




# Border Control and Immigration on Blockchain

Dhiren Patel<sup>1</sup>, Balakarhikeyan<sup>1</sup>, and Vasu Mistry<sup>2</sup>(✉) 

<sup>1</sup> Veermata Jijabai Technological Institute, Mumbai 400019, India

<sup>2</sup> National Institute of Technology, Surat 395007, India  
vasu5235@gmail.com

**Abstract.** In this paper, we propose a system using Blockchain technology to create a decentralized, secure, and scalable departure and arrival records of passengers. We provide a framework using Hyperledger Fabric, for maintaining the inter-port records of the passenger's entry and exit into a country as well as to facilitate gateless entry back to the passenger's country. We attempt to mitigate privacy and legal concerns over biometric data storage on the blockchain. We also explore the possibility of modifying the existing kiosks to work with the blockchain architecture at the backend so that passengers are not required to get familiar with a new procedure.

**Keywords:** Blockchain and distributed ledger technology · Immigration Gateless entry

## 1 Introduction

Borders between countries are strictly enforced to prevent illegal movement of people/goods into the country. A huge number of people cross international borders daily. This means effective, secure and scalable record keeping of entries and exits must be performed. For national security, it is crucial that these records are immutable to any attack/alterations. It is important to ensure that movement of people across borders happens easily and seamlessly. It is also imperative for nations to share their records to provide a strict and efficient control mechanism. At the same time these records must be securely stored complying with privacy laws and regulations of that country. This has made it all the more imperative to implement systems to alleviate all the above concerns.

The aim of this work is to implement a secure, decentralized, immutable seamless border control system to enable governments to easily and effectively log people exiting and entering their nations. This system also brings into sync every other port of entry into a unified decentralized system. It also aims to create separate data-paths for international transmission of departure records. The system will also include methods to securely store biometric information to validate/verify passengers automatically.

Rest of the paper is organized as follows: Sect. 2 discusses motivation based on Existing systems, strength of Blockchain Technology and enlists the Security Vulnerabilities in Existing Systems. Section 3 explains fundamentals and basics of Hyperledger framework. Section 4 discusses the proposed workflow for maintaining

arrival and exit records of the passengers. Section 5 discusses the implementation architecture on Hyperledger. Section 6 discusses Mitigating privacy and legal concerns over biometrics on the blockchain with Conclusions and References at the end.

## 2 Motivation and Background

### 2.1 Existing Border Control Systems

In a typical workflow for a passenger leaving a country via a port of exit he/she would be required to swipe passport at a passport scanner which records the details and then an immigration officer validates and stamps the exit out of the country. The passenger now proceeds towards the boarding gates and takes the flight to the destination.

The next stage is the submission of the Advanced Passenger Information System (APIS) data to the destination country by the flight carrier. The APIS was introduced by the US Customs and Border Protection and is a required criterion for many nations [1]. In India, each flight vessel is obligated to send the APIS data to the destination airport within 15 min of take-off from the origination point [2]. The UK government also has similar rules [3].

Similar systems are there for entry into a country. Common system operational in many countries including the United States are fast-tracked and quick-entry systems like the Global Entry Program. Global Entry allows rigorous background checks verified passport holders to skip lines and an immigration desk and walk to a Global Entry Kiosk to generate an exit pass. The kiosk scans the passports and collects fingerprints to verify authenticity of the passport holder [4].

Mobile passports have also made headway into certain nations as an easy process to clear immigration and skip the lines [5].

All of these systems have clearly contributed to ease of air-travel especially for citizens and have helped alleviate extensive screening and congestions at airports. Similar systems exist for entry through rail/sea. The downside being each of these systems have exposed us to new points of failures and security breaches.

The centralized structure, and the difficulty involved in keeping records securely synchronized across entry points are key issues the proposed system tackles. It also paves the way for nations to receive passenger information seamlessly without trust on a third-party. The proposed system allows nations to leverage biometric identities of their citizens to validate their entry. In addition the system can be integrated with existing automated kiosks making adoption easier.

