City University of Hong Kong

IS6200 – Blockchain Technology and Business Application

Group Project Final Report

City Express Application Based on Blockchain (Design)

Group name: Rookie Miners

Group Members:

XU	Rui	(Student ID: 56365524)
HONG	Yuan	(Student ID: 56373801)
WANG	Yuchun	(Student ID: 56532945)
HON	Yuen Ching	(Student ID: 56255901)
LU	Jie	(Student ID: 56509782)

Instructor: Prof. ZUO Zhiya

28th April 2021

Contents

1. Introduction of the Project

1.1 Overview

Our project is Rookie City Express (RCE), which is a peer-to-peer intra-city delivery platform based on blockchain technology. The main target is to provide users a decentralized, trustless, reliable online express system for both sending and receiving packages. Thanks to Hyperledger, an open source blockchain technology, the marked features of RCE like automatic billing and confidential signing will be realized in this application via smart contract.

1.2 Use Case Analysis

1.2.1 Industry Description

In modern life, everyone is chasing after the high efficiency of work to make the revenue maximization. That is why some new industries like express and delivery occur. But the process of the express is always complex, and it will take at least one day to deliver the package even the sender and recipient are in the same city. City express is aiming at the transportation which happens within city. The sender creates an order, the deliveryman will come and pick up the goods at once then delivery it as soon as possible. It only takes hours to finish the transportation in average, sometimes even half an hour, which improves the efficiency of express to a large extent.

1.2.2 Industry Pain Points and Challenges

Current pain points in express industry:

1. Lack of automatic billing system based on package weight, delivery distance and time.
2. Risk of illegal modifications of package information and order detail.
3. Risk of losing packages and delivery errors, like signed by a wrong person.
4. Less protection for users' information resulting in privacy leakage.
5. The delay and inaccuracy of the location of the packages.
6. Too much trust needed in the process between users and express companies.

1.2.3 Blockchain Assessment Framework

Unlike the traditional delivery platform with many manual processing of data, our platform provides the automated audit trail of the orders, so it's possible to update the state of the transaction in time, which can make transactions faster and cost less.

Based on the smart contracts of the blockchain, there are briefly rules in our platform. Since the orders update only when transactions are approved by the appropriate participants, the risk of fraudulent transactions is reduced. Moreover, since the transaction in blockchain is immutable and irreversible, there are no modify risk in our platform.

With the hash and the signature in blockchain, the platform can ensure the safety of the transaction. One the other hand, the privacy of user is provided by the private key.

1.2.4 Advantages and Disadvantages of Blockchain Technology in Proposed Industry

Advantages:

1. The blockchain tech will make the process of the whole transportation more automagical.
2. The encryption tech can protect the customers privacy.
3. The non-modification will ensure the goods would not be changed during the delivery.
4. The digital signature could ensure the goods would not be taken by others.

Disadvantages:

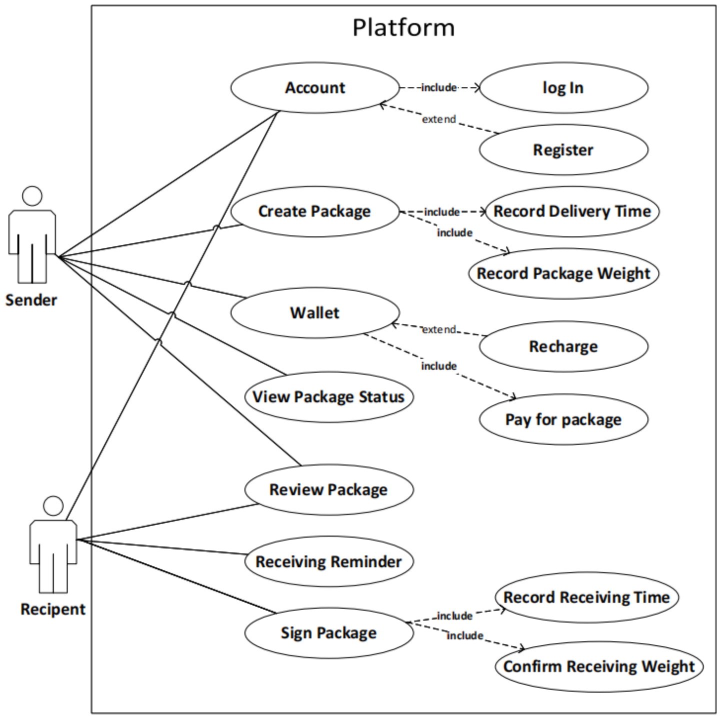
1. Some specific features like the quality of the goods are difficult to be assessed. Only the weight can be detected then in currently.
2. Because of the protection of the privacy, there may exist some transportations which are illegal.
3. Users need to buy particular crypto coins used in the platform before they create orders, which maybe consider as inconvenient.

2 Design of Blockchain Application

RCE is an online peer-to-peer delivery platform for users in a same city, which provides express service with smart billing solution, immutable order records, and valid signing guarantee. RCE will be deployed on IBM Hyperledger Fabric blockchain platform.

2.1 Use Case Demonstration

As the use case diagram below illustrated, this RCE application has 2 kinds of users: sender and recipient, as well as 7 use cases. Both sender and recipient can build an account to log in or register and review the packages before. The sender can create a package including the delivery time and the sending weight of the packages, and he can also view the status of the package like where the package is located by the IoT device which will send the location and other information to the system in real time. The recipient can sign the package including recording the receiving time, accepting the reminder of package's receiving and confirming receiving weight.



2.2 Entity-relation Diagram

2.2.1 Entity

- User

The user of the blockchain application is one who registers on our platform using their telephone number or email. Only people with registration can access the full service. The entity contains seven attributes: UserID, Name, Gender, Password, Email, Telephone and City.

- Sender

The people who create an order and send out the package would be the sender. It is a subtype of User. Besides the attributes of user, it contains the SenderID and Balance.

- Recipient

The people who receive the package would be the recipient. It is a subtype of User. Besides the attributes of user, it contains the ReciepientID.

- Order

This table records the basic information of the order the sender created. It concludes the OrderID, Delivery time, Receiving time, ShippingAddress, ReceivingAddress, SenderTelephone and RecipientTelephone.

- Package

This is what the order delivers. This table records the PackageID, Weight and Type of the package.

2.2.2 Relationship

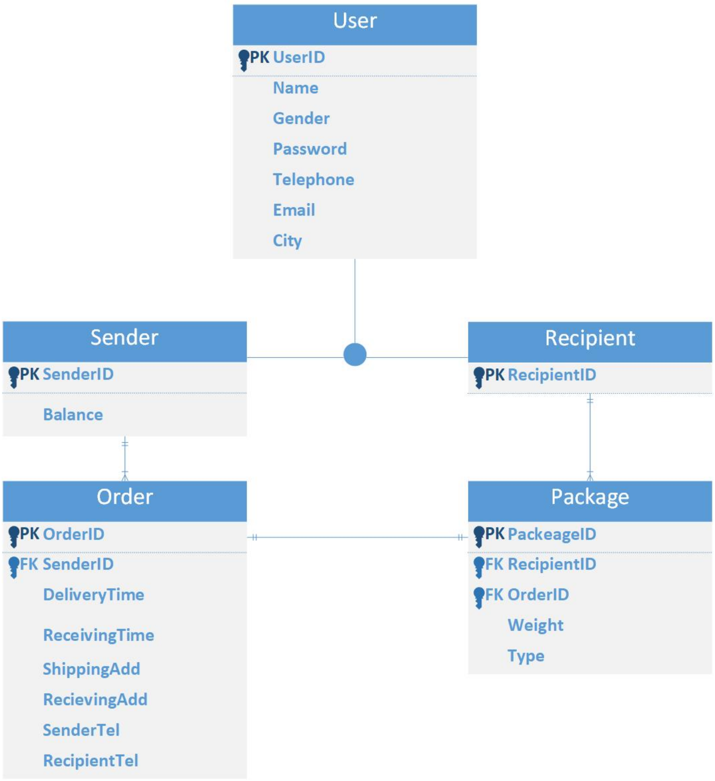
- One to One relationship

Order and package: One order can only contain one package; one package can only belong to one order.

- One to Many relationship

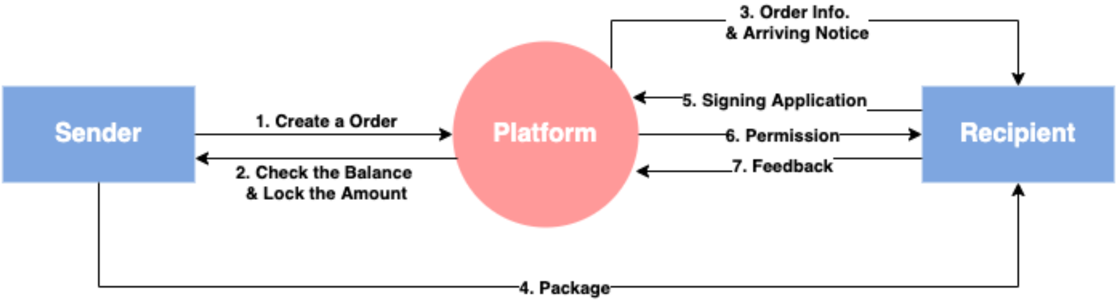
Sender and order: One sender can create many orders to deliver the package, one order can only be created by one sender.

