# RESTChain: a Blockchain-based Mediator for REST Interactions in Service Choreographies

Francesco Donini
Institute of Informatics and Telematics (IIT)
National Research Council (CNR), Pisa, Italy
francesco.donini@iit.cnr.it

Alessandro Marcelletti
University of Camerino
Camerino, Italy
alessand.marcelletti@unicam.it

Andrea Morichetta
University of Camerino
Camerino, Italy
andrea.morichetta@unicam.it

Andrea Polini
University of Camerino
Camerino, Italy
andrea.polini@unicam.it

## ABSTRACT

In inter-organizational contexts, different organizations cooperate exchanging information, to reach specific and shared objectives. The achievement of such interactions raises the need for a trusted communication environment to be used by the participants. This is a particularly relevant challenge when such interactions are specified in a peer-to-peer style, as in the case of Service Choreographies. Indeed, in such situations, the involved participants expect that all the interactions are performed abiding by the agreed specification. To support such a scenario, blockchain technology is gaining interest thanks to its security, trust, and decentralization characteristics. However, technological barriers still limit its adoption in real context due to the costly and time-consuming learning process. For this reason, we propose RESTChain, a general framework relying on blockchain technology enabling in an automatic way the interactions that take place among the participants in a service choreography. Starting from a choreography specification, the framework automatically derives a set of Mediators and a Smart Contract that coordinates the service interactions. In this way, each organization can communicate with the other services through the blockchain in a secure, auditable, and transparent manner.

## CCS CONCEPTS

• **Computer systems organization** → **Distributed architectures**; • **Information systems** → **RESTful web services**;

## KEYWORDS

REST services, service choreography, blockchain technology, multi-parties applications, trust management

**ACM Reference Format:**
Francesco Donini, Alessandro Marcelletti, Andrea Morichetta, and Andrea Polini. 2023. RESTChain: a Blockchain-based Mediator for REST Interactions in Service Choreographies. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27-March 31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3555776.3577826

## 1 INTRODUCTION

The integration of software systems in an inter-organizational context has been for a long time both a technical and methodological challenge. Nowadays, the emergence of technologies and protocols in the Service-Oriented Computing (SOC) realm, has made it rather easy, using for instance REST based frameworks, to connect different services and let them interact, even when they belong and are managed by different organizations. However, when considering inter-organizational contexts, two different aspects emerged as particularly relevant and challenging.

The first one refers to the specification of the distributed coordination among the services involved in an inter-organizational application. The second one instead asks for the definition of strategies and mechanisms that permit to increase trust among the various participants.

With respect to the first challenge, approaches considering the **choreography** specification emerged. A choreography mainly constitutes a specification that considers all the participants as peer entities that can independently interact to reach a specific common goal. The coordination logic is fully distributed among the participants, and each of them has to behave accordingly to the agreed specification without the need for a central authority. In this case, a lack of trust could arise due to the unknown contexts where the organizations collaborate or due to possible malicious behaviors diverging from the initial specification. These can lead to "your word against mine" cases which are particularly hard to handle, especially in a distributed context without a "regulator" entity.

To this aim, SOC applications can certainly take advantage of **blockchain** technology despite this is still largely unexplored, and it brings several challenges for organizations that have to spend resources, in terms of time and money, in forming new skills. Passing from an abstract specification, as it is a service choreography, to a blockchain infrastructure is indeed not trivial, and it can lead to security and technical issues.

For these reasons, we propose RESTChain, an approach that foresees the automatic generation of both a blockchain infrastructure and a set of service mediators. In particular, the blockchain coordinates and enforces the interactions among the different services specified within a choreography specification, while mediators

*SAC '23, March 27-March 31, 2023, Tallinn, Estonia*

ACM ISBN 978-1-4503-9517-5/23/03.
https://doi.org/10.1145/3555776.3577826

are distributed software components that enable the run-time interaction with the blockchain, exposing REST APIs embedding a behavior that by construction is compliant with the role played by the organization within the specified choreography. In this paper, we rely on a public permissionless environment, and, in particular, we consider Ethereum-like blockchain technologies as possible targeted implementations. In general, there is a wide plethora of blockchain technologies that come with different characteristics, and that could be selected according to specific needs. In this case, the characteristics of a public and permissionless infrastructure make it a natural choice for our purposes.

