



# Blockchain Electronic Health Record

JUNAID QAMER  
CHRISTOPHER ROBINSON  
NATHALIE KELLEY

# WHAT IS EHR (ELECTRONIC HEALTH RECORD)?

- A digital form of a medical record.
- EHRs have actually been in use since the 1970s but were limited to large hospital systems as they were costly to maintain.
- With the passage of the Patient Protection and Affordable Care act on 6/28/2012. All medical providers were mandated to transition from paper charts to digital medical charts.
- Originally the thought was to have a unified charting system that could be accessed anywhere in the country by any medical provider. Of course, that's not what happened.
- By most estimates there's currently 500 EHR vendors and none of them communicate with each other.

# ISSUES WITH THE STATUS QUO

- Too many vendors.
- Lack of communication and uniformity between each EHR vendor.
- Patient's lack of control and access to their health record.
- If you get sick outside of your residence, it is difficult for outside providers to have access to health records.
- Mom and Pop physician offices can't usually afford EHRs and it become a huge expense forcing them to join large health systems in order to turn a profit and stay afloat.
- Even with blocks put in place, health records can still be changed after being closed. Yes, it's recorded but they still can be changed.



# HOW CAN BLOCKCHAIN FIX IT?

- By decentralizing health records, there's no ownership of the records by any one vendor or provider.
- Patients keep their medical record in their blockchain wallet. They do not need to fill out forms or have to call providers for copies of their records.
- By creating a decentralized electronic record anyone anywhere can have access without having to contact for **Access** to server, as long as you have the appropriate credentials.
- Blockchain EHR can be more affordable to small fish providers, as the costly fees required to run and establish servers no longer apply.
- Blockchain can't be altered, therefore the medical record stays intact.



# IS ANYONE DOING THIS ALREADY?

Medicalchain is a British blockchain EHR which began in 2016. As of today, they mostly function in countries with less regulations of health records such as South America and eastern Europe. They have found it difficult to break into Europe and the U.S as most providers are fearful of risks taken with new technology.



# WHAT'S HOLDING YOU BACK? Are you scared of HIPPA?

- UMM yeah, I am scared of HIPPA and you should be too.
- The Health Insurance Portability and Accountability Act of 1996, commonly known as HIPAA, is a series of regulatory standards that outline the lawful use and disclosure of protected health information (PHI). HIPAA compliance is regulated by the Department of Health and Human Services (HHS) and enforced by the Office for Civil Rights (OCR).
- HIPPA was established to protect your health records and financial health records. It provides ACCESS to patient's to their own medical records, as well as dictating who this information can be shared with.
- It also sets standards of how private health information can be shared, transmitted, and handled.
- Bad things happen to those who mishandle PHI.

**JUNAID**

# WHAT REQUIREMENTS MUST BE MET?

- PATIENT SAFETY- Everything needs to be reviewed, nothing can be dropped, FOLLOW UP.
- DATA SHARING- The ability to share data easily.
- DOCUMENTATION- Everything needs to be well documented and signed off.
- PRIVACY AND SECURITY- Everything done in an EHR needs to be tracked. This is a huge requirement by HIPPA.
- PATIENT ENGAGEMENT- Patient have to be involved with their care and kept up to date with everything pertaining to their health.



# HOW DOES OUR BLOCKCHAIN EHR MEET THESE?

1. **PATIENT SAFETY**- Through the emitting events- flags and notifications.
2. **DATA SHARING**- Data will be shared by sharing a private key. Patient have read only capabilities and their medical record will sit in their wallets but providers will use the private key to create and deploy a contract/office visit. Private keys serves the same function as a medical record number or even a social security number.
3. **DOCUMENTATION**- This fits in with privacy. Records can't be tampered with once the block is closed.
4. **PRIVACY & SECURITY**- Blocks are secured through cryptography. Each participant has their own keys/ digital signatures this would allow for easy transaction identification. Because each visit will be a block, once it is closed it can not be reopened and it keeps records from being tampered.
5. **PATIENT ENGAGEMENT**- Patient's are kept in the loop of their own health as their medical records are kept in their wallet versus in a server that patient's can't access.