### 2.2 Blockchain

Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is linked to the previous one after validation and consensus of all participating nodes. As new blocks are added, older blocks become more difficult to modify. New blocks are replicated across all copies of the ledger in the network, and any conflicts are resolved automatically using established rules [6].

At their most basic level, blockchain enables a community of users to record transactions in a ledger that is public to that community, such that no transaction can be changed once published. A block is an individual unit of a blockchain, composed of a collection of transactions and a block header. A block header keeps a collection of metadata about the block that contains a hash-value of its parent in the blockchain, and a hash of the aforementioned metadata and the data of the block itself [7].

In a public or permissionless blockchain anyone can participate without a specific identity. Public blockchains typically involve a native cryptocurrency and often use consensus based on “proof of work” (PoW) and economic incentives. Permissioned blockchains, on the other hand, run a blockchain among a set of known, identified participants. A permissioned blockchain provides a way to secure the inter-actions among a group of entities that have a common goal but which do not fully trust each other, such as businesses that exchange funds, goods, or information. By relying on the identities of the peers, a permissioned blockchain can use traditional Byzantine-fault tolerant (BFT) consensus [8].

### 2.3 Security Vulnerabilities and Weak Points in the Current System

The major associated problem with the current method of entry/exit is the centralized nature of data, making it an easy target of attacks and attempts at manipulation may cause complete or partial data loss.

With multitude of kiosks validating data and entering data into this centralized database, it could be disastrous if security flaws are found in the system.

The next point of failure might be the trust on the Airline carrier to report advance information of passengers’ arrival. This inherent trust might be misused along with existing automated passport control kiosks.

Lastly the entry/exit records are always prone to modification either by malicious third parties or due to internal political pressures etc. This data must be immutable.

The intercommunication between multiple port-of-entries of a country with a centralized database is also a serious cause of concern in terms of security and scalability.

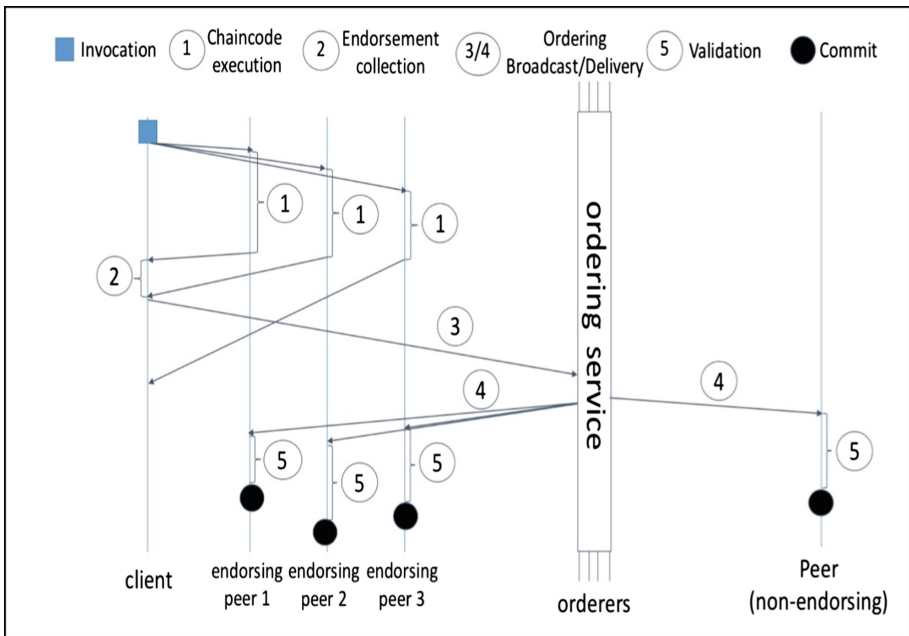
Apart from security issues, a simple lapse in stamping/recording of information might render a person as an invalid entrant into a country with no entry record and with no way of leaving the nation. This is especially true since many countries rely on a stamp on the passport to validate arrival, sometimes vis-a-vis maintaining a centralized record [9]. The adoption of this system prevents possibilities of such incidences especially for countries not maintaining centralized records.

## 3 Hyperledger Fabric

Fabric is a distributed operating system for permissioned blockchains that executes distributed applications written in general-purpose programming languages (e.g., Go, Java, Node.js) [8].

