IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 5, 2020, accepted September 10, 2020, date of publication October 2, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028430

The Merits of a Decentralized Pollution-Monitoring System Based on Distributed Ledger Technology

MARKUS LÜCKING[1], NICLAS KANNENGIEßER[2], MAURICE KILGUS[1], TILL RIEDEL[3], MICHAEL BEIGL[3], (Member, IEEE), ALI SUNYAEV[2], AND WILHELM STORK[4]

[1]Research Division for Embedded Systems and Sensors, FZI Research Center for Information Technology, 76131 Karlsruhe, Germany
[2]Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
[3]Chair for Pervasive Computing Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
[4]Institute for Information Processing Technologies, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

Corresponding author: Markus Lücking (luecking@fzi.de)

This work was funded by the Project COOLedger, Helmholtz Association of German Research Centers, under Grant HRSF-0081, and by the Russian Science Foundation under Project 19-41-06301.

ABSTRACT Pollution-monitoring systems (PMSs) are used worldwide to sense environmental changes, such as air quality conditions or temperature increases, and to monitor compliance with regulations. However, organizations manage the environmental data collected by such PMSs in a centralized manner, which is why recorded environmental data are vulnerable to manipulation. Moreover, the analysis of pollution data often lacks transparency to outsiders, which may lead to wrong decisions regarding environmental regulations. To address these challenges, we propose a software design for PMSs based on distributed ledger technology (DLT) and the long-range (LoRa) protocol for flexible, transparent, and energy-efficient environment monitoring and data management. To design the PMS, we conducted a comprehensive requirements analysis for PMSs. We benchmarked different consensus mechanisms (e.g., BFT-SMaRt and Raft) and digital signature schemes (e.g., ECDSA and EdDSA) to adequately design the PMS and fulfill the identified requirements. On this basis, we designed and implemented a prototype PMS and evaluated it in the field. The evaluation shows the effectiveness of DLT-based PMSs that include portable low-energy sensor nodes and demonstrates the applicability of the proposed software design for PMSs in contexts other than air pollution.

INDEX TERMS Blockchain, distributed ledger technology (DLT), Internet of Things (IoT), LoRa, low-energy sensors, pollution monitoring systems.

I. INTRODUCTION

The implementation of ever stricter environmental protection regulations over the past decade has increased the demand for reliable pollution data (e.g., particle pollution in the air) to support researchers, policy makers, and planners to make informed decisions on managing and improving the living atmosphere [1], [2]. To collect and store pollution data and allow for detailed analyses of environmental conditions (e.g., air quality), reliable pollution monitoring systems (PMSs) are required [3], [4]. Currently, local authorities (e.g., public environmental agencies) are given much of the responsibility to operate PMSs and carry out the provision of clean air, including monitoring air pollution and developing strategies to reduce air pollution [5]. Centralizing the responsibility of operating and maintaining PMSs in local authorities, however, caused a lack of transparency regarding the collecting, processing, and storage of sensor data (e.g., in terms of authenticity, integrity, and nonrepudiation) [6]. Consequently, the validation of pollution analyses is challenging for external parties because only few analysts ultimately perform data cleaning, calibration, applied analytical methods, and sensor data interpretation [7], [8], which can lead to incorrect assessments of environmental pollution [9] and mislead public regulation decisions and thus cause threats to human health. To counter such misguided regulations by enabling cross-validation by third parties [10] and to support better decision making regarding measures to improve air

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Quan.

quality [8], transparency should be increased in the collection, storage, and analysis of pollution data [7].

Local authorities primarily use stationary PMSs and predominantly investigate long-term effects on an urban macro scale (e.g., climate change). The large size and high cost of stationary PMSs (approximately 200,000 USD for installation and approximately 30,000 USD per year for maintenance [11]) limit the number of people who can participate in the collection of pollution data and how people can access stored data [8]. Moreover, the use of stationary PMSs hampers the collection of pollution data in a flexible and fine-grained way [12]. To support finer-grained air pollution monitoring, more detailed data on the spatial and temporal variability of air pollutants (e.g., particulate matter) are required [8]. Fast technological advancements regarding portable sensor nodes enable pollution sensing in a flexible way due to their low cost, small size, and battery-based power supply. In contrast to stationary PMSs, PMSs integrating portable sensor nodes allow for flexible ad hoc measurements and can achieve a high spatiotemporal resolution because such PMSs usually comprise a large number of sensor nodes [13], [14]. Although the use of portable sensor nodes in PMSs is promising to improve pollution monitoring, such PMSs have downsides regarding constrained resources of portable sensor nodes (e.g., low computational resources and constrained energy supply because of the use of batteries) [15]. These downsides make the design of PMSs using portable sensor nodes particularly challenging. Improvement of pollution monitoring requires a thorough analysis on how to design viable, transparent, and fraud-resistant PMSs that include portable sensor nodes.

Distributed ledger technology (DLT) promises to overcome many open challenges for the operation of PMSs (e.g., lack of data authenticity or tamper-proneness of stored data) [16]. DLT allows operating a transparent and tamper-resistant distributed database through a highly available and fault-tolerant infrastructure in which various storage and computing devices (referred to as DLT nodes) replicate data [17]. Many use cases of DLT applications exist that have successfully provided access management to data (e.g., [18]), identity management (e.g., for individuals, organizations, or devices [19]), and tamper-resistant logging and data storage (e.g., [20], [21]). Nonetheless, DLT is replete with various downsides, such as low performance compared to central and conventional distributed databases (e.g., poor scalability [22]) and extensive resource consumption (e.g., high storage requirements due to numerous ledger replications) [17]. The high resource consumption of distributed ledgers compared to conventional systems [23] is a particular challenge for the design of PMSs that employ portable, low-energy sensor nodes. At first glance, the downsides of DLT are particularly at odds with the core strengths of PMSs that include portable sensor nodes (e.g., easy maintenance and no place-boundness) [24] and call into question the effective use of DLT for PMSs using portable sensor nodes.

Since research on the applicability of DLT in the context of PMSs using portable low-energy sensor nodes is still in its infancy, little is known about the effectiveness of DLT in the context of PMSs incorporating portable low-energy sensor nodes and the appropriate resolution of the trade-off between flexibility and resource consumption. Thus, to combine the advantages of DLT and PMSs with low-cost portable sensor nodes and increase data authenticity and reliability of PMSs, we aim to answer the following research question:

RQ: How to design reliable PMSs that incorporate DLT and portable low-energy devices?

To answer our research question, we carried out an extensive requirements analysis for PMSs with low-energy sensor nodes and DLT by conducting a comprehensive literature review. Based on the derived requirements catalog, we designed and implemented a DLT-based PMS prototype[1] that uses low-energy sensors. To appropriately dimension the prototypical PMS, we benchmarked different signature algorithms (e.g., ECDSA and EdDSA) against different consensus mechanisms (e.g., BFT-SMaRt and Raft). To show that our prototypical PMS meets the requirements for PMSs (e.g., accuracy, low-energy consumption, and reliable data transmission), we conducted a field test over a 24-h period. Finally, we discussed to what extent the proposed PMS fulfills the identified requirements.

Our work presents essential requirements for components of PMSs (e.g., energy consumption of digital signature algorithms), which help to design effective PMSs and enable a better evaluation of conceptual and implemented PMS designs. Our detailed discussion and evaluation of alternative implementations (e.g., different consensus mechanisms) serves as a guide for the design of PMSs and similar Internet of Things (IoT) systems using low-energy devices and provides actionable insights into potential advantages and disadvantages of alternative system designs prior to implementation. Moreover, we show an overall concept on how a distributed ledger can be employed as IoT integration middleware in an entirely decentralized way. Therefore, our work addresses extant challenges regarding environmental data collection using portable sensor nodes, while promoting data authenticity, data availability, and tamper-resistance in PMSs.

The remainder of this work is structured as follows. First, we present the requirements for PMSs that we derived from scientific literature. Second, we describe the concept, design, and implementation of the proposed PMS. Third, we present and discuss the results of our evaluation efforts. Fourth, we compare the proposed PMS with extant approaches from the literature. The manuscript concludes with a discussion and an outlook for future research in the field of PMSs using portable sensor nodes and DLT.

[1] see https://github.com/lopess-project

II. REQUIREMENTS ANALYSIS FOR THE DESIGN OF POLLUTION MONITORING SYSTEMS

To identify requirements for PMSs incorporating portable low-energy sensor nodes and DLT, we conducted a literature review including extant scientific documents (e.g., journal articles and conference papers). We focused on scientific documents that reveal best-practices, goals, and challenges in the design of PMSs incorporating portable sensor nodes. This knowledge formed our foundation for deriving functional and nonfunctional requirements for PMSs to be considered in this work.

For the literature search, we followed established approaches [25], [26]. First, we developed and refined a search string focusing on the scientific investigation of the interplay of wireless sensor networks (WSN), low-power wide area networks (LPWAN), sensors nodes, and DLT. We applied the search string *(WSN* OR LPWAN* OR sensor* OR network*) AND (blockchain* OR "distributed ledger technology*")* to scientific databases we deemed relevant for requirements analysis: ACM DigitalLibrary, EBSCOhost, IEEEXplore, ProQuest, and ScienceDirect. The search revealed 217 documents in total (i.e., articles and papers). After screening the title, abstract, and keywords of each document, we excluded duplicates (46) and documents unrelated to our topic (155) and produced a preliminary set of 16 potentially relevant documents. We deemed a document relevant if it met the criteria of a peer-reviewed scientific paper or article that was written in English and described the implementation of a PMS. We carefully assessed the relevance of these documents by reading through their full texts and finally selected 14 documents relevant to the requirements analysis. Subsequently, we analyzed the remaining documents to identify requirements for PMSs by performing *open coding* and *axial coding* of the relevant literature [27]. First, we extracted goals and requirements discussed in the documents and noted the requirements' names and descriptions (*open coding*). For example, if authors strove for a high degree of provable data integrity by applying digital signatures, we extracted a requirement for high integrity. If mentioned, we also coded reasons and consequences for the requirements (*axial coding*), which helped us to aggregate similar requirements across articles. For example, we merged the requirements *integrity* [28] and *immutability* [29] into the requirement for *integrity* [30].

From a functional perspective, PMSs should collect and record data from (outdoor) sensor nodes (e.g., [31], [32]). We define a sensor node as a device composed of at least one sensor, a microcontroller, and other peripherals such as a GPS receiver (cf. Section IV-A). Only data of authorized sensor nodes should be stored by the PMS, which is why the PMS should allow for the registration and unregistration of sensor nodes (e.g., [33]). To do so, the PMS should integrate identity management for sensor nodes and consortium members who own these sensor nodes (e.g., [34]). The registration of new sensor nodes should be confirmed by all consortium members operating the PMS. Furthermore, the PMS should allow the public to retrieve recorded sensor data [35], for example, via a browser application [30]. For all recorded sensor data, the assigned organization for the sensor node should be visible to allow for transparency regarding the owner and operator of the respective sensor node [36]. Therefore, all consortium members should prove their identity before they are allowed to join the consortium running the PMS [33].

We identified thirteen nonfunctional requirements that a decentralized PMS incorporating portable sensor nodes needs to fulfill. Table 1 summarizes related nonfunctional quality characteristics for PMSs based on low-energy sensor nodes and DLT.

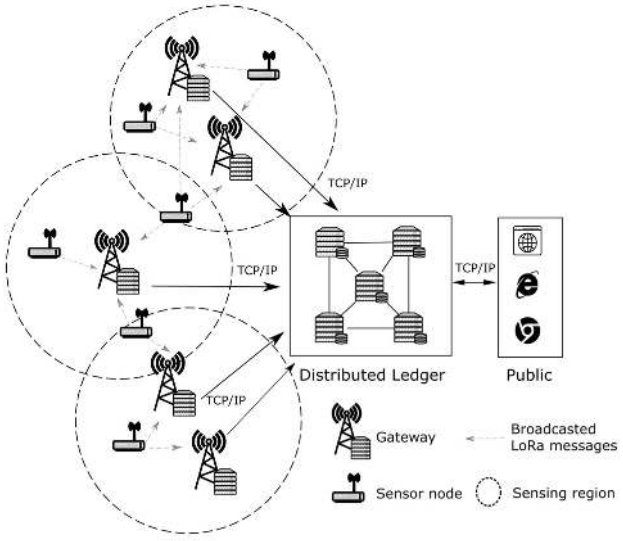

FIGURE 1. Schematic overview of the PMS architecture.

