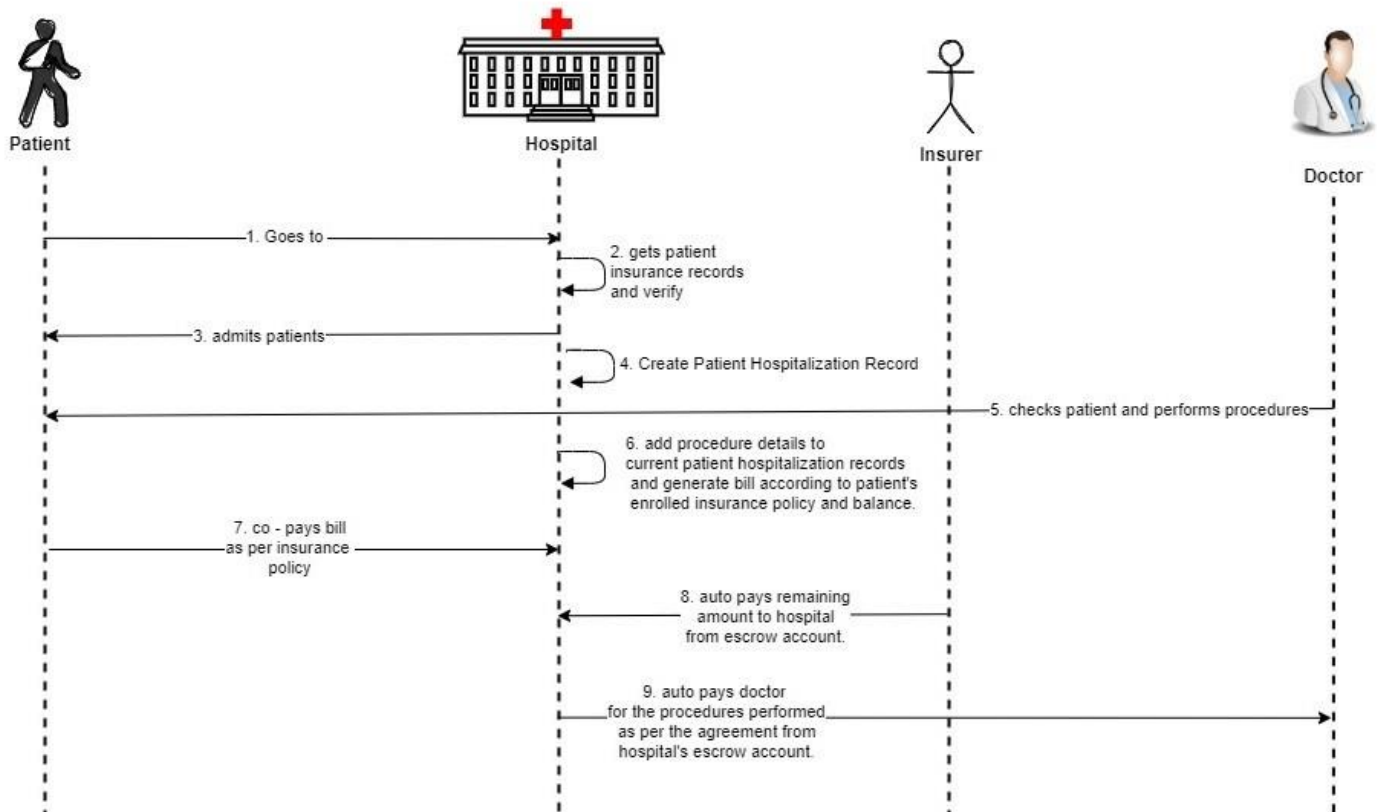


MediC: Medical Settlement Ecosystem

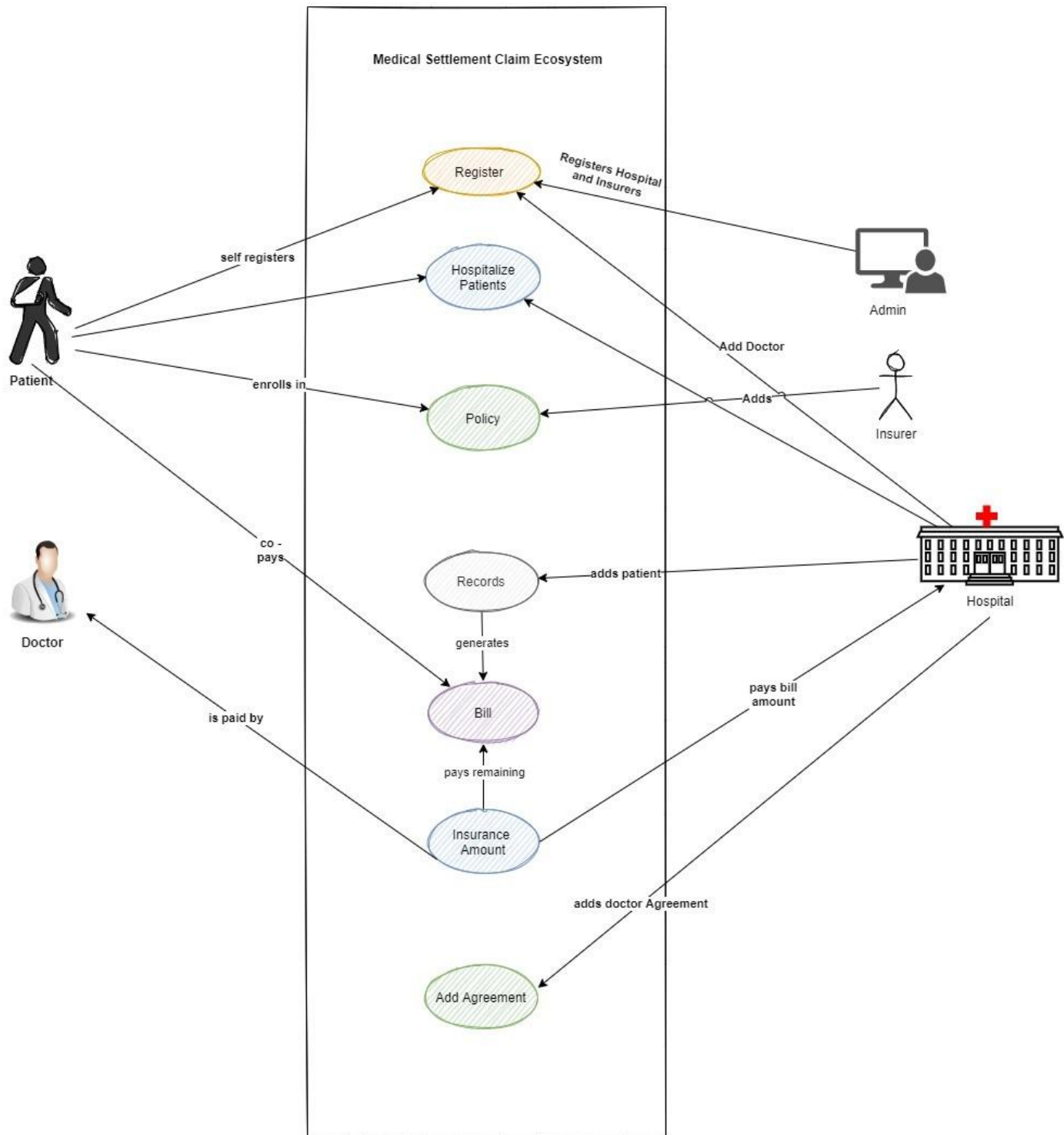
Repo URL: <https://github.com/vchaurasiya95/MediC-The-Medical-Payments-Ecosystem>

Sequence Diagram:



- Patient goes to the hospital and the hospital fetches patient records and verifies his insurance policy and admits him.
- The doctor working in the hospital checks the patient for illnesses and performs medical procedures.
- The hospital adds these costs for performing medical procedures to the patient's record.
- The hospital then generates the patients' medical bills.
- The patient pays the co-pay part of his bill himself to the hospital and the remaining part is automatically paid by the insurer as per the patient's insurance policy.
- The hospital then pays the doctor his share of the amount for performing medical procedures.

