

Land Registration System using Permissioned Blockchain



Group Information

Course Code: CSE-4800

Supervised By:

**Abdullahil Kafi Sir
Assistant Professor
Department of CSE**

Presented By:

**Md. Rokib Khan [C181053]
Ali Khatami [C191068]
Md. Rafidul Islam [C191065]**

Contents

- Introduction
- Problem Statement
- Literature Review
- Project Objective
- Methodology
- Results & Discussion
- Future Work
- Conclusion
- Reference

Introduction

- We will build a Land Registration solution for Bangladesh
- We will use Permissioned Blockchain Hyperledger Fabric
- Registration info & Land details will be stored in Hyperledger Fabric
- Client application will be built with Express.js framework of Node.js
- Login information will be stored in MongoDB

Problem Statement

- Some bad people work with helpers and dishonest officials to make fake papers and take control of land by force. This leaves the real owners in long fights in court to get their land back.
- These land sharks use bribes to change papers, selling the land to people they don't know, causing big legal problems between the true owner and the new buyer with fake documents.
- The original landowners might make many agreements, taking money from different buyers using a special paper called a pledge deed (Binah Dalil), leading to common court problems.
- Dishonest landowner may sell the same land multiple times without completing registration and mutation processes. Post-registration, some buyers may fail to fulfil the agreed-upon payments.

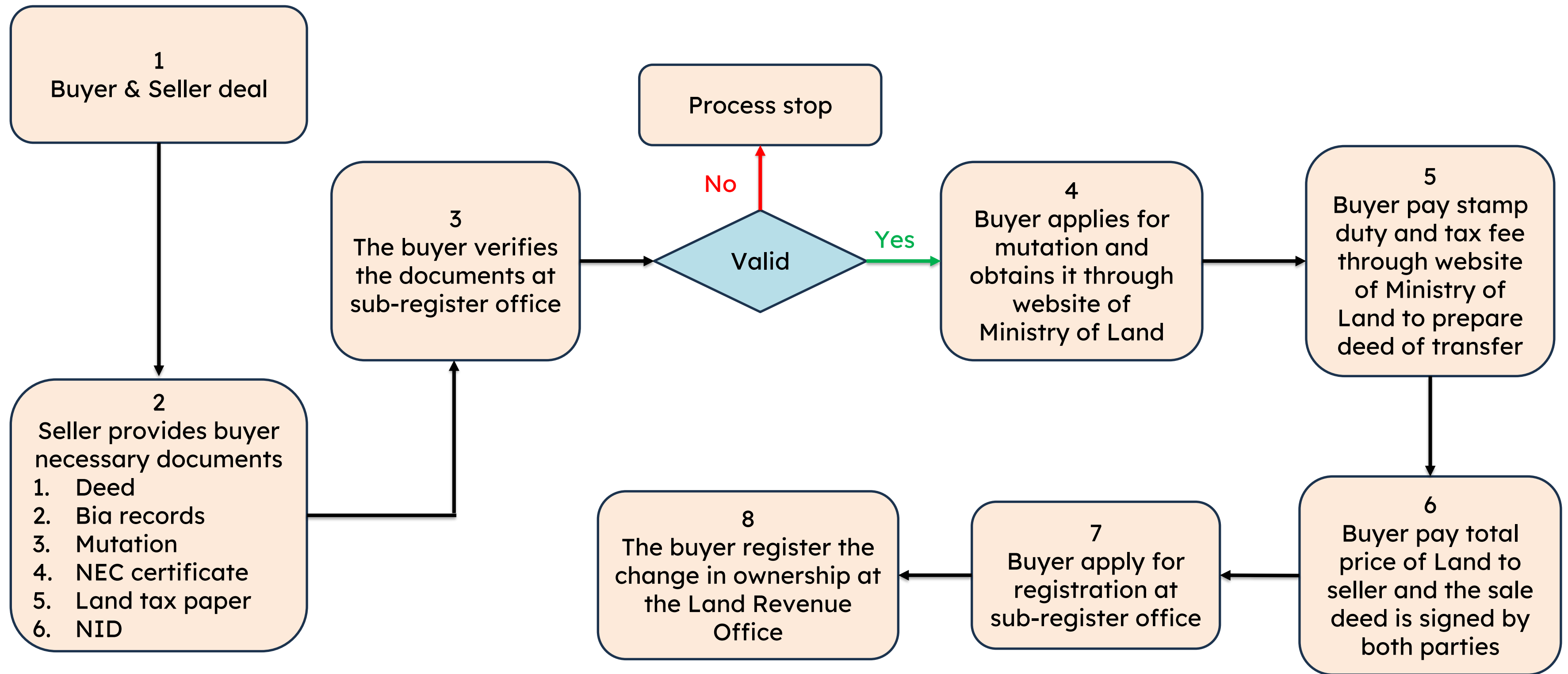
Problem Statement

- Fixing these issues usually means making friendly agreements or going to the police. But dealing with the police in the legal system brings more money and time problems, lasting for months or even years.

Literature Review & Related Work

Author	Methodology	Result	Limitation
Thakur, V., Doja, M. N., Dwivedi, Y. K., Ahmad, T., Khadanga, G.A (2020)	The proposal suggests leveraging blockchain, particularly smart[1] contracts, to document property transactions encompassing sales, inheritance, court orders, and land acquisition.	Successfully addressed the issue of implementing both public and, private blockchain on land registration systems in India and successfully solved it.	The load on the Blockchain system will gradually increase and the transactions will increase because of population growth
Fernando, D., Ranasinghe, N. (2019)	Implemented on Hyperledger Fabric v1.2, we assessed performance on AWS t2.large with 2 vCPUs and 8 GB of memory, considering transaction density and node failures.	AM2 unifies channels, employing a single chaincode and island-wide land ledger, outperforming AM1's separate channels.	Using Kafka for ordering service doesn't decentralize nodes when operated by different organizations, as all connect to a single-controlled Kafka cluster.
Lazuashvili, N., Norta, A., Draheim, D. (2019)	They have used Bitcoin blockchain technology.	Project achieved heightened safety, security for citizen data, and improved transparency, ensuring traceability of information.	Citizen data can be altered before being stored on immutable blockchain storage.
Alam, K. M., Rahman, J. A., Tasnim, A., Akther, A. (2022)	Ethereum deployed smart land title contracts. using Solidity. Ganache-cli for local testing, Remix as an online IDE, and Metamask for browser blockchain Interaction.	Efficiently manage land ownership, ensure traceability, minimize travel, time, costs.	The transaction fee varies with the ETH value, which will not be acceptable for people.
Zhang, L., Ci, L., Wu, Y., Wiwatan apataphee, B. (2023).	They built their system on the Ethereum blockchain network.	Cuts real estate authentication expenses, speeds up transactions, eliminates third party verification	Uncertain blockchain legality impacts practical use for businesses and organizations.