III. ARCHITECTURE OF THE POLLUTION MONITORING SYSTEM WITH PORTABLE SENSOR NODES

The architecture of our PMS comprises a consortium of organizations (e.g., research institutions, private contributors) that operate and use the PMS and four classes of technical components: *sensor nodes*, *gateways*, a *public key infrastructure (PKI)*, and a *distribution ledger* (cf. Figure 1). Each consortium member must register with the PKI of the PMS. Registered consortium members are allowed to register their sensor nodes with the PKI of the PMS to enable a transparent management of sensor nodes' identities and to achieve data integrity and nonrepudiation [42], [43]. After registering the sensor nodes, each consortium member can position their registered sensor nodes (e.g., in urban areas) and gather sensor data over a certain period in specific measurement intervals. At the end of each measurement interval, the sensor nodes digitally sign the collected data and broadcast the sensor message over a wireless network to surrounding gateways. The gateways forward the sensor message to the DLT nodes of the distributed ledger. On the DLT nodes, the digital signature of the sensor message is validated by a smart contract, which is a program running on the distributed ledger [17]. If the digital signature is found to be valid, the sensor message is stored

TABLE 1. Nonfunctional requirements for PMSs.

Name	Description	Exemplary reference
Accuracy	The proposed PMS should collect pollution data with an accuracy that serves the minimum requirements of the intended analysis.	[37]
Availability	The probability that a distributed ledger is operating at any point in time should be very high.	[30]
Bandwidth	The maximum bandwidth required by all devices connected within the PMS should be sufficient regarding the number of sensor nodes and the amount of sensor data to be transmitted over the network.	[31]
Censorship Resistance	It should not be possible for consortium members in PMSs, including private contributors and institutions, to deliberately prevent other consortium members from interacting (e.g., storing or reading sensor data) with the distributed ledger.	[30]
Energy Consumption	The energy consumption for the computing effort should be very low so that the sensor nodes can be supplied with power at least for the required measurement duration.	[38]
Independence	The components of the PMS and their interaction should be independent from proprietary hardware, software, and trusted authorities.	[30]
Integrity	The data submitted by sensor nodes and data stored on the distributed ledger are protected against unauthorized (or unintended) modification or deletion.	[28]
Nonrepudiation	The PMS should must prevent consortium members from successfully manipulating information about which sensor node sent which data and the owner of the respective sensor node.	[39]
Portability	Consortium members should be able to set up and operate the PMS independent of ambient conditions (e.g., power supply).	[40]
Reliability	The PMS should process and store the sensor data despite arbitrary failures (e.g., temporary network failures and malicious behavior of certain consortium members).	[41]
Scalability	The PMS should efficiently handle decreasing or increasing amounts of required resources (e.g., network traffic).	[31]
Throughput	The PMS should be able to commit at least the maximum number of expected sensor data per second under consideration of, for example, the maximum network bandwidth for communication between sensor nodes and the data storage.	[31]
Transparency	The stored sensor data, their originating sensor nodes, and the respective sensor node owners should be visible in the PMS and mappable to corresponding identities.	[41]

on the distributed ledger. All sensor messages are stored once and duplicates are discarded. Consortium members can access the stored sensor data directly via their DLT nodes. Additionally, each consortium member hosts an application programming interface (API) for the public to access stored sensor data (e.g., via a browser application).

IV. DESIGN AND IMPLEMENTATION OF THE POLLUTION MONITORING SYSTEM

Due to complex dependencies between PMS components (e.g., network bandwidth and throughput of the distributed ledger), it is necessary to implement the conceptual PMS and test it directly in realistic conditions in order to evaluate its fulfillment of the identified requirements presented in Section II. For the implementation of the proposed PMS concept (cf. Section III), we used customized, battery powered sensor nodes, the open LoRa protocol, standard dual-channel gateways, a distributed ledger, and digital signatures. To adequately dimension the PMS, we considered extant recommendations for pollution measurement intervals and the required coverage of the measurements (e.g., [44]), which form a focal base for the calculations regarding, for example, the required bandwidth and throughput.

A. SENSOR NODE

Sensor nodes comprise four components: one *microcontroller unit (MCU)* with its related *hardware security capabilities* for private key storage, at least one *sensor*, one *GPS module*, and one *LPWAN chip*. The core of each sensor node is a low-cost ESP32 MCU [45] because of its minimalist design and low power consumption compared to other low-cost single-board computers (e.g., Raspberry Pi Zero) [46]. The MCU starts each measurement in a predefined order by retrieving data from the particulate matter sensor (Nova SDS011) and the humidity and temperature sensor (Grove DHT22; cf. Appendix B). Afterwards, the MCU retrieves its location and the recent timestamp using the attached GPS module (Ublox Neo-6M). Subsequently, the MCU digitally signs recorded sensor data using its private key, which is stored in encrypted storage (eFuse). The sensor nodes send the digitally signed data over a long distance (up to approximately 5 km [47]) in an energy-efficient way using LPWAN technology (cf. Section IV-B). To do so, the MCU has an integrated LPWAN chip (Semtech LoRa transceiver SX1276), which can be extended with an external antenna to improve its range to up to 10 km.

The placement of sensor nodes in the public space enables any (nonauthorized) person to easily access the senor nodes software and hardware. Having access to the sensor node's on-board USB and serial peripheral interface, a person could read its data or manipulate its firmware and thus harm authenticity (e.g., by leaked private keys of sensor nodes).

To protect the sensor nodes' private keys, we enabled the MCUs' flash encryption using an Advanced Encryption Standard (AES) key. Flash encryption is a feature to encrypt the nonvolatile MCU memory storage (flash memory). The AES key is stored on the electronic fuse (eFuse) of the MCU, which is an one-time programmable read-only memory. Once the eFuse of the MCU is used for key storage, it cannot be modified again because the data are physically burned on

the eFuse. When flash encryption is enabled, application-based flash partitions (e.g., digital signing of measurements) are encrypted with the AES key. From there, the decryption can only occur at runtime via the MCU itself [48]. Each sensor node can decrypt its own flash memory, while unauthorized persons cannot access any data stored on the MCU (e.g., the private key) [49]. Moreover, we implemented a secure boot process to detect change of the sensor node software (e.g., modification of sensor data). The secure boot process cryptographically checks all software components of the MCU to be signed and verified before executing [50]. If software components are manipulated, the sensor node will refuse to boot [51].

B. WIRELESS COMMUNICATION

1) PROTOCOLS

LPWAN technology enables sending and receiving small amounts of data over a range of 1 to 10 km at low power consumption and includes three predominant protocols: *NB-LTE*, *Sigfox*, and *LoRa* [52]. NB-LTE operates within a licensed frequency band, which is why there are no restrictions regarding the maximum number of messages per day. Furthermore, the licensed spectrum achieves a higher degree of reliability and quality of service compared to Sigfox and LoRa [53]. NB-LTE primarily establishes a random access procedure, where each sensor node sends a sequence of signal messages to request resources of the base station. This resource allocation procedure consumes additional energy, shortens battery life, and reduces cost efficiency compared to unlicensed protocols (e.g., LoRa) [54].

TABLE 2. Comparison of LPWAN protocols based on [54].

Criteria	NB-LTE	Sigfox	LoRa
Uplink data rate	20 kB/s	100 B/s	300 B/s-50 kB/s
Max. payload size	1600 B	12 B	243 B
Range (urban)	1 km	10 km	2 to 5 km
Private networks	No	No	Yes
ISM band	No	Yes	Yes
Sending limitations	No	Yes	Yes

LoRa and Sigfox use unlicensed but duty-cycle-regulated industrial, scientific, and medical (ISM) bands below 1 GHz, which can transmit data over several kilometers depending on their environment (cf. Table 2). Compared to LoRa, Sigfox strictly limits the uplink data rate, the maximum payload size, and the number of messages that can be sent per day. These limitations are intended to achieve an ultralow energy consumption, long transmission range, and increased receiver sensitivity. For the proposed PMS, a minimum payload size of 121 B is required (28 B sensor raw data, 29 B GPS data, and 64 B digital signature; cf. Section IV-B3). The Sigfox protocol limits the maximum payload size of 16 B, and is therefore unsuitable for the proposed PMS. In contrast, LoRa offers a maximum payload size of 243 B [55].

Since LoRa best fulfills the requirements for the PMS, we decided to use LoRa in our own wireless network (cf. Section IV-B2). We chose a predefined LoRa configuration for applications in urban areas (spreading factor of nine; code rate of one) and a bandwidth of 250 kHz (maximum transmission rate of 439 bit/s) because a detailed performance analysis of LoRa networks is not within the scope of this work [56].

To design the PMS regarding throughput and scalability (cf. Table 1), an appropriate pollution measurement interval must be determined. The measuring interval of the sensor nodes to determine the air quality depends on the targeted resolution of the air pollution monitoring [57] and varies from once each minute [58] to once each hour [59]. To achieve a sufficient and feasible temporal resolution of the sensor data considering LoRa's technical capabilities, we defined a measuring interval of five minutes.

2) NETWORK ARCHITECTURE

LoRa wide area network (LoRaWAN) is an LPWAN protocol that specifies the upper network layers of the LoRa protocol including sensor nodes (or more general terminal devices), gateways, network servers, and application servers [60]. LoRaWAN predominantly serves as a routing protocol for the communication between application servers and sensor nodes. Sensor nodes use the LoRa protocol to transmit data to gateways. Subsequently, gateways use the standard transmission control protocol (TCP) and the internet protocol (IP) to send sensor messages to the network server to register and authenticate the respective sensor node before forwarding data to the target application server. By doing so, LoRaWAN ensures that only registered devices can send data to application servers (e.g., DLT nodes) [61].

Public network servers of a LoRaWAN (e.g., The Things Network) are usually operated by private organizations (e.g., ChirpStack). Public network servers have the ability to reject legitimate data from sensor nodes. Therefore, the use of public LoRaWANs comes with uncertainties regarding the network operation and potential loss of data integrity through network servers [62]. To avoid having a single organization that manages the PMS's network communication, we decided to set up our own dual-channel LoRa gateways (Dragino LG02) for communication between the sensor nodes and the distributed ledger. Each consortium member can individually set up a LoRa network to enable the communication between sensor nodes and the distributed ledger, as described in Section IV-C4.

3) DIGITAL SIGNATURES

A widespread approach to prove authenticity of data is the use of digital signatures and public key cryptography, which is also applied in DLT. The use of cryptography increases energy consumption due to more computationally intensive operations [17]. Several signature algorithms have been presented that differ, for example, in their space and time complexity, their energy consumption, and the degree of security for signing data or approving authenticity of signed data. To find a suitable signature algorithm for the PMS, we evaluated five different signature algorithms for signing data under

TABLE 3. Comparison of different signature algorithms on an ESP32 micro-controller.

Signature algorithm	Exec [ms]	Size [kB]	Memory [kB]	Energy [J]	Signature use	Quant.-safe
BLISS	56	4*	17.4	0.0154	Multiple	Yes
ECDSA	123	0.064	4.9	0.0331	Multiple	No
EdDSA	32	0.064	4.9	0.0082	Multiple	No
LMS	1,120	2.5	18.9	0.3095	Single	Yes
XMSS	330	1.45	18.5	0.0897	Single	Yes

Exec. — *Execution time for signature generation*
Size — *Size of the produced signature*
Memory — *Overall memory consumption for signature generation*
Energy — *Overall energy consumption for signature generation*
Signature use — *Private key can be used either for a single signature or for multiple signatures*
Quant. safe — *To be secure against an attack by a quantum computer*
* *Compressible to about* 700 B *using Huffman tables*

consideration of the following factors: *period for signing data*, *memory consumption*, *energy consumption*, *one-time signature use*, and *quantum safety* (cf. Table 3). To evaluate the energy consumption of the digital signature generation on the used MCU, we used a National Instruments USB-6216 module running at 10 kHz. We measured the voltage draw from the MCU while performing the signing operation and excluded the voltage draw of other MCU components such as the LPWAN chip or LEDs (cf. Figure 2).

