Smart Data: A New Perspective of Tackling the Big Data Phenomena Leveraging a Fog Computing System

* Farhoud Hosseinpour, Juha Plosila, Hannu Tenhunen
University of Turku, Department of Information Technology,
{farhos;juplos;hatenhu}@utu.fi

Abstract

The management of Big Data is a very important issue in emerging IoT technologies. Conventional methods are not sufficient to deal with the ever-increasing amount of raw data originating from the sensors. In this paper we approach this problem from the data structure perspective. We design and develop a concept that we call "Smart Data". Smart Data is an active and intelligent data structure using a fog computing system that facilitates the management of Big Data in IoT based applications. Such a data cell is initially very simple and lightweight, but it evolves (grows) when traveling through the hierarchical fog computing system towards the cloud, merging with other cells or vice-versa, if the data moves from the cloud towards the actuators. Using Smart Data, we aim to facilitate the pre-processing of data to reduce the load from cloud computing and improve the quality of service and energy efficiency in IoT applications. Our main targets for pre-processing of Big Data using Smart Data and fog computing platform include data filtering, aggregation, compression, and encryption. Moreover, our design goal is to reduce volume, velocity and increase value and veracity of Big Data considering other parameters such as energy efficiency, throughput, scalability and quality of service.

Keywords*: Smart Data, Big Data, Fog Computing, Internet of Things*

1. Introduction

Big Data is currently becoming a critical research focus along with the growth of Internet of Things (IoT) and Internet of Everything (IoE) technologies. The European Commission defines Big Data as "large amounts of data produced very quickly by many different sources such as people, machines or sensors [1]". The Big Data phenomenon was emerged with the rapid growth of social networks, machine to machine (M2M) communication, and Internet of Things paradigm. Continuous advancement in emerging IoT and IoE technologies face challenges regarding to managing, storing and processing Big Data to deliver flawless services especially for latency sensitive applications such as video streaming or health management systems. Relying on current technologies such as cloud computing is not efficient for addressing the requirements of Big Data management [2]. Hence, new technologies are needed to reduce the complexity, ease management and boost the processing of Big Data. In this paradigm, Big Data is complex and multidimensional and involved with multifaceted relations between generators, end users and intermediate intelligent processing units. Thus, it requires appropriate technologies and infrastructures for managing, storing and processing. The pre-processing of raw sensory data is one of the most efficient ways to reduce the load of Big Data in cloud computing. To this end, a fog computing platform at the edge of the network is introduced to reduce the processing load from the cloud by delegating some simple and frequent tasks to the fog (pre-processing) [3]. A virtualized and hierarchical architecture of fog computing provides a distributed computing and storage platform near to the edge sensors for local and latency sensitive applications. In this model, the raw data generated in edge sensors is pre-processed in the local fog, and more meaningful and efficient data with reduced volume and velocity are sent to the cloud for further and global processing and storage. Expected benefits of utilizing the fog computing technology in the IoT architecture include: local and hence faster processing and storage for geo-distributed and latency sensitive applications, reduced communication overhead, and reduced volume and velocity of Big Data before sending it to the cloud.

It is evident that recently the research trend for dealing with Big Data has focused on technological advancement in computing platforms and architectural designs for IoT based systems. Yet, relying

merely on the advancement of computing technologies for Big Data processing and management will not completely address the involved issues. In this paper, approaching the problem from a different perspective, we aim to reshape the existing raw, passive and unstructured form of data in IoT to an intelligent and active form, while preserving and enhancing many other important parameters such as energy efficiency, scalability, throughput, quality of service as well as privacy and security. For this, we will introduce a new efficient, intelligent, self-managed and lightweight data structure that we call Smart Data. We believe that Smart Data will revolutionize the current perspective of data and will open many potential research directions to tackle emerging Big Data issues. The Smart Data is a package of encapsulated structured or semi-structured data generated by IoT sensors, a set of metadata, and a virtual machine. It is controlled and managed through the metadata that accommodates a set of rules that define its behavior and govern its security, privacy and other functionalities. The virtual machine, in turn, executes the rules set in the metadata.

The remainder of the paper is organized as follows. In Section **2** we briefly review the Big Data phenomena. We discuss the fog computing technology and its integration with IoT systems in Section **3**. In Section **4** we introduce our Smart Data model and present its structure and, finally, we discuss and conclude in Section **5**.

2. Big Data

Big Data in general is characterized with five major features, namely *Volume, Velocity, Value, Veracity and Variety* which are known as the *5 Vs* of Big Data [4]. *Volume* in Big Data features the size, scale or amount of the data that is required to be processed. *Velocity* of Big Data represents the speed of data generation. It becomes a problematic issue, if data is generated faster than it can be analyzed and stored. *Variety* of Big Data expresses the complexity of data as structured, semi-structured, unstructured or mixed data. *Value* of the data reflects the added value of the data to the underlying process. In fact, the added value of the data represents the gap between the business demand and technological solutions for managing the Big Data. *Veracity* of Big Data refers to the consistency and trustworthiness of the data being processed. Data veracity ensures the integrity, availability, and accountability of the data. In addition to these general characteristics, a new feature, *geo-distribution* [2], is also introduced by emerging IoT applications. In IoT-based applications, sensors in different geographical locations, which have to be managed as a coherent whole, are generating the data. It is evident that centralized approaches such as cloud computing are not efficient for such a highly distributed infrastructure because of the low latency requirements of these applications. One efficient approach to deal with this issue is devising local computing/storage units for each geographical location to respond to local application requests that require fast interaction times and larger computing/storage units (i.e. cloud) for global processing and storage for processes that effect an IoT-based system as a whole.

