Article

Novel Air Pollution Measurement System Based on Ethereum Blockchain

Daniele Sofia [1],* [ID]**, Nicoletta Lotrecchiano [1], Paolo Trucillo [1,2]** [ID]**, Aristide Giuliano [3]** [ID] **and Luigi Terrone [4]**

[1] DIIN-Department of Industrial Engineering, University of Salerno, Via Giovanni Paolo II, 132, 84084 Fisciano SA, Italy; nlotrecchiano@unisa.it (N.L.); paolo.trucillo@unina.it (P.T.)

[2] Department of Chemical, Material and Industrial Production Engineering, University of Naples Federico II, Piazzale V. Tecchio, 80, 80125 Napoli, Italy

[3] ENEA-Italian National Agency for New Technologies, Energy and Sustainable Economic Development, Department of Energetic Technologies, Trisaia Research Centre, S.S. 106 Ionica, km 419+500, 75026 Rotondella, MT, Italy; aristide.giuliano@enea.it

[4] DIEM-Department of Computer Engineering, Electrical Engineering and Applied Mathematics, University of Salerno, Via Giovanni Paolo II, 132, 84084 Fisciano SA, Italy; l.terrone2@studenti.unisa.it

* Correspondence: dsofia@unisa.it

Abstract: The need to protect sensitive data is growing, and environmental data are now considered sensitive. The application of last-generation procedures such as blockchains coupled with the implementation of new air quality monitoring technology allows the data protection and validation. In this work, the use of a blockchain applied to air pollution data is proposed. A blockchain procedure has been designed and tested. An Internet of Things (IoT)-based sensor network provides air quality data in terms of particulate matter of two different diameters, particulate matter $(PM)_{10}$ and $PM_{2.5}$, volatile organic compounds (VOC), and nitrogen dioxide (NO_2) concentrations. The dataset also includes meteorological parameters and vehicular traffic information. This work foresees that the data, recovered from traditional Not Structured Query Language (NoSQL) database, and organized according to some specifications, are sent to the Ethereum blockchain daily automatically and with the possibility to choose the period of interest manually. There was also the development of a transaction management and recovery system aimed at retrieving data, formatting it according to the specifications and organizing it into files of various formats. The blockchain procedure has therefore been used to track data provided by air quality monitoring networks unequivocally.

Keywords: air quality monitoring; pollution; environmental security; sensitive data; IPFS; Ethereum blockchain

---

1. Introduction

Air quality, water purity, atmospheric CO_2 concentration, etc., are some examples of environmental parameters that are degrading due to human activities [1].

In particular, the levels of air pollution have changed in recent years, attracting the interest of governance, municipalities, and citizens [2]. As is well known, air pollution has a direct impact on urban pollution [3] and human health [4], especially in developing and industrialized countries [5] that should, instead, implement air quality management.

According to the World Health Organization (WHO), air pollution is the main environmental risk to health in the European Union [6]. Every year, it is the cause of around 400,000 premature deaths in the EU and leads to health-related diseconomies for hundreds of billions of euros. In 2013, the European Commission estimated that the health diseconomies caused by atmospheric pollution

totally reach between EUR 330 and EUR 940 billion per year. According to the European Environment Agency (EEA), in 2015, about a quarter of Europeans living in urban areas were exposed to levels of air pollution higher than those required by some EU air quality standards and up to 96% of EU citizens who live in urban areas have been exposed to harmful levels of air pollutants according to the WHO [6]. Air pollution tends to affect those who live in cities that are larger than the inhabitants of rural areas, since the population density of urban areas entails a greater release of atmospheric pollutants and because dispersion is more difficult in cities due to the high urbanization compared to rural areas. Particulate matter (PM), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), and tropospheric ozone (O_3), due to their characteristics, are classified by the WHO as the most dangerous air pollutants to human health.

Particulate matter (PM) consists of solid and liquid particles suspended in the air. It is characterized by a wide range of compounds, including human carcinogens such as benzo[a]pyrene and carbonaceous particulate matter. The PM is classified as PM10 (particulate matter coarse with diameter < 10 μm) and PM2.5 (fine particulate matter with diameter < 2.5 μm). In regions of Europe where fossil fuels are often still used for domestic heating, the emission of pollutants in the atmosphere, in particular PM, tends to increase with the decreasing of the temperature [7]. Nitrogen dioxide is a toxic reddish-brown gas belonging to the family of nitrogen oxides (NO_X). Sulfur dioxide is a colorless toxic gas with a pungent smell belonging to the class of sulfur oxides (SO_X). Tropospheric ozone is a colorless gas that forms near the ground following the chemical reaction of pollutants (such as volatile organic compounds (VOC) and NO_X) in the presence of sunlight.

Air quality monitoring and control are necessary to implement abatement strategies [8] and improve environmental consciousness in the community. Traditional methods, such as monitoring stations represented by public networks, are precise but expensive and bulky and, for these reasons, do not have a high space–time resolution. These stations provide measurements according to environmental regulations and are based generally on chemical analyzers. The growth of the Internet of Things (IoT) industry is presenting new possibilities for air quality monitoring and management for smart cities. Due to the high level of interest to air pollution, from the new point of view of smart cities, municipalities and private citizens have implemented air quality monitoring stations on their own. From this perspective, high space–time resolution monitoring networks are born [9] that enable a massive amount of data describing the most sensitive part of the environment investigated [10], to support decisions [11], and inform the general public. Measuring the air quality levels in smart cities should provide useful tools for alert services, forecasting, and warning. In particular, the daily forecasting of the atmospheric concentrations of pollutants is a vital support for policymakers in developing regulatory programs. Air quality networks provide public information, and the analysis of a historical time series allows for the definition of the air quality forecasting models [12] as a piece of harmful information to modify events that will occur and limit social activities ahead of schedule on the most polluted days. Some of the significant manufacturers of monitoring networks have developed portable devices in recent years. Their small size allows their direct use by citizens, measuring all the polluting parameters supported by the device. However, none of these ready-to-use devices have efficient real-time detection systems or high performance [13]. Recent studies allowed for the development of mobile monitoring systems in real time, such as platforms active on vehicles [14]. Monitoring systems directly on the road, such as those used in the Netherlands [15] and in Zurich, Switzerland [16,17], demonstrated how this measurement system can be a valid tool for assessing the spatial variability of the pollutant concentration. Moreover, they allow us to investigate specific areas that it would not be possible to assess with fixed monitoring systems. By increasing the temporal resolution of these systems, it is also possible to improve the reliability of the measurements and the quantity of data collected. This new technology, defined as a Real-Time On-Road Monitoring Station (ROM), represents a new concept of air quality sensing devices designed to be located on a moving vehicle within the urban area, to measure pollutant levels in space and time. By using sensing devices on vehicles traveling through well-defined routes, it is possible to measure the space

and time distribution of pollutants. This new technology increases the spatial resolution; in addition, considering the high time resolution, it allows us to improve the amount of data collected. The new air quality measurement technologies cited above are reported to contextualize the monitoring networks at an innovative level. In fact, the innovative aspect of this work is linked both to how data is collected (ROMs) and, as will be explained later, to how data is managed (sending them to the Ethereum blockchain).

The data validity of the data is often questioned, especially if they are labelled as sensitive. Today, the problem of data validity is widespread and concerns every sector, who must guarantee that data have not undergone changes of any kind. Having a system that helps to increase the data validity is a significant step forward to ensuring that their integrity has been provided, especially if they are sensitive. The fields affected by this problem are many, involving all the areas where data are connected to people to make public decisions. Technological innovation comes into play to support the validity of data with the advent of the blockchain. A blockchain is a shared and immutable data structure. It is defined as a digital register whose entries are grouped in blocks, linked in chronological order, and whose integrity is guaranteed by the use of cryptographic techniques. Among the various principles that lie behind this structure, the most interesting feature is its immutability since, as a rule, its content, once it is written, is no longer either modifiable or eliminable, unless the entire structure is invalidated.

Blockchains are based on transactions that take place between two digital wallets that allow users to store and manage their Bitcoins and Ether. The idea is to send transactions, containing the data involved, between the two wallets of the same owner, in order to modify the real purpose of the latter, thus becoming real registers where each transaction becomes the instrument that validates the data.

The blockchain concept has been applied in the past year to the food and agri-food sector, in healthcare, in lifecycle management for Industry 4.0, in security and anonymity and in personal data protection [18–23]. The novelty of the work proposed is represented by the use of a blockchain to protect air quality measurements that could be altered by users for unethical purposes. The concept of a blockchain has been extended to air quality only in the past two years in relation to the concept of smart cities [24]. Air quality data often require manual intervention for their access and use. Unlike previous studies [25], the air quality data used in this study were obtained from a real and fully working monitoring network running in the Italian city of Milan [14]. Moreover, the use of the InterPlanetary File System (IPFS) protocol allows for the storage of a large amount of data while keeping a low transaction price, unlike the smart contract that is commonly used [26].

The blockchain was used to ensure immutability and transparency in the measured data. The monitoring systems and the data provided are not perceived in the same way all over the world, but there are some portions of the population that do not consider such data reliable and trustworthy. One way to increase user trust with respect to environmental monitoring is certainly to somehow guarantee the integrity of such data, and this is achieved through the implementation of a blockchain.