Literature Review Existing Land Registration System in Bangladesh



Project Objectives

- To create a system that stops bad people from making fake papers to take over land, make a secure and easy way to check who owns land, and keep clear records of land deals to stop dishonest officials from cheating.
- We aim to implement the Raft ordering service, allowing each organization to have its own ordering nodes. This promotes a more decentralized system.
- Our project design ensures citizens' data remains unmodifiable until the title is transferred to the immutable storage on the blockchain.
- We have implemented measures to prevent common individuals from being burdened by transaction fees.
- We've implemented measures to prevent a gradual increase in the load on the blockchain system and transactions due to population growth.
- We will try to solve uncertain blockchain legality issues in many countries, such as the use of cryptocurrency.

Methodology

On this section We will discuss the following:

- Tools used
- Algorithm
- Network Architecture
- Step by step process
- Chaincode
- Diagrams
- User Interface

Methodology: Tools

The tools we used are:

- | | |
|---------------------------|-------------------|
| • Backend Server | Node.js (Express) |
| • Fabric API | Node.js |
| • Chaincode | Go |
| • Database Authentication | MongoDB, Firebase |
| • Database Storage | LevelDB |
| • API Testing | Postman |
| • Testing with frontend | Ngrok |
| • Consensus Algorithm | Raft |
| • Certificate Format | X.509 |
| • Hashing Algorithm | SHA-256 |

Methodology: Consensus Algorithm



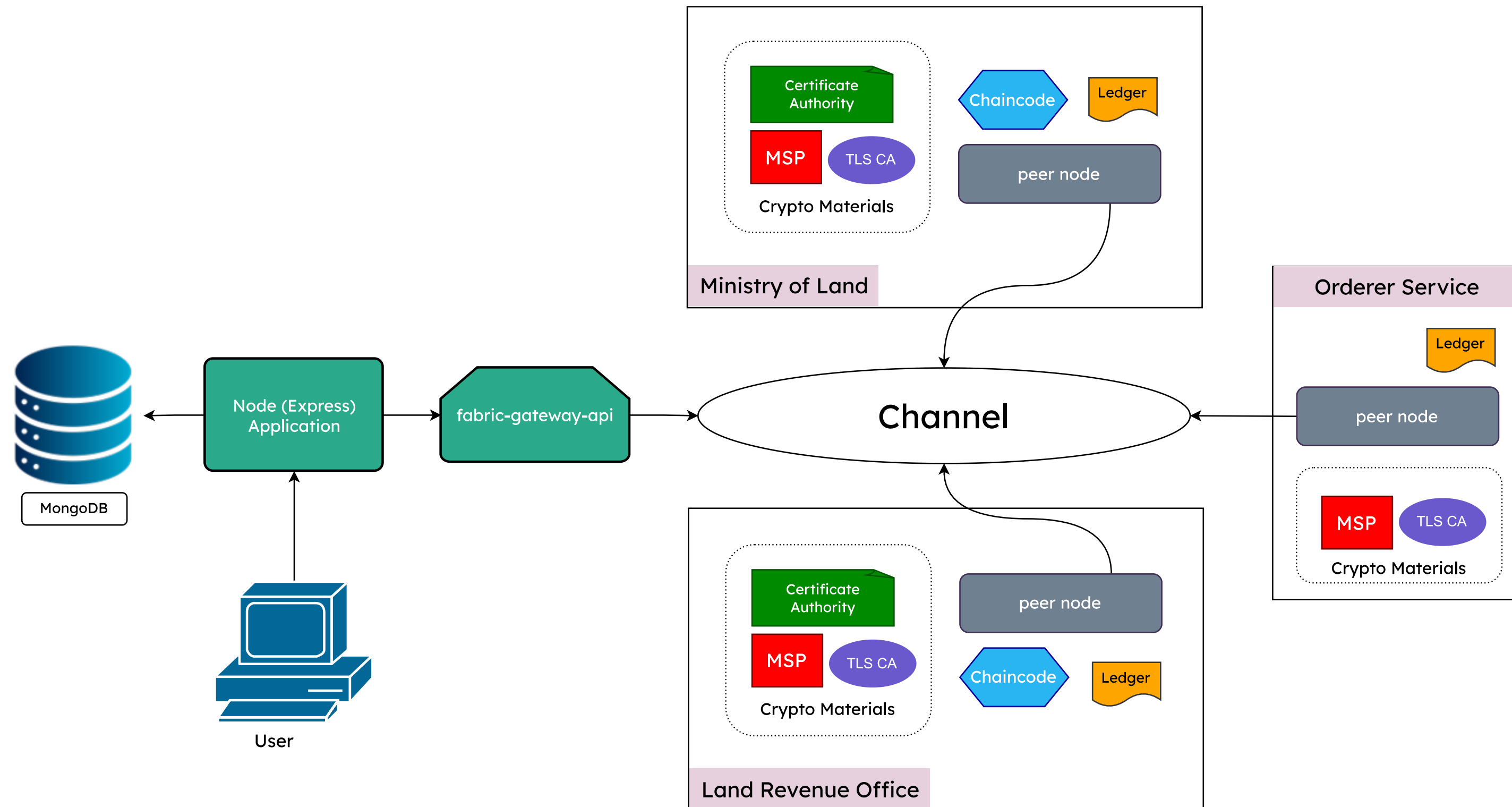
```
1  # SampleDevModeEtcdRaft defines a configuration that differs from the
2  # SampleDevModeSolo one only in that it uses the etcd/raft-based orderer.
3  SampleDevModeEtcdRaft:
4      <<: *ChannelDefaults
5      Orderer:
6          <<: *OrdererDefaults
7          OrdererType: etcdraft
8      Organizations:
9          - <<: *SampleOrg
10         Policies:
11             <<: *SampleOrgPolicies
12         Admins:
13             Type: Signature
14             Rule: "OR( 'SampleOrg.member' )"
```

