




Article

# A Smart Water Metering Deployment Based on the Fog Computing Paradigm

Dimitrios Amaxilatis <sup>1,\*</sup>, Ioannis Chatzigiannakis <sup>2</sup>, Christos Tselios <sup>3</sup>,  
Nikolaos Tsironis <sup>1</sup>, Nikos Niakas <sup>4</sup> and Simos Papadogeorgos <sup>4</sup>

<sup>1</sup> SparkWorks ITC Ltd, Derbyshire DE11 8HS, UK; ntsironis@sparkworks.net

<sup>2</sup> Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome, 00185 Roma, Italy; ichatz@diag.uniroma1.it

<sup>3</sup> Department of Electrical and Computer Engineering (ECE), University of Patras, 265 04 Patras, Greece; tselios@ece.upatras.gr

<sup>4</sup> Power Made SA, 14568 Kryoneri, Greece; ntniakas@gmail.com (N.N.); s.papadogeorgos@gmail.com (S.P.)

\* Correspondence: d.amaxilatis@sparkworks.net; Tel.: +44-0161-818 7082

† Current address: 74 High Street, Swadlincote, Derbyshire DE11 8HS, UK.

Received: 12 February 2020; Accepted: 7 March 2020; Published: 13 March 2020



**Featured Application:** Smart water metering enabling continuous, on-demand and bidirectional data exchange between metering devices, water flow equipment and utilities within a smart city context.

**Abstract:** In this paper, we look into smart water metering infrastructures that enable continuous, on-demand and bidirectional data exchange between metering devices, water flow equipment, utilities and end-users. We focus on the design, development and deployment of such infrastructures as part of larger, smart city, infrastructures. Until now, such critical smart city infrastructures have been developed following a cloud-centric paradigm where all the data are collected and processed centrally using cloud services to create real business value. Cloud-centric approaches need to address several performance issues at all levels of the network, as massive metering datasets are transferred to distant machine clouds while respecting issues like security and data privacy. Our solution uses the fog computing paradigm to provide a system where the computational resources already available throughout the network infrastructure are utilized to facilitate greatly the analysis of fine-grained water consumption data collected by the smart meters, thus significantly reducing the overall load to network and cloud resources. Details of the system's design are presented along with a pilot deployment in a real-world environment. The performance of the system is evaluated in terms of network utilization and computational performance. Our findings indicate that the fog computing paradigm can be applied to a smart grid deployment to reduce effectively the data volume exchanged between the different layers of the architecture and provide better overall computational, security and privacy capabilities to the system.

**Keywords:** Internet of Things; smart sensors; smart actuators; water metering; smart grid; real-world deployment; experimentation

## 1. Introduction

The concept of a smart water grid refers to augmenting existing water distribution grids by introducing continuous, on-demand and bidirectional data exchange between metering devices, water flow equipment, utilities and end-users [1]. Smart water grids that integrate Information and Communication Technologies (ICT) in water systems can bring benefits analogous to those of

smart electrical grids and improve management in risk-minimization for water infrastructure [2,3]. Smart metering deployments are considered of paramount importance towards the realization of smart cities as they provide multiple benefits to water utilities and end-users. Compared to traditional, manual, metering devices, smart water meters (a) provide accurate data collection also during low water flows, (b) measure backflow, which can indicate a problem in the system, and (c) are less susceptible to corrosion from particles in the system [4]. The data collected from the resulting infrastructure enables the analysis of water demand, which helps better comprehend water end-use and in this way influence the design of urban water supply networks [5].

The availability of real-time data at high temporal frequency can help water utilities identify leaks and fixture malfunctions, timely schedule maintenance or upgrades of the infrastructure, and ultimately help them meet goals for sustainable water use [6]. Such knowledge extracted from the data can also be used in conjunction with water consumption demand models to identify the factors contributing to peak demand [7]. Moreover, advanced optimization techniques can be employed to generate water savings throughout the urban water life cycle process resulting in operational efficiencies for the owner of the water distribution network [8]. Moving beyond the use of data, recent developments in smart networked actuators help create autonomous smart grids, where metering and actuation devices cooperate to manage the distribution network more efficiently than any manned service could ever do [9]. A detailed review of existing smart water grids and the functionalities provided for water utilities was presented in [10].

The deployment of smart water grids offers many benefits also to the end-users, that is, the clients of water utilities [3]. The real-time data collected on water quality and the consumption along with the respective knowledge extracted are used to develop environmental awareness and organize campaigns regarding the use of scarce resources such as freshwater, targeting mainly young people and their everyday habits [11]. Schools are the centre of such acts, using the data provided by real-world smart grids and Internet of Things (IoT) devices. The outcomes of these acts are studied with regards to the behavioural changes achieved in the direction of promoting sustainability on various ongoing research projects [12]. Easy to implement, water quality monitoring solutions allow us to develop education-related applications and scenarios that help achieve such goals [13]. Mainly IoT technologies initiatives can benefit from immediate, direct, feedback regarding the actions performed and the solutions applied [14].

The design and development of open and expandable end-to-end information-based grid management systems is a challenge. A vision of how smart technologies can be implemented at several scales and combined to contribute to more sustainable and resilient water systems was presented in [15]. From an ICT perspective, one of the key challenges is the very large volume of data generated by the smart meters and how it is effectively communicated across all elements of the systems [16]. The multi-modal contexts of city-scale applications and the need for data access between different organizations introduce further technical challenges on transferring massive datasets collected across disparate infrastructures [17,18].

During the past few years, a number of pilot deployments have been realized that constitute the first smart water grids, allowing water utilities to improve optimization of system operation, manage leakage control more effectively and reduce the duration and disruption of repairs and maintenance. WaterWiSeis an integrated, end-to-end platform for real-time monitoring of water distribution systems deployed in down town Singapore [19]. More recently, a pilot smart water grid was deployed in Mori, a village in the East Godavari district in Andhra Pradesh situated near the Bay of Bengal. The water grid management system deployed involves different sensors that measure the quality of water, provide an alert mechanism that notifies the different levels of authorities and allow remote control of the locks that have been employed in and around the village to control the flow of water in a timely manner [20]. An inexpensive, open-source, water metering system was deployed at the Utah State University to help to measure water use quantity and behaviour at high temporal frequency [6]. These pilot studies provide interesting indications of how smart water grids can improve the operational efficiency of

the water supply system in the city and rural locations. They provide clear evidence that integrated, on-line decision-support systems based on continuous in-network monitoring of hydraulic and water quality parameters can help water utilities better manage the ageing water distribution infrastructures that encounter failures with increasing frequency and also end-users optimize water consumption and adopt environmentally friendly behaviours.

Today, the dominant architecture implemented in smart grid deployments is cloud-based, where collected data are transmitted to cloud services for storage, integration and big data analysis. A detailed analysis of the enabling technologies for the smart water quality monitoring system related to data acquisition, data transmission and data processing was explored in [21]. Unfortunately, cloud data centre deployment around the globe is conducted in a centralized way, rendering it geographically secluded for large numbers of end-users. Although providers have intensively tried to increase their Points of Presence (PoP) and seamlessly unify their infrastructure using ultra-high-speed connections, users are still experiencing increased round-trip delay, network congestion, as well as service quality degradation. As anticipated, these constitute compromising factors for latency-sensitive applications, which need to be handled in a real-time manner. In addition, due to the rapid growth in a number of interconnected devices, their diverse characteristics and the minimum functional requirements, it is of paramount importance to introduce a novel, most probably heterogeneous, architecture to tackle some of the fundamental functionality issues and shortcomings of the existing ecosystem.

Very recently, an alternative approach has been proposed, so-called fog computing, which utilizes the computational resources already available throughout the network infrastructure to enable a fine-grained analysis of the smart meter data in a more optimized manner [22,23]. The recent developments of the upcoming Fifth-Generation communication network backbones (5G) are expected to influence the development of fog computing with huge benefits with regards to response times, transmission delays, and energy management costs in delay-sensitive applications [24]. The hierarchical structures of cloud-fog computing can provide different types of computing services that improve resource management in smart grids [25,26]. Evidence from real-world deployments indicates the benefits of middleware technologies of storing and processing smart meter data within the smart devices and the network hierarchy [27–30]. Moreover, fog computing allows application developers to provide analytics and real-time actionable data insights directly from IoT end-devices with minimal data exchanges (on-site) and low latency, using user-specific resources [31]. Consider that even a simple water usage counter as a real-time, direct, feedback mechanism can significantly influence user behaviour in terms of water usage, leading to a modified, more sustainable, behaviour [6].

It is evident that the private nature of water consumption data is of paramount importance in smart metering applications [32,33]. In current smart grid deployments where meters transmit all measurements to cloud-based services, privacy requirements are dealt with as an after-thought. It is evident that in doing so, personal information can be extracted from the fine-grained measurements [6,34]. User lifestyle profiles can easily be extracted from the detailed consumption data collected, generating information about, but not limited to, house occupancy, meal times, working hours or even religious habits [35,36]. One approach is to use a secure data aggregation scheme for smart grids that follows the fog computing paradigm to ensure the privacy of collected data [37]. Studies indicate that low-end devices participating in a fog computing architecture are capable of executing advanced cryptographic mechanisms in an energy-efficient manner [38]. Utilizing such mechanisms can help increase the privacy of the user's data while also reducing the communication and storage overhead. The hierarchical-nature of the fog computing architecture in combination with additive homomorphic encryption protects consumer privacy from third parties. Alternatively, differential privacy techniques can be applied to develop effective privacy-preserving schemes for load monitoring. Recently, new consumption behaviour models have been developed that take into consideration the fog computing elements of the architecture to add noise to the data collected at specific points, thus providing a better trade-off between the utility of the data and privacy compared with other popular methods [39].

In this work, we present the design and implementation details of a smart water grid system that follows the fog computing approach. A series of intermediate layers is strategically introduced in the computing and communication infrastructure that interconnects the secluded smart devices, the cloud services and the end-user equipment. The resulting system provides a fully functional edge-processing prototype for obtaining, analysing, storing and efficiently retrieving datasets from smart, off-the-shelf water consumption meters and water valve controllers. The benefits of using fog computing with regards to real-time analysis needs are summarized as follows:

**Reduction of bandwidth requirements:** The packets collected from an installation of smart water metering devices in a relatively small-sized city comprised of 50,000 houses can accumulate up to 13 GB per day [40]. This constitutes a load above the capacity of currently available narrow-band wireless networking technologies [41]. Real-world evidence of deployments that adopt the fog computing paradigm indicates that the load can be reduced up to a level of 24MB per day by using the processing capabilities already available at the edges of the network [40].

**Improved responsiveness and reliability:** A significant reduction in the wastage of resources can be achieved through fine-grained monitoring of consumption and relaying of this information back to the consumers [42]. Multiple studies suggest that real-time, direct, feedback mechanisms on the level of consumption can help to reduce consumption when compared to traditional, indirect mechanisms such as, e.g., bimonthly bills or periodic advice on resource consumption [43]. The ability to process data at the edges of the network and the automatic analysis of water consumption data at a local level allow providing direct feedback based on the actual consumption without the need to rely on data available on the cloud. Moreover, the operation of safety-critical systems, such as water pressure valves, is always operational and secure besides any potential and unexpected connectivity issues.

**Improved privacy:** Detailed and continuous sensing of water consumption raises various privacy concerns since it is possible to identify certain actions of the users while analysing the data from a central point [36,44,45]. Following the fog computing paradigm, detailed data collected from the smart water metering devices are analysed at the nearest available edge-based processing point, and only aggregated information reaches the cloud servers. In this way, disaggregation techniques that rely on detailed, real-time data cannot be effectively used from a central point of view, thus significantly improving the privacy of the end-users.

The resulting system is deployed in a city environment and evaluated in real-world conditions. The performance of the system is evaluated in terms of (a) network performance, (b) responsiveness and computational capabilities at the network's edges, (c) I/O operations on the edges of the network and (d) data encryption and decryption capabilities and resilience against threats faced.