3. Fog Computing

In a typical IoT infrastructure, raw data are generated through hundreds of thousands of sensors. Cloud computing has been widely adopted in this paradigm because of the immense processing load caused by the collected data, reducing service delivery cost and achieving better interoperability with other cooperated systems [5], [6]. Cloud computing is being recognized as a success factor for IoT, providing ubiquity, reliability, high-performance and scalability [7]. However, due to its geographically centralized nature as well as communication implications, cloud Computing-based IoT fails in applications that require a very low and predictable latency, are geographically distributed, are fast mobile, or are large-scale distributed control systems [3]. Fog computing is a promising technology proposed and developed by Cisco [8], complementing the cloud computing services by extending the computing paradigm to the edge of the network. Bringing the computational intelligence geographically near to the end users will provide new or better services for latency sensitive, location-aware and geo-distributed applications that due to their characteristics are not feasible merely through cloud computing. In this paradigm, smart devices and communication components with both computation and storage capabilities, i.e., intelligent routers, bridges, gateways, and smart devices such as tablets and mobile phones compose the fog computing platform near to the edge of the network.

Therefore, some simple yet frequent tasks of the cloud could be delegated to fog resulting in better performance for IoT based applications [9].

3.1. IoT Architecture

A typical IoT-based system is composed of a set of sensors and/or actuators with built in communication capabilities, connected to a gateway through wireless technologies such as Wi-Fi, Bluetooth or ZigBee. As we are dealing with resource-constrained devices, lightweight protocols, such as CoAP, 6LoWPAN, and MQTT, are used for communication at this level of the system. The gateways, on the other hand, are usually devices with higher processing power and storage capacity. They are responsible for harvesting data from sensors and redirecting data to processing units (typically cloud), or the other way around, from processing units to actuators. In some IoT systems, the gateways also carry out some basic pre-processing on raw sensory data. Communication between gateways and cloud platforms is accomplished through high performance communication technologies such as 3G and broadband technologies over Internet protocols such as TCP/IP, IPv4, and IPv6. At the other side of the network, end user systems and applications access processed data and visualize results of various analyses (Figure 1).

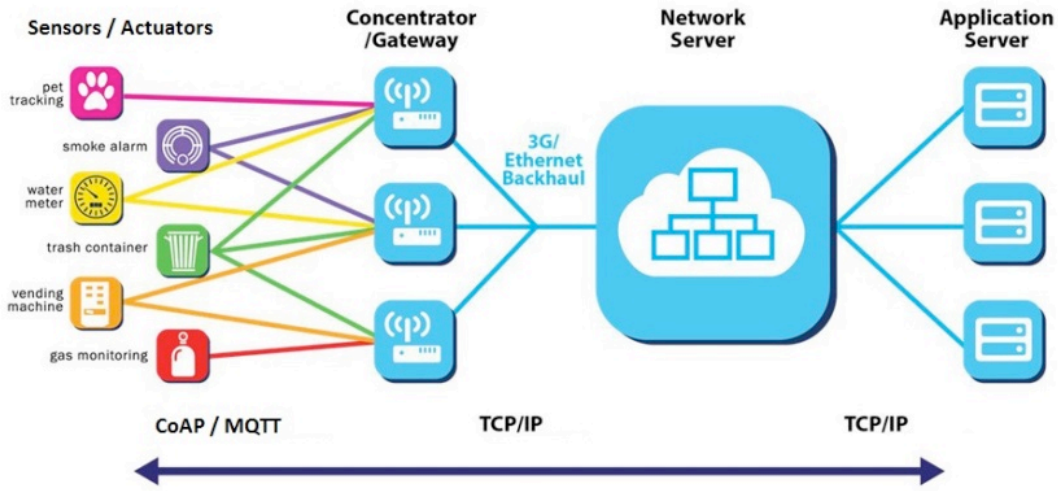

Figure 1: A typical IoT architecture.

Fog computing introduces an intermediate layer between the edge network or the end nodes and the traditional cloud computing layer. The fog layer can be implemented using almost the same components and resources and can utilize many of the same mechanisms and attributes as the cloud layer. Fog computing extends the computing intelligence and storage to the edge of the network. Furthermore, it also extends the computing and storage capability of gateways in conventional IoT systems to a broader and faster distributed, hierarchical computing platform. Fog computing does not outsource the cloud computing's components and services but instead, it aims to provide a computing and storage platform physically closer to the end nodes provisioning new breed of applications and services with an effective interplay with the cloud layer [3]. The expected benefit is faster computation times for requests that require low latency. This plays a crucial role in promotion of the Internet of Things (IoT). Utilizing fog computing reduces the overhead of communication with the cloud through the Internet and provides a faster response for applications that requires lower latency. This is made possible by locally computing IoT data in the fog layer and forwarding only those which does not require real-time computation or require higher processing power to the cloud layer. For example, in the case of a smart community [9], where homes in a neighborhood are connected to provide community services, lower latency can be expected for making urgent decisions, and so, data is sent to a geographically closer computation units instead of a fairly distant cloud node.

