Blockchain Development
Week: 10
Title: Case Study

Dr Ian Mitchell

Middlesex University,
Dept. of Computer Science,
London

September 26, 2019

Lecture Aims

Aims

To gain a holistic view of a blockchain application by investigating the parts, then see how these parts combined add up to more than the whole.

Lecture Objectives

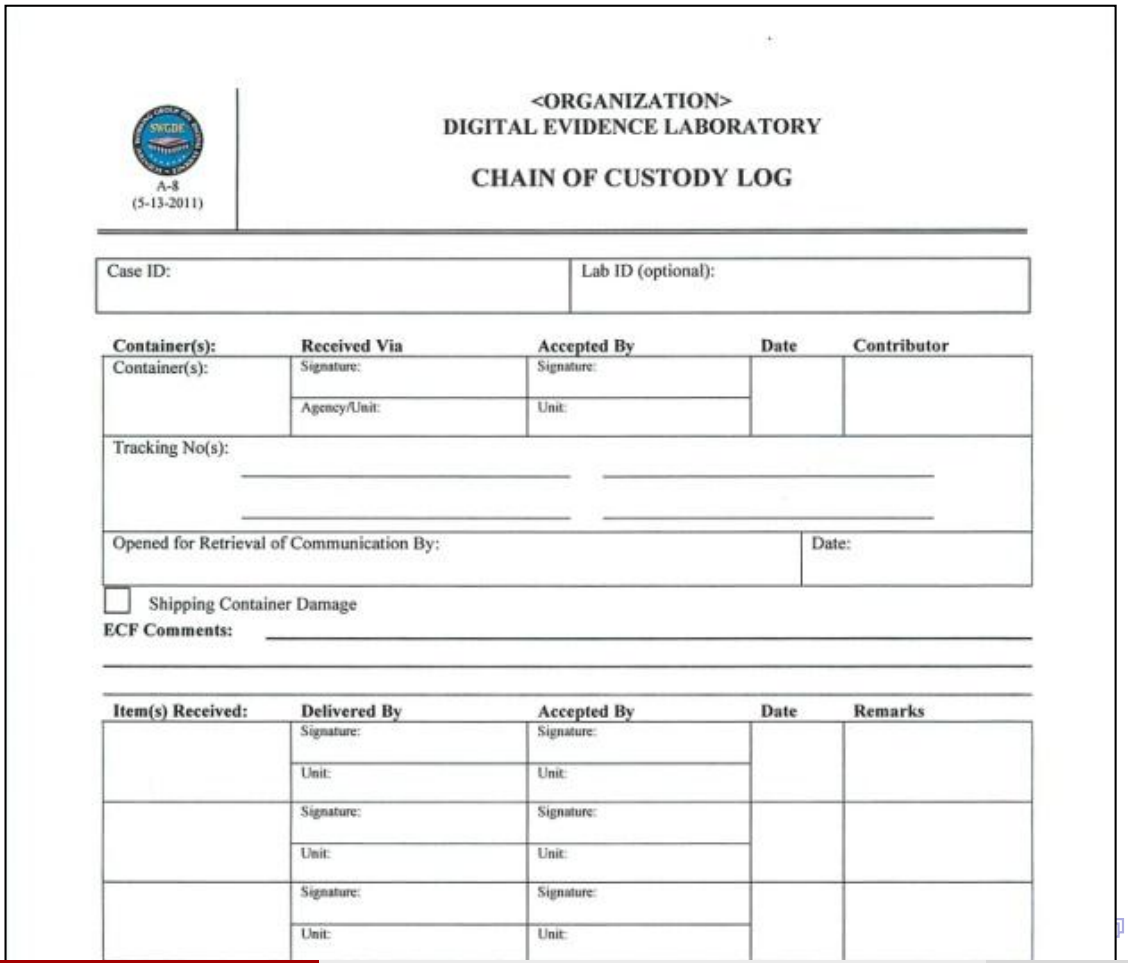
Case Study

Today's lecture will investigate the blockchain implementation of evidence management for a forensic lab, colloquially known as a chain-of-custody [1]

Chain of Custody [2]

Scientific Working Group on Digital Evidence

A-8: Chain of Custody Log



A-8
(5-13-2011)

<ORGANIZATION>
DIGITAL EVIDENCE LABORATORY

CHAIN OF CUSTODY LOG

Case ID:	Lab ID (optional):

Container(s):	Received Via	Accepted By	Date	Contributor
Container(s):	Signature:	Signature:		
	Agency/Unit:	Unit:		

Tracking No(s): _____ _____

_____ _____

Opened for Retrieval of Communication By:	Date:

☐ Shipping Container Damage

ECF Comments: _____

Item(s) Received:	Delivered By	Accepted By	Date	Remarks
	Signature:	Signature:		
	Unit:	Unit:		
	Signature:	Signature:		
	Unit:	Unit:		
	Signature:	Signature:		
	Unit:	Unit:		