# HOW WILL THIS WORK?

- Patients will own their health record in their personal wallet. Although, they will have view only capabilities to avoid health records from being altered.
- When a new provider visit is scheduled, patients will hand over their personal key to their wallet to their new provider.
- Providers will essentially begin a new contract or launch a new block in the patient's chain. (Each new visit is a block). (Providers have viewing and writing capabilities).
- To launch a new contract, patients will have a minimum amount of ETH in their wallet to be able to pay for their visit. (essentially there will be no outstanding bills in our future).
- After a visit is closed, a request will be made for payment to the patient's wallet.



# TEMPLATING A VISIT WITH A SMART CONTRACT

- DEMOGRAPHICS- Patient ID, name, DOB, relationship status, smoking status, gender, preferred pronouns, address, city, state, zip Code, family history, pre-existing conditions, and medication list.
- CHIEF COMPLAINT- Why are you here today?
- HPI (history of present illness)- What got you here today? What have you been doing thus far? Short history.
- Allergies, surgical history
- PHYSICAL EXAM
- VITALS
- LAB AND RADIOLOGY RESULTS
- ASSESSMENT AND FINDINGS: Conclusions and plans based on today's visit.
- TEST ORDERED- RX, lab test, radiology.
- COST OF VISIT



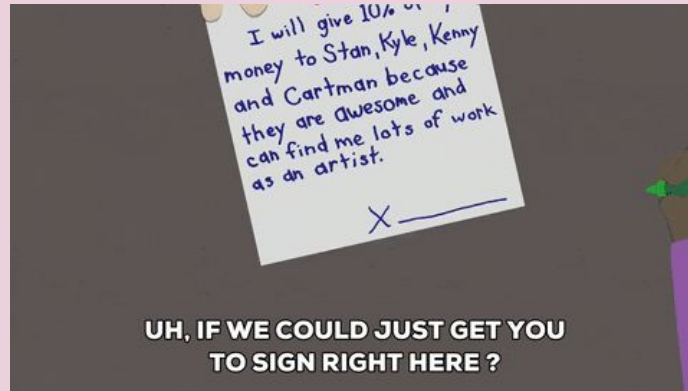
CHRIS

SOLIDITY CODE

THE CONTRACT IS BROKEN DOWN INTO 4 PARTS:

1. CREATING THE CONTRACT
2. FLAGS AND NOTIFICATIONS
3. FUNCTIONS
4. MAPPING

# CREATING A CONTRACT



```

pragma solidity ^0.5.5;
// A template for Doctor Visits
// Each visit is a contract that will be deployed onto the Blockchain
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/drafts/Counters.sol";

contract patientVisit {

    constructor() public {
        address payable provider = msg.sender;
    }

    uint charges;
    address payable patientWallet;
    string reasonForVisit;
    string finalAssessment;
    string visitSummary;
    uint[] labOrders;
    address[] referralAddresses;
    string referralNotes;
    address[] medicationList;
    string medicationNotes;

    using SafeMath for uint;

    using Counters for Counters.Counter;
    Counters.Counter private visitCount;

    //Only the provider can make edits and etc.
    modifier onlyAdmin{
        require(provider==msg.sender);
        _;}

```

Add Local file to the Browser Storage Explorer



# FLAGS AND NOTIFICATIONS



```
//Only the provider can make edits and etc.
modifier onlyAdmin{
    require(provider==msg.sender);
    _;}

//New patient added
event newPatient(string condition);

//Duplicate patient, increment visitcount.
event previousPatient(string condition);

//Labs ordered.
event labsOrdered(string condition);

// Referral ordered.
event referralOrdered(string condition);

//Medication ordered.
event medicationOrdered(string condition);

//Reschedule/ Follow-Up Notification.
event followUpNeeded(string condition);

//All charges paid/ Visit/Contract closed.
event visitClosed(string condition);

struct questionnaire{
    string patientName;
```