A distributed application using Fabric consists of two parts:

- A smart contract, called chaincode, which is program code that implements the application logic and runs during the execution phase. Special chaincodes are for managing the blockchain system and maintaining parameters, collectively called system chaincodes [8].
- An endorsement policy that is evaluated in the validation phase. An endorsement policy acts as a static library for transaction validation in Fabric, which can merely be parameterized by the chaincode. Only designated administrators may run system management functions and have the right to modify the endorsement policy [8] (Fig. 1).



**Fig. 1.** Hyperledger Fabric transaction workflow [8]

As Fabric is permissioned, all nodes that participate in the network have an identity. Nodes in a Fabric network take up one of three roles:

- ‘Clients’ submit transaction proposals for execution, help orchestrate the execution phase, and, finally, broadcast transactions for ordering [8].
- ‘Peers’ execute transaction proposals, validate transactions and maintain the blockchain. Only the ‘Endorsing peers’ execute all transactions while all peers maintain the blockchain ledger [8].

- ‘Ordering Service Nodes’ (OSN) (or, simply, orderers) are the nodes that collectively form the ordering service. The ordering service establishes the total order of all transactions in Fabric, where each transaction contains state updates and dependencies computed during the execution phase, along with cryptographic signatures of the endorsing peers that computed them [8].

## 4 Proposed Workflow of Border Control System on Blockchain

### 4.1 Maintaining Entry/Exit Records on the Blockchain

The proposed workflow is similar to the current workflow with an essential difference that the immigration officer now marks his immigration decision which is recorded onto the blockchain. Once all details are recorded the passenger is allowed to pass through.

- In case of a departure workflow, the system automatically finds the corresponding arrival record into the country for a foreign citizen and validates this departure.
- In case of arrival, the system automatically finds the corresponding departure record of the citizen and validates his entry.