## 2. Technical Enablers

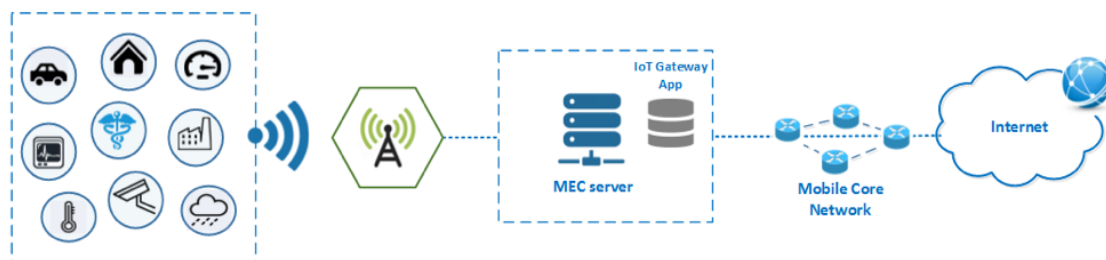
### 2.1. Deploying Intelligence on the Network Edge

Edge computing, as originally envisioned in [46], is a major concept targeting reduced latency and communication delays. The fundamental idea of edge computing is to place computational resources closer to the raw data sources and users. To this end, edge computing [47] integrates components categorized as end-user equipment (e.g., smartphones), edge networking devices (e.g., border routers, bridges, base stations, wireless access points) and edge servers. This approach bounds the specific solution to the localized computing paradigm, but at the same time, it provides faster response to computational service requests and consequently eliminates the need for bulk raw data transfer to distant data centres located on the network core. Since it is indeed a step towards the right direction, certain deployment paradigms have been introduced in an attempt to combine the prime aspects of edge and cloud computing, namely Multi-access Edge Computing (MEC) and fog computing.

### 2.1.1. Multi-Access Edge Computing

Multi-access Edge Computing (MEC) is an architectural paradigm that aims to deliver some of the capabilities of cloud computing bundled with an IT service environment, within the Radio Access Network (RAN). This approach introduces a processing environment closer to service subscribers, characterized by ultra-low latency, high bandwidth, real-time access to radio network and context information, location awareness, efficient network operation and enhanced service delivery. The elevated Quality of Experience (QoE) levels [48] for all end-users create additional value through novel application delivery, thus introducing many advantages for all stakeholders.

The design philosophy behind MEC renders it flexible enough to play numerous roles in use cases related to edge applications with infrastructure similar to commodity servers and intelligent networking appliances. The European Telecommunications Standards Institute (ETSI), an independent, not-for-profit, standardization organization in the telecommunications industry, recently issued such a scenario in which MEC servers deployed at an LTE base station can be utilized for delivering IoT applications, essentially acting as a low latency aggregation point for managing communication protocol enforcement, rapid message distribution and analytics processing (see Figure 1). Through the proposed blueprint, MEC enables the aggregation and distribution of IoT services for real-time responses inside a highly distributed environment. Data round trip time is reduced for the core network, and the cloud applications are abstracted in a new single layer.



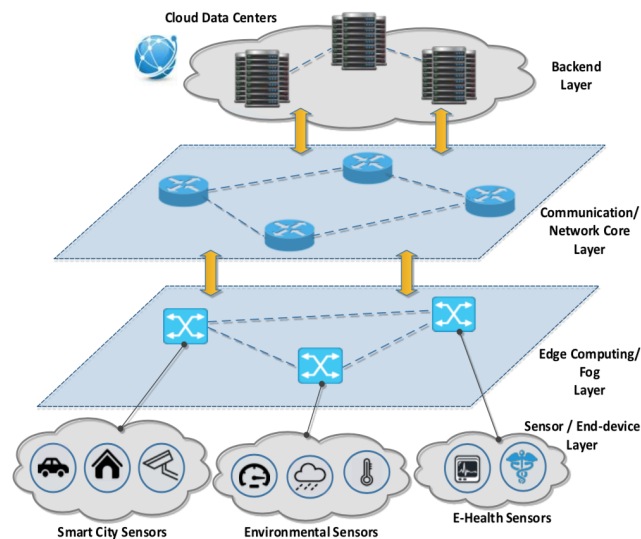
**Figure 1.** European Telecommunications Standards Institute (ETSI) use case blueprint in which a Multi-access Edge Computing (MEC) node is deployed as the IoT gateway.

### 2.1.2. Fog Computing

Fog computing provides computational resources, storage and control to consumers using an intermediate layer between remote and isolated traditional data centres and on-premise equipment or IoT sensors/devices. By exploiting resources available on the network edge, end-user and sensor-oriented datasets are analysed on the spot, eradicating the need for transferring bulk data to backend data centres, but only transmitting aggregated information for permanent storage or specialized more complex analysis.

Figure 2 presents a simplified fog computing topology and identifies the most important layers of the overall communication stack. Cloud data centres reside in the backend layer and have direct access to the network core layer, which is dedicated to perform tasks related to routing and switching and act as the gateway of cloud data centres to the outside world. The fog layer is placed between the network core and the sensor/end-device layer and is capable of communicating with both. In contemporary deployments, it is expected that Software-Defined Networking (SDN) nodes are used as part of the network core layer to allow for extensive governance and precise supervision of the network [49,50]. The introduction of an intermediate layer between the loosely supervised and easily compromised sensor/end-device layer also improves security since packets are not directly forwarded to the cloud services of the platform, but they are inspected and cleared from malicious, harmful or problematic content throughout their journey in the network. This tackles issues generated due to limited computational and energy capacity that hinder the use of data security and privacy mechanisms [38,51]. Yet, the most significant contribution of fog computing is the inherent solution

it introduces to specific IoT-related constraints and limitations, which could otherwise compromise ubiquitous service delivery for a wide variety of applications.



**Figure 2.** Introducing fog computing in the communication stack [52].

**Massively inflated bandwidth requirements:** The remarkable growth of IoT and its transformation into an ecosystem that supports information aggregation from billions of nodes generate a data-related problem. The interconnected sensors and all end-user devices constantly collect many data that are then forwarded to the network backend for processing. This approach is, by all means, ineffective as it requires huge bandwidth and does not guarantee that transmitted data will be meaningful, valid or worthy of storage for future reference. It is often the case that networking resources are allocated for pushing null or corrupted datasets to the cloud, while at the same time, the data aggregation rate constantly rises. This renders data pre-processing at the network edge more or less compulsory since the consequent trimming will reduce bandwidth requirements, network traffic cost and necessary cloud storage [53]. As stated in [52], cloud-based provisioning of resources could be simplified by using dedicated fog computing nodes where only valid data are processed for much lower networking costs.

**Contemporary application requirements for decreased latency and semi-autonomous operation:** The constantly increasing numbers of interconnected nodes introduce severe challenges for cloud deployments aiming to provide uninterrupted services since connectivity irregularities will statistically become more often and much more difficult to predict [54,55]. The anticipated advent of 5G [56,57] will resolve some of the most common causes of compromised networking, but since low latency and guaranteed robustness are prerequisites for existing infrastructure as well, one should examine fog computing as a method for addressing these issues. This problem becomes more obvious when considering the functional requirements of many industrial or safety-critical systems such as patient monitoring platforms [58–60], automated production lines and road traffic optimization applications [61], which only tolerate just milliseconds of end-to-end latency and therefore remain based on obsolete architectures. In addition, a certain type of autonomous operational capacity is important for reaching the aforementioned level of service continuity. All safety-critical systems need to stay operational and secure besides any potential and unexpected connectivity issues. Data reporting is to be restored, as soon as connectivity is available, as normal. This example showcases the desired behaviour of a dedicated node in such unfortunate, but definitely expected situations. Sensor data need to be temporarily stored and even pre-processed in the intermediate layers of the application producing notifications and warnings as normal. Preliminary system responses are generated at a smaller distance from the data producers and in a more direct manner without the need for always-on connectivity.

Enhanced reliability and platform security prerequisites: As the actual size of the transmitted data increases, so do the error rates, packet droppings and transmission times. However, increased error rates cannot be tolerated in the case of emergency and safety-critical applications. Successful IoT deployments need to rely on uninterrupted service delivery in combination with elevated security. As the fog computing layer is placed between the cloud and end-user devices, it provides the necessary guarantees to ensure service cohesion [55]. Its nodes are intermediates for providing security and sensor management capabilities, such as deep packet inspection or message encryption and benefiting from context and location-related information to easily detect threats [62]. Without a doubt, resource-constrained device protection and security update delivery to the very end of distributed system nodes, in a trustworthy manner without causing intolerable disruptions, increase the overall efficiency and performance of the system to a whole new level [63].

In conclusion, fog computing is an architecture that distributes computation, communication, control and storage closer to the end-users, along the cloud-to-things continuum [64], and will at the same time unveil a novel architectural concept that will most likely also enable fascinating business models for computing and networking.

## 2.2. Deploying Low-Power Long-Range Networks

An essential part of smart grid infrastructure is wireless networking protocols and the underlying technologies used for interconnecting smart meters, the network backbone and cloud services deployed over the Internet. Several wireless communication technologies have been proposed during the previous years to facilitate communication within a smart city deployment. These networking technologies mainly focus on increasing network throughput instead of reducing power consumption [65]. More recently, various networking technologies have been proposed that focus on wireless low-power transmissions, such as IEEE 802.15.4 (ZigBee, Z-Wave) and IEEE 802.15.11 (Bluetooth, BLE) [66]. When it comes to smart water metering devices, the Open Meter Standard (OMS) Wireless M-Bus protocol was designed to accommodate specialized industry-grade deployments, operating at the 169 and 868 MHz Industrial, Scientific and Medical (ISM) frequencies [67], with adequately high bit rate exchange capability in a short range. Due to the combination of low-power and short-range operational design, large-sized network deployment requires the combination of protocols that follow the multi-path approach [68] to achieve seamless and error-free message delivery. This approach provides a series of benefits like the ability to overcome obstacles in communication [69,70] and, most importantly, enhanced network security [71]. However, recent trials carried out in real-world wireless sensor networks have highlighted certain difficulties and limitations of the multi-hop, short-range connectivity paradigm [72]. Thus said, a series of alternative solutions was proposed specifically to mitigate these limitations, for instance deploying nodes with different transmission ranges [70], enforcing a hierarchical network structure [73] or using dedicated mobile nodes for managing the network [74]. Unfortunately, all these efforts to overcome the difficulties of the reduced transmission range of communication interfaces result in complex real-world deployments that span over multiple networking technologies in order to provide urban-scale coverage [75,76].

Exploiting sub-GHz communication bands, which allow transmission over longer distances, paired with very low data transmission rates, has been the most recent proposition to enable a significant decrease in power consumption [77] and reduce the complexity of urban-scale deployments. Such communication technologies are referred to as Low-Power Wide Area Networks (LPWANs) as opposed to other short-range high-frequency communication bands. Low-frequency signals are significantly less attenuated by multi-path propagation or physical obstacles compared to the high-frequency ones, thus increasing both signal reliability and robustness [69,78–80]. In LPWAN, concentrators (also called a collector) play an important role by collecting information from all embedded devices located several kilometres away. LPWANs provide huge benefits for IoT

deployments, including higher autonomy due to lower energy requirements and deployment costs as only a few concentrators can cover large geographical areas [26,78].

LPWAN communication is based on proprietary modulation techniques that have evolved from the Chirp Spread Spectrum (CSS) and operate in the sub-GHz bands. This allows for independent and asynchronous communication between nodes, partially resembling an ALOHA protocol, that result in an LPWAN concentrator capable of receiving data originating from multiple IoT devices and thus increasing the density of nodes in an LPWAN. Alas, due to unlicensed sub-GHz bands' regulation restrictions, LPWAN may end up in an asymmetric situation with concentrators directly connected to many IoT devices being somehow compromised in terms of functionality compared to less central concentrators connected to only a handful of IoT nodes. Existing designs dictate that IoT devices and concentrators should only transmit at about 1% of their total uptime to maintain a low power consumption while supporting high deployment numbers. Moreover, since the low-power operating mode is essential for IoT nodes, being constantly attached to a communication channel for down-link messages is not an option and is carried out only occasionally. In order to accommodate all the needs of the future, very dense, LPWAN deployments specifically designed as extensions need to be proposed and implemented [26,81,82].

The suitability of LPWAN technologies to interconnect smart water metering devices with cloud-based services has been evaluated in real-world deployments [20]. The evaluation of the LPWAN technology was based on a pilot deployment of different sensors deployed at various rural locations near to the Bay of Bengal to measure the quality of water by generating real-time data. The system enabled bi-directional communication to accommodate both sensing and control of the flow of water in a timely manner. In the system presented in this paper, although using similar LPWAN technologies, the data collected from the sensors and the decisions to control the flow of water do not take place at the cloud part of the system. In contrast to [20], the system presented here follows the fog computing paradigm, and in this way, the data are analysed at the edges of the network. For a comparative evaluation of the leading low-power, wide area network technologies useful for developing IoT networks in terms of a battery lifetime, cost, network coverage, latency, range and security, the reader is pointed to [83].