Recipient and package: One package can only be sent to one recipient, but one recipient can receive many packages.



2.3 Data Flow Diagram

In the data flow diagram (DFD) below, there are two participants and a platform. The precondition is that senders buy crypto coins used in the platform and charge into their account wallet, which is not shown in diagram. Then senders create the order on the platform, and platform will check and lock senders' balance. The order information will circulate between the recipients and platform. Once recipients get permission and sigh the package, the transaction finished, the records of payment and balance will circulate between them.



2.4 Business Architecture

2.4.1 Create an Order

Once an order is created by a sender, fabric channel will receive the order information including delivery time, location information (shipping address, destination), package information (category, weight), ID of the sender and the recipient. Notice that if the account is in arrears, this order will not be conducted until the sender recharges.

Also, with the help of IoT, from this moment, the sensor devices embedded in the package will keep reporting its location and other critical indicators continuously.

2.4.2 Apply to Sign

Once a package is delivered to the destination address, the recipient will be noticed to apply for sign. The application information contains the order information and its encrypted version by the private key of the recipient, so that the identity of the recipient will be verified by decrypting the encrypted information with the public key of the recipient recorded the order. Meanwhile, the applying time will be recorded as the receiving time.

2.4.3 Sign for a Package

Once the correct recipient is verified, the package will be unlocked. Then, the package information can be confirmed by the recipient. If the real status of the goods, like category, weight, description, etc., is the same as that of initial status before delivery, the recipient will give a feedback to the platform.

2.4.4 Confirm a Payment

Once the platform receives the recipient's positive feedback, this order will be confirmed, and the corresponding amount of money, which is defined by smart contract, will be deducted from the sender's bank account. The formula is as follows:

$$C = k * W * D / T$$

Where C is the cost of an order, k is a constant coefficient, W is the weight of package, D is the distance between shipping address and receiving address, T is the interval time between delivery and receiving.

3 Deployment of Blockchain Application

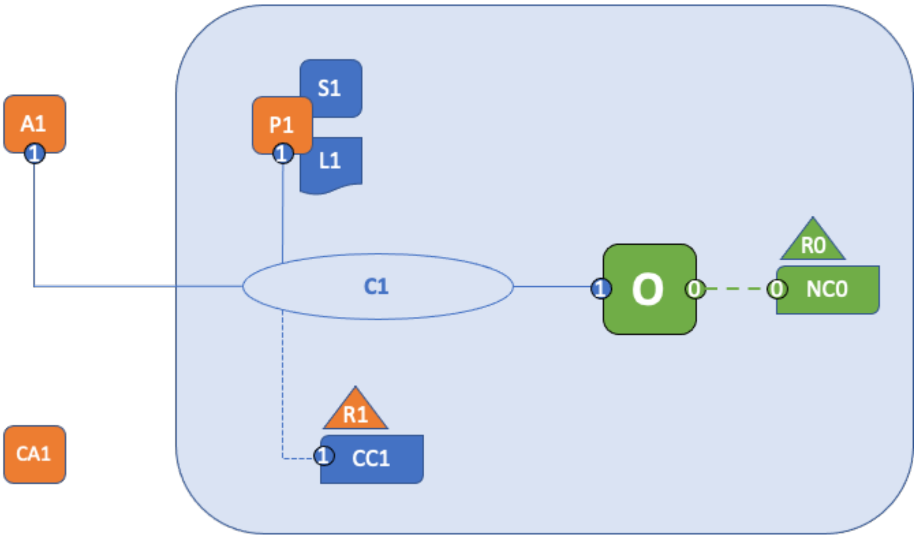
Nowadays, a large number of blockchain frameworks have emerged in industry and academia, which can be made use of in this project. IBM, a pioneer company in the blockchain field, provides us many useful blockchain development tools. In this stage, Hyperledger will be applied for the RCE deployment.

3.1 Blockchain Framework

In this stage, Hyperledger will be applied for the RCE deployment. It is a user-friendly platform to develop the blockchain applications. Generally, the composer could be used to create the business network definition, as well as determining the logic of our transaction and access control rules of blockchains. We will define the transactions, and participants and assets involved in the transaction in the model file. The participants are those users include senders and recipients who create orders to deliver packages and receive packages. In our particular application, the assets are only the packages being transported. Also, the packages will have their own unique identifiers.

3.2 Blockchain Network Diagram

Hyperledger Fabric and Composer allows organizations to collaborate in the formation of blockchain networks. In order to configure and launch RCE network, some typical elements should be declared above all.



As shown in this diagram, organization R0 represent our project group who define the network order server, and organization R1 is RCE which defines the channel and smart contract of the business system. Peer node P1 represents the management staff of RCE who has access to record transaction on ledger of the express system. Users of RCE can connect to the channel by application A1. Moreover, there is a certification authority CA1 for R1 to use when performing transaction.

3.3 Business Network Deployment

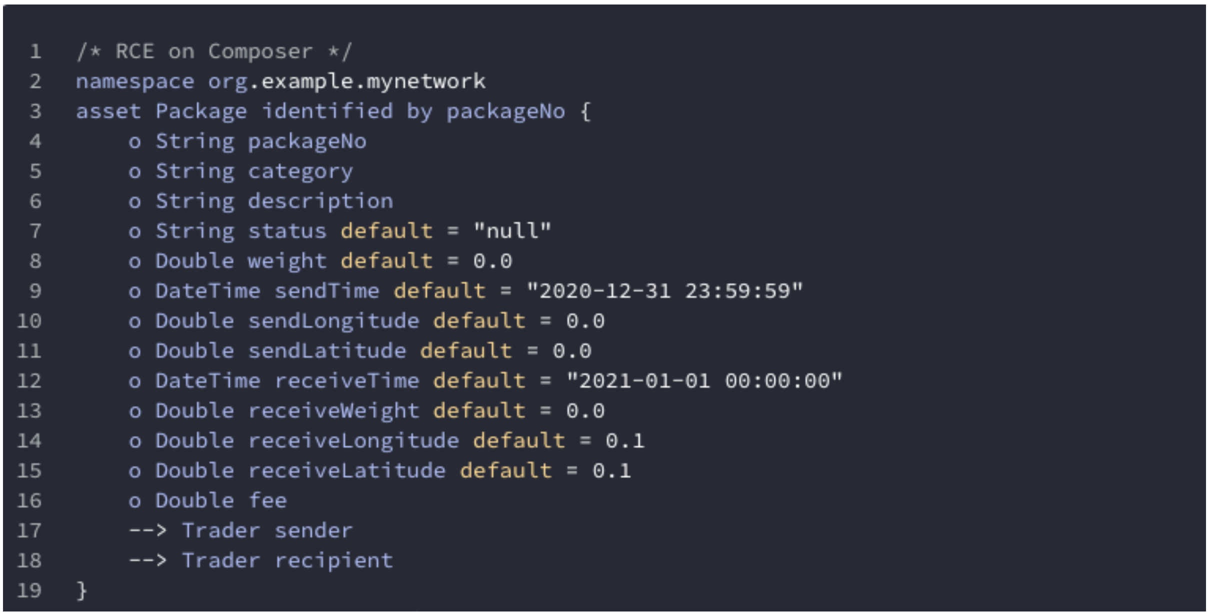
3.3.1 Model Definition

With the convenience of Hyperledger Composer, a simple network constructure defined by model file, script file and access control file.

Firstly, in the model file, basic class asset, participant and transaction are defined. As the code shows, an asset class called **Package** is created which has following attributes:

packageNo (the number that identify a certain package), category, description, weight (default set as 0), sendTime, sendLongitude, sendLatitude, receiveTime, receiveWeight, receiveLongitude, receiveLatitude, fee, sender and recipient.