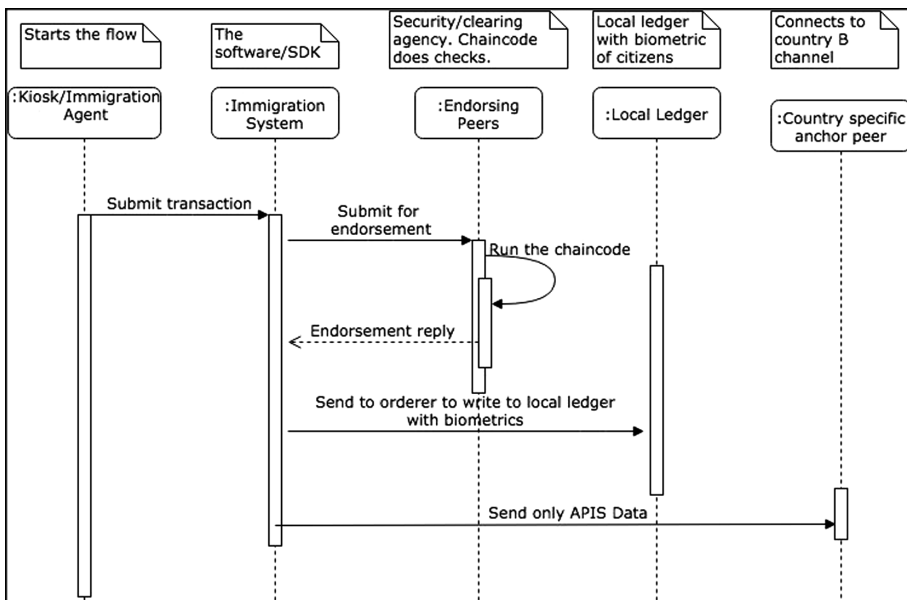


Fig. 2. Departure sequence diagram

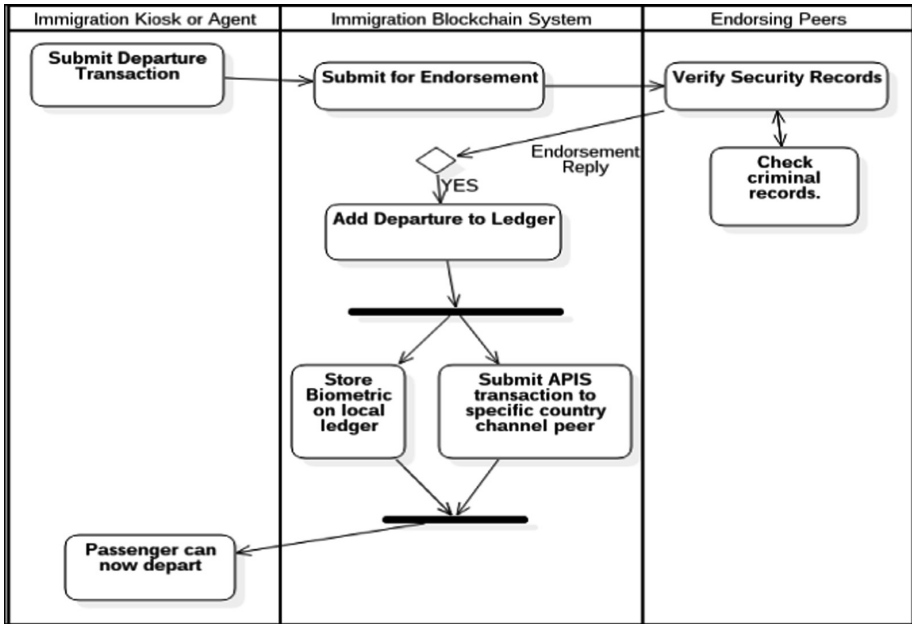


Fig. 3. Departure activity diagram

### 4.2 Detailed Example Workflow During Departure

Referring to Fig. 2 we see that the immigration officer submits a transaction for departure to the system. Here, a transaction refers to the departure of the passenger.

The transaction is then sent to endorsement to the corresponding endorsement peers. The endorsement peers could be from local security agencies to a central no-flyer database etc. Once the endorsements are successful, the entry is added into the departure ledger and biometric information stored in the local-ledger if the departing passenger is a citizen of the given country. In case of a rejection, it is added to a separate log with rejection comments by the immigration officer. The departure sequence diagram shows as various stages through with the system passes and its interaction with the ledger.

The two critical phases involve capturing and storage of biometrics securely to validate a passenger on return and to post the APIS data on to the specific country anchor peer so that the destination country is aware of the passenger. The robustness of the system lies in the fact that since there is no central machinery involved all ledgers in the airport work in a distributed fashion holding everyone’s records. Additionally all other ports in the nation are also now aware of this departure and biometric record which can be validated in case of a citizen on his return. Thus the system lacks central failure points but at the same time maintains a copy accessible with everyone.

A similar system is followed during the arrival of a passenger as noted in Fig. 4. The passenger admittance is subject to endorsement from the endorsing peers. This includes security agencies and validating departure record for a citizen. For an

