# ChorChain: A Blockchain-Based Framework for Executing and Auditing BPMN Choreographies

Flavio **Corradini**¹, Alessandro **Marcelletti**¹,*,†, Andrea **Morichetta**¹,†, Andrea **Polini**¹, Barbara **Re**¹ and Francesco **Tiezzi**²

¹*University of Camerino, Camerino, Italy*

²*University of Florence, Florence, Italy*

### Abstract

Adopting the BPMN standard and, in particular, its choreography diagrams is promising to increase business possibilities. Their adoption in practice has been challenging and faced complex hurdles due to the lack of concrete support for the choreography execution, and the absence of distributed infrastructures allowing the involved participants to trust each other. To overcome these challenges, we propose ChorChain, a novel model-driven framework based on Blockchain technology. It supports the whole lifecycle of choreographies, from their modelling to their distributed execution and auditing. The feasibility and effectiveness of the framework are assessed through experiments conducted on the Rinkeby Ethereum Testnet.

### Keywords

ChorChain, BPMN, Choreography, Blockchain

## 1. Introduction

Nowadays, a prominent modelling language to describe distributed collaborative systems is the BPMN standard [1]. BPMN provides choreography diagrams, a flow-chart based notation that permits to describe system interactions in terms of the exchange of messages from a global perspective, without exposing the internal behaviour of the participants. Despite the wide acceptance of BPMN from both industry and academia, in practice until now the usage of such diagrams has been mainly confined to the design phase. Indeed, there is a lack of concrete support for the other phases, in particular in relation to choreography execution. In addition, the full adoption of choreography specifications for the engineering of distributed systems has been hindered by the difficulty of ensuring trust among the distributed participants [2], which indeed will engage in cooperation only if they trust each other. We aim to address the above issues by leveraging blockchain technology, as envisioned in [3]. Blockchain is an emerging technology for decentralised and transactional data sharing across a network of untrusted participants. It guarantees the integrity and immutability of data without relying on a central authority or any particular participant. Thus, blockchain enables the development of new forms of distributed systems, where all participants have a clear view of the ongoing

system execution and can have tangible proof of the actions performed by the counterpart [4]. In particular, in our choreography-based setting, blockchain technology is exploited to allow choreography participants to achieve trust without a central authority, by enabling the auditing of the choreography execution in a non-repudiable manner. In this way, for example, it can be checked that a given participant has actually sent/received a given message as specified in the choreography. The main challenges in relation to the development of choreography-based distributed systems on top of the blockchain are: *(i)* the need to fill the gap between the high-level description given by the choreography, and the low-level one given by the code deployed and executed on the blockchain, without annoying the end-user with convoluted technicalities; and *(ii)* to support the whole life-cycle of choreographies.

We face these challenges by providing a model-driven framework, called ChorChain, for the development of trustable choreography-based systems. ChorChain exploits blockchain technology, in particular Ethereum, as the base for executing BPMN choreographies. The framework supports the full lifecycle of choreographies, from their modelling to their publishing and instantiation, until their deployment and execution in the Ethereum blockchain. In particular, ChorChain supports the automatic generation of smart contracts (i.e., programs to be run over the blockchain), which is completely transparent to the final user, and the controlled execution of the choreography, which allows only the sequence of actions prescribed by the specification. The smart contract executed in the blockchain, indeed, is specifically generated to implement the choreography workflow, thus forcing the correct behaviour of each participant. Moreover, the smart contract generated by ChorChain can manage payments in Ether specified in the choreography model. In addition, the guarantees on data stored in the blockchain allow the auditing of exchanged messages during the choreography execution, without the need for a monitoring infrastructure.

## 2. ChorChain Innovations and Main Features

ChorChain acts as a Business Process Management System for tamper-proof choreographies execution, differentiating from approaches available in the literature with similar objectives (e.g., those introduced in [5, 6]). In particular, differences relate to the modelling abstraction adopted (i.e., choreographies, neither processes nor collaborations), the usability of the resulting tool, and the comprehensive functionalities covering the whole choreography lifecycle.

ChorChain is accessible via web browser and the main phases with their features are synthesised below. When entering the tool, it is necessary to register and login. To ease access to the platform, the user can exploit the Metamask browser plugin (https://metamask.io/), which provides a web interface for managing Ethereum accounts. The account selected in Metamask then constitutes the identifier of the participant during the ChorChain usage.