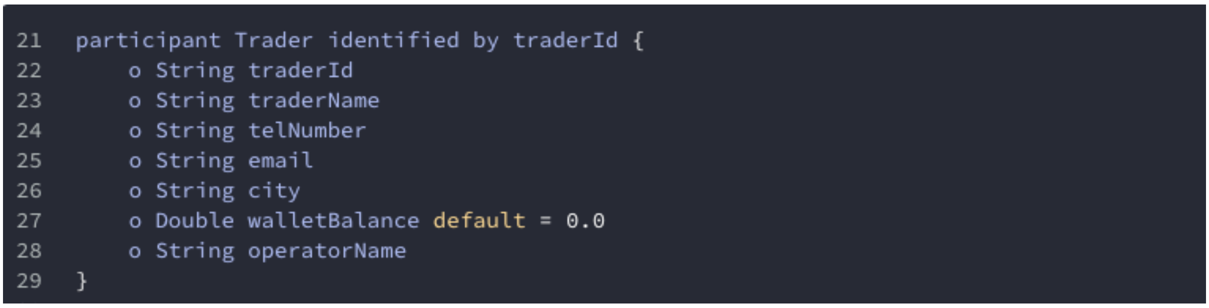
The attribute named status is to describe the status of the package, like "On Shipping" or "Already Signed". Additionally, if with the IoT interface and help of GPS and other sensors, the package information like location, temperature and humidity will be synchronized after the package is sent out.



```
/* RCE on Composer */
namespace org.example.mynetwork
asset Package identified by packageNo {
    o String packageNo
    o String category
    o String description
    o String status default = "null"
    o Double weight default = 0.0
    o DateTime sendTime default = "2020-12-31 23:59:59"
    o Double sendLongitude default = 0.0
    o Double sendLatitude default = 0.0
    o DateTime receiveTime default = "2021-01-01 00:00:00"
    o Double receiveWeight default = 0.0
    o Double receiveLongitude default = 0.1
    o Double receiveLatitude default = 0.1
    o Double fee
    --> Trader sender
    --> Trader recipient
}
```

For the participant part, a **Trader** class is created with attributes:

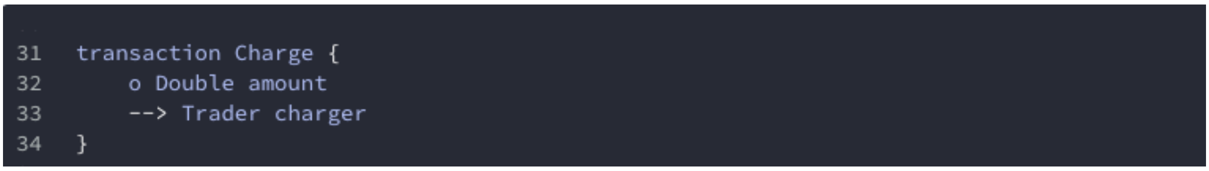
traderId (the number that identify a certain trader), traderName, telNumber, email, city, walletBalance (default set as 0.0), operatorName.



```
participant Trader identified by traderId {
    o String traderId
    o String traderName
    o String telNumber
    o String email
    o String city
    o Double walletBalance default = 0.0
    o String operatorName
}
```

Next, the transaction named **Charge** is claimed with parameters:

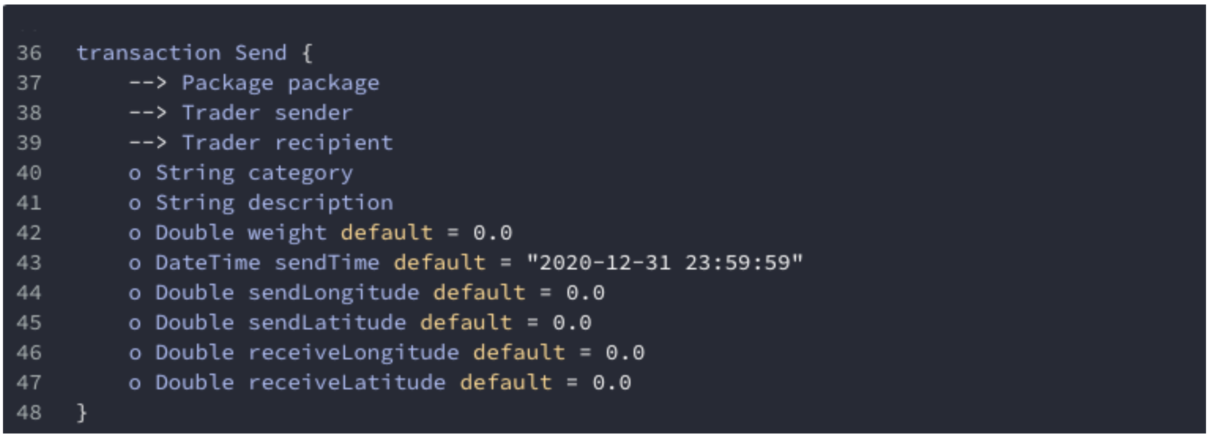
Amount (the amount of a charging operation), charger (the trader whose balance get charged)



```
transaction Charge {
    o Double amount
    --> Trader charger
}
```

The second transaction is **Send** defined with parameters:

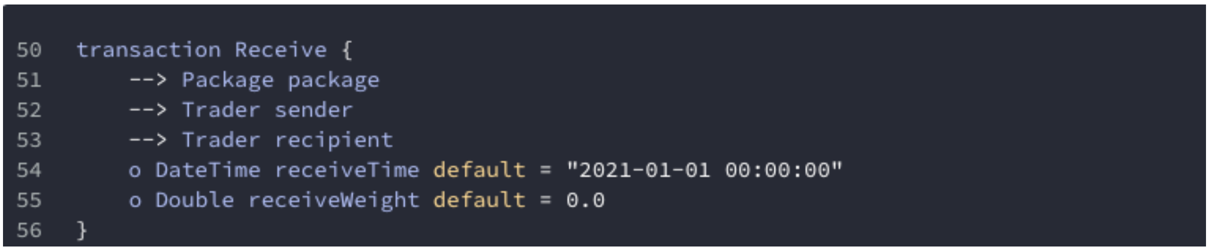
package, sender, recipient, category, description, weight (default set as 0.0), sendTime, sendLongitude, sendLatitude, receiveLongitude, receiveLatitude.



```
transaction Send {
    --> Package package
    --> Trader sender
    --> Trader recipient
    o String category
    o String description
    o Double weight default = 0.0
    o DateTime sendTime default = "2020-12-31 23:59:59"
    o Double sendLongitude default = 0.0
    o Double sendLatitude default = 0.0
    o Double receiveLongitude default = 0.0
    o Double receiveLatitude default = 0.0
}
```

The last transaction is called Receive with input as:

package, sender, recipient, receiveTime, receiveWeight.

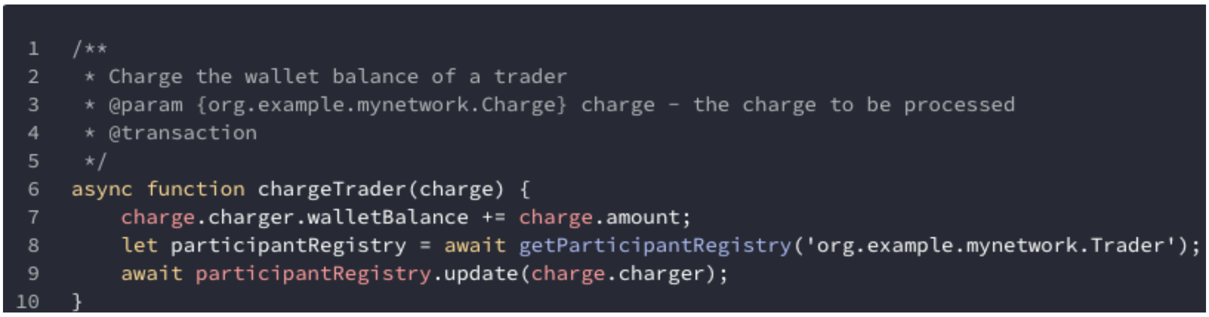


```
transaction Receive {
    --> Package package
    --> Trader sender
    --> Trader recipient
    o DateTime receiveTime default = "2021-01-01 00:00:00"
    o Double receiveWeight default = 0.0
}
```

3.3.2 Smart Contract

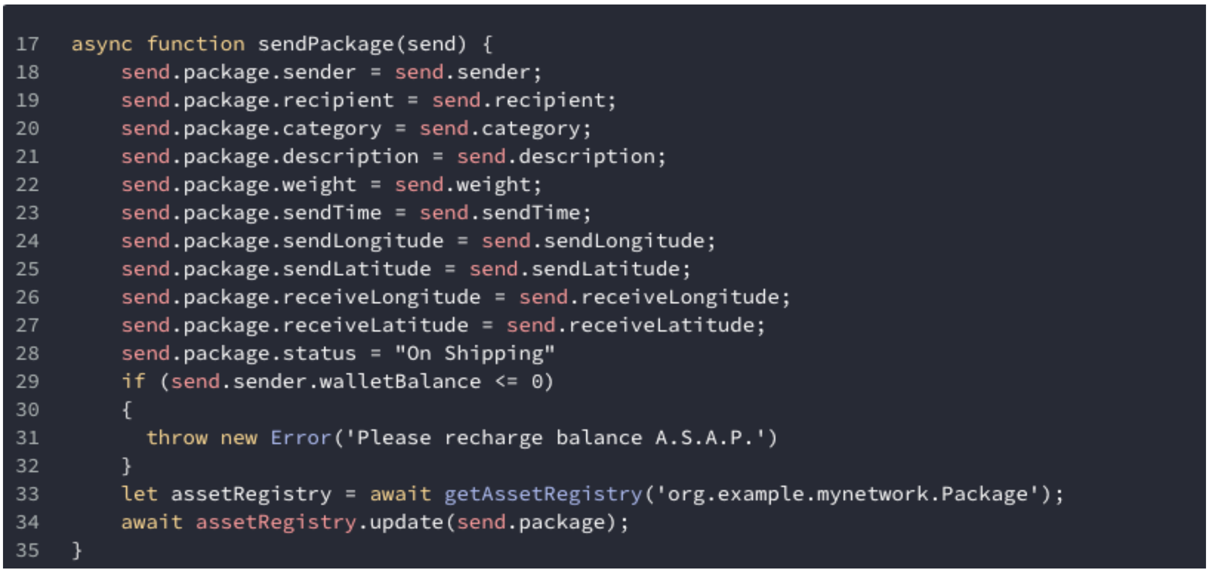
Smart contract or known as chain code is defined by JavaScript file. In this stage, the logical of each function defied in the .cto file will be specified.