The paper is organized as follows. Section 2 provides a general description of the RESTChain approach, while Section 3 gives more details about the RESTChain architecture and the motivational choices. Section 4 discusses previous works found in literature, and finally, Section 5 summarizes the paper and reflects upon some opportunities for future work.

## 2 THE RESTCHAIN APPROACH

The main objective of RESTChain is to provide a distributed and trusted communication infrastructure among the possible participants in a service choreography. Moreover, RESTChain permits service developers to use the generated infrastructure without the need to acquire knowledge regarding blockchain technology. To reach such objectives, RESTChain supports an approach organized over a set of steps as exemplified in Figure 1.

(1) **Service choreography design**: the first phase of the approach foresees the derivation of a choreography specification that defines the expected run-time interactions among the various services. This step can be performed using any modeling environment supporting the creation of BPMN choreography diagrams.
(2) **Choreography instance derivation**: this step permits one or more organizations, willing to collaborate, to select a suitable choreography, and then to derive an instance of it for successive enactment and execution. Such an instance is initially set in an inactive state and RESTChain provides a reference to it allowing other organizations to join.
(3) **Participants registration**: in this phase, each interested organization can subscribe to an existing choreography instance acting as the provider of a service and playing a role that has not been covered, yet. The subscription enables the participating organization to download its RESTChain mediator. This is a software component that allows the organization service to enter the choreography execution when the instance is activated. It provides an interface for communications with other services passing through the blockchain in a transparent and secure manner. The mediator has to be maintained by the organization within its premises. The code generated by RESTChain for the mediator is available for being scrutinized by the organization. This is required in order to not reduce the trust, at the same time, the availability of pre-configured software makes the setting of the environment easier.
(4) **Trusted communication infrastructure generation and deployment**: this step is performed once all the roles in a choreography instance have been subscribed. At this point, RESTChain automatically generates a smart contract enabling the various choreography participants to immutably store the traces of their interactions. The generated contract keeps the status of the choreography and does not propagate those messages that are not in line with the specification, thus, enforcing the correct execution flow. In case of a deviation, the smart contract will return an error to the invoking partner specifying the non-conformance to the choreography specification.
(5) **Enactment and Execution**: the final step of the approach relates to the run-time support for the enactment and execution of the choreography. During this phase, the RESTChain mediator of each involved organization is directly connected with the previously generated smart contract, which stores and delivers the various messages to the proper recipients as specified in the choreography. The smart contract ensures and enforces the invocation of the correct service, while the execution is performed directly by the enabled mediator. Indeed, the mediators continuously listen to the events happening in the blockchain until a specific action is requested. The designed mediator will then handle the request, executing the service and storing the resulting payload.
(6) **Auditing**: after the execution phase, it is possible to check if the interactions among the participants abide by the initial specification. This is possible by exploiting the transparency of the blockchain that allows retrieving all the past interactions that took place in a specific choreography. To concretely support this, the proposed approach includes an auditing mechanism to automatically retrieve and show the exchanged information in order to solve potential issues or disputes.

## 3 THE RESTCHAIN FRAMEWORK

In this section, we provide technical details of the framework details. The overall architecture is reported in Figure 2 and it contains the core components of the proposed approach.

*IPFS.* The IPFS plays a crucial role inside the RESTChain approach. Indeed, it stores all the structures of the REST calls that must be invoked during the execution of the service choreography. For each REST call, the IPFS produces an ID that will indicate the location where the content can be lately retrieved. The collection of all the IDs is then stored inside the smart contract which will maintain an overall trace. The choice of the IPFS comes from the limitation of blockchain technology that restricts the expressiveness and the size of the stored information. Indeed, each operation inside the blockchain has a cost calculated with respect to the amount of data collected and processed. With the use of the IPFS, we are able to reduce those costs by decoupling the content of the REST calls from their execution. Indeed, while the information about RESTs (e.g. endpoint, body, URL, and others) is stored in the IPFS and read by each organization, their execution is started by the blockchain and completed by the organizations' services.

