

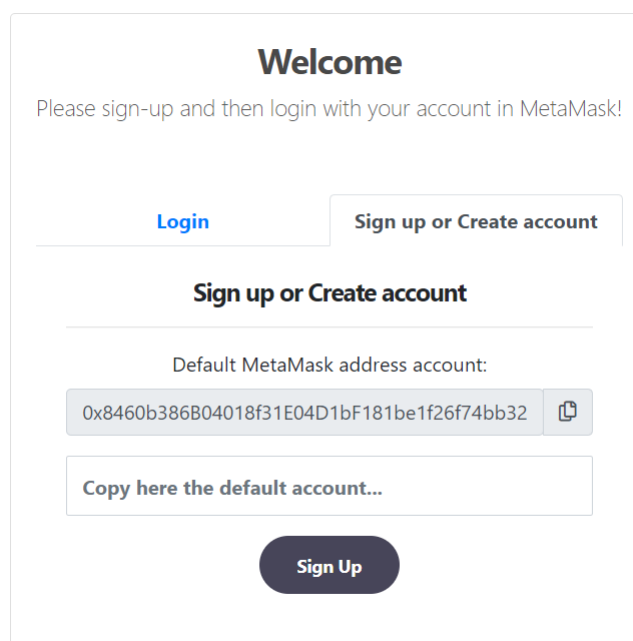
Deployment, Execution and Auditing of Bike Rental case study

Platform registration

The first page of the ChorChain tool handles the **Sign-up** and **Sign-in** of a user.

In particular, the **registration** requires an Ethereum address to be inserted. During this first step, the account currently logged in the **Metamask** plugin in the active browser is selected and inserted for the sign-up. This one is also suggested into a bar over the input field, together with the possibility to copy and paste it.

Through this procedure, the user can then **log into** the system, the only requirement is to insert an already registered account that must also match the one currently selected in Metamask.



The image shows a 'Welcome' screen for the ChorChain tool. At the top, it says 'Welcome' in bold, followed by the instruction 'Please sign-up and then login with your account in MetaMask!'. Below this, there are two buttons: 'Login' (in blue text) and 'Sign up or Create account' (in a grey box). The 'Sign up or Create account' button is selected, and below it, the text 'Sign up or Create account' is repeated. Underneath, it says 'Default MetaMask address account:' followed by a grey box containing the address '0x8460b386B04018f31E04D1bF181be1f26f74bb32' and a copy icon. Below the address box is a text input field with the placeholder text 'Copy here the default account...'. At the bottom, there is a dark blue button labeled 'Sign Up'.

Model Creation

After the user is correctly authenticated, it is possible to model the choreography diagram directly into the ChorChain modeller.

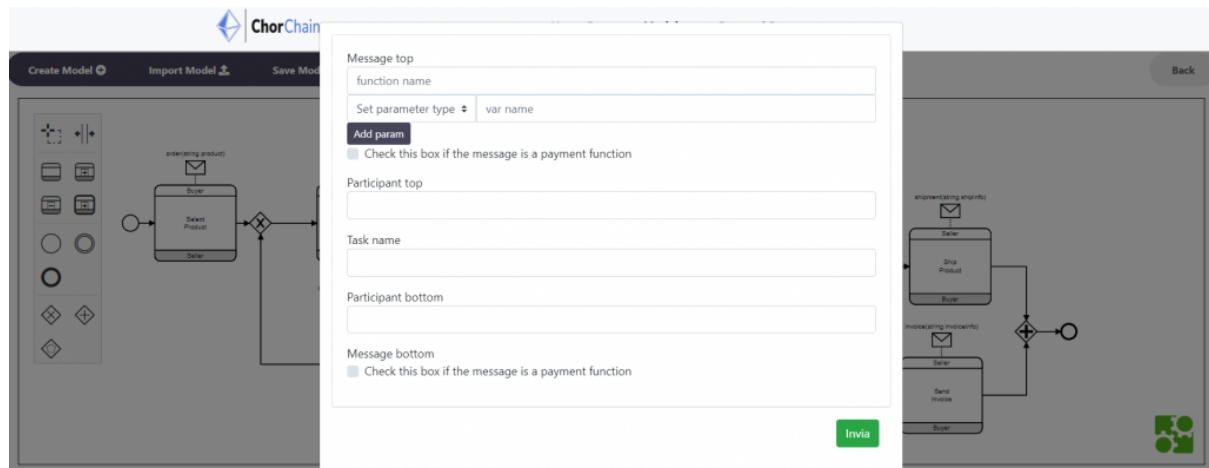
The modeller works like most of the others BPMN tools, in addition, ChorChain allows the insertion of information through the use of an intuitive panel, double-clicking the wanted task.