As the .js code attached, the first function is **charge**, with the input **amount** and trader identified by **traderID**, the walletBalance of the trader will be accumulated.



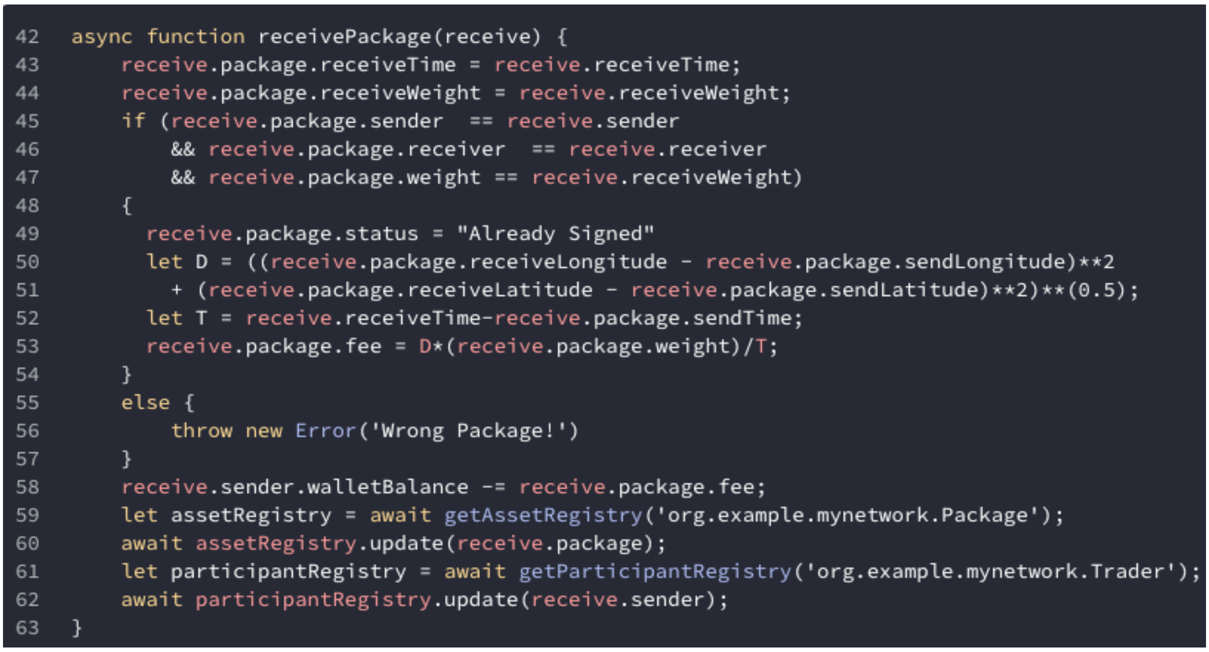
```
/**
 * Charge the wallet balance of a trader
 * @param {org.example.mynetwork.Charge} charge - the charge to be processed
 * @transaction
 */
async function chargeTrader(charge) {
    charge.charger.walletBalance += charge.amount;
    let participantRegistry = await getParticipantRegistry('org.example.mynetwork.Trader');
    await participantRegistry.update(charge.charger);
}
```

And for the **send** function, the input parameters like shipment information (**sender, recipient, sendTime, sendLongitude, sendLatitude, receiveLongitude, receiveLatitude**), package information (**category, description, weight**) in this step will be passed to the package instance. And errors will come out if the sender's account is less than or equal to zero. Also, the package **status** can be updated per minute based on the location information returned via IoT interface.



```
async function sendPackage(send) {
    send.package.sender = send.sender;
    send.package.recipient = send.recipient;
    send.package.category = send.category;
    send.package.description = send.description;
    send.package.weight = send.weight;
    send.package.sendTime = send.sendTime;
    send.package.sendLongitude = send.sendLongitude;
    send.package.sendLatitude = send.sendLatitude;
    send.package.receiveLongitude = send.receiveLongitude;
    send.package.receiveLatitude = send.receiveLatitude;
    send.package.status = "On Shipping"
    if (send.sender.walletBalance <= 0)
    {
      throw new Error('Please recharge balance A.S.A.P.')
    }
    let assetRegistry = await getAssetRegistry('org.example.mynetwork.Package');
    await assetRegistry.update(send.package);
}
```

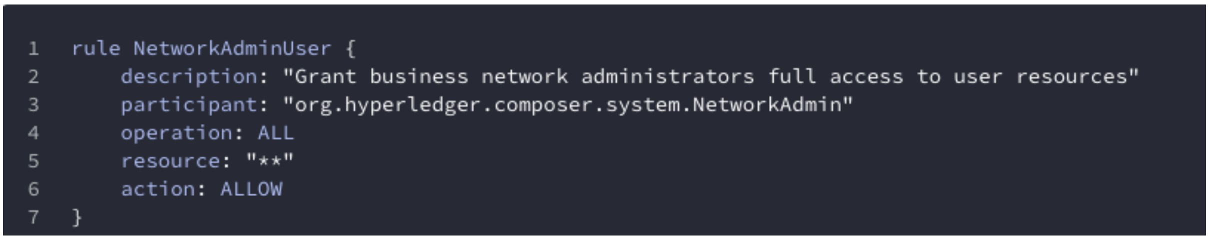
The last function is known as **receive**, which will be triggered while signing the package. At this moment, the package attributes like **receiveTime, receiveWeight** will be assignment, followed by a judgment sentence to make sure this package is the correct one. After the validation, the smart contract will calculate the shipping fee based on the latest parameters of the package. Some errors will come out if the package is wrong. Finally, if everything looks good, the corresponding amount will be deducted from the sender's wallet balance.



```
async function receivePackage(receive) {
    receive.package.receiveTime = receive.receiveTime;
    receive.package.receiveWeight = receive.receiveWeight;
    if (receive.package.sender  == receive.sender
        && receive.package.receiver   == receive.receiver
        && receive.package.weight == receive.receiveWeight)
    {
      receive.package.status = "Already Signed"
      let D = ((receive.package.receiveLongitude - receive.package.sendLongitude)**2
        + (receive.package.receiveLatitude - receive.package.sendLatitude)**2)**(0.5);
      let T = receive.receiveTime-receive.package.sendTime;
      receive.package.fee = D*(receive.package.weight)/T;
    }
    else {
        throw new Error('Wrong Package!')
    }
    receive.sender.walletBalance -= receive.package.fee;
    let assetRegistry = await getAssetRegistry('org.example.mynetwork.Package');
    await assetRegistry.update(receive.package);
    let participantRegistry = await getParticipantRegistry('org.example.mynetwork.Trader');
    await participantRegistry.update(receive.sender);
}
```

3.3.3 Access Control

Access control file is necessary for the Hyperledger blockchain network as well, for it defines the certification access towards the system. In this case, only one management staff has all the privilege to modify the instance in this channel, claimed as:



```
rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}
```

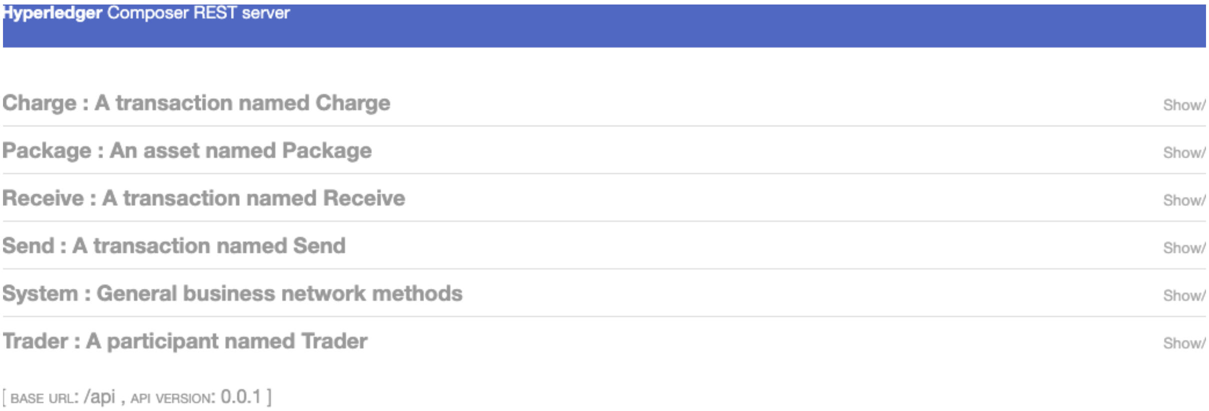
For the users of this platform, they have no rights to modify the asset attributes and delivery details, which is designed by the management staff of each company in a real business context.