# FUNCTIONS

```
    }).done(function(response) {  
        for (var i = 0; i < response.length; i++) {  
            var layer = L.marker(  
                [response[i].latitude, response[i].longitude]  
                // ,{icon: myIcon}  
            );  
            layer.addTo(group);  
  
            layer.bindPopup(  
                "<p>" + "Species: " + response[i].species + "<br>" +  
                "<p>" + "Description: " + response[i].description + "<br>" +  
                "<p>" + "Seen at: " + response[i].latitude + " " + response[i].longitude + "<br>" +  
                "<p>" + "On: " + response[i].sighted_at + "</p>"  
            );  
        }  
  
        $('select').change(function() {  
            species = this.value;  
        });  
    });  
}  
$.ajax({  
    url: queryURL,  
    method: "GET"  
}).done(function(response) {  
    for (var i = 0; i < response.length; i++) {  
        var layer = L.marker(  
            [response[i].latitude, response[i].longitude]  
            // ,{icon: myIcon}  
        );  
        layer.addTo(group);  
    }  
});
```

*// Demographics will always be first thing reviewed.*

```
function addPatientDemographics(  
    bool _newPatient,  
    string memory _patientAddress,  
    string memory _patientCity,  
    string memory _patientState,  
    string memory _patientZip,  
    string memory _patientCountry,  
    string memory _patientPhoneNo) onlyAdmin public {  
    if (_newPatient == false) {  
        emit previousPatient('This patient has been previously seen here, this will be a duplicate patient. Look at previous bl  
        visitCount.increment();  
    }  
    else {  
        patientDemographics = demographics(_newPatient, _patientAddress,_patientCity,_patientState,_patientZip,_patientCountry,..  
    }  
}
```

```
function addPatientQuestionnaire(  
    string memory _patientName,  
    string memory _patientDoB,  
    string memory _patientGenderId,  
    string memory _patientRelationshipStatus,  
    string memory _patientSexuality,  
    string memory _patientPreferredPronoun,  
    bool _patientMarried,  
    bool _patientSmoker,  
    bool _patientAlcohol,  
    string memory _patientFamilyHistory,  
    string memory _patientPreExistingConditions,  
    string memory _surgicalHistory,  
    string memory _patientAllergies,  
    string memory _patientMedications) onlyAdmin public {
```

