Utilization Of Algorithms, Dynamic Programming, Optimal Memory Utilization

Team Id	NM2023TMID04410
Project Name	Drug Traceability

Dynamic Programming, Optimal Memory Utilization:

.

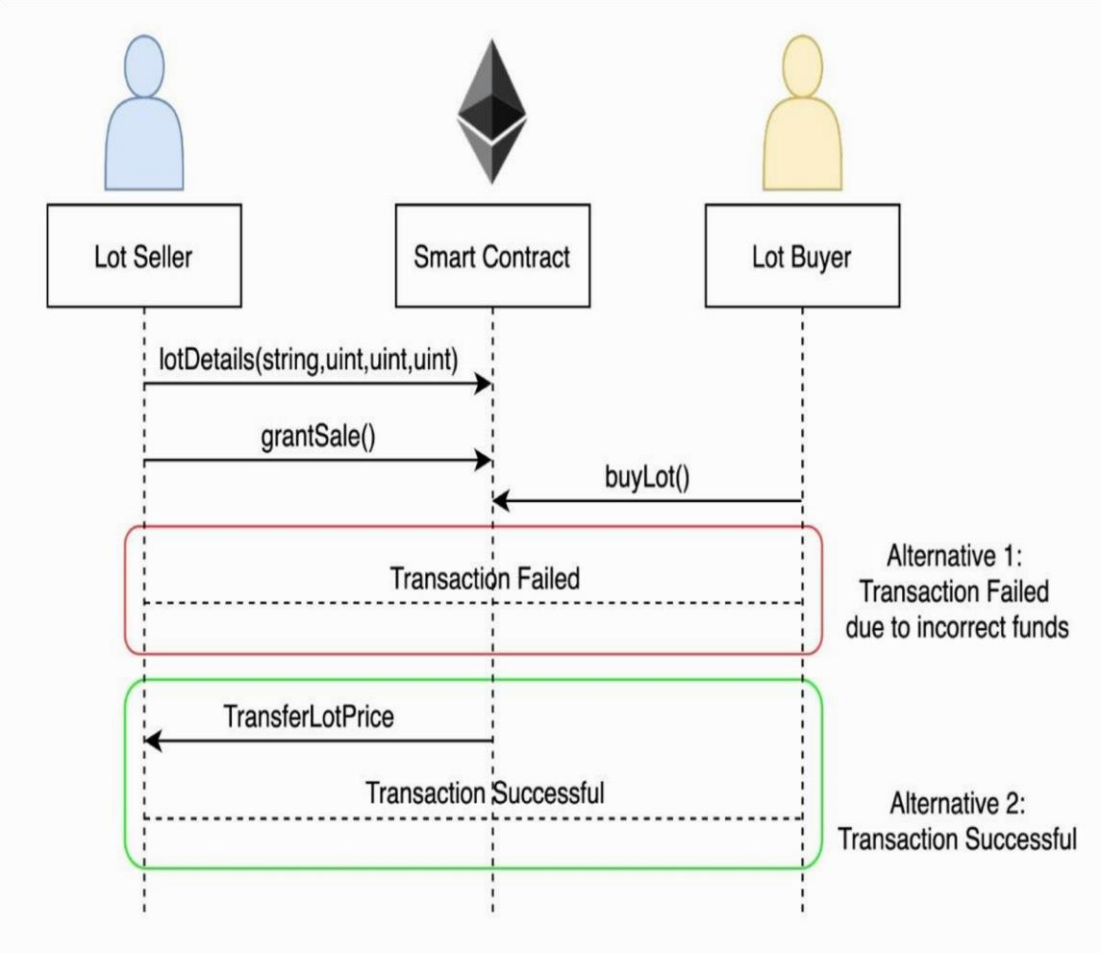
The manufacturer will first deploy the smart contract in which details of the manufactured drug Lot will be defined, declared and an event will be triggered and announced to all participants in the supply chain

- In case new participants are added to the network, they will have access to the events since they are permanently stored on the ledger and therefore they can track and trace the history of any manufactured drug Lot.

