Demo: SenShaMart - A Sensor Sharing Marketplace for IoT

Anas Dawod, Dimitrios Georgakopoulos, Prem Prakash Jayaraman, and Josip Karabotic Milovac
Swinburne University of Technology
Melbourne, Australia
{adawod, dgeorgakopoulos, pjayaraman, jkaraboticmilovac}@swin.edu.au

Kewen Liao
HilstLab, Peter Faber Business School
Australian Catholic University
Sydney, Australia
kewen.liao@acu.edu.au

Panos K. Chrysanthis
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA, USA
panos@cs.pitt.edu

2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-3986-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDCS57875.2023.00122

***Abstract*—The Sensor Sharing Marketplace (SenShaMart) enables IoT applications to find IoT sensors, which are owned and managed by other parties, integrate them, and pay for using their data. To provide corresponding services that implement that FAIR (*Findable, Accessible, Interoperable, Reusable*) principles of IoT, SenShaMart incorporates a specialized blockchain that manages all the information its services need to allow different parties in IoT to describe, query, integrate, pay for, and use IoT sensors and their data. The paper presents the SenShaMart's architecture, implementation, evaluation, and demonstration.**

Keywords—Blockchain, Internet of Things, Sensor Discovery, Sensor Sharing, Data Marketplace

I. Introduction

Tens of billions of IoT devices and sensors are currently connected to the Internet and major industry players project that their number will reach anywhere between 25 to 125 billion in 2030 [1]. These vast number of IoT devices and sensors provides an exceptional opportunity to observe the physical world and distil valuable timely information to address major challenges (e.g., bushfire prediction) left from the past due to a lack of timely and accurate information[2]. However, the potential of IoT has not been fully realised, as IoT applications currently operate in *silos*, i.e., IoT sensors are usually owned by different individuals or organizations for private use. Hence, vast opportunities exist to use and share the cost of sensors provided by other parties (i.e., sensor providers).

Existing solutions for sharing IoT sensors (e.g., [3] and [4]) are deficient in 1) standards to describe IoT sensors, their data and the cost for their usage; 2) a discovery mechanism forIoT sensors, which involves formulating and querying the description of IoT sensors supplied by their providers; 3) ensuring IoT applications have an unfettered right to discover, pay, and use any sensor offered by any provider without any control or management from any entity; 4) scalability to support the rapidly expanding volume and variety of IoT sensors; and 5) a payment mechanism that allows IoT applications to regularly pay for used sensors (i.e., pay as you go) and enables cost-sharing between IoT applications.

This paper presents SenShaMart - a Sensor Sharing Marketplace that implements the *Findable, Accessible, Interoperable, Reusable* (FAIR) [5] principles of IoT. More specifically, Section II presents SenShaMart's decentralised marketplace architecture, that novelty contributes: 1) a specialized SenShaMart blockchain that manages the descriptions of available sensors and their data, their integration end-points and protocols, and related costs, which are necessary for IoT sensor discovery, integration, and payment; 2) Services for semantic registration and query processing of sensor metadata in the SenShaMart blockchain; and 3) Services for sensor payment and sensor integration via the MQTT protocol. Section III introduces the implementation of SenShaMart, and it is followed by the SenShaMart evaluation in Section IV. Finally, Section V provides the demonstration of SenShaMart.

II. SenShaMart Archetecture

SenShaMart consists of distributed nodes, called SenShaMart nodes or SSM Nodes, as shown in Figure 1(a). Some SSM Nodes include a SenShaMart (SSM) Broker or simply Broker that is responsible for helping with integrating the IoT sensors (will be detailed in the subsection II.C). SSM Nodes interact with the underneath layer, which is a specialized blockchain, called SSM Blockchain. The SSM Blockchain is responsible for storing the IoT *sensor metadata*, i.e., all the required information for IoT sensor registration, query, integration, and payment, Broker metadata, and a log of payments for IoT sensors. The decentralization of the SSM Blockchain ensures that no entity controls SenShaMart or manages it. Figure 1(b) shows the detailed architecture of a SSM Node, which is built around a corresponding SSM Blockchain Node.