Use Case Diagram:



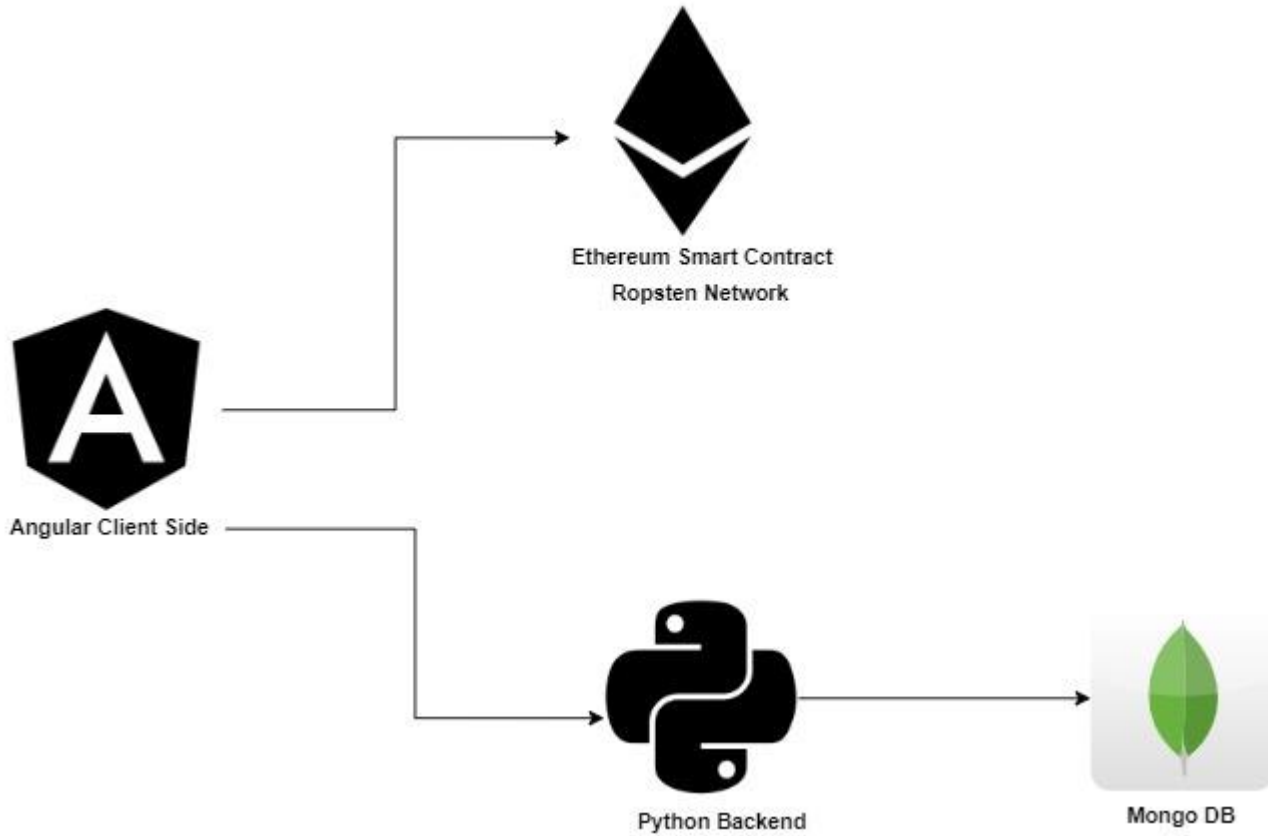
Contract Diagram

CONTRACT : ERC 20
totalSupply
balanceOf
allowance
transfer
approve
transferFrom
Transfer
Approval

Data Store
MODIFIERS
onlyAdmin
onlyHospital
onlyInsurer
onlyPatient
onlyDoctor
onlyHospitalOrInsurer
FUNCTIONS
requestTokens
addEscrowBalance
replenishContractTokens
addNetworkParticipants
addDoctors
addPatients
addProcedureType
getProcedureType
getUserType
replenishEscrowBalance
addPolicy
viewPolicy
enrollPolicy
updatePatientPolicyBal
getPatientPolicy
addProcedureCost
getProcedureCost
addHospitalDoctorAgreement
getDoctorAgreement
getHospitalAgreement
addHospitalizationRec
getPatientHospitalizationRecords
getPatientHospitalizationDetails
addProcedure
getProcedure
getProcedureDetails
getInsuranceClaim
settleDoctorPayment
Fields
Struct HospitalizationRecord(hospital_id, patient_id, timestamp, settled?)
Struct Policy(coverage[], max_value, meta_id, cost, provider)
Struct Agreement(part1, part2, timestamp)
Struct ProcedureDetails(doctor, procedure_id)

Application Architecture

Application Architectue Overview



The above diagram illustrates Application Architecture:

1. Python along with MongoDB server provides the capability to store and retrieve off-chain data through REST end point.
2. The client side is connected to Smart contract by using the web3 instance injected by the Metamask Wallet.