- The manufacturer also has the option of uploading an image of the Lot to the IPFS so that it can be accessed by participating entities to visually inspect the drug Lot.

- Prior to the sale of the newly manufactured Lot, it has to be packaged, the manufacturer will announce to other participants that the newly manufactured Lot is available for sale by sending an event.

- The manufacturer will not be eligible to deploy the smart contract for the drug Lot unless it is approved by the FDA but for the sake of simplicity, this approval is not implemented in the smart contract.
- Figure 4 illustrates the relationship among the different entities with the smart contract.



Function calls and events for two different scenarios for Lot sale

- The buyer initiates the process by executing the buyBox function with the number of boxes needed and can result in two outcomes. This scenario usually happens in a real pharmaceutical supply chain between

- If the transferred amount does not match the price of the boxes it will result in failure of the transaction.

- However, if the transferred amount is exactly equal to the price of the requested boxes, the price of the boxes will be transferred to the seller.
- Moreover, the number of boxes owned by the two entities will be updated according to the quantity purchased.

- Finally, the transaction will be finalized and declared successful to all participants. The following are the main functions with their corresponding algorithms.

Creating a Lot: Algorithm 1 explains the steps in creating a Lot. The inputs to the smart contract needed by the functions are shown with their descriptions.

- The function executes only if the address of the caller is the same as the address of the ownerID.
- If the caller is granted access, he/she will have the authority to update the fields in Algorithm 1.
- Once all fields have been updated the two events will update the status as shown in Algorithm 1.

Buying Lot: Algorithm 3 describes the transactions between the buyer and the seller of the drug Lot.

- It requires the caller of the function (buyer) to not have the same address as seller (to ensures Lot owner doesn't buy his own Lot) and requires the transferred amount to be exactly equal to the Lot price.

Buying Lot Boxes: Algorithm 4 is similar to Algorithm 3, with subtle difference.

- The initiation of this algorithm is similar to Algorithm 3 but the buyer is required to specify the exact number of boxes within the Lot.
- The amount transferred by the buyer has to be exactly equal to the number of boxes the buyer wants to buy multiplied with the price of each box.
- The main difference here is that there is mapping for the addresses of the buyers with the number of boxes purchased, and the mapping gets updated every time this function gets executed successfully.

Algorithm 1: Creating a Lot in Smart Contract

Input: lotName, lotPrice, numBoxes, boxPrice, IPFShash, Caller, OwnerID
Output: An event declaring that the Lot has been manufactured
An event declaring that the image of the Lot has been uploaded
Data:
lotName: is the name of the Lot
lotPrice: is the specified price of the Lot
numBoxes: is the total number of boxes within a Lot
boxPrice: is the price of each box within a Lot
IPFShash: is the IPFS hash of the Lot image
ownerID: is the Ethereum address of the owner of the Lot
initialization;
if *Caller == ownerID* **then**
  Update *lotName*
  Update *lotPrice*
  Update *numBoxes*
  Update *boxPrice*
  Add *IPFShash*
  Emit an event declaring that the Lot has been manufactured
  Emit an event declaring that the Lot image has been uploaded to the IPFS server
else
  Revert contract state and show an error.

Algorithm 2: Granting Lot Sale

Output: An event declaring that the Lot is for sale
initialization;
if *Caller == ownerID* **then**
  Emit an event stating that the Lot is up for sale
else
  Revert contract state and show an error.

Algorithm 3: Buying Lot

Input: ownerID, Buyer, Seller, Transferred Amount, lotPrice
Output: An event declaring that the Lot has been sold
Data: *ownerID*: The Ethereum address of the current Lot owner
Buyer: The Ethereum Address of the buyer
Seller: The Ethereum Address of the Seller
Transferred Amount: The amount transferred to the function
lotPrice: The price of the Lot
initialization;
if $Buyer \neq Seller \wedge TransferredAmount = lotPrice$ **then**
  Transfer the price of the Lot to the seller
  Update ownerID by replacing the seller Ethereum address to the buyer Ethereum Address
  Emit an event declaring that the Lot has been sold
else
  Revert contract state and show an error.