The only digital signature schemes that meet the limitation of LoRa's payload size (243 B) are the elliptic curve digital signature algorithm (ECDSA) and the Edwards-curve digital signature algorithm (EdDSA). The results of the energy consumption evaluation revealed that a shorter signature generation time is correlated to a lower energy consumption. Thus, EdDSA is the most fitting algorithm for our purpose.

C. DISTRIBUTED LEDGER AS IoT INTEGRATION MIDDLEWARE

1) BACKGROUND

Distributed ledgers are (Byzantine) fault-tolerant [63] and append-only distributed databases whose operation is enabled by DLT [17]. In most distributed ledgers (e.g., Bitcoin or Ethereum), each DLT node stores and maintains a local copy of the data stored on the ledger and new data are appended to the local ledger in the form of transactions. When a DLT node receives a new transaction, the DLT node first validates the new transaction using digital signatures [43]. If the transaction is valid, the DLT node keeps the valid transaction in memory and forwards the transaction to adjacent DLT nodes in the network. These DLT nodes also validate and forward the transaction accordingly. Finally, the validated transactions are directly appended to the distributed ledger or placed in a higher-level data structure (referred to as a block), which is then appended to the distributed ledger. To achieve consistency among the local replications of the ledger kept on the DLT nodes, distributed ledgers use a consensus mechanism.

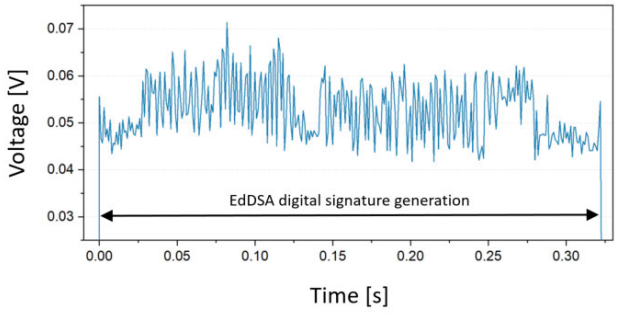

FIGURE 2. Voltage draw during EdDSA signature generation on a sensor node's MCU.

Most consensus mechanisms used in DLT are at least crash fault tolerant (e.g., Kafka or Raft) or even Byzantine fault tolerant (e.g., Nakamoto consensus). Crash fault tolerance refers to the ability of a consensus mechanism to achieve consensus among all validating nodes despite (temporarily) unavailable nodes, for example, due to network latency, hardware errors, or because a node has left the distributed ledger network. Byzantine fault-tolerant consensus mechanisms are crash fault tolerant and can additionally handle malicious behavior of nodes [63]. For example, when a user transfers the same digital asset to different recipients at the same time (referred to as double-spending [64]), Byzantine fault-tolerant consensus mechanisms manage to agree on storing one of these transactions. Consensus mechanisms tolerate crash faults and/or Byzantine failures only to a certain threshold (e.g., 1/3 of malicious nodes [65], [66]). This maximum is referred to as *fault tolerance*.

Distributed ledgers also differ regarding their read and write permissions. There are four types of distributed ledgers [17]: *private-permissionless*, *private-permissioned*, *public-permissionless*, and *public-permissioned*. The terms *public* and *private* refer to read permissions of DLT nodes in a distributed ledger, which means that nodes must first be authorized to join the distributed ledger. The terms *permissioned* and *permissionless* refer to DLT nodes' permission to validate transactions and take part in the consensus mechanism. These permissions are comparable to writing permissions in conventional databases. DLT nodes that take part in consensus finding are called validating nodes.

2) SELECTION OF A DISTRIBUTED LEDGER

Based on the identified requirements for PMSs (cf. Table 2), trade-offs between DLT characteristics [17], and DLT archetypes [17], we selected a suitable distributed ledger for the proposed PMS. Since the consortium members should be verified before participation and the requirements for non-repudiation and transparency, including organizations' identities, are particularly high in PMSs, we decided to use a permissioned distributed ledger in which only verified consortium members are allowed to operate a validating node. Every consortium member operating a DLT node should have access to the stored data. To reduce storage consumption, only the consortium members store replications of the ledger. The consortium members enable outsiders to access stored sensor data via a web application or to set up their own nodes with read permissions.

Compared to public-permissionless distributed ledgers, private-permissioned distributed ledgers mostly offer a higher degree of flexibility (e.g., maintainability), better performance (e.g., fast transaction confirmation and high throughput), and a high degree of transparency [17]. In permissioned distributed ledgers, transparency is increased because all consortium members are known and their real identities are assigned to public keys using a PKI. Since all transactions in a distributed ledger are digitally signed, their corresponding issuer is easy to identify using the consortium members' public keys. In addition, private distributed ledgers mostly do not employ a pricing scheme for the execution of smart contracts (e.g., *gas* in Ethereum), which can decrease operational cost (e.g., for the execution of smart contracts) when using distributed ledgers [17]. Therefore, we decided to use Hyperledger Fabric (HLF) to implement a blockchain for the proposed PMS [67] (cf. Section IV-C3).

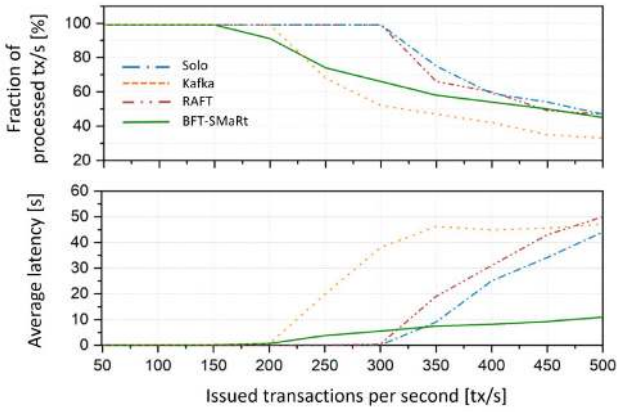

FIGURE 3. Throughput of different consensus mechanisms (i.e., Solo, Kafka, Raft, and BFT-SMaRt) for varying transaction issuance rates on Hyperledger Fabric for four orderer and four peer nodes.

3) HYPERLEDGER FABRIC

HLF incorporates three software components: *clients*, *peer nodes*, and *orderer nodes* [67]. Clients form the distributed ledger's endpoints that enable transaction issuance and the interaction with, for example, browser applications. Peer nodes endorse transactions and maintain replications of the distributed ledger. Orderer nodes generate new blocks, propagate them through the network of the distributed ledger, and participate in the consensus mechanism.

To interact with the HLF blockchain, every software component (i.e., clients, peer nodes, and orderer nodes) needs to first acquire a certificate to prove its identity from the consortium member's PKI (referred to as membership service provider in HLF) [67]. Within the PKI, certification authorities issue certificates to these software components based on their public keys. These certificates allow the verification of identities and their roles of software components of the distributed ledger and the PMS. For example, peer nodes use their private keys to digitally sign transactions. To validate the digital signature attached to each transaction, the PKI stores the peer nodes' public keys to which their identities and responsible consortium members are assigned. By doing so, the PKI enables the recognition of identities without revealing the members' private key.

HLF (v 1.4.1) offers the choice between three consensus mechanisms: *Solo*, *Kafka*, and *Raft*. In addition, there is the custom developed and Byzantine fault-tolerant consensus mechanism *BFT-SMaRt* applicable to HLF (v 1.3) [68]. To evaluate these consensus mechanisms for HLF regarding their performance characteristics (i.e., max. throughput and transaction latency) in changing configurations (i.e., number of peer nodes and transaction issuance rate; cf. Appendix A and Figure 3), we performed benchmarking using Hyperledger Caliper [69].

Solo is a centralized consensus mechanism with a single orderer node that sends new blocks to all peer nodes. Although Solo represents the fastest consensus mechanism among the HLF built-in consensus mechanisms [70] (cf. Figure 3 and Appendix A), Solo is not meant to be used productively because it is neither crash fault tolerant nor Byzantine fault tolerant [67]. In addition, Solo does not meet requirements for censorship resistance and comes with a single point of failure due to its centralized design, directly impeding availability of the overall PMS. Therefore, Solo is unsuitable for the PMS.

Kafka is a decentralized consensus mechanism that includes a cluster of Apache Kafka nodes in addition to the DLT nodes (i.e., orderer nodes and peer nodes) in HLF [71]. Order nodes retrieve data from Kafka nodes, which the Apache ZooKeeper nodes keep track of. In the Kafka cluster, a Kafka node, which is elected as the current cluster leader, initiates the replication of data among all Kafka nodes in the cluster. If the cluster leader is no longer available, the ZooKeeper ensemble elects a new leader [72]. Compared to other crash fault-tolerant consensus mechanisms (e.g., Paxos or Raft), Kafka has a significantly higher message complexity, which inhibits scalability regarding a large number of validating nodes [73]. Scaling the Kafka peer nodes from four to twelve decreases the transaction throughput from 200 tx/s to 50 tx/s. This decrease in throughput may form a bottleneck in the PMS (cf. Appendix A). Although different organizations independently operate the order nodes, one single organization controls the entire Kafka cluster and ZooKeeper ensemble [74]. All orderer nodes communicate with the same, centralized Kafka cluster. Therefore, the requirements for high availability and censorship resistance for the PMS (cf. Table 1 are not fulfilled.

Raft is a leader-based consensus mechanism, which has been widely applied due to its high performance [75]. In Raft, all orderer nodes are assigned to one of the three roles: *candidate*, *follower*, or *leader*. Initially, all orderer nodes are followers. If no leader exists over a certain period, a leader election is triggered and the followers change their role to candidate. The candidates vote for a new leader. One of the candidates becomes the new leader after a majority is achieved. The leader receives all transactions and forwards

them to the follower [76]. To keep its leader position, the leader periodically sends a heartbeat to the follower. When a follower times out after waiting for a heartbeat from the leader, the follower elects a new leader [77]. Our measurements reveal that Raft has a higher throughput and shorter transaction latency compared to Kafka. In addition, Raft has better scalability than Kafka and an increased number of Raft peer nodes leads to a lower decrease in throughput (cf. Appendix A). Raft is crash fault tolerant up to 50 % of the number of orderer nodes in a distributed ledger [78]. Nevertheless, Raft is vulnerable to malicious behavior of assigned consortium members (Byzantine failures), which may impede censorship resistance. The unsafe conditions happens when the current leader crashes and transactions are blocked in the committing queue until a new leader is chosen [79]. This kind of attack could impede censorship resistance or even lead to a denial of service. Due to these vulnerabilities, Raft does not fulfill the requirements for *availability*, *censorship resistance*, and *reliability* for PMSs (cf. Table 1).

BFT-SMaRt executes transactions similar to the practical Byzantine fault tolerance (PBFT) [65] but comes with improved reliability and higher scalability regarding the number of orderer nodes [80]. Similar to PBFT, clients trigger the execution of the consensus mechanism by sending a transaction to all orderer nodes. The leader (referred to as the primary leader) broadcasts a batch of transactions (e.g., a block), which should be appended to the distributed ledger and to its followers (referred to as secondary nodes). The followers reach consensus by voting on whether or not to append the transaction batch. To have a request successfully appended to the distributed ledger, more than two-thirds of the followers must reply that they appended the new request to their local replication of the ledger. BFT-SMaRt achieves a lower throughput and a better scalability than Raft (cf. Appendix A).

BFT-SMaRt achieves a lower average latency for an issued transaction to be committed than the other consensus mechanisms (cf. Figure 3). This difference in the average latency is mainly caused by the way blocks are stored: while blocks in BFT-SMaRt are stored in the random access memory (RAM), the other HLF-supported consensus mechanisms store blocks on the hard drive storage [81].