*Blockchain.* The blockchain stores the smart contracts used for managing the services' interactions. In particular, the blockchain

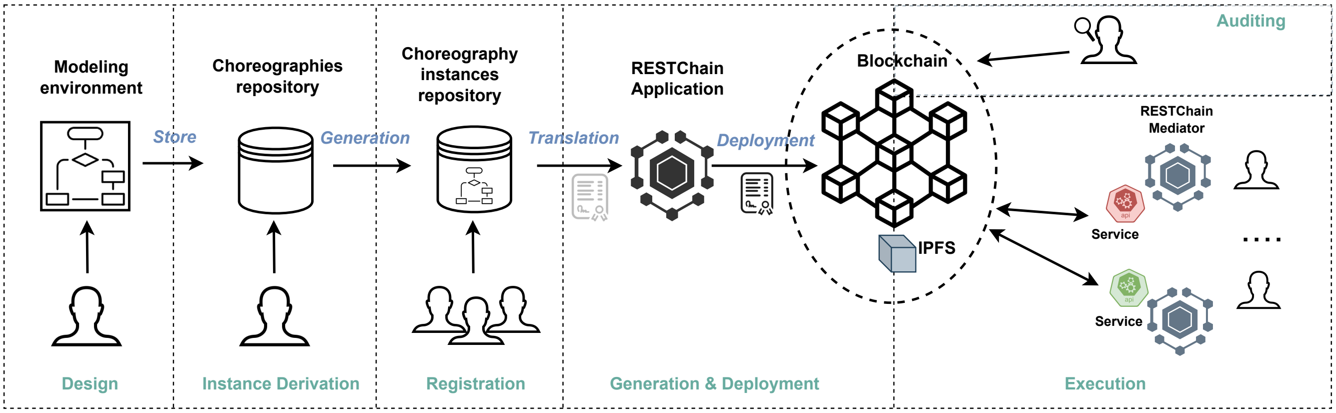

**Figure 1: RESTChain conceptual framework**

maintains in an immutable and transparent way the IDs of the IPFS where the REST calls are stored. Furthermore, the contracts coordinate and enforce the execution of choreography messages enabling only the expected interactions.

*RESTChain application.* The application is in charge of supporting the different RESTChain phases starting from the choreography design to the smart contract creation, execution, and deployment. In particular, this component is based on a back-end for supporting user management and providing the interaction primitives for communication with the blockchain. The Front-end provides instead the user interface and, also in this case, interaction primitives for the communication with the blockchain. The front-end enables the user’s interactions with the choreography’s life cycle, conveying data through the back-end and allowing communication with the blockchain, permitting the automatic deployment of the generated smart contracts. Therefore, it exposes all the functionalities for the choreography design and role subscription. Thanks to this application, RESTChain is able to provide support to the organizations, without acting as a central authority since it exposes only utility functionalities. Once the smart contract has been deployed within the blockchain the server is no anymore used, and the service’s collaboration takes advantage of the mediators distributed among involved organizations.

*Organization RESTChain mediator.* The mediator is a distributed software component that communicates directly with the blockchain and IPFS repositories and every organization is in charge of its maintenance. The mediator is placed under the organization’s supervision, and it has the role of executing any REST call belonging to the organization. Indeed, it is directly connected to the respective choreography smart contract, providing a bidirectional communication channel. There are several advantages to using this kind of component. First of all, the mediator allows the distribution of the computation and the execution of services without relying only on the RESTChain application avoiding a single central entity. On the opposite side, distributing completely the approach may affect the organizations, that have to host and run a heavy application without the support of another component. For this reason, RESTChain relies on a sort of decentralized approach with a balance between the provided application and the distributed mediators.