In addition to the requirement of low latency, fog computing is a promising technology for dealing with Big Data generated from IoT-based systems with a vast number of nodes spread across a large

area. IoT-based systems introduce a new dimension for characterizing Big Data, namely geo-distribution, along with its generally known characteristics. Cloud based IoT systems dealing with Big Data [3] have to process a large amount of data at any time. Fog computing, as a middleware, can pre-process raw data coming from the edge nodes before sending them to the cloud layer. As a result, the fog layer not only reduces the amount of work needed in the cloud layer by generating meaningful results from raw data, but also reduces the monetary cost of computing in the cloud layer. Figure 2 illustrates the architecture of an IoT system which uses fog computing.

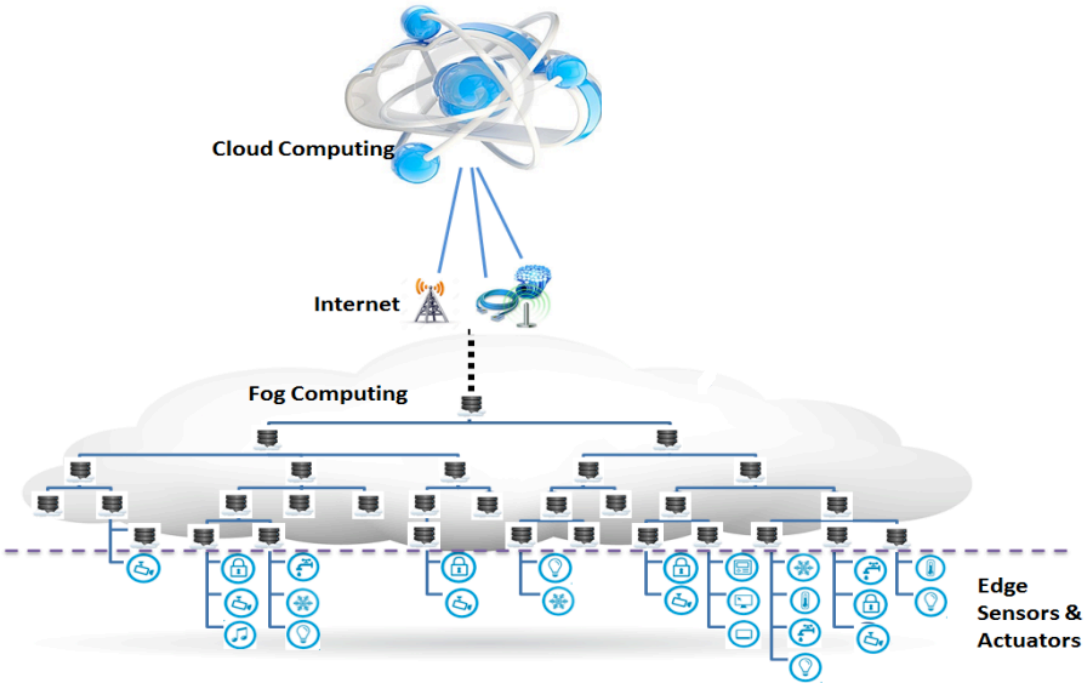

Figure 2: Hierarchical Fog Computing Architecture in IoT systems.

3.2. Fog Computing Architecture

In a systematic view, the fog computing is composed of distributed and heterogeneous resources that are deployed based on a hierarchal model. At the edge of the network, fog computing is extended and distributed from the network's gateways and routers to intelligent access points or smart phones that communicate directly with the edge devices. In this model, fog nodes deploy a virtualized and hierarchical topology and provide a distributed computing platform. Figure 3 illustrates a physical fog node that is composed of several virtual fog nodes based on the structure introduced by [10]. Each physical node is composed of computing and storage components and has interfaces for communication with neighboring fog nodes at the same, one step higher, or one step lower level of hierarchy. A virtual fog node is also composed of computing, storage and communication components and provides a multi-layered and hierarchical structure as well as collaborative distributed computing. The virtualized topology of fog computing supports multi-tenancy of different applications and processes and enables fog computing to provide seamless computing services for different applications within each local fog node. Moreover, a hierarchical architecture that is composed of several physical and virtual fog nodes forms the fog computing platform. In this paradigm, the virtual fog nodes are defined as software agents that are composed of a virtual machine with the ability of running independently in different physical nodes. Figure 4 illustrates a hierarchical architecture of physical fog Computing nodes in different levels [10].