### 2.3. Deploying in Heterogeneous Computing Environments

The main benefit of the development of IoT enabled systems that are cloud-centric is the simplicity of their design with straight-to-cloud data collection and central processing. During the previous years numerous platforms have been presented, such as Amazon Web Services (AWS) IoT (<https://aws.amazon.com/iot/>), Google Cloud IoT (<https://cloud.google.com/solutions/iot/>) or Microsoft Azure IoT (<https://azure.microsoft.com/en-us/overview/iot/>). These cloud-based platforms provide a uniform and homogeneous computing environment that allows developers to focus on the analysis of the data and the integration of the extracted knowledge with existing enterprise infrastructures.

Systems that are designed using the fog computing paradigm inherently incorporate different platforms with heterogeneous computing, processing and networking capabilities. In such deployments, heterogeneous computing elements usually contain one or more CPUs, each one with a set of computing cores, and possibly also a GPU [84]. Moreover, the physical implementation of MEC within the RAN currently integrates more and more heterogeneous hardware, such as Field Programmable Gate Arrays (FPGAs) in their servers, enabling their clients to accelerate their applications via programmable hardware. Ideally, programmers want to make use of as many resources as possible, primarily to increase performance and save energy [85]. However, contemporary systems do not provide a homogeneous programming environment that at the same time abstracts the heterogeneous hardware characteristics and yet allows taking advantage of all heterogeneous resources available within a computer system [84,86].

Some of the acclaimed frameworks for stream-processing and big data analytics such as Apache Flink [87], Spark [88] and Storm [89] are implemented on top of Java Virtual Machines (JVMs), due to



their fundamental requirements for portability and interoperability with commonly used high-level programming languages. These frameworks became popular partially because they are able to operate on a large variety of platforms and operating systems, despite the fact that most production JVMs only generate CPU-oriented code. Consequently, the task of creating code suitable for execution on heterogeneous devices such as GPUs or FPGAs falls back to the developers. Despite the fact that this approach is common in several frameworks such as HadoopCL [90], HeteroSpark [91] and Glasswing [92], there are certain omnipresent disadvantages that compromise any chance of a wider adoption or applicability [93].

**Code fragmentation:** Every developers' code base needs to provide inherent support for more than one programming language while at the same time integrate several different programming models [90–92]. For instance, in an attempt to provide out-of-the-box GPU acceleration, developers need to consolidate Scala and Java source code with low-level APIs such as CUDA or OpenCL [94]. This poses great programmability, as well as maintainability challenges since it demands familiarity with a variety of different concepts, APIs and tool-chains. Developers with such a broad spectrum of expertise are substantially hard to find.

**Lack of code portability:** Given the fact that code segments are optimized for a particular device or hardware [95], migrating to a new cloud ecosystem, hardware architecture or sometimes to similar devices having different firmware requires extensive source code porting. This problem partially derives from the fact that low-level programming models used in heterogeneous accelerator programming do not comply with the “write-once-run-anywhere” development paradigms of high-level programming languages like Java; however, it remains a reality that must be properly considered.

**Lack of dynamic reconfiguration:** JVM cannot reconfigure accelerated code segments on runtime [91,92,95], a limitation with severe impact on the application performance along with the overall deployment cost. From the application perspective, it is only possible to utilize resources allocated in advance for the specific functionality. The ability for dynamic, ad-hoc reconfiguration at runtime is directly inherited by the underlying platform, and this is where JVM's limitations, especially when operating on heterogeneous hardware, become crystal clear. Adding heterogeneous hardware support on JVM is a highly perplexing task that must be addressed with respect to the JVM semantics [96].

As a result, with the exception of IBM's J9 GPU support [97], which only recently started accelerating Spark workloads on GPUs [98], most JVM-based solutions for heterogeneous code execution are nothing more than sophisticated prototypes, significantly far from production-ready deployments. Thus said, it is compelling for heterogeneous JVMs designed for stream-processing and large scale data manipulation to support GPUs together with application-specific accelerators, but at the same time allowing dynamic code migration across devices without a reboot [84,99].

### 3. System Architecture, Software Services and Deployment

The main entities of the fog computing-based smart water metering system presented in this paper are: (i) the smart meters deployed throughout the water distribution network that provide continuous monitoring of the water consumption and pressure; (ii) the smart water valves installed near the water meters that offer the ability to control the flow of water remotely; (iii) the edge computing infrastructure that interconnects wireless networks of smart devices with the core network, bundled with a fully customized edge processing and analytics framework for collecting and analysing the data produced; (iv) the cloud services that facilitate identity management, telemetry, asynchronous notifications and historic data storage, as well as the dedicated interfaces that will be utilized by both users and external services for getting access to the platform logs and data (e.g., for billing, etc.).

#### 3.1. Hierarchical Network Architecture

Smart devices deployed for monitoring and remotely controlling water distribution are organized in a hierarchical network architecture. Since the majority of integrated third-party smart water

monitoring solutions use wM-Bus communication interfaces, the specific protocol’s functional requirements play a fundamental role in the design of the overall architecture. Additional communication protocols are also supported rendering the solution future-proof; however, performance optimization will be carried out in the near future.

The proposed architecture introduces an intermediate layer of gateway devices, each one equipped with two distinct networking interfaces. The first interface uses the wM-Bus protocol for interacting with the affiliated smart devices, while the second interface delivers long-range low-rate communication based on LPWAN technology. This inevitably forms a certain type of network hierarchy in which (i) LPWAN gateways constitute the top layer connecting the IoT deployments with the Internet, (ii) gateways form an intermediate layer and, lastly, (iii) smart devices are placed at the bottom. An example of such a hierarchical topology is shown in Figure 3, while the architecture is aligned with the paradigm proposed in [26]. This scheme is considered appropriate due to its adaptability to various internal and external conditions, supporting scalable and robust communication. It only requires a limited degree of fine-tuning, and it is partially affected by the actual environmental conditions of the deployment.

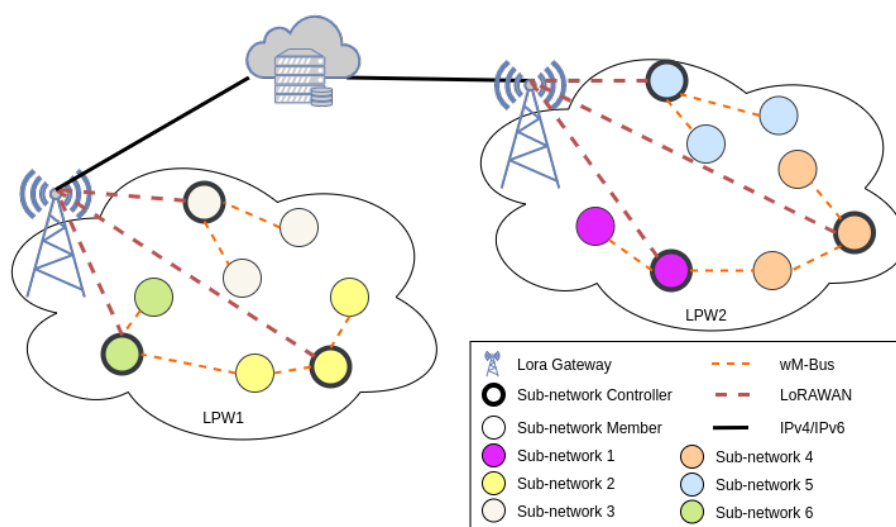


Figure 3. A hierarchical network architecture based on LPWAN and wM-Bus.

Moreover, the specific hierarchical organization structure provides a series of additional benefits such as the establishment of secure communication between sub-networks by Public Key Infrastructure (PKI) incorporation. This essentially provides an increased level of confidentiality and information integrity, while at the same time facilitating the execution of services related to data analysis at the network edge. Each transmitted message is individually encrypted, thus demanding a separate key per device to decrypt its content on the receiving end. Each device key is stored permanently on the device and cannot be changed during its lifetime (more than 10 years in most cases). The transmitted packets are collected by the gateway devices through the supported wM-Bus network interface and are forwarded via the LoRaWAN interface to the higher layers for further processing.

### 3.2. Fog Computing-Based Data Processing Hierarchy

When viewed in an abstract perspective in terms of data processing, the entities of the system can be organized into two logical layers: (i) the Sensors and Actuators Layer (SAL), comprised of the smart devices physically attached to the source of monitoring that collect the information, digitize and transmit it, and (ii) the Processing layer (PL), comprised of all the system entities that convert received bits of information into meaningful datasets.

The second layer is organized into additional sub-layers based on their processing ability and the communication interface they support. From a fog computing perspective, as presented in Figure 2,

processing nodes reside in the edge/fog computing layer, while the backend layer hosts all cloud services of the prototype. This design inherently exploits the existing intermediate layer between the smart water meters and the cloud, while at the same time, the approach of having all available cloud services deployed in large backend data centres alleviates several scalability concerns along with bandwidth consumption problems, thus ensuring the seamless operation for the whole prototype. The prototype architecture blueprint is presented in Figure 4.

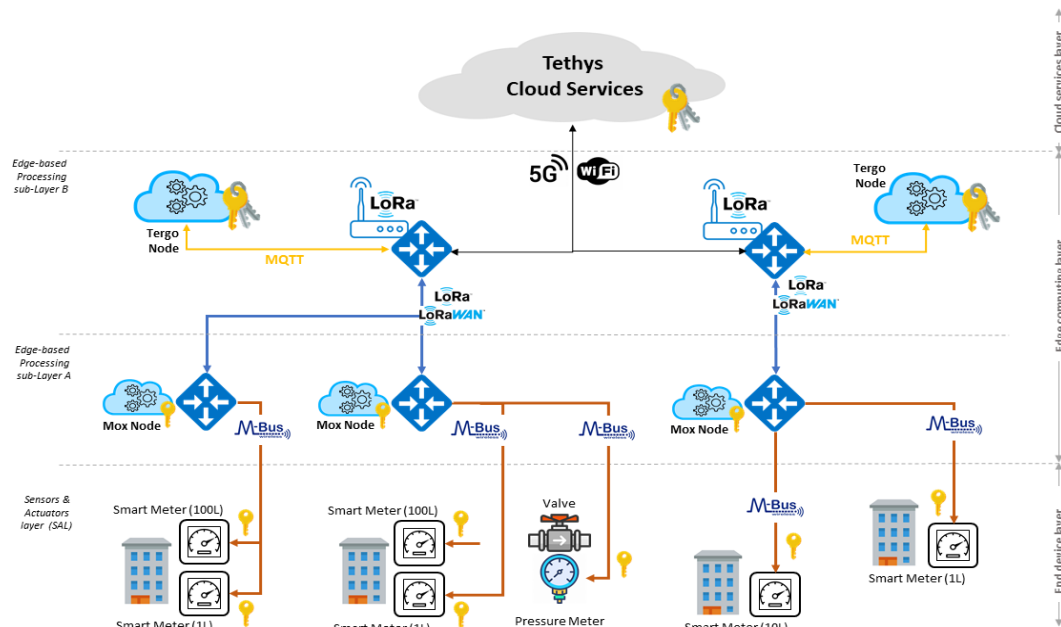


Figure 4. The fog computing-based data hierarchy.

As shown in Figure 4, the Sensing and Actuation equipment Layer (SAL) contains all the smart water sensing and control equipment, which can be off-the-shelf (i) water consumption meters, (ii) water pressure meters or (iii) remote-controlled valves. These devices broadcast their sensing data via wM-Bus over predefined time intervals and provide metrics related to (i) total water consumption, (ii) current water pressure, (iii) water and environment temperature and (iv) water valve status.

The first node equipped with data processing, storage and analysis capabilities is called the Mox node, located in the Edge-based Processing sub-Layer A (EPL-A) and interlinked with a bridge that can deliver transmitted packets for local processing through wM-Bus and LoRaWAN interfaces. Mox nodes have limited available computational resources, provide only basic data manipulation functionality and can only be used for elementary tasks. Designed for operating on resource-scarce hardware, Mox nodes integrate a trimmed-down version of the analytics software deployed in full scale into other nodes of the prototype and are therefore capable of obtaining and efficiently handling metrics from just a small number of sources. One could state that their main function is to collect packets from the smart devices, identify the source of each message and prioritize its upload to a higher capacity node located in a different layer than EPL-A. Moreover, Mox nodes can be used for remotely controlling smart meters and valves, which supports this type of functionality. Given the fact that all transmitted messages are encrypted, Mox nodes incorporate authentication mechanisms. However, due to their limited computational capacity, as well as the static topology in which they tend to operate having only specific smart water metering devices interlinked, authentication is carried out using pre-existing permanent keys, each directly related to an existing sensing node (see Section 3.1).

Data that are not processed with the Mox nodes are transferred for further processing into different, more capable nodes of the prototype, through the LoRaWAN interface, and are picked up by the LoRa gateways available in the area. After this point, messages reach the Tergo nodes, which form the so-called Edge-based Processing sub-Layer B (EPL-B) together with the affiliated LoRa gateways.