The security model of BFT-SMaRt requires a total number of followers n to tolerate $f < \frac{n}{3} - 1$ fraudulent followers f [68]. In the case of a fraudulent leader, a majority of the honest followers can vote on the legitimacy of the current leader and replace it with the another DLT node after a predefined period [68]. Nevertheless, the leader may drop certain requests [82], [83]. Because BFT-SMaRt is Byzantine fault tolerant and offers sufficient performance for the PMS, we find BFT-SMaRt best suitable for the PMS among the evaluated consensus mechanisms.

4) WORKFLOW

The proposed PMS requires each consortium member to set up their own computer running a client, a certification authority as part of the PKI, a peer node, an orderer node, and a publicly accessible API. Each client includes a Node.js server, which represents the access point for incoming requests (e.g., sensor messages to be processed). Moreover, the Node.js server hosts a certification authority and stores the public keys and roles of the consortium members' sensor nodes, orderer nodes, and peer nodes. When consortium members register a sensor node with the PMS, the certification authority generates a cryptographic key pair and a unique sensor node ID for the sensor node. The private key is stored exclusively on an encrypted storage of the sensor node. The public keys of all sensor nodes are assigned to the respective sensor node ID and stored by the PKI. To establish the communication between sensor nodes and the distributed ledger, each consortium member can use an individual LoRa network. Already existing public LPWANs can be used for wireless data transmission of sensor messages provided that digitally signed sensor messages are forwarded from public LPWAN servers to the distributed ledger.

To measure air pollution, consortium members place multiple sensor nodes in the environment of interest. Each sensor node detects its GPS coordinates, the relative humidity, the temperature, and the particulate matter. Each sensor node's MCU reads the data from its sensors and defines a universally unique identifier (UUID) for each measurement by using a random number generator with a length of 16 B according to recommendations of the Internet Engineering Task Force (IETF) [84]. Sensor nodes digitally sign the sensor data and the UUID to make data authenticity provable. Subsequently, the sensor node broadcasts a sensor message (including the sensor data, the measurement UUID, its digital signature of the sensor data, and its sensor node ID) to all adjacent LoRa gateways. Next, the sensor nodes switch into a power save mode before the next measurement is carried out. During the power save mode, the different sensors and the GPS module are turned off.

When the LoRa gateways receive a sensor message, the LoRa gateways broadcast the sensor message to the clients of the DLT nodes via TCP/IP. Subsequently, each client invokes a smart contract on the peer nodes to process the received sensor message. First, the sensor node ID is extracted from the incoming sensor message to query the public key of the sensor node from the PKI and to verify the digital signature. If the sensor node's public key is not registered with the PKI, the sent sensor message is rejected by the smart contract. Otherwise, the transaction is verified and the sensor node ID, the measurement UUID, and the sensor data are included into a new block appended the ledger.

To analyze stored sensor data, the PMS offers a publicly accessible API that can be integrated into various applications (e.g., a browser application). Via the API, functions of the smart contract deployed to the distributed ledger can be invoked to fetch all stored sensor data from the distributed ledger.

V. EVALUATION OF THE PROPOSED POLLUTION MONITORING SYSTEM

For the evaluation of the PMS, we assumed a consortium of four members each operating a validating node. We set up the PMS in an urban area using five sensor nodes, three LoRa gateways, four clients, four peer nodes, and four orderer nodes. The sensors nodes issued new messages every five minutes, including the sensor nodes' locations, the current timestamp, the temperature, the relative humidity, and pollutants (PM_{10}) at different locations. Similar to other field tests of mobile PMSs [85]–[87], we conducted a 24 h evaluation of our PMS. In the following, we discuss to which extent the PMS meets the requirements stated in Section II.

Accuracy. According to a common sensor calibration method [88], we evaluated the sensor nodes' accuracy by comparing our measurements with those from a stationary reference PMS. Therefore, we colocated a portable sensor node next to a government-run stationary reference PMS [89], which was equipped with calibrated air quality sensing instruments [90]. As illustrated in Figure 4, the pollutants (PM_{10}) recorded by the proposed PMS (solid line) closely align with the pollution data recorded by the stationary reference PMS (dash-dotted line) [90]. Compared to other air pollution monitoring studies [91], [92], our portable PMS achieved a low mean absolute error (MAE)[2] of 1.7 μg/m^3 during the 24 h evaluation. We observed the smallest deviations of the measured pollutants (PM_{10}) during late afternoon rush hour (3:00-6:00 *pm*; MAE = 0.2 μg/m^3) and a strong change in the sensor nodes' accuracy during night (3:00-6:00 *am*; MAE = 6.1 μg/m^3). The sensor nodes' measurement accuracy might be affected by air temperature and relative humidity changes [93] and can be further improved by applying a particle distribution–based correction algorithm (e.g., kappa-Köhler theory [94]) [95]. However, with regard to the low overall MAE, the PMS satisfies the accuracy requirement.

Availability. During the 24 h evaluation time, no communication errors occurred such as crashed or unreachable devices. Due to the high level of redundancy regarding the LoRa gateways and DLT nodes, the PMS reaches higher availability compared to centralized PMSs that prefer a lean bandwidth use over redundant sensor message broadcasting. Hence, we claim that the availability requirement of our PMS is fulfilled even for long monitoring duration.

Bandwidth. The maximum duty-cycle in LoRa represents the maximum percentage of time during which a device (e.g., sensor node) can occupy a channel [96]. The maximum duty-cycle of the EU 868 ISM band is 1 % per channel and results in a maximum total transmission time (referred to as air time) of 864 s/d per channel [96]. With the bandwidth of 250 kHz, spreading factor of nine, code rate of one and payload size of 121 B per transmission, the air time per transmission of the proposed PMS is approximately 328 ms [97].

[2]*Mean absolute error* describes the average deviation between two measurements and is defined as the sum of the absolute values of the residual divided by the total number of measurements.

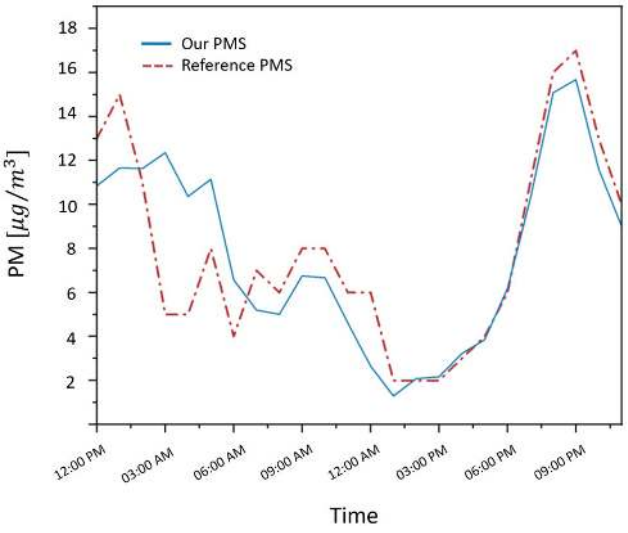

FIGURE 4. Comparison of PM_{10} exposure measured by the proposed PMS (red) and reference measurements by a stationary air quality station operated by the Bavarian State Office for the Environment (LfU) in Germany (dot-dashed line) [90]. All measurements were carried out at the same location in Augsburg (Germany).

According to the measurement interval of 5 min/msg per sensor node and an evaluation time of 24 h, the total air time of all five sensor nodes was approximately 472 s/d. Thus, the PMS fulfills the bandwidth requirement.

Censorship resistance. Since each consortium member self-operates an individual LoRa network, no third party is involved (e.g., The Things Network). All consortium members can set up and integrate their own gateways into the PMS network to assure censorship resistance. In addition, broadcasting sensor messages to all reachable gateways further increases censorship resistance. With an increasing number of independently operated DLT nodes, the PMS's censorship resistance increases. However, the number of independent consortium members is limited by the scalability of the BFT-SMaRt consensus mechanism. Therefore, the proposed PMS fulfills the requirement for censorship resistance regarding the network communication and achieves limited censorship resistance regarding the distributed ledger.

Energy consumption. Our energy consumption measurements reveal that the sensor node's average energy consumption is 60 mA/h. By disabling the permanently running GPS module, the total energy consumption of the sensor node could be reduced to 19 mA/h. Therefore, the GPS module is only turned ON once a day to synchronize the MCU clock and to check the sensor nodes location. Without permanent use of the GPS module, the sensor node would be powered for approximately 22 d using a battery capacity of 10 A h. We consider the requirement for energy consumption to be fulfilled.

Independence. The sensor nodes and LoRa gateways are replaceable by a wide range of other devices (e.g., Raspberry Pi). The LoRa protocol, the HLF blockchain, and the applied digital EdDSA signature algorithm are open source software

and free to use. By using these software frameworks and protocols and replaceable devices, we avoided dependencies from proprietary software and hardware. Therefore, the PMS fulfills the requirement for independence.

Integrity. Data integrity is provable from the time of sending the digital signed messages from the sensor node due to the use of digital signatures. If the smart contract on the distributed ledger validates that the sensor message has not been tampered with during its transmission, the sensor messages are appended to the distributed ledger. Thus, only if all gateways tamper with the sensor message of the same measurement are no new sensor messages appended to the distributed ledger. This scenario is unlikely because consortium members set up their own LoRa network and should have no incentive to tamper with their own data on the gateways. Hence, we find the PMS meets the requirement for data integrity.

Nonrepudiation. Each sensor node is identifiable and signs its grasped measurements with its unique private key [98]. Thus, all transactions stored on the distributed ledger can be unambiguously assigned to a sensor node. The private key is protected by various security checks of the ESP32 MCU, which makes it hard to leak the private key. In combination with the tamper resistance of the distributed ledger, it is hard to corrupt the PMS regarding nonrepudiation. Since the recomputation of EdDSA-generated private keys by attackers is still not feasible [99], the PMS fulfills the requirement for nonrepudiation.

Portability. All sensor nodes are battery powered and put into a $13 \times 7 \times 5$ cm^3 weather-resistant box (cf. Appendix B) to reduce their environmental impact restrictions. Consortium members can place registered sensor nodes anywhere without consideration of particular environmental conditions other than those specified by the manufacturer. The portability of the LoRa gateways is limited by their permanent power consumption of 60 W [100] and degree of weather resistance. However, compared to stationary PMSs, the LoRa gateways have very low energy consumption and installation effort and can be easily relocated. Therefore, the PMS meets requirements for portability.

Reliability. During the evaluation, no communication channel faults or outliers in the period between the issuance of sensor messages and their confirmation on the distributed ledger were detected. The PMS reliably recorded all 1,440 measurements, and all duplicate sensor messages have been filtered as intended. All stored data were publicly accessible using the API hosted by the individual consortium members. In the proposed PMS, a single gateway failure does not necessarily lead to data loss because sensor nodes redundantly broadcast their messages to all adjacent LoRa gateways within reach. BFT-SMaRt tolerates up to one-third of malicious orderer nodes among the total number of orderer nodes in the distributed ledger. Since it was shown that consensus mechanisms with probabilistic finality (e.g., Nakamoto consensus in Bitcoin) can even be compromised by a minority (e.g., 30 % of the overall hashing power [66]), we find the fault tolerance of BFT-SMaRt sufficient for the PMS with a small number of stakeholders compared to large distributed ledgers such as Bitcoin; moreover, consensus cannot be influenced by computational power such as in Bitcoin. Due to the ability of the PMS to deal with crashed or corrupted gateways and malicious consortium members, we find the PMS fulfills the requirements for reliability.

Scalability. We evaluated the scalability of the overall PMS from data collected at the sensor nodes to the storage of the data on the distributed ledger. We identified the gateways as a scalability bottleneck in the PMS that arises due to LoRa's low duty-cycle and the mesh network topology of sensor nodes and gateways. Changing the number of gateways has a stronger effect on the performance of the PMS than does changing the number of sensor or peer nodes because an increasing number of gateways also increases the number of duplicate sensor messages. In our field test, for example, five sensor nodes and three gateways generated 15 sensor messages to be processed by the distributed ledger and slightly decreased the distributed ledger's confirmation rate[3] to 99.40% (cf. Appendix A). The integration of twelve peer nodes in the PMS using the BFT-SMaRt consensus mechanism decreases the throughput from 150 tx/s to 100 tx/s, which shows better scalability than the alternative consensus mechanisms (cf. Appendix A). The overall capability of the PMS to process 100 tx/s would cover up to six different sensing regions,[4] simultaneously sending data to the distributed ledger. Since the PMS does not require real-time data (e.g., new measurements of single sensor nodes every second) and the workload should be processed within the determined measurement interval of five minutes, we consider the requirement for scalability of the PMS to be fulfilled.