Chain of Custody [2]

Scientific Working Group on Digital Evidence

Item(s) Received:	Delivered By	Accepted By	Date	Remarks
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature: Unit:	Signature: Unit:		
	Signature:	Signature:		

Chain of Custody[2]

Scientific Working Group on Digital Evidence

For Major Deviations Only:

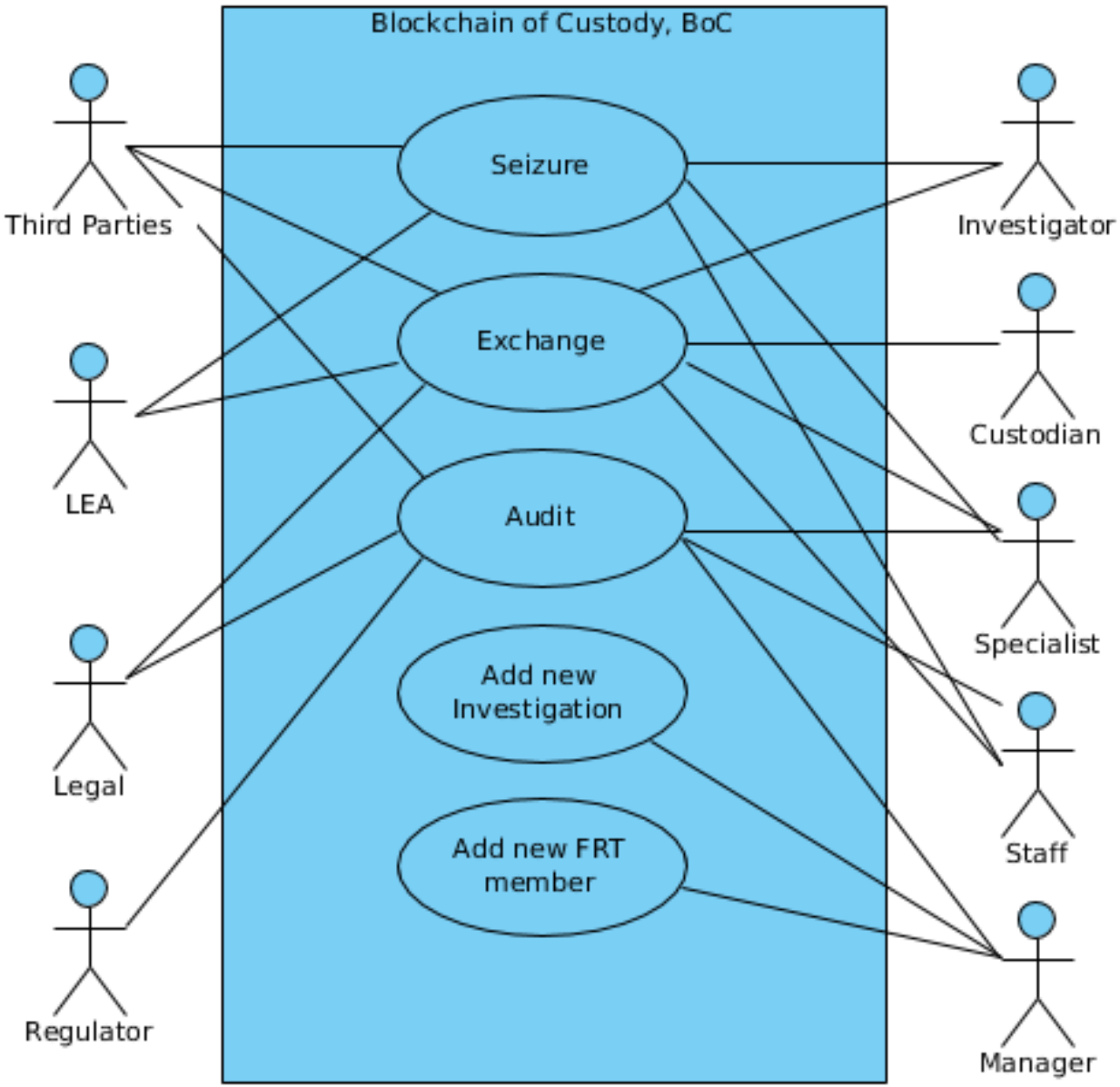
1st Reviewer Name and Title (Unit Supervisor or Laboratory Director):	
1st Reviewer Signature:	1st Review Date:
(Optional) Quality Assurance Reviewer Name, Signature and Date:	

Final Review and Approval (Unit Supervisor or Lab Director for Minor; Lab Director for Major):