Tergo nodes have significantly improved computational capacity as they operate on better hardware than the Mox nodes and thus support a wider variety of features by integrating an almost full version of the analytics software deployed in the prototype. They are able to support services for (i) analysing incoming packet rates, as well as the signal quality from the installation's meters, (ii) key management for storing and accessing the meter's decryption keys, (iii), advanced analytics over the obtained datasets, (iv) local storage and (v) syncing data with the central cloud infrastructure. Tergo nodes support additional communication protocols such as MQTT [100], which allow direct links with the available LoRa gateways.

The top-most layer of the prototype incorporates the Tethys Cloud Services, which are designed to collect datasets from the whole infrastructure and provide APIs together with the necessary communication interfaces for accessing the aggregated data over the Internet. The cloud services reside in a layer that corresponds to the backend or Cloud Services Layer (CSL) of fog computing as presented in Figure 4 and are accessible via common IP interfaces, supported by both cellular and landline networking infrastructure. In addition, Tethys Cloud Services also integrate sophisticated authentication mechanisms for mitigating malicious attacks, which could potentially lead to sensitive data loss.

### 3.3. Data Engineering on the Fog Layer

All data messages that are accumulated in SAL are then circulated within EPL-A and EPL-B through a dedicated, highly customized message bus system that essentially distributes all obtained information to the various subsystems responsible for data processing, data storage or alert generation. The existence of a dedicated message bus submodule introduces a certain degree of flexibility allowing the integration of data transformation and processing mechanisms. These mechanisms stretch over several different hierarchical layers and accommodate all end-user services and functions. When viewed from the inside, the message bus delivers a fully autonomous stream processing pipeline compatible with all interconnected sensors and last-mile devices. Each data stream is provided with a unique tag/name resembling the network addresses of the sensors where data are originated from and extended with the unique identifier of the EPL-A device that received it.

This overall approach renders the framework capable of storing data in a scalable manner across several different architectural layers. Data can be formatted for imminent retrieval in which case time-consuming queries or aggregations are avoided, while at the same time, caching mechanisms are integrated to store recent data retrieved from specific sensors or series of sensors temporarily. Moreover, the framework also stores historical data in a more permanent repository via a dedicated process that identifies obsolete measurements and forwards them accordingly. The decoupling of data generation and storage entities facilitates the implementation of a wide variety of services in a more independent way since the necessity of optimizing code for meeting vastly different performance requirements no longer exists.

#### 3.3.1. Generic Data Processing

Once the stream processing pipeline is formally established, all inbound data are processed based on a predefined set of data processing functions. This practice renders edge devices as the "primary" processing points for all sensor-oriented data, triggering a series of steps that effectively handle and extract all meaningful information from the recently obtained datasets. The first step of the overall data processing pipeline is to track the existence of improper, fragmented or erroneous data that will adversely compromise the coherence of any time series. It is critical that outliers are identified (i.e., by checking new observation points that appear to be somehow irrelevant to normal values). The origin of these outliers often derives from transmission errors, which are sometimes frequent on the sensor device, but should be deleted on time to not "pollute" the dataset. The second step applies an ongoing, but predefined time window that allows averaging and smoothing out potential fluctuations. The third step is introduced to handle temporary sensor device disconnections, which

may lead to missing values and consequently disrupted datasets. Moreover, a dedicated algorithm for identifying data gaps and inserting mean values for complementing the datasets is also implemented.

### 3.3.2. Continuous Data Analysis

Once all values within the predefined timeslot are processed, regardless of the sensor stream they belong to, an aggregated output for a variety of functions, as well as time intervals is produced. This output contains information regarding minimum, maximum or average values in time intervals spanning from one hour up to one month, whereas customization is also possible for producing even more precise results. Additional analysis can be applied to each separate stream, allowing the framework to identify local spikes and perform basic measurements of the data slope. The resulting dataset is stored in the integrated memory of the EPL-A and is also forwarded to the cloud layer.

### 3.3.3. Application-Specific Processing

Not all end-user-facing services have common data processing tasks that are being executed at the backend. Each and every one of those tasks is specified through Java Specification Requests JSR-000335 lambda expressions (<https://jcp.org/aboutJava/communityprocess/final/jsr335/index.html>), with the specifications being pushed into the EPL-B and analysed based on two factors: (i) the sensor data that are processed and (ii) the time frame in which the data stream process must have been finalized. In addition, as part of the services delivered to end-users, the association of factors such as pressure and consumption together with the various environmental conditions must be evaluated in order to assess the efficiency, as well as the accuracy of the water distribution monitoring framework. Modelling the relationship between all types of variables can be achieved by applying linear regression methods, which in general are considered to be highly effective for the specific task.

For meeting the functional requirements of smart water metering services regarding data interpretation and analytics, customized lambda expressions for data processing and manipulation are implemented. These lambda expressions are divided into three main categories as follows:

**Expressions that involve linear regression:** Certain high-level services employ linear regression for generating effective models to identify existing relationships between scalar-dependent and explanatory variables. The majority of currently implemented models are based on simple linear regressions; however, there are cases in which multivariate linear regressions are also utilized. Such a use case may be the prediction of water consumption within a confined area of the city or the estimation of the pressure of a specific block of buildings for the foreseeable future.

**Expressions that employ clustering and classification of data:** Across the different user-facing functionalities implemented, datasets need to become as homogeneous as possible by using unsupervised learning techniques. The proposed framework addresses the specific necessity through the highly popular k-means clustering method of vector quantization and the KNN data classification method. For getting a better overview of more specific examples for this category of expressions, one could consider the fully automated characterization of the water consumption for each individual house across all buildings or water consumption classification of each building during each day/time-slot.

**Fast Fourier transform expressions:** The last group of transformation expression is directly linked to the Fast Fourier Transform (FFT) algorithms toward computing the sequence's Discrete Fourier Transform (DFT). Through Fourier analysis, it is possible to convert a signal originally in the time/space domain to accurate frequency domain representation and vice versa. Through FFT, it is possible to compute such transformations rapidly by generating the corresponding DFT matrix as a product of sparse factors. Such expressions can be used for analysing the water consumption of the building in order to track whether specific household equipment was in use within given day/time-slots. An additional example has to do with the quantification of events that lead to low water pressures.

### 3.4. Cloud-Based Services

The deployed framework supports cutting edge features that ensure full compliance with the pillars of fog computing, namely efficiency, security and augmented reliability. In particular, the framework has integrated subsystems for identity management, rapid access to historic events, real-time telemetry support. along with the inherent capability of issuing asynchronous notifications, all bundled with specialized interfaces used by authorized users and external services alike to get access to the available services.

#### 3.4.1. Identity Management

Identity management was designed to allow reliable user authentication services, taking into consideration user equipment, profile or customized security preferences. It supports secure and private mobile device authentication and dedicated web access per user taking into consideration customized profile management selections, all delivered in a way that ensures high levels of privacy regarding the disposition of personal data. The user is capable of generating a new identity only by providing a minimum set of information, which is then stored in an encrypted database. After creating the new identity, the user is able to authenticate with the identity management subsystem and becomes capable of accessing all available cloud services via the OAuth 2.0 protocol using https credentials. Moreover, since the whole prototype framework follows a micro-service-oriented architecture, user records are stored in different, yet interconnected components, which interact with each other through the unique identifier generated by the identity management subsystem. Lastly, the identity management system handles all authentication codes such as passwords or access tokens and is responsible for verifying the authentication roles for each system function.

#### 3.4.2. Historic Events

Efficient and rapid access to historical datasets is considered of paramount importance for smart water management applications, e.g., when comparing historical data from different time spans trying to identify consumption patterns or detecting anomalies related to malfunctions. The specific cloud service ensures that all datasets obtained from many different smart water measuring devices are aggregated and then stored on a common repository, following pre-defined tagging patterns, and are available in near-real-time regardless of the time window dictated by the request.

#### 3.4.3. Telemetry

The telemetry service provides remote access to real-time records and alerts issued directly by the smart water consumption metering devices. The service aggregates all data generated by the affiliated, interconnected smart devices through the nearest gateway and processes them accordingly.

#### 3.4.4. Asynchronous Notifications

The asynchronous notifications service pushes notifications associated with alerts generated by the interconnected smart metering devices to all authorized users. Authorized users may also receive a notification related to data analytics and other relevant services. It should be stated here that smart meter-related notifications are only forwarded to users in the case of an alert, indicating that a user-defined threshold for a specific metric is violated. This approach allows notification to be issued in the case of an emergency or detected anomaly, i.e., due to a node malfunction or water leakage. The asynchronous notification service renders the framework administrators capable of defining complex criteria that potentially introduce additional computational costs for efficiently handling the alerts generated by the device (e.g., leak detected). Remote users get notifications through a smartphone application that integrates the available push notification mechanism.

## 4. Evaluation and Benchmarking

We now present a set of evaluation scenarios and quantitative evidence on the performance of the system that we collected from the operation of our real-world installation present in Section 3. We split our evaluation into multiple domains, based on the technologies used and the devices involved.

### 4.1. *wM-Bus Connectivity Evaluation*

An important element of the architecture proposed in this paper is the ability of the infrastructure to collect the data produced by the smart water metering devices properly. The fog computing-based data hierarchy depicted in Figure 4 relies on the wM-Bus network to deliver the continuous water quality and usage measurements, collected from the smart devices that constitute the Sensors and Actuators Layer (SAL), to the Mox nodes. Recall from Section 3.2 that the latter are used to form the first edge-based processing sub-layer (EPL-A). In the first set of experiments, we wished to measure the performance of the wM-Bus network in terms of signal quality, the stability of the wireless communication medium and the observed delivery rate. Both the smart metering devices and the Mox nodes were fixed in specific locations, but the quality of the communication was severely influenced by external parameters and installation characteristics. In all cases, the water meters were located inside cast iron-covered inspection pits below their respective buildings, while the Mox nodes were placed in the buildings' upper floors to increase coverage as much as possible. The main hindrance for the communication was the enclosure of the device, which severely diminished the signal's quality, but as we will showcase below, they were also affected by other unpredictable events and environmental agents (e.g., birds).

Under normal conditions and without any external interventions, the RSSI of the received messages from three smart meter devices is presented in Figure 5 (Location 1). In this figure, we see the RSSI of the messages received by a single Mox node from three separate smart meters positioned in a single inspection pit of a building. The RSSI value was relatively stable throughout the four months of the evaluation period. Figure 6, on the other hand, shows us an example of an unexpected outside intervention (Location 2). We again see the RSSI of two different smart meters as they were received, again, from a single Mox node. The two smart meters were also placed in the same inspection pit, but in different orientations, thus causing a difference of at least 10 db. The sudden drop that happened almost half the time into the evaluation period was caused by a shift in the Mox node's antenna, caused by extreme wind or birds interacting with the antennas of the devices. We can also observe how the first device had more fluctuations in the RSSI values, which became even more intense as the quality of the received signal degraded.

The next two figures (Figures 7 and 8) show two cases that were also interesting to note. In Figure 7, we again present the RSSI values from two smart meters recorded from their respective Mox node (Location 3). The second device had a significantly less stable behaviour at the beginning of the evaluation, which was observed also in the first device towards the end of the period, while the RSSI values of the other device stabilized. This gave us the impression that an external agent again affected the quality of the communication, and for some reason, the effect of the agent shifted from one device gradually to the other. Finally, Figure 8 shows a similar case where in the second half of the evaluation period, the RSSI of the first device became unstable and degraded greatly, while the second one was affected at some points, but in a less intense manner (Location 4).

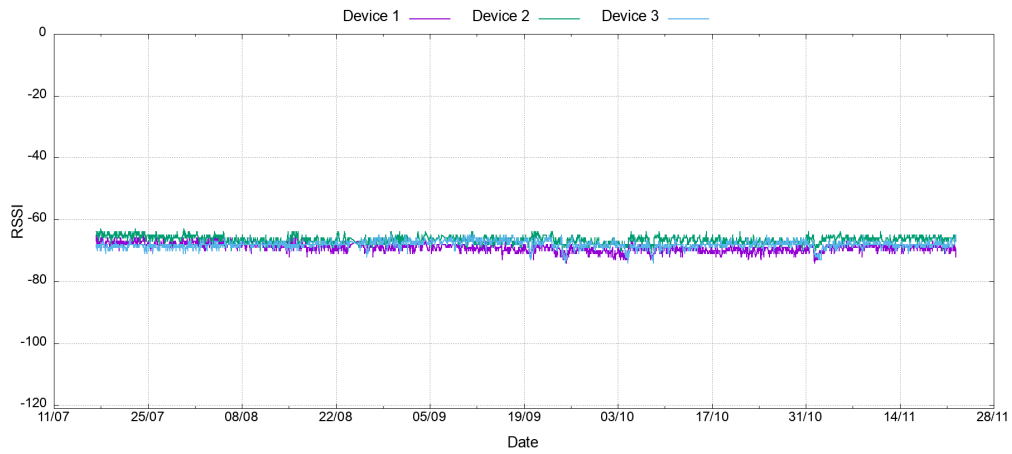


Figure 5. wM-Bus smart meter devices with stable RSSI (Location 1).