Throughput. During the field test, the proposed PMS did not show performance bottlenecks. Using the available air time of 864 s/d for each public LoRa channel, each sensor node can even transmit a maximum of 21 msg/h. Each dual-channel gateway can process up to two sensor messages simultaneously [100], which allows for fast processing of queued sensor messages. Assuming that every gateway receives all five sensor messages simultaneously, 15 msg/s are forwarded to the distributed ledger for a single measurement interval. The evaluation of different consensus mechanisms (cf. Section IV-C3 and Appendix A) indicates that the distributed ledger in the presented PMS can reliably handle up to 150 tx/s incorporating four peer nodes and four orderer nodes. With a maximum throughput of 150 tx/s our PMS can theoretically process a maximum of 450 msg/s.[5] Hence,

[3] Ratio in percentage of issued transactions to max. throughput.

[4] Equal to our field test, each sensing region is equipped with three gateways and five sensor nodes.

[5] We assume that each sensor node transmits a maximum of three messages every second based on an air time of (0.328 s) for each message. Note that this assumption is a theoretical worst-case estimation and often larger measuring intervals are chosen (e.g., one measurement every 5 min).

we regard the PMS to have fulfilled the requirements of throughput.

Transparency. The presented PMS is based upon a private-permissioned distributed ledger, and stored sensor data are fully transparent for the consortium members who operate the distributed ledger. To make stored sensor data publicly accessible, the consortium members must individually host an API that allows the public to interact with the distributed ledger from outside the consortium [101]. The public keys and addresses of the consortium members are retrievable via the API and can be mapped to the real identities of the consortium members. The individual APIs of the consortium members may be subject to malicious behavior of consortium members. To decrease the impact of malicious behavior of individual consortium members, users, which are not part of the consortium, should request all APIs of all consortium members and compare the retrieved sensor data for inconsistencies. By doing so, corrupted APIs should stand out. Since we decided to prioritize low resource consumption over full openness of the PMS, the requirement for transparency is only partially fulfilled because outsiders cannot directly access the distributed ledger.

VI. COMPARISON WITH RELATED WORK

Existing research on IoT applications (including PMSs incorporating portable, low-energy sensor nodes) has shown a special interest on achieving an appropriate equilibrium regarding the trade-off between performance and security (e.g., [102]–[106]). To find such an equilibrium for PMSs, various software designs using the LoRa communication protocol have been proposed (e.g., [30], [103], [107]) that comprise the design of IoT applications from the sensor node to the data storage. The integration of battery powered, low-energy sensor nodes into DLT was considered in LoRaWAN using digital signatures and public key cryptography [108]. During an initial enrollment process, a certification authority issues certificates to each sensor node of the network. To verify the identities of the sensor nodes, the nodes send their certificates to a single network server, which represents a centralized certification authority. The network server needs to verify the identity of each sensor node before the network server can issue a transaction to a private distributed ledger [108]. However, the network server forms a single point of failure and might impact the availability of the infrastructure. The certification authority in the PMS proposed in this work is redundant and thus overcomes this challenge and fulfills the requirements for availability (cf. Section II).

Compared to portable sensor nodes, LoRa gateways are mostly power socket-operated and thus have sufficient computational resources to be directly integrated as (validating) nodes in a distributed ledger [30], [107]. The use of standard LoRa gateways as a light client of a distributed ledger (e.g., Ethereum Light Client) allows a simple integration of different sensor nodes into IoT applications. The compatibility with existing LoRaWAN devices is maintained because no changes of the LoRa communication protocol are required. Nevertheless, it is likely that LoRa gateways will become targets of cyberattacks because they act as a bridge between IoT devices and the distributed ledger and might not hold the requirement for censorship resistance (cf. Section II). In the PMS proposed in this work, LoRa gateways are not part of the distributed ledger and are only used to forward digitally signed transactions. Even if a single LoRa gateway is corrupted (e.g., temporally switched off), sensor messages can still be transmitted to the distributed ledger because sensor messages are broadcast simultaneously to multiple LoRa gateways to avoid a single point of failure and meet the requirement for censorship resistance.

For secure data transmission from sensor nodes through LoRa gateways to a distributed ledger, various studies used LoRaWAN [109], [110]. Concepts of integrating DLT into LoRaWAN have been developed to achieve secure and decentralized public networks [111]–[113]. Instead of relying on a LoRaWAN operated by third-party providers, passive roaming techniques were used to create a fully decentralized LoRaWAN [112]. To enable roaming agreements between different network and application servers in a decentralized and open way, smart contracts were used. These smart contracts acted as a domain name service for gateways and ran on a public-permissionless Ethereum blockchain [112]. The execution of smart contracts on a public-permissionless distributed ledger (e.g., Ethereum) is subject to a pricing scheme that requires users to pay for the smart contract execution in proportion to the computing resources allocated to the execution. The execution cost required to execute smart contract functions (e.g., join-request) is highly dependent on current demand (market price). The extremely volatile [114] and generally high prices of popular cryptocurrencies (e.g., Bitcoin or Ether) challenge the cost-efficient design of DLT-based systems [115]. The PMS proposed in this work does not require users to pay for the sensor node registration or smart contract execution and represents a cost-efficient DLT-based system.

VII. CONCLUSION

In this work, we present the design and implementation of a PMS using portable low-energy sensor nodes, the LoRa protocol, and an HLF blockchain. The PMS is characterized by an energy-efficient and secure end-to-end data transfer between portable sensor nodes and a distributed ledger. The evaluation shows that the proposed PMS design effectively works and that DLT is applicable to the collection of environmental data. We showed that DLT can be employed as a shared, decentralized infrastructure among consortium members in the field of environment analysis to overcome the prevalent challenges regarding scarcity and validity of environmental data and to reduce the inconsistency of the evidence about air pollution.

During the design process of the PMS, we realized that HLF (v 1.4.1) only provides crash fault-tolerant consensus mechanisms (i.e., Solo, Kafka, and Raft), in contrast to our definition of DLT that requires Byzantine fault tolerance. Although the need for Byzantine fault tolerance in private

distributed ledgers is often considered to be of no particular importance (e.g., [116]–[118]), we present a use case for DLT that specifically requires fraud resistance through Byzantine fault tolerance in this paper.

Even though Byzantine fault tolerance should be an inherent characteristic of distributed ledgers [17], we found only few Byzantine fault tolerant consensus mechanisms applicable to HLF (e.g., BFT-SMaRt [119], [120] or PBFT [65], [121]). In addition to the security model, we determined that the implementations of the different consensus mechanisms strongly differ regarding their performance (cf. Appendix A). Surprisingly, the Byzantine fault-tolerant consensus mechanism BFT-SMaRt outperformed the HLF-inherent consensus mechanisms, which is likely due to the different concept underlying the storage of blocks. Beyond the consensus mechanism, the HLF architecture is not fully optimized and currently does not use internal-memory data structures whose lack of durability guarantees can be compensated by the blockchain itself [122].

We found that the PMS sufficiently scales up to twelve validating nodes with a transaction issuance rate of 500 tx/s. To use the PMS in larger consortia (or in multiple, interconnected consortia), sharding [123], [124] or the use of a public-permissioned distributed ledger (e.g., Steem) should be considered. The use of a public-permissionless distributed ledger would increase censorship resistance and transparency but requires self-implementing identity management, which is still a prevalent challenge in DLT (e.g., [125], [126]). Nevertheless, benchmarking these different types of distributed ledgers would reveal insights into the achievement of the identified requirements. Although the measurement period of 24 h in the field test reflects commonly used measurement periods of PMSs that incorporate portable sensor nodes (e.g., [87], [127], [128]), the evaluation is not representative for long-term use of PMSs.

More research needs to be carried out to investigate the likelihood for potential attacks and the impact of network delays on the entire PMS performance. In this context, different broadcasting strategies for sensor messages should be investigated to find a Pareto optimum between the number of deployed gateways and sufficient redundancy to achieve high reliability in sensor message transmission. In addition, more work needs to be carried out to define the local air pollution probability in order to adjust the measurement interval of the sensor nodes and to maximize their battery lifetime. This work points out the particular importance of Byzantine fault tolerance in DLT, and future research should emphasize the development of robust Byzantine fault-tolerant consensus mechanisms applicable to private distributed ledgers. From a hardware perspective, the investigation of integrated circuits for digital signatures of low-energy sensors should be of particular interest in order to decrease energy consumption and to increase fraud resistance at the sensor node endpoint.

We believe that low-cost accessible sensor networks and distributed database systems need to evolve together to increase benefits for their users beyond closed measurement infrastructures. Our evaluation results indicate that developers must consider constraints of wireless networks (e.g., available air time) to successfully integrate battery-powered sensor nodes into distributed ledgers. We contribute to research and practice by presenting results of a comprehensive requirements analysis of existing PMSs incorporating DLT and low-energy devices that help to design and assess PMSs. The detailed evaluation of alternative approaches for network protocols, consensus mechanisms, and digital signature algorithms helps the development of tamper-resistant and transparent PMSs and similar IoT applications (e.g., shipping container tracking and monitoring).

This work follows calls from extant research (e.g., [129]–[133]) regarding environmental data collection by increasing stakeholder engagement in the data collection and serves as a guide to facilitate new participatory research designs. Our evaluation of different consensus mechanisms, digital signature schemes, and the proposed design for a decentralized PMS that includes low-energy sensor nodes helps to resolve the trade-off between performance and security in the IoT field.

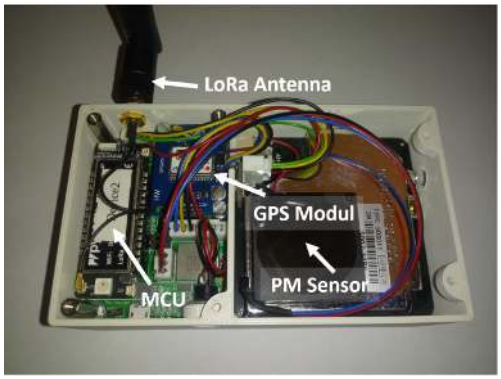

FIGURE 5. Portable sensor node composed of a particulate matter (PM_{10}) sensor (Nova SDS011), a humidity and temperature sensor (Grove DHT22), a GPS module (Ublox Neo-6M), a microcontroller unit (ESP32), a rechargeable lithium polymer battery, and an external LoRa antenna.

APPENDIX

APPENDIX A PERFORMANCE OF DIFFERENT CONSENSUS MECHANISMS IN HYPERLEDGER FABRIC

Performance of different consensus mechanisms applied to the Hyperledger Fabric blockchain with four peer nodes for varying transaction issuance rates, different consensus mechanisms (i.e., Solo, Kafka, Raft, and BFT-SMaRt), and a varying number of peer nodes (i.e., 4, 8, and 12; except for the centralized Solo).

See Table 4.

APPENDIX B DESIGNED SENSOR NODE

We designed a portable sensor node composed of a particulate matter (PM_{10}) sensor (Nova SDS011), a humidity and temperature sensor (Grove DHT22), a GPS module (Ublox Neo-6M), a microcontroller unit (ESP32), a rechargeable lithium polymer battery, and an external LoRa antenna.

See Figure 5.

TABLE 4. Performance of different consensus mechanisms applied to the Hyperledger Fabric blockchain with four peer nodes for varying transaction issuance rates, different consensus mechanisms, and a varying number of peer nodes.