- Every drug Lot is manufactured with a smart contract that is specifically designed for it and is responsible for triggering events and logging them on the ledger.

- A unique Ethereum address is generated for every drug Lot. However, copying Ethereum address of each drug is cumbersome, time consuming, and error prone process.

- Therefore, a QR code is used which can be easily scanned using smartphones.

- A QR code is a two-dimensional barcode that is readable by smartphones, and it can allow encoding over 4000 characters in a two dimensional barcode.

- Mapping an Ethereum address to a QR code can be done by using an Ethereum QR code generator in which the Ethereum address is passed and a unique QR

code is generated which will exclusively map to that Ethereum address every time it gets scanned.

- Once the QR code gets attached to the drug Lot, it can be dispensed to patients.
- Figure 7 illustrates the steps to verify the authenticity of a drug.
- The first step is scanning the QR code that is attached to the drug by using a DApp which interacts with the Ethereum node (local or remote node) through web3j.
- To map the QR code to its corresponding Ethereum address, the DApp has to interact with the Ethereum node (Infura for example) through JSON-RPC.
- The Ethereum node has a replica of the ledger, and it is extremely important for the users because it makes the process smooth and easy by saving them the effort of having to set up their own Ethereum node which takes a lot of time

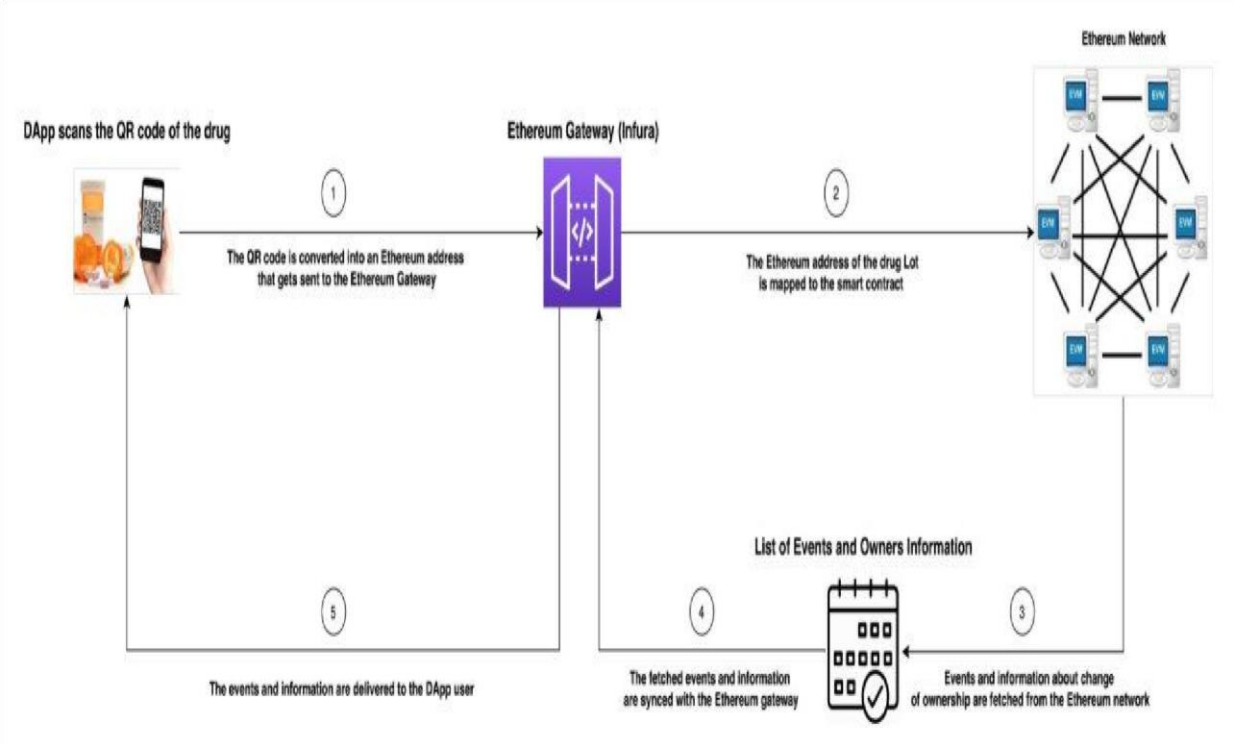


FIGURE 7. Application use case of the proposed blockchain-based solution

Algorithm 4: Buying Lot Boxes

Input: ownerID, Buyer, Seller, Transferred Amount, boxPrice, numBoxes, numBoxesToBuy, Transferred Amount, boxesPatient