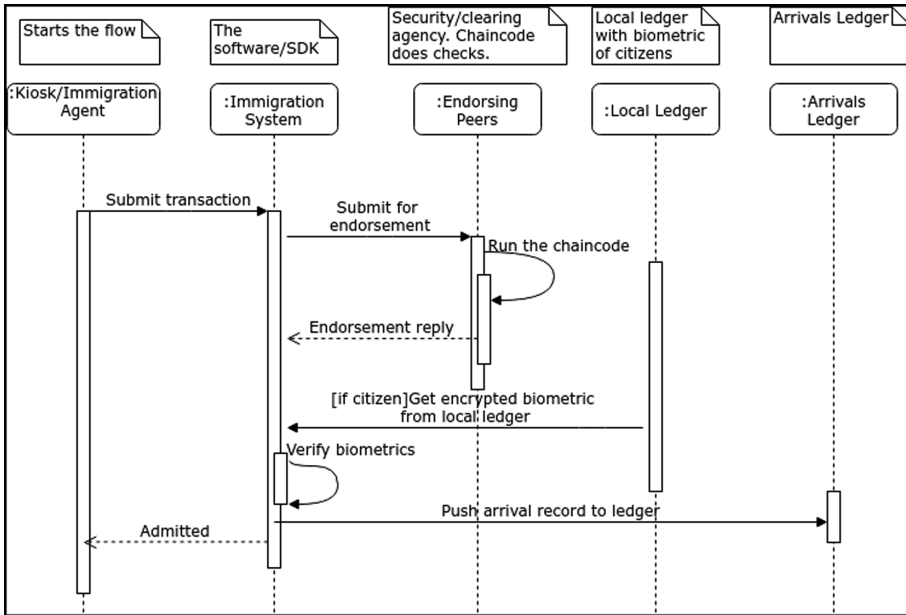


Fig. 4. Arrival sequence diagram

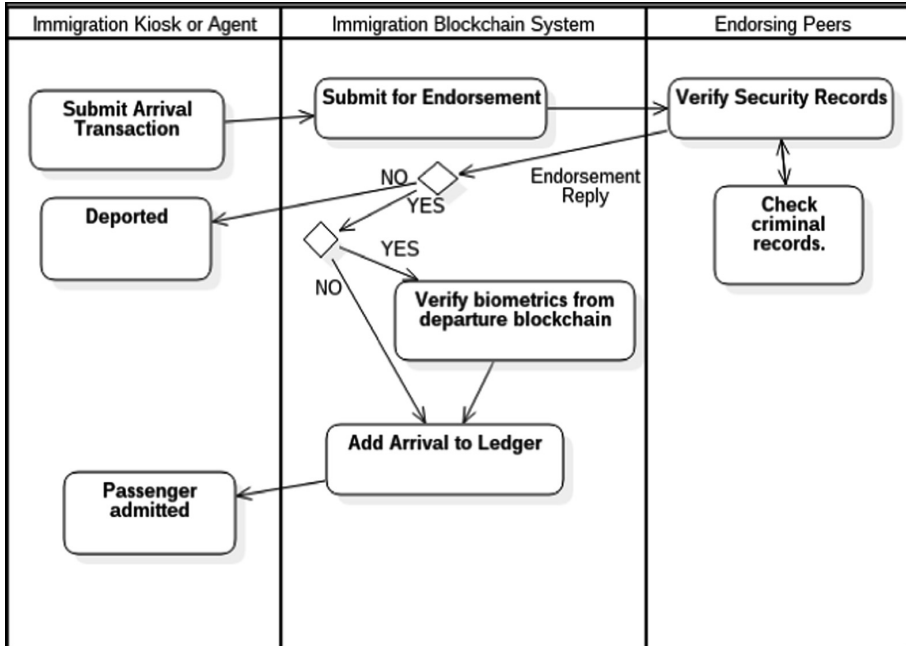


Fig. 5. Arrival activity diagram

incoming citizen, additionally biometrics shall be compared before final admission. This automates the procedure for citizens and removes involvement of immigration officers who now will have a supervisory role.

The following points shall be noted from the swim lane diagrams:-

- The check for corresponding arrival/departure records happens at an endorsement peer in the defined chaincode.
- Each security agency can define chaincodes, which can search databases and/or perform complex actions to complete security procedures and provide with an endorsement.
- The immigration officer can chose not to submit the transaction and directly reject it. Rejections are recorded in a separate peer node.
- The system can integrate with existing passport-control kiosks, with minor re-configuration. The kiosks act as the client triggering the transactions.

## 5 Implementation Architecture

Hyperledger is the preferred choice for this use case as it provides a fast and scalable system with features complementing the specific needs and deployment of permissioned blockchains.

Hyperledger does not require mining and instead uses Endorsement for the Ordering Services. The Hyperledger Certificate Authority (CA) allows developers to enroll peers using existing public key infrastructure.

The endorsement policy ensures that the Immigration officer might not be the only endorser for approving a passenger (in this case, giving a vote as to whether the person gets in or not), additional security agencies can be made part of this endorsement process. Since endorsement chaincode can be written in non-deterministic languages like Go/Java, a quick Banned Flier list data lookup can be performed as an additional endorsing peer.

Each port maintains a minimum of one peer node (Immigration officer endorsing peer) although multiple peers can be maintained.