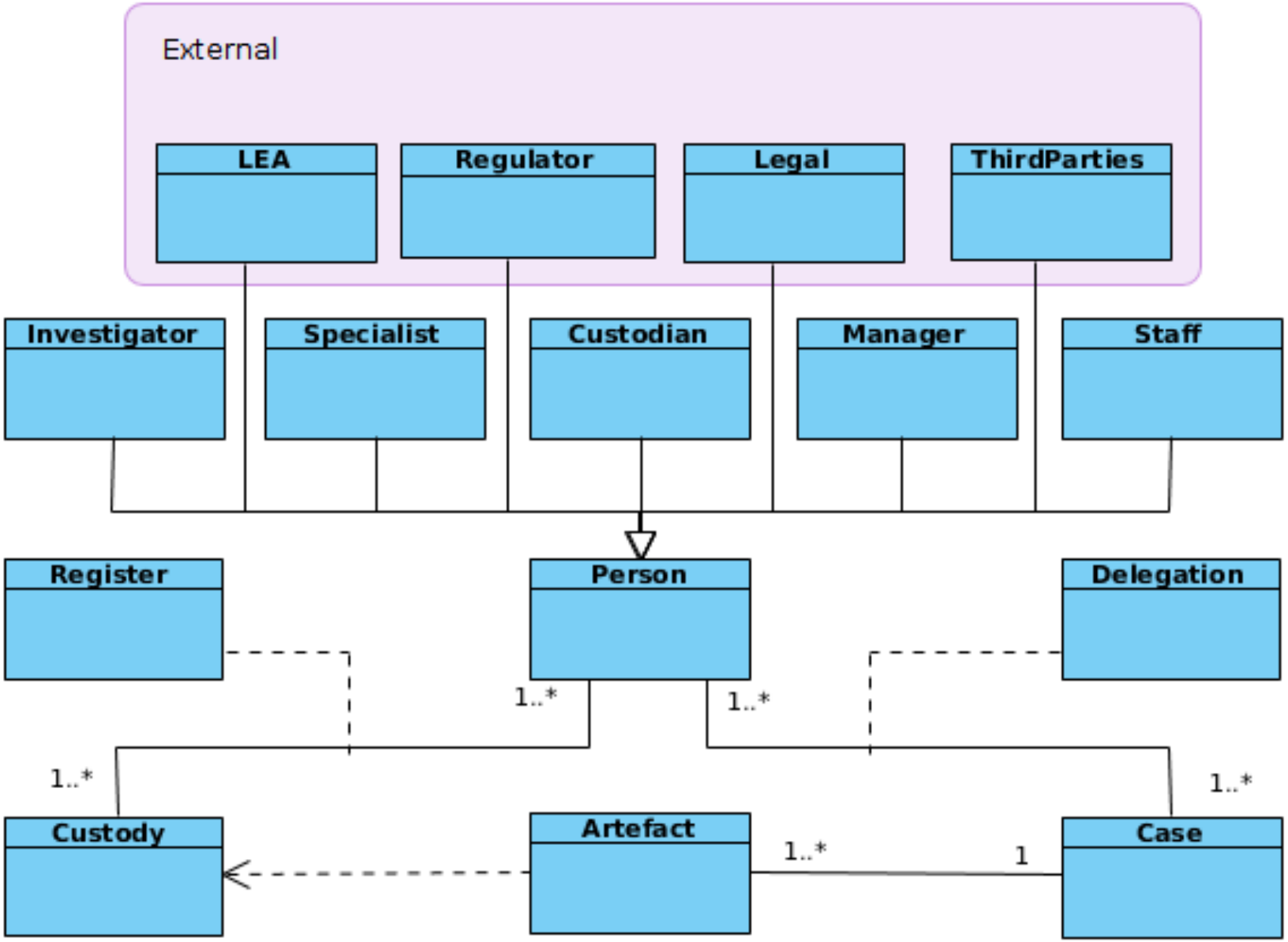
Reviewer Name and Title:	
Reviewer Signature:	Review Date:
Decision: ☐ Approved ☐ Denied	
Approval limitations:	

A copy of the completed form must be supplied to the Laboratory Quality Assurance Program Manager for both Minor and Major Deviations. Fax to <fax number>.