Output: An event declaring that the Lot boxes have been sold

Data: *ownerID*: The Ethereum address of the current Lot owner
Buyer: The Ethereum Address of the buyer
Seller: The Ethereum Address of the Seller
Transferred Amount: The amount transferred to the function
boxPrice: The price of the Lot box
numBoxes: The total number of boxes in the Lot
numBoxesToBuy: The number of boxes the buyer wants to buy
boxesPatient: Maps the number of boxes bought to the buyer address
initialization;

if $Buyer \neq Seller \wedge TransferredAmount =$ *numBoxesToBuy*boxPrice* **then**
    Transfer the price of the boxes to the seller
    Update ownerID by replacing the seller Ethereum address to the buyer Ethereum address
    Update numBoxes owned by the seller by decreasing the sold amount from it
    Update boxesPatient by assigning the purchased amount to the buyer address
else
    Revert contract state and show an error.

TESTING AND VALIDATION:

In order to asses the smart contracts developed via Ethereum, Remix IDE in-browser developing and testing environment was used to test and validate different functions.

The scenarios involved three different participants and their corresponding Ethereum Addresses as presented in Table 3.

We further present the transactions and logs of the smart contract's functions below

	Ethereum Address
Participant1	0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c
Participant2	0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C
Participant3	0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB


```
"from": "0x5e72914535f202659083db3a02c984188fa26e9f",
"topic": "0x44c99ce1ec0af6519400dc5641e20fd507c596f90096ffe116181619d7ab1a25",
"event": "lotManufactured",
"args": {
        "0": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
        "manufacturer": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
        "length": 1
```


FIGURE 8. Successful execution of lotDetails Function

TABLE 3. The Ethereum address of each participant in the testing scenario

- **lotDetails:** In this function, it was tested whether the current owner of the smart contract is able to add the details of a newly manufactured Lot such as the Lot name, Lot price, number of boxes within the Lot, and the price of each box. A successful execution of the function and its corresponding logs and events are displayed in Figure 8.

```
"from": "0x5e72914535f202659083db3a02c984188fa26e9f",
"topic": "0x15a51b79663b36aa87b7e256eddbad58070b43d374c4294e41b9e76ad43a4c04",
"event": "lotSale",
"args": {
        "0": "Aspirine",
        "1": "200",
        "2": "1000000000000000000",
        "3": "100000000000000000",
        "_lotName": "Aspirine",
        "_numBoxes": "200",
        "_lotPrice": "1000000000000000000",
        "_boxPrice": "100000000000000000",
        "length": 4
```


FIGURE 9. Successful execution of grantSale Function

- **grantSale:** The grantSale function has a simple task yet it's very important, it basically notifies all the entities that the manufactured Lot is currently for sale. A successful execution of the function is given in Figure 9.


```
"from": "0x5e72914535f202659083db3a02c984188fa26e9f",
"topic": "0xeb373dc4c684e4ae6135618e7fc15d654b409d8071dc8126b4a5d18ac86590db",
"event": "lotSold",
"args": {
        "0": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
        "newownerID": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
        "length": 1
```


FIGURE 10. Successful execution of buyLot Function

- **buyLot**: In this function, Participant2 (Table 3 shows the corresponding Ethereum address) is used to buy the Lot from Participant1. Participant1 has specified the

correct amount of ether (which is 1Ether as shown in Figure 8) to transfer and the successful execution of the function is shown in Figure 10.


```
"from": "0x5e72914535f202659083db3a02c984188fa26e9f",
"topic": "0x82c28ddbad097bd1003a55cdb6788f38fbe3033fa91c813a8a00652716c0d45b",
"event": "boxesSold",
"args": {
        "0": "50",
        "1": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
        "_soldBoxes": "50",
        "newownerID": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
        "length": 2
```