Channels provide a way to replace the APIS system with the destination port's receiving peer which can now easily be notified of arrival of passengers with all APIS data encoded on to it.

The following Fig. 6 shows the deployment architecture of the system. Here, symbols are derived from Hyperledger's default set of symbols for showing interactions. The oval represents an organization. Peers in an organization are always connected to each other.

- In this deployment, we see Port-1, Port-2 and Endorsing Organizations on a common channel.
- A separate organization called Airports Authority is maintained as the peer to send APIS information to specific destination countries. The chaincode in this peer shall be invoked by the system after successful departure as shown in Fig. 4.



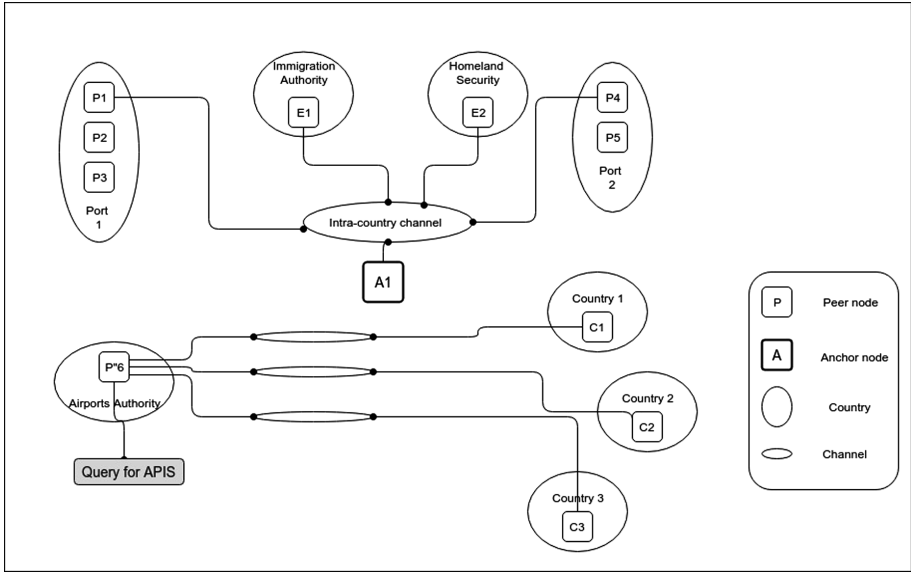


Fig. 6. Deployment architecture on Hyperledger

- For the sake of brevity, ordering services on the channel are omitted in the diagram but are assumed to exist.

The system will integrate with existing kiosks/immigration desks as they will act as clients interacting with the Immigration System APIs as shown in Figs. 3 and 5.

The data stored on the ledger shall be the basic details as recorded by APIS systems internationally. This include names, passport number, source and destination ports, along with carrier vessel identification and boarding pass details [2].

In addition the passenger’s country would store biometric information to validate the passenger seamlessly on his return.

### 5.1 Pseudocode for the Immigration System

We provide the pseudocodes using the Hyperledger Composer CTO language containing the `model` and the `script` which forms the chaincode in Composer. Hyperledger Composer is a fast and rapid prototype deployment tool to be used with Hyperledger Fabric.

In Hyperledger Composer all interactions are performed using participants and assets. Assets are entities that change their state during the course of a transaction. The assets are recorded and saved on the blockchain as well as the transactions [10].

We model the system in terms of a passenger asset due to the constraints of Hyperledger CTO language. It must be noted that in a real installation we need not model it in a similar way but it is a good representation to store the details of a passenger and his passport. A passenger asset is identified by his `passportNo` as a unique key. In addition we show participants immigration-officer-peer node and the

endorsing agency as assets. In a real deployment we might have multiple endorsing agencies. We also have defined our depart transaction here and its attributes. Similarly a transaction for arrival can be created.