# MAPPING



```
mapping(address => uint) balances;

//Get patient wallet information.
function accessPatientWallet(address payable _patientWallet) public {
    patientWallet = _patientWallet;
}

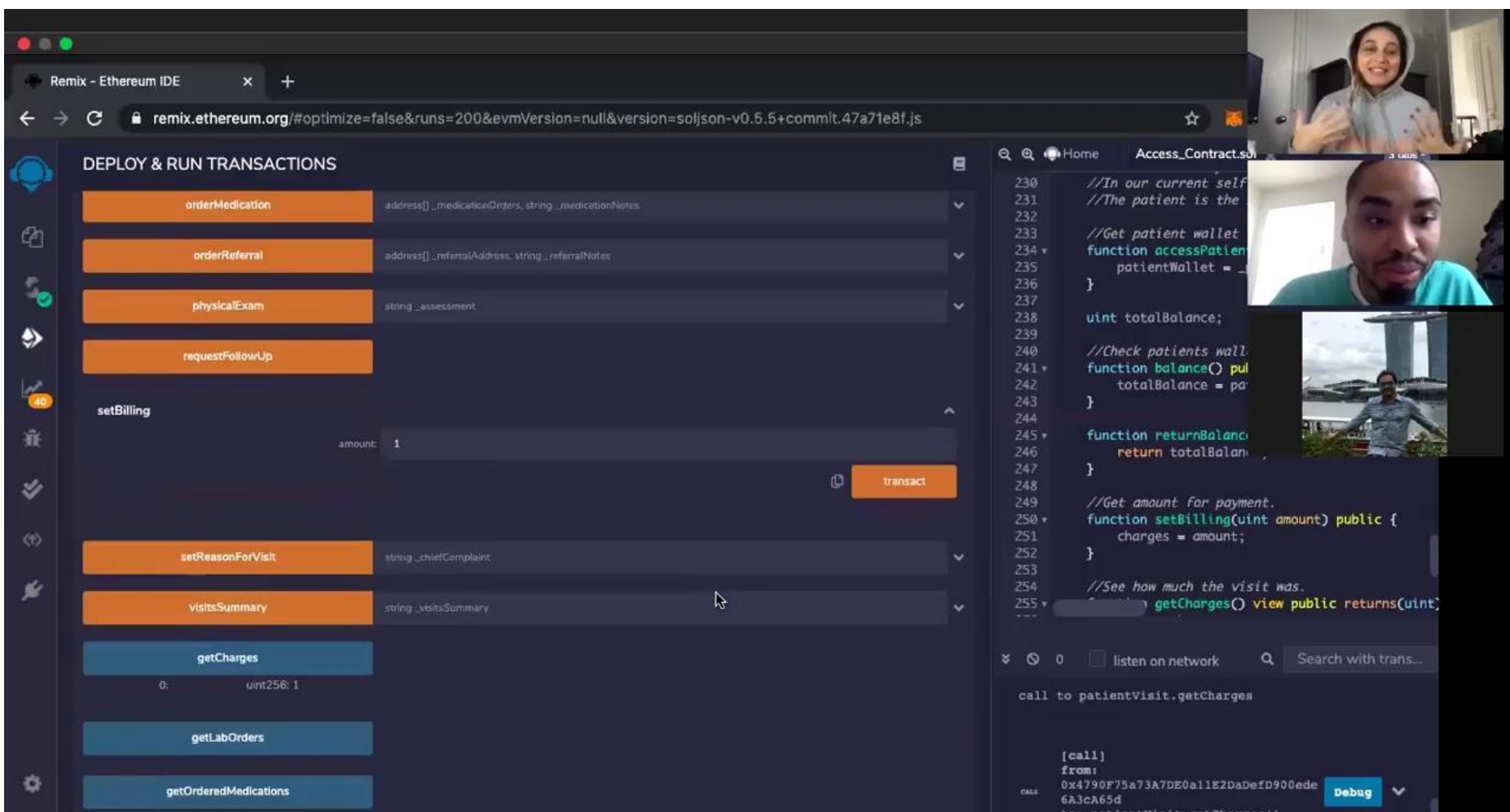
//Check patients wallet balance.
function balance() public view returns(uint) {
    return balances[patientWallet]; // Questions: Why is my balances not working.?
}

//Get amount for payment.
function setBilling(uint amount) public {
    charges = amount;
}

//See how much the visit was.
function getCharges() view public returns(uint){
    return charges;
}

//Take payment.
function completeBilling() public {
    balances[patientWallet] = balances[patientWallet].sub(charges);
    balances[provider] = balances[provider].add(charges);
    emit visitClosed('Everything has been charged and the billing has been completed. This visit is now closed.');
```

ACCESS B-EHR - DEMO





**JUNAID**

# THE FUTURE?



**In the future, we foresee insurances minting tokens to be given to patients depending on their health plan and/or health status. Billing issues will be a thing of the past and issues like surprise billing would be obsolete.**

- **Insurances mint tokens**
- **Tokens are given to patients based on a health plan.**
- **Patients pay providers directly with tokens received by their insurance.**
- **Insurance exchange provider tokens for working currency.**

# MINTING CONTRACT

A separate smart contract is used to generate tokens.

ERC 777 standard is used to add insurance companies as mint operators in the contract.

Each insurance company mint tokens (Access tokens: ACS) as needed according to predefined exchange rate.

Operator (insurance) transfer Tokens to each patient's wallet based on patient's health plans.  
Operators transfer ACS tokens in exchange of ETH/USD.

Patients use their tokens in each provider visit and make payments directly to the medical provider.

# QUESTIONS, CONCERNS, ADVICE?