Ethereum is the best available blockchain and allows us to uniquely write data. Coupled to the IPFS protocol, it is possible to increase the number of writable data without increasing the gas price.

The private blockchain has lost the advantage of trust over data. Public blockchains have public ledgers to which everyone can connect by writing in the Ethereum wallet in which the data was written. A private blockchain does not have the same agreement mechanism as the public one. The private blockchain publishes only a limited number of ledgers; however, its use is not free, since the agreement mechanism is managed by a private company. The mechanism to modify the private blockchain is, however, in the hands of a company, which does not guarantee the integrity of the data. So, to ensure the immutability and transparency of the data, in this work, the public blockchain of Ethereum was chosen.

The novelty of this work is shown in Figure 1. The wallet sender dispatches the transaction fee to the wallet receiver. After the transaction, a string encoded in hexadecimals is written in the Ethereum

blockchain. This string contains the key to access the database and therefore all the data written inside. In this way, a single string is written in the blockchain, therefore containing a reduced number of characters with an extremely low transaction fee, which contains thousands of data.
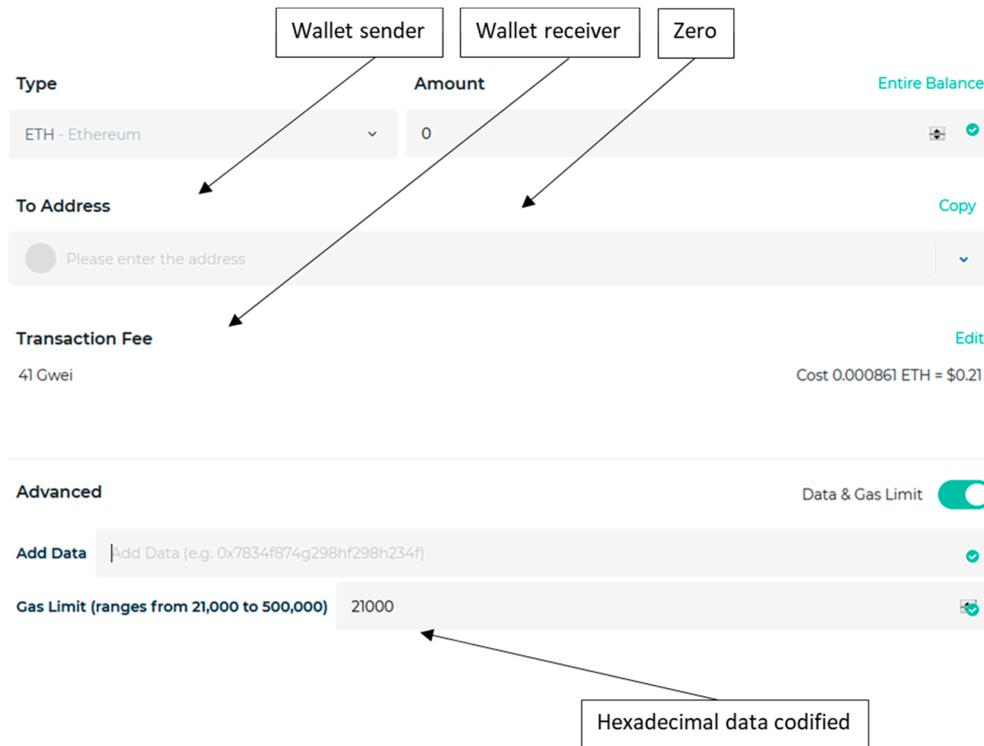

Figure 1. Ethereum blockchain transaction.

2. Experiments

As described in detail in the following paragraphs, several tools were used to build the informatics system to write in the blockchain data from mobile air quality monitoring stations. The use of a developed environment that is able to interact with the Test-Net was crucial to optimize costs and reduce the time taken to develop this application for the blockchain.

2.1. Data

The data written in the blockchain come from a real-time on-road monitoring station network operating in Milan (Italy), as already presented in our previous study [14]. The network provides measures of air quality in terms of particulate concentrations (PM_{10}, $PM_{2.5}$, PM_1), gas concentrations such as NO_2 and volatile organic compounds (VOC). This innovative technology is an IoT-based monitoring station located on a commercial van that samples the air during its operating time. The technology installed allows for a high time resolution with a sampling period of 3–5 min. Due to the dynamic working mode of the monitoring network, the data are collected with a high spatial resolution that is 1 square km large, considering a total area of 100 square km. The data collected are sent via Long-Term Evolution (LTE) to the Firebase database. The dataset also includes the meteorological data measured by the monitoring station, such as temperature, relative humidity and pressure and integrates the data of traffic, wind intensity and direction derived with an application programming interface key (API) network from external databases.

2.2. Blockchain

A blockchain is a timestamp set (a digital record of the time of occurrence of a particular event) of immutable data records managed by a cluster of computers not owned by a single entity. Furthermore,

it is defined as a sub-family of technologies structured as a chain of blocks containing the transactions whose validation is entrusted to an approval mechanism, distributed on all the nodes of the network. Network nodes represent all the nodes that are authorized to participate in the transaction validation process to be included in the register. The main features of blockchain technologies are immutability, transaction traceability, and security based on cryptographic techniques [27]. The blockchain network has no central authority. Since it is a shared and immutable register, the information contained inside is accessible to anyone. Due to the transparency of its nature, all the users involved are responsible for their actions.

Ethereum is a distributed public blockchain network based on blockchain technology that allows developers to create and distribute decentralized applications. The Ethereum blockchain focuses on the execution of the programming code of any decentralized application. In the Ethereum blockchain, miners work to earn Ether, a type of cryptographic token that powers the network. In addition to a negotiable crypto-currency, Ether is also used by application developers to pay commissions and transaction services on the Ethereum network. There is a second type of token that is used to pay miners commission for the inclusion of transactions in their block, or gas, and any smart contract execution requires sending a certain amount of gas to entice the miners to insert it into the blockchain. The Ethereum engine is represented by the Ethereum Virtual Machine (EVM), which represents the runtime environment for the development and management of smart contracts. EVM operates securely and separately from the network. The code managed by the virtual machine does not have access to the network; moreover, the same smart contracts generated are independent and separated from each other. Smart contracts are available in the blockchain in EVM byte code (an Ethereum-specific binary format). Contracts are written in Ethereum high-level language, transformed into byte code with an EVM compiler and uploaded to the blockchain with an Ethereum client.

2.3. IPFS Protocol

The InterPlanetary File System (IPFS) is a protocol and a distributed system for storing and accessing files, websites, applications and data. IPFS tries to create a permanent and distributed web using a content-oriented system instead of a location-based system introduced by the Hypertext Transfer Protocol (HTTP), which is the most used protocol to transfer data over the web [28].

- An HTTP request is of the type http://10.20.30.40/folder/file.txt
- An IPFS request is of the type ipfs/QmT5NvUtoM5n/folder/file.txt

HTTP relies on addressing via IP addresses to identify the location of the specific server that is hosting the requested information by retrieving it from the source server. An IP address is a unique address that identifies a device on the Internet or a local network. It allows a system to be recognized by other systems connected via the Internet protocol. IPFS does not have a single point of failure as it is a peer-to-peer distributed file system that decentralizes the Internet and prevents information being published from suddenly disappearing. IPFS uses a representation of the content required for addressing, using a cryptographic hash on a file whose result is used as an addressing method. The hash is associated with the file contents; therefore, the IPFS file address is created from the content. The IPFS protocol is not limited to downloading files from other servers, but the computer also helps to distribute them, making it possible to download not only web pages but also any type of file that a computer could store, such as a document, an email, a MP3 file or even a database record. The use of IPFS is characterized by several advantages:

1. It makes it difficult for a website to go offline. In the case of a sudden failure or accident that damages the servers, it will still be possible to get the same webpage at a different address.
2. It makes it more difficult for authorities to censor the content because files on IPFS can come from different places, and it is very difficult for private or public authorities to block their contents.
3. It can speed up the web when the user is away or disconnected. It is quicker to get a file from someone nearby instead of hundreds or thousands of miles away; with IPFS, the latter

happens more quickly. Organizations with enough money and expertise can do this with Content Distribution Networks (CDNs) or with Multiple Data Centers (MDCs), but IPFS aims to make it possible for everyone.

The use of IPFS is participatory and collaborative; if an IPFS user does not share the content identified by a given address, it will not be possible to obtain it. On the other hand, the content cannot be removed from IPFS as long as someone is interested enough to make it available, regardless of whether they are the original author or not. IPFS is based on very sophisticated technology. The ideas are based on how networks of people and computers communicate. Today's World Wide Web (WWW) is structured in ownership and access, while IPFS is developed on the ideas of ownership and participation, in which many users own files, while others participate in making them available.