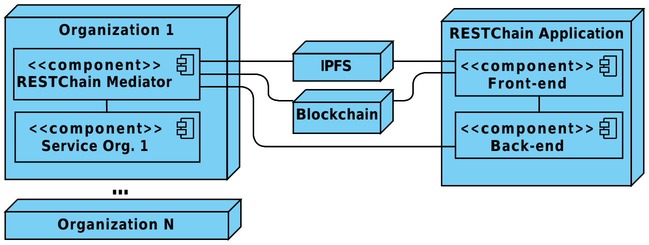

**Figure 2: RESTChain Architecture**

## 4 RELATED WORKS

Service composition tries to address the challenge of coordinating heterogeneous services based on different technologies. In particular with model-based techniques it is possible to abstract from specific and low-level implementation details of services, specifying only their interaction [10].

Service choreographies is a topic of interest that in the years has generated many works financed by the European Commission [3, 13] also for the adoption of these models in the IT companies [2]. The challenge of generating a REST architecture starting from BPMN choreographies has been faced in [11] where the authors propose a method for the semi-automatic generation of REST requests via natural language analysis. In our case, we are more interested in providing a flexible and trust layer, in which the specific actions and services to invoke can be defined at run-time, delegating in the modeling phase the definition of which components must communicate.

In [12] the authors present a web service orchestrator allowing the execution of business processes through a set of business rules. They propose also an XML-based language for the bindings of the process activities with data and services. These are then evaluated by the orchestrator during the execution phase in which, interacting with the events and checking the business rules, the process is updated. However, the execution of service choreographies using blockchain technology is still challenging [8], and just a few solutions are actually available [4–7]. However, in those contexts, the choreography is used to derive mainly on-chain contracts that coordinate a human-driven execution. With our idea instead, services have to communicate in an independent and autonomous way, thus exploiting the generated set of mediators. An approach for coordinating services is proposed in [1] where services choreographies are implemented using a permissioned blockchain. In particular, it allows connecting external services related to the participants, and communication is validated through the use of oracles. In our case instead, to address the trust issues in this kind of context, we adopt a direct binding between services and public blockchain components, relying only on the Ethereum protocol for validating messages.

In [14] the authors propose an approach in which service requesters can select and execute tasks from service providers interacting through smart contracts. However, the parties have to manually write and deploy their smart contracts, specifying the QoS and the prices. Also, the multiplicity of deployed contracts could significantly impact the costs that the providers and the requesters have to sustain. Differently, our approach aims at automatically supporting the process with the auto-generation and deployment of a single contract, focusing not on the service selection, but on the communication of already defined ones.

Other approaches are instead highly specific like the one proposed in [15] where the blockchain is used to compose services in the cloud manufacturing context. Here the main objective is to decentralize this kind of architecture, increasing also information transparency. Finally, the Ethereum blockchain is used in combination with the IPFS also seen in [9], in this case, to provide a trusted decentralized service marketplace.

## 5 CONCLUSIONS

This paper presented RESTChain, an approach, and a corresponding architecture that intends to permit the usage of blockchain infrastructure in the enactment and execution of a service choreography specification. The objective of the proposed approach is to provide an automatically generated infrastructure that can be easily used to integrate a set of inter-organizational REST-based services, that cooperate to reach the objectives of a choreography. The inclusion of blockchain technology permits to increase trust and successively audit the performed message exchanges. The approach has been explored over a public permissionless environment (e.g. Ethereum-like blockchains). The choice of such a paradigm permits us to say that the approach is viable if the time constraints foreseen by the choreography specification are somehow not much relevant and significant delays for each interaction can be accepted. If this is not the case the cost of the execution will probably overtake in many cases the benefit of having a trustful infrastructure. Such issues directly relate to the possible usage of one of the currently available scaling solutions or even to different blockchain technologies depending on context. Indeed, such alternatives provide a reduced overhead and cost for the inclusion of transactions that, in some cases such as private blockchains, can be reduced to almost zero. In our case, RESTChain wants just to constitute a first proposal to permit the combined usage of a REST protocol with a blockchain infrastructure in the enactment and execution of a choreography. In the future, we intend to further investigate the topic trying to characterize the different blockchain technologies and scaling solutions available, and their possible adoption in the discussed context.

## REFERENCES

[1] Marco Autili, Francesco Gallo, Paola Inverardi, Claudio Pompilio, and Massimo Tivoli. 2019. Introducing Trust in Service-Oriented Distributed Systems through Blockchain. In *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 149–154.