The panel makes easier the process of filling the model in a translator-suitable way. Indeed, each message must be written in the structure *function_name(input_type input_name)*, where:

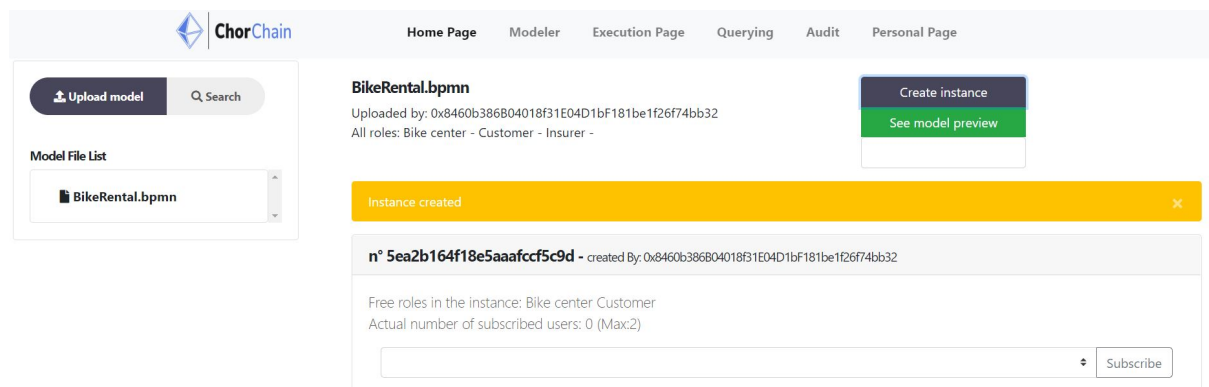
- *Function_name* represents the message name
- *Input_type* is the type of exchanged information through the message
- *Input_name* is the name of the information

In case the chosen message is a payment function, so it sends ether from one participant to another, it's just necessary to check the box and automatically the modeller generates it.

The other forms are used to fill in the participants and the name of the task. Once created, it's also possible to upload the model directly into the system repository with the **Save Model** option.



At this point, the Bike Rental choreography is publicly available on the ChorChain home page which displays all the available models on the left side of the page. Clicking on it, the instance list will appear, displaying more details, such as the creator address, the instance ID, the available roles in the instance and the actual number of subscribed users. At this point, two main operations are available: **Create an instance** and **Subscribe**.



Instance Creation

This function will show a panel that consents to make two operations. The first allows the user to select the choreography roles that will become optional while the second makes the instance visible just for the inserted addresses. Once compiled (or not) the instance will be created and will be visible on the home page.

Create new instance

Select the optional roles

The roles not selected are considered **MANDATORY**

☐ Participant 1

☐ Participant 2

Set visibility

Entering the addresses, the instance will become private and will be visible only to those addresses.

1.

Subscription

If there is a vacant role in the instance the user can cover it, taking part in the choreography. When the maximum number of mandatory participants is reached (so every role is covered) the deployment will automatically start, translating the model to a smart contract executed in the Blockchain. If a role is still vacant (so it's optional) the user can cover it calling a dynamic subscription through a smart contract function, made visible in the interface.

n° 5e2c61328df71510fb204d14 - created By: 0x0DBb0BaB522F880e7FfA5e227241f5ae8571B37E

Free roles in the instance: Seller

Actual number of subscribed users: 1 (Max:2)

Seller

Execution

Each user can access the execution page that contains all the active processes on the Blockchain visible on the left side of the page.

