" Eur. Fed. Med. Inform., Cardiff, U.K., Tech. Rep., 2011, doi: 10.3233/978-1-60750-806-9-300.

[18] *Organ Procurement and Transplantation Network*. Accessed: Nov. 27, 2021. [Online]. Available: https://optn.transplant.hrsa.gov/governance/public-comment/standardize-organ-coding-and-tracking-system/

[19] A. Bougdira, A. Ahaitouf, and I. Akharraz, "Conceptual framework for general traceability solution: Description and bases," *J. Model. Manage.*, vol. 15, no. 2, pp. 509–530, Oct. 2019.

[20] N. Mattei, A. Saffidine, and T. Walsh, "Mechanisms for online organ matching," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 345–351, doi: 10.24963/ijcai.2017/49.

[21] S. Zouarhi, "Kidner—A worldwide decentralised matching system for kidney transplants," *J. Int. Soc. Telemed. E-Health*, vol. 5, Apr. 2017, Art. no. e62. [Online]. Available: https://journals.ukzn.ac.za/index.php/JISfTeH/article/view/287

[22] *Kidner Project*. Accessed: Dec. 28, 2021. [Online]. Available: https://www.kidner-project.com/

[23] L. A. Dajim, S. A. Al-Farras, B. S. Al-Shahrani, A. A. Al-Zuraib, and R. M. Mathew, "Organ donation decentralized application using blockchain technology," in *Proc. 2nd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, May 2019, pp. 1–4, doi: 10.1109/cais.2019.8769459.

[24] A. Soni and S. G. Kumar, "Creating organ donation system with blockchain technology," *Eur. J. Mol. Clin. Med.*, vol. 8, no. 3, pp. 2387–2395, Apr. 2021.

[25] G. Alandjani, "Blockchain based auditable medical transaction scheme for organ transplant services," Tech. Rep., 2019, doi: 10.17993/3ctecno.2019.specialissue3.