3.4 Web Interface and Function Demonstration

After exporting the .bna file from Composer Playground, the Fabric based blockchain network can be upload and conducted on the secure shell (SSH) server. Then, to be more friendly, we give it a user interface based on Angular frame.

3.4.1 Representational State Transfer Interface (REST)

REST API is like a middle stage to record and display any operation and transaction in the RCE platform. In this case the URL is http://45.76.212.69:3001/.



Hyperledger Composer REST server

Charge : A transaction named Charge   Show/

Package : An asset named Package   Show/

Receive : A transaction named Receive   Show/

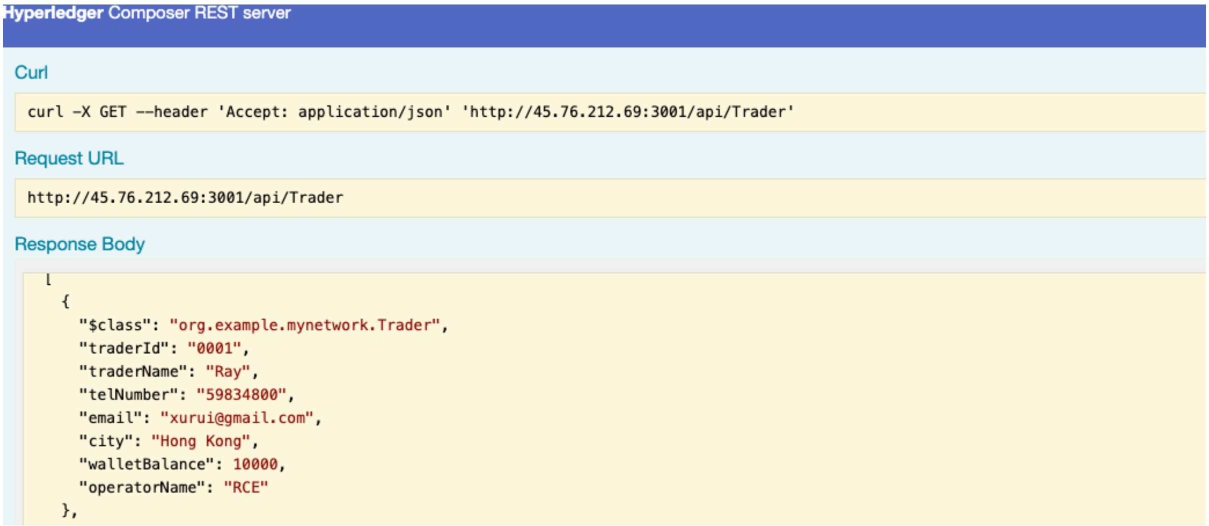
Send : A transaction named Send   Show/

System : General business network methods   Show/

Trader : A participant named Trader   Show/

[BASE URL: /api , API VERSION: 0.0.1]

Take the **Trader** for example, you can check the information of the all traders in the network and can even delete a certain account if necessary.



Hyperledger Composer REST server

Curl

```
curl -X GET --header 'Accept: application/json' 'http://45.76.212.69:3001/api/Trader'
```

Request URL

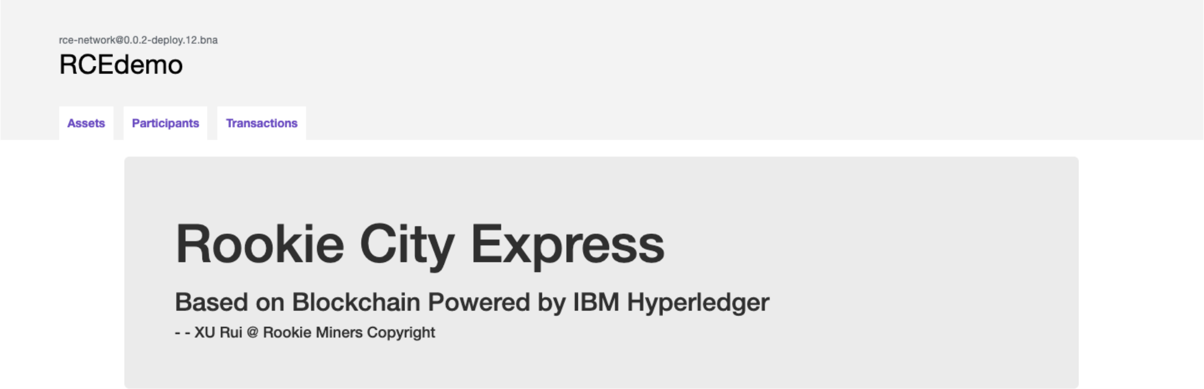
```
http://45.76.212.69:3001/api/Trader
```

Response Body

```
[
  {
    "$class": "org.example.mynetwork.Trader",
    "traderId": "0001",
    "traderName": "Ray",
    "telNumber": "59834800",
    "email": "xurui@gmail.com",
    "city": "Hong Kong",
    "walletBalance": 10000,
    "operatorName": "RCE"
  },
```

3.4.2 Website Application Interface

A website user interface is generated roughly in this stage, with the help of Angular, a platform for building mobile and desktop web applications. The URL is http://45.76.212.69:4201/.



rce-network@0.0.2-deploy.12.bna

RCEdemo

Rookie City Express

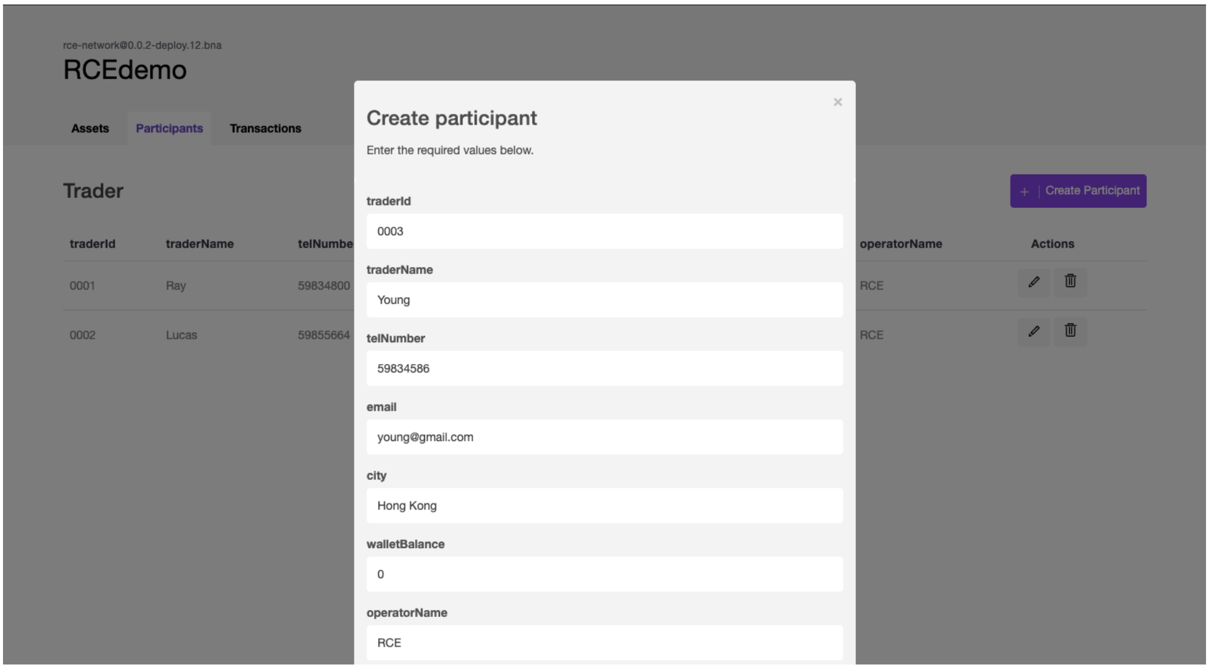
Based on Blockchain Powered by IBM Hyperledger

- - XU Rui @ Rookie Miners Copyright

3.4.3 Main Function Illustration

In this simple Angular website mentioned above, some basic functions can be demonstrated.