	Solo			Kafka			Raft			BFT-SMaRt			Transaction issuance rate
Criterion	4	8	12	4	8	12	4	8	12	4	8	12	
Issued transactions	10.10	10.10	10.10	10.10	10.10	10.01	10.10	10.10	10.10	10.40	10.40	10.40	10
Max. throughput	10.00	10.01	10.01	10.01	10.00	10.00	9.90	10.00	10.00	10.10	10.3	10.30	
Avg. latency	0.34	0.35	0.30	0.35	0.39	0.41	0.35	0.36	0.37	0.77	0.51	0.78	
Max. latency	0.56	0.59	0.64	0.59	0.62	0.70	0.67	0.60	0.61	2.28	1.80	2.30	
Issued transactions	50.10	50.10	50.10	50.01	50.01	50.10	50.10	50.00	50.10	50.40	50.40	50.40	50
Max. throughput	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.10	50.10	
Avg. latency	0.12	0.13	0.14	0.14	0.16	0.18	0.13	0.14	0.15	0.15	0.15	0.16	
Max. latency	0.22	0.25	0.26	0.48	0.45	0.42	0.28	0.29	0.24	0.28	0.25	0.28	
Issued transactions	100.10	100.10	99.80	100.10	100.10	100.10	100.10	100.10	100.00	100.40	100.40	100.40	100
Max. throughput	100.00	100.00	100.00	99.90	99.90	96.90	100.00	100.00	99.90	100.10	99.90	99.70	
Avg. latency	0.08	0.09	0.12	0.11	0.13	1.68	0.10	0.10	0.11	0.11	0.10	0.15	
Max. latency	0.14	0.27	0.44	0.47	0.42	3.93	0.16	0.20	0.22	0.24	0.19	0.34	
Issued transactions	150.10	150.10	150	150.10	150.10	150.10	150.1	150.1	145.21	150.30	150.30	150.30	150
Max. throughput	150.00	149.90	149.69	149.69	117.60	90.41	149.99	144.47	149.59	99.53	146.80	125.31	
Avg. latency	0.07	0.08	4.51	0.12	8.73	31.58	0.07	0.09	1.66	0.18	0.28	1.76	
Max. latency	0.31	0.26	16.25	0.71	21.24	42.86	0.15	0.28	4.10	0.43	0.69	3.01	
Issued transactions	200.10	200.10	200.10	200.10	200.10	200.00	200.00	200.10	200.10	200.30	200.30	200.40	200
Max. throughput	199.89	199.80	137.91	198.70	114.20	85.84	199.90	127.48	128.20	182.40	163.91	145.39	
Avg. latency	0.06	0.10	18.19	0.61	30.57	54.63	0.08	2.41	25.56	0.69	1.92	3.47	
Max. latency	0.16	0.26	36.90	0.97	42.33	68.99	0.42	4.73	37.55	1.82	3.31	5.03	
Issued transactions	249.2	250.10	250.10	250.10	250.10	250.10	250.1	248.30	250.00	250.40	250.30	250.30	250
Max. throughput	249.71	185.09	139.01	169.99	118.70	92.71	249.80	140.91	134.95	185.40	173.90	149.99	
Avg. latency	0.07	10.55	26.31	19.75	45.56	71.00	0.09	23.11	42.34	3.77	4.45	6.60	
Max. latency	0.22	22.16	49.29	30.69	56.26	88.91	0.63	35.09	54.34	5.30	6.30	7.83	
Issued transactions	300.10	298.00	300.10	300	300	300	300.00	300.10	300.10	300.20	300.20	300.20	300
Max. throughput	299.71	185.09	137.30	155.70	121.59	63.51	299.70	148.25	139.95	198.19	178.71	167.39	
Avg. latency	0.09	26.11	34.35	38.01	59.67	78.20	0.41	34.88	52.42	5.50	6.72	6.92	
Max. latency	0.29	37.08	58.20	48.63	74.83	100	1.05	46.13	77.70	7.38	9.54	9.46	
Issued transactions	350.00	350.00	350.10	350	350.10	350.10	350.00	350.0	348.80	350.20	350.30	350.10	350
Max. throughput	264.50	196.70	122.18	166.60	124.99	39.11	229.50	150.15	136.31	203.99	201.11	172.11	
Avg. latency	9.02	36.67	40.85	46.21	73.88	82.59	18.48	46.71	65.25	7.45	6.94	8.46	
Max. latency	21.78	48.97	68.14	58.19	92.50	100	28.84	63.06	85.72	10.78	9.48	15.32	
Issued transactions	400.00	400.00	400.1	400	400.10	397.50	396.20	391.4	400.00	400.10	400.20	400.20	400
Max. throughput	239.32	192.52	129.19	169.08	127.99	30.01	236.81	167.91	140.20	219.29	208.58	179.49	
Avg. latency	25.30	46.14	43.53	43.91	63.45	87.97	18.48	58.92	57.05	8.16	8.43	10.16	
Max. latency	35.73	65.88	59.25	54.51	81.01	100	28.84	76.67	85.19	11.26	12.35	13.88	
Issued transactions	450.00	447.30	437.10	450	449.20	414.1	444.90	443.5	449.20	450.10	450.20	448.2	450
Max. throughput	245.30	181.29	130.39	161.10	147.57	39.59	218.00	173.99	141.00	229.28	210.20	181.40	
Avg. latency	34.14	54.19	46.05	45.59	63.99	88.57	30.74	67.50	57.05	9.24	10.45	12.14	
Max. latency	46.12	75.10	69.01	60.05	83.29	100.33	42.28	88.19	85.19	12.79	13.79	18.05	
Issued transactions	499.99	494.40	499.2	499.90	498.10	384.8	500.10	486.20	456.6	500.2	500.1	500.3	500
Max. throughput	235.07	182.19	130.89	166.92	127.91	35.71	223.39	173.62	142.09	226.89	210.29	179.81	
Avg. latency	44.04	60.67	46.01	47.13	67.44	88.57	49.92	76.53	58.83	10.98	11.87	13.85	
Max. latency	57.41	82.54	61.86	60.43	89.93	100.30	68.56	99.38	82.40	13.94	15.51	19.08	

Issued transactions	The rate at which Hyperledger Caliper issued the transactions [tx/s].	*Max. throughput*	The maximum number of successfully processed transactions per second [tx/s].
Avg. latency	The average time span between the issuance of a transaction by a DLT node and its appending to the ledger [s].	*Max. latency*	The maximum time span between the issuance of a transaction by a DLT node and its appending to the ledger [s].

We carried out the measurements for Solo, Kafka, and Raft using Hyperledger Fabric (v 1.4.1). For BFT-SMaRt, we used Hyperledger Fabric (v 1.3) due to constrained compatibility. All measurements were carried out in Docker containers using the Hyperledger Caliper framework. For all measurements, we used an Intel(R) Core(TM) i7-8700 with a 3.20 GHz CPU and 16 GB memory.

ACKNOWLEDGMENT

The authors thank C. Fries and R. Lamberti for supporting the evaluation of the different consensus mechanisms, S. Essig for his contribution to the hardware design of the sensor nodes, and B. Sturm and T. Dehling for their friendly review.

REFERENCES

[1] B. R. Gurjar, A. Jain, A. Sharma, A. Agarwal, P. Gupta, A. S. Nagpure, and J. Lelieveld, "Human health risks in megacities due to air pollution," *Atmos. Environ.*, vol. 44, no. 36, pp. 4606–4613, Nov. 2010.

[2] Z. Idrees and L. Zheng, "Low cost air pollution monitoring systems: A review of protocols and enabling technologies," *J. Ind. Inf. Integr.*, vol. 17, Dec. 2020, Art. no. 100123, doi: 10.1016/j.jii.2019.100123.

[3] D. Ghanem and J. Zhang, "'Effortless perfection': Do Chinese cities manipulate air pollution data?" *J. Environ. Econ. Manage.*, vol. 68, no. 2, pp. 203–225, Sep. 2014.

[4] Y. Chen, G. Z. Jin, N. Kumar, and G. Shi, "Gaming in air pollution data? Lessons from China," *B.E. J. Econ. Anal. Policy*, vol. 12, no. 3, pp. 1–43, Jan. 2012.

[5] C. I. Beattie, J. W. S. Longhurst, and N. K. Woodfield, "Air quality management: Evolution of policy and practice in the UK as exemplified by the experience of English local government," *Atmos. Environ.*, vol. 35, no. 8, pp. 1479–1490, Jan. 2001.

[6] L. Zhang, A. Mol, and G. He, "Transparency and information disclosure in China's environmental governance," *Current Opinion Environ. Sustainability*, vol. 18, pp. 17–24, Feb. 2016, doi: 10.1016/j.cosust.2015.03.009.

[7] J. Brainard, "Publishers try out alternative pathways to open access," *Science*, vol. 367, no. 6483, p. 1179, 2020.

[8] E. G. Snyder, T. H. Watkins, P. A. Solomon, E. D. Thoma, R. W. Wil-liams, G. S. W. Hagler, D. Shelow, D. A. Hindin, V. J. Kilaru, and P. W. Preuss, "The changing paradigm of air pollution monitoring," *Environ. Sci. Technol.*, vol. 47, no. 20, pp. 11369–11377, 2013.

[9] D. Randall. (2018). The Irreproducibility Crisis of Modern Science—Causes, Consequences and the Road to Reform. National Associatoin of Scholars. [Online]. Available: https://www.nas.org/storage/app/media/Reports/IrreproducibilityCrisisReport/NAS_irreproducibilityReport.pdf

[10] W. Yi, K. Lo, T. Mak, K. Leung, Y. Leung, and M. Meng, "A survey of wireless sensor network based air pollution monitoring systems," *Sensors*, vol. 15, no. 12, pp. 31392–31427, Dec. 2015.

[11] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proc. 19th SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2013, pp. 1436–1444. [Online]. Available: https://www.microsoft.com/en-us/research/publication/u-air-when-urban-air-quality-inference-meets-big-data/

[12] S. Steinle, S. Reis, and C. E. Sabel, "Quantifying human exposure to air pollution—Moving from static monitoring to spatio-temporally resolved personal exposure assessment," *Sci. Total Environ.*, vol. 443, pp. 184–193, Jan. 2013.

[13] A. Mukherjee, S. G. Brown, M. C. McCarthy, N. R. Pavlovic, L. G. Stanton, J. L. Snyder, S. D'Andrea, and H. R. Hafner, "Measuring spatial and temporal PM2.5 variations in sacramento, california, communities using a network of low-cost sensors," *Sensors*, vol. 19, no. 21, p. 4701, Oct. 2019.

[14] A. Gałuszka, Z. M. Migaszewski, and J. Namieśnik, "Moving your laboratories to the field–advantages and limitations of the use of field portable instruments in environmental sample analysis," *Environ. Res.*, vol. 140, pp. 593–603, Jul. 2015.

[15] K. Hu, V. Sivaraman, B. G. Luxan, and A. Rahman, "Design and evaluation of a metropolitan air pollution sensing system," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1448–1459, Mar. 2016.

[16] A. Sunyaev, "Distributed ledger technology," in *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, A. Sunyaev, Ed. Cham, Switzerland: Springer, 2020, pp. 265–299.

[17] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–37, Jul. 2020.

[18] A. Roehrs, C. A. da Costa, and R. da Rosa Righi, "OmniPHR: A distributed architecture model to integrate personal health records," *J. Biomed. Informat.*, vol. 71, pp. 70–81, Jul. 2017, doi: 10.1016/j.jbi.2017.05.012.

[19] A. G. Abbasi and Z. Khan, "VeidBlock: Verifiable identity using blockchain and ledger in a software defined network," in *Proc. Companion 10th Int. Conf. Utility Cloud Comput.*, 2017, pp. 173–179.

[20] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," in *Proc. Cloud Comput. Secur. Workshop (CCSW)*, 2017, pp. 45–50.

[21] H. G. Do and W. K. Ng, "Blockchain-based system for secure data storage with private keyword search," in *Proc. IEEE World Congr. Services (SERVICES)*, Jun. 2017, pp. 90–93.

[22] G. O. Karame, "On the security and scalability of bitcoin's blockchain," in *Proc. ACM Conf. Comput. Commun. Secur.*, vols. 24–28, 2016, pp. 1861–1862.

[23] J. Sedlmeir, H. U. Buhl, G. Fridgen, and R. Keller, "The energy consumption of blockchain technology: Beyond myth," *Bus. Inf. Syst. Eng.*, pp. 1–10, Jun. 2020, doi: 10.1007/s12599-020-00656-x.