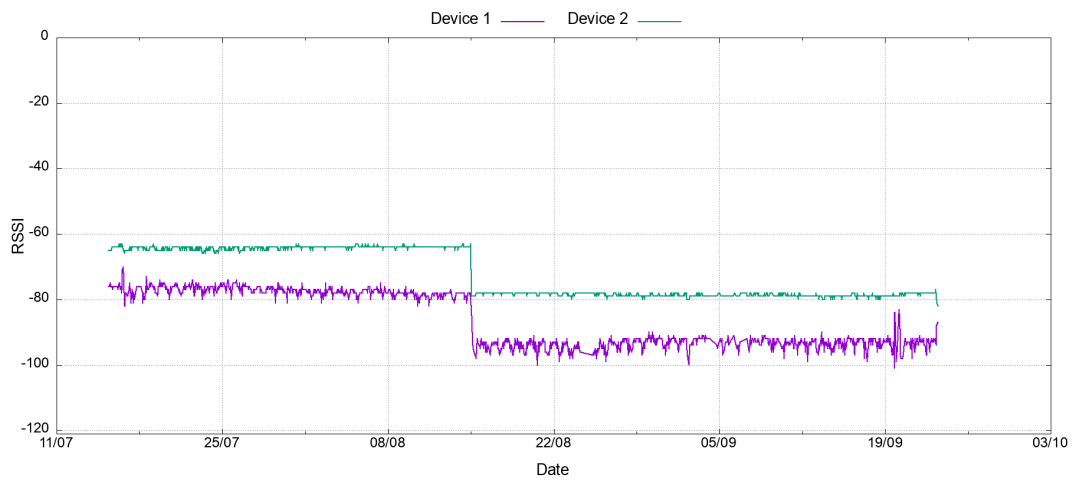


Figure 6. wM-Bus smart meter devices with sudden RSSI drop (Location 2).

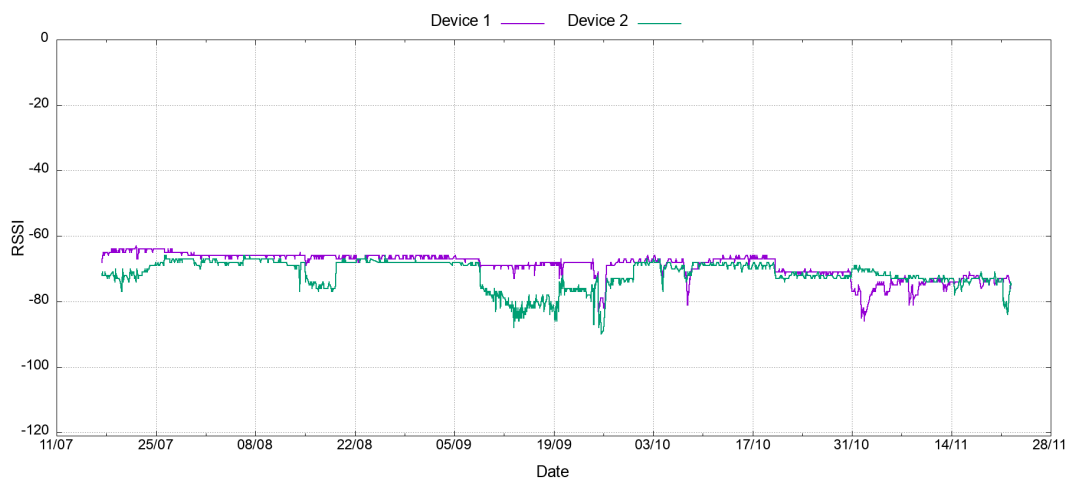
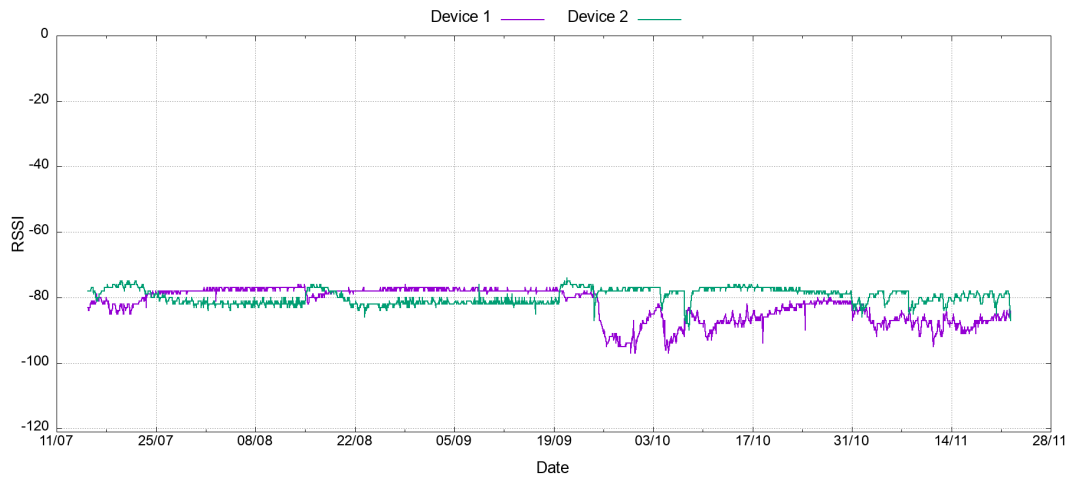


Figure 7. wM-Bus smart meter devices with unstable RSSI 1 (Location 3).





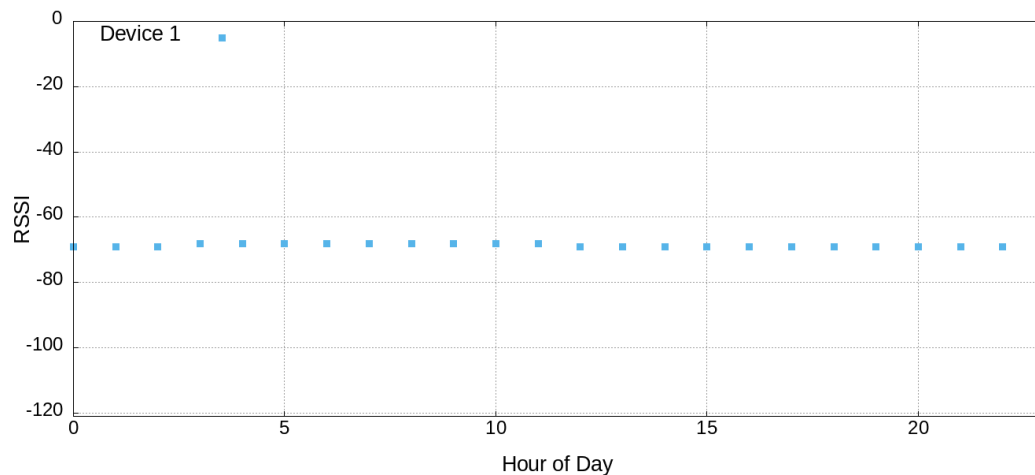
**Figure 8.** wM-Bus smart meter devices with unstable RSSI 2 (Location 4).

Table 1 presents the absolute numbers for the received and expected packets from all the smart meters in the four locations presented above. We can see that the received packet rate was quite low in some cases, especially when the RSSI dropped closer to  $-100$  db. We can also see that the numbers we recorded seemed quite similar to the ones presented in [23], depending on the exact installation characteristics.

**Table 1.** Percentage of received packets based on the manufacturer’s setting for hourly broadcasts.

Location	Device	Expected Packets	Received Packets	(%)
1	1	40,701	3059	65
1	2	40,701	3321	70
1	3	40,701	3021	64
2	1	40,701	1997	42
2	2	40,701	2017	43
3	1	40,701	2632	56
3	2	40,701	2628	56
4	1	40,701	3092	66
4	2	40,701	3140	67

Another important metric we wanted to evaluate was the effects of human presence in the vicinity of the installation. Figure 9 shows the average RSSI measurement for the packets received from a single smart meter and by its assigned Mox node. As we can seem, these values did not present any huge deviation throughout the day, allowing us to be sure that there were no daily events that may periodically affect our communication. The same behaviour was presented in all devices of our installation.



**Figure 9.** wM-Bus RSSI variability during a day.

#### 4.2. LoRaWAN Connectivity Evaluation

The data packets collected from the first sub-layer of the edge-based processing layer (EPL-A) were pre-processed and forwarded to the Tergo nodes (EPL-B) for detailed stream-based analysis. Recall from Section 3.2 that the Mox nodes support basic data processing functionality and can only be used for elementary tasks. The pre-processed data were forwarded to the nearest Tergo nodes using the LoRaWAN. It is now important to assess the connectivity of networking technology. In this second experiment, we wished to measure signal quality and the resulting number of re-transmissions over a given period of time, as well as the achieved packet delivery rate. Communication was carried out over a private LoRaWAN network that operated on the publicly accessible, license-free 868 MHz radio frequency band and was easily affected by nearby transmission and increased background noise [101,102]. This noise originated from any electronic equipment with wireless connectivity or RF transmitters (e.g., garage door openers) and could be especially harmful to the Internet of Things devices. Figures 10 and 11 showcase the effects of this background noise on our LoRaWAN installation. The first figure shows the re-transmissions required to upload the collected data over a weekend from the Mox node to the LoRa server. When we compare these data to the ones from Figure 11, we see how the transmissions became more difficult, resulting in many more re-transmissions. We also see how on a bank holiday, which was on Thursday, the number of re-transmissions was again low, similar to the conditions of the weekend.

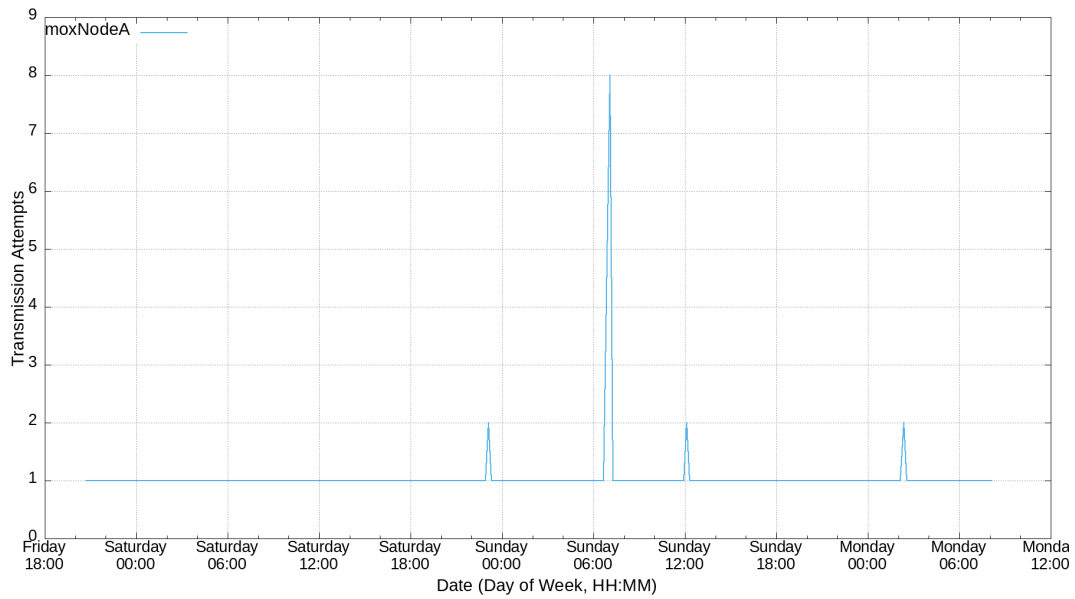


Figure 10. LoRaWAN re-transmissions during the weekend.