### Composer – Model (Fig. 7)

---

#### Pseudocode 1 Composer Model

---

```

1: namespace org.india.immigration
2: asset Passenger identified by passportNo:
    passportNo
    firstName
    lastName
    middleName
    passportExpiry
    visaNo
    destCountry
    depCountry
    depStatus
    endorsers
3: Port
    portNo
    portName
4: participant ImmigrationOfficerPeer identified by officerID
    officerId
    officerName
    port
5: participant EndorserAgency identified by agencyCode
    agencyCode
    agencyName
6: transaction Depart
    departId
Passenger p
ImmigrationOfficerPeer i
    officerRemark

```

---

Fig. 7. Hyperledger composer – model pseudo-code

### Composer – Script (Fig. 8)

```

/**
 * A sample Immigration Script API to submit a passenger
 departure transaction.
 * A dummy endorsement function is shown. This is just to
 model it like a hyperledger endorsement returning from
 * a chaincode running at a hyperledger endorsement peer.
 */

```

---

**Pseudocode 2** Composer Script
 

---

**Input:**

```

    tx: Transaction data
1: function GETENDORSEMENT( tx )
2:   if (tx and passenger data are valid) then
3:     return "Departure Granted"
4:   else
5:     return "Departure Not Granted"
6:   end if
7: function DEPART( tx )
8:   tx.p.departure_status ← GETENDORSEMENT(tx)
9:   tx.p.endorsers.push(endorserId)
10:  assetRegistry ← GETASSETREGISTRY('org.india.immigration.Passenger')
11:  return assetRegistry.update(tx.p)

```

---

**Fig. 8.** Hyperledger composer – script pseudo-code

The Script shown here is representative of our Immigration System deployment and in an actual use case the client machines shall interface with the API of our Immigration System to carry out the transaction i.e. letting a passenger depart and our arrive and collecting valid endorsements etc. The Client only needs to call the API to initiate the transaction and wait for results to come back. The system will do all the business logic processing and call appropriate chaincodes to execute the transaction.

Since Hyperledger Composer does not currently support custom Endorsement policies we have created a dummy endorsement function. In real life as soon as the transaction is submitted the endorsing peers would run their chaincodes to validate the transaction and such a dummy function is not needed. Here the function only serves to remind us about how the transaction would be endorsed.

The `Depart(transaction)` sets the departure status on the passenger asset and updates the asset.

```

1  {
2  "$class": "org.india.immigration.Depart",
3  "departId": "001",
4  "p": "resource:org.india.immigration.Passenger#5523",
5  "i": "resource:org.india.immigration.ImmigrationOfficerPeer#7609",
6  "officerRemark": "ECNR, Ok to Board",
7  "transactionId": "f3fa7a0f-e510-4c1d-8893-9b7948b2a329",
8  "timestamp": "2018-03-31T06:34:29.586Z"
9  }

```

**Fig. 9.** Demo transaction

Figure 9 shows a sample executed transaction summary based on our initial model on the Composer Playground Web UI [11]. Here we see that the passenger departure is logged with an ID for easy query. Along with it the passenger asset and the immigration officer asset is also logged. The `officerRemark` along with timestamp and `transactionID` show a valid departure status.

## 6 Addressing Privacy and Legal Concerns Over Biometrics on the Blockchain

A very important question arises with regards to privacy of the passenger's biometric information and legal issues with its dissemination. Firstly, no biometric information shall be transmitted outside the passenger's country and shall be stored as per the laws of the country e.g. Aadhaar UID system in India.

Countries like Germany provides users an option to use biometric passwords and the data can be stored in double RSA hashing to implement the same. The above pseudocode provides a small example of using Double RSA encryption technique. The code can be easily written in Javascript, by utilizing the popular RSA library called JSEncrypt [12]. Here a public-private key pair based on passport information or other has to be generated for a passenger so that his private key is needed for reading his biometric data from the stored asset. The `biometricHash` in the stored asset corresponds to this encrypted biometric data (Fig. 10).

---

### Pseudocode 3 Double RSA technique

---

**Input:**

`biometricData`: An object holding biometric data.