[24] P. J. Sallis, *Wireless Sensor Networks—Insights and Innovations*. Rijeka, Croatia: InTech, 2017.

[25] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584908001390

[26] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future," *MIS Quart.*, vol. 26, no. 2, pp. 8–23, 2002.

[27] J. M. Corbin and A. L. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 4th ed. Newbury Park, CA, USA: Sage, 2015.

[28] A. Durand, P. Gremaud, and J. Pasquier, "Resilient, crowd-sourced LPWAN infrastructure using blockchain," in *Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst.*, 2018, pp. 25–29.

[29] D.-H. Shih, P.-Y. Shih, and T.-W. Wu, "An infrastructure of multi-pollutant air quality deterioration early warning system in spark platform," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2018, pp. 648–652.

[30] S. R. Niya, S. S. Jha, T. Bocek, and B. Stiller, "Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and LoRaWAN," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–4.

[31] O. Attia, I. Khoufi, A. Laouiti, and C. Adjih, "An IoT-blockchain architecture based on hyperledger framework for healthcare monitoring application," in *Proc. 10th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jun. 2019, pp. 1–5.

[32] M. Pavani and P. Rao, "Urban air pollution monitoring using wireless sensor networks: A comprehensive review," *Int. J. Commun. Netw. Inf. Secur.*, vol. 9, no. 3, pp. 439–449, 2017.

[33] I. D. Buldin, M. G. Gorodnichev, S. S. Makhrov, and E. N. Denisova, "Next generation industrial blockchain-based wireless sensor networks," in *Proc. Wave Electron. Appl. Inf. Telecommun. Syst. (WECONF)*, Nov. 2018, pp. 1–5.

[34] Y. M. Yussoff, H. Hashim, and M. D. Baba, "Identity-based trusted authentication in wireless sensor networks," *Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 230–239, 2012.

[35] O. Lamtzidis and J. Gialelis, "An IOTA based distributed sensor node system," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Jun. 2018, pp. 1–6.

[36] F. J. Kelly, G. W. Fuller, H. A. Walton, and J. C. Fussell, "Monitoring air pollution: Use of early warning systems for public health: Monitoring and communicating air quality," *Respirology*, vol. 17, no. 1, pp. 7–19, Jan. 2012.

[37] H. Cheng, R. Guo, and Y. Chen, "Node selection algorithms with data accuracy guarantee in service-oriented wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 4, Apr. 2013, Art. no. 527965.

[38] M. T. Hammi, P. Bellot, and A. Serhrouchni, "BCTrust: A decentralized authentication blockchain-based mechanism," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2018, pp. 1–6.

[39] X. Zhao, S. Zuo, R. Ghannam, Q. H. Abbasi, and H. Heidari, "Design and implementation of portable sensory system for air pollution monitoring monitoring," in *Proc. IEEE Asia Pacific Conf. Postgraduate Res. Microelectron. Electron. (PrimeAsia)*, Oct. 2018, pp. 47–50.

[40] R. T. Tse and Y. Xiao, "A portable wireless sensor network system for real-time environmental monitoring," in *Proc. IEEE 17th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2016, pp. 1–6.

[41] S. Ibba, A. Pinna, M. Seu, and F. E. Pani, "CitySense: Blockchain-oriented smart cities," in *Proc. XP Sci. Workshops*, 2017, pp. 1–5.

[42] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Boston, MA, USA: Springer, 1983, pp. 199–203.

[43] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 5th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2010.

[44] European Environment Activity. (2019). *Exceedance of Air Quality Standards in Urban Areas*. [Online]. Available: https://www.eea.europa.eu/data-and-maps/indicators/exceedance-of-air-quality-limit-3/assessment-5

[45] B. S. Sarjerao and A. Prakasarao, "A low cost smart pollution measurement system using REST API and ESP32," in *Proc. 3rd Int. Conf. for Converg. Technol. (I2CT)*, Apr. 2018, pp. 1–5.
[46] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things," in *Proc. Internet Technol. Appl. (ITA)*, Sep. 2017, pp. 143–148.
[47] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. Skarmeta, "Performance evaluation of LoRa considering scenario conditions," *Sensors*, vol. 18, no. 3, p. 772, Mar. 2018.
[48] B. Pearson, L. Luo, C. Zou, J. Crain, Y. Jin, and X. Fu, "Building a low-cost and state-of-the-art IoT security hands-on laboratory," in *Internet of Things. A Confluence of Many Discipline*. Cham, Switzerland: Springer, 2020, pp. 289–306.
[49] K. Sovani. (2018). *No Title*. [Online]. Available: https://medium.com/the-esp-journal/understanding-esp32s-security-features-14483e465724
[50] *Secure Boot*. Accessed: Apr. 9, 2020. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/security/secure-boot-v1.html
[51] L. H. Adnan, Y. M. Yussoff, and H. Hashim, "Secure boot process for wireless sensor node," in *Proc. Int. Conf. Comput. Appl. Ind. Electron.*, Dec. 2010, pp. 646–649.
[52] Y. Song, J. Lin, M. Tang, and S. Dong, "An Internet of energy things based on wireless LPWAN," *Engineering*, vol. 3, no. 4, pp. 460–466, Aug. 2017.
[53] S. Martiradonna, G. Piro, and G. Boggia, "On the evaluation of the NB-IoT random access procedure in monitoring infrastructures," *Sensors*, vol. 19, no. 14, pp. 1–25, 2019.
[54] A. Khalifeh, K. A. Aldahdouh, K. A. Darabkh, and W. Al-Sit, "A survey of 5G emerging wireless technologies featuring LoRaWAN, sigfox, NB-IoT and LTE-M," in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw. (WiSPNET)*, Mar. 2019, pp. 561–566.
[55] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Exp.*, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.
[56] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, Sep. 2016.
[57] Y. J. Jung, Y. K. Lee, D. G. Lee, K. H. Ryu, and S. Nittel, "Air pollution monitoring system based on geosensor network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2008, vol. 3, no. 1, p. III-1370.
[58] L. J. Chen, Y. H. Ho, H. C. Lee, H. C. Wu, H. M. Liu, H. H. Hsieh, Y. T. Huang, and S. C. C. Lung, "An open framework for participatory PM2.5 monitoring in smart cities," *IEEE Access*, vol. 5, pp. 14441–14454, Jul. 2017.
[59] W. Chen, L. Yan, and H. Zhao, "Seasonal variations of atmospheric pollution and air quality in Beijing," *Atmosphere*, vol. 6, no. 11, pp. 1753–1770, Nov. 2015.
[60] *LoRa Alliance: LoRaWAN Specification V1.1*. Accessed: Apr. 5, 2020. [Online]. Available: https://lora-alliance.org/resource-hub/lorawantm-specificationv11
[61] K. L. Tsai, Y. L. Huang, F. Y. Leu, I. You, Y. L. Huang, and C. H. Tsai, "AES-128 based secure low power communication for LoRaWAN IoT environments," *IEEE Access*, vol. 6, pp. 45325–45334, 2018.
[62] S. M. Danish, M. Lestas, W. Asif, H. K. Qureshi, and M. Rajarajan, "A lightweight blockchain based two factor authentication mechanism for LoRaWAN join procedure," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.
[63] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.
[64] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 906–917, doi: 10.1145/2382196.2382292.
[65] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Oper. Syst. Design Implement. (OSDI)*. Berkeley, CA, USA: USENIX Association, 1999, pp. 173–186.
[66] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 3–16, doi: 10.1145/2976749.2978341.
[67] Hyperledger Foundation. (2018). *A Blockchain Platform for the Enterprise*. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/
[68] J. Sousa, A. Bessani, and M. Vukolic, "A Byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, no. 1, 2018, pp. 51–58.
[69] Hyperledger. *Hyperledger Blockchain Performance Metrics*. Accessed: May 4, 2020. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/10/HL_Whitepaper_Metrics_PDF_V1.01.pdf
[70] L. Zanzi, A. Albanese, V. Sciancalepore, and X. Costa-p, "NSBchain: A secure blockchain framework for network slicing brokerage," 2020, *arXiv:2003.07748*. [Online]. Available: https://arxiv.org/abs/2003.07748
[71] *Introduction Kafka—A Distributed Streaming Platform*. Accessed: Apr. 6, 2020. [Online]. Available: https://kafka.apache.org/intro.html
[72] K. Christidis. (2016). *A Kafka-Based Ordering Service for Fabric*. [Online]. Available: https://docs.google.com/document/d/19JihmW-8blTzN99lAubOfseLUZqdrB6sBR0HsRgCAnY/edit
[73] R. Estrada, *Apache Kafka 1.0 Cookbook: Over 100 Practical Recipes on Using Distributed Enterprise Messaging to Handle Real-Time Data*. Birmingham, U.K.: Packt, 2017.
[74] *The Ordering Service*. Accessed: Apr. 4, 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.0/orderer
[75] K. Wang, M. Liu, X. Jiang, C. Yang, and H. Zhang, "A novel vehicle blockchain model based on hyperledger fabric for vehicle supply chain management," in *Blockchain and Trustworthy Systems*, Z. Zheng, H.-N. Dai, M. Tang, and X. Chen, Eds. Singapore: Springer, 2020, pp. 732–739.
[76] G. Carrara, L. Burle, D. Medeiros, C. Albuquerque, and D. Menezes, "Consistency, availability, and partition tolerance in blockchain: A survey on the consensus mechanism over peer-to-peer networking," in *Proc. Ann. Telecommun.*, 2020, pp. 1–12.
[77] V. Arora, T. Mittal, D. Agrawal, A. E. Abbadi, X. Xue, and Z. Zhiyanan, "Leader or majority: Why have one when you can have both? Improving read scalability in raft-like consensus protocols," in *Proc. 9th USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2017, p. 6. [Online]. Available: https://www.usenix.org/conference/hotcloud17/program/presentation/arora
[78] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2019, pp. 305–319.
[79] G. Zhang and C.-Z. Xu, "An efficient consensus protocol for real-time permissioned blockchains under non-Byzantine conditions," in *Proc. 14th Int. Conf. Green, Pervas. Cloud Comput.*, 2019, pp. 298–311.
[80] A. Bessani, J. Sousa, and E. E. Alchieri, "State machine replication for the masses with BFT-SMART," *Proc. 44th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, 2014, pp. 355–362.
[81] *Byzantine Fault-Tolerant (BFT) State Machine Replication (SMaRt) v1.2*. Accessed: May 5, 2020. [Online]. Available: https://github.com/bft-smart/library
[82] S. Rüsch, K. Bleeke, and R. Kapitza, "BLOXY: Providing transparent and generic BFT-based ordering services for blockchains," in *Proc. Int. Symp. Reliable Distrib. Syst.*, 2019, p. 30509. [Online]. Available: https://www.cs.tu-bs.de/users/ruesch/papers/srds19-bloxy.pdf
[83] S. He, Q. Tang, C. Q. Wu, and X. Shen, "Decentralizing IoT management systems using blockchain for censorship resistance," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 715–727, Jan. 2020.
[84] (2005). *IETF*. [Online]. Available: https://tools.ietf.org/html/rfc4122
[85] P. Zappi, E. Bales, J. H. Park, W. Griswold, and T. Šimuni, "The CitiSense air quality monitoring mobile sensor node," in *Proc. 11th ACM/IEEE Conf. Inf. Process. Sensor Netw.*, 2012, pp. 23–24.
[86] K. K. Johnson, M. H. Bergin, A. G. Russell, and G. S. W. Hagler, "Field test of several low-cost particulate matter sensors in high and low concentration urban environments," *Aerosol Air Qual. Res.*, vol. 18, no. 3, pp. 565–578, 2018.
[87] G. Ramachandran, J. L. Adgate, G. C. Pratt, and K. Sexton, "Characterizing indoor and outdoor 15 minute average PM 2.5 concentrations in urban neighborhoods," *Aerosol Sci. Technol.*, vol. 37, no. 1, pp. 33–45, Jan. 2003.
[88] D. H. Hagan, G. Isaacman-VanWertz, J. P. Franklin, L. M. M. Wallace, B. D. Kocar, C. L. Heald, and J. H. Kroll, "Calibration and assessment of electrochemical air quality sensors by co-location with regulatory-grade instruments," *Atmos. Meas. Techn.*, vol. 11, no. 1, pp. 315–328, Jan. 2018.
[89] M. Pitz, W. Birmili, O. Schmid, A. Peters, H. E. Wichmann, and J. Cyrys, "Quality control and quality assurance for particle size distribution measurements at an urban monitoring station in Augsburg, Germany," *J. Environ. Monitoring*, vol. 10, no. 9, pp. 1017–1024, 2008.