Methodology: Hashing Algorithm



```
1 #####
2 # BCCSP (BlockChain Crypto Service Provider) section is used to select which
3 # crypto library implementation to use
4 #####
5 bccsp:
6     default: SW
7     SW:
8         hash: SHA2
9         security: 256
10    filekeystore:
11        # The directory used for the software file-based keystore
12        keystore: msp/keystore
13
14 #####
```

Methodology: Network Architecture



Methodology: Step by step process

- Login: Verified users will login with valid credentials into their account.
- Dashboard: After login users can see their most recent 10 applications and some statistics about Pending, Rejected, Approved and Registered application.
- Apply for Registration: After logging in users can apply for land registration from Register Application menu. They will provide valid information of the land with NEC. If no previous record is already been submitted for the same land the application will be stored to ledger for verification.
- Approve application Ministry of Land: Any admin from Ministry of Land will see all pending application in their dashboard. They will verify the applications and approve if information is correct and reject if not.

Methodology: Step by step process

- Register application Land Revenue Office: Admin from Land Revenue office will see all application approved by Ministry of Land in their dashboard. They will verify the applications and register the land and generate Deed if information is correct and reject if not.
- Transfer Ownership: Users can transfer the ownership of their registered land to other users. After successful transfer of ownership the land will again have to be verified from Ministry of Land and Land Revenue Office to register the land to new user.

Methodology: Chaincode (CreateApplication)



```
1 // Create Land Application
2 func (s *SmartContract) CreateApplication(ctx contractapi.TransactionContextInterface,
3     dagNo int, dist string, div string, khatianNo int, mouza string, nec string, oName
4     string, oNID int, payTx string, upazila string) (string, error) {
5
6     id := fmt.Sprintf("%d%s%d%s", dagNo, prefix1, prefix2, khatianNo, prefix3, prefix4)
7     exists, err := s.AssetExists(ctx, id)
8     if exists { return "", error }
9
10    land := Land{
11        // store land info
12    }
13    landJSON, err := json.Marshal(land)
14    err = ctx.GetStub().PutState(id, landJSON)
15    return ctx.GetStub().GetTxID(), nil
16 }
```

Methodology: Chaincode (AdminDashboard)

```
1 // Admin Dashboard
2 func (s *SmartContract) GetAllAdminApplications(ctx contractapi.TransactionContextInterface,
3     oNID int, userType string) ([]*Dashboard, error) {
4
5     if userType == userString { return nil, fmt.Errorf("User not authorized") }
6     resultsIterator, err := ctx.GetStub().GetStateByRange("", "")
7
8     clientMSPID, err := ctx.GetClientIdentity().GetMSPID()
9     if err != nil { return nil, error }
10
11     var lands []*Dashboard
12     for resultsIterator.HasNext() {
13         queryResponse, err := resultsIterator.Next()
14         var land Land
15         err = json.Unmarshal(queryResponse.Value, &land)
16
17         if (clientMSPID == moMSPID && land.Status == "pending") ||
18             (clientMSPID == lroMSPID && land.Status == "approved") {
19             dashboardLand := Dashboard{ // append into dashboard }
20             lands = append(lands, &dashboardLand)
21         }
22     }
23     return lands, nil
24 }
```

Methodology: Chaincode (ApproveApplication)



```
1 // Approve Land Application by Ministry of Land
2 func (s *SmartContract) ApproveApplication(ctx contractapi.TransactionContextInterface,
3     id string, comment string, adminNID int, response bool, userType string) error {
4
5     clientMSPID, err := ctx.GetClientIdentity().GetMSPID()
6     if userType == userString || clientMSPID != molMSPID { return error }
7
8     landJSON, err := ctx.GetStub().GetState(id)
9     var land Land
10    err = json.Unmarshal(landJSON, &land)
11    if land.Status != "pending" { return error }
12
13    land.Status = status
14    land.Comment = comment
15    appendUnique(&land, adminNID)
16
17    landJSON, err = json.Marshal(land)
18    return ctx.GetStub().PutState(id, landJSON)
19 }
```

Methodology: Chaincode (RegisterApplication)



```
1 // Register Land Application by Land Revenue Office
2 func (s *SmartContract) RegisterApplication(ctx contractapi.TransactionContextInterface,
3     id string, comment string, adminNID int, response bool, userType string) error {
4
5     clientMSPID, err := ctx.GetClientIdentity().GetMSPID()
6     if userType == userString || clientMSPID != lroMSPID { return error }
7
8     landJSON, err := ctx.GetStub().GetState(id)
9     if landJSON == nil { return error }
10
11     var land Land
12     err = json.Unmarshal(landJSON, &land)
13     if land.Status != "approved" { return error }
14
15     land.Status = status; land.Comment = comment; appendUnique(&land, adminNID)
16     if land.DeedID == "" { land.DeedID, err = s.GenerateDeed(ctx, land) }
17
18     landJSON, err = json.Marshal(land)
19     return ctx.GetStub().PutState(id, landJSON)
20 }
```

Methodology: Chaincode (GenerateDeed)