Use Case Diagram



Class Diagram



CTO I

Listing

```
1  /*
2  * Blockchain of Custody
3  * Prototype for all organisations involved in the transfer of evidence
4  * Single blockchain application, with the intention of becoming many inter-dependent
       applications
5  */
6  namespace org.boc.net
7  /*
8  * PARTICIPANTS
9  */
10 abstract participant Person identified by PIN{    // other personal details can be
       included
11   o String PIN regex = /P[0-9]{1,6}/              //PIN has format P9, P99, .., P999999
12   o String firstName optional                     // prototype firstName is optional
13   o String lastName optional                      // prototype lastName is optional
14 }
15 participant Investigator extends Person{}         // Investigating crime, e.g., DCI
16 participant Specialist extends Person{}          // Forensic Specialist, e.g., Coroner
17 participant Staff extends Person{}               // Other internal staff
18 participant Custodian extends Person{}           // Custodian of evidence
19 participant MLA extends Person{}                 // Mutual Legal Assistance
20 participant Manager extends Person{}
21 abstract participant Organisation identified by OID{ // other data can be added
22   o String OID regex = /B[0-9]{1,4}/        // B for Business, since O=0? unambiguous
       unique identifier
23   o String OrganisationName                 // name of organisation
24   o String Abbr                    // abbreviate name
25 }
```

CTO II

Listing

```
26 participant Regulator extends Organisation{}     // Regulator, e.g., FSR
27 participant Auditor extends Organisation{}       // Audit, e.g., for ISO17025
28 participant Legal extends Organisation{            // Legal Professionals external to
       organisation
29   o String email                                   // email - was candidate for unique
       key
30 }
31 participant TP extends Organisation{              // 3rd parties
32   o String description                            // how they are related to investigation,
         e.g., defence
33   o String email                                  // communication
34 }
35 participant LEA extends Organisation{
36   o String email optional                          // email
37 }
38 /*
39 * ASSETS
40 */
41 asset Artefact identified by ArtefactID{     // added to investigation list
42   o String ArtefactID //regex = /A[0-9]{1,5}/ // relates to unique id for evidence
43   o String description                        // description of artefact, usually required
44   -->Staff custodian                          // intially -->STAFF; custodian could be any
       participant
45 }
46
47
48 asset Investigation identified by InvestigationID{  // Case is reserved word and can be
       confusing
```

CTO III

Listing

```
49    o String InvestigationID regex = /C[0-9]{3}/      // Case ID == InvestigationID
50    --> Manager MoI                                   // Manager of Investigation, people
51    o Person[] FRT                                    // list of staff eligible to seize
          evidence
52    o Artefact[] ArtefactList                         // intially empty, populated by
          seizure
53  }
54  /*
55  * TRANSACTIONS
56  */
57  transaction seizure{                    // initial seizure, adding an artefact
58    -->Investigation currentInvestigation //investigation to update
59    //-->Staff submitter                  // authorised staff
60    o Artefact evidence                 // new item of evidence, concat to investigation's
          artefactList
61  }
62  transaction exchange{                   // exchange custody of evidence, only current owner
          's can swap
63    -->Staff receiver                     // receiver can be custodian - needs update
64    -->Artefact evidence                  // existing item of evidence
65  }
66  transaction addInvestigation{
67    //o Investigation ni    // add new investigation
68    o String ID   ///regex=/C[0-9]{3}/
69  }
70  transaction newMemberFRT{       // add existing member of staff to FRT
71    --> Investigation currentInvestigation  // current investigation, exists
72    --> Staff newMember             // member of staff to add, repeated 'n' times
```

CTO IV

Listing

```
73    }
74
75  /*transaction newMemberREAD{          // add member of staff who can have read access to
        investigation
76    --> Investigation currentInvestigation // current investigation
77    --> Staff newMember                 // member of staff to add to read list, can be in FRT
78  }
79  transaction newMemberALL{       //add member of staff who can have ALL access to
        investigation
80    --> Investigation currentInvestigation // current investigation
81    --> Staff newMember               // staff to add to ALL access list
82  }
83  transaction removeMemberREAD{    //remove a member of staff from the read list
84    --> Investigation currentInvestigation // current investigation
85    --> Staff exMember               // staff to remove from READ list
86  }
87  transaction removeMemberALL{     //remove a member of staff from ALL access list
88    --> Investigation currentInvestigation // current investigation
89    --> Staff exMember
90  }*/
```

ACL I

Listing

```
/*
* access control list for evidence network, boc
*/

rule staffSeeSelf{
    description:    "staff can only see themselves"
    participant(p): "org.boc.net.Staff"
    operation:      READ
    resource(r):    "org.boc.net.Staff"
    condition:      (p.getIdentifier()===r.getIdentifier())
    action:         ALLOW
}
rule managerSeeSelf{
    description:    "manager see self"
    participant(p): "org.boc.net.Manager"
    operation:      READ
    resource(r):    "org.boc.net.Manager"
    condition:      (p.getIdentifier()===r.getIdentifier())
    action:         ALLOW
}
rule managerSeeStaff{
    description:    "manager see staff"
    participant:    "org.boc.net.Manager"
    operation:      ALL
    resource:       "org.boc.net.Staff"
    action:         ALLOW
}
/*rule managerInvestigation{
```

