Project Health Genesis

Health IT Track #1: Identity Management and APIs

PRESENTED BY TEAM NUCLEUS HEALTH
MARCH 15, 2017
Medical Imaging procedures are involved in 28.7% of patient encounters.

7.7% of images were repeated within 90 days.

Connectivity reduced the odds of repeated imaging by 25%.

Industry is aggressively transitioning to cloud based image sharing systems.

Medical Imaging could be the catalyst to establishing a global network of trust for medical records.

Introducing Health Genesis

- Identity management via public key cryptography
- Authentication via digital signatures
- Authorization via smart contracts
- Bridges RESTful standard API's to blockchain

Health Genesis bridges existing Health IT investments into the new world of blockchain, global access and patient controlled data.
Use Case: Provider to Patient Sharing

1) Get Studies
   <https>
2) Create Contract
   <transaction>
3) Discover Contract
   <event>
4) Get Study
   <https>
5) Check Contract
   <query>

Architecture is generalized for sharing any URL based resource between any users – patients and providers.

Provider Intranet

ShareApp
   <Meteor>

AuthProxy
   <Meteor>

Orthanc
   <DICOMWeb>

PhrApp
   <Meteor>

Viewer
   <OHIF Viewer>

staff

patient
Use Case: Provider to Provider Sharing

1) Get Studies

2) Create Contract

3) Discover Contract

4) Get Study

5) Check Contract

Architecture is generalized for sharing any URL based resource between any users – patients and providers
Demo

Note: All code was written during the 24 hour code-a-thon period and available on github
We provide a complete Docker-based container deployment

- Health Genesis Applications
- Modified OHIF Viewer
- Private Ethereum Network (10 pre-built identities)
- Orthanc Image Archive
- Mongo Database

Works on Linux or Mac OS X
• Digital Wallets integrate with an application to prove a person’s identity
• Currently available solutions:
  – MetaMask (browser extension)
  – uPort (mobile app + APIs)
  – Mist (custom browser)
• Identity is established through proof of private key ownership – wallet signs a message using person’s private key, which can be validated by an application using the public key
• Application stores person’s blockchain address as user identifier
• Off-chain data is associated with blockchain address using traditional database
function signRequest() {
    // prompt user to sign
    var msSinceEpoch = new Date().getTime();
    var ctx = Session.get('ethereumContext');
    var contractAddress1 = ctx.wadoContract;
    var contractAddress2 = ctx.wadoRsContract;

    var message = contractAddress1 + contractAddress2 + msSinceEpoch;
    var msgHex = web3.sha3(message);

    web3.eth.sign(web3.eth.accounts[0], msgHex, function(err, result) {
        if(err) {
            console.log('error signing:', err);
            return;
        }

        var sig = hexSigToRSV(result);

        signedHeaders = {
            "x-secp256k1-r" : sig.r.toString('hex'),
            "x-secp256k1-s" : sig.s.toString('hex'),
            "x-secp256k1-v" : sig.v,
            "x-timestamp" : msSinceEpoch,
            "x-contractaddresses" : contractAddress1 + ',' + contractAddress2,
        }

        Session.set('signedHeaders', signedHeaders);
        userSigned.set(true);
        console.log(result);
    });
}
contract Resource {

  // The account that created this resource
  address public owner;

  // Base URL to a resource
  string public url;

  // The address of the recipient who has access to the resource. The recipient
  // can access the resource by signing the request using web3.eth.sign()
  // The resource server can validate the signature to ensure the
  // recipient owns the private key.
  address public recipient;

  // an identifier that can be used to group multiple resource smart contracts
  // together. In the case of DICOM, it could be the study instance uid
  string public resourceID;

  // the type of resource (e.g. wadouri, wadors)
  string public resourceType;

  // Event that is fired every time a resource smart contract is created.
  // Needed to reliably find resource smart contracts
  event ResourceCreated(address indexed recipient);

  function Resource(address _recipient, string _url, string _resourceID, string _resourceType) {
    owner = msg.sender;
    recipient = _recipient;
    url = _url;
    resourceId = _resourceId;
    resourceType = _resourceType;
    ResourceCreated(recipient);
  }

}
Authorization Checks

checkTimeStamp()
- prevents replay attacks

checkOwner()
- makes sure the smart contract was issued by this resource owner

checkRecipient()
- makes sure the caller is who they say they are
- make sure the caller is the recipient in the smart contract

checkUrl()
- make sure the URL matches what is in the smart contract
Thank You!

Durand Van Arnem
Matt Brinkman
Jason Klotzer
Chris Hafey
Jason Nye
Health Genesis!

Please [Register] or [Login]
Health Genesis ShareApp

Medical Record Search

Please enter the registration code provided by your doctor (use "topsecret")

**topsecret**

or [Login](#)
Staff authentication using digital signature

Healthcare ShareApp

Medical Records

Please enter the registration code:
topsecret

or Login

CONFIRM TRANSACTION

Sign Message

Signing this message can have dangerous side effects. Only sign messages from sites you fully trust with your entire account. This will be fixed in a future version.

Chris H...

ADDRESS 0x9332827e...0C31
BALANCE None

MESSAGE 0x4a29bc329589b6ea0e2024e91bd7d42a99b4

Cancel  Sign
Health Genesis ShareApp

Medical Record Search

QIDO URL: http://localhost:9042/dicom-web/studies

<table>
<thead>
<tr>
<th>Patient Name</th>
<th>TCGA-4Z-AA7M</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA-4Z-AA7M</td>
<td>TCGA-4Z-AA7M</td>
<td></td>
</tr>
</tbody>
</table>
Staff share to patient/provider identity

Health Gen

Hello, 0x5cab4C6b92c4266f535fe8da18092b0545c70AE3

Medical Record Search

QIDO URL: http://localhost:9042/dicom-web/studies

<table>
<thead>
<tr>
<th>Patient Name</th>
<th>Patient ID</th>
<th>Study UID</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA-4Z-AA7M</td>
<td>TCGA-4Z-AA7M</td>
<td>1.3.6.1.4.1.14519.5.2.1.6354.4016.325142651224193401620709391720</td>
</tr>
</tbody>
</table>

Share Exam

Recipient Address

0x5cab4C6b92c4266f535fe8da18092b0545c70AE3

Share Close
Health Genesis PHR

Please [Register] or [Login]
Patient authenticate via digital signature

Health

METAHR

CONFIRM TRANSACTION

1 of 3 ➔
Sign Message

Signing this message can have dangerous side effects. Only sign messages from sites you fully trust with your entire account. This will be fixed in a future version.

ADDRESS 0x5cab4C6b9...0AE3
BALANCE None

MESSAGE 0x9d10cd48598cf3c3d99f9b678d961ba4c2a8b

Cancel Sign
Hello, 0x5cab4cb92c4266ff535fe8da18092b0545c70ae3

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Block#</th>
<th>StudyUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed Mar 15 2017 01:08:43 GMT-0400 (EDT)</td>
<td>29</td>
<td>1.3.6.1.4.1.14519.5.2.1.6354.4016.325142651224193401620709391720</td>
</tr>
</tbody>
</table>
Patient authenticates using digital signature

MetaMask Notification

CONFIRM TRANSACTION

Sign Message

Signing this message can have dangerous side effects. Only sign messages from sites you fully trust with your entire account. This will be fixed in a future version.

ADDRESS 0x5cab4Cb9...0AE3
BALANCE None

ACCOUNT

MESSAGE 0x7562c64c4071c156dea6ef6b57e3a03b614d50

Cancel Sign
Patient accesses images