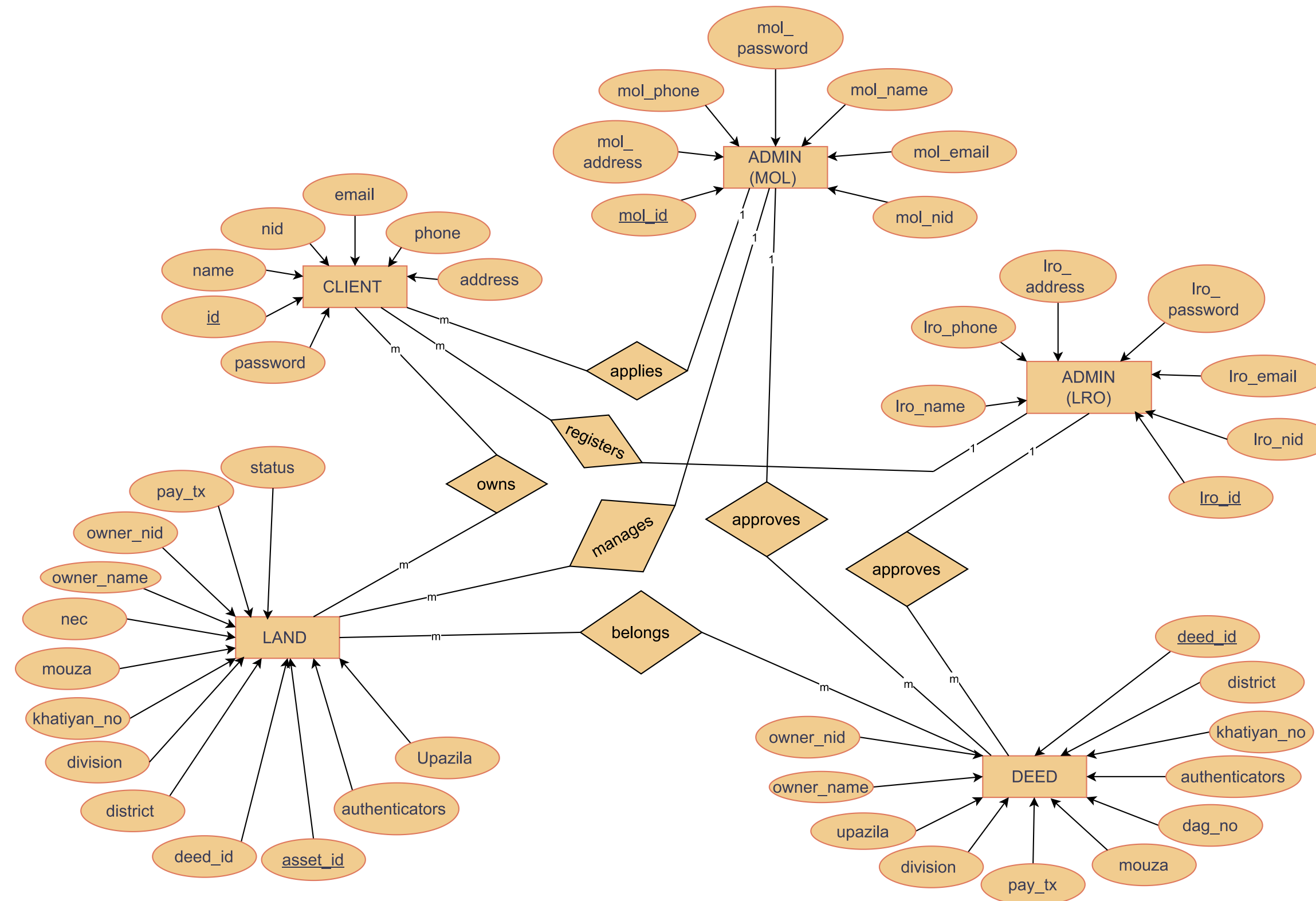


```
1 // Generate Deed of Land
2 func (s *SmartContract) GenerateDeed(ctx contractapi.TransactionContextInterface,
3     land Land) (string, error) {
4
5     prfx := time.Now().Unix()
6     id := fmt.Sprintf("%d%s", prfx, land.AssetId)
7
8     exists, err := s.AssetExists(ctx, id)
9     if exists { return "", error }
10
11     deed := Deed{
12         // store deed info
13     }
14
15     deedJSON, err := json.Marshal(deed)
16     err = ctx.GetStub().PutState(id, deedJSON)
17     return id, nil
18 }
```