A. SenShaMart Blockchain

The SSM Blockchain is a decentralized registry of IoT sensors, which includes the required information to make IoT sensors and their data *Findable, Accessible, Interoperable, and Reusable*. Just like many other existing blockchains (e.g., the Bitcoin blockchain [6]), the SSM Blockchain allows SSM Blockchain Nodes to generate new blocks, contribute to SSM Blockchain consensus, and verify newly generated blocks across the entire SSM Blockchain. Unlike other existing blockchain-based solutions for IoT (e.g., [7]) SenShaMart uses the SSM Blockchain to store *only* IoT sensor metadata and related information that is required for IoT sensor description, registration, query, integration, and payment.

2575-8411/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDCS57875.2023.00122

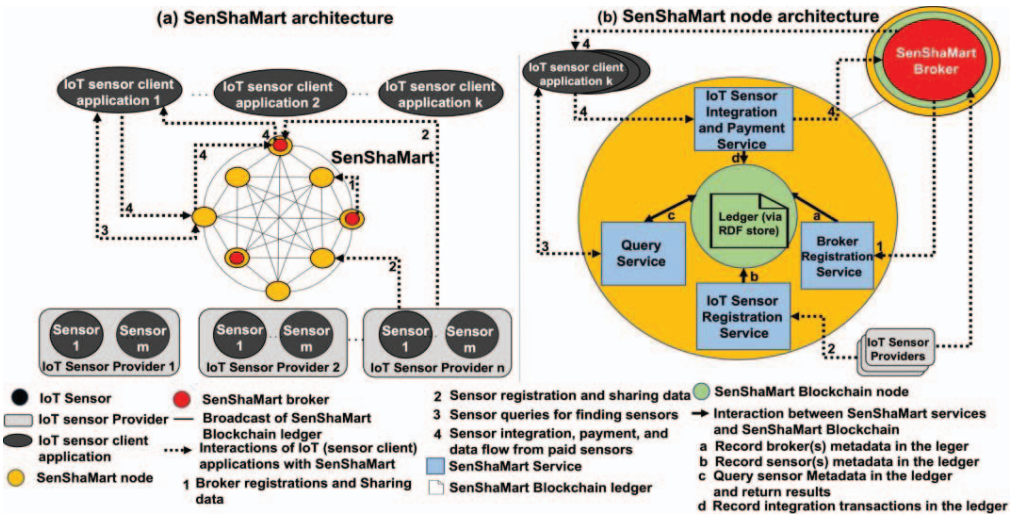


Fig. 1. SenShaMart high-level architecture(a) and node architecture (b).

SenShaMart avoids blockchain-related bottlenecks by not storing any IoT sensor data in the SSM Blockchain as this would have significantly reduce its scalability.

The SSM Blockchain provides novel features that support IoT sensor sharing via the SenShaMart services. The SSM Blockchain includes a novel ledger (we refer to it as the SSM Blockchain Ledger) for storing IoT sensor metadata organized in the form of blocks, which we refer to as SSM Blockchain Blocks. The SSM Blockchain Ledger uses our developed Semantic Sensor Network (SSN) [8]–based ontology for describing sensors and their data [9, 10]; Furthermore, SSM Blockchain incorporates an RDF triple store that is used to record IoT sensor metadata as triples. The SenShaMart's RDF triple store supports efficient processing of semantic queries involving IoT sensor metadata.

B. IoT Sensor and Broker Registration and Query Services

The IoT Sensor Registration Service (SRS) allows sensor providers to register their IoT sensors in SenShaMart to make them discoverable by IoT client applications via the Query Service (QS). The sensor providers are currently the parties responsible for submitting all sensor metadata via the SRS, which automatically submits them to SSM Blockchain to be validated and stored. The IoT sensor metadata will be stored in the SSM Blockchain in blocks and these are synchronised with the triple store that stores IoT sensors metadata in a form of triplets that are compliant with our developed ontology mentioned in Subsection A. As soon as the metadata of an IoT sensor is stored in the SSM Blockchain, any sensor query submitted via the QS of any SSM Node can "find" this IoT sensor. Similarly, the Broker Registration Service (BRS) allows any SSM broker to register itself in SenShaMart via submitting the metadata of the brokers to SSM Blockchain to be validated and stored.