Concerning data storage, IPFS uses a distributed hash table (DHT). Once the hash is obtained, a request is sent to the peer network. The content is located in such a hash, so it is downloaded directly from the node containing the requested data. Data are transferred between the nodes of the network using mechanisms similar to BitTorrent: a user searching for some content on the IPFS web finds neighbors who have access to such content, then downloads small bits of content from those neighbors. Moreover, like the DHT and BitTorrent protocols, IPFS uses a Merkle tree structure, a data structure similar to that on which Git is based, to track content across the web. The Merkle tree represents a procedure for entering transactions in which the leaves are the txid transactions, the branches (forked) the intermediate hashes, and the root the final hash, the product of all the other hashes: the Merkle root. Thanks to the Merkle tree structure, it is not necessary to know all the transactions included in a block to verify that a single transaction is part of it. Instead, it is sufficient to follow a particular branch connecting a leaf (i.e., a transaction) to the Merkle root. IPFS was borne by the blockchain gene. Juan Benet, the inventor of this project, created it using the blockchain Bitcoin protocol, considering that the protocol stores and shares the content of immutable information and removes duplicated content [29]. When implementing a blockchain with the IPFS protocol, the transfer of data between nodes will be faster, more secure, and unalterable.

The idea of a permanent network that is durable and efficient was undoubtedly also the goal of the original inventors of Internet protocols. However, over time, as web usage has changed, weaknesses have emerged in these protocols, even though, in its early stages, IPFS showed the promise of being a crucial piece of a new decentralized technology stack. IPFS was essential for the development of the present work; in fact, what characterizes the sending of the transaction is a fee set by the miners. This fee is directly proportional to the amount of data entered in the transaction; this implies that, when sending a weekly sampling of the pollutants of an entire city, the transaction cost, in the case of a low gas price, can even reach the order of a few euros. With IPFS, regardless of how much data are sent, the generated hash will always be 256 bits; therefore, it is elementary to guess that the fee per transaction is around hundredths of euros and, thus, is a significant saving when considering that the data is sent daily.

2.4. Python

Python is a high-level programming language, released in 1991 by its creator Guido van Rossum. Python supports several programming paradigms, such as object-oriented (with support for multiple inheritances), imperative and functional, and offers strong dynamic typing. It is equipped with a rich built-in library, which, together with automatic memory management and robust exception management constructs, makes Python one of the more complete and most comfortable languages to use. Python is a pseudo-compiled language: an interpreter takes care of analyzing the source code (simple text files with a .py extension) and, if syntactically correct, executing it. Python was used for all the scripts produced for this work using some feature libraries.

2.5. Web3.py

Web3 is one of the best libraries in this work, since it allows for a direct interaction with the Ethereum blockchain and can carry out the primary operations such as the execution and recovery of transactions. The connection to the blockchain occurs through a provider that defines the way in which Web3 connects to the blockchain. The Web3 library comes with the following built-in providers, suitable for the most standard cases of use:

- Web3.HTTP provider for connection to JavaScript Object Notation (JSON-RPC) servers based on http and https.
- Web3.IPCP provider for connection to JSON-RPC sockets based on the Inter-process communication (ipc) sockets.
- Web3.Websocket provider for connection to websocket and secure websocket connection (wss) based JSON-RPC servers.

The HTTP provider runs the complete uniform resource locator (URL) where the server can be found. The IPCP provider starts the file system path where the IPC socket can be found. If no argument is provided, the default path for the operating system in use will be used. The sendRawTransaction () method was used to send the transaction. Before sending, the transaction must be signed using the signTransaction () function in order to create the nonce or the transaction ID that remains pending before the actual sending takes place. The transaction parameters must be supplied in JavaScript Object Notation (JSON), which represents a lightweight data-interchange JSON format, using the following fields:

- from: sender address;
- to: recipient address;
- gas: gas limit;
- gas price: the price of the gas to pay;
- value: the value of the sending Ether;
- date: data to be sent in the transaction;
- nonce: this allows the overwriting of the pending transaction using the same nonce.

2.6. Firebase.py

The Firebase.py library allows for an interaction via URL and secret key with the NoSQL Firebase database. The NoSQL database provides a mechanism for the storage and retrieval of data, which is different from the tabular relations used in relational databases. Among the main features of the library are the possibilities to create, update and delete data, and query the database. By using the functions FirebaseAuthentication () and FirebaseApplication (), it is possible to access and manage the database. The functions get () and post () allow us to query and to manage the database, respectively.

2.7. Zlib.py

The functions in the zlib.py library allow data compression and decompression. These functions have many options ad usually have to be used in a well-defined sequence. In particular, the function compress () has two parameters as inputs—the data to compress and the compression level, which varies between one and nine. Decompress () is the function that allows data decompression and takes, as inputs, the data to decompress.

2.8. Ipfshttpclient.py

The ipfshttpclient.py library allows for the use of the IPFS protocol.

2.9. Requests.py

The Requests.py library is the standard tool for HTTP requests in Python.

2.10. Firebase

Firebase is a platform created primarily for the development of web and mobile applications, but, thanks to integrated services, this platform performs highly for remote data collection and processing. Firebase offers several services, providing Software Development Kits (SDKs) and multi-platform application programming interfaces (Android, IOS, JavaScript, C ++, Unity) to interact with them. The database provided by Firebase is an NoSQL cloud Database based on a hierarchical tree structure with a single root and with a structure that does not have a predefined scheme that can change over time. One of the advantages offered by this service is the complete synchronization between each client in real time; in fact, when using the SDK, all requests made to the real-time database are stored locally in the client's cache and are updated in real time as changes and events occurring within the database. When the previously offline device regains connection, the real-time database SDK synchronizes local data changes with remote updates that occurred while the client was offline, resolving conflicts, and then making any data available at the same time as any application that uses the database closes the connection.

In this work, Firebase was crucial in the development of the data recovery system, as there was a need to interact with the database using specific requests to obtain the data. Furthermore, to increase the current database performance and organization, a new node has been added with all the fields related to sampling, so that, by adding a new field in this node, all the fields in the various sampling sections are updated.

2.11. Ganache

Ganache is an open-source software that allows for rapid Ethereum blockchain development. A JavaScript-generated blockchain will replicate the behavior and characteristics of the Ethereum blockchain. Because it is a test blockchain, it is possible to configure it according to customer needs. Figure 2 shows the core of Ganache: a list of ten Ethereum accounts, each with ETH 100. These accounts were generated in the local blockchain, and it is possible to display the private key of each account, which is necessary to send transactions.

Figure 2. Ganache front page.

During the development of smart contracts and decentralized applications (dApps), it is necessary to monitor transactions: on the transaction page, it is possible to check the details of each transaction at any time. Figure 3 shows all the transaction information. At the top, there is the transaction (TX), which is not the ID of the transaction; immediately below the sender address and the contract address are, respectively, the sending and the receiving wallets. The value is the Ether amount sent to the receiving wallet, while the gas is used to calculate the transaction. Based on this last parameter, the miners get their fees. Finally, the TX represents the data sent in the transaction.
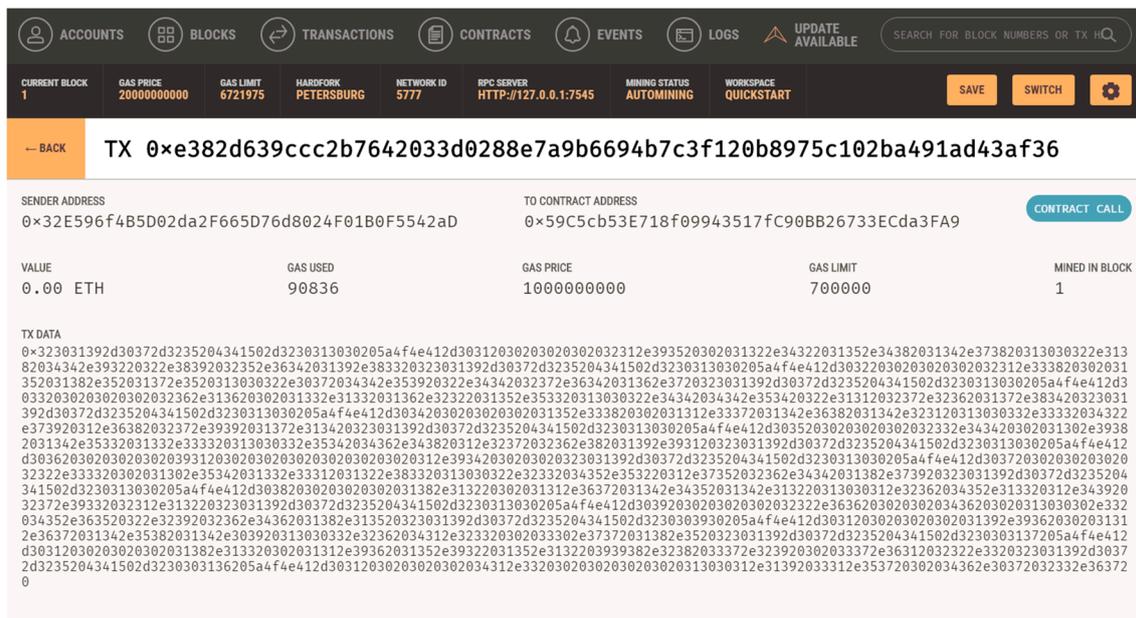

Figure 3. Transaction example.