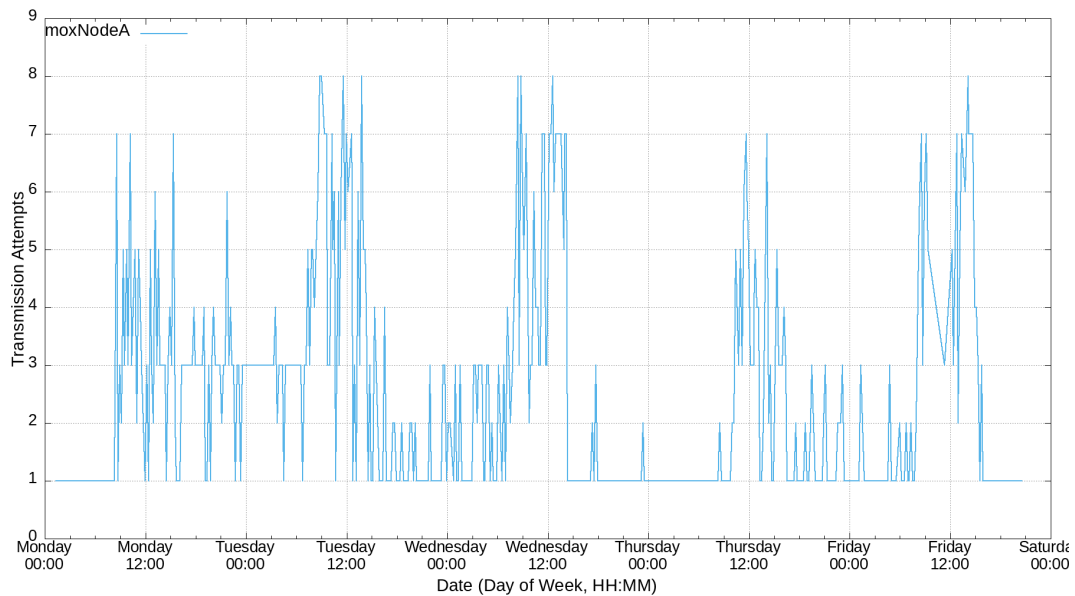


Figure 11. LoRaWAN re-transmissions during weekdays and a bank holiday (Thursday).

Table 2 presents the absolute numbers for the received and expected LoRaWAN packets from Mox nodes in the four locations presented above. We can see that the number of received packets was much higher than the ones observed for the wM-Bus network. These results also agreed with the ones presented in [23], with LoRaWAN appearing to be a much more reliable communication protocol than wM-Bus, even at larger distances. Unfortunately, the power requirements of LoRaWAN were much higher than those of wM-Bus and could limit the overall lifetime of a LoRaWAN-enabled smart water meter. Furthermore, we need to note that during our evaluation, we had no LoRaWAN-enabled smart meter that we could place inside the cast iron-covered inspection pits to evaluate its behaviour in a straight comparison.

**Table 2.** Percentage of received packets based on the manufacturer’s setting for hourly broadcasts.

Location	Expected Packets	Received Packets	(%)
1	69,120	67,456	98
2	69,120	61,367	89
3	69,120	67,162	97
4	69,120	65,954	95

#### 4.3. Edge Data Processing Evaluation

The data packets were now delivered to the Tergo nodes, the second part of the edge processing layer (EPL-B). First, we were interested in evaluating the overall performance of the layer to reduce effectively the volumes of data collected from the sensors and thus reduce the overall dependence on the cloud back-end services. We measured the volume of data in terms of Input/Output (I/O) operations induced for each part of the processing, as described in Section 3.3. We measured the latency of the processing pipeline and the resulting processing rate in terms of time required to process each individual data packet. We used these measurements to identify the minimum processing capabilities required in the real-world deployment. Three different hardware platforms were used to evaluate the performance of the Tethys fog computing hierarchy. For the edge-based processing Sub-layer B, we used two different off-the-shelf single-board devices. We also included a standard Intel-based processor that is commonly available in cloud provider infrastructures in order to compare the performance of the edge-based processing layer, if it was reallocated to the cloud processing layer. The available computational, memory and storage capabilities of the three different units are described in detail in Table 3. We bundled the software components using different Docker containers so that the same configuration parameters and software versions were deployed on each device. As indicated in Table 3, the Raspberry PI Tergo node provided a single core for execution. For this reason, we configured the Docker containers to use only one core throughout all three platforms.

**Table 3.** Technical specifications of edge devices.

	Tergo Raspberry	Tergo Zotac	Tethys
Processor	BCM2836 Arm7	Intel i3-3120M	Intel E5-2630V4
Frequency	900 MHz	2.2 GHz	2.2 GHz
Cores	1	2	6
Memory	736 MB	8 GB	24 GB
Disk	64 GB	120 GB	600 GB
Type	SD Card class10	SSD	SSD

In the pilot deployment, a total of 21 smart metering devices was installed with an average of two devices per building [40]. For each building, we deployed one Mox node controlling the devices of the corresponding building. Each smart metering device transmitted every 60 min several sensor readings. Throughout the duration of the pilot operation, a total of 164,343 messages containing sensor readings was generated.

All the messages containing sensor readings that were collected during the pilot operation were stored in a way that allowed us to replay the network traffic. Such network traffic replay allowed us to evaluate the performance of the system under different operating parameters. In particular, we could inject into each Mox node those messages that corresponded to the smart metering devices located on the same building. We could, therefore, evaluate the overall performance of the system by examining different message generation rates. For example, we increased the message generation rate of each device to 12 messages per second, about 20 times higher than the actual requirements of the real-world

deployment. In this way, we could evaluate the performance of each different type of edge-based processing device under increased load.

We started the experimentation by measuring the network performance of the fog computing-based deployment. One of the main goals of the fog computing paradigm is that sensor values are processed using the computational resources available at the edges of the network, thus reducing the need to transfer the data to the cloud servers. We were therefore interested in examining the average volume of data exchanged by the different services executed on the edge-based processing layer. This was measured in terms of Input/Output (I/O) network operations. The results are depicted in Table 4. Recall from Section 3.2 that each sensor received at the edge-based processing layer was organized by the stream processing pipeline. The actual processing of each sensor reading took place within the generic data processing module. The processing of the message concluded within the continuous data analysis module that guaranteed that data were stored uniformly both at the cloud and the edge layers. For example, the network traffic arriving at the Tergo Raspberry Pi was at the level of 101 MBs for a total of 164,343 messages, while only one-fourth exited the service. By combining all three services, only about 5% of the network traffic was transmitted to the cloud. Clearly, the results indicated that the ability to process each sensor value by taking advantage of the processing power available significantly reduced the overall network traffic.

**Table 4.** Network I/O operations in MB.

Service	Tergo Raspberry I/O	Tergo Zotac I/O	Tethys I/O
Stream Processing Pipeline	101/26.9	88.6/17.3	89.1/15.9
Generic Data Processing	79.8/82	78.2/79.9	79.4/80.4
Continuous Data Analysis	6.62/5.37	2.03/1.97	0.45/0.46

A critical question, however, is to evaluate the resulting response time. That is, we were interested in evaluating the average time required by each different service to process a single message. In Table 5, the processing latency is measured (in ms) for each different service and for each edge device. It was evident that the more powerful Tergo Zotac and Tethys nodes achieved very low latency, which in fact could even accommodate applications that require real-time response times. However, even the low-power Tergo Raspberry Pi node, which did not support fast integer and floating-point calculations, managed to complete the processing pipeline with a response time of as low as one-tenth of a second. At this point, we were also interested in evaluating the operations required for the encryption/decryption of the messages at each different layer. Once again, it was evident that all platforms considered managed to carry out the operations without any significant delays. In this experiment, we also decided to include in the comparison the Mox nodes as well. Recall that these were the nodes that bridge the wM-Bus with the LPWAN and utilize an Arm Cortex M processor. We note that the ultra-low-power processor was capable of carrying out the cryptographic operations without imposing any significant delays.

**Table 5.** Services' processing latency (ms).

Service	Mox Node	Tergo Raspberry	Tergo Zotac	Tethys
Stream Processing Pipeline	-	1.620	0.016	0.004
Generic Data Processing	-	52.183	0.400	0.244
Continuous Data Analysis	-	63.817	0.186	0.145
CMAC calculation	1.199	0.036	0.0017	0.001

The last aspect of the evaluation related to the pilot deployment in a real-world environment was the utilization of the computational resources available at the edges of the network. Now, we were interested in understanding whether the processing power provided by each platform considered

sufficed to handle the load created by the sensors. Therefore, in this experiment, we examined the usage of the processors of each platform considered. Table 6 depicts the processing rates achieved by each platform in a number of messages processed per second. Evidently, the Tergo nodes, those based on the Raspberry Pi single-board computer, achieved the lowest processing rate among the three platforms considered. Interestingly, the processing rate on average reached 15.36 messages per second, which although low, was still higher than the message generation rate of the experiment. In this sense, it was safe to conclude that the processing power of the Raspberry Pi was enough to handle the load. However, given that the message generation rate used in the real-world deployment was about 20 times lower than the one used in the experiments, we concluded that the processing power provided by this low-power single-board computer was adequate. We were therefore confident that a Raspberry Pi could be used to handle more than 20–30 smart meters. On the other hand, the other two platforms considered, that is the Tergo Zotac and Tethys nodes, had a much larger processing rate and could, therefore, accommodate a much higher number of sensor metering devices.

**Table 6.** Processing rate in messages per second.

Device	Processing Rate
Tergo Raspberry	15.36
Tergo Zotac	2692.31
Tethys	5833.33

## 5. Conclusions

This paper presented an IoT-enabled platform, designed based on the fog computing paradigm, for monitoring and controlling water distribution grids. The key element of the design approach was the introduction of an intermediate layer, between remote cloud data centres and the smart metering devices, capable of executing computational tasks. This intermediate layer was connected to the network backbone based on long-range low-power networking technologies. The platform implemented a generic execution environment that was flexible enough to accommodate a broad range of data analytic tasks that could be executed on this intermediate layer. These tasks were deployed at the remote cloud data centres and were delivered to the intermediate layer of execution.

The platform was deployed in a real-world environment comprised of several smart metering devices and smart valves. The performance of the deployment was evaluated both in terms of network connectivity and computational performance. The experimental evaluation indicated that with proper fine-tuning of the various parameters, the intermediate layer could operate over narrow-band connectivity, enhance the reliability of the deployment and reinforce the platform security. In particular, the results indicated that stable connectivity was maintained for the majority of the period. At the same time, the intermediate layer was used to analyse the data collected from the smart metering devices, thus significantly reducing the amount of data that needed to be delivered to the cloud data centres.

As future work, we would like to extend the deployment by introducing additional devices comprised of heterogeneous hardware capabilities, possibly combining also a GPU. In such a heterogeneous intermediate layer, it is important that the execution environment dynamically identifies the most suitable location where the computational tasks can be executed given the network and computational load of the platform. Given such an intermediate processing layer, it is important to implement different data analysis tools to enrich water consumption information and assist water utilities and their clients in obtaining benefits from data sampling and the cost of high-resolution metering and real-time data-model coupling. The availability of such data analysis tools at the edges of the network will also enable the development of richer, direct feedback mechanisms to help reduce water usage and thus benefit the customers; billing and costs.

The work presented in this paper focused on the benefits of the fog computing paradigm in terms of processing the data collected from the smart water metering devices. Important future work is to examine the levels of cyber-security and cyber-protection achieved. As a system composed of different smart devices, with data processing carried out at remotely located micro-data centres, guaranteeing the overall security of the system is a challenge [62,103,104]. Towards this direction, we need to develop efficient methods and technologies to benchmark the smart water micro-components' protection efficiency.

Finally, an important future research direction is to continue the operation of the pilot deployment in order to collect long-term data related to the actual data usage and the network performance. Smart water grids have an increased complexity compared to traditional water infrastructures. Therefore, more real-world tests will help better realize the resilience of the smart water system related to automatic and online operations.

**Author Contributions:** Conceptualization and methodology, D.A., I.C., and S.P.; software, D.A., N.T., and N.N.; writing, original draft preparation, D.A., I.C., and C.T.; writing, reviewing and editing, I.C. and C.T.; data curation and visualization, D.A., N.T., and N.N.; supervision, project administration and funding acquisition, I.C. and S.P. All authors read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by the EU Horizon 2020 E2DATA Grant Number 780245.