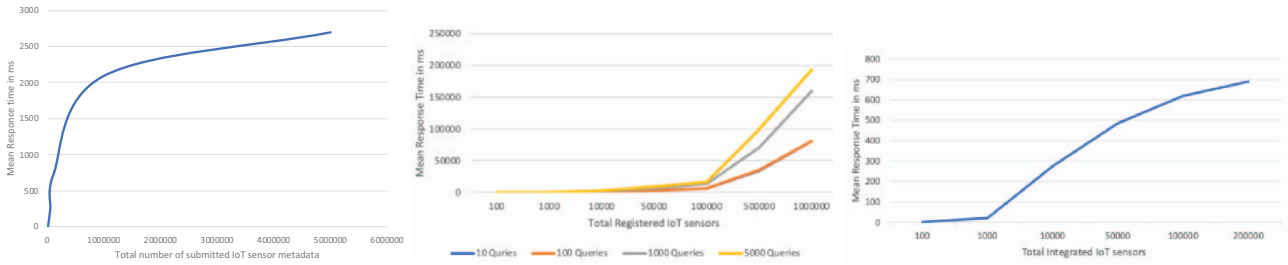
To search for available IoT sensors or Brokers, IoT (client) applications submit queries to the QS via any SSM Node. The QS currently supports SPARQL queries that use our developed ontology and cover the entire spectrum of IoT sensor and Broker metadata. The queries submitted to QS are processed using the RDF triple store built by the SSM Blockchain. For each sensor query submitted by a sensor client application, the QS returns the sensor metadata (in the form of triplets) of all available IoT sensors that satisfy the submitted sensor query. The IoT applications are responsible for selecting which sensor to use from the sensor query results. They submit integration requests for the selected sensors to the Sensor Integration and Payment Service (SIPS).

C. IoT Sensor Integration and Payment Service

To integrate the selected sensors, the IoT sensor client applications first use the sensor query results they obtained from QS to extract the IDs of the sensors they select. The sensor client applications then send an integration request to the SIPS service, which includes the identity of the requesting IoT application, the IDs of the selected IoT sensors, and the payment amount for the selected sensors that determine the duration of and/or amount of data each selected sensor will provide. IoT sensor metadata contain which SSM Broker will help to share the data for each IoT sensor. The SSM Broker will then activate the data flow of selected sensor to the sensor client application when it has verified the SSM Blockchain integration transaction. The SIPS automatically terminates the Broker of each client application and stops the sensor data flow to the IoT client application whenever the payment made by its sensor payment transaction runs out.

III. Implementation

SenShaMart is implemented as three layers. The first layer is the Application Programming Interfaces (APIs) which allow IoT applications and IoT sensor providers to communicate with SSM Services. The second layer is the SSM Services, which contain the functionalities for discovering, using, and cost-sharing of IoT sensors. The last layer is the SSM Blockchain which is responsible for managing the required information for making IoT sensors and their data *Findable, Accessible, Interoperable, and Reusable*.



(a) (b) (c)

Fig. 2. Evaluation plots: The relationship between the mean response time and a) the number of submitted IoT sensor metadata for sensor registration, b) the number of registered IoT sensors with varies number of queries, and c) the number of integrated IoT sensors.

We have used *NodesJS* to implement SenShaMart including the SSM Blockchain and SSM Services. *The SHA-256* hash function has been used to provide hash values to chain the blocks in the SSM Blockchain Ledger. The peer-to-peer communication inside the SSM Blockchain is supported by Web Socket protocol (*ws)*. The *N3 RDF triple store* was used in each of the SSM Blockchain Nodes to store IoT sensor metadata triples in the SSM Blockchain Ledger. *MQTT.js* has been used to implement MQTT protocol.

IV. Evaluation

An experimental-based evaluation has been conducted to measure SenShaMart's performance and scalability in terms of the number of sensor registrations, queries, and integrations. Up to 100,000 of sensor (client) applications, 5,000,000 IoT sensors, and 5,000 queries have been used in this evaluation. As shown in Figure 2, we have measured the response time of IoT sensor registration, query, and integration (payment time is included in the integration) with respect to the increasing number of IoT sensors.