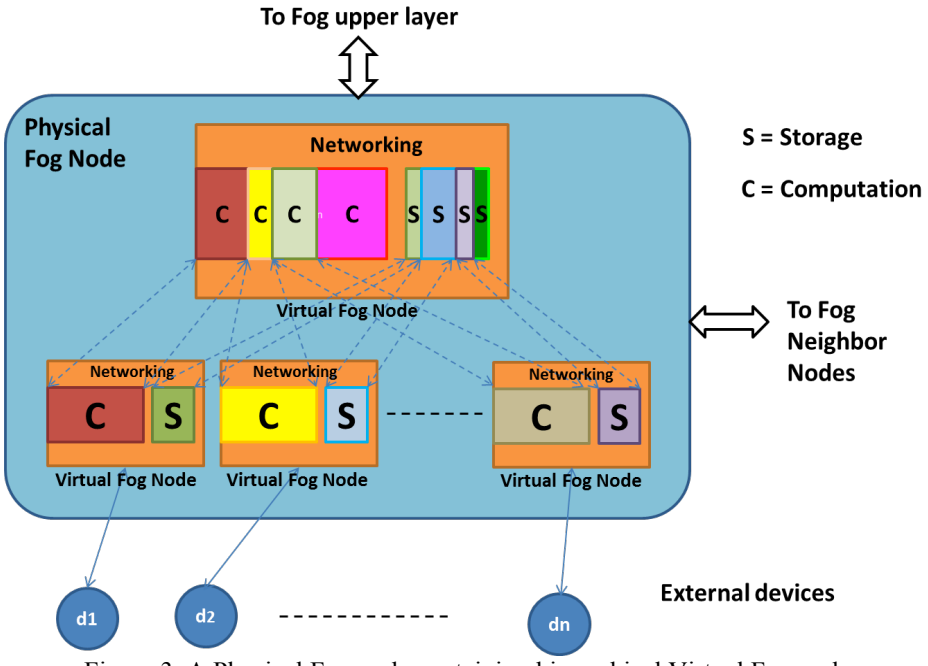

Figure 3. A Physical Fog node containing hierarchical Virtual Fog node.

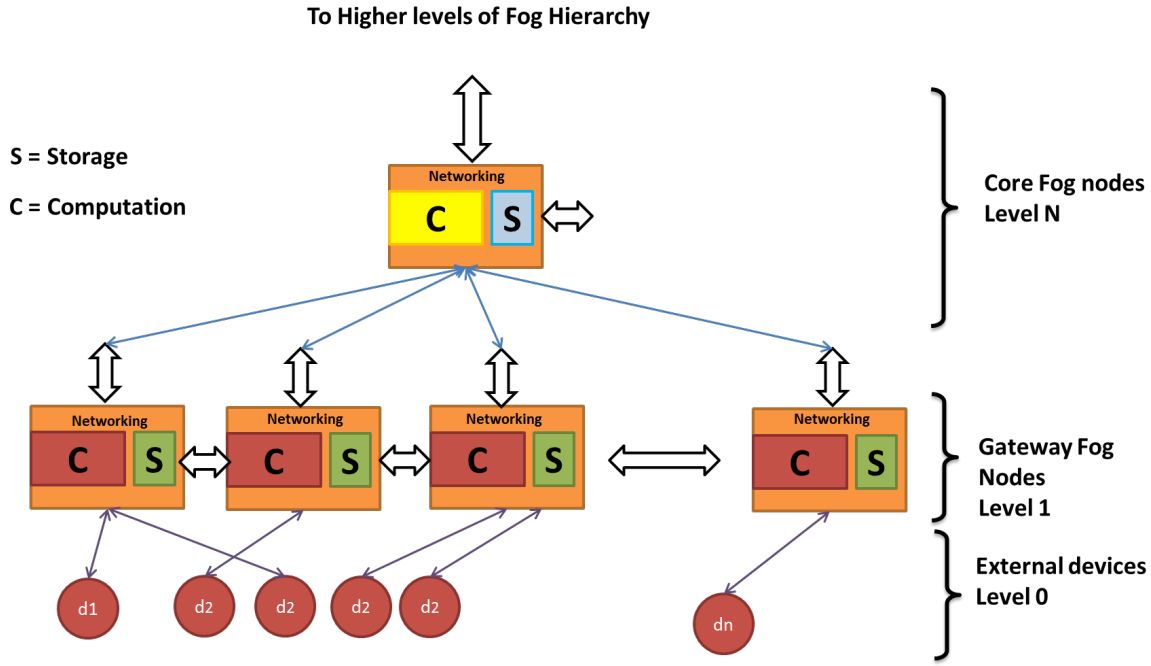

Figure 4. Hierarchical architecture of Fog Computing.

4. Smart Data

Management of Big Data is one of the most important issues in emerging IoT technologies. The massive amount of data generated through M2M communication will require new data management solutions [11]. In fact, at a low level of a large IoT system, the data generated by sensors has raw and passive nature with a high volume, velocity and variety. Conventional methods for managing

(computing and storing) such data do not provide sufficient solutions. Hence, new and efficient technologies are required to ease the management efforts and to reduce computational and network overheads. Typically, cloud computing is recognized as a scalable and robust approach for processing and storage of data in IoT systems. Pre-processing of raw sensory data is one of most efficient ways to reduce the load of Big Data from the cloud. To this end, a fog computing platform at the edge of the network is introduced to reduce the processing load from the cloud by delegating some simple and frequent tasks from the cloud to the fog (pre-processing).

In order to reduce the management efforts, computing load and communication overhead imposed by Big Data, and to improve efficiency of Big Data computing in general, we approach the problem from a different perspective. We aim to reshape the existing raw, passive and unstructured form of data in IoT to an intelligent and active form, while preserving and enhancing many other important attributes such as energy efficiency, scalability, throughput, quality of service, and privacy and security. We refer to our new data structure as "Smart Data". Smart Data are first generated by sensors in a very basic form and then they evolve through their lifecycle by being processed and converted to meaningful information as well as complemented and amended with new features such as security and privacy. A fog computing platform is the main enabler for implementing our Smart Data concept.