[90] Landesamt. *Messstation Augsburg Königsplatz*. [Online]. Available: https://www.lfu.bayern.de/luft/immissionsmessungen/doc/lueb_dokumentation/aktiv/07_Schwaben/03_augsburg_koenigsplatz.pdf
[91] E. Lagerspetz, S. Tarkoma, T. Hussein, N. H. Motlagh, M. A. Zaidan, P. L. Fung, J. Mineraud, S. Varjonen, M. Siekkinen, P. Nurmi, and Y. Matsumi, "MegaSense: Feasibility of low-cost sensors for pollution hot-spot detection," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2019, pp. 1083–1090.
[92] M. Penza, D. Suriano, V. Pfister, M. Prato, and G. Cassano, "Urban air quality monitoring with networked low-cost sensor-systems," *Proceedings*, vol. 1, no. 4, p. 573, Aug. 2017.
[93] A. P. K. Tai, L. J. Mickley, and D. J. Jacob, "Correlations between fine particulate matter (PM2.5) and meteorological variables in the United States: Implications for the sensitivity of PM2.5 to climate change," *Atmos. Environ.*, vol. 44, no. 32, pp. 3976–3984, Oct. 2010, doi: 10.1016/j.atmosenv.2010.06.060.
[94] M. D. Petters and S. M. Kreidenweis, "A single parameter representation of hygroscopic growth and cloud condensation nucleus activity," *Atmos. Chem. Phys.*, vol. 7, no. 8, pp. 1961–1971, Apr. 2007.
[95] A. Di Antonio, O. Popoola, B. Ouyang, J. Saffell, and R. Jones, "Developing a relative humidity correction for low-cost sensors measuring ambient particulate matter," *Sensors*, vol. 18, no. 9, p. 2790, Aug. 2018.
[96] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017.
[97] *LoRa-Calculator*. Accessed: Apr. 8, 2020. [Online]. Available: https://www.loratools.nl/#/airtime
[98] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," *J. Cryptograph. Eng.*, vol. 2, no. 2, pp. 77–89, Sep. 2012.
[99] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[100] Dragino. *Manuel_LG02*. [Online]. Available: http://www.dragino.com/downloads/downloads/LoRa_Gateway/LG02-OLG02/LG02_LoRa_Gateway_User_Manual_v1.5.1.pdf
[101] D.-D. Truong, T. Nguyen-Van, Q.-B. Nguyen, N. H. Huy, T.-A. Tran, N.-Q. Le, and K. Nguyen-An, "Blockchain-based open data: An approach for resolving data integrity and transparency," in *Future Data and Security Engineering*, T. K. Dang, J. Küng, M. Takizawa, and S. H. Bui, Eds. Cham, Switzerland: Springer, 2019, pp. 526–541.
[102] X. Zheng, Y. Zhu, and X. Si, "A survey on challenges and progresses in blockchain technologies: A performance and security perspective," *Appl. Sci.*, vol. 9, no. 22, pp. 1–24, 2019.
[103] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018, doi: 10.1016/j.future.2017.11.022.
[104] S. R. P. Benedict and J. Kaur, "IoT blockchain solution for air quality monitoring in SmartCities," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2020, pp. 1–6. [Online]. Available: http://arxiv.org/abs/2003.12920
[105] C. Tazoult, R. Chiky, and V. Foltescu, "A distributed pollution monitoring system: The application of blockchain to air quality monitoring," in *Proc. Comput. Collective Intell.*, 2019, pp. 688–697.
[106] Y. Han, B. Park, and J. Jeong, "A novel architecture of air pollution measurement platform using 5G and blockchain for industrial IoT applications," *Procedia Comput. Sci.*, vol. 155, pp. 728–733, Jan. 2019, doi: 10.1016/j.procs.2019.08.105.
[107] K. R. Özyılmaz and A. Yurdakul, "Designing a blockchain-based IoT infrastructure with Ethereum, Swarm and LoRa," in *Proc. IEEE Consum. Electron. Mag.*, Sep. 2018, pp. 28–34.
[108] I. Mayer, "LoRaWan-hyperledger robust network integrity on IoT devices," in *Proc. Disruptive Technol. Inf. Sci. II*, May 2019, p. 28.
[109] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018.
[110] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni, "Evaluation of the IoT LoRaWAN solution for distributed measurement applications," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 12, pp. 3340–3349, Dec. 2017.
[111] J. Lin, Z. Shen, C. Miao, and S. Liu, "Using blockchain to build trusted LoRaWAN sharing server," *Int. J. Crowd Sci.*, vol. 1, no. 3, pp. 1–13, Sep. 2018.
[112] A. Durand, P. Gremaud, and J. Pasquier, "Decentralized LPWAN infrastructure using blockchain and digital signatures," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 12, pp. 1–10, Jun. 2020.
[113] S. M. Danish, M. Lestas, H. K. Qureshi, K. Zhang, W. Asif, and M. Rajarajan, "Securing the LoRaWAN join procedure using blockchains," *Cluster Comput.*, vol. 23, no. 3, pp. 2123–2138, Sep. 2020, doi: 10.1007/s10586-020-03064-8.
[114] Ethereumprice. *Ethereum Average Price*. Accessed: Jun. 29, 2020. [Online]. Available: https://ethereumprice.org/eth-eur/
[115] L. Bader, J. C. Burger, R. Matzutt, and K. Wehrle, "Smart contract-based car insurance policies," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Jul. 2018, pp. 1–7.
[116] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," 2018, *arXiv:1808.01081*. [Online]. Available: http://arxiv.org/abs/1808.01081
[117] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A blockchain-based framework for supply chain provenance," *IEEE Access*, vol. 7, pp. 157113–157125, 2019.
[118] H. Xu, L. Zhang, Y. Liu, and B. Cao, "RAFT based wireless blockchain networks in the presence of malicious jamming," *IEEE Wireless Commun. Lett.*, vol. 9, no. 6, pp. 817–821, Jun. 2020.
[119] S. N. Sm, P. Reddy Yasa, N. Mv, S. Khadirnaikar, and P. Rani, "Mobile monitoring of air pollution using low cost sensors to visualize spatio-temporal variation of pollutants at urban hotspots," *Sustain. Cities Soc.*, vol. 44, pp. 520–535, Oct. 2019, doi: 10.1016/j.scs.2018.10.006.
[120] R. Han, V. Gramoli, and X. Xu, "Evaluating blockchains for IoT," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jan. 2018, pp. 1–5.
[121] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)," in *Proc. IEEE 36th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2017, pp. 253–255.
[122] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling hyperledger fabric to 20,000 transactions per second," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 455–463.
[123] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 17–30.
[124] C. Ferris. (Apr. 2019). *Does Hyperledger Fabric Perform at Scale?* [Online]. Available: https://www.ibm.com/blogs/blockchain/2019/04/does-hyperledger-fabric-perform-at-scale/
[125] X. Zhu and Y. Badr, "Identity management systems for the Internet of Things: A survey towards blockchain solutions," *Sensors*, vol. 18, no. 12, pp. 1–18, 2018.
[126] M. Luecking, C. Fries, R. Lamberti, and W. Stork, "Decentralized identity and trust management framework for Internet of Things," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency*, May 2020, pp. 1–9.
[127] Y.-C. Wu, A. Shiledar, Y.-C. Li, J. Wong, S. Feng, X. Chen, C. Chen, K. Jin, S. Janamian, Z. Yang, Z. S. Ballard, Z. Göröcs, A. Feizi, and A. Ozcan, "Air quality monitoring using mobile microscopy and machine learning," *Light, Sci. Appl.*, vol. 6, no. 9, Sep. 2017, Art. no. e17046, doi: 10.1038/lsa.2017.46.
[128] A. McNabola, A. McCreddin, L. W. Gill, and B. M. Broderick, "Analysis of the relationship between urban background air pollution concentrations and the personal exposure of office workers in Dublin, Ireland, using baseline separation techniques," *Atmos. Pollut. Res.*, vol. 2, no. 1, pp. 80–88, Jan. 2011, doi: 10.5094/APR.2011.010.
[129] J. Gray. (2016). *Changing What Counts—How Can Citizen Generated and Civil Society Data Be Used as an Advocacy Tool to Change Official Data Collection?* [Online]. Available: http://civicus.org/thedatashift/wp-content/uploads/2016/03/changing-what-counts-2.pdf
[130] J. Gabrys, H. Pritchard, and B. Barratt, "Just good enough data: Figuring data citizenships through air pollution sensing and data stories," *Big Data Soc.*, vol. 3, no. 2, pp. 1–14, 2016.
[131] A. L. Clements, W. G. Griswold, R. S. Abhijit, J. E. Johnston, M. M. Herting, J. Thorson, A. Collier-Oxandale, and M. Hannigan, "Low-cost air quality monitoring tools: From research to practice (a workshop summary)," *Sensors*, vol. 17, no. 11, pp. 1–20, 2017.
[132] P. B. English, M. J. Richardson, and C. Garzón-Galvis, "From crowdsourcing to extreme citizen science: Participatory research for environmental health," *Annu. Rev. Public Health*, vol. 39, no. 1, pp. 335–350, Apr. 2018.
[133] Nature Editorial. (2015). *Rise of the Citizen Scientist*. [Online]. Available: http://www.nature.com/news/riseof-%0Athe-citizen-scientist-1.18192

MARKUS LÜCKING is currently a Research Associate at the Research Department for Embedded Systems and Sensors (ESS), FZI Research Center for Information Technology, Germany. His research is mainly concerned with industrial applications based on distributed ledger technology in the field of the Internet of Things.

NICLAS KANNENGIEßER is currently a Research Associate at the Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT), Germany. His main research interests are software engineering, the analysis of system behavior of distributed systems (e.g., distributed ledger technology), and the decentralization of digital applications.

MAURICE KILGUS received the degree in information engineering and management. He is currently working as a Software Engineer. During his studies at the Karlsruhe Institute of Technology (KIT), he focused on Information Security with a particular interest in distributed ledger technology and the Internet of Things. Within the scope of his master's thesis, he designed and evaluated a secure, low-power environmental sensing system that leverages blockchain.

TILL RIEDEL is currently a Postdoctoral Researcher and the Lab Leader at the Research Group for Technology for Pervasive Computing (TECO), Karlsruhe Institute of Technology (KIT), Germany. His current research interests include the application of data analysis methods in industrial domains and the Internet of Things.

MICHAEL BEIGL (Member, IEEE) is currently a Professor at the Karlsruhe Institute of Technology (KIT) and holds the Chair position for Pervasive Computing Systems. He has been the Head of the TECO Research Group since 2010, a Spokesperson and a Co-Founder of the Smart Data Innovation Laboratory (SDIL), and the Co-Director of the Smart Data Solution Center Baden-Württemberg (SDSC-BW). His research interests evolve around people and services at the center of communication and information technology in the field of ubiquitous, pervasive, and mobile computing.

ALI SUNYAEV is currently a Professor of computer science at the Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT), Germany. His research interests include trustworthy Internet technologies and complex health IT applications. His research work accounts for the multifaceted use contexts of digital technologies with research on human behavior affecting Internet-based systems and vice versa. His research has appeared in various journals, including ACM CSUR, JIT, JMIS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and *Communications of the ACM*.

WILHELM STORK is currently a Professor of electrical engineering and information technology at the Institute for Information Processing Technologies (ITIV), Karlsruhe Institute of Technology (KIT), Germany. His research interests include the broad field of the Internet of Things technologies in telemedicine and health care services. His current research focus is on using machine learning techniques for the assessment of vital signs and visual parameters based on brain–computer interfaces.