Ganache was used in this work for the initial testing of Ethereum dynamics. Thanks to this tool, it was possible to analyze the transaction costs and, based on the data sent, how they increased. Furthermore, it has been possible to build a robust management system for any impossibilities of sending transactions, caused either by the insufficient gas price limit inserted to compute the transaction, or the low gas price inserted, or a case in which there is not enough Ether in the wallet.

2.12. Infura

Infura is a scalable backend infrastructure for creating DApps on the Ethereum blockchain. It is a method to connect to the Ethereum network without running a complete node. The most straightforward interface to access Ethereum is hosted through Amazon cloud servers and is the most commonly used method by DApp developers to connect to the Ethereum network. Infura is a collection of complete nodes on the Ethereum network that allows developers to connect to these nodes through its interface. Therefore, a significant portion of DApp traffic crosses Infura, thanks to its ease of use, which does not require anything for developers to perform a complete node locally or continuous maintenance. Not running a complete node is suitable for developers, who can then concentrate more on building DApps, rather than consistently managing the complete connection of the node to the network. Infura offers numerous development tools, documentation, and API keys to work with Ethereum, also enabling distributed storage via IPFS. Infura's IPFS gateway is a useful feature of its design, and the consistency of IPFS with the blockchain should continue to increase its growth and knowledge among developers.

Infura was used in this work for the real completion of the transactions. Once the connection with the Infura provider was established, after some operations, the real transaction was sent. Moreover,

through the IPFS address and the hash recovered from the transaction, it is possible to verify the data loaded.

2.13. Concept Design

The blockchain management system stems from the need to have a tool that allows the system administrator to send data back to the structure as well as receive it. All this has the objective of having a certification that the data are valid. The work is divided into one part executed automatically by the company server, and another part executed manually, dedicated to the user. The part dedicated to the user is based on the concept of simplicity. Even if the system does not have a graphic interface, it has been designed to be user-friendly and executable by all users [29]. The manual sending system is intended to be executed by the user and allows for the choice of various alternatives, such as:

- The sending of the generated IPFS protocol hash or the sending of compressed data.
- The possibility to choose the sending of only a sample of the interested zip code or the Milan zip code directly.
- The possibility to choose to send a sample that refers to the following day, to a past day or to several days in the past.

The possibility to choose a fee for the arbitrary transaction, having—as a reference—the current fee.

Once chosen, all the options will be sent either as a compressed string or as the hash generated by the IPFS protocol. This module, apart from sending older sampled data, allows us to re-send a transaction that has not been confirmed due to the low "fee" entered or the low credit left in the sending wallet. The automatic sending system is intended for automatic execution on the company server; in fact, it is timed and is executed, after some analysis, in a time slot where the "fee" is—on average—lower, to provide a further saving in the transaction. The system sends the hash generated by the IPFS protocol in the transaction since the execution occurs daily. Finally, an email is sent if the remaining credit on the sending wallet is not sufficient to carry out the transaction. The system is sent automatically by the server one hour later than the execution of the automatic submission script. The primary purpose of this module is user notification in the cases where a transaction has been successful or if it is still "pending" or not accepted due to a low fee. The notification would be the sending of an email with an "ok message" if the transaction was successful, while it would contain a link with the transaction ID if the latter had not been confirmed. The data recovery system is run by the user, who can arbitrarily choose the number of transactions to recover from the blockchain. Once this is done, the system searches for the chosen transactions and retrieves the data within the latter. The script also distinguishes whether the data within the transaction are compressed or comprise an IPFS hash; for this reason, there is an IPFS decompression or request for data recovery. Once the data are obtained, they need to be reformatted and, finally, the following are created:

- Two files in JSON and text file format, containing data as well as recovery data from the blockchain.
- Two files in JSON and text file format with the possibility of choosing whether, in addition to sampling, to have only the date of the blockchain or only the data recording date in Firebase.

The generated files are intended, in addition to being analyzed by the data analyst, to be used to update the dashboard with the various concentrations per zone.

3. Results and Discussion

3.1. Methodology Implementing the String to Send in the Blockchain

The idea that comprises the basis of this work can be deduced easily from Figure 4, where the logical path followed is reported.
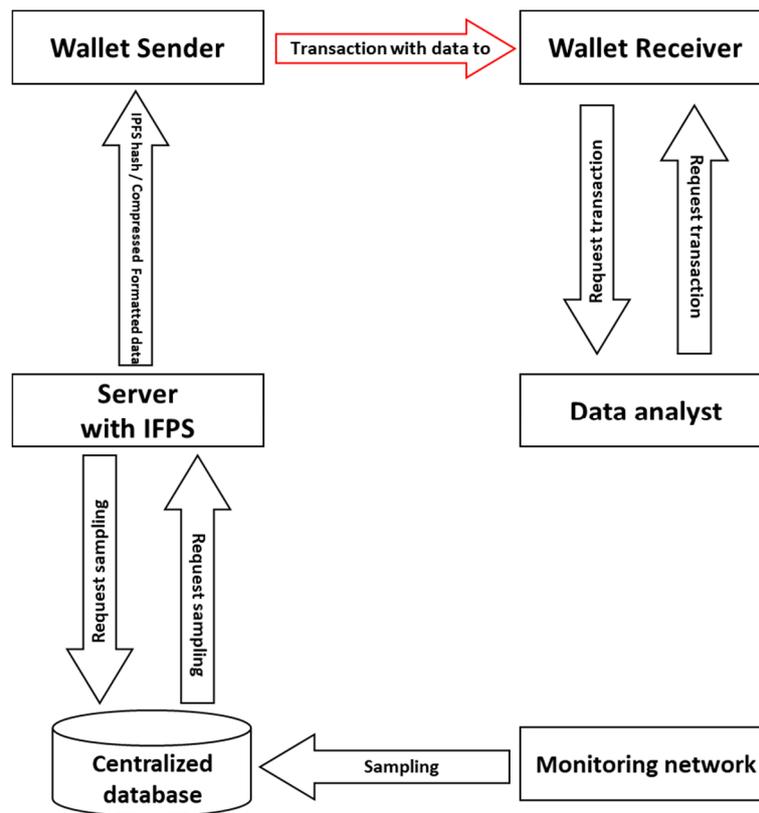
Figure 4. Conceptual path of the project.

As described in Figure 4, the blockchain is not integrated directly into the sensor but it is the final part of data validation and storage. The air quality monitoring sensor sends the measured data to the server via an LTE connection. These data are stored in a centralized database, which, in this work, is the Firebase web platform. Once the data have been written to the server, they are compressed into a string that is written in the blockchain using the IPFS protocol. Writing in the blockchain makes the data immutable and means that it cannot be manipulated later.

Every day, through the user interface for manual sending, the sending system retrieves the required zone data from Firebase. The response from the database will be null if the requested data are not stored or will contain data if measurements are taken that day. As explained previously, the price per transaction is directly proportional to the content of the transaction, and it is therefore necessary to pay attention to the design of the string to be sent to Ethereum. Sending saved data directly from Firebase, without being formatted, wastes money. It was therefore decided to implement a string optimization procedure. Once the request has been made to Firebase, the data are shown in the following format (Figure 5).

[{'co': 0, 'direzione_vento': 0, 'intensita_vento': 0, 'no2': 18.12, 'o3': 0, 'pm1': 11.67, 'pm10': 14.45, 'pm2_5': 14.12, 'pressione': 1001.26, 'temperatura': 45.13, 'traffico': 1.49, 'umidità': 27.93, 'voc': 21.12}]

Figure 5. Data from Firebase.

As shown in Figure 5, the data appear not to be contextualized. The date, zip code, and area to which they belong are unknown in advance; moreover, in this format, many fields and strings considerably increase the weight of the transaction. All the fields were deleted to have, at the beginning of the string, only the sampling values, such as date, zip code, and area; this allows us to have temporal and spatial reference data. Furthermore, not all monitoring stations have the same sensors enabled

and, therefore, the same sampling fields; therefore, it was necessary to organize the string in such a way that, if there is an inconsistency of fields, a zero is added.

The string obtained as a result is the following (Figure 6).

```
2019-07-25 CAP-20100 ZONA-01 0 0 0 21.95 0 12.42 15.48 14.78 1002.18 44.92 2.89 25.64 19.83
```

Figure 6. Formatted string.