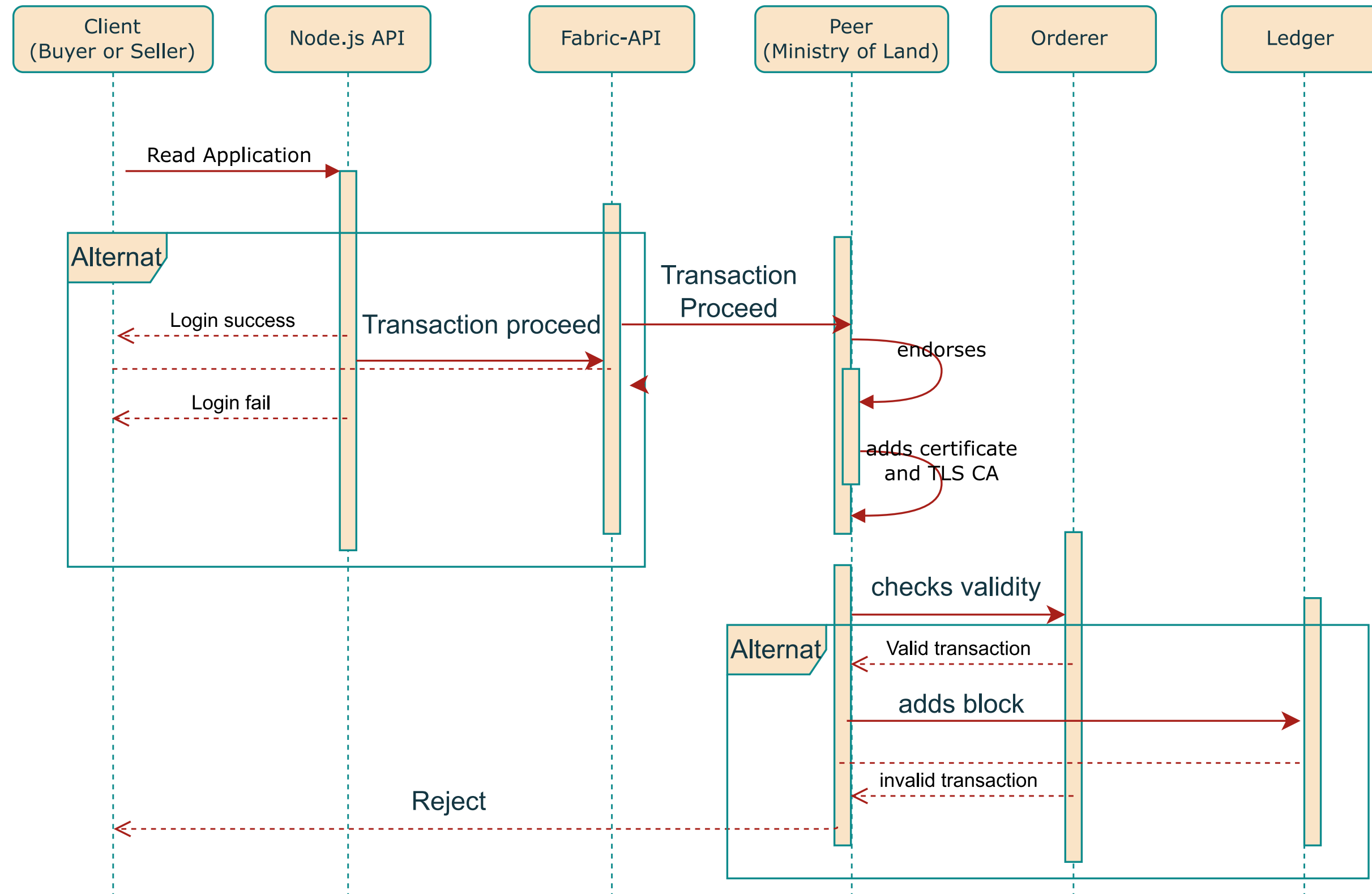

Methodology: Chaincode (TransferOwnership)

```
1  // Transfer ownership
2  func (s *SmartContract) TransferOwnership(ctx contractapi.TransactionContextInterface,
3      id string, oNID int, newOwnerName string, newONID int) error {
4
5      landJSON, err := ctx.GetStub().GetState(id)
6      if landJSON == nil { return error }
7      var land Land
8      err = json.Unmarshal(landJSON, &land)
9
10     if land.OwnerNID != oNID { return error }
11     err = s.DeleteAsset(ctx, land.DeedID)
12
13     land.OwnerName = newOwnerName
14     land.OwnerNID = newONID
15     land.Status = "pending"
16     landJSON, err = json.Marshal(land)
17     return ctx.GetStub().PutState(id, landJSON)
18 }
```

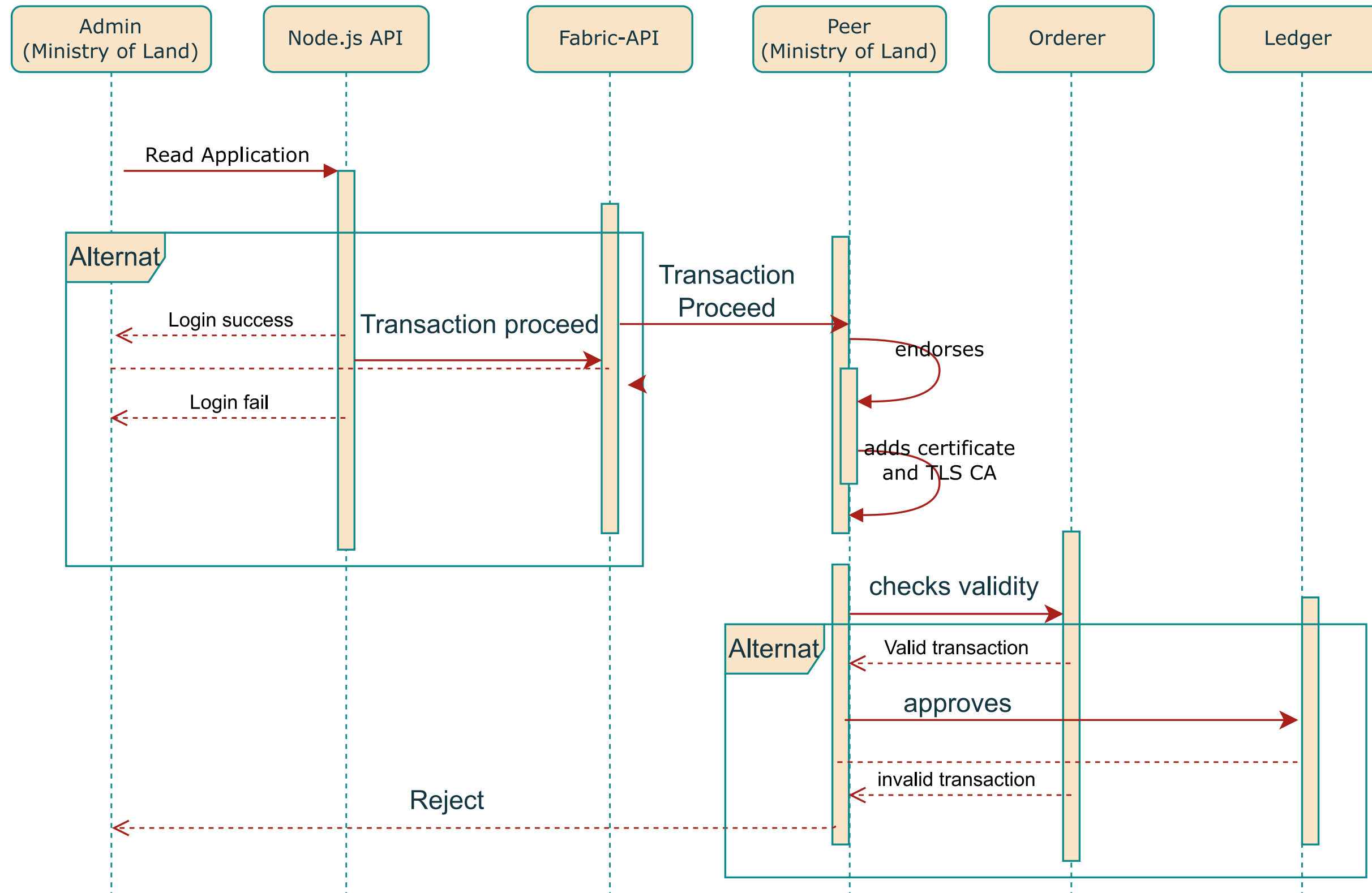
Methodology: Diagram (Entity Relationship Diagram)