FIGURE 11. Successful execution of buyBox Function

- **buyBox:** This function deals with transactions related to purchase of specific number of boxes from the Lot (usually happens between a patient and the pharmacy). Figure 11 shows a successful execution of this function where Participant3 purchases 50 boxes from Participant2. The price of the boxes has been selected arbitrarily and they may not be logical but the purpose here is to confirm that the execution of the functions properly.

DISCUSSION AND EVALUATION:

In this section, we discuss generalization of the proposed Ethereum blockchain-based solution, present cost and security analysis for drug traceability in supply chain, and discuss blockchain limitations in supply chains.

A. GENERALIZATION:

- The proposed work in this paper demonstrates how blockchain technology can be applied for drug traceability in a pharmaceutical supply chain.

Although the functions in the smart contract were defined in a way that fits the pharmaceutical supply chain specifically, it can be easily extended to other types of supply chains.

- The main difference between the pharmaceutical supply chain and any other supply chain is the products/items that are being shipped, distributed, and sold and the way they are handled throughout the process.

- For example, some pharmaceutical drugs require very specific conditions like temperature and humidity while they are being transferred from a point to another whereas a spare part supply chain for example would have very different conditions.

- The entity relationship diagram can be also modified, for example, if a supply chain has an application that requires the use of more than one parent smart contract then it will have to be added and define its relationship with the other entities.

- Another possibility is the creation of more than one product at a time which requires an extension to the functions to accommodate the additional products, and this can be achieved by modifying the existing smart contract.

- Finally, the defined algorithms follow simple and easy to grasp steps, and similar algorithms are followed in many other supply chains.

- This fact can be used to adjust the customize the algorithms used in this paper to fit the needs of specific supply chain application.

B. COST ANALYSIS:

- This subsection presents cost analysis of the Ethereum smart contract code and the function calls.

- When a transaction is executed on the Ethereum blockchain, it costs gas to send it to the Ethereum blockchain.
  Remix IDE is a very useful and easy to use tool to estimate the gas costs for the execution and transaction which are the main types of gas costs.

- The execution cost is the cost of executing different functions in the smart contract whereas the transaction cost deals with several factors such as the deployment of the contract, and any data that gets sent to the blockchain network.

- Table 4 shows the gas costs of the different functions used in the smart contract, and it also shows the costs converted into fiat currency (USD).

TABLE 4. Gas Costs of the Smart Contract Functions.

Function Caller	Function Name	Transaction Gas	Execution Gas
SC Owner	lotDetails	107356	83844
SC Owner	grantSale	29745	8473
Buyer	buyLot	40845	19573
Buyer	buyBox	62305	40841

Table 3 presents that the cost in USD is very minimal for all four functions.

- The function that costs the most is the lotDetails which is executed by the smart contract owner (manufacturer).
  This relatively high cost can be explained due to changes in five different variables in the function which requires storage.

- On the other hand, grantSale function costs the least, as this function only broadcasts an event to notify the participants that the Lot is available for sale.

C. SECURITY ANALYSIS FOR THE BLOCKCHAIN-BASED HEALTHCARE SUPPLY CHAIN:

In this subsection, we discuss briefly the security analysis of the proposed blockchainbased solution for the healthcare supply chain where integrity, accountability, authorization, availability, and non-repudiation are considered as key security goals. Moreover, we discuss how our solution is resilient against common attacks including Man-In-The-Middle(MITM) and Distributed Denial of Service (DDoS).

- **Integrity:** The primary objective of the proposed blockchain solution is to keep track of all the transactions that occur within the healthcare supply chain ensuring traceability of the history of the Lots, ownership transfers and their corresponding boxes.