Clicking on one of them the relative model will appear, in which the active message is highlighted in green. In the centre of the page, the enabled message generates the function form, however, only the participant with the associated role can trigger it. Indeed, if the user doesn't cover the role actually involved he can't see the input form. In the opposite case, a Metamask transaction will be generated, waiting for confirmation from the user. To see the previously exchanged information it is necessary to double-click a message and the right-side panel will show the desired data.

ChorChain

Home Page Modeler Execution Page Querying Audit Personal Page

Model File List

BikeRental.bpmn 5e9ad1428df7157a5976d72a

MetaMask Notification

Account 1 → 0x364b...63...

INTERAZIONE CONTRATTO

DETAILS DATA

GAS FEE 0.0002 Tasso di conversione non disponibile

AMOUNT + GAS FEE 0.0002 Tasso di conversione non disponibile

TOTAL 0.0002 Tasso di conversione non disponibile

Annulla Conferma

BikeRental.bpmn

Contract deployed: 0x364be381db2692c91fee4b5f7c4688cf0596371 visible at: etherscan.io

insuranceRequest(bool insuranceReq) Customer Status: Enabled

false

Submit

request(string bikeType)
string bikeType : city bike

When a model is deployed it is accessible also in the auditing pages, Audit, Personal and Querying.

Audit Page

Here, all the instances deployed in the blockchain via ChorChain are stored, in particular, they are grouped by model in a left-side list. clicking one of the models, the BPMN diagram will appear in the centre, while in the right-side list the deployed instances are shown (green tick for completed ones). It is also possible to select a role of the model to see all the participants that covered it during the different executions. Additional information is displayed, like the instances execution time, the gas used and fees. Selecting an instance, all its related data is shown: the related blockchain contract address, gas and fees (related only to it), the list of participants involved with their role, all the transactions generated and the messages exchanged.

ChorChain

Home Page Modeler Execution Page Querying Audit Personal Page

Model File List

BikeRental.bpmn

BikeRental.bpmn

BikeRental.bpmn

Uploaded by: 0x8460b386804018f31E04D1bF181be1f26f74bb32

5e946814f18e5a4bd8fd628a Transactions Messages

created By: 0x8460b386804018f31E04D1bF181be1f26f74bb32
contract: 0x981eab61760ffe507aad401d1fc8a3d423cf10b

Total gas used: 8035709
Total fee: 0.0900319970 Ether

Role	User	Gas used	Fee (Ether)
Bike center (mandatory)	0x8460b386804018f31E04D1bF181be1f26f74bb32	605621 (7.536621846311259%)	0.0006056210 (0.6726730719968369%)

Model roles

Bike center Customer

Model instances

Instances completed: 6/8 (75%)

Execution time

Max. 2610s Min. 615s Av. 1237.5s

Total gas used

Max. 8544691 Min. 8035709 Av. 8144377.16666667

Total fee (Ether)

Max. 0.1000609910 Min. 0.0900319970 Av. 0.091727338

5e946814f18e5a4bd8fd628a

5e95bcf46b3b6e7b806bf949

5e95c3e56b3b6e7b806c02ad

The **transactions** panel shows all the transactions generated by the contract from a blockchain point of view. Indeed it is possible to find the hash of the transaction, its sender,

the timestamp, the value exchanged (if present) and the message executed with its parameters.

0x92ff902c0f317d091e6f0298216a3fe082be612478b1a94242ef46f6576ea814

GasUsed: 83895

Fee: 0.0000838950 Ether

From: 0x26de9861972308303c955fa9ee4bbb63a5a413db

To: 0x9be6cf1adf70d6db095a36c5fd04af6e757f9ba1

Timestamp: 30/4/2020 17:15:27

Message: Message_02ckm6k(bikeType: city)

The **Message** panel is instead oriented to show all the messages of the model and their related transaction hashes. These also the participant that executed it and the number of times that it was sent (if there is a cycle).