The result shows the response time grows approximately linearly, demonstrating superior performance.

V. DEMONSTRATION

In this demonstration, SenShaMart will be deployed on 10 SSM Nodes using Nectar research cloud. Each SSM Node used the NeCTAR Ubuntu 16.04 LTS (Xenial) amd64 [v37] operating system. Also, each SSM Node has 16 GB of RAM, a public IP, and 500 GB of Hard Disk. Also, four different IoT sensors, two actuators, and two IoT applications will be deployed. These sensors are 1) a temperature/humidity sensor DHT22 connected to Arduino Uno board; 2) a PT100 temperature sensor connected to a Raspberry pi 3; 3) a weather station API that provides weather information; and 4) a camera connected to a raspberry pi 4. The first two sensors are deployed at Swinburne IoT laboratory, while the last sensor is deployed at the Factory of the Future-Swinburne University. Regarding to actuators, the first one is a heater and the second one is an alarm. The deployment of the two IoT applications are explained after showing the demonstration steps. SenShaMart's demonstration includes the following steps:

- SSM Nodes who want to be involved in integrating IoT sensors use the Broker Registration Service (BRS) to register as SSM Broker. We will register three SSM Brokers with different characteristics. Figure 3(a) shows an example of SSM Broker registration;
- IoT sensor providers use the Query Service (QS) to find a list of available SSM Brokers. Then, providers can choose one of them to help with sharing their sensor data with different IoT applications. Figure 3(b) shows an example for querying available SSM Brokers;
- Providers use the Sensor Registration Service (SRS) to register the four sensors mentioned above. Figure 3(c) shows an example of a registered IoT sensor;
- Two IoT applications use the Query Service (QS) to find suitable IoT sensors for their needs from the various registered IoT sensors. The QS return a list of related IoT sensors to be selected by the IoT applications. This step along with the previous one make IoT sensors and their data *Findable*. Figure 3(e) shows a query example to find registered IoT sensors;
- Two IoT applications use the Sensor Integration and Payment Service (SIPS) to integrate the selected IoT sensors, pay them, and access their data. This step ensures the *Accessibility* and *Reusability* of IoT sensors and their data. Figure 3(d) show an integration transaction example that used to pay and integrate an IoT sensor;
- The four IoT sensors share their data via SSM Broker. Without a loss of generality, in this demonstration, we use MQTT protocol, which is adopted by virtually all known IoT platforms. Other widely adopted protocols like CoAP and HTTP are similarly supported, but they are not implemented in this demonstration. This step shows the data *Interoperability* between IoT sensors and IoT applications.

The two IoT applications that deployed in this demonstration use the deployed IoT sensors for the following two scenarios. The first IoT application is used to solve the problem of roof ice dam. Ice dam is created at the roof's edge in cold environment and blocks the melted snow from draining off the roof causing celling damages as it accumulates water behind it, which leaks into the house. The IoT application uses the weather station and

(a)

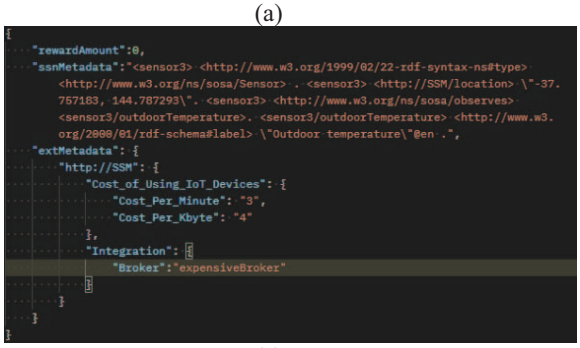
Query:

```
SELECT ?name ?costPerMinute ?costPerKilobyte WHERE  {
  ?name <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://SSM/Broker>.
  ?name <http://SSM/Cost_of_Using_IoT_Devices/Cost_Per_Minute> ?costPerMinute.
  ?name <http://SSM/Cost_of_Using_IoT_Devices/Cost_Per_Kbyte> ?costPerKilobyte.}
```