ACL II

Listing

```
29      description:    "manager see case"
30      participant(p): "org.boc.net.Manager"
31      operation:      ALL
32      resource(r):    "org.boc.net.Investigation"
33      condition:      (p.getIdentifier()===r.MoI.getIdentifier())
34      action:         ALLOW
35  }*/
36  rule managerInvestigation{
37    description: "manager sees cases"
38    participant: "org.boc.net.Manager"
39    operation: ALL
40    resource:   "org.boc.net.Investigation"
41    action: ALLOW
42  }
43  rule managerTXnewMemberFRT{
44      description: "manager is allowed updates for all transactions"
45      participant: "org.boc.net.Manager"
46      operation:   ALL
47      resource:  "org.boc.net.newMemberFRT"
48      action: ALLOW
49  }
50  rule managerTXaddInvestigation{
51    description: "add new investigation"
52    participant: "org.boc.net.Manager"
53    operation: ALL
54    resource: "org.boc.net.addInvestigation"
55    action: ALLOW
56  }
```

ACL III

Listing

```
57 rule managerSeeArtefacts{
58    description:      "manager see case artefacts"
59    participant:  "org.boc.net.Manager"
60    operation:        READ
61    resource:     "org.boc.net.Artefact"
62    action:           ALLOW
63 }
64 rule staffSeeArtefacts{
65   description:  "staff see artefacts"
66   participant:  "org.boc.net.Staff"
67   operation:  READ,CREATE,UPDATE
68   resource:   "org.boc.net.Artefact"
69   action:   ALLOW
70 }
71 rule staffUpdateArtefactList{
72   //only those allowed to seize can do this
73   description: "upon seizure staff need to update both artefacts and artefact list in
        investigation"
74   participant: "org.boc.net.Staff"
75   operation: READ, CREATE, UPDATE
76   resource: "org.boc.net.Investigation"
77   action: ALLOW
78 }
79
80 rule staffSeeHistory{
81   description: "Staff can only see their own history"
82   participant(t): "org.boc.net.Staff"
83   operation: READ
```

ACL IV

Listing

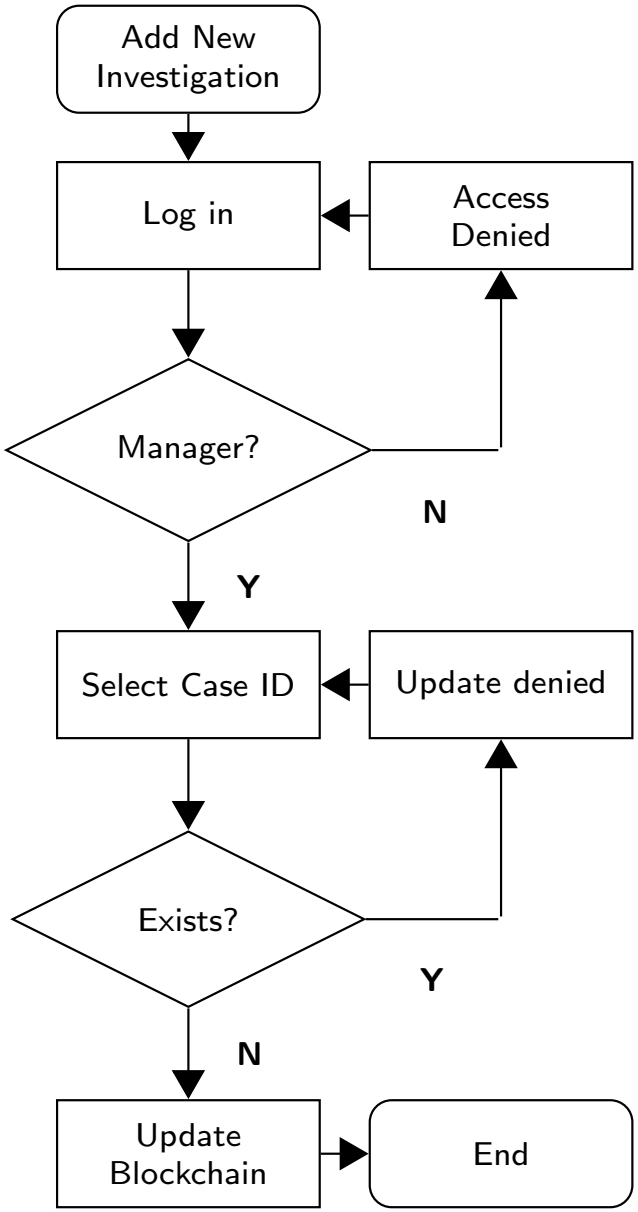
```
84    resource(v): "org.hyperledger.composer.system.HistorianRecord"
85    condition: (v.participantInvoking.getIdentifier() != t.getIdentifier())
86    action: DENY
87  }
88  rule staffTXseizure{
89    description: "Staff can seize new artefacts"
90    participant: "org.boc.net.Staff"
91    operation:  ALL
92    resource: "org.boc.net.seizure"
93    action: ALLOW
94  }
95
96  rule staffTXexchange{
97    description:  "staff can exchange existing artefacts"
98    participant:  "org.boc.net.Staff"
99    operation:  ALL
00    resource:   "org.boc.net.exchange"
01    action: ALLOW
02  }
03
04  rule SystemACL {
05      description:  "System ACL to permit all access"
06      participant: "org.hyperledger.composer.system.Participant"
07      operation: ALL
08      resource: "org.hyperledger.composer.system.**"
09      action: ALLOW
10  }
11
```