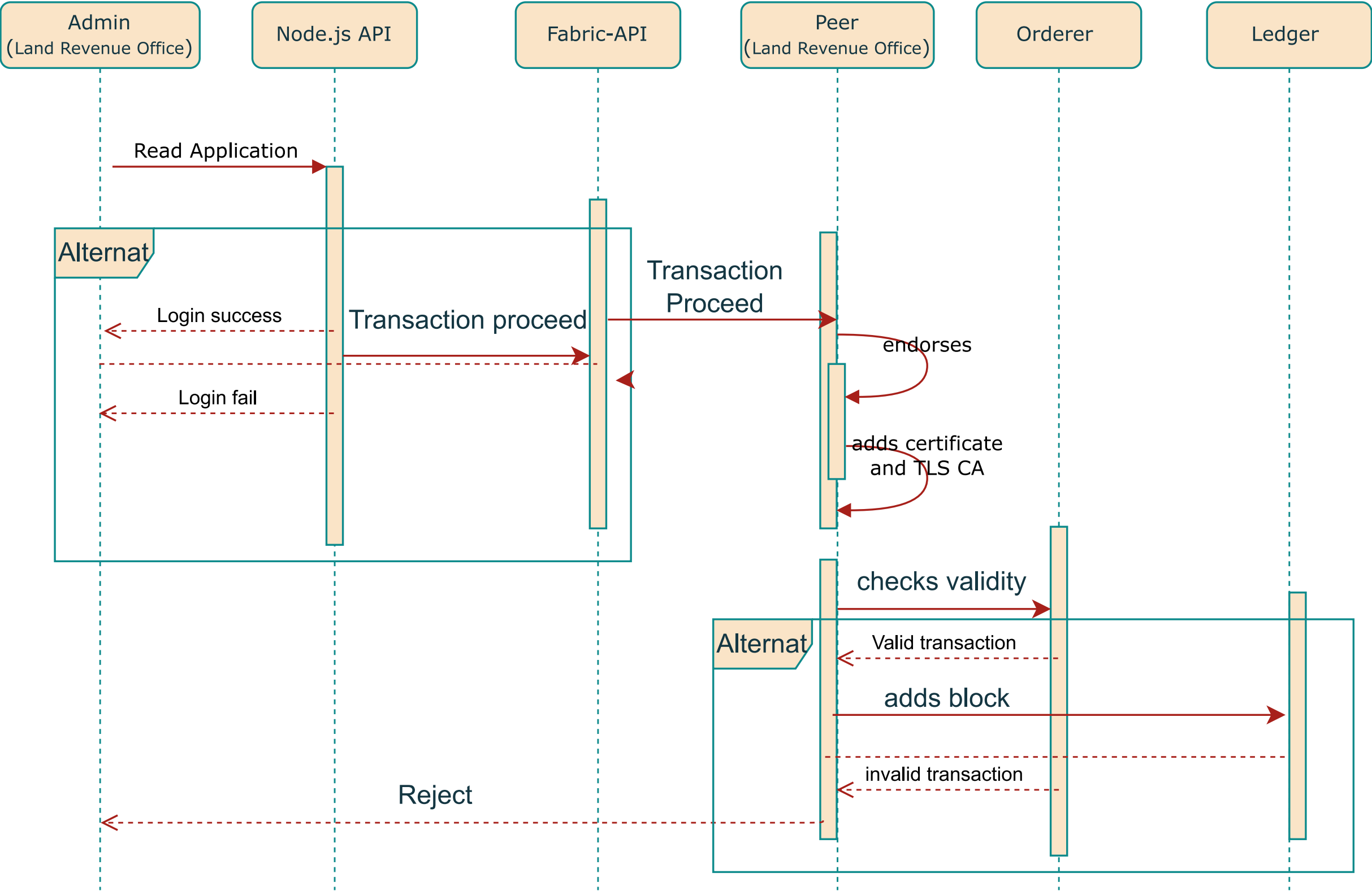
Methodology: Diagram (Sequence Diagram Client)



Methodology: Diagram (Sequence Diagram Ministry of Land)



Methodology: Diagram (Sequence Diagram Land Revenue Office)



Methodology: UI (Land Registration Form)

PropertyBlock

user1Logout

Dashboard

Register

All Application

Registered Land

Profile

Land Registration form

Provide Land Information

Dag/Plot Number

4543

District

Chittagong

Division

Chittagong

Khatian Number

116

Mouza

Kattali

Payment Transaction

00x8f6a7b2f9e1c4d3fe8c6f0a1b9d7b3

Upazila


Dabalmoring

NEC

CHOOSE FILE

NEC.pdf

Submit



27

Department Of CSE, IIUC

Methodology: UI (Application Details)

PropertyBlock

user1LogOut

Dashboard

Register

All Application

Registered Land

Profile

Application Info

Application ID	4543chittagongchittagong117kattalidabalmuring	
Owner Name	user1	
Owner ID	9876543211	
Payment Transaction ID	00x8f6a7b2f9e1c4d3fe8c6f0a1b9d7c3	
Status	registered	
NEC	9876543211-1705915576303.pdf	Download NEC

Land Info

Division	Chittagong
District	Chittagong
Upazila	Dabalmuring
Mouza	Kattali

Methodology: UI (Application Approve)

- Dashboard
- + Register
- All Application
- Registered Land
- Profile

Land Info

Division	Chittagong
District	Chittagong
Upazila	Dabalmuring
Mouza	Kattali
Khatian Number	116
Dag/Plot Number	4543

Deed Info

Deed ID

Authenticators

[Ministry of Land](#)

[Land Revenue Office](#)

Official Statement

Download Deed

Add Comment

Reject

Approve

Methodology: UI (Application Register)

Dashboard

+ Register

All Application

Registered Land

Profile

Land Info

Division	Chittagong
District	Chittagong
Upazila	Dabalmuring
Mouza	Kattali
Khatian Number	116
Dag/Plot Number	4543

Deed Info

Deed ID	
Authenticators	Ministry of Land Land Revenue Office
	9876543219
Official Statement	Land Application Approved by Ministry of Land

Download Deed

Add Comment

Reject

Register

Methodology: UI (Certificate of Registered Land)

Land Revenue Office Bangladesh

This is to certify that the rightful owner of the land is

ID: 4543chittagongchittagong117kattalidabalmuring

Name: user1

NID: 9876543211

Land Information

Division	: Chittagong	Dag/Plot Number	: 4543
District	: Chittagong	Mouza	: Kattali
Upazila	: Dabalmuring	Khatian Number	: 117