Before sending the actual transaction, we used Ganache for testing. The first transaction was sent by inserting the string in the format of Figure 7 without any further processing. For the following examples, a sample of four Milan zip codes was sent, consisting of 17 total areas.

```
2019-07-25 CAP-20100 ZONA-01 0 0 0 21.95 0 12.42 15.48 14.78 1002.18 44.92 2.89 25.64 19.83 2019-07-25 CAP-20100 ZONA-02 0 0 0 21.38 0 15 18.5 17.5 1002.07 44.59
2.44 27.64 16.7 2019-07-25 CAP-20100 ZONA-03 0 0 0 26.16 0 13.13 16.22 15.53 1002.44 44.54 2.11 27.26 17.84 2019-07-25 CAP-20100 ZONA-04 0 0 0 15.38 0 11.37 14.68
14.21 1003.33 42.79 1.68 27.99 17.14 2019-07-25 CAP-20100 ZONA-05 0 0 0 23.44 0 10.98 14.53 13.33 1003.54 46.48 1.27 26.8 19.91 2019-07-25 CAP-20100 ZONA-06 0 0
0 91 0 0 0 0 0 0 1.94 0 0 2019-07-25 CAP-20100 ZONA-07 0 0 0 22.33 0 10.54 13.31 12.83 1002.23 45.52 1.75 26.44 18.79 2019-07-25 CAP-20100 ZONA-08 0 0 0 18.12 0 1
1.67 14.45 14.12 1001.26 45.13 1.49 27.93 21.12 2019-07-25 CAP-20100 ZONA-09 0 0 0 22.66 0 0 46 0 1000.3 45.65 2.29 26.46 18.15 2019-07-25 CAP-20090 ZONA-01 0 0 0
19.96 0 11.67 14.58 14.09 1003.26 41.23 0 30.77 18.5 2019-07-25 CAP-20017 ZONA-01 0 0 0 18.13 0 11.96 15.92 15.12 998.28 37.29 0 37.61 22.3 2019-07-25 CAP-20016
ZONA-01 0 0 0 41.3 0 0 0 0 1001.19 31.57 0 46.07 23.67
```

Figure 7. Formatted string referring to a one-day sample in Milan.

The string before sending is in the following format (Figure 7).

Before sending the actual string in the transaction, it must be converted into hexadecimals, in accordance with the blockchain policies. It is easy to understand, before showing the results clearly, that in sending a string of this type, the cost per transaction is not favorable for the purpose of cost optimization.

In Figure 8, it is clear that 90,836 gases are needed to send a daily sample of Milan, which, translated in terms of money, using a low gas price, amounts to EUR 0.38. The price reported in Figure 8 may seem acceptable, but it is sufficient to think that this type of transaction is carried out daily and therefore it is about EUR 11.40 monthly, which becomes EUR 136.80 yearly. Moreover, the system administrator could send transactions related to several days in the past, and the expense to be faced could become more conspicuous. The sending of non-optimized data leads to a considerable expense; therefore, a data compression phase was carried out. In this way, a solution was obtained, which, compared to the previous one, leads to a very advantageous saving. Upon recovering the data, instead of sending them directly (as done previously), a compression phase was performed, obtaining the following result (Figure 9).

Figure 8. Raw data transaction.

Message compressed : b'x\xda}\x93Kn\xc50\x08E\xb7\xc2\x06\x82\x00\x7f0\xc3\xa7\xce\xdb\x8e\xbb\xff\x85\x94K\x12\xbd\xea\xb5\xb5"\xd9R\x12\x9f\x1c.\xc4D\xe3\x10?1
\xd0\xdb\xe3\xf30Q\x11\xfa\xfax\x7f\x1c\xa2$u\x99r\x8c\xdc\xd5\xb8\x1b\xe9\xe0\xbeH;{\xae"\xc6\xba\xa8w\x0e#\xe3\x15d\x83g\'\r^\x8d\xec\x7f\xb6=\xd9m\x81=H\x17\xe
7\xe2X\x80\x15\x07v$\x91{\'\xf3\xc2N\xf6\x1d\xb5\xdd\xd4\xc9:Am\xac\r\xa7\xac\xbcG;\xd1\xc9\x03:\xa9\xac\n\xb4M|x\xf5\x1d\xbb_\xec\xe4\x9c\xc6i\xee\xc8aV\x1a\xa6^
7n\x8d\xba\xb1\x07)\x1e$;\x02l\xdd\xb2\xc7\xed\xdd\xe0\x961\xe1(*\x8c\x8bY\xec4\xee\xb3\xd2gs\x14\xb9\x10t\xe8\x0e=/t\xdc\xdd\xbc\xaa\xe08\x0b\xda\x1c\xf5\xdb\xca
`PVi\x00\x1f\xc5,\xac+M\xcb\x8a3\xdbL\x98}\xc0*K\xc8^f\x04\x1b\xf6\xba=\x16\xab\x9di\xceJ\xb3\x0f\xacy/\xdfU4&\xd9\xe8!\xf7\xa84\x1bF&\x1fo\xd8\xf1\xf4\x9eg\xf9\x
bd\x86AD\xb8\`gj\xb2E\xb9\xceR\x18\xbfy\x12\xaf\xff\x01\xb2\x9e?]G\xf5{\xbfW\xdd\x81\xab"\r\xa1&\xec~\x0e\xf4o\xae\xfa+w\xa1\xc2\xe2&?\xe7+jZ\xb3\xc8\x88\xc5\xb6\
xa89l\x05\xfb\xd4j\xc7_\xd8\xf9\x82M\x99\xf6\xec7\xd2\xd4\xa0\xa6<\xbc\x12\xc1\x0f\x96\xf3\x96\x95|\x03\xd6r\xc3\t'

Figure 9. Compressed string.

Following the conversion into hexadecimal, as per protocol, there was a transaction corresponding to the string's sending, as shown in Figure 10.
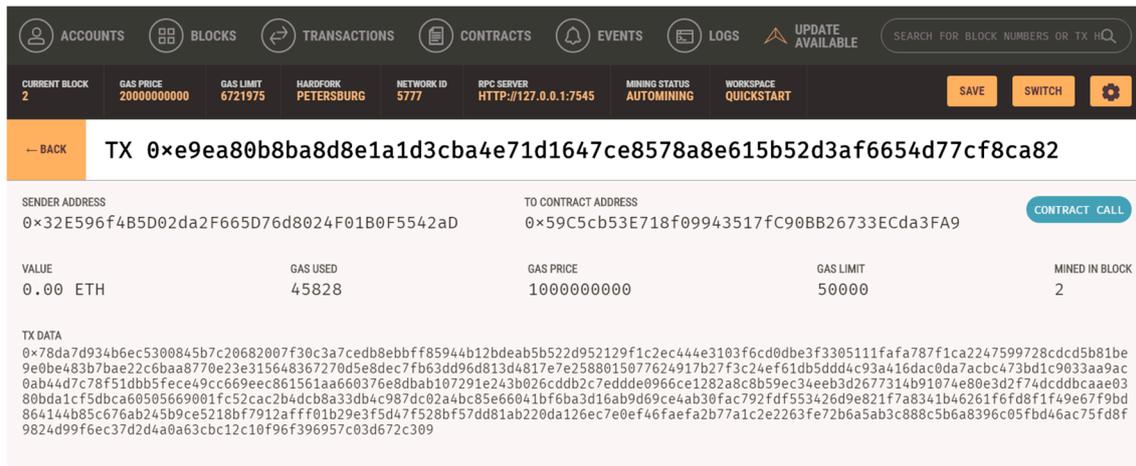
Figure 10. Compressed string transaction.

The parameter of interest is the gas used. Compared with values in Figure 10, it is noted that the gas used to compute this transaction has halved. In terms of expenditure, with the same parameters used for the previous calculation, it is possible to get a price of EUR 0.19 per transaction. In addition to the ability to send the compressed string, a sending system via IPFS was developed. IPFS and blockchain are a perfect combination. It is possible to address large amounts of data with IPFS by inserting immutable, permanent IPFS links (hashes) in a blockchain transaction to provide a timestamp of the data and protect content without sending data directly into the blockchain. In Figure 11, the string has been inserted into a temporary file, and the following hash has been generated from it.
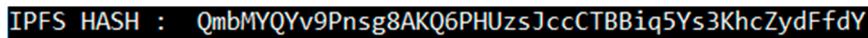

Figure 11. Hash InterPlanetary File System (IPFS).

Through the hash, from a local or from a public IPFS gateway, it is possible to access the content that it refers to, namely the string (Figure 12).
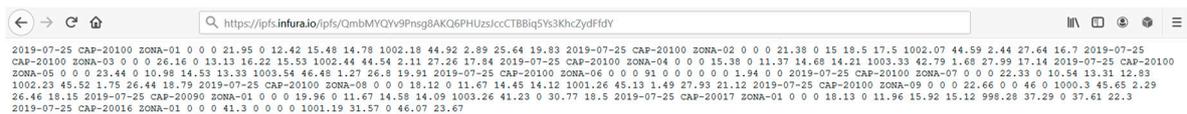

Figure 12. IPFS content through public gateway.

