A blockchain system to empower healthcare accuracy rating

Chongdan Pan, Karen Chan, Sujit Lakshmikanth, Matteo Di Mario

Problem Statement And Solution

Problem and Goal:

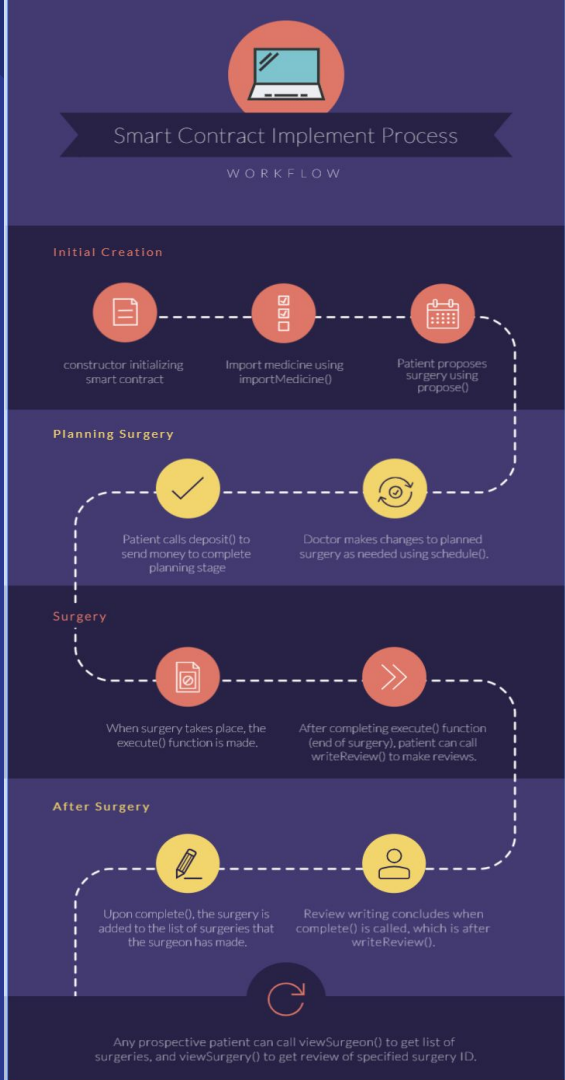
- Improving the rating system of the healthcare environment
- The reputation of surgeons and physicians is in the hands of reviewers, which authenticity is hard to identify
- Currently impossible to automate the determination of whether a patient has actually gone through the surgery process
- Unfeasible to check whether the patient is verified on a person by person case

Solution:

- Using blockchain technology to ensure the accuracy of customer reviews, verifying the authenticity of reviews
- Building a blockchain-empowered system that can record securely the surgeries that happened and allow customers to make reviews that accurately reflect the experience with the surgery

Our Progress

- Created an implementation of sample smart contract
- Currently done in four major steps:
  - Initial Creation
  - Planning Surgery
  - Execution of Surgery
  - After Surgery
- Current payment system using sample tokens.



What we've learned

Strong Contention

- High gases to get instruction executed
- 0.05 ETH (150USD) for a simple transfer

Execution Order

- MVE: Miner may switch the order of execution in one block to gain profits
- Atomicity: No concurrency issue for one contract

Design Considerations

- Execute as less instruction on the blockchain as possible
- Specific data location

```
function execution(uint256 surgeryId, string memory information)
public {
    Surgery storage surgery = _surgeries[surgeryId];
    // After the confirmation, the doctor execute the surgery
    and upload related information to the blockchain
    require(msg.sender==surgery.surgeon&&surgery.
    surgeryState==State.CONFIRMED);
    surgery.information = information;
    surgery.surgeryState = State.EXECUTED;
    _medicineUsage[surgery.medicineId] = surgeryId;
    emit SurgeryUpdate(_surgeryId, surgery.patient, msg.sender,
    State.EXECUTED);
}
```

Thank you

“I tried to make a joke using SHA-256, but people are still trying to figure it out.”