**Conflicts of Interest:** No conflict of interest is reported by the authors.

## References

1. Cavoukian, A.; Polonetsky, J.; Wolf, C. Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation. *Identity Inf. Soc.* **2010**, *3*, 275–294. [[CrossRef](#)]
2. Lee, S.W.; Sarp, S.; Jeon, D.J.; Kim, J.H. Smart water grid: The future water management platform. *Desalin. Water Treat.* **2015**, *55*, 339–346. [[CrossRef](#)]
3. Bracciali, A.; Chatzigiannakis, I.; Vitaletti, A.; Zecchini, M. Citizens Vote to Act: Smart contracts for the management of water resources in smart cities. In Proceedings of the 2019 First International Conference on Societal Automation (SA), Krakow, Poland, 4–6 September 2019.
4. Engle, D. Leak busters: Ultra-accurate new residential meters and methods can zap-ghost water accounts. *Water Effic.* **2010**, *5*, 22–29.
5. Gurung, T.; Stewart, R.; Beal, C.; Sharma, A. Smart meter enabled informatics for economically efficient diversified water supply infrastructure planning. *J. Clean. Prod.* **2016**, *135*, 1023–1033. [[CrossRef](#)]
6. Horsburgh, J.S.; Leonardo, M.E.; Abdallah, A.M.; Rosenberg, D.E. Measuring water use, conservation, and differences by gender using an inexpensive, high frequency metering system. *Environ. Model. Softw.* **2017**, *96*, 83–94. [[CrossRef](#)]
7. Beal, C.; Stewart, R.; Huang, T.; Rey, E. *South East Queensland Residential End Use Study*; Urban Water Security Research Alliance: Brisbane, Australia, 2011.
8. Nguyen, K.A.; Stewart, R.A.; Zhang, H.; Sahin, O.; Siriwardene, N. Re-engineering traditional urban water management practices with smart metering and informatics. *Environ. Model. Softw.* **2018**, *101*, 256–267. [[CrossRef](#)]
9. Pau, M.; Patti, E.; Barbierato, L.; Estebarsari, A.; Pons, E.; Ponci, F.; Monti, A. A cloud-based smart metering infrastructure for distribution grid services and automation. *Sustain. Energy Grids Networks* **2018**, *15*, 14–25. [[CrossRef](#)]
10. Li, J.; Yang, X.; Sitzenfren, R. Rethinking the Framework of Smart Water System: A Review. *Water* **2020**, *12*. [[CrossRef](#)]
11. Tziortzioti, C.; Mavrommati, I.; Mylonas, G.; Vitaletti, A.; Chatzigiannakis, I. Scenarios for educational and game activities using internet of things data. In Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games (CIG), Maastricht, The Netherlands, 14–17 August 2018.
12. Mylonas, G.; Amaxilatis, D.; Chatzigiannakis, I.; Anagnostopoulos, A.; Paganelli, F. Enabling Sustainability and Energy Awareness in Schools Based on IoT and Real-World Data. *IEEE Pervasive Comput.* **2018**, *17*, 53–63. [[CrossRef](#)]

13. Tziortzioti, C.; Amaxilatis, D.; Mavrommati, I.; Chatzigiannakis, I. IoT sensors in sea water environment: Ahoy! Experiences from a short summer trial. *Electron. Notes Theor. Comput. Sci.* **2019**, *343*, 117–130. [[CrossRef](#)]
14. Tziortzioti, C.; Andreetti, G.; Rodinò, L.; Mavrommati, I.; Vitaletti, A.; Chatzigiannakis, I. Raising Awareness for Water Pollution Based on Game Activities Using Internet of Things. In Proceedings of the European Conference on Ambient Intelligence, Larnaca, Cyprus, 12–14 November 2018; pp. 171–187.
15. Mutchek, M.; Williams, E. Moving towards sustainable and resilient smart water grids. *Challenges* **2014**, *5*, 123–137. [[CrossRef](#)]
16. Amaxilatis, D.; Akrivopoulos, O.; Chatzigiannakis, I.; Tselios, C. Enabling stream processing for people-centric IoT based on the fog computing paradigm. In Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2017, Limassol, Cyprus, 12–15 September 2017.
17. Theodoridis, E.; Mylonas, G.; Chatzigiannakis, I. Developing an IoT smart city framework. In Proceedings of the IISA 2013, Piraeus, Greece, 10–12 July 2013.
18. Chatzigiannakis, I.; Mylonas, G.; Vitaletti, A. Urban pervasive applications: Challenges, scenarios and case studies. *Comput. Sci. Rev.* **2011**, *5*, 103–118. [[CrossRef](#)]
19. Allen, M.; Preis, A.; Iqbal, M.; Whittle, A.J. Case study: A smart water grid in Singapore. *Water Pract. Technol.* **2012**, *7*. [[CrossRef](#)]
20. Saravanan, M.; Das, A.; Iyer, V. Smart water grid management using LPWAN IoT technology. In Proceedings of the 2017 Global Internet of Things Summit (GIoTS), Geneva, Switzerland, 6–9 June 2017.
21. Dong, J.; Wang, G.; Yan, H.; Xu, J.; Zhang, X. A survey of smart water quality monitoring system. *Environ. Sci. Pollut. Res.* **2015**, *22*, 4893–4906. [[CrossRef](#)] [[PubMed](#)]
22. Okay, F.Y.; Ozdemir, S. A fog computing based smart grid model. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, 11–13 May 2016.
23. Alvisi, S.; Casellato, F.; Franchini, M.; Govoni, M.; Luciani, C.; Poltronieri, F.; Riberto, G.; Stefanelli, C.; Tortonesi, M. Wireless Middleware Solutions for Smart Water Metering. *Sensors* **2019**, *19*, 1853. [[CrossRef](#)] [[PubMed](#)]
24. Kumari, A.; Tanwar, S.; Tyagi, S.; Kumar, N.; Obaidat, M.S.; Rodrigues, J.J.P.C. Fog Computing for Smart Grid Systems in the 5G Environment: Challenges and Solutions. *IEEE Wirel. Commun.* **2019**, *26*, 47–53. [[CrossRef](#)]
25. Zahoor, S.; Javaid, N.; Khan, A.; Ruqia, B.; Muhammad, F.J.; Zahid, M. A Cloud-Fog-Based Smart Grid Model for Efficient Resource Utilization. In Proceedings of the 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 1154–1160.
26. Amaxilatis, D.; Chatzigiannakis, I. Design and Analysis of Adaptive Hierarchical Low-Power Long-Range Networks. *J. Sens. Actuator Netw.* **2018**, *7*, 51. [[CrossRef](#)]
27. Yan, Y.; Su, W. A fog computing solution for advanced metering infrastructure. In Proceedings of the 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T D), Dallas, TX, USA, 3–5 May 2016.
28. Fang, X.; Yang, D.; Xue, G. Evolving Smart Grid Information Management Cloudward: A Cloud Optimization Perspective. *IEEE Trans. Smart Grid* **2013**, *4*, 111–119. [[CrossRef](#)]
29. Amaxilatis, D.; Akrivopoulos, O.; Mylonas, G.; Chatzigiannakis, I. An IoT-based solution for monitoring a fleet of educational buildings focusing on energy efficiency. *Sensors* **2017**, *17*, 2296. [[CrossRef](#)]
30. Akrivopoulos, O.; Chatzigiannakis, I.; Koninis, C.; Theodoridis, E. A web services-oriented architecture for integrating small programmable objects in the web of things. In Proceedings of the 2010 Developments in E-systems Engineering, London, UK, 6–8 September 2010; pp. 70–75.
31. Chen, Y.Y.; Lin, Y.H.; Kung, C.C.; Chung, M.H.; Yen, I.H. Design and Implementation of Cloud Analytics-Assisted Smart Power Meters Considering Advanced Artificial Intelligence as Edge Analytics in Demand-Side Management for Smart Homes. *Sensors* **2019**, *19*, 2047. [[CrossRef](#)]
32. Anderson, R.J.; Fuloria, S. On the Security Economics of Electricity Metering. In *WEIS*; Citeseer: Princeton, NJ, USA, 2010.
33. Giurco, D.P.; White, S.B.; Stewart, R.A. Smart metering and water end-use data: Conservation benefits and privacy risks. *Water* **2010**, *2*, 461–467. [[CrossRef](#)]
34. Laughman, C.; Lee, K.; Cox, R.; Shaw, S.; Leeb, S.; Norford, L.; Armstrong, P. Power signature analysis. *IEEE Power Energy Mag.* **2003**, *1*, 56–63. [[CrossRef](#)]



35. Rial, A.; Danezis, G. Privacy-preserving Smart Metering. In Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, Chicago, IL, USA, 17 October 2011; ACM: New York, NY, USA, 2011; pp. 49–60.
36. Nguyen, K.A.; Stewart, R.A.; Zhang, H. An autonomous and intelligent expert system for residential water end-use classification. *Expert Syst. Appl.* **2014**, *41*, 342–356. [[CrossRef](#)]
37. Okay, F.Y.; Ozdemir, S. A secure data aggregation protocol for fog computing based smart grids. In Proceedings of the 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), Doha, Qatar, 10–12 April 2018.
38. Chatzigiannakis, I.; Pyrgelis, A.; Spirakis, P.G.; Stamatou, Y.C. Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. In Proceedings of the 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, Valencia, Spain, 17–22 October 2011; pp. 715–720.
39. Cao, H.; Liu, S.; Wu, L.; Guan, Z.; Du, X. Achieving differential privacy against non-intrusive load monitoring in smart grid: A fog computing approach. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4528. [[CrossRef](#)]
40. Amaxilatis, D.; Chatzigiannakis, I.; Tselios, C.; Tsironis, N. Sparks-Edge: Analytics for Intelligent City Water Metering. In Proceedings of the Joint Proceeding of the Poster and Workshop Sessions of Aml-2019, the 2019 European Conference on Ambient Intelligence, Rome, Italy, 13–15 November 2019; pp. 19–26.
41. Yang, W.; Wang, M.; Zhang, J.; Zou, J.; Hua, M.; Xia, T.; You, X. Narrowband Wireless Access for Low-Power Massive Internet of Things: A Bandwidth Perspective. *IEEE Wirel. Commun.* **2017**, *24*, 138–145. [[CrossRef](#)]
42. Darby, S. The effectiveness of feedback on energy consumption. *Rev. DEFRA Lit. Metering Billing Direct Displays* **2006**, *486*, 26.
43. Ehrhardt-Martinez, K.; Donnelly, K.A.; Laitner, S. *Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities*; American Council for an Energy-Efficient Economy: Washington, DC, USA, 2010.
44. Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S. Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors* **2012**, *12*, 16838–16866. [[CrossRef](#)]
45. Lisovich, M.A.; Mulligan, D.K.; Wicker, S.B. Inferring personal information from demand-response systems. *IEEE Secur. Priv.* **2010**, *8*, 11–20. [[CrossRef](#)]
46. Garcia Lopez, P.; Montresor, A.; Epema, D.; Datta, A.; Higashino, T.; Iamnitchi, A.; Barcellos, M.; Felber, P.; Riviere, E. Edge-centric Computing: Vision and Challenges. *SIGCOMM Comput. Commun. Rev.* **2015**, *45*, 37–42. [[CrossRef](#)]
47. Varghese, B.; Wang, N.; Barbhuiya, S.; Kilpatrick, P.; Nikolopoulos, D.S. Challenges and Opportunities in Edge Computing. In Proceedings of the 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 18–20 November 2016; pp. 20–26.
48. Tselios, C.; Tsolis, G. On QoE-awareness through virtualized probes in 5G networks. In Proceedings of the 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), Toronto, ON, Canada, 23–25 October 2016; pp. 159–164.
49. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76. [[CrossRef](#)]
50. Tselios, C.; Politis, I.; Kotsopoulos, S. Enhancing SDN security for IoT-related deployments through blockchain. In Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, Germany, 6–8 November 2017; pp. 303–308.
51. Boukerche, A.; Chatzigiannakis, I.; Nikolettas, S. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. *Comput. Commun.* **2006**, *29*, 477–489. [[CrossRef](#)]
52. Akrivopoulos, O.; Chatzigiannakis, I.; Tselios, C.; Antoniou, A. On the Deployment of Healthcare Applications over Fog Computing Infrastructure. In Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 4–8 July 2017; Volume 2, pp. 288–293.
53. Papageorgiou, A.; Cheng, B.; Kovacs, E. Real-time data reduction at the network edge of Internet-of-Things systems. In Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain, 9–13 November 2015; pp. 284–291.
54. Zhu, N.; Anagnostopoulos, A.; Chatzigiannakis, I. On Mining IoT Data for Evaluating the Operation of Public Educational Buildings. In Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018, Athens, Greece, 19–23 March 2018; pp. 278–283.