At this point, the hash is sent in the transaction, and the result appears as in Figure 13.
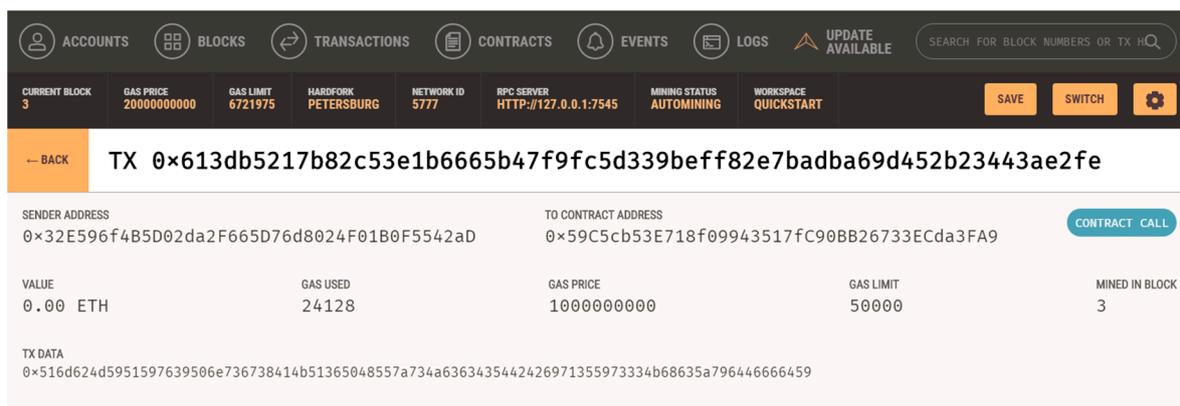
Figure 13. IPFS hash transaction.

Figure 13 shows the vast opportunities available to IPFS; through a 256-bit hash, it is also possible to refer to files larger than several gigabytes. Since the size of the hash is always the same, the gas used per transaction will always be 24,128, which can be translated into EUR 0.05 per transaction. The cost per transaction changes according to the content carried, but it is not the only parameter of influence. A fundamental part of the present study is the optimization of the transition price, which, when compared with that estimated by de Tazoult et al. [30], is reduced by two orders of magnitude. In terms of sending via the Infura mainnet, it has been noted that the prices indicated previously are only indicative. Other parameters that influence the transaction cost beyond the weight of the sending data are the Ethereum price and the Ethereum network congestion. The price of the Ethereum currency, like all virtual currencies, does not have a stable value over time. In fact, the price varies frequently based on current economic policies. It is interesting to analyze how this affects the cost of a transaction. Considering market capitalization (market cap), the unit price, and the volume traded per time unit, it is clear that, based on the principle of supply and demand, the price of money has an oscillatory behavior (Figure 14). Nevertheless, what is interesting is how this price affects the transaction cost related to network congestion. As anticipated, the other parameter of influence for the cost of sending the transaction is network congestion (Figure 15). This parameter is essential in the study of expenditure for the blockchain.

Looking at the curves described by Figures 14 and 15, the Ethereum price approximately grows with the increase in transactions and it decreases otherwise. This is linked to the optimization analysis, before sending the transactions, of the best period to make the transaction. The time chosen for sending the transaction is set at around 7:00 AM, as the network is not congested.

The use of the IPFS protocol allows us to write a data access key in the blockchain with a very low cost related to the transaction. The key written in the blockchain contains within it access to a large amount of data of different sizes, ranging from a few gigabytes to many terabytes. Since the price of the transaction depends on the amount of data registered in the blockchain, the use of the IPFS protocol allows for the writing of thousands of data, always maintaining a low price. The implementation of the smart contract includes an opening phase, a writing phase and a closing phase of the contract to which a price is associated. The solution implemented in this work, using the IPFS protocol, compared to the smart contract, therefore provides for a decrease in price by 1/3. The economic scalability of the proposed solution is provided by the IPFS, which guarantees the decentralized storage of data available to the network by sending the data on the Ethereum transaction by reducing it to a key, thus removing the constraints on the number of data to be sent.

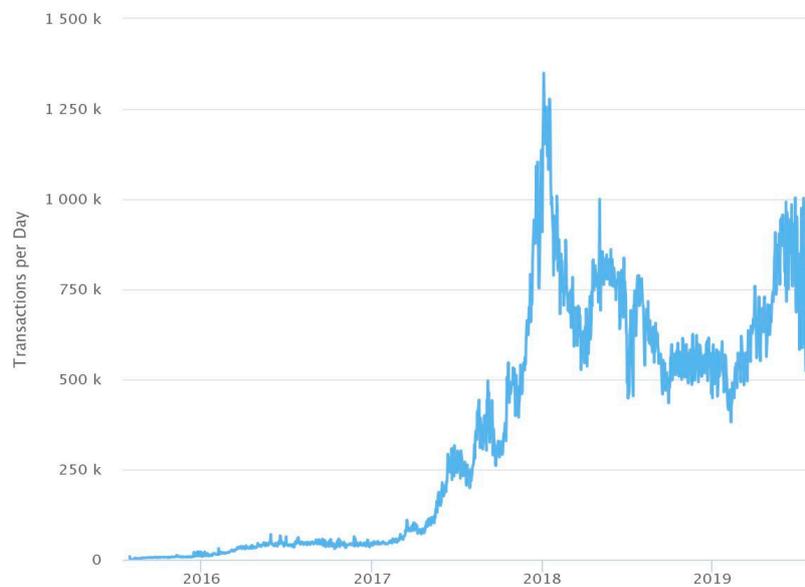**Figure 14.** Chart of Ethereum price.

Figure 15. Transactions chart.

Once the real sending in the blockchain has been carried out, it is necessary to read the data that are inside the transactions and organize them so that the data analyst can process them. A wallet, acting as a sender, dispatches transactions with the data to be validated to a wallet receiver, which performs the function of validating and registration. In fact, by querying the latter, all the validated data are available, since the data that are still pending are only visible in the wallet sender for a given period and are then deleted from the system. In this way, the blockchain develops around the wallet receiver. The user will have the possibility to choose an arbitrary number of transactions to read. Excluding the manual sending function, a transaction is sent every day. So, the arbitrary number chosen corresponds to the number of sampling days to recover.

In data recovery, it is necessary to distinguish the type of data in the transaction since, if it is a compressed string, a decompression process would be needed, whereas, if it is a hash, an IPFS request must be sent for data retrieval. Once the content of the transaction is obtained, there is a need to provide references to the string. In addition to the data reconstruction, there is the need to add the transaction confirmation date, as there is the possibility that a transaction can be sent after a certain period due to a low gas price being entered. The reconstruction of the string consists of a field-given association process. The current fields are retrieved from the Firebase node, with the addition of the zip code, area, sampling date, and transaction date, and everything is sent to a reconstruction algorithm. The reconstruction algorithm, as well as performing the field data association, also converts the characters into numbers. Both the transfer input and the recovery output are strings composed of characters that do not have a useful format for data analysis. If the association does not include dates or combinations of characters and dates, the algorithm also proceeds to a conversion from a string to numbers. When the algorithm has fulfilled all its functions, a structured vector in JSON format will be the output. This format not only follows the Firebase structure, but it is the perfect solution when it comes to handling information organized in field data; moreover, the JSON format is also recommended for more efficient data analysis. As mentioned before, the data read by the blockchain must be supplied to the data analyst. For their characteristics, the JSON and CSV formats are considered the most useful for data analysis. The JSON format is preferred for data analysis using mathematical software such as Matlab, which makes the process of extracting and contextualizing data less laborious, quickly obtaining information on monthly, daily, weekly, and hourly pollution levels. The CSV format is used to get an overview of the pollutant concentrations by zone and date. In fact, from this format, it is much simpler to generate graphs that provide a view of how the pollution of some areas varies due to critical parameters such as humidity, or wind intensity and direction.

3.2. Blockchain Implementation

This system creates a user-friendly blockchain solution. There are no problems related to interfacing with external software to create and send data through transactions, but, through some simple options chosen from the command line, it is possible to send a transaction and find the information from the latter in such a way that the data used are the validated ones.

The system can be summarized with the scheme reported in Figure 16.

The wallet receiver contains the validating registers and the confirmed transactions are sent to the latter, meaning that the information retrieved from the transactions in this wallet is valid and immutable. Transactions that are not confirmed remain visible in the wallet sender for a specified period and, in the worst case, are eliminated. A critical step is the introduction of timestamps. Having a time reference both to confirm the transaction in the blockchain, and to define the sampling day, is necessary for validation. The transactions could not be confirmed on the same day of dispatch, and therefore the introduction of the two timestamps was fundamental in order to have a precise and correct historical classification of the data. The data retrieved from the transactions are organized in various formats, allowing them to be analyzed through specific software, and used to update the dashboard of the company's website (Figure 17).

The data written in the blockchain are available using the IPSF protocol. As reported in Tables 1 and 2, both the sending and receiving transitions are characterized by a unique Tx-hash. The Tx-hash allows for direct access to the IPFS key to be used directly in the protocol in order to visualize and download the data. The advantage of using the IPFS protocol is that it is possible to write a large amount of data in the blockchain while paying a low fee. This is allowed by the IPFS protocol that assigns the Tx-hash to each transaction, ensuring that the dimensions of the described string always remain short and of the same length.