As a management staff of RCE, the **trader** as participants can be created at the begin. Just click “Trader” under the menu of “Participants” first and then find “Create Participant” button to declare a new **trader**, where some attributes need to be specified in the blanks.



traderId	traderName	telNumbe	operatorName	Actions
0001	Ray	59834800	RCE	
0002	Lucas	59855664	RCE	

Create participant

Enter the required values below.

traderId: 0003

traderName: Young

telNumber: 59834586

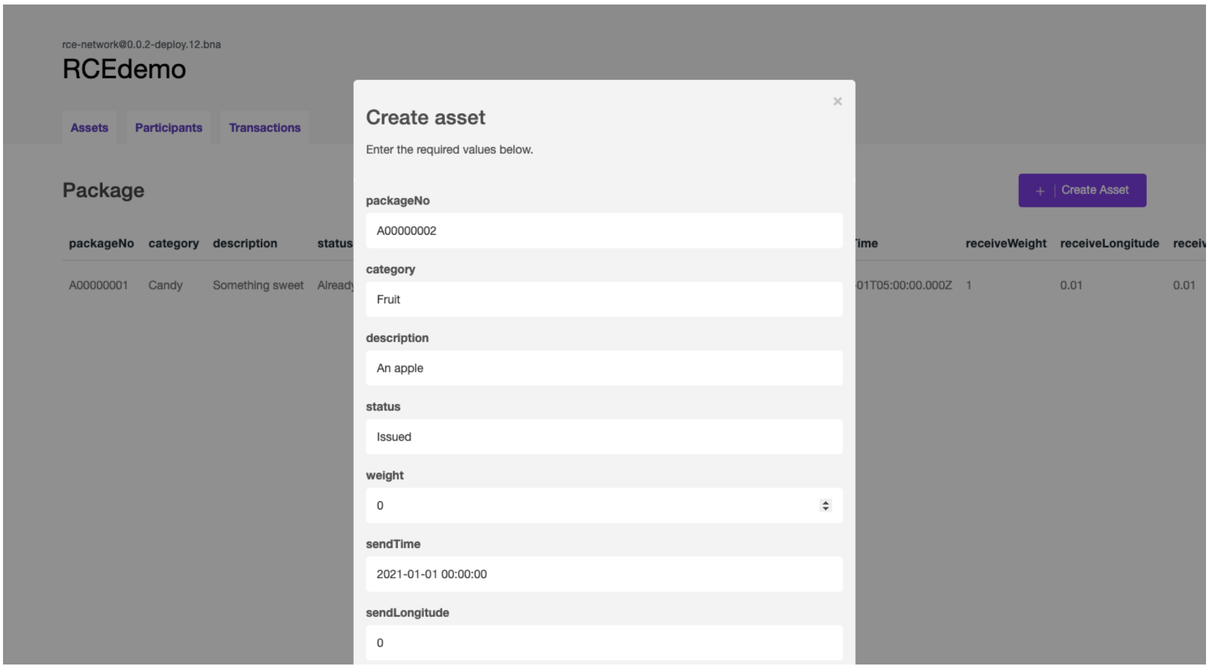
email: young@gmail.com

city: Hong Kong

walletBalance: 0

operatorName: RCE

And then, the **package** can be issued in the “Package” under the “assets” menu. Similar to a **trader**, all the attributes should be initialized in this pop-up window after a click on “Create Asset”, so that a line of package information will merge in this webpage.



packageNo	category	description	status	ime	receiveWeight	receiveLongitude	receiv
A00000001	Candy	Something sweet	Already	01T05:00:00.000Z	1	0.01	0.01

Create asset

Enter the required values below.

packageNo: A00000002

category: Fruit

description: An apple

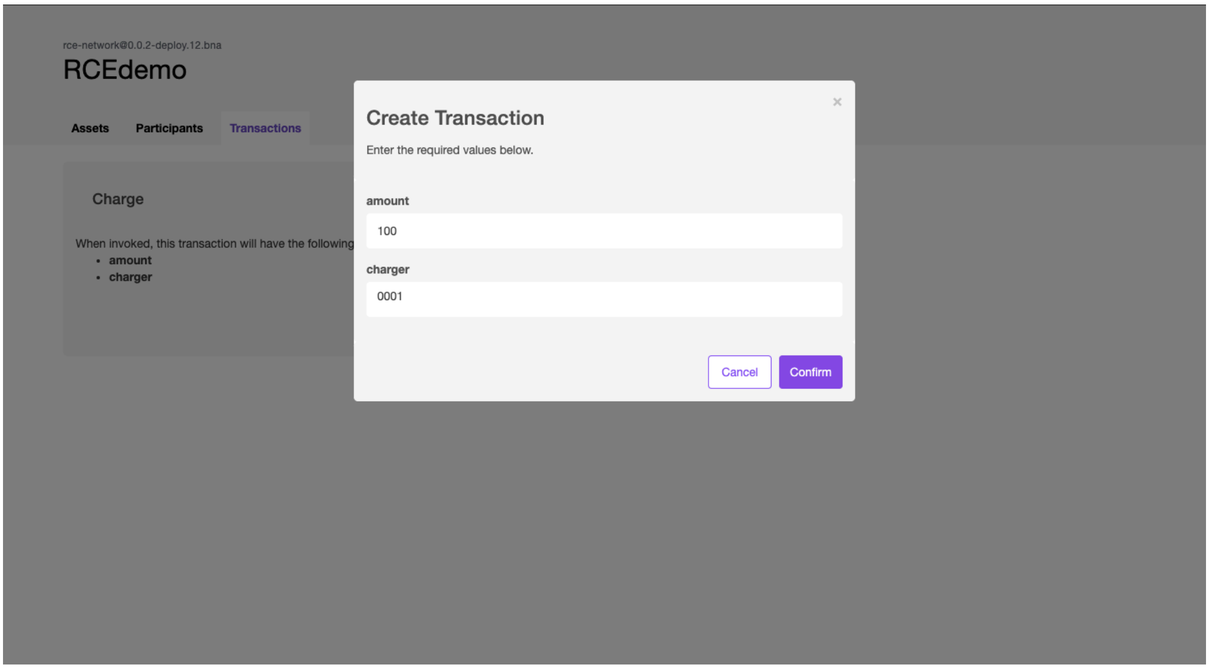
status: Issued

weight: 0

sendTime: 2021-01-01 00:00:00

sendLongitude: 0

Additionally, the three kinds of **transaction** can be found under the “Transaction” menu. The first one is **charge** that is to recharge the wallet balance for a **trader**. After this operation, a certain amount of token input in the blank will be accumulated into the trader’s account, which can be checked in the “Participants” page.



Charge

When invoked, this transaction will have the following

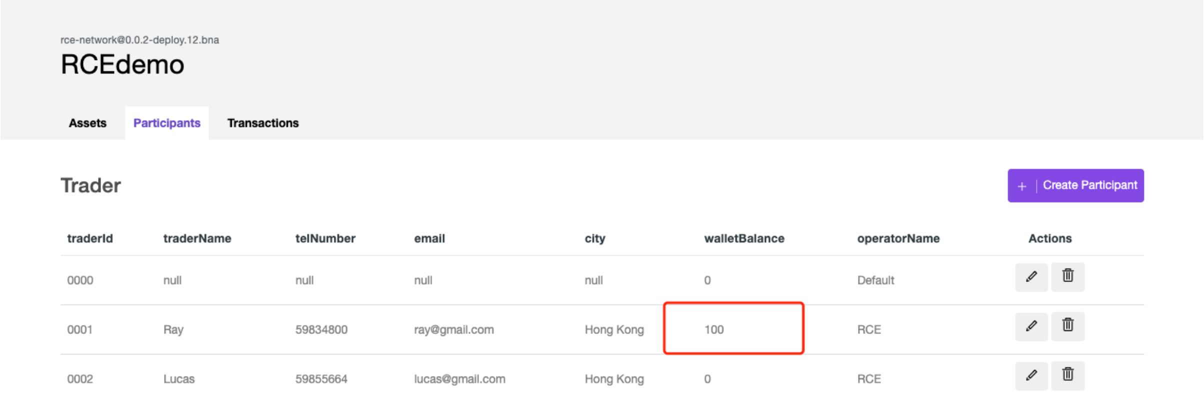
- amount
- charger

Create Transaction

Enter the required values below.