Land Deed

Deed ID : 17060153804543chittagongchittagong117kattalidabalmuring

9876543219
Ministry of Land

9876543218
Land Revenue

Methodology: UI (Transfer Ownership Form)

PropertyBlock

user1Logout

Dashboard

Register

All Application

Registered Land

Profile

Transfer Ownership

Give buyers information

Name

user2

NID

9876543212

Transfer OwnerShip

Result: API Test (Apply for land registration)

POST {{URL}}/user/register Send

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	Key		Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	dagNo	Text	4541			
<input checked="" type="checkbox"/>	dist	Text	Chittagong			
<input checked="" type="checkbox"/>	div	Text	Chittagong			
<input checked="" type="checkbox"/>	khatianNo	Text	123			
<input checked="" type="checkbox"/>	mouza	Text	Kattali			
<input checked="" type="checkbox"/>	nec	File	nec.pdf			
<input checked="" type="checkbox"/>	payTx	Text	00x8f6a7b2f9e1c4d3fe8c6f0a1b9d7c3			
<input checked="" type="checkbox"/>	upazila	Text	Dabalmuring			

Body Cookies Headers (8) Test Results

Status: 201 Created Time: 2.53 s Size: 398 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "msg": "Asset created Successfully",
4   "txID": "c9b7e5d47a27955de74a56e3acb11a4ef6c5623532d8c614491c0fa4df7efec9"
5 }
```

Result: API Test (Admin verification)

POST

{{URL}}/user/application/verify/4541chittagongchittagong123kattalidabalmuring

Send

Params

Authorization

Headers (12)

Body

Pre-request Script

Tests

Settings

Cookies

Headers

9 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input type="checkbox"/>	Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NTd...				
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NTd...				

Body

Cookies

Headers (8)

Test Results

Status: 201 Created

Time: 2.66 s

Size: 328 B

Save as example

...

Pretty

Raw

Preview

Visualize

JSON

...

...

...

1 {

2 "success": true,

3 "msg": "Land application status updated"

4 }



Result: API Test (Deed of Registered Land)

GET {{URL}}/public/application/certificate/17052220204541chittagongchittagong123kattalidabalmuring Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NTdjNzU0MjR...				

Body Cookies Headers (8) Test Results Status: 200 OK Time: 19 ms Size: 621 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "data": {
4     "AssetID": "4541chittagongchittagong123kattalidabalmuring",
5     "Authenticators": [
6       9876543219,
7       9876543218
8     ],
9     "DagNo": 4541,
10    "DeedID": "17052220204541chittagongchittagong123kattalidabalmuring",
11    "District": "Chittagong",
12    "Division": "Chittagong",
13    "KhatianNo": 123,
14    "Mouza": "Kattali",
15    "OwnerName": "user1",
16    "OwnerNID": 9876543211,
17    "Upazila": "Dabalmuring"
18  }
19 }
```

Result: API Test (Ownership Transfer)

The screenshot displays a REST client interface with a POST request to the endpoint `{{URL}}/user/application/transfer/4541chittagongchittagong123kattalidabalmuring`. The request body is a JSON object with the following structure:

```
1 {  
2   "newOName": "user2",  
3   "newONID": 9876543212  
4 }
```

The response status is 201 Created, with a time of 2.37 s and a size of 332 B. The response body is a JSON object indicating success:

```
1 {  
2   "success": true,  
3   "msg": "Land Ownership successfully updated"  
4 }
```

The interface includes tabs for Params, Authorization, Headers (12), Body, Pre-request Script, Tests, and Settings. The Body tab is active, showing the request and response JSON. The response is formatted as JSON, and the status bar indicates the request was successful.

Result: Tamperproof Test

```
Terminal
File Edit View Search Terminal Help

[mohammadrokib ~/Codes/PropertyBlock/test-network] main ≡ ?3 ~1
$ export PATH=${PWD}/../bin:$PATH
export FABRIC_CFG_PATH=$PWD/../config/

[mohammadrokib ~/Codes/PropertyBlock/test-network] main ≡ ?3 ~1
$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganization
s/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCe
rtFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCe
rtFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"ResubmitApplication","Args
":["4543chittagongchittagong116kattalidabalmuring", "4543", "Chittagong", "Chittagong", "116", "Kattali", "9876543211-1705979121346.pdf", "user2", "9
876543212", "00x8f6a7b2f9e1c4d3fe8c6f0a1b9d7c3", "Dabalmuring"]}'
Error: endorsement failure during invoke. response: status:500 message:"Not authorized"

[mohammadrokib ~/Codes/PropertyBlock/test-network] main ≡ ?3 ~1 error
$
```

Conclusion

In conclusion, our project aims to address land registration challenges in Bangladesh by leveraging Hyperledger Fabric for a secure and permissioned blockchain solution.

The utilization of Express.js and Node.js for the client application, coupled with MongoDB for user authentication, reflects our commitment to a robust and scalable system. The tamperproof nature of Hyperledger Fabric ensures the integrity of land registration data, providing an extra layer of security.

As we move forward, future iterations will focus on enhancing user authentication through certificate authority, enhancing security. Which will pave the way for an efficient and trustworthy land registration solution tailored for the specific needs of Bangladesh.

References

[1] Hyperledger Fabric, “The Ordering Service.”

https://Hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html

Accessed: 06/12/2023.

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., “Hyperledger fabric: a distributed operating system for permissioned blockchains,” in Proceedings of the thirteenth EuroSys conference, pp. 1–15, 2018.

[3] M. Xu, X. Chen, and G. Kou, “A systematic review of blockchain,” Financial Innovation, vol. 5, no. 1, pp. 1–14, 2019.

[4] A. Mardan and A. Mardan, “Using express.js to create node.js web apps,” Practical Node.js: Building Real-World Scalable Web Apps, pp. 51–87, 2018.