4.1. Structure of Smart Data

In general, a Smart Data element is a cell of encapsulated data consisting of a payload, metadata, and a virtual machine. Smart Data is inspired by the Active Bundle (AB) technology introduced by Ben Othmane in 2009 [12] for protection of sensitive data: "An active bundle or AB is a software construct, which bundles together the following three components: (1) sensitive data (2) metadata, which contain information describing sensitive data and prescribing its use; they can include a privacy policy for the sensitive data (which control the access to sensitive data or their portions), as well as the rules for AB dissemination; and (3) virtual machine (VM), which controls and manages how its AB behaves, thus making the AB active; the essential task of the VM is enforcement of the privacy policy specified by metadata". The Smart Data has a similar structure to the active bundles. However, our Smart Data, rather than only protecting the data within the bundle, is an intelligent unit that is able to evolve and participate in the operation of an IoT application. Generally, the Smart Data is a standalone unit that through the resources provided by the underlying hierarchical fog computing platform undergoes a series of pre-processing steps, evolving by getting more attributes, such as security and privacy aspects, and involved rules. IoT sensors generate a basic and lightweight version of Smart Data. It evolves (grows) when it travels through the hierarchical fog computing system towards the cloud, merging with other cells. The process is the opposite when data moves from the cloud towards the actuators, i.e., data are transformed stepwise into a distributed set of elementary cells. Figure 5 illustrates the general structure of a Smart Data cell. The Smart Data is composed of three main parts: payload data, metadata and virtual machine. In IoT based scenarios, where data is generated and transferred continuously in a resource-constrained environment, communication activities can consume a considerable amount of energy. In our Smart Data, data generated by each sensor are encapsulated into Smart Data bundles and are communicated to their gateways in specific intervals. The main objective of encapsulating a set of data already at the sensor level, instead of constantly sending discrete data, is to reduce the communication overheads in a very resource constrained environment as well as to reduce the data velocity in the Big Data context. The payload component of the Smart Data undergoes a series of processing or pre-processing steps and is thereby converted into more meaningful information. Processing or pre-processing of data includes different operations such as aggregation, filtering, compression, and encryption.

The metadata part of Smart Data contains key information such as the source of data (sensors), destination of data, the physical entity which data belongs to, timestamps, current status and logs as well as rules for accessing, fusing or diffusing, and processing data, for example. In addition, the metadata part stores information extracted by processing the payload data. Such information obtains more accurate values when the data is processed and aggregated with other data from different sensors or the same sensor over a longer period of time.

The virtual machine part, in turn, acts as a platform that enables and manages the execution of the rules specified in the metadata part. The VM at the very beginning stage contains only basic application codes. Then, it evolves by adding other code modules of the application when they are

needed. Each code module provides specific functionalities and services to the Smart Data. The modular structure of the VM component makes Smart Data extendable, allowing it to manage the overhead of carrying the code by removing unnecessary code modules and adding the required modules only when they are needed. The code modules are installed into a Smart Data cell as plugins. To enable this, we consider a remote code repository node that contains all necessary code modules as plugins. The code repository is located in the core of the network and transmits the code modules upon requests coming from any Smart Data cells. Whenever a Smart Data cell requires a specific code module, it communicates with the code repository node and requests for the required code module. To minimize the communication involved in downloading the plugins, the most recently downloaded plugins are also cached in the physical fog nodes for some period of time. So, if the requested code module does not exist in the local fog node, it will be downloaded from the remote code repository node.

To clarify the idea, let us consider a simple room temperature control system using IoT sensors and actuators as an example. Each sensor senses the temperature within its designated area with a frequency of one measurement every 2 seconds. It is obvious that continuously transmitting every single sensor reading will cause high network traffic. The Smart Data reduces this overhead by encapsulating a series of raw sensor readings and storing the data into a Smart Data bundle and transmitting it in specific time intervals. A basic form of Smart Data is built in each sensor in each interval and is transmitted to the gateways. The data at this stage is raw and have a very basic structure, but it is pre-processed with operations such as filtering, aggregation and compression during its lifecycle. The metadata part of the Smart Data includes information that describes the data. In this example, the average temperature in each sensing interval is calculated and stored in the metadata already at the sensors. Once the Smart Data is transmitted to the gateways, it is aggregated with the previously transmitted Smart Data and hence, an average temperature over a longer period will be stored in the metadata. Moreover, the Smart Data also can be aggregated with other Smart Data collected from other sensors, for example temperature sensors of other rooms, and the average temperature of the whole house in specific time slots as well as the average temperature of the whole house in a longer period could be calculated and stored in the metadata.

In order to avoid the overhead of integrating the whole application code to each Smart Data, at early stage Smart Data includes only basic application code that provides it some basic functionalities such as communication and lightweight encryption. A new module of the application code integrates to the base code whenever there is a need for a new functionality. Each module contains a set of program codes that provides a certain service or accomplishes a certain operation on the data. For example, there could be an aggregation module, an encryption module, and compression modules.

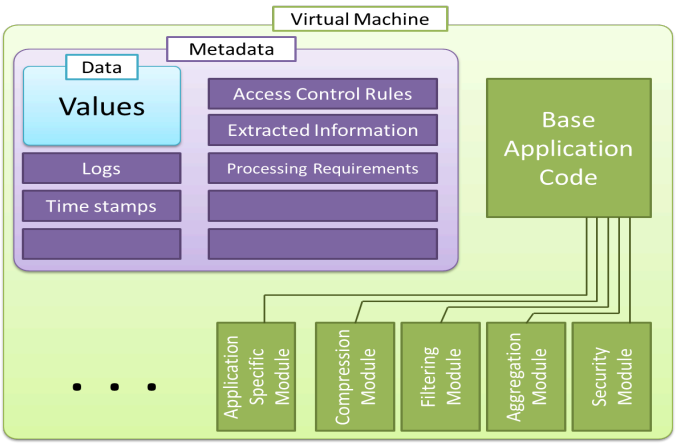

Figure 5. Structure of Smart Data

4.2. Smart Data Lifecycle

A lifecycle of Smart Data is a time span in which a data cell is generated, processed, stored, used and destroyed. Figure 6 illustrates the Smart Data life cycle. The Smart Data technology facilitates