MESSAGES

request(string bikeType) [Sent 1 times]

Transactions	Users
0x92ff902c0f317d091e6f0298216a3fe082be612478b1a94242ef46f6576ea814 Instance: 5eaaeb368df7157a597641e4	0x26de9861972308303c955fa9ee4bbb63a5a413db

Personal Page

This page is similar to the previous, however, it is focused only on the processes in which the logged user is involved. Indeed, all the information displayed is related to the instances below that don't correspond to all the instances of the BikeRental model stored in the repository.

ChorChain

Home PageModelerRunning ContractsQueryingAuditPersonal Page

Your models

BikeRental.bpmn

BikeRental.bpmn

Model instances: 5
Completed instances: 5 (100%)

Your roles

Bike center (Mandatory)

Fee (Ether)

Max. 0.0896578890Min. 0.0896384370Av. 0.0896424402

Execution time

Max. 2400sMin. 885sAv. 1336s

Gas used

Max. 8120565Min. 8101113Av. 8105116.2

Instances

✓5e860593f18e5ac9e0338ae0

Execution time: 885s
Total gas used: 8120565
Total fee: 0.0896578890 Ether

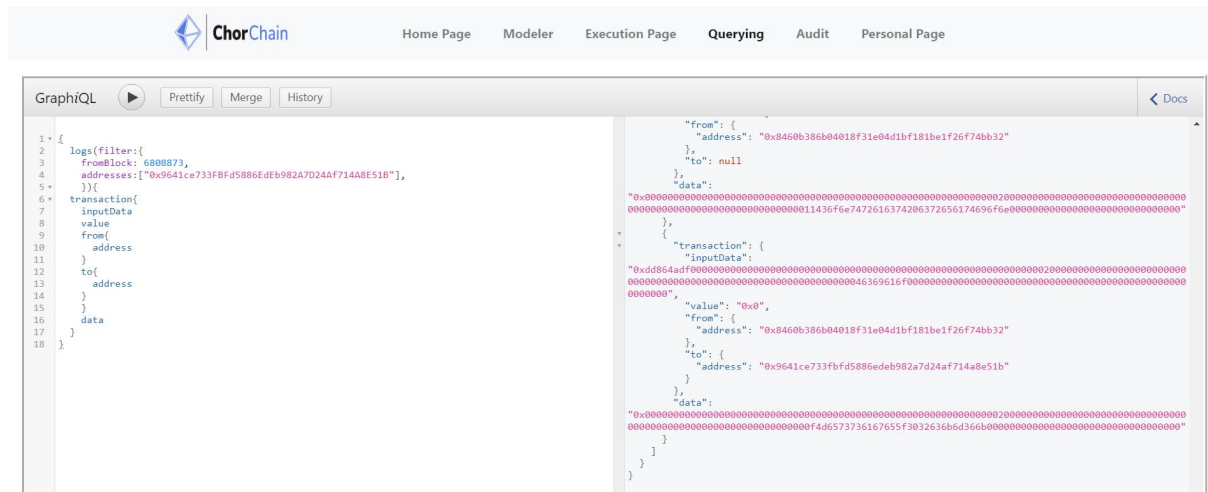
Your roles
Bike center (Mandatory)

✓5e86f400abe6b807003017de

The main purpose of this personal page is to give a clear view to the user of its processes, allowing him to analyse data that can help him to find possible issues or possible upgrades.

Querying Page

The last page allows the user to perform personalised queries oriented to explore the entire blockchain. Indeed it contains the GraphQL interface connected to an Ethereum node.



This functionality is not limited to the ChorChain models, indeed its expressiveness can be used to retrieve information regarding blocks, transactions and logs of the entire network. To notice, the use of this section requires a good level of knowledge by the user.