55. Akrivopoulos, O.; Zhu, N.; Amaxilatis, D.; Tselios, C.; Anagnostopoulos, A.; Chatzigiannakis, I. A Fog Computing-Oriented, Highly Scalable IoT Framework for Monitoring Public Educational Buildings. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018.
56. Bianchi, G.; Biton, E.; Blefari-Melazzi, N.; Borges, I.; Chiaraviglio, L.; Cruz Ramos, P.; Eardley, P.; Fontes, F.; McGrath, M.J.; Natarianni, L.; et al. Superfluidity: A flexible functional architecture for 5G networks. *Trans. Emerg. Telecommun. Technol.* **2016**, *27*, 1178–1186. [[CrossRef](#)]
57. Bolivar, L.T.; Tselios, C.; Mellado Area, D.; Tsolis, G. On the Deployment of an Open-Source, 5G-Aware Evaluation Testbed. In Proceedings of the 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Bamberg, Germany, 26–29 March 2018; pp. 51–58.
58. Nousias, S.; Tselios, C.; Bitzas, D.; Orfila, O.; Jamson, S.; Mejuto, P.; Amaxilatis, D.; Akrivopoulos, O.; Chatzigiannakis, I.; Lalos, A.S.; et al. Managing nonuniformities and uncertainties in vehicle-oriented sensor data over next generation networks. In Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, Greece, 19–23 March 2018; pp. 272–277.
59. Nousias, S.; Tselios, C.; Bitzas, D.; Lalos, A.S.; Moustakas, K.; Chatzigiannakis, I. Uncertainty Management for Wearable IoT Wristband Sensors Using Laplacian-Based Matrix Completion. In Proceedings of the 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Barcelona, Spain, 17–19 September 2018.
60. Akrivopoulos, O.; Amaxilatis, D.; Mavrommati, I.; Chatzigiannakis, I. Utilising fog computing for developing a person-centric heart monitoring system. *JAISE* **2019**, *11*, 237–259. [[CrossRef](#)]
61. Tselios, C.; Nousias, S.; Bitzas, D.; Amaxilatis, D.; Akrivopoulos, O.; Lalos, A.S.; Moustakas, K.; Chatzigiannakis, I. Enhancing an Eco-Driving Gamification Platform Through Wearable and Vehicle Sensor Data Integration. In *Ambient Intelligence*; Chatzigiannakis, I., De Ruyter, B., Mavrommati, I., Eds.; Springer: Cham, Switzerland, 2019; pp. 344–349.
62. Chatzigiannakis, I.; Maiano, L.; Trakadas, P.; Anagnostopoulos, A.; Bacci, F.; Karkazis, P.; Spirakis, P.G.; Zahariadis, T. Data-Driven Intrusion Detection for Ambient Intelligence. In Proceedings of the European Conference on Ambient Intelligence, Rome, Italy, 12–15 November 2019; pp. 235–251.
63. Amaxilatis, D.; Tselios, C.; Akrivopoulos, O.; Chatzigiannakis, I. On the Design of a Fog Computing-Based, Driving Behaviour Monitoring Framework. In Proceedings of the 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 11–13 September 2019.
64. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet Things J.* **2016**, *3*, 854–864. [[CrossRef](#)]
65. Kumar, A.; Sengupta, J.; Liu, Y.f. 3GPP LTE: The future of mobile broadband. *Wirel. Pers. Commun.* **2012**, *62*, 671–686. [[CrossRef](#)]
66. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [[CrossRef](#)]
67. Communication Systems for Meters. Available online: [https://ec.europa.eu/eip/ageing/standards/ict-and-communication/data/en-13757\\_en](https://ec.europa.eu/eip/ageing/standards/ict-and-communication/data/en-13757_en) (accessed on 6 September 2019).
68. Chatzigiannakis, I.; Dimitriou, T.; Nikolettseas, S.E.; Spirakis, P.G. A probabilistic algorithm for efficient and robust data propagation in wireless sensor networks. *Ad Hoc Netw.* **2006**, *4*, 621–635. [[CrossRef](#)]
69. Chatzigiannakis, I.; Mylonas, G.; Nikolettseas, S.E. Modeling and Evaluation of the Effect of Obstacles on the Performance of Wireless Sensor Networks. In Proceedings of the 39th Annual Simulation Symposium, Huntsville, Alabama, 2–6 April 2006; pp. 50–60.
70. Chatzigiannakis, I.; Kinalis, A.; Nikolettseas, S. An adaptive power conservation scheme for heterogeneous wireless sensor networks with node redeployment. In Proceedings of the Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, Las Vegas, NV, USA, 18–20 July 2005; pp. 96–105.
71. Chatzigiannakis, I.; Konstantinou, E.; Liagkou, V.; Spirakis, P.G. Design, Analysis and Performance Evaluation of Group Key Establishment in Wireless Sensor Networks. *Electron. Notes Theor. Comput. Sci.* **2007**, *171*, 17–31. [[CrossRef](#)]
72. Baumgartner, T.; Chatzigiannakis, I.; Fekete, S.P.; Fischer, S.; Koninis, C.; Krölller, A.; Krüger, D.; Mylonas, G.; Pfisterer, D. Distributed algorithm engineering for networks of tiny artifacts. *Comput. Sci. Rev.* **2011**, *5*, 85–102. [[CrossRef](#)]

73. Amaxilatis, D.; Chatzigiannakis, I.; Dolev, S.; Koninis, C.; Pyrgelis, A.; Spirakis, P.G. Adaptive Hierarchical Network Structures for Wireless Sensor Networks. In Proceedings of the 3rd International Conference on Ad Hoc Networks, Paris, France, 21–23 September 2011; Volume 89, pp. 65–80.
74. Chatzigiannakis, I.; Kinalis, A.; Nikolettseas, S.E. Efficient data propagation strategies in wireless sensor networks using a single mobile sink. *Comput. Commun.* **2008**, *31*, 896–914. [[CrossRef](#)]
75. Sanchez, L.; Muñoz, L.; Galache, J.A.; Sotres, P.; Santana, J.R.; Gutierrez, V.; Ramdhany, R.; Gluhak, A.; Krco, S.; Theodoridis, E.; et al. SmartSantander: IoT experimentation over a smart city testbed. *Comput. Netw.* **2014**, *61*, 217–238. [[CrossRef](#)]
76. Chatzigiannakis, I.; Vitaletti, A.; Pyrgelis, A. A privacy-preserving smart parking system using an IoT elliptic curve based security platform. *Comput. Commun.* **2016**, *89–90*, 165–177. [[CrossRef](#)]
77. Centenaro, M.; Vangelista, L.; Zanella, A.; Zorzi, M. Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios. *IEEE Wirel. Commun.* **2015**, *23*. [[CrossRef](#)]
78. Raza, U.; Kulkarni, P.; Sooriyabandara, M. Low power wide area networks: An overview. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 855–873. [[CrossRef](#)]
79. Chatzigiannakis, I.; Kinalis, A.; Mylonas, G.; Nikolettseas, S.E.; Prasinos, G.; Zaroliagis, C.D. TRAILS, a Toolkit for Efficient, Realistic and Evolving Models of Mobility, Faults and Obstacles in Wireless Networks. In Proceedings of the 41st Annual Simulation Symposium (ANSS-41 2008), Ottawa, ON, Canada, 14–16 April 2008; pp. 23–32.
80. Chatzigiannakis, I.; Mylonas, G.; Nikolettseas, S.E. A Model for Obstacles to be used in Simulations of Wireless Sensor Networks and its Application in studying Routing Protocol Performance. *Simulation* **2007**, *83*, 587–608. [[CrossRef](#)]
81. Mikhaylov, K.; Petaejaervi, J.; Haenninen, T. Analysis of capacity and scalability of the LoRa low power wide area network technology. In Proceedings of the 22th European Wireless Conference, Oulu, Finland, 18–20 May 2016.
82. Varsier, N.; Schwoerer, J. Capacity limits of LoRaWAN technology for smart metering applications. In Proceedings of the IEEE International Conference on Communications, Paris, France, 21–25 May 2017.
83. Gaddam, S.C.; Rai, M.K. A comparative study on various LPWAN and cellular communication technologies for IoT based smart applications. In Proceedings of the 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), Arakkunnam, Kerala, 11–13 July 2018.
84. Fumero, J. Accelerating Interpreted Programming Languages on GPUs with Just-In-Time and Runtime Optimisations. Ph.D. Thesis, The University of Edinburgh, Edinburgh, UK, 2017.
85. Chatzigiannakis, I.; Giannoulis, G.; Spirakis, P. Scheduling tasks with dependencies on asymmetric multiprocessors. In Proceedings of the Twenty-Seventh ACM Symposium on Principles of Distributed Computing, Toronto, ON, Canada, 18–21 August 2008; p. 454.
86. Kotselidis, C.; Komnios, I.; Akrivopoulos, O.; Bress, S.; Doka, K.; Mohammed, H.; Mylonas, G.; Spitadakis, V.; Luxembourg, N.; Strimpel, D. Efficient Compilation and Execution of JVM-Based Data Processing Frameworks on Heterogeneous Co-Processors. In Proceedings of the Design, Automation and Test in Europe Conference (DATE 2020), Grenoble, France, 9–13 March 2020.
87. Carbone, P.; Katsifodimos, A.; Ewen, S.; Markl, V.; Haridi, S.; Tzoumas, K. Apache Flink<sup>TM</sup>: Stream and Batch Processing in a Single Engine. *IEEE Data Eng. Bull.* **2015**, *38*, 28–38.
88. Zaharia, M.; Xin, R.S.; Wendell, P.; Das, T.; Armbrust, M.; Dave, A.; Meng, X.; Rosen, J.; Venkataraman, S.; Franklin, M.J.; et al. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* **2016**, *59*, 56–65. [[CrossRef](#)]
89. Apache Storm. Available online: <https://storm.apache.org> (accessed on 27 January 2020).
90. Grossman, M.; Breternitz, M.; Sarkar, V. HadoopCL: MapReduce on Distributed Heterogeneous Platforms through Seamless Integration of Hadoop and OpenCL. In Proceedings of the 2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum, Cambridge, MA, USA, 20–24 May 2013; pp. 1918–1927.
91. Li, P.; Luo, Y.; Zhang, N.; Cao, Y. HeteroSpark: A heterogeneous CPU/GPU Spark platform for machine learning algorithms. In Proceedings of the 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), Boston, MA, USA, 6–7 August 2015; pp. 347–348.
92. El-Helw, I.; Hofman, R.F.H.; Bal, H.E. Glasswing: Accelerating mapreduce on multi-core and many-core clusters. In Proceedings of the 23rd International Symposium on High-Performance Parallel and Distributed Computing, HPDC '14, Vancouver, BC, Canada, 23–27 June 2014.

93. Xekalaki, M.; Fumero, J.; Kotselidis, C. Challenges and Proposals for Enabling Dynamic Heterogeneous Execution of Big Data Frameworks. In Proceedings of the 2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Nicosia, Cyprus, 10–13 December 2018; pp. 335–341.
94. AMD Aparapi. Available online: <http://aparapi.github.io> (accessed on 27 January 2020).
95. Sabne, A.; Sakdhnagool, P.; Eigenmann, R. HeteroDooP: A MapReduce Programming System for Accelerator Clusters. In Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing, Portland, OR, USA, 15–19 June 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 235–246.
96. Fumero, J.; Papadimitriou, M.; Zakkak, F.S.; Xekalaki, M.; Clarkson, J.; Kotselidis, C. Dynamic Application Reconfiguration on Heterogeneous Hardware. In Proceedings of the 15th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, Providence, RI, USA, 14 April 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 165–178.
97. Ishizaki, K.; Hayashi, A.; Koblents, G.; Sarkar, V. Compiling and Optimizing Java 8 Programs for GPU Execution. In Proceedings of the 2015 International Conference on Parallel Architecture and Compilation (PACT), San Francisco, CA, USA, 18–21 October 2015; pp. 419–431.
98. Transparent GPU Exploitation on Apache Spark. Available online: <https://tinyurl.com/yxdf4oqn> (accessed on 27 January 2020).
99. Papadimitriou, M.; Fumero, J.; Stratikopoulos, A.; Kotselidis, C. Towards Prototyping and Acceleration of Java Programs onto Intel FPGAs. In Proceedings of the 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), San Diego, CA, USA, 28 April–1 May 2019; p. 310.
100. Light, R. Mosquitto: Server and client implementation of the MQTT protocol. *J. Open Source Softw.* **2017**, *2*, 265. [[CrossRef](#)]
101. Staniec, K.; Kowal, M. LoRa performance under variable interference and heavy-multipath conditions. *Wirel. Commun. Mob. Comput.* **2018**, *2018*. [[CrossRef](#)]
102. Vejlggaard, B.; Lauridsen, M.; Nguyen, H.; Kovács, I.Z.; Mogensen, P.; Sorensen, M. Interference impact on coverage and capacity for low power wide area IoT networks. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017.
103. Chatzigiannakis, I.; Liagkou, V.; Spirakis, P.G. Brief Announcement: Providing End-to-End Secure Communication in Low-Power Wide Area Networks. In *Cyber Security Cryptography and Machine Learning*; Dinur, I., Dolev, S., Lodha, S., Eds.; Springer: Cham, Switzerland, 2018; pp. 101–104.
104. Chatzigiannakis, I.; Strikos, A. A decentralized intrusion detection system for increasing security of wireless sensor networks. In Proceedings of the 2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007), Patras, Greece, 25–28 September 2007; pp. 1408–1411.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).