efficient monitoring and controlling of data during its lifecycle. The idea is that throughout its lifecycle data is traceable, linkable and accountable. This is made possible by logging the events that Smart Data has undergone within the integrated metadata part of a data cell. From the data flow perspective (Figure 7) a basic form of Smart Data is first generated in sensors. At this point, Smart Data is considered immature. Immature Smart Data contains a semi-structured and encapsulated set of raw data in its payload. The data is stored in the payload of the Smart Data in a very basic format already in the sensors. This semi-structured data needs to be re-formatted and structured according to the application's context to make complex analysis and access possible for the application. Furthermore, some basic information, such as the source of data, destination of data, and time stamps are specified in the metadata part of each smart data cell already in the sensors. The payload and metadata of the Smart Data are encrypted using lightweight symmetric ciphers such as Tiny Encryption Algorithm (TEA), PRESENT Cipher and HIGHT Cipher [13] in the sensors to ensure their integrity and confidentiality during inter-node transfers.

Once the Smart Data has been transmitted from the sensors to their corresponding gateways, they are decrypted and then aggregated with other Smart Data collected from the neighbor sensors. The objective of the data aggregation is to combine the Smart Data from different sensors of an IoT application and transform them into a single Smart Data cell that is smaller in size. Then, the aggregated Smart Data is encrypted and access control policies for its payload are set in its metadata at this point. For this purpose, the Smart Data cell communicates with the code repository unit to fetch the required code modules and rules for the access control and aggregation. The data aggregation in Smart Data is done in an intelligent way. Instead of the gateways performing the aggregation, Smart Data cells have the ability to aggregate with each other using the pre-specified rules which are set in their metadata parts and required intelligence provided by the code modules downloaded from the code repository.

The gateways in IoT systems are usually resource unconstrained devices and responsible for managing the sensors and their data. In the case of Smart Data, the gateways provide computing and storage resources as well as communication interfaces to the Smart Data. In the gateways, after the aggregation stronger encryption mechanisms such as Advanced Encryption Standard (AES) could be applied to Smart Data cells. At this point, Smart Data becomes semi-mature, having aggregated data and developed its metadata to a higher level of complexity.

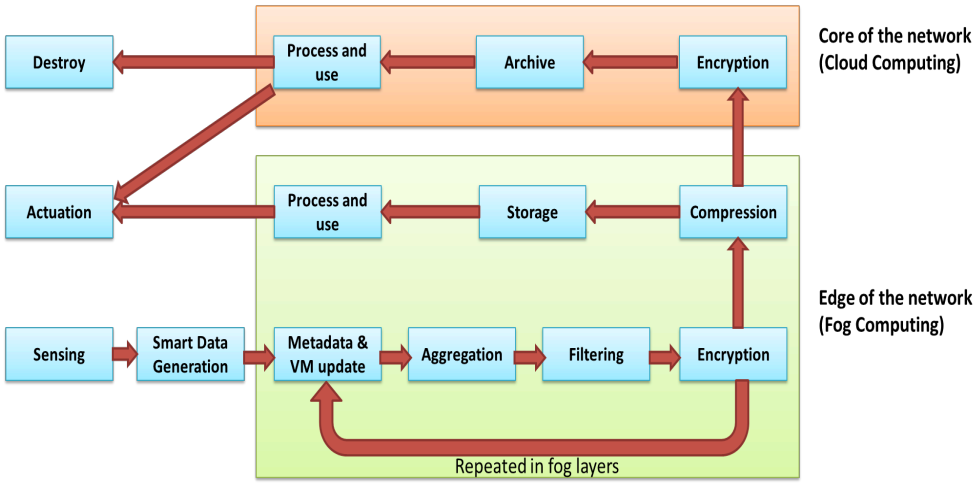

Figure 6: Smart Data Life Cycle

The semi-mature Smart Data undergoes then a series of further pre-processing tasks, such as filtering and compression through the next higher levels of the fog Computing hierarchy and becomes mature. Furthermore, the Smart Data might be aggregated with other Smart Data collected from the neighbor networks updating their metadata and VM parts if required. Filtering involves elimination of noise from the data. Noise refers to useless or unwanted records in the data that degrades the quality of data. Spurious readings, measurement errors, and background data are three main causes of noise in the data [14]. Filtering makes further analysis and processing of the data easier and more accurate. In some

applications, in order to significantly decrease the amount of data and to preserve the system's energy efficiency, the filtering process eliminates normal features of monitored parameters and only reports any abnormal trend of those parameters. Such filtering rules are also set within the metadata part of the Smart Data. For example, in the case of a video streaming application whose aim is to detect the motions in the environment, filtering the unwanted part of a video stream that contains no movements and keeping and processing only the part that contains movements of an object will significantly reduce the size of the data that needs to be processed. This will also considerably affect the network traffic. Clustering techniques provide an efficient way to filter the data.

In addition to combining the Smart Data collected from different sensors or from a same sensor in different time periods, the aggregation process also involves transforming data from different sources to a single summative data. Such summative data will have a smaller size compared to the sum of the original data before the aggregation that results in lower network traffic and improved performance in processing the data. The aggregation of sensory data will also reduce the variety of Big Data in Smart Data. Depending on the application, filtering and aggregation might be applied on the smart data in the fog Computing network hierarchy until the Smart Data reaches a desired level of quality. So, defining an optimal fog computing cluster in which the Smart Data is processed with the least energy consumption and time, and with the best values for the five Vs of the data, is a challenging research problem in designing the Smart Data.