Go!

costPerMinute	costPerKilobyte	name
10	15	broker1
2	3	cheapBroker
30	50	expensiveBroker

(b)



(c)

(d)

Query:

```
SELECT ?name ?sensorCostPerMinute ?sensorCostPerKilobyte ?brokerCostPerMinute ?brokerCostPerKilobyte ?location ?observes WHERE {
  ?name <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/ns/sosa/Sensor>.
  ?name <http://SSM/location> ?location.
  ?name <http://SSM/Cost_of_Using_IoT_Devices/Cost_Per_Minute> ?sensorCostPerMinute.
  ?name <http://SSM/Cost_of_Using_IoT_Devices/Cost_Per_Kbyte> ?sensorCostPerKilobyte.
  ?name <http://SSM/Integration/Broker> ?broker.
  ?broker <http://SSM/Cost_of_Using_IoT_Devices/Cost_Per_Minute> ?brokerCostPerMinute.
  ?broker <http://SSM/Cost_of_Using_IoT_Devices/Cost_Per_Kbyte> ?brokerCostPerKilobyte.
  ?name <http://www.w3.org/ns/sosa/observes> ?observes }
```

Go!

sensorCostPerKilobyte	brokerCostPerMinute	brokerCostPerKilobyte	location	observes	sensorCostPerMinute	name
3	2	3	-37.823942, 145.174408	sensor1/atmosphericPressure	2	sensor1
4	2	3	-37.759516, 144.788173	sensor2/outdoorTemperature	3	sensor2
4	30	50	-37.757183, 144.787293	sensor3/outdoorTemperature	3	sensor3

(e)

Fig. 3. Demo examples for a) Broker registration, 2) Broker query, c) IoT sensor registration, d) IoT sensor integration and payment, and e) IoT sensor query.

temperature sensors' data to detect the creation of the ice dame and turn the heater *ON* and *off* accordingly.

The second IoT application is used to monitor the number of people inside the Swinburne Factory of the Future building. The IoT application uses the camera (IoT sensor) to count the number of incoming and outgoing people from the building. The IoT application turns the alarm *ON* if the number of people become more than 20.

ACKNOWLEDGMENT

This work is supported by the Australian Research Council (ARC) No. DP220101420.

REFERENCES

[1] ACS, "Australia's IoT Opportunity: Driving Future Growth." [Online]. Available: http://bit.ly/32rdri9

[2] A. Dawod, D. Georgakopoulos, P. P. Jayaraman, A. Nirmalathas, and U. Parampalli, "IoT device integration and payment via an autonomic blockchain-based service for IoT device sharing," *Sensors,* vol. 22, no. 4, p. 1344, 2022.

[3] K. Mišura and M. Žagar, "Data marketplace for Internet of Things," in *2016 International Conference on Smart Systems and Technologies (SST)*, 2016: IEEE, pp. 255-260.

[4] P. Gupta, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Towards a blockchain powered IoT data marketplace," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2021: IEEE, pp. 366-368.

[5] M. D. Wilkinson *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific data,* vol. 3, no. 1, pp. 1-9, 2016.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[7] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, "A trust architecture for blockchain in IoT," in *Proceedings of the 16th EAI international conference on mobile and ubiquitous systems: computing, networking and services*, 2019, pp. 190-199.

[8] M. Compton *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *Web semantics: science, services and agents on the World Wide Web,* vol. 17, pp. 25-32, 2012.

[9] A. Dawod, D. Georgakopoulos, P. P. Jayaraman, and A. Nirmalathas, "An IoT-owned service for global IoT device discovery, integration and (Re) use," in *2020 IEEE International Conference on Services Computing (SCC)*, 2020: IEEE, pp. 312-320.

[10] D. Georgakopoulos and A. Dawod, "Sensor Sharing Marketplace," in *Edge Computing–EDGE 2022: 6th International Conference, Held as Part of the Services Conference Federation, SCF 2022, Honolulu, HI, USA, December 10–14, 2022, Proceedings*, 2022: Springer, pp. 64-79.