***Modelling.*** The modelling phase is the starting point of the choreography lifecycle. To support it, ChorChain integrates the modelling environment chor-js (https://github.com/bptlab/chor-js) directly in the front end, which is based on AngularJS. The modelling area offers several functionalities, such as the creation, the import, the export and the saving of a model in the ChorChain repository. To derive a blockchain-based infrastructure, supporting a choreography-based system at run-time, ChorChain provides a panel where the user can specify a list of parameters for the exchanged messages and data; this information is included in the *.bpmn* file
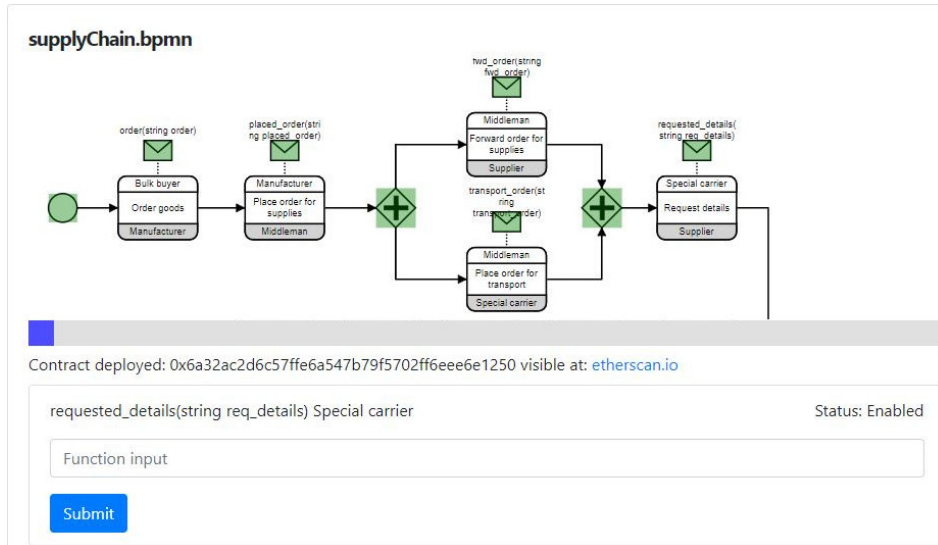
**Figure 1:** ChorChain execution page

and is completely portable and compatible with other modellers.

***Publishing, Instantiation and Subscription.*** After the model is saved inside the ChorChain repository, it is accessible in a dedicated page via an intuitive user interface. Here the user can generate a new instance for a specific model generated by him/her or by other users. During the instantiation, it is possible to select optional roles and limit the visibility to selected parties. Once created, the choreography instance will be kept in a "suspended" state while waiting for all mandatory roles to subscribe. The subscription is done by selecting the role to cover in the choreography. This will automatically associate the user's Ethereum account to the role identifying him/her also in the smart contract. When a choreography instance has no more vacant mandatory roles, ChorChain considers the partnership complete and starts the generation of the Solidity smart contract. After the generation, the contract is deployed automatically on the blockchain.

***Execution.*** In order to enable the participants interaction, ChorChain provides an execution page accessible to each participant (see Fig. 1). The execution page shows the current state of the execution and the active actions with the respective forms that are dynamically constructed for executing them. Notably, the execution form is visible only to the participant in charge of performing such activity. ChorChain also provides the automatic execution via REST API to permit the interactions by external services.

***Auditing.*** The proposed auditing strategy provided by ChorChain is choreography-centric, as it relies on the retrieval of information related to a choreography model and in relation to the execution of possible instances (see Fig. 2). The objective is to enable the assessment of constraints related to data exchanged by the participants in a choreography instance, as well as time and gas-related aspects with respect to a choreography instance execution. The combined usage of enforcement and auditing mechanisms can highly increase accountability and trust in a multi-organisational interaction context. ChorChain provides three main auditing pages: *(i)* general auditing, *(ii)* personal auditing and *(iii)* querying auditing. In the first case, it is
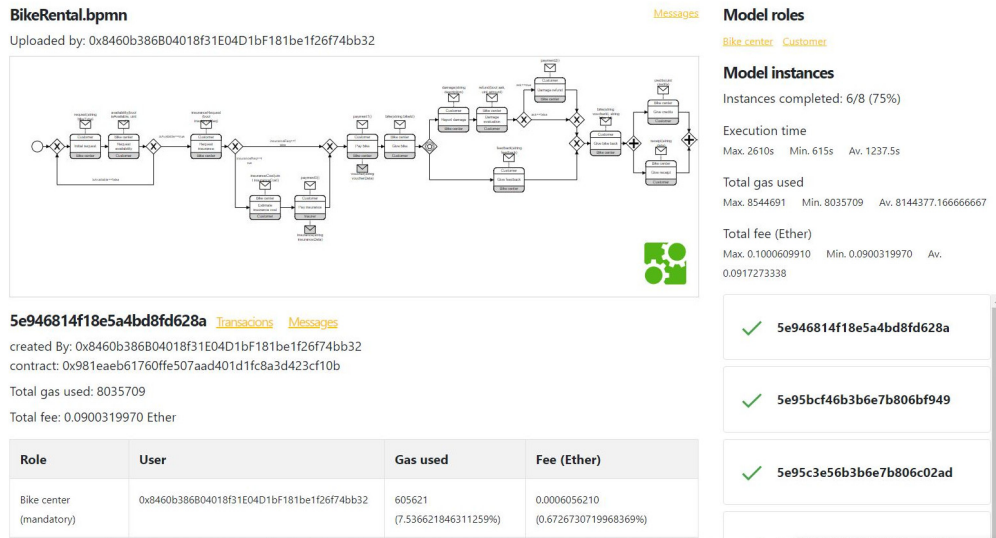
**Figure 2:** ChorChain auditing page

possible to retrieve information for all the choreographies deployed in the blockchain, in the second the system will show a focus just on the instances in which the user requesting the auditing is involved, and the last one is for expert users and permits the definition and execution of custom queries based on the GraphQL language.