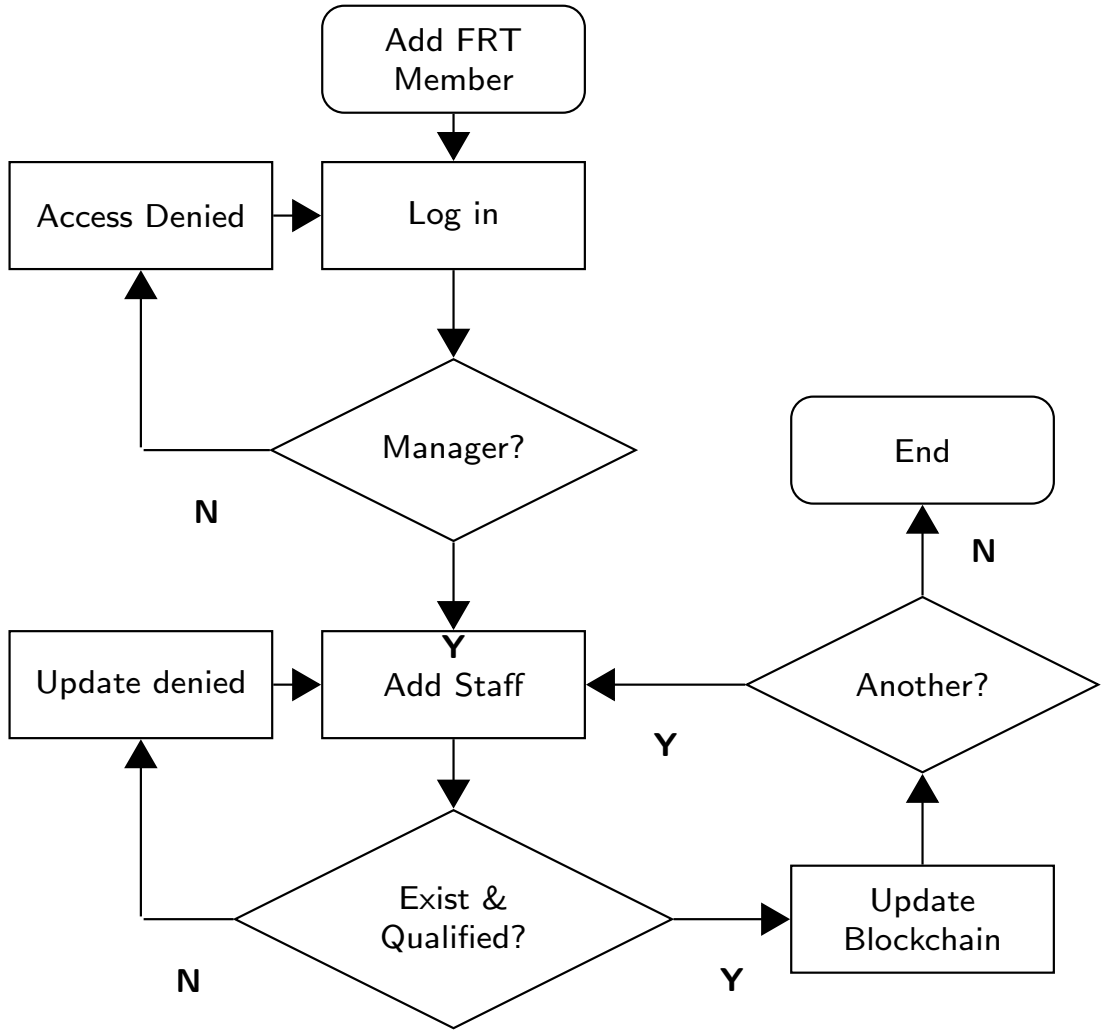
ACL V

Listing

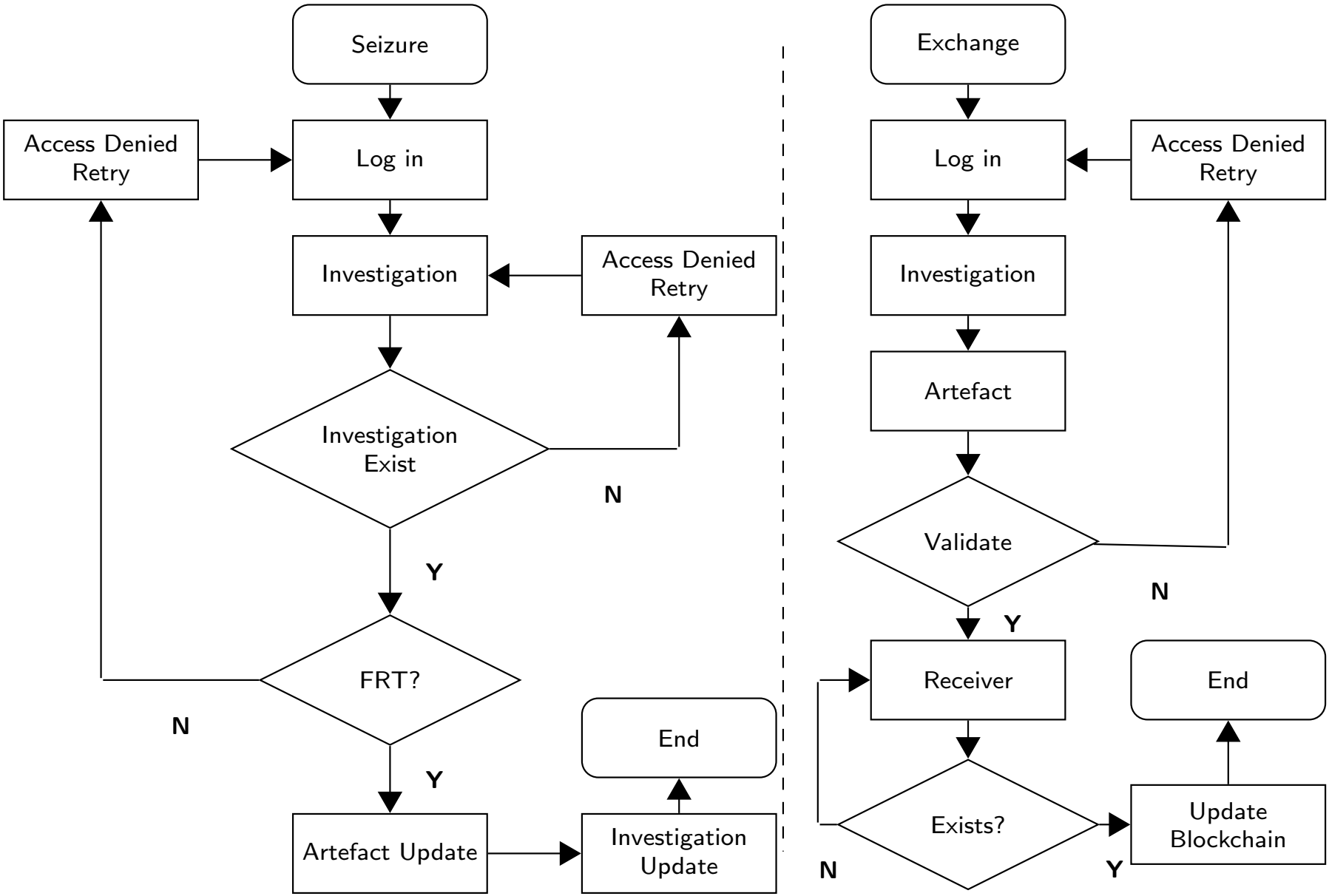
```
12 rule NetworkAdminUser {
13     description: "Grant business network administrators full access to user resources"
14     participant: "org.hyperledger.composer.system.NetworkAdmin"
15     operation: ALL
16     resource: "**"
17     action: ALLOW
18 }
19
20 rule NetworkAdminSystem {
21     description: "Grant business network administrators full access to system resources"
22     participant: "org.hyperledger.composer.system.NetworkAdmin"
23     operation: ALL
24     resource: "org.hyperledger.composer.system.**"
25     action: ALLOW
26 }
```

Adding a new investigation





Seizure and Exchange



Logic I

Listing

```
1  /*
2  * script file for evidence network, boc
3  * each transactions is described below and provides the business logic
4
5
6  * TRANSACTIONS */
7
8  /*
9  * add a member of staff to FRT
10 * @param{org.boc.net.newMemberFRT} newMemberFRT - add member of staff to FRT list
11 * @transaction
12 */
13 async function newMemberFRT(tx) {
14     //check info.newMember exists as member of staff
15     // shortcomming - could be a specialist or other, for now just check staff
16     var ns='org.boc.net.';
17     var me=getCurrentParticipant();
18     return getParticipantRegistry(ns+'Staff')
19         .then(function (staffReg){
20             return staffReg.exists(tx.newMember.getIdentifier());
21         })
22         .then(function(exists){
23             //if member of staff exists then update & current user is manager of
       investigation (MoI)
24             // if((exists)&&(me.getIdentifier()===tx.currentInvestigation.MoI.
       getIdentifier()))
25             // current manager is check by ACL, no need to repeat
26             // if member of staff exists
```