[2] Marco Autili, Alexander Perucci, Leonardo Leite, Massimo Tivoli, Fabio Kon, and Amleto Di Salle. 2021. Highly collaborative distributed systems: Synthesis and enactment at work. *Concurrency and Computation: Practice and Experience* 33, 6 (2021), e6039.

[3] Marco Autili, Amleto Di Salle, Francesco Gallo, Claudio Pompilio, and Massimo Tivoli. 2019. CHOReVOLUTION: Automating the Realization of Highly-Collaborative Distributed Applications. In *International Conference on Coordination Languages and Models (Lecture Notes in Computer Science)*, Vol. 11533. Springer, 92–108.

[4] Flavio Corradini, Alessandro Marcelletti, Andrea Morichetta, Andrea Polini, Barbara Re, Emanuele Scala, and Francesco Tiezzi. 2021. Model-driven engineering for multi-party business processes on multiple blockchains. *Blockchain: Research and Applications* 2, 3 (2021), 100018.

[5] Flávio Corradini, Alessandro Marcelletti, Andrea Morichetta, Andrea Polini, Barbara Re, and Francesco Tiezzi. 2020. Engineering Trustable Choreography-based Systems using Blockchain. In *SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, online event, [Brno, Czech Republic], March 30 - April 3, 2020*. ACM, 1470–1479.

[6] Flavio Corradini, Alessandro Marcelletti, Andrea Morichetta, Andrea Polini, Barbara Re, and Francesco Tiezzi. 2022. Engineering Trustable and Auditable Choreography-based Systems Using Blockchain. *ACM Transactions on Management Information Systems (TMIS)* 13, 3 (2022), 1–53.

[7] Flavio Corradini, Alessandro Marcelletti, Andrea Morichetta, Andrea Polini, Barbara Re, and Francesco Tiezzi. 2022. Flexible Execution of Multi-Party Business Processes on Blockchain. In *2022 IEEE/ACM 5th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*. IEEE, 25–32.

[8] Florian Daniel and Luca Guida. 2019. A Service-Oriented Perspective on Blockchain Smart Contracts. *IEEE Internet Computing* 23, 1 (2019), 46–53.

[9] Markus Klems, Jacob Eberhardt, Stefan Tai, Steffen Härtlein, Simon Buchholz, and Ahmed Tidjani. 2017. Trustless Intermediation in Blockchain-Based Decentralized Service Marketplaces. In *International Conference on Service-Oriented Computing (Lecture Notes in Computer Science)*, Vol. 10601. Springer, 731–739.

[10] Yun Ma, Xuanzhe Liu, Yihan Wu, and Paul Grace. 2013. Model-Based Management of Service Composition. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*. IEEE Computer Society, 103–112.

[11] Adriatik Nikaj, Mathias Weske, and Jan Mendling. 2019. Semi-automatic Derivation of RESTful Choreographies from Business Process Choreographies. *Software & Systems Modeling* 18, 2 (2019), 1195–1208.

[12] Natália Silva, Renata Carvalho, Ricardo Lima, and Cesar Oliveira. 2013. Integrating Declarative Processes and SOA: A Declarative Web Service Orchestrator. In *Proceedings of the International Conference on Semantic Web and Web Services (SWWS)*. The Steering Committee of The World Congress in Computer Science, Computer, CSREA Press, 5.

[13] Hugues Vincent, Valérie Issarny, Nikolaos Georgantas, Emilio Francesquini, Alfredo Goldman, and Fabio Kon. 2010. CHOReOS: Scaling Choreographies for the Internet of the Future. In *Middleware'10 Posters and Demos Track*. 1–3.

[14] Puwei Wang, Xiaohe Liu, Jinchuan Chen, Ying Zhan, and Zhi Jin. 2018. QoS-aware service composition using blockchain-based smart contracts. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*. ACM, 296–297.

[15] Chunxia Yu, Luping Zhang, Wenfan Zhao, and Sicheng Zhang. 2020. A blockchain-based service composition architecture in cloud manufacturing. *International Journal of Computer Integrated Manufacturing* 33, 7 (2020), 701–715.