The mature Smart Data will have significantly less volume and velocity and higher value and veracity. During its lifecycle, semi-mature or mature smart data would be compressed and stored in the local memory systems provided by the fog nodes, so that it would be swiftly available for local applications and processes. On the other hand, mature data from each local fog node in multiple geographical areas is sent to the cloud for global processing and stored for long-term purposes. Also, specific rules for destroying Smart Data can be set in the metadata part to destroy the payload part in the case of detected security breaches, or if a given validity period expires. Table 1, briefly compares immature, semi-mature and mature Smart Data.

Table 1: Immature, Semi-mature and Mature Smart Data

	Payload	Metadata	VM
Immature	- semi-structured (basic format) - include noise - lightweight encyption	- basic information about the payload - logs - lightweight encryption	- base application code only
Semi-mature	- structured - reletively reduced size - still include noise - partially aggregiated	- developed metadata - information extracted from the payload - logs - stronger encryption - aggregation, access control, and dissemination rule	- base application cod + - partially code integration (access control, filtering, aggregation code modules)
Mature	- structured - significant information - reduced size - noise are eliminated - fully aggregiated	- developed metadata - meaningful information extracted from the payload - logs	- base application code + - full code integration

The data flows in this paradigm are not only unidirectional (from sensors to the cloud) but, once the Smart Data has been processed it could be returned to the edge of the network performing a task through actuators.

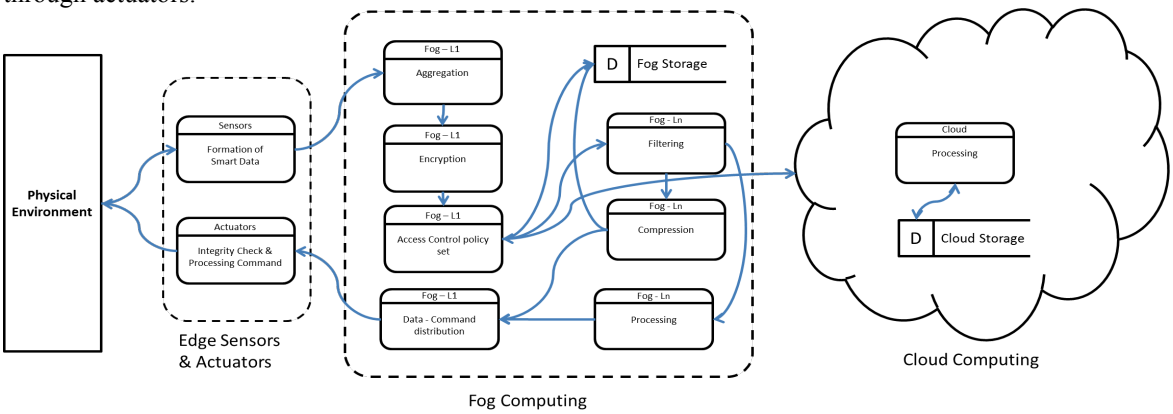

Figure 7: Data Flow Diagram for Smart Data in an IoT based system.

4.3. Proposed System Architecture

In our model, utilizing the virtualized architecture of fog computing, we integrate Smart Data as virtual fog computing nodes in an existing fog computing platform (Figure 3&4). In this model, we bring cog computing not only to the nearest point (gateways) to the edge of the network (sensors) but also the sensor nodes themselves are involved and are organic parts of fog computing. Smart Data is generated by sensors and sent to the gateways managing the sensor nodes. Then it is (pre-)processed and stored in the local fog networks. The sensor nodes have built-in communication capability and are able to communicate with their neighbors. This has two main advantages. First, in case a sensor node has lost its connection with a gateway, it can redirect the data to another node in its vicinity that has connectivity with the gateway. This is especially crucial and effective for mobile sensors. Second, the sensor nodes can collaborate with each other for very swift decision-making that does not require heavy processing.

The sensors are the lowest layer of our fog computing hierarchy (Figure 4). Since the sensor nodes in our model are not primarily meant for actual data processing, we consider the sensors as the level zero (L_0) of the computing hierarchy. Once Smart Data is generated by the sensor nodes, a set of encapsulated data is sent to the gateways within a particular time frame. The gateways in this model can also be mobile. The nearest available gateway which receives Smart Data directly from a set of sensors is considered the current managing fog node, or the gateway fog node, for these underlying sensors. Moreover, as it is the nearest point to the edge sensors, it is considered the level one (L_1) fog node for this set of the sensor nodes. Each fog node is capable of communicating with other fog nodes in its neighborhood.

A gateway fog node has the primary responsibility for managing Smart Data received from its underlying reporting sensors, initiating and managing fog computing-based processing and storage for this data and also forwarding actuation commands to specific actuators within its area. The gateway fog nodes, once having received Smart Data from multiple sensors, supervise aggregation of this data and assign required computing and memory resources for the involved processing. They are also responsible for setting access control policies for smart data according to predetermined rules. Stronger encryption of data is also applied at this point. Depending on the underlying IoT service, the gateway nodes also establish distributed local storage within the local fog computing platform or (pre-)process the contents of the Smart Data in the fog computing platform. Evolution of Smart Data involves procedures such as aggregation, filtering, compression, encryption, and access control.