Logic II

Listing

```
27              if (exists)
28              {
29                  tx.currentInvestigation.FRT = tx.currentInvestigation.FRT.concat(tx.
      newMember);
30                  return getAssetRegistry(ns+'Investigation')
31                      .then(function(investigationReg){
32                          return investigationReg.update(tx.currentInvestigation)
33                      })
34              } else {
35                  throw new Error('Staff member'+tx.newMember.getIdentifier()+' does not
      exist! ')
36              }
37          })
38     }
39
40     /*
41     * add a new investigation
42     * @param {org.boc.net.addInvestigation} addInvestigation - add a new investigation,
       with MoI
43     * @transaction
44     */
45    async function addInvestigation(tx){
46        //acl means only managers can view their own cases
47        // does not check if case number already exists
48        var ns='org.boc.net.';
49        var me=getCurrentParticipant();
50        return getAssetRegistry(ns+'Investigation')
51          .then(function (investigationReg){
```

Logic III

Listing

```
52                  var factory = getFactory();
53                  var ni = factory.newResource('org.boc.net','Investigation',tx.ID);
54                  ni.MoI=me;
55              //ni.Authority='';
56              ni.ArtefactList=[];
57              ni.FRT=[];
58                  return investigationReg.add(ni);
59          })
60    }
61 /*
62 * add a new artefact
63 * @param {org.boc.net.seizure} seizure - add a new artefact
64 * @transaction
65 */
66 async function seizure(tx){
67     var ns='org.boc.net.';
68     var me=getCurrentParticipant();
69    return getAssetRegistry(ns+'Investigation')
70         .then(function(iReg){
71            //check investigation exists
72             return iReg.get(tx.currentInvestigation.getIdentifier())
73                 .then(function(singleInvestigation){
74                 //check current user is member of FRT exists
75                     var needle=me.getIdentifier();
76                     var haystack=singleInvestigation.FRT;
77                     var filteredHaystack = haystack.filter((item)=>item.PIN===needle);
78                     if (filteredHaystack.length>0){
```

Logic IV

Listing

```
79                          //cannot submit on behalf of others, default option of ownership
      is me
80                          tx.evidence.custodian = me;
81                          singleInvestigation.ArtefactList = singleInvestigation.
      ArtefactList.concat(tx.evidence);
82                          iReg.update(singleInvestigation);
83                          //update artefact
84                          return getAssetRegistry(ns+'Artefact')
85                            .then(function(aReg){
86                              return aReg.add(tx.evidence);
87                              });
88                      } else {
89                          throw new Error('User is not a member of FRT for this
      investigation')
90                      }
91                  })
92
93              })
94  }
95
96  /*
97  * exchange an existing artefact
98  * @param {org.boc.net.exchange} exchange - exchanging the custodian of an existing
        artefact
99  * @transaction
00  */
01  async function exchange(tx){
02      const me=getCurrentParticipant();
```

Logic V

Listing

```
03      //staff cannot update on behalf of others
04      if(me.getIdentifier()!==tx.evidence.custodian.getIdentifier()){
05          throw new Error('Staff cannot update on behalf of other staff');
06      } else {
07              // artefact is in artefactList, since it has been submitted
08              tx.evidence.custodian = tx.receiver
09              //update
10              let assetRegistry = await getAssetRegistry('org.boc.net.Artefact');
11              await assetRegistry.update(tx.evidence);
12      }
13
14  }
```

Output


```
{
 "$class": "org.boc.net.addInvestigation",
 "ID": "C111",
 "transactionId": "ae86b543-54a4-444c-8aeb-28268cf02ff5",
 "timestamp": "2018-08-28T15:57:21.212Z"
}
{
 "$class": "org.boc.net.addInvestigation",
 "ID": "C222",
 "transactionId": "c8b05472-9299-4579-9c34-2788b4a731d7",
 "timestamp": "2018-09-04T09:53:00.558Z"
}
```

Summary

Conclusions

Chain of Custody

References I

[1] I. Mitchell et al. "Blockchain of Custody, BoC". In: *Cyber Security Practitioner's Guide* (2019). Ed. by Hamid Jahankhani.

[2] Scientific Working Group on Digital Evidence (SWDGE). *Model Standard Operation Procedures for Computer Forensics (ver. 3)*. https://www.swgde.org/. Version 3. [Accessed June 2016].

Web Resources

- `http://hyperledger.org`
- `https://nodejs.org`