References

- [5] M. F. Karagoez and C. Turgut, “Design and implementation of restful wireless sensor network gateways using node.js framework,” in European Wireless 2014; 20th European Wireless Conference, pp. 1–6, VDE, 2014.
- [6] V. Thakur, M. Doja, Y. K. Dwivedi, T. Ahmad, and G. Khadanga, “Land records on blockchain for implementation of land titling in india,” International Journal of Information Management, vol. 52, p. 101940, 2020.
- [7] D. Fernando and N. Ranasinghe, “Permissioned distributed ledgers for land transactions; a case study,” in Business Process Management: Blockchain and Central and Eastern Europe Forum: BPM 2019 Blockchain and CEE Forum, Vienna, Austria, September 1–6, 2019, Proceedings 17, pp. 136–150, Springer, 2019.

References

- [8] N. Lazuashvili, A. Norta, and D. Draheim, “Integration of blockchain technology into a land registration system for immutable traceability: a case study of georgia,” in Business Process Management: Blockchain and Central and Eastern Europe Forum: BPM 2019 Blockchain and CEE Forum, Vienna, Austria, September 1–6, 2019, Proceedings 17, pp. 219–233, Springer, 2019.
- [9] K. M. Alam, J. A. Rahman, A. Tasnim, and A. Akther, “A blockchain-based land title management system for bangladesh,” Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 6, pp. 3096–3110, 2022.
- [10] L. Zhang, L. Ci, Y. Wu, and B. Wiwatanapataphee, “The real estate time-stamping and registration system based on ethereum blockchain,” Blockchain: Research and Applications, p. 100175, 2023.

References

[11] Hyperledger Fabric, “Planning for an ordering service.”

<https://hyperledger-fabric.readthedocs.io/en/latest/deployorderer/ordererplan.html>

Accessed: 06/12/2023.

[12] X. Piao, M. Li, F. Meng, and H. Song, “Latency analysis for raft consensus on hyperledger fabric,” in International Conference on Blockchain and Trustworthy Systems, pp. 165–176, Springer, 2022.

[13] Hyperledger Fabric, “Channels.”

<https://hyperledger-fabric.readthedocs.io/en/release-2.2/channels.html>

Accessed: 07/09/2023.

References

[14] Hyperledger Fabric, “Peers.”

<https://hyperledger-fabric.readthedocs.io/en/release-2.5/peers/peers.html>

Accessed: 02/11/2023.

[15] Hyperledger Fabric, “Organization.”

<https://hyperledger-fabric.readthedocs.io/en/release-2.5/glossary.html>

Accessed: 06/11/2023.

[16] Hyperledger Fabric, “Fabric CA User’s Guide.”

<https://hyperledger-fabric-ca.readthedocs.io/en/latest/deployguide/cadeploy.html>

Accessed: 06/11/2023.

References

[17] M. G. Santhosh and T. Reshmi, “Enhancing pki security in hyperledger fabric with an indigenous certificate authority,” in 2023 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA), pp. 1–5, IEEE, 2023.

[18] D. Westerveld, API Testing and Development with Postman: A practical guide to creating, testing, and managing APIs for automated software testing. Packt Publishing Ltd, 2021.

[19] Postman, Inc, “What is Postman?”

<https://www.postman.com/product/what-is-postman/>

Accessed: 10/11/2023.

[20] OpenJS Foundation and expressjs.com contributors, “Fast, unopinionated, minimalist web framework for Node.js.” <https://expressjs.com/>

Accessed: 10/11/2023.

References

[21] OpenJS Foundation and Node.js contributors, “About Node.js.” <https://nodejs.org/en/about>

Accessed: 10/11/2023.

[22] C. Gy őr ödi, R. Gy őr ödi, G. Pecherle, and A. Olah, “A comparative study: Mongodb vs. mysql,” in 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), pp. 1–6, IEEE, 2015.

[23] MongoDB, Inc., “What Is MongoDB?” <https://www.mongodb.com/what-is-mongodb>

Accessed: 16/11/2023.

[24] Google, “The Go Programming Language Specification.” <https://go.dev/ref/spec>

Accessed: 20/11/2023.

References

- [25] J. Meyerson, “The go programming language,” IEEE software, vol. 31, no. 5, pp. 104–104, 2014.
- [26] Meta Platforms, Inc., “React.” <https://legacy.reactjs.org/>
Accessed: 20/11/2023.
- [27] A. Javeed, “Performance optimization techniques for reactjs,” in 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1–5, IEEE, 2019.
- [28] ngrok, “What is ngrok?.” <https://ngrok.com/docs/what-is-ngrok/>
Accessed: 02/11/2023.

References

- [29] Google for Developers, “Make your app the best it can be.” <https://firebase.google.com/>
Accessed: 10/12/2023.
- [30] L. Moroney and L. Moroney, “An introduction to firebase,” The Definitive Guide to Firebase: Build Android Apps on Google’s Mobile Platform, pp. 1–24, 2017.
- [31] Docker,Inc, “Docker overview.” <https://docs.docker.com/get-started/overview/>
Accessed: 10/12/2023.
- [32] T. Combe, A. Martin, and R. Di Pietro, “To docker or not to docker: A security perspective,” IEEE Cloud Computing, vol. 3, no. 5, pp. 54–62, 2016.

References

[33] A. M. Potdar, D. Narayan, S. Kengond, and M. M. Mulla, “Performance evaluation of docker container and virtual machine,” *Procedia Computer Science*, vol. 171, pp. 1419–1428, 2020.

[34] Docker,Inc, “Use containers to Build, Share and Run your applications.”
<https://www.docker.com/resources/what-container/>

Accessed: 22/12/2023.

[35] C. V. Helliard, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, “Permissionless and permissioned blockchain diffusion,” *International Journal of Information Management*, vol. 54, p. 102136, 2020.

[36] Ministry of Land, “Record of Rights (Khatiyan) (RoR).”
<https://minland.gov.bd/site/page/c14f084e-8974-4255-8183-bae66d436ebb/>

Accessed: 29/12/2023.

References

- [37] Ministry of Land, “Land Development Tax: Civil.” <https://ldtax.gov.bd/>
Accessed: 01/12/2023.
- [38] Q. Li and Y.-L. Chen, “Entity-relationship diagram,” in Modeling and analysis of enterprise and information systems, pp. 125–139, Springer, 2009.
- [39] X. Li, Z. Liu, and H. Jifeng, “A formal semantics of uml sequence diagram,” in 2004 Australian Software Engineering Conference. Proceedings., pp. 168–177, IEEE, 2004.

Thank You