## 3. Evaluation of ChorChain

The ChorChain tool was evaluated and reviewed under different aspects. In particular, we tested ChorChain using different approaches and case studies based on real worlds scenarios. For each case study, the related choreography was executed using the ChorChain tool in the Ethereum Rinkeby testnet, generating a complete track of the performed operations. We used a total of 4 different case studies that we synthesise below with their description, the links to their execution inside the blockchain and the pointers to additional information.

**Online Purchase**[1]: it describes the process of acquiring goods between a buyer and a seller. The process handles the offer for a good, the payment and the final shipment to the buyer [3].

**Bike Rental**[2]: it describes a rental process involving a customer, requesting a bike, a bike center, offering different rent possibilities, and an optional insurer providing insurance policies in conjunction with the bike rental [7].

**Supply Chain**[3]: it describes the message exchange of a supply chain. Specifically, a bulk buyer starts by placing an order with the manufacturer, who orders the materials from a middleman. The latter then forwards the request to a supplier and arranges also the transportation by means of a special carrier [7].

**Incident Management**[4]: it describes the processing of a help request by a customer made to a

---

[1]https://rinkeby.etherscan.io/address/0x251d5c389f9ab2b5ac705fd26b71ecfb167e5ed6
[2]https://bit.ly/Shortest_Path and https://bit.ly/Longest_Path
[3]https://bit.ly/Sup_Chain
[4]https://bit.ly/IncidentMan

software manufacturer. The key account manager tries to handle the request and, in the negative case, forwards this issue to different support agents, until to reach the software developer for receiving a solution [7].

ChorChain was evaluated by using 30 synthetically generated choreographies. The experiments were conducted on models to measure the effectiveness of the approach by isolating and executing all the BPMN choreography elements supported. Specifically, each model was tested with a growing number of elements in order to capture the incremental trend of costs [7]. In such experiments, we observed that each new message added to the model costs on average around 123,000 units of gas in the deployment. Each pair of gateways, consisting of a split and a join, impacts around 108,000 units of gas and the transactions range from a minimum of 201,000 to a maximum of 582,000 units of gas.

Finally, ChorChain usability was assessed by experiments involving students at the University of Camerino at the 1st year of the MSc in Computer Science. All of the 12 participants have got a BSc degree in Computer Science and have taken two semester courses at master level about business process modelling and enterprise software infrastructure. We provided a scenario referring to an Internship process flow in which a company, the university office and a student interact in order to publish, select, assign and complete a given internship. ChorChain obtained quite good results in the integration and usage; the interested reader can see the details in [7].

## 4. Screencast and Website

The ChorChain tool, as well as its source code, examples and tutorial document are available at http://pros.unicam.it/chorchain/, while a screencast is available at https://youtu.be/_RV7r9xbx9w. In particular, the screencast shows a typical scenario where the user models, deploys, executes and audits a bike rental scenario. ChorChain can be redistributed and/or modified under the terms of the MIT License.

## References

[1] F. Corradini, A. Morichetta, A. Polini, B. Re, F. Tiezzi, Collaboration vs. choreography conformance in BPMN 2.0: From theory to practice, in: EDOC, IEEE, 2018, pp. 95–104.

[2] A. Gemino, Y. Wand, Complexity and clarity in conceptual modeling: comparison of mandatory and optional properties, Data & Knowledge Engineering 55 (2005) 301–326.

[3] F. Corradini, A. Marcelletti, A. Morichetta, A. Polini, B. Re, F. Tiezzi, Engineering trustable choreography-based systems using blockchain, in: SAC, ACM, 2020, pp. 1470–1479.

[4] J. Mendling, I. Weber, et al., Blockchains for business process management - challenges and opportunities, ACM Transactions on Management Information Systems 9 (2018) 1–16.

[5] B. Carminati, C. Rondanini, E. Ferrari, Confidential business process execution on blockchain, in: Web Services, IEEE, 2018, pp. 58–65.

[6] L. García-Bañuelos, A. Ponomarev, M. Dumas, I. Weber, Optimized execution of business processes on blockchain, in: Business Process Management, volume 10445 of *LNCS*, Springer, 2017, pp. 130–146.

[7] F. Corradini, A. Marcelletti, A. Morichetta, A. Polini, B. Re, F. Tiezzi, Engineering trustable and auditable choreography-based systems using blockchain, ACM Transactions on Management Information Systems 13 (2022).