Personalized Smart Contracts for IoT Data Certification

Alessia Pisu*, Livio Pompianu*, Salvatore Castello*, Daniele Riboni*, Salvatore Carta*

***Abstract*—Monitoring sensor data demands continuous and transparent processes. Existing centralized solutions for data monitoring grant sole authority to a single entity, risking data manipulation without consensus. Distributed ledger technologies offer promising solutions, but their scalability limitations pose challenges in handling the vast volume of IoT-related data. This paper proposes an innovative approach that leverages distributed ledger technologies to certify sensor data, allowing user groups to customize their certification policies. We develop our solution as a Hyperledger Fabric smart contract and test our approach using the energy consumption values we extract from a public dataset. This approach enhances trust and transparency in IoT monitoring while accommodating diverse stakeholder requirements.**

I. Introduction

Applications within the IoT sector, ranging from industrial plant management to Renewable Energy Communities [1], require continuous and transparent monitoring of values among the various involved entities. Standard solutions for monitoring IoT data include distributed platforms that collect data from field devices and provide a web interface for data observation [2]. However, such systems are not always considered reliable by all stakeholders, as they are controlled by a single entity with authority over the data, which can decide to delete or alter them without the agreement or awareness of other entities. Consequently, given the conflicting interests among stakeholders, making the certification process more dependable is imperative.

A possible solution to this trust issue is to exploit distributed ledger technologies, particularly blockchain [3] since such technologies offer tools for exchanging reliable information among parties that do not inherently trust each other. Unfortunately, simply inserting all sensor data into a blockchain is not feasible in IoT applications. Indeed, the blockchain's inability to remove old data conflicts with the large volume of data generated in these contexts, thus limiting the system's scalability [4]. Hence, limiting the amount of data saved in the blockchain is necessary to make the system usable. At the same time, the policy on which data to save cannot be trivially determined in advance, as each application has its own requirements, and any pre-selected configuration could harm some stakeholders while benefiting others.

In this context, we introduce an innovative approach for monitoring IoT values and certifying data using smart contracts. Our system allows each user group to customize their certification policy and choose which subset of data gives them the correct balance between the amount of blockchain space to occupy and the amount of data they need for a trustworthy certification. Users create custom policies as smart contracts and vote on which policies to use among the existing ones. Next, the system monitors the transactions users submit and verifies compliance with the contract policy voted on. The system recognizes and tracks users who do not adhere to the contract.

Our work brings the following contributions. (i) We propose an innovative approach for IoT data certification based on personalized smart contracts (Section II). (ii) We implement our solution as a Hyperledger Fabric smart contract (Section III). (iii) We validate our approach by simulating a case study in the energy sector (Section IV).

Although other works explored IoT data certification via distributed ledger technologies (e.g., [5]–[7]), they are often tailored for specific use cases, while our work focuses on the creation of policies customizable for each particular use case.

II. Our Proposal

We introduce a methodology for building and deploying certification policies customized for each user group. Our system allows users to build policies as custom instances of a smart contract we developed for the Hyperledger Fabric blockchain [8]. Accordingly, our system supports multiple users groups running on the same blockchain and smart contract. Figure 1 depicts the main steps of our approach.

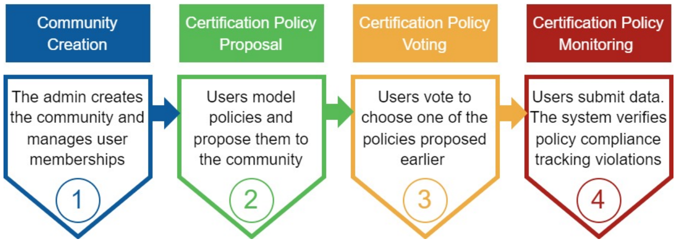
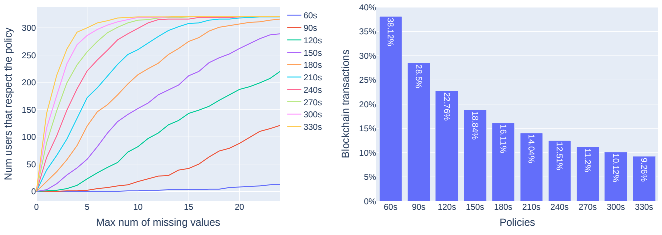

Community Creation
The admin creates the community and manages user memberships
1
Certification Policy Proposal
Users model policies and propose them to the community
2
Certification Policy Voting
Users vote to choose one of the policies proposed earlier
3
Certification Policy Monitoring
Users submit data. The system verifies policy compliance tracking violations
4

Fig. 1: Main steps of our methodology for data certification.

Phase 1. A *User* creates a new group of users, which we call a *Community*, and becomes its *Admin*. A community gathers users with a common goal, e.g. those belonging to the same Renewable Energy Community. Other users can apply to join the community. The Admin manages user membership; e.g., it may accept only active users of the community.

Phase 2. In the next step, the community negotiates a *Certification Policy*. A policy is a data acquisition strategy that must be adhered to by users in the same group. Users can create new policies as custom instances of our proposed smart

*University of Cagliari, Via Ospedale 72, Cagliari, Italy
Corresponding author: pompianu.livio@unica.it



(a) Policies.

(b) Transactions.

Fig. 2: (a): number of users who comply with the community policy calculated as the maximum number of missing values tolerated changes. Each line indicates the maximum time interval the user submits a value to avoid a missing value for the policy.
(b): for each policy, the percentage of transactions that minimize total system violations relative to the total number of transactions in the dataset.

contract. Our policy template allows users to customize various IoT-related parameters, such as: (i) the maximum timeframe for submitting a new measurement value; (ii) the maximum number of *missing values* tolerated. A missing value is an event that happens when a user does not send a measurement within the timeframe set. Indeed, since we can assume various transmission problems, it is legitimate for these policies to set tolerance thresholds of no transmissions. For example, a user can define a policy with a maximum timeframe of 60 seconds (i.e., the user must transmit at least one measurement per minute from the field to the blockchain) and a tolerance of 5 missing values before a violation occurs.