amount: 100

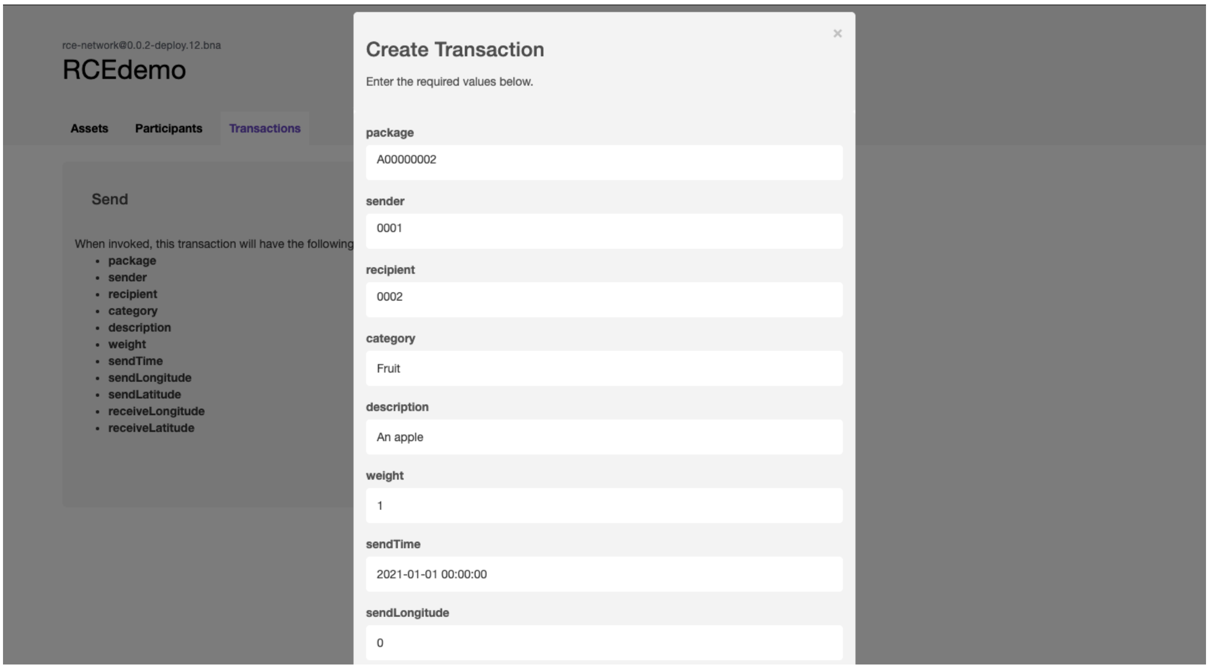
charger: 0001



Trader

traderId	traderName	telNumber	email	city	walletBalance	operatorName	Actions
0000	null	null	null	null	0	Default	✎ 🗑
0001	Ray	59834800	ray@gmail.com	Hong Kong	100	RCE	✎ 🗑
0002	Lucas	59855664	lucas@gmail.com	Hong Kong	0	RCE	✎ 🗑

Secondly, the **send** function allows a trader to send a package issued already. The attributes will be updated when invoke this transaction, including package information, delivery time, shipping and receiving address. Notice that only the sender with positive wallet balance can invoke this **send** function, based on common sense designed in chain code, otherwise, some errors will come out to remind trader to recharge the wallet balance. Meanwhile, if everything is good, the status of the package will be update to indicate that the package is sent out.



rce-network@0.0.2-deploy.12.bna

RCEdemo

Send

When invoked, this transaction will have the following

- package
- sender
- recipient
- category
- description
- weight
- sendTime
- sendLongitude
- sendLatitude
- receiveLongitude
- receiveLatitude

Create Transaction

Enter the required values below.

package: A00000002

sender: 0001

recipient: 0002

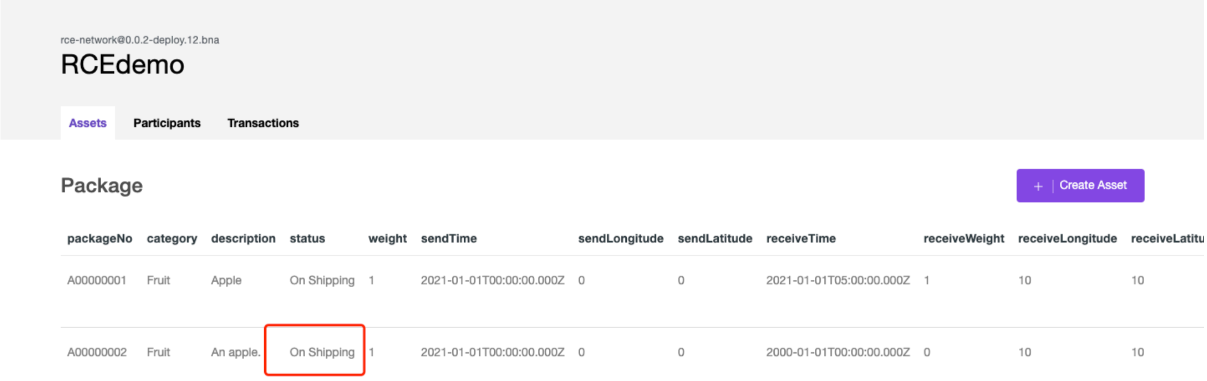
category: Fruit

description: An apple

weight: 1

sendTime: 2021-01-01 00:00:00

sendLongitude: 0



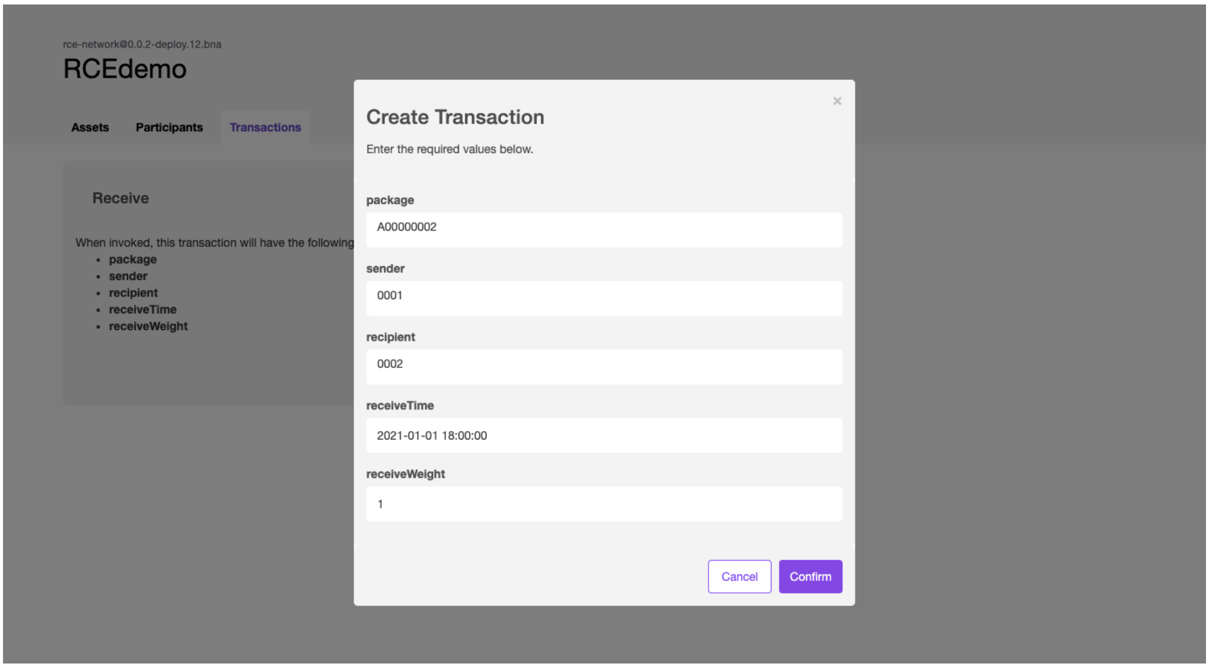
rce-network@0.0.2-deploy.12.bna

RCEdemo

Package

packageNo	category	description	status	weight	sendTime	sendLongitude	sendLatitude	receiveTime	receiveWeight	receiveLongitude	receiveLatitu
A00000001	Fruit	Apple	On Shipping	1	2021-01-01T00:00:00.000Z	0	0	2021-01-01T05:00:00.000Z	1	10	10
A00000002	Fruit	An apple.	On Shipping	1	2021-01-01T00:00:00.000Z	0	0	2000-01-01T00:00:00.000Z	0	10	10

The last step in this business process is to sign the package, which is done by **receive** function. Receiving time should be input in the input box to calculate the delivery fee based on the formula written in the smart contract. If all the input information about the package is exactly the same as it claimed when sending, such as the weight without any shrinking, which indicates a successful sign-in, then the amount of the shipping fee will be deducted from the senders account balance eventually as shown in the screenshot.



rce-network@0.0.2-deploy.12.bna

RCEdemo

Assets Participants Transactions

Receive

When invoked, this transaction will have the following

- package
- sender
- recipient
- receiveTime
- receiveWeight

Create Transaction

Enter the required values below.