- This is ensured in the proposed solution because all events and logs are stored in the immutable blockchain ledger.
- Moreover, the use of IPFS to store images of the manufactured Lots adds integrity to the proposed solution.
- This will ensure that every transaction within the healthcare supply chain can be tracked and traced.

Accountability: As demonstrated in section V, each execution of a function has the Ethereum address of the caller stored on the blockchain which means tracing the function caller is always possible.

- Therefore, all the participants are accountable for their actions. In the healthcare supply chain, the manufacturer will be accountable for any drug Lot he produces using the lotDetails function and pharmacies will be accountable for any prescription they give to a function because buyBox function will show where each patient is getting the drugs from

Authorization: The critical functions in the smart contract can only be executed by authorised participants by using the modifier.

- This ensures protection against unprivileged access and prevention of any unwanted entities from using the implemented functions.
- This is very important for the healthcare supply chain because the manufacturing of the drug Lot should only be done by a verified manufacturer and the prescription of drugs should be only done by a verified pharmacy.

Availability: Blockchains are decentralized and distributed by nature.

- Therefore, once the smart contract is deployed on the blockchain, all logs and transactions are accessible to all participants.
- Contrary to centralized approaches, the transaction data is stored at all participating nodes therefore loss of a node does not result in the loss of transaction data.
- The blockchain network needs to be up and running all the time for the application of healthcare supply chain to be successful. Any downtime might result in delays that are very costly in the healthcare industry.
- **Non-Repudiation:** As transactions are cryptographically signed by the private key of their initiators, cryptographic properties of PKI guarantee that private keys cannot be deduced from public keys.
- Therefore, a transaction signed by a specific private key can be attributed to the owner of the key. This is similar to accountability where the participants of the blockchain-based healthcare supply chain cannot deny their actions since they are already signed by their private key which is associated with their real identity.
- **MITM Attacks:** Every transaction in the blockchain needs to be signed by its initiator's private key, and therefore if an intruder tries to modify any of the original data and information in the blockchain it will not be confirmed unless it gets signed by the initiator's private key.
- Therefore, MITM attacks are not possible in the blockchain environment. This feature is indispensable for the application of healthcare supply chain because it

ensures that only the verified entities can perform actions within the supply chain, and intruders who illegally try to produce counterfeit drugs in the name of a verified manufacturer will no longer be able to that.

D. SMART CONTRACT SECURITY ANALYSIS:

- The developed Ethereum smart contract for drug traceability was analyzed using specialized tools to reveal any code vulnerabilities in addition to the aforementioned security analysis.

- Those tools were used in code development iterations to improve the reliability of the smart contract.

- Remix IDE that was used to develop the smart contract provides some code debugging and run-time error warnings.

- However, they are not sufficient to establish trust in the smart contract robustness.

- Therefore, SmartCheck was used to detect vulnerabilities in the code at different severity levels.

- After multiple iterations of smart code modification, the smart code was bug-free as reported by the output.

- SmartCheck analyzed the smart contract comparing it to its knowledge base and verified that it was free from risks that would make it susceptible to exploitation and cyber-attacks.