- 1: **procedure** ENCRYPTION(`biometricData`)
- 2:     `crypt1`  $\leftarrow$  ENCRPYT()
- 3:     `crypt2`  $\leftarrow$  ENCRPYT()
- 4:     `crypt1.setPublicKey(PassengerPublicKey)`
- 5:     `crypt2.setPublicKey(ImmigrationAgencyPublicKey)`
- 6:     `biometrics`  $\leftarrow$  `biometricData`
- 7:     `enc1`  $\leftarrow$  `crypt1.encrypt(biometrics)`
- 8:     `encFinal`  $\leftarrow$  `crypt2.encrypt(enc1)`                   ▷ Double RSA encryption
- 9:     **return** `encFinal`

**Input:**

`encFinal`: Holds cipher text after performing double RSA.

- 10: **procedure** DECRYPTION(`cipherText`)
  - 11:     `crypt1.setPrivateKey(PassengerPrivateKey)`
  - 12:     `crypt2.setPrivateKey(ImmigrationAgencyPrivateKey)`
  - 13:     `dec1`  $\leftarrow$  `crypt2.decrypt(encFinal)`
  - 14:     `biometricData`  $\leftarrow$  `crypt1.decrypt(dec1)`
  - 15:     **return** `biometricData`
- 

**Fig. 10.** Double RSA for biometric data pseudo-code

The first round of hashing is performed by the passenger's private key while an additional hashing is performed by the Border control agencies private key. Passenger's private key can be generated from his own biometrics/passport, making it mandatory for the passenger to be present to enable decryption of his record. Thus although passenger's biometric identity might exist on the blockchain without the passenger such data cannot be accessed and utilized thus alleviating any concerns over privacy of this sensitive data.

## 7 Conclusions

We have looked at an interesting use case of global nature and implemented a secure, decentralized, immutable seamless border control system to enable governments to easily and effectively log people exiting and entering their nations.

In this paper we have addressed the security and reliability issues of current systems. Using blockchain on Hyperledger, this system brings into sync every other port of entry into a unified decentralized system. We have shown creation of separate data-paths for international transmission of departure records. A permissioned infrastructure is envisioned where the government security agencies act like gate-keepers (endorsers) automatically allowing entry/exit. We have also envisioned storage of biometrics in a local ledger allowing citizens to easily enter their own country while also keeping privacy concerns at bay. Currently the system is under active development. We look forward to further improve the system with simple, automated interfaces.

## References

1. Advanced Passenger Information System Air Canada. <https://www.aircanada.com/us/en/aco/home/plan/travel-requirements/advancepassenger-information.html>. Accessed 20 Feb 2018
2. APIS, Bureaus of Immigration India. <https://boi.gov.in/content/apis-advanced-passenger-information-system>. Accessed 11 Mar 2018
3. Transfer e-Borders data: general aviation and maritime. <https://www.gov.uk/government/publications/transfer-e-borders-data-general-aviation-and-maritime>. Accessed 20 Feb 2018
4. Global Entry U.S. Customs and Border Protection homepage. <https://www.cbp.gov/travel/trusted-traveler-programs/global-entry>. Accessed 7 Mar 2018
5. Mobile Passport Control App U.S. Customs and Border Protection. <https://www.cbp.gov/newsroom/national-media-release/new-mobile-passport-control-app-available>. Accessed 7 Mar 2018
6. Yaga, D., Mell, P., Roby, N., Scarfone, K.: Blockchain technology overview. Draft NISTIR 8202, NIST, U.S. (2018)
7. Wurster, S., et al.: Specification on blockchain technology. ISO/TC 307, Tokyo (2017)
8. Cachin, C., Barger, A., Manevich, Y.: Hyperledger fabric: a distributed operating system for permissioned blockchains. <https://arxiv.org/abs/1801.10228v1>. Accessed 20 Mar 2018
9. No arrival stamp on Indian's passport at Mumbai airport. <https://timesofindia.indiatimes.com/city/mumbai/No-arrival-stamp-on-Indians-passport-at-Mumbai-airport-he-cant-return-to-UAE/articleshow/47930826.cms>. Accessed 15 Mar 2018

10. Hyperledger Composer. <https://hyperledger.github.io/composer/latest/reference/reference-index>. Accessed 28 Mar 2018
11. Hyperledger Composer Playground. <https://composer-playground.mybluemix.net/>. Accessed 31 Mar 2018
12. JSEncrypt. <http://travistidwell.com/jsencrypt/>. Accessed 28 Mar 2018