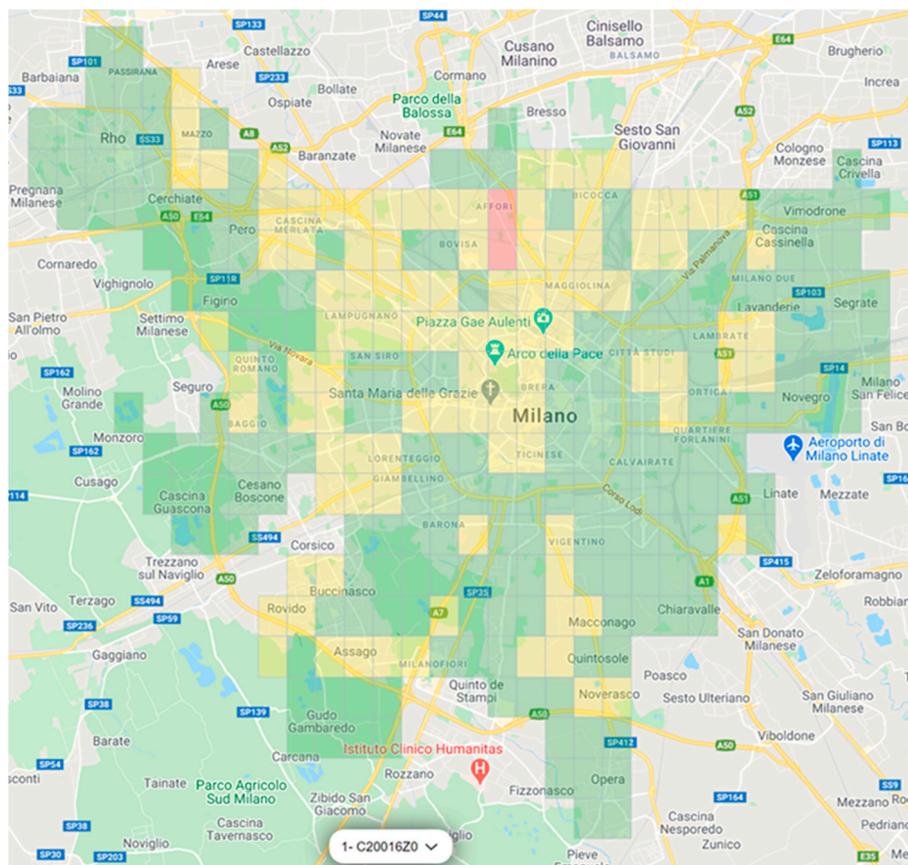**Figure 16.** Procedure for the blockchain implementation.

Figure 17. Particulate matter (PM)10 daily average concentrations measured in Milan, retrieved from the blockchain.

Table 1. Data written in the sender wallet.

Tx-hash	Block no.	Unix Timestamp	Date/Time	From	To	Value In (ETH)	Value Out (ETH)	Current Value @ USD 154.96/ETH	Txn Fee (ETH)	Txn Fee (USD)	Historical USD Price/ETH
0x2f0b603ba2c061d270d722690b0f2 e18c70d47883a1642b2e079293f7c02c744	8269236	1564722075	08/02/2019 05:01	0x428ebe9232f8d68c02e5c369693c 60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	217.70
0x7a40d356eb2e21ad532b2f7d924 ce9bae5bf7827ed04e9249bd8e00548f8eddc	8277036	1564826279	08/03/2019 09:57	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	217.73
0x1d5d662e89b07f94274e5463d557 dac3a8e9a5baab8f7d7fa872973482aa0a5c	8295101	1565067623	08/06/2019 05:00	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	228.70
0xbc30a0bbcdbdd8220e14ee0883 d812e9a9d090f64b5212e4a5cec80045c21a18	8301513	1565154025	08/07/2019 05:00	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	225.99
0x25e0c047c81aa2c9151fede6f7444 f7573d0b144f902bcc6d71fd487d41ee956	8307942	1565240479	08/08/2019 05:01	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	221.18
0xde6ecdcf5346b5d7921176b7db3 d413f6bdbe0c3c4c5d6dfec916098c57cda00	8314437	1565326841	08/09/2019 05:00	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	210.47
0x5572d61666ac167752e046bfd3b 3737db6a395460d507eda6cfd15c3c0b8e229	8320815	1565413238	08/10/2019 05:00	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	206.26
0x172b5545a5dbd6ab19aa1920769 cdf06c1ef2090c2c831f1af0522af6fa82f0a	8340186	1565672442	8/13/2019 5:00:42 AM	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	208.62
0xa51280a92edf9c23998062619ad 460776f17738ae00b82466eb9224994c5288a	8346608	1565758834	8/14/2019 5:00:34 AM	0x428ebe9232f8d68c02e5c369693 c60bb7ce5f45d	0x60aabce99f44f8b8a1 e11934cd04f7671b16fd53	0	0	0	0,00024	0,03739	186.49

Table 2. Data written in the receiver wallet.

Tx-hash	Block no.	Unix Timestamp	Date/Time	From	To	Value In (ETH)	Value Out (ETH)	Current Value @ USD 154.96/ETH	Txn Fee (ETH)	Txn Fee (USD)	Historical USD Price/ETH
0x2f0b603ba2c061d270d722690b0f2 e18c70d47883a1642b2e079293f7c02c744	8269236	1564722075	08/02/2019 05:01	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	217.70
0x7a40d356eb2e21ad532b2f7d924 ce9bae5bf7827ed04e9249bd8e00548f8eddc	8277036	1564826279	08/03/2019 09:57	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	217.73
0x1d5d662e89b07f94274e5463d557 dac3a8e9a5baab8f7d7fa872973482aa0a5c	8295101	1565067623	08/06/2019 05:00	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	228.70
0xbc30a0bbcdbdd8220e14ee0883 d812e9a9d090f64b5212e4a5cec80045c21a18	8301513	1565154025	08/07/2019 05:00	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	225.99
0x25e0c047c81aa2c9151fede6f7444 f7573d0b144f902bcc6d71fd487d41ee956	8307942	1565240479	08/08/2019 05:01	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	221.18
0xde6ecdcf5346b5d7921176b7db3 d413f6bdbe0c3c4c5d6dfec916098c57cda00	8314437	1565326841	08/09/2019 05:00	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	210.47
0x5572d61666ac167752e046bfd3 b3737db6a395460d507eda6cfd15c3c0b8e229	8320815	1565413238	08/10/2019 05:00	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	206.26
0x172b5545a5dbd6ab19aa1920769 cdf06c1ef2090c2c831f1af0522af6fa82f0a	8340186	1565672442	8/13/2019 5:00:42 AM	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	208.62
0xa51280a92edf9c23998062619ad 460776f17738ae00b82466eb9224994c5288a	8346608	1565758834	8/14/2019 5:00:34 AM	0x428ebe9232f8d68c02 e5c369693c60bb7ce5f45d	0x60aabce99f44f8b8 a1e11934cd04f7671b16fd53	0	0	0	0.00024128	0.037386336	186.49

In Figures 18 and 19, the validated transactions are reported for 9 days of testing. Each transaction is defined by a blockchain number that identifies the dataset written. The resulting CSV file is organized into a table containing each field of measurements. In the reported example (Figure 20), data have been written in the blockchain two days after the request due to high demands that led to the slowing down of the network.

Figure 18. Sent transactions.

Figure 19. Received transactions.

DATA_BLOCKCHAIN	DATA_FIREBASE	CAP	ZONA	co	direzione_vento	intensita_vento	no2	o3	pm1	pm10	pm2_5	pressione	temperatura	traffico	umidità	voc
27/07/2019	25/07/2019	CAP-20100	ZONA-01	0		0	0	21,95	0	12,42	15,48	14,78	1002,18	44,92	2,89	25,64	19,83
27/07/2019	25/07/2019	CAP-20100	ZONA-02	0		0	0	21,38	0	15	18,5	17,5	1002,07	44,59	2,44	27,64	16,7
27/07/2019	25/07/2019	CAP-20100	ZONA-03	0		0	0	26,16	0	13,13	16,22	15,53	1002,44	44,54	2,11	27,26	17,84
27/07/2019	25/07/2019	CAP-20100	ZONA-04	0		0	0	15,38	0	11,37	14,68	14,21	1003,33	42,79	1,68	27,99	17,14
27/07/2019	25/07/2019	CAP-20100	ZONA-05	0		0	0	23,44	0	10,98	14,53	13,33	1003,54	46,48	1,27	26,8	19,91
27/07/2019	25/07/2019	CAP-20100	ZONA-06	0		0	0	91	0	0	0	0	0	0	1,94	0	0
27/07/2019	25/07/2019	CAP-20100	ZONA-07	0		0	0	22,33	0	10,54	13,31	12,83	1002,23	45,52	1,75	26,44	18,79
27/07/2019	25/07/2019	CAP-20100	ZONA-08	0		0	0	18,12	0	11,67	14,45	14,12	1001,26	45,13	1,49	27,93	21,12
27/07/2019	25/07/2019	CAP-20100	ZONA-09	0		0	0	22,66	0	0	46	0	1000,3	45,65	2,29	26,46	18,15
27/07/2019	25/07/2019	CAP-20090	ZONA-01	0		0	0	19,96	0	11,67	14,58	14,09	1003,26	41,23	0	30,77	18,5
27/07/2019	25/07/2019	CAP-20017	ZONA-01	0		0	0	18,13	0	11,96	15,92	15,12	998,28	37,29	0	37,61	22,3
27/07/2019	25/07/2019	CAP-20016	ZONA-01	0		0	0	41,3	0	0	0	0	1001,19	31,57	0	46,07	23,67

Figure 20. CSV formatted data.