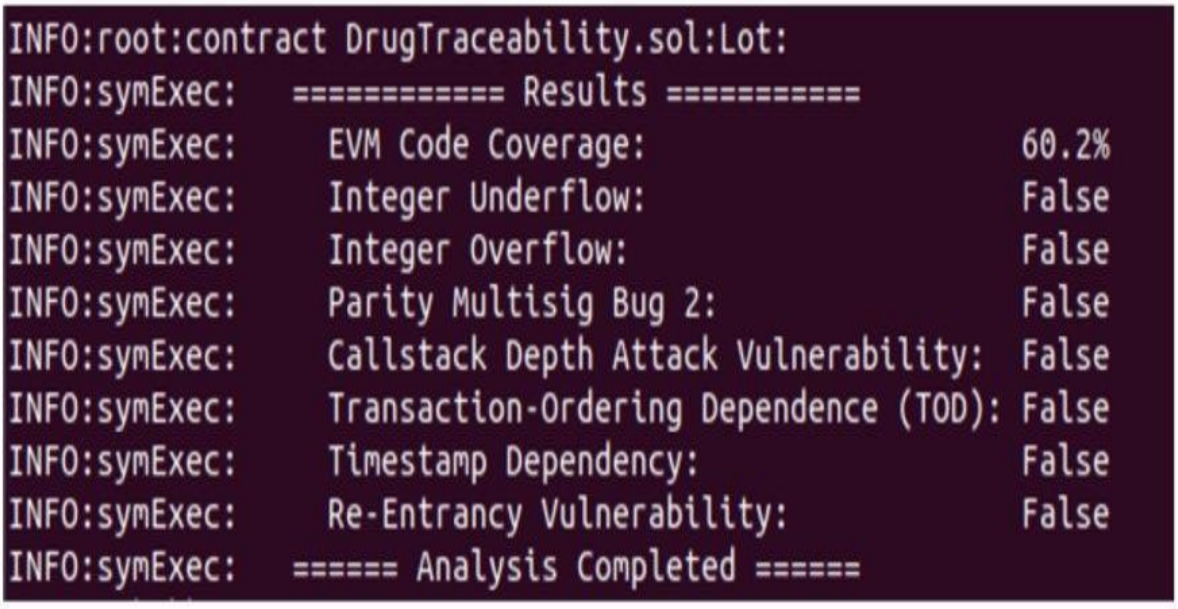


FIGURE 12. Smart contract vulnerability analysis

E. BLOCKCHAIN LIMITATIONS IN HEALTHCARE SUPPLY CHAINS:

Although the proposed system leverages prominent benefits of blockchain technology, there are number of potential limitations which should be highlighted to aid deeper understanding of their potential impact on the proposed system. We present a discussion of such potential limitations of blockchain in healthcare supply chains below.

Immutability: Blockchains are immutable where any information appended to the ledger cannot be altered or removed.

- While this can be beneficial for data integrity, it presents a major challenge, there is no way to correct inaccuracies on a blockchain because they are immutable.
- For example, the operators conducting the physical tasks in the drug supply chain can still make errors when recording information to the ledger.
- Consequently, these errors cannot be corrected even if it’s detected. In a healthcare supply chain, this can have unwanted consequences.

- For example, if the manufacturer inserts wrong details of a drug Lot, it can cause issues later on when it reaches the pharmacy where a pharmacist might incorrectly prescribe a drug to a patient.

Data Privacy: Although immutability is considered one of the main advantages of blockchains, it can be in conflict with emerging laws that address information storage issues.

- For example, the General Data Protection Regulation (GDPR) in Europe requires that organizations accurately control where and how data is stored because the person it is collected from have the right to modify or delete it any time, and if actions are not taken according to their requests, the organization can be liable to heavy fines [50]. In healthcare supply chains, patients might refuse to have their data stored permanently on the blockchain and they can legally sue the healthcare center.

Scalability: Blockchain requires individual nodes to process every transaction on the entire network which provides security and verifiability to the system, but it limits scalability.

- However, there is active research to address this challenge.
- For instance, Sharding and Plasma are two scaling solutions for Ethereum that would eliminate the need for every Ethereum node to process every transaction on the network.
- In healthcare supply chains, this might not be an issue if the manufacturing is done for small to moderate quantities.
- However, if a drug is being manufactured in large scale, the process will be difficult and very slow

Interoperability: Blockchain networks other than Ethereum work in their own unique way which leads to interoperability issues where the different blockchains are not able to communicate with each other.

- If a unified blockchain-based solution is used among healthcare centers, this problem can be avoided.
- However, if healthcare centers decide to use different blockchainbased solutions with different platforms, it will be very difficult to make them interoperable.

Efficiency: The efficiency of the blockchain solution is highly dependent on the coding of the smart contract and also the consensus algorithm used to verify and confirm a transaction.

- The former determines how costly the implementation and execution process will be, and the latter determines the energy consumption level.
- The healthcare supply chain involves many transactions, therefore it's very important for the smart contract to be coded properly so that it executes quickly and efficiently.