The fog computing platform in our model is hierarchical, which enables stepwise evolution of Smart Data at each level of hierarchy. At the level one (L_1) of this hierarchy, the gateway nodes are connected directly to the sensor nodes (L_0) from one side and to their upper fog level (L_2) from the other side. The nodes at each level are able to communicate with each other (typically with their neighbors at the same

level) and accomplish distributed tasks. Once the tasks have been completed the nodes pass data to their parent node at the next higher level of hierarchy. This stepwise process is continued until data reaches a desired degree of maturity at the highest hierarchy level of the fog computing platform. Consequently, the architectural model is scalable and extendable both vertically and horizontally. The identification of fog nodes is done according to their level and position within a specific level. For example F_{nm} identifies a fog node at the level "n" and the position "m" within the level "n".

Generally, the tasks at the lower levels of hierarchy are more detailed and concern relatively small amounts of data, while at the higher levels of hierarchy more general tasks on larger amounts of data are carried out. The data collected from the lower levels is aggregated at the higher levels. Moreover, a higher hierarchy level deals with data from a larger geographical region than a lower hierarchy level. From the real-time performance point of view, task allocation is done in such a way that processes with low/tight latency requirements are executed at lower hierarchy levels, closer to users. This enables real-time tasks to execute and deliver results very fast, improving user experience. Such processes have some specific characteristics. First, they typically use data related to fewer sensors (corresponding to smaller geographical areas), and therefore a relatively small amount of data is involved. Second, generally at lower levels, a larger number of individual computing devices are involved in processing compared to higher levels. Tasks with no or loose latency requirements can be executed at higher levels of hierarchy. Smart Data processed in the fog computing platform becomes eventually mature and refined and is stored either in the distributed fog storage or sent to the cloud utilizing high performance communication technologies such as 3G and broadband connections. In case a local decision is made within the fog computing platform, the resulting commands will be sent to corresponding actuators through the involved gateways, without disturbing the cloud. Therefore, by pre-processing data locally in the fog computing platform, the workload of coud Computing can be significantly reduced.

5. Conclusion

With wide application of IoT/IoE technologies, new techniques will be required to deal with Big Data. In the Big Data paradigm, the data generated by sensors has a passive and raw form that requires a significant amount of processing and storage resources. Consequently, these new technologies are facing challenges in regard to management, storage and processing of Big Data. In this paper, by reshaping the current raw and passive structure of data into an intelligent and active form, we introduced the Smart Data concept. This concept combined with the fog computing technology will have potential to revolutionize the current perspective of data in IoT and will open many potential research opportunities to tackle emerging Big Data issues. Smart Data takes advantage of hierarchical and virtualized model of fog computing and provides better means for pre-processing Big Data originated from IoT sensors. As the next phase of this project, our purpose is to implement the Smart Data model in a realistic scenario and to design and develop efficient solutions for relevant operations such as filtering, aggregation, compression and encryption based on the proposed Smart Data concept.

References

[1] "Public-Private Partnership (PPP) for Big Data," no. October. European Commission, 2014.
[2] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, "Addressing big data issues in Scientific Data Infrastructure," *Proc. 2013 Int. Conf. Collab. Technol. Syst. CTS 2013*, pp. 48–55, 2013.
[3] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence*, vol. 546, N. Bessis and C. Dobre, Eds. Cham: Springer International Publishing, 2014, pp. 169–186.
[4] I. W. Jang, H. G. Nam, M. W. Pyeon, and K. J. Jeong, "New Trend Analysis of Spatial Information using Big Data Technology," *JNIT (Journal Next Gener. Inf. Technol.*, vol. 4, no. 8, pp. 1–6, 2013.
[5] C. Doukas and I. Maglogiannis, "Enabling data protection through PKI encryption in IoT m-Health devices," *... (BIBE), 2012 IEEE ...*, no. November, pp. 11–13, 2012.

[6] X. Chen, D. Berry, and W. Grimson, "Identity management to support access control in e-health systems," ... *Int. Fed. Med. ...*, pp. 880–886, 2009.
[7] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," *2014 IEEE World Forum Internet Things*, pp. 375–376, Mar. 2014.
[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things Characterization of Fog Computing," pp. 13–15, 2012.
[9] V. K. Sehgal, A. Patrick, A. Soni, and L. Rajput, "Smart Human Security Framework Using Internet of Things, Cloud and Fog Computing," vol. 321, pp. 251–263, 2015.
[10] M. Nemirovsky, "Fog Computing," Barcelona Supercomputing Center, 2012.
[11] H. Park, I. Y. Yeo, H. Jang, and S. Noh, "Simulation based Analysis on Big Data Service Bottleneck for Data Center," *JNIT (Journal Next Gener. Inf. Technol.*, vol. 4, no. 8, pp. 185–189, 2013.
[12] L. Ben Othmane and L. Lilien, "Protecting Privacy of Sensitive Data Dissemination Using Active Bundles," *2009 World Congr. Privacy, Secur. Trust Manag. E-bus.*, pp. 202–213, Aug. 2009.
[13] T. Bhattasali, "LICRYPT : Lightweight Cryptography Technique for Securing Smart Objects in Internet of Things Environment," no. May, pp. 26–28, 2013.
[14] S. N. Kashwan K. R., "Gradient Based Bilateral Filtering in Wavelet Domain for Removing Rician Noise," *JDCTA Int. J. Digit. Content Technol. its Appl.*, vol. 10, no. 2, pp. 61–77, 2016.