package: A00000002

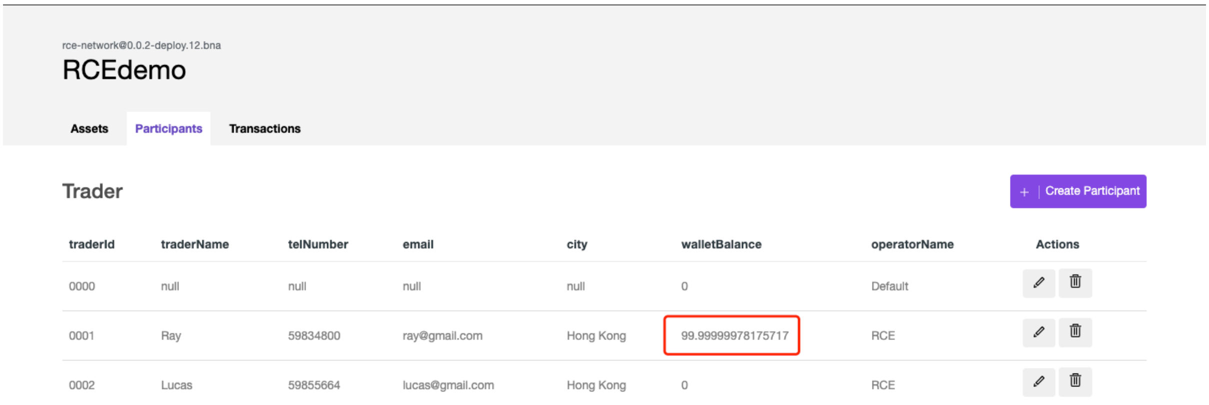
sender: 0001

recipient: 0002

receiveTime: 2021-01-01 18:00:00

receiveWeight: 1

Cancel Confirm



rce-network@0.0.2-deploy.12.bna

RCEdemo

Assets Participants Transactions

Trader

＋ | Create Participant

traderId	traderName	telNumber	email	city	walletBalance	operatorName	Actions
0000	null	null	null	null	0	Default	
0001	Ray	59834800	ray@gmail.com	Hong Kong	99.99999978175717	RCE	
0002	Lucas	59855664	lucas@gmail.com	Hong Kong	0	RCE	

4 Future Work and Possible Extensions

4.1 Assessment

Advantages of our design:

- In the data layer of blockchain, information is distributed among nodes, and no node can control the formation of account data. Only through the coordination among nodes, a consistent account can be generated. This means that our data is not easy to be changed, making the whole express transportation process more open and transparent.

- The blockchain data layer is a chain structure starting from the genesis block, including the block data, chain structure, random number, timestamp, public key, private key data, etc. Each block is associated with each other in the way of hash signature. If a transaction in the block changes, its hash value will also change. This makes the whole process of data security higher, customer privacy protection problem can be solved, to ensure the security of the transaction.
- Our database transport billing transaction process is fully automated, and in the user transaction to join the private key encryption, data update, avoid the express delivery error or loss.

Drawbacks of our design:

- Users need to buy crypto coins used in our platform before they use it. This may be some inconveniences in use.
- There may be some illegal transportations due to the protection of the privacy.

4.2 Future Extension and Plan

The framework we built is a prototype of our blockchain application. However, it can be improved from different aspects. More functions can be added into Recipient such as a brief information of delivery man for a service (Name of the delivery man, Rating of service, Contact No.). Therefore, recipient can directly contact the delivery man if there is any enquiry of a package.

Moreover, we can modify Platform to be more intelligent. For instance, if a sender or recipient entered a specific shipping or receiving address frequently, the system may save the address and ask the user whether set it to default address next time. As a result, it can decrease the chance that enter a wrong address.

Senders needed to charge their wallet by crypto coin by now. However, it is still not a common currency in public. It will cause our blockchain application may not reach more users generally. In the future, we would like to include incorporating banks into the blockchain network. Then, more users can charge their wallet by their default currency in bank. In the course of time, our application can be universalized in our society.

Generally speaking, it is a peer-to-peer express system based on blockchain network for packages. If we take a wider view of this application, it can be more powerful and meaningful. For insurance industry such as AIA and Manulife, if they have some cooperated project that need to share some internal data with adequate transparency. Our system can help with that. Because we can open the network to both company and all the staffs in these companies will be able to access and collect some public data.

At this stage, we would like to apply our blockchain system to city express. In long-term development, we hope this system can be apply to some fresh ingredient delivery service. Since some fresh ingredients especially organic ingredient without preservative are easily to be spoiled during the logistic. Also, the food security is a huge concern when delivery the ingredients. If we want to improve the process, we can use this blockchain system to record every stop of the delivery like our city express. By real time record of the shipment, we are able to figure the problem effortless since we can trace it back.

5 Potential Challenges and Impact

5.1 Normalize the Fee of the Delivery

In the platform, due to the smart contract can execute arbitrary computations on the blockchain, the cost of the package is calculated strictly following by the normalized regulation in the smart contract. And in the regulation, the price of the package is calculated by the real sending time and the real weight of the package, which avoid the phenomenon that the price is same of the package that arrives in five hours and the package arrives in three hours in the traditional city express. Based on the normalized calculating regulation, the sender can calculate the price by himself and avoid pay more many for latter package.

Moreover, all the users can learn the regulation of the price calculation, and the transparent rules make prices more reasonable.

5.2 Make the Speed of Delivery Faster

Though traditional express already provides different service to clients and charge for different fees according to what kind of speed the clients choose. It still doesn't have a positive effect on the speed because the express fee was settled down once the clients chose the service. That the package just needs to be delivered within the limited time should be enough. Someone may spend 10 for a traditional speed while someone maybe willing to spend extra 10 and require it to be delivered before the night of second day. It would not encourage the speed to be faster and faster, after clients pay for the fee, which is what our platform pursues. In our application, the express fee would never be decided only if the package is sighed. There is a timer to record the total time of the delivering, it corrects to hour even minute. The longer time this deliver spends, the lower fee it will charge.

5.3 Reduce the Delivering Mistakes

Nowadays, it is significantly common that the package is lost in the transportation process or signed by unexpected people. In our RCE platform, due to the IoT device with the package, the location information of the package can be transmitted to system in real time. Therefore, the sender can view the status of the package and once the package is lost, the system can find it back by the IoT device. As a result, RCE makes it possible to avoid the package's lost in the transportation process.

In another hands, only when the expected recipient sends the application of signing the package by using his own private key, he or she can sign the package after the RCE check the application by the public key. It is no doubt that the package will be signed only by the expected recipient.

In addition, the package can only be signed when the weight of the package is the same as the record of the sender made, which ensures there is no loss of package in the transportation.

5.4 Reduce the Risk of Modification

In the traditional city express service, the package can be signed by the courier before the recipient receive the package and the information like weight can be changed in the process. As a result, modification of package information and package detail and mistake records of package are prevalent in the traditional express industry, which leads to the cost of package becoming more expensive or cheap than fact. Since package information like weight, the receiving time is recorded automatically by the system in the RCE platform and the records of the blockchain is immutable, there will be less false information of the during the package sending process, thereby solving the problem that the platform and the user, including both the sender and the recipient, do not trust each other.

5.5 Make the Speed of Delivery Faster

Though traditional express already provides different service to clients and charge for different fees according to what kind of speed the clients choose. It still doesn't have a positive effect on the speed because the express fee was settled down once the clients chose the service. That the package just needs to be delivered within the limited time should be enough. Someone may spend 10 for a traditional speed while someone maybe willing to spend extra 10 and require it to be delivered before the night of second day. It would not encourage the speed to be faster and faster, after clients pay for the fee, which is what our platform pursues. In our application, the express fee would never be decided only if the package is sighed. There is a timer to record the total time of the delivering, it corrects to hour even minute. The longer time this deliver spends, the lower fee it will charge.

5.6 Promote the Development of Specific Industries

Since the trustless RCE platform can ensure the quality of package as well as some other improvement in the transportation of the multi-party organizations, the RCE can be used not only in the express delivery industry, but also in other industries that need to guarantee transportation time and quality to improve their service.

For example, for transportation in the agricultural industry, the most important thing is ensuring the products like fresh ingredients to be transported as the package in time. Using the RCE platform can encourage the delivery to pursue the speed to avoid the spoilage of ingredient during the delivery. Besides, the feature that orders can be traced can solve the problem of the food security to some extent.

Another aspect is to protect the package. For the main feature of blockchain, everything is determined once the order created, and package sent out. Especially for some industries like insurance or financial which always deliver important files. The transparency and traceable records will guarantee that no one can modify the information even change the packages. That right package sent to right person seems like a baseline target but also the most important one.

Based on the platform's ability to ensure quality and time in the delivery, the pain points of some industries have been solved. Therefore, the industries will get further development.

6 Statement

6.1 Acknowledgment

ZUO Zhiya, HU Wei

6.2 Contribution

Contribution of Each Member in the Group:			
Student ID	**Last Name**	**First Name**	**Contribution %**
56365524	XU	Rui	40%
56373801	HONG	Yuan	20%
56255901	HON	Yuen Ching	20%
56532945	WANG	Yuchun	20%
56509782	LU	Jie	0%
		Total	100%