4. Conclusions

The blockchain management system represents a solution to the problem of the validity of sensitive information. The temporal traceability of the data sent by an air quality monitoring network, with high spatial and temporal resolution, was obtained.

It was possible to write, in the Ethereum blockchain, the data on the average concentrations of zones referred to as the main areas of the city of Milan. The written data refer, in particular, to the average concentrations of NO_2, O_3, PM1, PM10, PM2.5, pressure, temperature, traffic, humidity, and VOC. Time traceability was guaranteed by considering the timestamps affixed by the blockchain concerning the data sampling dates.

From the economic analysis, it was necessary to use a compression tool initially identified through the use of the Python zlib library, which allowed us to save about 50% compared to sending data

without optimization. Given the extreme variability of the weight of the data to be sent, the Ethereum price and blockchain congestion, the use of a novel IPFS protocol was considered.

This tool has allowed the data amount reduction to be sent and, subsequently, the transaction's cost, optimizing the economic aspect of this tool. After writing the data in the blockchain through the transaction recovery and management system, it was possible to process and represent information regarding the measured parameters.

References

1. Giuliano, A.; Catizzone, E.; Barisano, D.; Nanna, F.; Villone, A.; De Bari, I.; Cornacchia, G.; Braccio, G. Towards methanol economy: A techno-envoronmental assessment for a bio-methanol OFMSW/Biomass/Carbon Capture-based integrated plant. *Int. J. Heat Technol.* **2019**, *37*, 665–674. [CrossRef]

2. European Environment Agency. Air quality in Europe—2019. Available online: https://www.eea.europa.eu/publications/air-quality-in-europe-2019 (accessed on 10 September 2020).

3. Sofia, D.; Giuliano, A.; Gioiella, F.; Barletta, D.; Poletto, M. Modeling of an air quality monitoring network with high space-time resolution. In *Computer Aided Chemical Engineering; Proceedings of the 28th European Symposium on Computer Aided Process Engineering*; Friedl, A., Klemeš, J.J., Radl, S., Varbanov, P.S., Wallek, T., Eds.; Elsevier: Amsterdam, The Netherlands, 2018; Volume 43, pp. 193–198. [CrossRef]

4. Buonanno, G.; Stabile, L.; Morawska, L. Personal Exposure to Ultrafine Particles: The Influence of Time-Activity Patterns. *Sci. Total Environ.* **2014**, *468–469*, 903–907. [CrossRef] [PubMed]

5. Morakinyo, O.M.; Mukhola, M.S.; Mokgobu, M.I. Ambient Gaseous Pollutants in an Urban Area in South Africa: Levels and Potential Human Health Risk. *Atmosphere* **2020**, 11. [CrossRef]

6. World Health Organization. *Ambient Air Pollution: A Global Assessment of Global Exposure and Burden of Disease*; WHO: Geneva, Switzerland, 2016.

7. Nguyen, T.H.; Nagashima, T.; Doan, Q.-V. Air Quality Modeling Study on the Controlling Factors of Fine Particulate Matter (PM2.5) in Hanoi: A Case Study in December 2010. *Atmosphere* **2020**, *11*, 733. [CrossRef]

8. Sofia, D.; Gioiella, F.; Lotrecchiano, N.; Giuliano, A. Cost-benefit analysis to support decarbonization scenario for 2030: A case study in Italy. *Energy Policy* **2019**, *173*. [CrossRef]

9. Sofia, D.; Giuliano, A.; Gioiella, F. Air Quality Monitoring Network for Tracking Pollutants: The Case Study of Salerno City Center. *Chem. Eng. Trans.* **2018**, *68*, 67–72. [CrossRef]

10. Sofia, D.; Lotrecchiano, N.; Giuliano, A.; Barletta, D.; Poletto, M. Optimization of number and location of sampling points of an air quality monitoring network in an urban contest. *Chem. Eng. Trans.* **2019**, *74*, 277–282. [CrossRef]

11. Gao, H.; Yang, W.; Yang, Y.; Yuan, G. Analysis of the Air Quality and the Effect of Governance Policies in China's Pearl River Delta, 2015–2018. *Atmosphere* **2019**, *10*, 412. [CrossRef]

12. Lotrecchiano, N.; Gioiella, F.; Giuliano, A.; Sofia, D. Forecasting Model Validation of Particulate Air Pollution by Low Cost Sensors Data. *J. Model. Optim.* **2019**, *11*, 63–68. [CrossRef]

13. Languille, B.; Gros, V.; Bonnaire, N.; Pommier, C.; Honoré, C.; Debert, C.; Gauvin, L.; Srairi, S.; Annesi-Maesano, I.; Chaix, B.; et al. A methodology for the characterization of portable sensors for air quality measure with the goal of deployment in citizen science. *Sci. Total Environ.* **2020**, *708*, 15. [CrossRef]

14. Lotrecchiano, N.; Sofia, D.; Giuliano, A.; Barletta, D.; Poletto, M. Real-time On-road Monitoring Network of Air Quality. *Chem. Eng. Trans.* **2019**, *74*, 241–246. [CrossRef]

15. Weijers, E.P.; Khlystov, A.Y.; Kos, G.P.A.; Erismana, J.W. Variability of particulate matter concentrations along roads and motorways determined by a moving measurement unit. *Atm. Environ.* **2004**, *38*, 2993–3002. [CrossRef]

16. Bukowiecki, N.; Kittelson, D.B.; Watts, W.F.; Burtscher, H.; Weingartner, E.; Baltensperger, U. Real-time characterization of ultrafine and accumulation mode particles in ambient combustion aerosols. *J. Aer. Sci.* **2002**, *33*, 1139–1154. [CrossRef]

17. Bukowiecki, N.; Dommen, J.; Preˆvoˆt, A.S.H.; Richter, R.; Weingartner, E.; Baltensperger, U. A mobile pollutant measurement laboratory—Measuring gas phase and aerosol ambient concentrations with high spatial and temporal resolution. *Atm. Environ.* **2002**, *36*, 5569–5579. [CrossRef]

18. Bumblauskas, D.; Mann, A.; Dugan, B.; Rittmer, J. A blockchain use case in food distribution: Do you know where your food has been? *Int. J. Inf. Manag. Sci.* **2019**, *52*, 2020–102008. [CrossRef]

19. Zhao, G.; Liu, S.; Lopez, C.; Lu, H.; Elguet, S.; Chen, H.; Boshkosk, B.M. Blockchain technology in agri-food value chain management: A synthesis of applications, challenges and future research directions. *Comput. Ind.* **2019**, *109*, 83–99. [CrossRef]

20. Tanwar, S.; Parekh, K.; Evans, R. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *J. Inf. Secur. Appl.* **2020**, *50*, 102407. [CrossRef]

21. Liu, X.L.; Wang, W.M.; Guo, H.; Barenji, A.V.; Li, Z.; Huang, G.Q. Industrial blockchain based framework for product lifecycle management in industry 4.0. *Rob. Comp. Int. Man.* **2020**, *63*, 101897. [CrossRef]

22. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. *J. Par. Distr. Comp.* **2019**, *134*, 180–197. [CrossRef]

23. Wei, P.C.; Wang, D.; Zhao, Y.; Kumar, S.; Tyagi, S.; Kumar, N. Blockchain data-based cloud data integrity protection mechanism. *Fut. Gen. Comp. Syst.* **2020**, *102*, 902–911. [CrossRef]

24. Benedict, S.; Rumaise, P.; Kaur, J. IoT Blockchain Solution for Air Quality Monitoring in SmartCities. In Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019; pp. 1–6. [CrossRef]

25. Yusuf, A.A.; Kurnia Basuki, D.; Sukaridhoto, S.; Pratama, Y.P.; Bramasta Putra, F.; Yulianus, H. ArmChain— A Blockchain Based Sensor Data Communication for the Vehicle as a Mobile Sensor Network. In Proceedings of the International Electronics Symposium (IES), Surabaya, Indonesia, 27–28 September 2019; pp. 539–543. [CrossRef]

26. Niya, S.R.; Jha, S.S.; Bocek, T.; Stiller, B. Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and LoRaWAN. In Proceedings of the NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–4. [CrossRef]

27. Wang, Q.; Zhu, X.; Ni, Y.; Gu, L.; Zhu, H. Blockchain for the IoT and industrial IoT: A review. *Internet Things* **2020**, *10*. [CrossRef]

28. Nyaletey, E.; Parizi, R.M.; Zhang, Q.; Choo, K. BlockIPFS—Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability. In Proceedings of the IEEE International Conference on Blockchain, Atlanta, GA, USA, 14–17 July 2019.

29. Easley, D.; O'Hara, M.; Basu, S. From mining to markets: The evolution of bitcoin transaction fees. *J. Financ. Econ.* **2019**, *134*, 91–109. [CrossRef]

30. De Tazoult, C.T.; Chiky, R.; Foltescu, V. A Distributed Pollution Monitoring System: The Application of Blockchain to Air Quality Monitoring. In *Computational Collective Intelligence ICCCI 2019, Lecture Notes in Computer Science*; Nguyen, N., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B., Eds.; Springer: Cham, Switzerland, 2019; Volume 11684.