At this stage, users have a fixed amount of time to propose one or more policies to vote on: they can be either newly created policies or policies already available in the system.

Phase 3. After collecting proposed policies, we start a voting phase in which each user votes for the policy they prefer among all the proposed policies. The voting phase has a time limit; once it expires, the one obtaining the most votes is set as the community policy.

Phase 4. Finally, the system monitoring starts. Users submit their *Measurements* for certification. The smart contract evaluates whether users comply with the community policy, detects policy *Violations*, and tracks both user-provided transaction data and system-detected violations in the blockchain.

III. Implementation

For developing the smart contract system, we decide to use the Hyperledger Fabric blockchain, due to its support for high-level programming languages, and its capability for tailored network management. We write the contracts in Java.

We develop each step of our methodology (presented in Section II) on the blockchain as a *transaction*. Specifically, the transactions we implement are: (i) *createUser*: Creates a new user; (ii) *createCommunity*: A user creates a community and becomes its admin; (iii) *joinCommunity*: A user asks permission to join a community; (iv) *acceptRequest*: An admin allows a user to join the community; (v) *registerDevice*: A user registers a new device; (vi) *startProposal*: An admin starts the policy proposal phase; (vii) *createPolicy*: A user creates a policy; (viii) *policyProposal*: A user adds a policy to the proposal list; (ix) *vote*: A user votes for one or more policies among those proposed; (x) *sendMeasurement*: A user sends its own measurements;

IV. Validation

To validate our approach, we performed experiments using a real-world dataset acquired from a smart-home testbed[1]. The dataset was acquired by researchers of the Center for Advanced Studies in Adaptive Systems of Washington State University [9]. The test-bed is a two-story apartment equipped with several sensors, including presence sensors, door sensors, temperature sensors, and a whole-apartment electricity usage meter [10]. The dataset comprises data acquired during more than 300 days while a set of people performed different activities within the apartment. For the sake of our study, we considered only power meter data readings. Totally, the dataset includes about 115,000 power meter readings.

We used this dataset to simulate a Renewable Energy Community by considering the various energy values of distinct users as if they came from different homes. Each entry includes a timestamp indicating when the value was recorded, which was used to verify when the data was produced. Our experimentation aims to simulate various policies to demonstrate how our smart contract allows users to create policies personalized for each specific scenario.

In our use case of energy communities, we created 10 example policies defining the time interval for which users must transmit data. In the graph shown in 2a, we describe the results obtained from implementing different policies with different tolerance thresholds for missed measurements. The y-axis indicates the number of users who fully comply with the policy. Raising the tolerance threshold and the sampling interval increases the number of users complying with the policy. While it is necessary to keep sampling short and fast for some use cases, this is only sometimes possible because the transaction load in the blockchain would grow too large. Figure 2b illustrates the percentage of transactions recorded in the blockchain with the different policies compared to the total number of records the various users collect.

V. Conclusions

In this paper, we proposed a novel approach for IoT data certification, leveraging smart contracts to enable personalized certification policies tailored to specific use cases. By empowering user groups to define and vote on certification policies, our system promotes transparency, accountability, and trust among stakeholders. Also, validating our solution through a simulated case study in the energy sector underscores our approach's practical applicability and effectiveness. In the future, we want to explore this approach to implement different use cases across various domains.

[1] https://casas.wsu.edu/datasets/assessmentdata.zip

References

[1] G. Dóci, E. Vasileiadou, and A. C. Petersen, “Exploring the transition potential of renewable energy communities,” *Futures*, vol. 66, pp. 85–95, 2015.

[2] Z. S. Ageed, S. Zeebaree, M. Sadeeq, M. B. Abdulrazzaq, B. W. Salim, A. A. Salih, H. M. Yasin, and A. M. Ahmed, “A state of art survey for intelligent energy monitoring systems,” *Asian Journal of Research in Computer Science*, vol. 8, no. 1, pp. 46–61, 2021.

[3] K. Wüst and A. Gervais, “Do you need a blockchain?” in *2018 crypto valley conference on blockchain technology (CVCBT)*. IEEE, 2018, pp. 45–54.

[4] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with iot. challenges and opportunities,” *Future generation computer systems*, vol. 88, pp. 173–190, 2018.

[5] D. Kirli, B. Couraud, V. Robu, M. Salgado-Bravo, S. Norbu, M. Andoni, I. Antonopoulos, M. Negrete-Pincetic, D. Flynn, and A. Kiprakis, “Smart contracts in energy systems: A systematic review of fundamental approaches and implementations,” *Renewable and Sustainable Energy Reviews*, vol. 158, p. 112013, 2022.

[6] W. Xiong and L. Xiong, “Data trading certification based on consortium blockchain and smart contracts,” *IEEE Access*, vol. 9, pp. 3482–3496, 2020.

[7] N. Elia, F. Barchi, E. Parisi, L. Pompianu, S. Carta, A. Bartolini, and A. Acquaviva, “Smart contracts for certified and sustainable safety-critical continuous monitoring applications,” in *European Conference on Advances in Databases and Information Systems*. Springer, 2022, pp. 377–391.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, “Hyperledger fabric: a distributed operating system for permissioned blockchains,” in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[9] D. J. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas, “Collecting and disseminating smart home sensor data in the CASAS project,” in *Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009.

[10] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, “CASAS: A smart home in a box,” *Computer*, vol. 46, no. 7, pp. 62–69, 2013.